ЭКСПЕРИМЕНТАЛЬНАЯ ЭКОНОМИКА И ФИНАНСЫ

№3
2024

СОДЕРЖАНИЕ

МАТЕМАТИЧЕСКИЕ, СТАТИСТИЧЕСКИЕ И ИНСТРУМЕНТАЛЬНЫЕ МЕТОДЫ В ЭКОНОМИКЕ

Булгаков А.Л., Урядников А.М. Оптимизация вычислений при определении параметров модели машинного обучения на языке python с использованием библиотеки dask. 3

Гумеров Р.К., Булгакова Т.Н. Автоматизированный сбор и систематизация цен акций компаний в информационном секторе 19

Гумеров Р.К., Булгакова Т.Н. Анализ и прогнозирование цен на перелеты с помощью инструментов машинного обучения 24

Чижикова В.А., Булгаков А.Л. Выявление оптимальных факторов для прогнозирования демографических показателей с помощью методов машинного обучения. 34

*Демидов А.Д., Алешина А.В., Милютин М.А.*Сравнительный анализ моделей прогнозирования кредитоспособности 53

ФИНАНСЫ

Гревцев М.Э. Формирование инвестиционного портфеля частным инвестором в современной России 71

РЕГИОНАЛЬНАЯ И ОТРАСЛЕВАЯ ЭКОНОМИКА

Кадетов А.В. Тенденции развития зарубежного гражданского авиастроения в современных условиях 80

НАУЧНЫЙ ЖУРНАЛ

Основан в июне 2021 года

Выходит 4 раза в год

Зарегистрирован Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций (РОСКОМНАДЗОР).

Рег. ПИ № ФС77-81413 от 30 июня 2021 года

ISSN 2782-3644

Учредители:

ООО «Издательство «КноРус»

ООО «Институт экспериментальной экономики и финансов МГУ имени М.В. Ломоносова»

Адрес редакции:

Россия, 117218, Москва,
ул. Кедрова, д. 14, корп. 2
Многоканальный телефон/факс: +7 (495) 741-46-28

Сайт: *www.eeaf.ru*

Почта: *welcome@eeaf.ru*

Отпечатано в типографии ООО «Русайнс», 117218, Москва, ул. Кедрова, д. 14, корп. 2

Тираж 300 экз. Формат А4. Подписано в печать: 30.09.2024
Цена свободная

Все материалы, публикуемые в журнале, подлежат внутреннему и внешнему рецензированию

Издание не подлежит маркировке согласно п. 2 ст. 1 Федерального закона от 29.12.2010 № 436-ФЗ «О защите детей от информации, причиняющей вред их здоровью и развитию»

К СВЕДЕНИЮ АВТОРОВ

Уважаемые коллеги! Обращаем ваше внимание на то, что материалы статей проходят обязательную экспертизу. После экспертизы статьи поступают в Редакцию журнала, где проходят редакторскую и корректорскую правку. Редакция оставляет за собой право сокращать объем статей и редактировать их в соответствии с требованиями научного журнала. Рукописи статей не возвращаются; с авторами в переписку Редакция не вступает; гонорар авторам не выплачивается.

ПЕРЕЧЕНЬ ДОКУМЕНТОВ И ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ СТАТЕЙ И СОПУТСТВУЮЩИХ МАТЕРИАЛОВ, НЕОБХОДИМЫХ ДЛЯ ПУБЛИКАЦИИ

Текст статьи, при оформлении которого необходимо соблюсти следующие требования: объем статьи - до 60 тыс. знаков (1,5 авт. листа); в статье должна быть следующая информация: ФИО автора(ов) полностью, место работы (учебы), контактная информация (телефон, E-mail); аннотация и ключевые слова к статье, список литературы (на русском и английском языках).

TABLE OF CONTENTS

MATHEMATICAL, STATISTICAL INSTRUMENTAL METHODS IN ECONOMICS

Bulgakov A.L., Uryadnikov A.M. Optimization of calculations when determining the parameters of a machine learning model in python using the dask library. 3

Gumerov R.K., Bulgakova T.N. Automated collection and systematization of stock prices of companies in the information sector 19

Chizhikova V.A., Bulgakov A.L. Analysis and forecasting of flight prices using machine learning tools 24

Gorbas D.A., Bulgakov A.L. Identification of optimal factors for forecasting demographic indicators using machine learning methods 34

Demidov A.D., Aleshina A.V., Milyutin M.A. Comparative analysis of creditworthiness forecasting models 53

FINANCE

Grevtsev M.E. Formation of an investment portfolio by a private investor in modern Russia 71

REGIONAL AND INDUSTRIAL ECONOMICS

Kadetov A.V. Trends in the development of foreign civil aircraft industry in modern conditions 80

SCIENTIFIC JOURNAL

Founded in June 2021
Published 4 times a year

Registered by the Federal Service for Supervision of Communications, Information Technology and Mass Media (ROSKOMNADZOR).

Reg. PI No. FS77-81413 dated June 30, 2021

ISSN 2782-3644

Founders:

Knorus Publishing House LLC, Institute of Experimental Economics and Finance of the Lomonosov Moscow State University LLC

EDITORIAL OFFICE:

Russia, 117218, Moscow, Kedrova St., 14, bldg. 2
Multi-channel phone/fax: +7 (495) 741-46-28

Website: *www.eeaf.ru*

Mail: *welcome@eeaf.ru*

CHIEF EDITOR
Anna Valentinovna Aleshina

EDITORIAL TEAM
Andrey Leonidovich Bulgakov

Printed at the printing house LLC Rusyns, 117218, Moscow, st. Kedrova, d.14, building 2

Circulation 300 copies. A4 format. Signed to print: 30/09/2024

Free price

All materials published in the journal are subject to internal and external review.

The publication is not subject to labeling in accordance with paragraph 2 of Art. 1 of the Federal Law of December 29, 2010 No. 436-FZ "On the Protection of Children from Information Harmful to Their Health and Development"

Оптимизация вычислений при определении параметров модели машинного обучения на языке python с использованием библиотеки dask

Урядников Анатолий Максимович
студент РЭУ им. Г. В. Плеханова
E-mail: u.a-98@yandex.ru

Булгаков Андрей Леонидович
к. э. н., доцент МГУ имени М. В. Ломоносова;
Доцент РЭУ им. Г. В. Плеханова
Профессор Московского института современного академического образования
E-mail: z3900207@mail.ru

В работе рассматриваеттся создание нового логического формата данных на python, полученного благодаря объединению разноформатных датасетов с открытыми статистическими данными на сайте ООН. Представлено решение задачи оптимизации вычислений для нахождения наиболее подходящих параметров модели машинного обучения. Сравнивается скорость обработки данных с помощью библиотеки pandas с распределенной обработкой данных, реализованной с помощью библиотеки dask. В дальнейшем возможно применение алгоритмов машинного обучения для прогнозирования различных показателей, в том числе популяция населения стран, количество умышленных убийств и других преступлений, расходы на здравоохранение, платежные балансы, и т. д.

Ключевые слова: логический формат данных, распределенная обработка данных, библиотека dask, популяция стран, расходами на здравоохранение, машинное обучение, искусственный интеллект, умышленные убийства и другие преступления, нейронные сети.

Введение

Возможность объединения разнородных данных позволяет более эффективно строить модели машинного обучения за счет большего числа параметров для обучения нейронной сети. Однако загрузка большого массива данных требует ресурсов и времени, что замедляет исследовательскую работу и уменьшает целесообразность данного подхода. Распределенная обработка данных значительно ускоряет процесс загрузки и обработки данных, что позволяет максимально эффективно использовать ресурсы для анализа статистических показателей и создать параметроемкие и эффективные для машинного обучения логические форматы данных.

Библиотека Dask, как эффективный инструмент для работы с данными

В рамках научного исследования и прогнозирования статистических показателей, библиотека Dask в Python проявляет себя, как инструмент для параллельных вычислений и масштабируемой работы с объемными данными. Эта библиотека обеспечивает эффективную обработку датасетов, размер которых превышает возможности одноузловой памяти, путем распределения вычислительных задач между множеством процессоров или различными вычислительными узлами. Dask расширяет функциональность стандартных структур данных, таких как DataFrame в Pandas и масси-

вы в NumPy, позволяя обрабатывать данные значительно большего объема.

Ключевые аспекты Dask в контексте повышения эффективности исследовательской работы включают:

1. **Ленивые вычисления.** Библиотека разрабатывает вычислительные графы, которые используются только при необходимости, что способствует оптимизации процессов обработки данных.
2. **Распределенная обработка данных.** Dask обладает способностью распределять вычислительные задачи по множеству ядер одного компьютера или через сеть кластеров.
3. **Совместимость с существующими библиотеками.** Производится интеграция с ведущими библиотеками Python, включая Pandas, NumPy и Scikit-Learn, обеспечивая беспрепятственное применение привычных инструментов в контексте больших данных.
4. **Гибкость.** Dask предоставляет возможности настройки и расширения для обработки различных типов данных, что делает его пригодным для широкого спектра задач в области анализа и обработки данных.
5. **Интерактивность.** Поддержка интерактивной работы с данными делает Dask удобным инструментом для исследовательского анализа данных и машинного обучения.

Таким образом, Dask является ценным инструментом для аналитиков и инженеров по обработке данных, а также научных исследователей, работающих с большими объемами информации.

Имея различные числовые показатели (индикаторы) стран за различные временные промежутки, можно построить прогнозную модель ИИ (искусственного интеллекта), для исследования факторов, влияющих на потенциально значимые для точного прогнозирования индикаторы, такие как популяция, продолжительность жизни, финансовые и социальные показатели. Для решения данной задачи был разработан логический формат файла, объединяющий некоторые количественные характеристики стран, взятые из открытых источников международных организаций. Был написан скрипт на языке Python, реализующий данный логический формат, а также проведено сравнение скорости работы программы стандартным методом, и с помощью библиотеки dask.

Импорт библиотек

Для начала работы с функциями и расширенными возможностями, которые не входят в стандартную библиотеку языка Python, необходимо импортировать дополнительные модули. Далее следует код, подключающий необходимые дополнительный библиотеки для решения задачи.

```
#Библиотека для обработки и анализа данных. Высокоуровневые структуры данных и операции для эффективного анализа
#pandas – https://pandas.pydata.org
import pandas as pd

#Встроенный модуль для чтения и записи файлов формата «CSV»
# csv – https://docs.python.org/3/library/csv.html
import csv

#Встроенный модуль для работы с регулярными выражениями
#re – https://docs.python.org/3/library/re.html
import re

#Встроенный модуль, для работы с системным временем
#time – https://docs.python.org/3/library/time.html
import time

# расширенный анализ DataFrame, позволяющий экспортировать анализ данных в различные форматы, такие как html и json
# ydata_profiling – https://ydata-profiling.ydata.ai/docs/master/pages/getting_started/overview.html
```

```
!pip install ydata-profiling
from ydata_profiling import
ProfileReport

#библиотека для параллельных вы-
числений и масштабирования работы
с данными, позволяющая эффективно
работать с большими датасетами.
#dask поддерживает структуры данных,
аналогичные тем, что есть в Pandas
(например, DataFrame) и NumPy (на-
пример, массивы).
#dask — https://www.dask.org/
import dask.dataframe as dd
```

```
Requirement already satisfied:
ydata-profiling in /usr/local/lib/
python3.10/dist-packages (4.6.4)

Requirement already satisfied:
scipy<1.12,>=1.4.1 in /usr/local/lib/
python3.10/dist-packages (from ydata-
profiling) (1.11.4)

Requirement already satisfied:
pandas!=1.4.0,<3,>1.1 in /usr/local/
lib/python3.10/dist-packages (from
ydata-profiling) (1.5.3)

Requirement already satisfied:
matplotlib<3.9,>=3.2 in /usr/local/
lib/python3.10/dist-packages (from
ydata-profiling) (3.7.1)

Requirement already satisfied:
pydantic>=2 in /usr/local/lib/
python3.10/dist-packages (from ydata-
profiling) (2.5.3)

Requirement already satisfied:
PyYAML<6.1,>=5.0.0 in /usr/local/lib/
python3.10/dist-packages (from ydata-
profiling) (6.0.1)

Requirement already satisfied:
jinja2<3.2,>=2.11.1 in /usr/local/lib/
python3.10/dist-packages (from ydata-
profiling) (3.1.2)

Requirement already satisfied:
visions[type_image_path]==0.7.5 in /
usr/local/lib/python3.10/dist-packages
(from ydata-profiling) (0.7.5)

Requirement already satisfied:
numpy<1.26,>=1.16.0 in /usr/local/lib/
python3.10/dist-packages (from ydata-
profiling) (1.23.5)

Requirement already satisfied:
htmlmin==0.1.12 in /usr/local/lib/
python3.10/dist-packages (from ydata-
profiling) (0.1.12)

Requirement already satisfied:
phik<0.13,>=0.11.1 in /usr/local/lib/
python3.10/dist-packages (from ydata-
profiling) (0.12.4)

Requirement already satisfied:
requests<3,>=2.24.0 in /usr/local/lib/
python3.10/dist-packages (from ydata-
profiling) (2.31.0)

Requirement already satisfied:
tqdm<5,>=4.48.2 in /usr/local/lib/
python3.10/dist-packages (from ydata-
profiling) (4.66.1)

Requirement already satisfied:
seaborn<0.13,>=0.10.1 in /usr/local/
lib/python3.10/dist-packages (from
ydata-profiling) (0.12.2)

Requirement already satisfied:
multimethod<2,>=1.4 in /usr/local/lib/
python3.10/dist-packages (from ydata-
profiling) (1.10)

Requirement already satisfied:
statsmodels<1,>=0.13.2 in /usr/local/
lib/python3.10/dist-packages (from
ydata-profiling) (0.14.1)

Requirement already satisfied:
typeguard<5,>=4.1.2 in /usr/local/lib/
python3.10/dist-packages (from ydata-
profiling) (4.1.5)

Requirement already satisfied:
imagehash==4.3.1 in /usr/local/lib/
python3.10/dist-packages (from ydata-
profiling) (4.3.1)

Requirement already satisfied:
wordcloud>=1.9.1 in /usr/local/lib/
python3.10/dist-packages (from ydata-
profiling) (1.9.3)

Requirement already satisfied:
dacite>=1.8 in /usr/local/lib/
python3.10/dist-packages (from ydata-
profiling) (1.8.1)

Requirement already satisfied:
numba<0.59.0,>=0.56.0 in /usr/local/
lib/python3.10/dist-packages (from
ydata-profiling) (0.58.1)

Requirement already satisfied:
PyWavelets in /usr/local/lib/
python3.10/dist-packages (from
imagehash==4.3.1->ydata-profiling)
(1.5.0)

Requirement already satisfied: pillow
in /usr/local/lib/python3.10/dist-
packages (from imagehash==4.3.1-
>ydata-profiling) (9.4.0)

Requirement already satisfied:
attrs>=19.3.0 in /usr/local/lib/
python3.10/dist-packages (from
```

```
visions[type_image_path]==0.7.5-
>ydata-profiling) (23.2.0)

Requirement already satisfied:
networkx>=2.4 in /usr/local/lib/
python3.10/dist-packages (from
visions[type_image_path]==0.7.5-
>ydata-profiling) (3.2.1)

Requirement already satisfied:
tangled-up-in-unicode>=0.0.4 in /usr/
local/lib/python3.10/dist-packages
(from visions[type_image_path]==0.7.5-
>ydata-profiling) (0.2.0)

Requirement already satisfied:
MarkupSafe>=2.0 in /usr/local/lib/
python3.10/dist-packages (from
jinja2<3.2,>=2.11.1->ydata-profiling)
(2.1.3)

Requirement already satisfied:
contourpy>=1.0.1 in /usr/local/
lib/python3.10/dist-packages (from
matplotlib<3.9,>=3.2->ydata-profiling)
(1.2.0)

Requirement already satisfied:
cycler>=0.10 in /usr/local/lib/
python3.10/dist-packages (from
matplotlib<3.9,>=3.2->ydata-profiling)
(0.12.1)

Requirement already satisfied:
fonttools>=4.22.0 in /usr/local/
lib/python3.10/dist-packages (from
matplotlib<3.9,>=3.2->ydata-profiling)
(4.47.0)

Requirement already satisfied:
kiwisolver>=1.0.1 in /usr/local/
lib/python3.10/dist-packages (from
matplotlib<3.9,>=3.2->ydata-profiling)
(1.4.5)

Requirement already satisfied:
packaging>=20.0 in /usr/local/lib/
python3.10/dist-packages (from
matplotlib<3.9,>=3.2->ydata-profiling)
(23.2)

Requirement already satisfied:
pyparsing>=2.3.1 in /usr/local/
lib/python3.10/dist-packages (from
matplotlib<3.9,>=3.2->ydata-profiling)
(3.1.1)

Requirement already satisfied:
python-dateutil>=2.7 in /usr/local/
lib/python3.10/dist-packages (from
matplotlib<3.9,>=3.2->ydata-profiling)
(2.8.2)

Requirement already satisfied:
llvmlite<0.42,>=0.41.0dev0 in /usr/
local/lib/python3.10/dist-packages
(from numba<0.59.0,>=0.56.0->ydata-
profiling) (0.41.1)

Requirement already satisfied:
pytz>=2020.1 in /usr/local/lib/
python3.10/dist-packages (from
pandas!=1.4.0,<3,>1.1->ydata-
profiling) (2023.3.post1)

Requirement already satisfied:
joblib>=0.14.1 in /usr/local/lib/
python3.10/dist-packages (from
phik<0.13,>=0.11.1->ydata-profiling)
(1.3.2)

Requirement already satisfied:
annotated-types>=0.4.0 in /usr/local/
lib/python3.10/dist-packages (from
pydantic>=2->ydata-profiling) (0.6.0)

Requirement already satisfied:
pydantic-core==2.14.6 in /usr/local/
lib/python3.10/dist-packages (from
pydantic>=2->ydata-profiling) (2.14.6)

Requirement already satisfied: typing-
extensions>=4.6.1 in /usr/local/
lib/python3.10/dist-packages (from
pydantic>=2->ydata-profiling) (4.9.0)

Requirement already satisfied:
charset-normalizer<4,>=2 in /usr/
local/lib/python3.10/dist-packages
(from requests<3,>=2.24.0->ydata-
profiling) (3.3.2)

Requirement already satisfied:
idna<4,>=2.5 in /usr/local/lib/
python3.10/dist-packages (from
requests<3,>=2.24.0->ydata-profiling)
(3.6)

Requirement already satisfied:
urllib3<3,>=1.21.1 in /usr/local/
lib/python3.10/dist-packages (from
requests<3,>=2.24.0->ydata-profiling)
(2.0.7)

Requirement already satisfied:
certifi>=2017.4.17 in /usr/local/
lib/python3.10/dist-packages (from
requests<3,>=2.24.0->ydata-profiling)
(2023.11.17)

Requirement already satisfied:
patsy>=0.5.4 in /usr/local/lib/
python3.10/dist-packages (from
statsmodels<1,>=0.13.2->ydata-
profiling) (0.5.6)

Requirement already satisfied:
six in /usr/local/lib/python3.10/
dist-packages (from patsy>=0.5.4-
>statsmodels<1,>=0.13.2->ydata-
profiling) (1.16.0)
```

Импорт сырых данных

В качестве источников использовались данные с сайта ООН. К сожалению, данные не структурированы, что делает необходимым их дальнейшее преобразование к общему виду.

Были загружены следующие данные:

- Население страны «Population, total» http://data.un.org/_Docs/SYB/CSV/SYB65_1_202209_Population,%20Surface%20Area%20and%20Density.csv
- Данные о преднамеренных убийствах другихпреступленияхhttp://data.un.org/_Docs/SYB/CSV/SYB65_328_202209_Intentional%20homicides%20and%20other%20crimes.csv
- Расходы на здравоохранение «Health expenditure» http://data.un.org/_Docs/SYB/CSV/SYB65_325_202209_Expenditure%20on%20health.csv
- Данные об использовании интернета в странах http://data.un.org/_Docs/SYB/CSV/SYB65_314_202209_Internet%20Usage.csv

На сайте ООН http://data.un.org/ данные доЮ ступны для загрузки в двух форматах: pdf и csv. Данные в pdf сведены в таблицу (см. таблицу 1) и разбиты по страницам.

Таблица 1. Данные по населению, территории и плотности расселения населения с сайта ООН.

2 Population, surface area and density

Population, superficie et densité

Country or area Pays ou zone	Year Année	Mid-year population estimates and projections (millions) Estimations et projections de population au milieu de l'année (millions) Total	Male Hommes	Females Femmes	Sex Ratio (males per 100 females) Rapport des sexes (hommes pour 100 femmes)	Population age distribution (percentage) Répartition par âge de la population (pourcentage) Aged 0 to 14 years old âgée de 0 à 14 ans	Aged 60+ years old âgée de 60 ans ou plus	Population density (per km²) Densité de population (pour km²)	Surface area Superficie (000 km²)
Total, all countries or areas	**2010**	**6 985.60**	**3 514.41**	**3 471.20**	**101.2**	**27.1**	**11.1**	**53.6**	...
	2015	**7 426.60**	**3 737.40**	**3 689.19**	**101.3**	**26.4**	**12.3**	**57.0**	...
Total, tous pays ou zones	**2020**	**7 840.95**	**3 943.61**	**3 897.34**	**101.2**	**25.7**	**13.5**	**60.1**	**130 094**
	2022[1]	**7 975.11**	**4 008.58**	**3 966.53**	**101.1**	**25.3**	**13.9**	**61.2**	...
Africa	2010	1 055.23	525.87	529.37	99.3	41.5	5.0	35.7	...
Afrique	2015	1 201.11	599.30	601.81	99.6	41.3	5.2	40.6	...
	2020	1 360.68	679.36	681.32	99.7	40.5	5.4	46.0	29 648
	2022[1]	1 426.74	712.43	714.31	99.7	40.1	5.5	48.3	...
Northern Africa	2010	207.11	104.59	102.52	102.0	32.4	6.6	26.9	...
Afrique	2015	228.36	115.31	113.05	102.0	32.8	7.3	29.7	...
septentrionale	2020	251.42	126.86	124.55	101.9	32.8	8.2	32.7	7 769
	2022[1]	259.97	131.12	128.85	101.8	32.5	8.5	33.8	...
Sub-Saharan Africa	2010	848.12	421.27	426.85	98.7	43.7	4.6	38.8	...
Afrique	2015	972.75	483.99	488.76	99.0	43.2	4.7	44.5	...
subsaharienne	2020	1 109.26	552.50	556.76	99.2	42.3	4.8	50.7	21 879[2]
	2022[1]	1 166.77	581.31	585.46	99.3	41.8	4.8	53.4	...
Eastern Africa	2010	342.74	169.60	173.15	97.9	45.0	4.1	51.5	...
Afrique orientale	2015	393.35	194.84	198.51	98.2	43.8	4.3	59.1	...
	2020	449.29	222.75	226.54	98.3	42.1	4.5	67.5	6 667
	2022[1]	473.00	234.54	238.47	98.4	41.4	4.5	71.1	...
Middle Africa	2010	133.61	66.32	67.29	98.6	45.6	4.5	20.6	...
Afrique centrale	2015	157.40	78.22	79.18	98.8	45.7	4.4	24.3	...
	2020	184.57	91.79	92.78	98.9	45.5	4.4	28.4	6 497
	2022[1]	196.08	97.51	98.57	98.9	45.4	4.4	30.2	...
Southern Africa	2010	59.10	28.50	30.60	93.1	29.7	7.3	22.2	...
Afrique australe	2015	63.72	30.99	32.72	94.7	29.2	7.8	24.0	...
	2020	67.27	32.74	34.53	94.8	29.5	8.3	25.3	2 651
	2022[1]	68.60	33.44	35.16	95.1	29.3	8.4	25.8	...
Western Africa	2010	312.67	156.86[3]	155.80[3]	100.7[3]	44.1[3]	4.7[3]	51.6	...
Afrique occidentale	2015	358.28	179.94[3]	178.35[3]	100.9[3]	44.1[3]	4.6[3]	59.1	...
	2020	408.12	205.22[3]	202.91[3]	101.1[3]	43.2[3]	4.7[3]	67.3	6 064
	2022[1]	429.08	215.82[3]	213.26[3]	101.2[3]	42.7[3]	4.7[3]	70.8	...
Americas [2]	2010	935.82	461.93	473.89	97.5	24.7	13.1	24.1	...
Amériques [2]	2015	983.54	485.81	497.73	97.6	23.2	14.7	25.4	...
	2020	1 025.79	506.71	519.09	97.6	21.8	16.5	26.4	38 791
	2022[1]	1 037.14	511.80	525.34	97.4	21.2	17.2	26.7	...
Northern America	2010	345.27	170.17[4]	175.10[4]	97.2[4]	19.5[4]	18.4[4]	18.5	...
Amérique	2015	360.46	178.29[4]	182.18[4]	97.9[4]	19.0[4]	20.5[4]	19.3	...
septentrionale	2020	373.96	185.39[4]	188.56[4]	98.3[4]	18.2[4]	22.8[4]	20.0	18 652
	2022[1]	376.87	186.67[4]	190.20[4]	98.1[4]	17.7[4]	23.7[4]	20.2	...
Latin America & the Caribbean	2010	590.55	291.77	298.78	97.7	27.7	10.0	29.1	...
	2015	623.08	307.53	315.55	97.5	25.6	11.3	30.7	...
Amérique latine et	2020	651.84	321.31	330.52	97.2	23.9	12.9	32.2	20 139
Caraïbes	2022[1]	660.27	325.13	335.14	97.0	23.2	13.4	32.6	...
Caribbean	2010	41.40	20.57[5]	20.84[5]	98.7[5]	26.3[5]	11.9[5]	186.9	...
Caraïbes	2015	42.75	21.21[5]	21.55[5]	98.4[5]	24.8[5]	13.1[5]	193.0	...
	2020	43.96	21.76[5]	22.20[5]	98.0[5]	23.6[5]	14.5[5]	198.5	226
	2022[1]	44.39	21.94[5]	22.45[5]	97.8[5]	23.2[5]	15.1[5]	200.4	...
Central America	2010	156.06	76.68	79.39	96.6	31.3	8.4	63.3	...
Amérique centrale	2015	167.19	82.15	85.04	96.6	28.9	9.5	67.8	...
	2020	176.34	86.54	89.81	96.4	26.6	10.9	71.6	2 452
	2022[1]	179.06	87.73	91.33	96.1	25.8	11.3	72.7	...

Данные в формате csv содержат 1 строку-заголовок, после чего идут через запятую. Стоит отметить, что некоторые данные из файла в формате PDF не соответствуют данным из файла csv.

```
Region/Country/Area,,Year,Series,Value,Footnotes,Source
1,"Total, all countries or areas",2000,Percentage of individuals using the internet,5.3,,"International Telecommunication Union (ITU), Geneva, the ITU
database, last accessed March 2022."
1,"Total, all countries or areas",2005,Percentage of individuals using the internet,15.8,Estimate.,"International Telecommunication Union (ITU), Geneva, the
ITU database, last accessed March 2022."
1,"Total, all countries or areas",2010,Percentage of individuals using the internet,28.9,Estimate.,"International Telecommunication Union (ITU), Geneva, the
ITU database, last accessed March 2022."
1,"Total, all countries or areas",2015,Percentage of individuals using the internet,40.5,Estimate.,"International Telecommunication Union (ITU), Geneva, the
ITU database, last accessed March 2022."
1,"Total, all countries or areas",2018,Percentage of individuals using the internet,49.1,Estimate.,"International Telecommunication Union (ITU), Geneva, the
ITU database, last accessed March 2022."
1,"Total, all countries or areas",2019,Percentage of individuals using the internet,53.6,Estimate.,"International Telecommunication Union (ITU), Geneva, the
ITU database, last accessed March 2022."
1,"Total, all countries or areas",2020,Percentage of individuals using the internet,59.1,Estimate.,"International Telecommunication Union (ITU), Geneva, the
ITU database, last accessed March 2022."
15,Northern Africa,2000,Percentage of individuals using the internet,0.6,,"International Telecommunication Union (ITU), Geneva, the ITU database, last accessed
March 2022."
15,Northern Africa,2005,Percentage of individuals using the internet,9.7,,"International Telecommunication Union (ITU), Geneva, the ITU database, last accessed
March 2022."
15,Northern Africa,2010,Percentage of individuals using the internet,22.8,,"International Telecommunication Union (ITU), Geneva, the ITU database, last
accessed March 2022."
15,Northern Africa,2015,Percentage of individuals using the internet,41.4,,"International Telecommunication Union (ITU), Geneva, the ITU database, last
accessed March 2022."
```

Следующий код демонстрирует процесс импорта (загрузки) данных в программу:

Импорт датасета с популяцией (Population, surface area and density)

```
dataPopulation=pd.read_
csv('http://data.un.org/_Docs/SYB/
CSV/SYB65_1_202209_Population,%20
Surface%20Area%20and%20Density.csv',
header = 1, thousands=',')
```

Импорт датасета с умышленными убийствами и другими преступлениями (Intentional homicides and other crimes)

```
dataCrime=pd.read_
csv('http://data.un.org/_Docs/SYB/
CSV/SYB65_328_202209_Intentional%20
homicides%20and%20other%20crimes.
csv', header = 1, thousands=',')
```

Импорт датасета с расходами на здравоохранение (Health expenditure)

```
dataHealth=pd.read_
csv('http://data.un.org/_Docs/SYB/
CSV/SYB65_325_202209_Expenditure%20
on%20health.csv', header = 1,
thousands=',')
```

Импорт датасета с данными об использовании интернета в странах (Internet usage)

```
dataInternet=pd.read_
csv('http://data.un.org/_Docs/
SYB/CSV/SYB65_314_202209_
Internet%20Usage.csv', header = 1,
thousands=',')
```

Создание новой логической структуры данных

Чтобы продемонстрировать принцип получения нового логического формата данных, рассмотрим классическую реализацию алгоритмов обработки данных посредством библиотеки pandas.

Настройка датасета с популяцией (Population, surface area and density)

```
dataPopulationProcessing = dataPo
pulation[dataPopulation[«Series»]
== «Population mid-year estimates
(millions)»]
dataPopulationProcessing = data
PopulationProcessing[dataPopul
ationProcessing[«Series»].str.
contains(«(millions)», flags=re.I)]

dataPopulationProcessing = data
PopulationProcessing[[«Region/
Country/Area», «Unnamed: 1», «Year»,
«Value»]]
dataPopulationProcessing.columns
= ['Id', 'Region', 'Year',
'Population']
```

Обработанный датасет выглядит следующим образом:

```
dataPopulationProcessing
```

	Id	Region	Year	Population
0	1	Total, all countries or areas	2010	6985.60
7	1	Total, all countries or areas	2015	7426.60
15	1	Total, all countries or areas	2020	7840.95
23	1	Total, all countries or areas	2022	7975.11
30	2	Africa	2010	1055.23
…	…	…	…	…
7836	894	Zambia	2022	20.02
7843	716	Zimbabwe	2010	12.84
7850	716	Zimbabwe	2015	14.15
7858	716	Zimbabwe	2020	15.67
7866	716	Zimbabwe	2022	16.32

1050 rows × 4 columns

Настройка датасета с умышленными убийствами (Intentional homicides)

```
dataCrimeProcessing =
dataCrime[dataCrime[«Series»] ==
«Intentional homicide rates per
100,000»]
dataCrimeProcessing =
dataCrimeProcessing[[«Region/
Country/Area», «Unnamed: 1», «Year»,
«Value»]]
dataCrimeProcessing.columns = ['Id',
'Region', 'Year', 'Intentional
homicide rates per 100,000']
```

Обработанный датасет выглядит следующим образом:

```
dataCrimeProcessing
```

	Id	Region	Year	Intentional homicide rates per 100,000
0	1	Total, all countries or areas	2005	6.3
3	1	Total, all countries or areas	2010	6.1
6	1	Total, all countries or areas	2015	5.9
9	1	Total, all countries or areas	2020	5.6
12	2	Africa	2005	12.5
…	…	…	…	…
4852	894	Zambia	2010	6.0
4853	894	Zambia	2015	5.4
4854	716	Zimbabwe	2005	11.2
4863	716	Zimbabwe	2010	5.6
4864	716	Zimbabwe	2012	7.5

927 rows × 4 columns

Настройка датасета с расходами на здравоохранение (Health expenditure)

```
dataHealthProcessing =
dataHealth[[«Region/Country/Area»,
«Unnamed: 1», «Year»,»Series»,
«Value»]]

dataHealthProcessing.columns = ['Id',
'Region', 'Year', 'Series','Value']

dfs = [dataHealthProcessing[dataHea
lthProcessing['Series'] == 'Current
health expenditure (% of GDP)'].set_
index(['Id','Region', 'Year']),

     dataHealthProcessing[data
HealthProcessing['Series'] ==
'Domestic general government
health expenditure (% of total
government expenditure)'].set_
index(['Id','Region', 'Year'])]

dataHealthProcessing = pd.concat(dfs,
axis=1).reset_index()

dataHealthProcessing.columns = ['Id',
'Region', 'Year', 'Series','Current
health expenditure (% of GDP)',
'Series','Domestic general
government health expenditure (% of
total government expenditure)']

dataHealthProcessing = dataHealthP
rocessing[['Id','Region', 'Year',
'Current health expenditure (% of
GDP)','Domestic general government
health expenditure (% of total
government expenditure)']]
```

Обработанный датасет выглядит следующим образом:

```
dataHealthProcessing
```

	Id	Region	Year	Current health expenditure (% of GDP)	Domestic general government health expenditure (% of total government expenditure)
0	4	Afghanistan	2005	9.9	3.4
1	4	Afghanistan	2010	8.6	2.3
2	4	Afghanistan	2017	12.6	2.3
3	4	Afghanistan	2018	14.1	1.9
4	4	Afghanistan	2019	13.2	3.9
…	…	…	…	…	…
1126	894	Zambia	2019	5.3	7.0
1127	716	Zimbabwe	2010	10.5	15.2
1128	716	Zimbabwe	2017	7.5	6.4
1129	716	Zimbabwe	2018	8.7	8.7
1130	716	Zimbabwe	2019	7.7	8.7

1131 rows × 5 columns

Настройка датасета с потреблением интернета (Internet usage)

```
dataInternetProcessing = dataInternet[dataInternet[«Series»] == «Percentage of individuals using the internet»]
dataInternetProcessing = dataInternetProcessing[[«Region/Country/Area», «Unnamed: 1», «Year», «Value»]]
dataInternetProcessing.columns = ['Id', 'Region', 'Year', 'Percentage of individuals using the internet']
```

Обработанный датасет выглядит следующим образом:

```
dataInternetProcessing
```

	Id	Region	Year	Percentage of individuals using the internet
0	1	Total, all countries or areas	2000	5.3
1	1	Total, all countries or areas	2005	15.8
2	1	Total, all countries or areas	2010	28.9
3	1	Total, all countries or areas	2015	40.5
4	1	Total, all countries or areas	2018	49.1
…	…	…	…	…
1458	199	LDC§	2010	3.1
1459	199	LDC§	2015	10.8
1460	199	LDC§	2018	20.0
1461	199	LDC§	2019	22.5
1462	199	LDC§	2020	24.6

1463 rows × 4 columns

Поскольку данные имеют разнородный характер, их необходимо привести к общему виду. В следующей функции происходит объединение общих столбцов из обработанных датасетов.

```
#Объединение таблиц
#Объединение строк с образованием в столбцы по стране и году
dfs = [dataPopulationProcessing.set_index(['Id','Region', 'Year']),
     dataCrimeProcessing.set_index(['Id','Region', 'Year']),
     dataHealthProcessing.set_index(['Id','Region', 'Year']),
     dataInternetProcessing.set_index(['Id','Region', 'Year'])]
logicData = pd.concat(dfs, axis=1).reset_index()
```

После обработки датасет выглядит следующим образом:

```
logicData
```

	Id	Region	Year	Population	Intentional homicide rates per 100,000	Current health expenditure (% of GDP)	Domestic general government health expenditure (% of total government expenditure)	Percentage of individuals using the internet
0	1	Total, all countries or areas	2010	6985.60	6.1	NaN	NaN	28.9
1	1	Total, all countries or areas	2015	7426.60	5.9	NaN	NaN	40.5
2	1	Total, all countries or areas	2020	7840.95	5.6	NaN	NaN	59.1
3	1	Total, all countries or areas	2022	7975.11	NaN	NaN	NaN	NaN
4	2	Africa	2010	1055.23	12.0	NaN	NaN	NaN
…	…	…	…	…	…	…	…	…
2351	199	LDC§	2010	NaN	NaN	NaN	NaN	3.1
2352	199	LDC§	2015	NaN	NaN	NaN	NaN	10.8
2353	199	LDC§	2018	NaN	NaN	NaN	NaN	20.0
2354	199	LDC§	2019	NaN	NaN	NaN	NaN	22.5
2355	199	LDC§	2020	NaN	NaN	NaN	NaN	24.6

2356 rows × 8 columns

Для создания модели машинного обучения данные не должны содержать неопределенных значений. Поэтому важно удалить пустые значения из полученной таблицы (отображаются как NaN), чтобы они не влияли на общую статистику.

```
#Удаление нулевых значений
logicData = logicData[pd.notnull(logicData[«Intentional homicide rates per 100,000»])]
logicData = logicData[pd.notnull(logicData[«Current health expenditure (% of GDP)»])]
logicData = logicData[pd.notnull(logicData[«Domestic general government health expenditure (% of total government expenditure)»])]
logicData = logicData[pd.notnull(logicData[«Percentage of individuals using the internet»])]
logicData = logicData[pd.notnull(logicData[«Population»])]

#Проверка нулевых значений
logicData.isnull().sum()
```

```
Id                                                0
Region                                            0
Year                                              0
Population                                        0
Intentional homicide rates per 100,000            0
Current health expenditure (% of GDP)             0
Domestic general government health expenditure
```

```
   (% of total government expenditure                    0
   Percentage of individuals using the internet          0
dtype: int64
```

Получен логический формат данных, в который собраны разнородные статистические данные. Его можно представить в виде таблицы:

```
logicData
```

	Id	Region	Year	Population	Intentional homicide rates per 100,000	Current health expenditure (% of GDP)	Domestic general government health expenditure (% of total government expenditure)	Percentage of individuals using the internet
124	4	Afghanistan	2010	28.19	3.4	8.6	2.3	4.0
128	8	Albania	2010	2.91	4.3	4.7	8.1	45.0
132	12	Algeria	2010	35.86	0.7	5.1	9.5	12.5
140	20	Andorra	2010	0.07	0.0	6.6	19.7	81.0
152	28	Antigua and Barbuda	2010	0.09	6.8	5.4	13.9	47.0
…	…	…	…	…	…	…	…	…
1022	862	Venezuela (Boliv. Rep. of)	2010	28.72	46.0	6.8	8.2	37.4
1026	704	Viet Nam	2010	87.41	1.5	4.7	7.8	30.7
1038	887	Yemen	2010	24.74	4.7	5.2	3.8	12.4
1042	894	Zambia	2010	13.79	6.0	3.7	4.7	3.0
1046	716	Zimbabwe	2010	12.84	5.6	10.5	15.2	6.4

140 rows × 8 columns

Полученный логический формат был составлен из датасетов сравнительно небольшого размера. В эпоху больших данных, эффективная обработка и анализ огромных наборов данных становятся ключевыми задачами для исследователей и специалистов по данным. Используемая выше библиотека Pandas ограничена однопоточными операциями и размером памяти одной машины. Чтобы эффективно произвести вышеописанные действия с большими данными стоит использовать библиотеку Dask, которая предоставляет мощные инструменты для параллельных вычислений и обработки данных, выходящих за рамки ограничений традиционных систем.

Основные Преимущества Dask по Сравнению с Pandas в рамках вышеизложенной задачи:

1. Масштабируемость:

Pandas: Ограничен обработкой данных, которые помещаются в память одного компьютера.

Dask: Позволяет обрабатывать наборы данных, размер которых превышает объем доступной памяти, распределяя данные по кластеру машин, что позволяет использовать датасеты большого размера.

2. Параллельные Вычисления:

Pandas: Преимущественно использует одно ядро процессора, что ограничивает его производительность.

Dask: Реализует параллельные вычисления, значительно ускоряя обработку данных за счет использования всех доступных ядер процессора и даже распределенных систем, что уменьшает время на обработку данных на этапе подготовки к объединению.

3. Ленивые Вычисления:

Pandas: Выполняет операции немедленно, что может быть неэффективно для очень боль-

ших наборов данных.

Dask: Использует ленивые вычисления, позволяя строить сложные вычислительные графы и выполнять их только при необходимости, оптимизируя память и время вычислений. В нашем примере нет необходимости загружать и использовать сразу все данные из выборки, поскольку в дальнейшем происходит их частичное отсечение.

4. Гибкость и Совместимость:

Pandas: Хорошо подходит для стандартного анализа данных и имеет широкую поддержку в сообществе Python.

Dask: Поддерживает API, совместимые с Pandas и NumPy, обеспечивая плавный переход от меньших к большим данным без необходимости полностью менять рабочий процесс.

5. Интерактивность:

Pandas: Идеально подходит для интерактивной работы с данными в Jupyter Notebooks на уровне отдельного компьютера.

Dask: Расширяет эту интерактивность, позволяя работать с гораздо большими наборами данных и обеспечивая интеграцию с распределенными вычислительными ресурсами.

Для демонстрации преимущества скорости обработки данных аналогичный код был написан с помощью методов библиотеки Dask, после чего произведено сравнение времени работы программы с использованием библиотек pandas и dask.

```
#Старт замера времени работы программы, использующей библиотеку Pandas
start_time_pandas = time.time()

#Использование pandas для загрузки датасетов:
dataPopulation = pd.read_csv('http://data.un.org/_Docs/SYB/CSV/SYB65_1_202209_Population,%20Surface%20Area%20and%20Density.csv', header = 1, thousands=',')
dataCrime = pd.read_csv('http://data.un.org/_Docs/SYB/CSV/SYB65_328_202209_Intentional%20homicides%20and%20other%20crimes.csv', header = 1, thousands=',')
dataHealth = pd.read_csv('http://data.un.org/_Docs/SYB/CSV/SYB65_325_202209_Expenditure%20on%20health.csv', header = 1, thousands=',')
dataInternet = pd.read_csv('http://data.un.org/_Docs/SYB/CSV/SYB65_314_202209_Internet%20Usage.csv', header = 1, thousands=',')

#Обработка данных о населении
dataPopulationProcessing = dataPopulation[dataPopulation[«Series»] == «Population mid-year estimates (millions)»]
dataPopulationProcessing = dataPopulationProcessing[dataPopulationProcessing[«Series»].str.contains(«(millions)», flags=re.I)]
dataPopulationProcessing = dataPopulationProcessing[[«Region/Country/Area», «Unnamed: 1», «Year», «Value»]]
dataPopulationProcessing.columns = ['Id', 'Region', 'Year', 'Population']

#Обработка данных о преступлениях
dataCrimeProcessing = dataCrime[dataCrime[«Series»] == «Intentional homicide rates per 100,000»]
dataCrimeProcessing = dataCrimeProcessing[[«Region/Country/Area», «Unnamed: 1», «Year», «Value»]]
dataCrimeProcessing.columns = ['Id', 'Region', 'Year', 'Intentional homicide rates per 100,000']

#Обработка данных о здравоохранении
dataHealthProcessing = dataHealth[[«Region/Country/Area», «Unnamed: 1», «Year»,»Series», «Value»]]
```

```
dataHealthProcessing.columns = ['Id', 'Region', 'Year', 'Series','Value']

#Слияние данных о здравоохранении
dfs = [dataHealthProcessing[dataHealthProcessing['Series'] == 'Current health expenditure (% of GDP)'].set_index(['Id','Region', 'Year']),
     dataHealthProcessing[dataHealthProcessing['Series'] == 'Domestic general government health expenditure (% of total government expenditure)'].set_index(['Id','Region', 'Year'])]
dataHealthProcessing = pd.concat(dfs, axis=1).reset_index()

dataHealthProcessing.columns = ['Id', 'Region', 'Year', 'Series','Current health expenditure (% of GDP)', 'Series','Domestic general government health expenditure (% of total government expenditure)']
dataHealthProcessing = dataHealthProcessing[['Id','Region', 'Year', 'Current health expenditure (% of GDP)','Domestic general government health expenditure (% of total government expenditure)']]

#Обработка данных об использовании интернета
dataInternetProcessing = dataInternet[dataInternet[«Series»] == «Percentage of individuals using the internet»]
dataInternetProcessing = dataInternetProcessing[[«Region/Country/Area», «Unnamed: 1», «Year», «Value»]]
dataInternetProcessing.columns = ['Id', 'Region', 'Year', 'Percentage of individuals using the internet']

#Объединение таблиц
dfs = [dataPopulationProcessing.set_index(['Id','Region', 'Year']),
     dataCrimeProcessing.set_index(['Id','Region', 'Year']),
     dataHealthProcessing.set_index(['Id','Region', 'Year']),
     dataInternetProcessing.set_index(['Id','Region', 'Year'])]

logicData = pd.concat(dfs, axis=1).reset_index()

#Конец замера времени работы программы, использующей библиотеку Pandas
pandas_duration = time.time() — start_time_pandas

#Старт замера времени работы программы, использующей библиотеку Dask
start_time_dask = time.time()

#Загрузка данных
dtype_dict = {'Footnotes': 'object'}

dataPopulation = dd.read_csv('http://data.un.org/_Docs/SYB/CSV/SYB65_1_202209_Population,%20Surface%20Area%20and%20Density.csv', header=1, thousands=',', dtype=dtype_dict)
dataCrime = dd.read_csv('http://data.un.org/_Docs/SYB/CSV/SYB65_328_202209_Intentional%20homicides%20and%20other%20crimes.csv', header=1, thousands=',', dtype=dtype_dict)
dataHealth = dd.read_csv('http://data.un.org/_Docs/SYB/CSV/SYB65_325_202209_Expenditure%20on%20health.csv', header=1, thousands=',', dtype=dtype_dict)
dataInternet = dd.read_csv('http://data.un.org/_Docs/SYB/CSV/SYB65_314_202209_Internet%20Usage.csv', header=1, thousands=',', dtype=dtype_dict)

#Обработка данных о населении
dataPopulationProcessing = dataPopulation[dataPopulation[«Series»]
```

```
== «Population mid-year estimates
(millions)»]
dataPopulationProcessing = data
PopulationProcessing[dataPopul
ationProcessing[«Series»].str.
contains(«(millions)», flags=re.I,
regex=False)]
dataPopulationProcessing = data
PopulationProcessing[[«Region/
Country/Area», «Unnamed: 1», «Year»,
«Value»]]
dataPopulationProcessing.columns
= ['Id', 'Region', 'Year',
'Population']

#Обработка данных о преступлениях
dataCrimeProcessing =
dataCrime[dataCrime[«Series»] ==
«Intentional homicide rates per
100,000»]
dataCrimeProcessing =
dataCrimeProcessing[[«Region/
Country/Area», «Unnamed: 1», «Year»,
«Value»]]
dataCrimeProcessing.columns = ['Id',
'Region', 'Year', 'Intentional
Homicide Rates']

#Обработка данных о здравоохранении
dataHealthProcessing =
dataHealth[[«Region/Country/Area»,
«Unnamed: 1», «Year», «Series»,
«Value»]]
dataHealthProcessing.columns =
['Id', 'Region', 'Year', 'Series',
'Health Value']

#Слияние данных о здравоохранении
health_exp_gdp = dataHealthProcess
ing[dataHealthProcessing['Series']
== 'Current health expenditure (% of
GDP)']
health_exp_gov = dataHealthProcessin
g[dataHealthProcessing['Series'] ==
'Domestic general government health
expenditure (% of total government
expenditure)']

#Обработка данных об использовании
интернета
dataInternetProcessing = dataInt
ernet[dataInternet[«Series»] ==
«Percentage of individuals using the
internet»]
dataInternetProcessing =
dataInternetProcessing[[«Region/
Country/Area», «Unnamed: 1», «Year»,
«Value»]]
dataInternetProcessing.columns =
['Id', 'Region', 'Year', 'Internet
Usage']

#Объединение таблиц
merged_df = dataPopulationProcessing.
merge(dataCrimeProcessing, on=['Id',
'Region', 'Year'], how='outer')
merged_df = merged_df.merge(health_
exp_gdp[['Id', 'Region', 'Year',
'Health Value']], on=['Id',
'Region', 'Year'], how='outer',
suffixes=('', '_gdp'))
merged_df = merged_df.merge(health_
exp_gov[['Id', 'Region', 'Year',
'Health Value']], on=['Id',
'Region', 'Year'], how='outer',
suffixes=('', '_gov'))
merged_df = merged_df.merge(dataInte
rnetProcessing, on=['Id', 'Region',
'Year'], how='outer')

#Переименование столбцов
merged_df = merged_
df.rename(columns={
  'Health Value': 'Current health
expenditure (% of GDP)',
  'Health Value_gov': 'Domestic
general government health expenditure
(% of total government expenditure)'
})

#Удаление нулевых значений
columns_to_check = [
  'Intentional Homicide Rates',
  'Current health expenditure (% of
GDP)',
  'Domestic general government
```

```
health expenditure (% of total
government expenditure)',
  'Internet Usage',
  'Population'
]
logicData_dask = merged_
df.dropna(subset=columns_to_check)

#Конец замера времени работы про-
граммы, использующей библиотеку Dask
dask_duration = time.time() — start_
time_dask

logicData = logicData_dask.compute()

print(f»Время выполнения с Pandas:
{pandas_duration} секунд»)
print(f»Время выполнения с Dask:
{dask_duration} секунд»)

#Вычисляем абсолютную разницу
во времени
time_difference = abs(pandas_
duration — dask_duration)

#Выбираем большее время для расчета
процентной разницы
longer_duration = max(pandas_
duration, dask_duration)

#Рассчитываем разницу в процентах
percentage_difference = (time_
difference / longer_duration) * 100

#Определяем, какой метод был быстрее
if pandas_duration < dask_duration:
  faster_method = «Pandas»
else:
  faster_method = «Dask»

print(f»Разница во времени выполне-
ния: {percentage_difference}%»)
print(f»Быстрее был метод: {faster_
method}»)
<ipython-input-37-a5352a7669a8>:12:
UserWarning: This pattern is
interpreted as a regular expression,
and has match groups. To actually
get the groups, use str.extract.
dataPopulationProcessing = data
PopulationProcessing[dataPopul
ationProcessing[«Series»].str.
contains(«(millions)», flags=re.I)]
```

Время выполнения с Pandas: 13.113258123397827 секунд

Время выполнения с Dask: 5.9607253074646 секунд

Разница во времени выполнения: 54.544284483892305%

Быстрее был метод: Dask

Как видно из сравнения, даже при небольшом размере датасетов методы dask выигрывают по скорости обработке данных у методов pandas в поставленной задаче, что заметно скажется при работе с большими данными.

Заключение

Полученные логические типы данных продемонстрировали потенциал использования библиотеки dask в сравнении с библиотекой pandas. Ее использование будет полезно в исследовательских проектах, где требуется обработать большое количество данных, выделить из них параметры и выявить наиболее значимые. Данный код можно использовать и для других данных с сайта ООН, в части определения зависимости между различными индикаторами стран, а также возможна реализация прогнозной модели для индикаторов страны, зная ее ближайшие показатели.

Список источников:

1. F. Alfaro-Almagro, M. Jenkinson, N. K. Bangerter, J. L. Andersson, L. Griffanti, G. Douaud, et al., «Image processing and Quality Control for the first 10000 brain imaging datasets from UK Biobank», NeuroImage, vol. 166, pp. 400–424, 2018.
2. V. Hayot-Sasson, S. T. Brown and T. Glatard, «Performance Evaluation of Big Data Processing Strategies for Neuroimaging», 19th IEEE/ACM

International Symposium on Cluster Cloud and Grid Computing (CC-Grid), 2019.
3. Matthew Rocklin, «Dask: Parallel Computation with Blocked algorithms and Task Scheduling», Proceedings of the 14th Python in Science Conference, pp. 126–132, 2015.
4. Y. Cheng and F. Rusu, «Parallel in-situ data processing with speculative loading», SIGMOD, pp. 1287–1298, 2014.
5. Anaconda, [online] // Электронный ресурс // URL: Available: https://www.continuum.io/.
6. S. Arumugam, A. Dobra, C. Jermaine, N. Pansare, and L. Perez. The DataPath System: A Data-Centric Analytic Processing Engine for Large Data Warehouses. In SIGMOD 2010.
7. J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. Commun. ACM, 51(1), 2008
8. Announcing the Consortium for Python Data API Standards. https://data-apis.org/blog/announcing_the_consortium/.
9. Dask. // Электронный ресурс // URL: https://dask.org/.
10. D. A. Watt. Programming language design concepts. John Wiley & Sons, 2004.
11. P. Sinthong and M. J. Carey. Aframe: Extending dataframes for large-scale modern data analysis. In 2019 IEEE International Conference on Big Data (Big Data), pages 359–371. IEEE, 2019.
12. Stephan Hoyer and Joseph J. Hamman: xarray: N-d labeled arrays and datasets in python. Journal of Open Research Software, 5, apr 2017.
13. Ryan Abernathey: Step-by-Step Guide to Building a Big Data Portal.
14. Adnan, K., Akbar, R.: An analytical study of information extraction from unstructured and multidimensional big data. Journal of Big Data 6(1), 91 (2019).
15. Angelova, R., Siersdorfer, S.: A neighborhood-based approach for clustering of linked document collections. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management. p. 778–779. CIKM '06, Association for Computing Machinery (2006).
16. Bach, M., Krstic, Z., Seljan, S., Turulja, L.: Text mining for big data analysis in financial sector: A literature review. Sustainability (Switzerland) 11(5) (2019).
17. Grishin, V.: Method of analysis and search for borrowings in the text. Problems of science 7, 31 (2018)
18. Jalali, S.M.J., Park, H.W., Vanani, I.R., Pho, K.H.: Research trends on big data domain using text mining algorithms. Digital Scholarship in the Humanities (04 2020). 19.Logica, B., Magdalena, R.: Using big data in the academic environment. Procedia Economics and Finance 33, 277–286 (2015).
19. Rocklin, M.: Dask: Parallel computation with blocked algorithms and task scheduling. In: Python in Science Conference. pp. 126–132 (2015).
20. Salvador, S., Chan, P.: Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In: 16th IEEE International Conference on Tools with Artificial Intelligence. pp. 576–584 (2004).
21. Zambelli, A.: A data-driven approach to estimating the number of clusters in hierarchical clustering. F1000Research 5 (2016).
22. Elgendy, N., Elragal, A.: Big data analytics: a literature review paper. Lect. Notes Comput. Sci. 8557, 214–227 (2014).
23. Gandomi, A., Haider, M.: Beyond the hype: Big data concepts, methods, and analytics. Int. J. Inf. Manage. 35(2), 137–144 (2015)
24. Shin, D., Shim, J.: A, systematic review on data mining for mathematics and science education. Int. J. Sci. Math. Educ. 19, 639–659 (2021)
25. Данные с сайта ООН // Электронный ресурс // URL: http://data.un.org/

Distributed data processing and creation of a new logical data format using the dask library

Bulgakov A. L., Uryadnikov A. M.
Russian Economic University. G. V. Plekhanov, MISAO, Lomovosov Moscow State University

The paper considers the creation of a new logical data format in python, obtained by combining different format datasets with open statistical data on the UN website. A solution to the problem of optimizing calculations for finding the most suitable parameters of the machine learning model is presented. The speed of data processing using the pandas library is compared with distributed data processing

implemented using the dask library. In the future, it is possible to use machine learning algorithms to predict various indicators, including the population of countries, the number of intentional murders and other crimes, health care costs, balances of payments, etc.

Keywords: logical data format, distributed data processing, dask library, population of countries, health care costs, machine learning, artificial intelligence, intentional murders and other crimes, neural networks.

References

1. F. Alfaro-Almagro, M. Jenkinson, N. K. Bangerter, J. L. Andersson, L. Griffanti, G. Douaud, et al., «Image processing and Quality Control for the first 10,000 brain imaging datasets from UK Biobank,» NeuroImage, vol. 166, pp. 400–424, 2018.
2. V. Hayot-Sasson, S. T. Brown and T. Glatard, «Performance Evaluation of Big Data Processing Strategies for Neuroimaging,» 19th IEEE/ACM International Symposium on Cluster Cloud and Grid Computing (CC-Grid), 2019.
3. Matthew Rocklin, «Dask: Parallel Computation with Blocked Algorithms and Task Scheduling,» Proceedings of the 14th Python in Science Conference, pp. 126–132, 2015.
4. Y. Cheng and F. Rusu, «Parallel in-situ data processing with speculative loading,» SIGMOD, pp. 1287–1298, 2014.
5. Anaconda, [online] // Electronic resource // URL: Available: https://www.continuum.io/.
6. S. Arumugam, A. Dobra, C. Jermaine, N. Pansare, and L. Perez. The DataPath System: A Data-Centric Analytic Processing Engine for Large Data Warehouses. In SIGMOD 2010.
7. J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. Commun. ACM, 51(1), 2008
8. Announcing the Consortium for Python Data API Standards. https://data-apis.org/blog/announcing_the_consortium/.
9. Dask. // Electronic resource // URL: https://dask.org/.
10. D. A. Watt. Programming language design concepts. John Wiley & Sons, 2004.
11. P. Sinthong and M. J. Carey. Aframe: Extending dataframes for large-scale modern data analysis. In 2019 IEEE International Conference on Big Data (Big Data), pages 359–371. IEEE, 2019.
12. Stephan Hoyer and Joseph J. Hamman: xarray: N-d labeled arrays and datasets in python. Journal of Open Research Software, 5, April 2017.
13. Ryan Abernathey: Step-by-Step Guide to Building a Big Data Portal.
14. Adnan, K., Akbar, R.: An analytical study of information extraction from unstructured and multidimensional big data. Journal of Big Data 6(1), 91 (2019).
15. Angelova, R., Siersdorfer, S.: A neighborhood-based approach for clustering of linked document collections. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management. p. 778–779. CIKM '06, Association for Computing Machinery (2006).
16. Bach, M., Krstic, Z., Seljan, S., Turulja, L.: Text mining for big data analysis in the financial sector: A literature review. Sustainability (Switzerland) 11(5) (2019).
17. Grishin, V.: Method of analysis and search for borrowings in the text. Problems of science 7, 31 (2018)
18. Jalali, S.M.J., Park, H.W., Vanani, I.R., Pho, K.H.: Research trends on big data domain using text mining algorithms. Digital Scholarship in the Humanities (04 2020). 19.Logica, B., Magdalena, R.: Using big data in the academic environment. Procedia Economics and Finance 33, 277–286 (2015).
19. Rocklin, M.: Dask: Parallel computation with blocked algorithms and task scheduling. In: Python in Science Conference. pp. 126–132 (2015).
20. Salvador, S., Chan, P.: Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In: 16th IEEE International Conference on Tools with Artificial Intelligence. pp. 576–584 (2004).
21. Zambelli, A.: A data-driven approach to estimating the number of clusters in hierarchical clustering. F1000 Research 5 (2016).
22. Elgendy, N., Elragal, A.: Big data analytics: a literature review paper. Lect. Notes Comput. Sci. 8557, 214–227 (2014).
23. Gandomi, A., Haider, M.: Beyond the hype: Big data concepts, methods, and analytics. Int. J. Inf. Manage. 35(2), 137–144 (2015)
24. Shin, D., Shim, J.: A, systematic review on data mining for mathematics and science education. Int. J. Sci. Math. Educ. 19, 639–659 (2021)
25. Data from the UN website // Electronic resource // URL: http://data.un.org/

Автоматизированный сбор и систематизация цен акций компаний в информационном секторе

Гумеров Руслан Камилевич
студент РЭУ им. Г. В. Плеханова
E-mail: 625ruslan@gmail.com

Булгакова Татьяна Николаевна
аналитик, ООО Институт экспериментальной экономики и финансов МГУ имени М. В. Ломоносова,
E-mail: tate-888@yandex.ru

В статье описываются методы автоматизированного сбора финансовой информации и систематизации цен акций компаний IT сектора. В работе проанализированы финансовые данные за 2023 год, проведена предобработка данных, проведен анализ и визуализация данных. Проведено сравнение наибольших/наименьших точек цен акций, изучен объем торгуемых акций и их общая стоимость. Анализ проводился на основе финансовых данных Московской биржи. При подготовке и сборе данных для данной статьи было использовано подключение через API для сбора и получения данных сайта.

Ключевые слова: Мосбиржа, финансовые данные, анализ данных, автоматизированная обработка данных, информационные технологии.

Информационные технологии (далее по тексту — ИТ) являются одной из перспективных и динамично растущих отраслей российского рынка. ИТ является связующим звеном между производством и эффективным распределением услуг и ресурсов.

В период за 2021–2022 гг. Россия заняла 14 место в топ-20 стран по развитию цифровых технологий[1]. При этом Россия входит в топ-10 стран по научной и изобретательской активности в робототехнике, квантовым технологиям и искусственному интеллекту, что влечет за собой рост привлекательности ИТ отрасли для вложения инвестиций.

Мосбиржа является крупной торговой площадкой для торговли акциями IT компаний. Существует большое количество брокерских приложений, через которые может осуществляться торговля ценными бумагами. «Финам»[2] предлагает возможность не только торговать ценными бумагами, но и осуществляет обучение тех, кто хочет получить квалификацию. Сбер предлагает инвестировать, как через компьютерные программы (система Quik[3]), так и с помощью мобильного приложения (Сбер инвестор). Банк Тинькофф — один из первых, кто реализовал возможность инвестировать с помощью мобильного приложения. Однако, получая биржевую информацию через третье лицо (брокера), инвестор получает информацию

[1] Россия вошла в топ-20 стран по развитию цифровых технологий. / https://rospatent.gov.ru/ru/news/top-20-stran-cifrovyh-tehnologiy-18012923 (дата обращения 10.01.2024)

[2] Финам / https://www.finam.ru/landings/about-finam/

[3] Интернет-трейдинг в системе QUIK / http://www.sberbank.ru/ru/person/investments/broker_service/quik (дата обращения 10.01.2024)

о котировках с небольшим запозданием, в связи с чем может упустить прибыль. Исходя из этого, было решено провести анализ данных, полученных напрямую от Мосбиржи. Сложность анализа и обработки эти данных была связана с тем, что указанные данные представлены в неструктурированном виде, при котором их визуальное восприятие и анализ затруднительны без осуществления обработки в Python.

Была поставлена цель — получение данных с Мосбиржи (с сайта moex.ru), в случае необходимости предобработать их, провести анализ данных и визуализировать. Для анализа были взяты 3 IT компании, которые предлагают технологические решения для бизнеса: YNDX, HHRU, CIAN.

Далее были проведены следующие действия:

1. Импорт модулей/библиотек;
2. Предобработка данных;
3. Анализ и визуализация данных.

Импорт модулей/библиотек.

```
import requests
import apimoex
import pandas as pd

import matplotlib.pyplot as plt
from matplotlib.dates import DateFormatter, MonthLocator
import matplotlib.dates as mdates
```

В целях извлечения данных были необходимы следующие библиотеки:

1. request — библиотека запросов.
2. apimoex — основная библиотека для получения информации об акциях через API.
3. pandas — для работы с данными в виде датафрейма.
4. matplotlib — библиотека для визуализации.

С помощью метода get_board_history класса apimoex были запрошены цены на момент закрытия торгов акциями, количество акций и их общая ценность за 2023 год. Сохранение производится в каждый отдельный файл формата Excel.

```
shares = [ "CIAN",  "OZON", "YNDX"]
with requests.Session() as session:
    for i in shares:
        data = apimoex.get_board_history(session, i, start='2023-01-01', end='2023-12-30',
                                         columns=('TRADEDATE', 'CLOSE', 'VOLUME', 'VALUE'))

        df = pd.DataFrame(data)

        df.to_excel(r"\\{}.xlsx".format(i), index=False)
```

В случае необходимости можно изменить изучаемый период, указав в параметре start необходимую стартовую границу.

Открываем файл, содержащий информацию о ценах дневного закрытия компании OZON

```
df_ozon = pd.read_excel(r"OZON.xlsx")
```

```
df_ozon
```

	TRADEDATE	CLOSE	VOLUME	VALUE
0	2023-01-03	1438.0	156273	2.247806e+08
1	2023-01-04	1436.0	70407	1.013383e+08
2	2023-01-05	1439.0	35744	5.141899e+07
3	2023-01-06	1424.5	48303	6.877347e+07
4	2023-01-09	1429.5	134162	1.918266e+08
...	...	...	...	...
249	2023-12-25	2662.0	294956	7.850213e+08
250	2023-12-26	2661.5	126141	3.347073e+08
251	2023-12-27	2688.5	250858	6.703450e+08
252	2023-12-28	2799.0	601152	1.657029e+09
253	2023-12-29	2804.5	374667	1.052692e+09

254 rows × 4 columns

Указанный файл содержит 254 строки, в котором имеется дата торгов, цена закрытия, количество акций и их общая стоимость в разрезе дня.

```
df_ozon.isnull().sum()
```

```
TRADEDATE    0
CLOSE        0
VOLUME       0
VALUE        0
dtype: int64
```

Пропуски в файле OZON отсутствуют. Проанализируем объем акций компании OZON (см. рис. 1).

20

```
fig, ax = plt.subplots(figsize=(10, 6))
ax.bar(df_ozon['TRADEDATE'], df_ozon['VOLUME'], color='blue')
ax.set_title('Столбчатая диаграмма объема торгов акции OZON по датам')
ax.set_xlabel('Дата')
ax.set_ylabel('Объем торгов')
ax.grid(True)

# Установка интервалов оси x

plt.xticks(rotation=45)
 # Устанавливаем интервал каждый месяц
ax.xaxis.set_major_locator(mdates.MonthLocator(bymonthday=1))

plt.xticks(rotation=45)

plt.show()
```

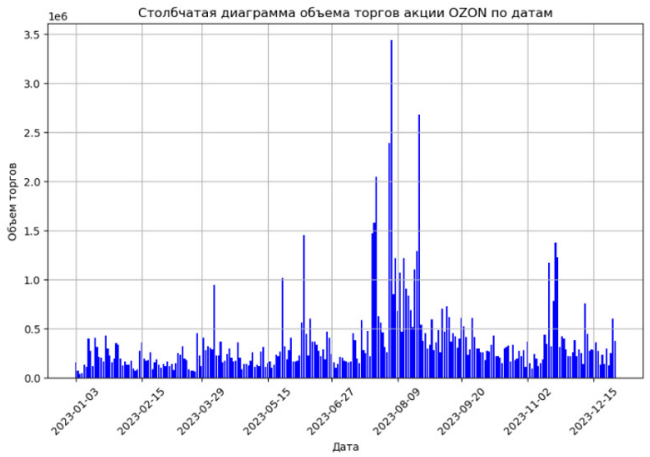

Рис. 1. Столбчатая диаграмма объема торгов акции OZON по датам [1].

Вывод наибольшее количество акций приходится на август 2023 — около 3.5 миллионов акций, наименьшее — в январе 2023. Проанализируем цены акций OZON (см. рис. 2).

```
# Построение графика
fig, ax = plt.subplots(figsize=(10, 6))
ax.plot(df_ozon['TRADEDATE'], df_ozon['CLOSE'], marker='o', linestyle='-')
ax.set_title('График цены закрытия по датам')
ax.set_xlabel('Дата')
ax.set_ylabel('Цена закрытия')
ax.grid(True)

# Установка интервалов оси x
ax.xaxis.set_major_locator(mdates.MonthLocator(bymonthday=1))
plt.xticks(rotation=45)


plt.show()
```

Рис. 2. График цены закрытия по датам акций OZON [1].

Наибольшая цена OZON составляет на декабрь 2023 — около 3000 рублей, наименьшая в начале года — 1400 рублей. Аналогично проанализируем акции компании Циан (см. рис. 3).

```
df_cian['TRADEDATE'] = pd.to_datetime(df_cian['TRADEDATE'])

# Сортировка DataFrame по дате
df_cian = df_cian.sort_values(by='TRADEDATE')

# Построение графика
fig, ax = plt.subplots(figsize=(10, 6))
ax.plot(df_cian['TRADEDATE'], df_cian['CLOSE'], marker='o', linestyle='-')
ax.set_title('График цены закрытия ЦИАН по датам')
ax.set_xlabel('Дата')
ax.set_ylabel('Цена закрытия')
ax.grid(True)

# Установка интервалов оси x
ax.xaxis.set_major_locator(mdates.MonthLocator(bymonthday=1))
plt.xticks(rotation=45)

plt.show()
```

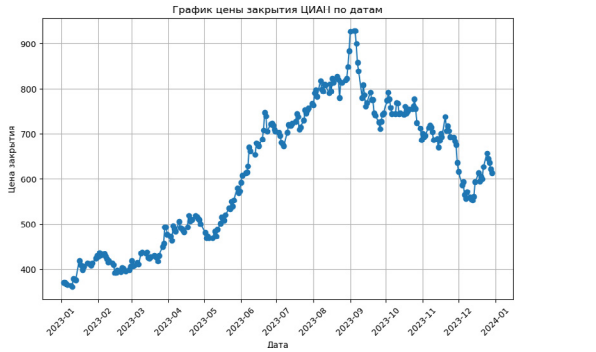

Рис. 3. График цены закрытия компании ЦИАН по датам [1].

Минимальная стоимость акций в январе 2023 — около 200, Максимальная в сентябре 2023–1000 (см. рис. 4).

```
fig, ax = plt.subplots(figsize=(10, 6))
ax.bar(df_cian['TRADEDATE'], df_cian['VOLUME'], color='blue')
ax.set_title('Столбчатая диаграмма объема торгов акции ЦИАН по датам')
ax.set_xlabel('Дата')
ax.set_ylabel('Объем торгов')
ax.grid(True)

# Установка интервалов оси x
ax.xaxis.set_major_locator(mdates.MonthLocator(bymonthday=1))
plt.xticks(rotation=45)

plt.show()
```

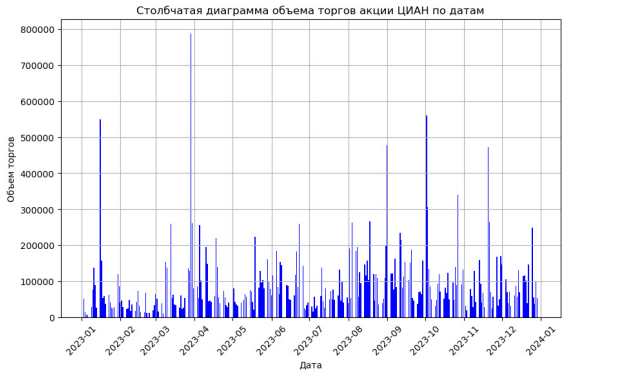

Рис. 4. Столбчатая диаграмма объема торгов акции ЦИАН по датам.

Минимальный объем торгов в январе 2023 менее 10000, в апреле 2023 около 800000. Проанализируем акции компании Яндекс (см. рис. 5).

```
fig, ax = plt.subplots(figsize=(10, 6))
ax.bar(df_yndx['TRADEDATE'], df_yndx['VOLUME'], color='blue')
ax.set_title('Столбчатая диаграмма объема торгов акции Yandex по датам')
ax.set_xlabel('Дата')
ax.set_ylabel('Объем торгов')
ax.grid(True)

# Установка интервалов оси x
ax.xaxis.set_major_locator(mdates.MonthLocator(bymonthday=1))
plt.xticks(rotation=45)

plt.show()
```

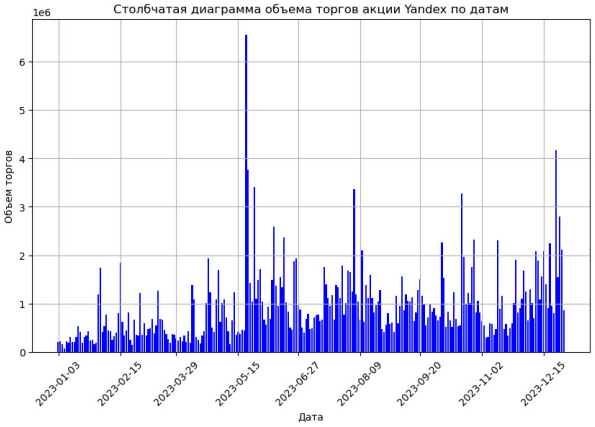

Рис. 5. Столбчатая диаграмма объема торгов акции Yandex по датам [1].

Минимальная стоимость в январе 2023–1800, максимальная в июле 2023–2700 (см. рис. 6).

```
# Построение графика
fig, ax = plt.subplots(figsize=(10, 6))
ax.plot(df_yndx['TRADEDATE'], df_yndx['CLOSE'], marker='o', linestyle='-')
ax.set_title('График цены закрытия Яндекс по датам')
ax.set_xlabel('Дата')
ax.set_ylabel('Цена закрытия')
ax.grid(True)

# Установка интервалов оси x
ax.xaxis.set_major_locator(mdates.MonthLocator(bymonthday=1))
plt.xticks(rotation=45)

plt.show()
```

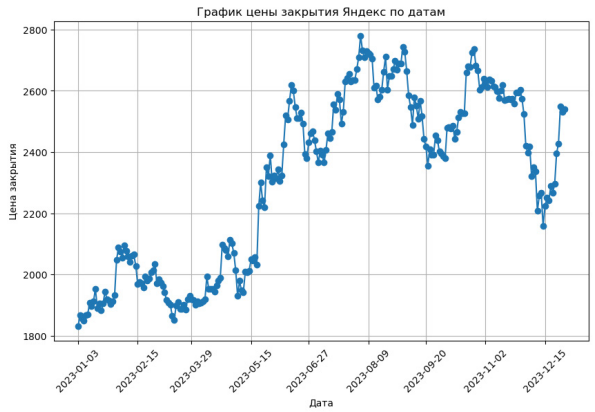

Рис. 6. Графи цены закрытия Яндекс по датам [1].

Далее рассмотрим диаграмму объема торгов акциями Yandex по датам (см. рис. 7).

```
fig, ax = plt.subplots(figsize=(10, 6))
ax.bar(df_yndx['TRADEDATE'], df_yndx['VOLUME'], color='blue')
ax.set_title('Столбчатая диаграмма объема торгов акции Yandex по датам')
ax.set_xlabel('Дата')
ax.set_ylabel('Объем торгов')
ax.grid(True)

# Установка интервалов оси x
ax.xaxis.set_major_locator(mdates.MonthLocator(bymonthday=1))
plt.xticks(rotation=45)

plt.show()
```

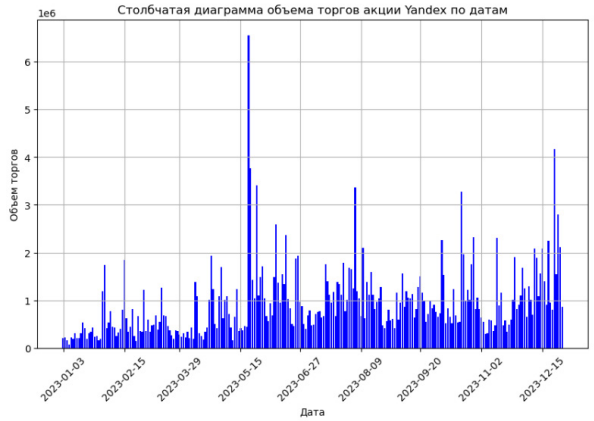

Рис. 7. Столбчатая диаграмма объема торгов акциями Yandex по датам [1].

Минимальный объем торгов составил в январе менее 10000 максимальная 6000000 (см. рис. 8).

Проведем сравнение цен закрытия акций.

```
fig, ax = plt.subplots(figsize=(10, 6))
ax.plot(combined_df['TRADEDATE'], combined_df['CLOSE'], marker='o', linestyle='-', label='CIAN')
ax.plot(combined_df['TRADEDATE'], combined_df['CLOSE_OZON'], marker='o', linestyle='-', label='OZON')
ax.plot(combined_df['TRADEDATE'], combined_df['CLOSE_YNDX'], marker='o', linestyle='-', label='YNDX')

ax.set_title('График цены закрытия по датам')
ax.set_xlabel('Дата')
ax.set_ylabel('Цена закрытия')
ax.legend()  # Добавляем легенду
ax.grid(True)

# Установка интервалов оси x
ax.xaxis.set_major_locator(mdates.MonthLocator(bymonthday=1))
plt.xticks(rotation=45)

plt.show()
```

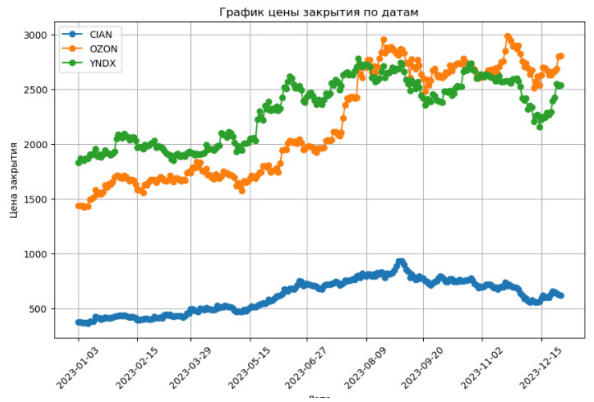

Рис. 8. График цены закрытия по датам [1].

Исходя из линейного графика мы видим, наибольшая цена была у компании OZON в декабре 2023 года, что подтверждается графиком.

Исходя из вышеизложенного, мы проанализировали акции трех IT компаний за 2023 год. Наибольшая цена на акции зафиксирована у компании OZON в начале декабря 2023 — около 3000 рублей за одну акцию, затем у компании Яндекс- около 2750 рублей за одну акцию. При этом как мы видим в графике IT компании развиваются интенсивно, цена акции каждой компании увеличилась почти в два раза в течении одного года.

Проведенные исследования показали, что для проведения анализа данных можно использоваться подключение через API для сбора и получения данных сайта Московской биржи.

Литература

1. Документация API Мосбиржи // Портал Мосбиржи // Электронный ресурс // URL: https://www.moex.com/a7939
2. Официальный сайт Мосбиржи // Портал Мосбиржи // Электронный ресурс // URL: https://www.moex.com/en
3. Россия вошла в топ-20 стран по развитию цифровых технологий // Портал ТАСС от 18 января 2023 // Электронный ресурс // URL: https://rospatent.gov.ru/ru/news/top-20-stran-cifrovyh-tehnologiy-18012923 (дата обращения 10.01.2024)
4. ФИНАМ // Электронный ресурс // URL: https://www.finam.ru/landings/about-finam/
5. Документация Pandas // Портал pandas.pydata.org // Электронный ресурс // URL: https://pandas.pydata.org/pandas-docs/stable/user_guide/merging.html
6. Документация Jupyter // Портал jupyter.org // Электронный ресурс // URL: https://docs.jupyter.org/en/latest/index.html
7. Документация Python // Портал python.org // Электронный ресурс // URL: — https://www.python.org/doc/
8. Интернет-трейдинг в системе QUIK/http://www.sberbank.ru/ru/person/investments/broker_service/quik (дата обращения 10.01.2024)

Automated collection and systematization of stock prices of companies in the information sector

Gumerov R. K., Bulgakova T. N.
Plekhanov Russian University of Economics, Institute of Experimental Economics and Finance of Moscow State University named after MV Lomonosov,

The article describes methods for the automated collection of financial information and systematization of stock prices of IT sector companies. The work analyzes financial data for 2023, pre-processes the data, analyzes and visualizes the data. A comparison of the highest/lowest points of stock prices was carried out, the volume of traded shares and their total value were studied. The analysis was carried out based on the financial data of the Moscow Exchange. When preparing and collecting data for this article, an API connection was used to collect and receive site data.

Key words: Moscow Exchange, financial data, data analysis, automated data processing, information technology.

References

Literature

1. Moscow Exchange API Documentation // Moscow Exchange Portal // Electronic resource // URL: https://www.moex.com/a7939
2. Moscow Exchange Official Website // Moscow Exchange Portal // Electronic resource // URL: https://www.moex.com/en
3. Russia Enters Top 20 Countries in Digital Technology Development // TASS Portal of January 18, 2023 // Electronic resource // URL: https://rospatent.gov.ru/ru/news/top-20-stran-cifrovyh-tehnologiy-18012923 (date of access 01/10/2024)
4. FINAM // Electronic resource // URL: https://www.finam.ru/landings/about-finam/
5. Pandas Documentation // pandas.pydata.org Portal // Electronic resource // URL: https://pandas.pydata.org/pandas-docs/stable/user_guide/merging.html
6. Jupyter Documentation // Portal jupyter.org // Electronic resource // URL: https://docs.jupyter.org/en/latest/index.html
7. Python Documentation // Portal python.org // Electronic resource // URL: — https://www.python.org/doc/
8. Internet trading in the QUIK system / http://www.sberbank.ru/ru/person/investments/broker_service/quik (date of access 10.01.2024)

Анализ и прогнозирование цен на перелеты с помощью инструментов машинного обучения

Чижикова Виктория Александровна
магистр РЭУ имени Г. В. Плеханова
E-mail: viktoria29092001@gmail.com

Булгаков Андрей Леонидович
к. э. н., доцент МГУ имени М. В. Ломоносова;
Доцент РЭУ им. Г. В. Плеханова
Профессор Московского института современного академического образования
E-mail: z3900207@mail.ru

В статье анализируется возможность использования методов машинного обучения для проведения маркетинговых исследований на рынке авиаперелетов. Определение оптимальной стоимости авиабилетов и времени их покупки может быть затруднено из-за большого количества факторов, влияющих на цены, в том числе факторов сезонности, уровня спроса, конкуренция между авиакомпаниями, и других условий. Была предпринята попытка учесть эти факторы при проведении прогнозирования цен на авиабилеты с использованием методов машинного обучения. Для прогнозирования цен на перелеты с сайта aviasales.ru был использован парсинг данных в качестве инструмента для обработки информации, которая отображается на веб-сайте. В статье приведен пример парсинга данных и исследуются полученные результаты, в том числе изучена возможность дальнейшего анализа данных и возможности визуализации данных.

Ключевые слова: туризм, большие данные, машинное обучение, парсинг, прогнозирование.

Введение

Стоимость авиабилетов может значительно варьироваться в зависимости от различных факторов, таких как сезонность, спрос, конкуренция между авиакомпаниями и другие экономические и маркетинговые факторы. Определить оптимальное время для покупки билетов может быть сложной задачей для потребителя. Поэтому использование инструментов машинного обучения для анализа и прогнозирования цен является важным и актуальным средством для потребителей, агентств по продаже билетов и авиакомпаний.

В перспективе, основная идея состоит в том, чтобы на основе исторических данных о ценах на перелеты, а также других характеристик, таких как дата бронирования, время вылета, длительность полета, класс обслуживания и т. д., создать модель, которая будет предсказывать будущие цены.

Контекст

С ростом численности населения и увеличением благосостояния увеличивается количество авиаперевозок. С развитием цифровых технологий появилась возможность анализа как финансовой, так и не финансовой информации для анализа возможных маршрутов путешествий разных групп людей.

В настоящее время на сайте aviasales.ru в открытом доступе имеется большой объём информации по вылетам в разные города и страны, интересующей самые разные категории пользователей — начиная от самых дешевых билетов и заканчивая бизнес-классом. Однако данные представлены в неструктурированном виде, при котором их визуальное восприятие и анализ затруднительны без осуществления обработки.

Очевидно, что ручной поиск информации и последующий ее анализ на текущий момент — не эффективный инструмент, поэтому все больше компаний переходят на применение программных средств для автоматизации данного процесса. Также есть несколько примеров статей авторов, которые уже затронули данную тему в своих статьях.

Пример таких статей со способами и инструментами для машинного обучения мониторинга цен перелетов представлен ниже:

1. «Modelling of Passenger Air Transportation Prices» Olga P. Sushko, Nickolay D. Koryagin [1]. В статье авторы уделили большое внимание динамике колебания цен на авиаперевозки методом регрессионного анализа.
2. «Predicting Flight Prices with Machine Learning» Vinicius Oliveira Lima and André Cunha [2]. Статья предлагает метод прогнозирования цен на перелеты с использованием машинного обучения, основанного на моделях регрессии.
3. «Airfare prediction using Machine Learning» Amanbir Singh and Gourav Ahuja [3]. В этой статье авторы исследуют различные алгоритмы машинного обучения для прогнозирования цен на перелеты и сравнивают их производительность
4. «Using Machine Learning to Predict Flight Prices» Debajyoti Das and Paul Duan [4]. Авторы предлагают модель прогнозирования цен на перелеты на основе временных рядов и используют алгоритмы машинного обучения, такие как ARIMA и LSTM.

Вопросы применения машинного обучения для анализа данных исследуется в большом количестве научных статей [6, 8–20]. Применение машинного обучения для анализа больших данных позволяет анализировать большие неструктурированные данные, обрабатывать данные и визуализировать выдачу данных.

Была поставлена задача проанализировать перелеты на сайте aviasales.ru [5, 7] (метапоисковик, в котором можно найти информацию по ценам на авиабилеты, а также по их продавцам) на разные расстояния и выявлены закономерности и причины колебания цен. Было решено с помощью подключения по API забирать данные с сайта, обрабатывать и визуализировать их, чтобы нагляднее видеть изменение цен в зависимости от даты вылета, продавца авиабилетов и дальности полета.

Инструмент: парсинг и визуализация данных с сайта aviasales.ru для мониторинга цен на авиабилеты с помощью Python.

Задача: разработать визуализацию данных с сайта Aviasales и продемонстрировать применение данной структуры для мониторинга цен на авиабилеты по направлениям Москва-Стамбул-Москва и Москва-Адлер-Москва с помощью применения языка программирования Python.

Шаги

1. Изучение сайтов с данными об авиаперелетах.
2. Ознакомление с технической документации для работы с данными сайта.
3. Подготовка и настройка окружения для парсинга данных.
4. Работа с данными, их анализ и визуализация.

Далее описано, как производилась работа с данными, какие основные инструменты использовались для парсинга данных и построения графиков с агрегированными показателями.

1. Импорт базы данных

Был проведен импорт необходимых библиотеки

```
#Импорт библиотек
import requests
import json
import pandas as pd
from itertools import product
from tqdm import tqdm
import matplotlib
import matplotlib.pyplot as plt
```

С помощью подключения к API получены данные о ценах на авиабилеты с сайта aviasales.ru в формате json.

```
#Подключение к API Aviasales
#Получаем токен на сайте Aviasales.
Для безопасности можно вынести токен
в отдельный файл.
TOKEN = 'token'
cur = 'rub'
limit = 1000
trip_class = 0
origin = 'MOW'


# Данные по вылету в Стамбул
destination = 'IST'
resp = requests.get(f'http://api.
travelpayouts.com/v2/prices/
latest?currency={cur}&period_type=ye
ar&page=1&limit={limit}&sorting=pric
e&trip_class={trip_class}&origin={or
igin}&destination={destination}&toke
n={TOKEN}')
price_data_0 = resp.json()

# Данные по вылету в Адлер
destination = 'AER'
resp = requests.get(f'http://api.
travelpayouts.com/v2/prices/
latest?currency={cur}&period_type=ye
ar&page=1&limit={limit}&sorting=pric
e&trip_class={trip_class}&origin={or
igin}&destination={destination}&toke
n={TOKEN}')
price_data_1 = resp.json()
```

Пример получаемых данных


```
{'currency': 'rub',
 'error': '',
 'data': [{'depart_date': '2024-01-17',
   'origin': 'MOW',
   'destination': 'AER',
   'gate': 'Utair',
   'return_date': '2024-01-17',
   'found_at': '2024-01-11T16:36:47',
   'trip_class': 0,
   'value': 4883,
   'number_of_changes': 0,
   'duration': 465,
   'distance': 1364,
   'show_to_affiliates': True,
   'actual': True},
  {'depart_date': '2024-01-16',
   'origin': 'MOW',
   'destination': 'AER',
   'gate': 'Utair',
   'return_date': '2024-01-17',
   'found_at': '2024-01-11T16:42:08',
   'trip_class': 0,
   'value': 4883,
   'number_of_changes': 0,
   'duration': 465,
...
```


Опишем основные столбцы полученной базы данных:

- beginning_of_period — начало периода, на который приходятся даты отправления (в формате ГГГГ-ММ-ДД, например, 2016–05–01). Должен быть указан, если period_type равен месяцу.
- period_type — период, за который были найдены билеты (обязательный параметр):
- year — за все время;
- month– за месяц.
- one_way — true — в одну сторону, false — последовательно. Значение по умолчанию — false.
- page– номер страницы. Значение по умолчанию — 1.
- limit — общее количество записей на странице. Значение по умолчанию — 30. Максимальное значение — 1000.
- show_to_affiliates — false — все цены, true — только цены, найденные с помощью маркера партнера (рекомендуется). Значение по умолчанию — true.
- sorting– сортировка цен:

- price — по цене (значение по умолчанию). Для направлений город — город возможна только сортировка по цене;
- route– по популярности маршрута;
- distance_unit_price — по цене за 1 км.
- trip_class — класс перелета:
- 0 — эконом-класс (значение по умолчанию);
- 1 — Бизнес-класс;
- 2 — Первый класс.
- trip_duration — продолжительность пребывания в неделях или днях (для period_type=-день).
- token — индивидуальный партнерский токен.

Ниже представлена визуализация данных в виде таблицы с данными непосредственно на сайте aviasales.ru (см. рис. 1).

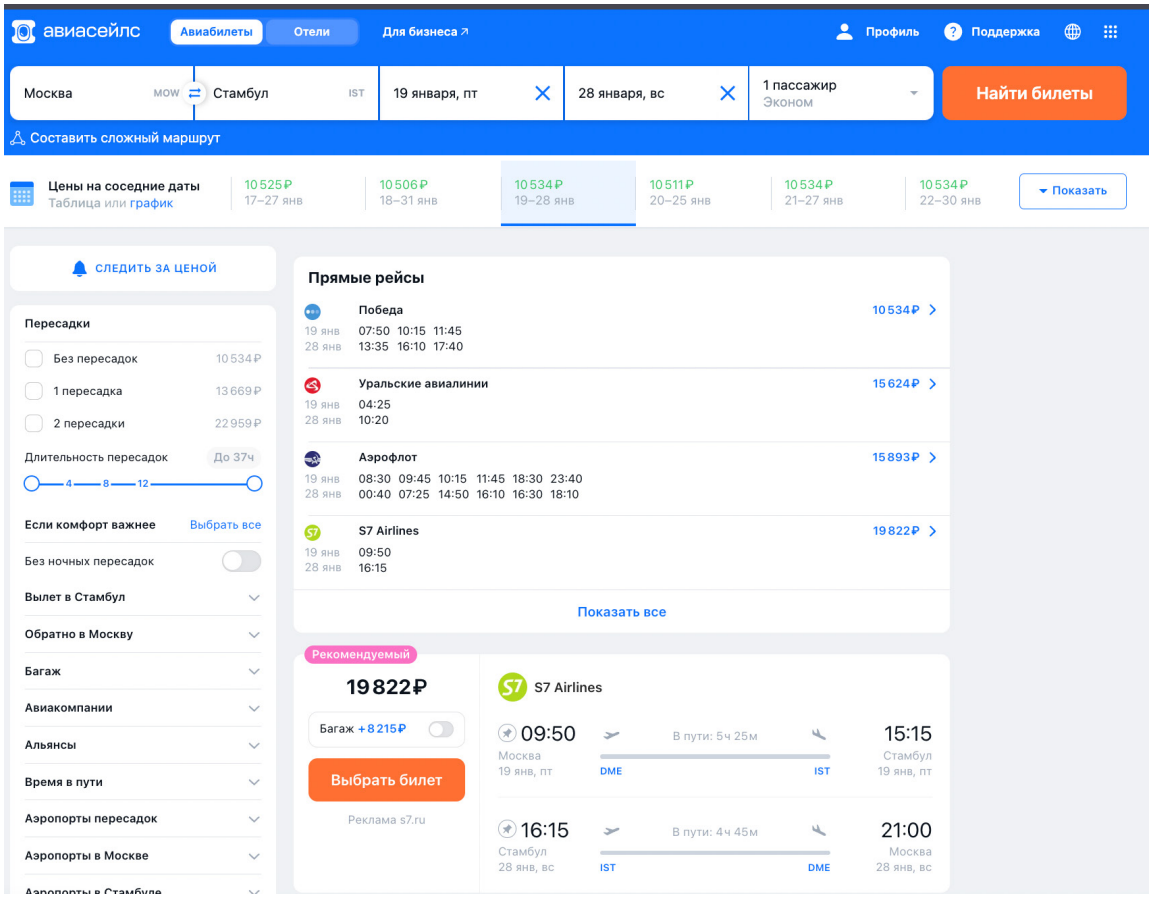

Рис. 1. Визуализация выдачи данных при поиске на сайте aviaseles.ru.

На сайте можно точечно увидеть цены на интересующие даты перелета, но нет агрегированной информации об изменениях цены.

2. Предобработка данных

Для того, чтобы нагляднее было использовать данные, можно преобразовать их из json в dataframe. Этот вариант подразумевает, что данные будут отображаться в формате таблицы.

```
#Преобразование данных из API
в dataframe
price_df_0 = pd.DataFrame(price_
data_0['data'])
price_df_1 = pd.DataFrame(price_
data_1['data'])
price_df_1
```

	depart_date	origin	destination	gate	return_date	found_at	trip_class	value	number_of_changes	duration	distance	show_to_affiliates	actual
0	2024-01-14	MOW	AER	Utair	2024-01-18	2024-01-10T16:50:53	0	4870	0	460	1364	True	True
1	2024-01-16	MOW	AER	Utair	2024-01-18	2024-01-10T16:51:21	0	4870	0	460	1364	True	True
2	2024-01-13	MOW	AER	Utair	2024-01-18	2024-01-10T16:52:17	0	4870	0	460	1364	True	True
3	2024-01-11	MOW	AER	Utair	2024-01-18	2024-01-10T16:51:44	0	4870	0	485	1364	True	True
4	2024-01-12	MOW	AER	Kupi.com	2024-01-30	2024-01-10T15:14:02	0	5466	0	455	1364	True	True
...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	2024-03-06	MOW	AER	Kupi.com	2024-03-28	2024-01-10T09:43:48	0	10817	0	445	1338	True	True
996	2024-03-21	MOW	AER	Kupi.com	2024-03-26	2024-01-10T14:46:06	0	10817	0	445	1338	True	True
997	2024-03-19	MOW	AER	Kupi.com	2024-03-26	2024-01-10T16:18:12	0	10817	0	445	1338	True	True
998	2024-03-04	MOW	AER	MEGO.travel	2024-03-13	2024-01-10T14:31:04	0	10817	0	445	1338	True	True
999	2024-03-15	MOW	AER	MEGO.travel	2024-04-06	2024-01-04T19:17:13	0	10821	0	435	1338	True	True

1000 rows × 13 columns

Проверим базу данных на отсутствие пропусков и соответствие типов.

```
price_df_1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 13 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   depart_date         1000 non-null   object
 1   origin              1000 non-null   object
 2   destination         1000 non-null   object
 3   gate                1000 non-null   object
 4   return_date         1000 non-null   object
 5   found_at            1000 non-null   object
 6   trip_class          1000 non-null   int64 
 7   value               1000 non-null   int64 
 8   number_of_changes   1000 non-null   int64 
 9   duration            1000 non-null   int64 
 10  distance            1000 non-null   int64 
 11  show_to_affiliates  1000 non-null   bool  
 12  actual              1000 non-null   bool  
dtypes: bool(2), int64(5), object(6)
memory usage: 88.0+ KB
```

Для того, чтобы сравнить цены на авиабилеты до Стамбула и Адлера, необходимо объединить их в один датафрейм.

```
# Мержим билеты до Стамбула и Адлера по датам для того, чтобы сравнить цены на графике
price_df__pre_merge_0 = price_df_0[['depart_date', 'value', 'destination']]
price_df__pre_merge_1 = price_df_1[['depart_date', 'value', 'destination']]
min_price_df__pre_merge_0 = price_df__pre_merge_0.groupby('depart_date')['value'].min()
min_price_df__pre_merge_1 = price_df__pre_merge_1.groupby('depart_date')['value'].min()

prices = price_df__pre_merge_0.merge(price_df__pre_merge_1, how='inner', on='depart_date', sort='depart_date', suffixes=('_ist', '_aer'))
min_prices = prices.groupby('depart_date')[['value_ist', 'value_aer']].min()
min_prices
```

depart_date	value_ist	value_aer
2024-01-10	18910	6539
2024-01-11	14184	4870
2024-01-12	14139	5466
2024-01-13	14126	4870
2024-01-14	14184	4870
2024-01-15	13911	5470
2024-01-16	14239	4870
2024-01-17	13911	5475
2024-01-18	14139	6070
2024-01-19	13911	6070
2024-01-20	14139	6399
2024-01-21	13923	6665
2024-01-22	13911	6666

3. Анализ данных

После получения обработанных данных проводится процесс визуализации данных. Для этого используется библиотека matplotlib, которая позволяет строить графики в различных форматах: гистограмма, линейный график, круговая диаграмма, диаграмму размаха и т. д.

1. Составлен линейный график с динамикой изменения цены за авиабилеты Москва-Стамбул-Москва (рис. 2).

```
price_df_0.plot(
x='depart_date',
y='value',
title = 'Динамика изменения цены
за билеты MOW-IST-MOW, руб',
color = 'green'
)
```

Динамика изменения цен на авиабилеты может быть сложной для отслеживания из-за множества факторов, которые влияют на ценообразование. Однако есть несколько общих принципов, которые могут помочь объяснить, почему цены на авиабилеты растут.

1. Сезонные колебания. В зависимости от времени года, спрос на авиабилеты может значительно колебаться.

2. Повышение спроса. Когда спрос на авиабилеты превышает предложение, авиакомпании могут увеличить цены, чтобы покрыть свои издержки. Это может произойти, например, во время крупных спортивных событий или международных мероприятий, когда большое количество людей путешествует одновременно.

3. Изменения в тарифах и правилах. Авиакомпании могут менять тарифы и правила в любое время, что также может повлиять на стоимость авиабилетов. Это может включать изменения в скидках, бонусных программах и специальных предложениях.

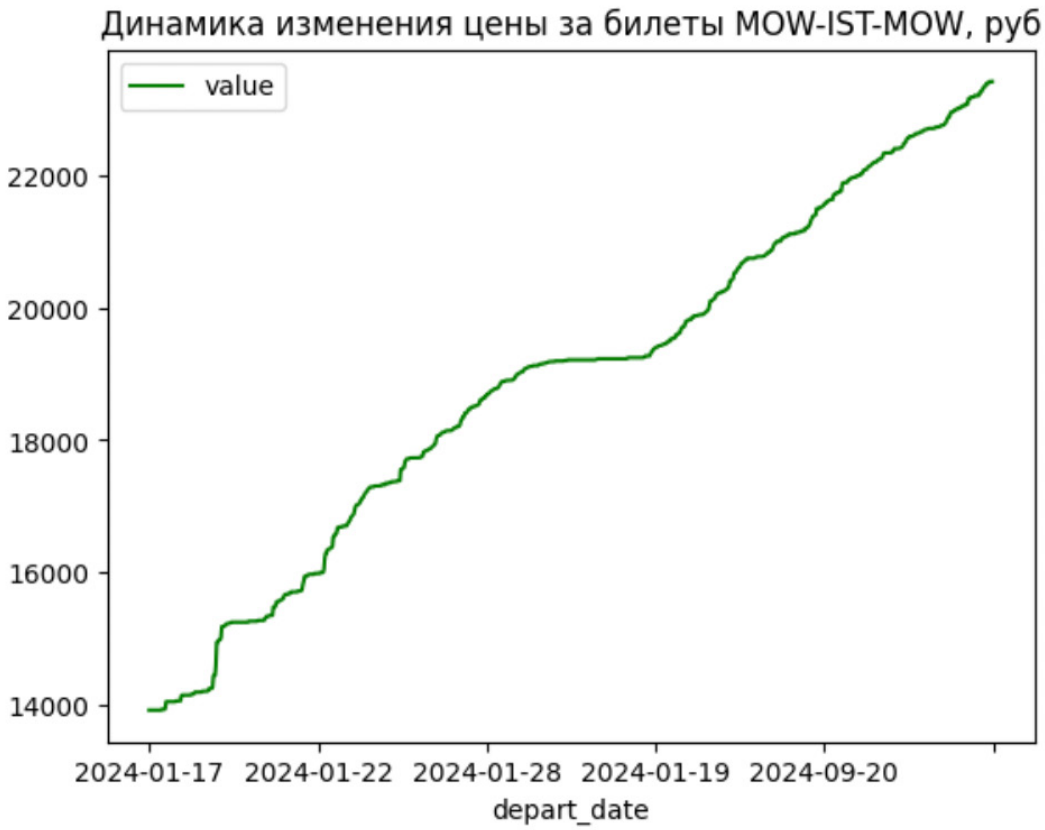

Рис. 2. Динамика изменения цены за билеты на направлении Москва-Стамбул-Москва в 2024 году.

4. Изменение курса валют. Цены на авиабилеты могут колебаться из-за изменений в обменных курсах валют. Если валюта, которую авиакомпании используют для расчета цен, ослабевает по отношению к другим валютам, цены могут вырасти.

2. Гистограмма с ценами на авиабилеты с группировкой по продавцам (рис. 3).

```
# Минимальная цена с группировкой по продавцам
min_price = price_df_0.
groupby('gate')['value'].min()
min_price.plot(
x='gate',
y='value',
kind = 'bar',
title = 'Минимальная цена билета MOW-IST-MOW по продавцам, руб',
color = 'orange'
)
```

3. Линейный график с ценами на авиабилеты до Стамбула и Адлера (см. рис. 4).

```
# Выводим график
min_prices.plot(
y=['value_ist','value_aer'],
title = 'Ежедневное сравнение цен по билетам в Стамбул и Адлер, руб',
kind = 'line',
fontsize=7
)
```

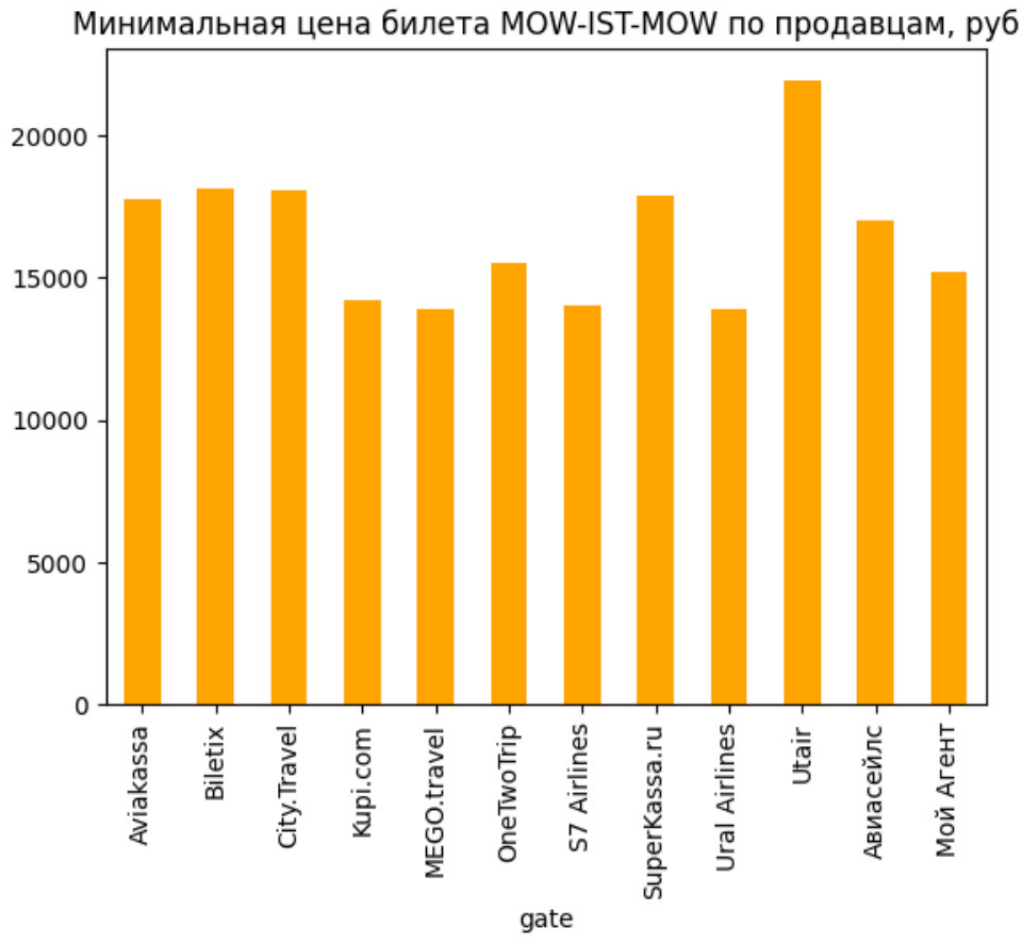

Рис. 3. Гистаграмма, демонстрирующая распределение минимальных цен на билеты Москва-Стамбул по различным продавцам.

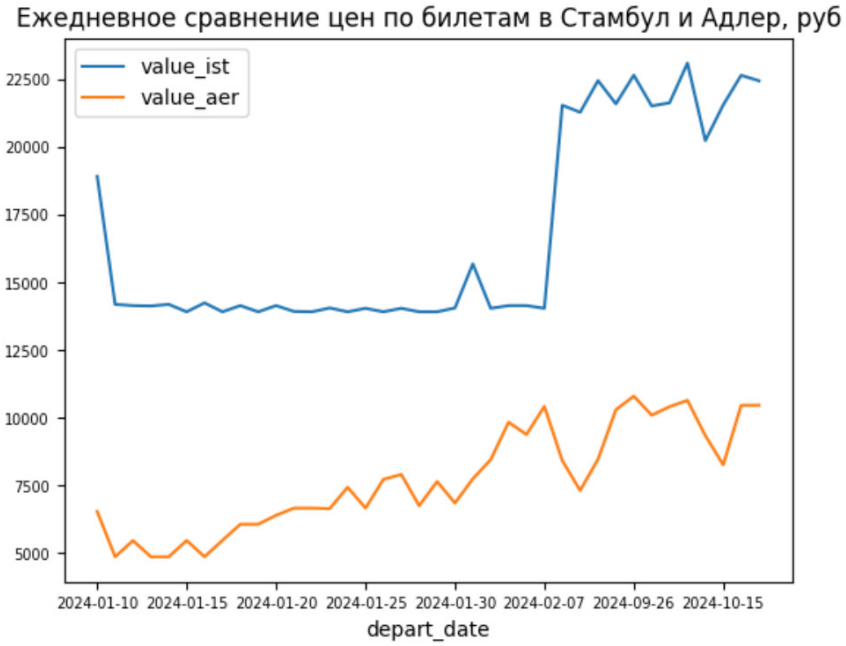

Рис. 4. Линейный график с ценами на авиабилеты до Стамбула и Адлера.

Весь код, представленный выше, может быть также применен к другим базам данных Aviasales.

Список ссылок, для которых адаптирован проект:

- Докум**ентация API Aviasales —** https://travelpayouts-data-api.readthedocs.io/en/latest/
- **Официальный сайт Aviasales —** https://www.aviasales.ru/

Для создания модели прогнозирования цен на авиабилеты было использовано программное средство Microsoft Excel, с помощью которого были проанализированы прошлые данные для прогнозирования будущих цен. В качестве исходных данных была использована информация о средней цене на внутренние рейсы за последние 6 месяцев и применена функция линейной регрессии для создания модели прогноза и функция «ТЕНДЕНЦИЯ» для получения прогноза на следующий месяц.

Заключение

Применение анализа больших данных, машинного обучения и использования парсинга данных в сфере прогнозирования цен на авиабилеты имеет высокое практическое значение. В статье представлен пошаговая схема процесса прогнозирования цен на авиабилеты с использванием данных с сайта *aviasales.ru*. Описан процесс парсинга данных, их последующего анализа и визуализации.

Определение оптимальной стоимости авиабилетов и времени их покупки может быть затруднено из-за большого количества факторов, влияющих на цены: сезонность, уровень спроса, конкуренция между авиакомпаниями, экономические и маркетинговые условия и др. Все эти факторы учитываются при использовании машинного обучения и больших данных для анализа и прогнозирования цен на билеты.

Возможности применения машинного обучения при анализе маркетинговых данных дают широкие возможности для построения модели прогнозирования цен с учетом таких дополнительных характеристик, как дата покупки, время отправления, продолжительность полета, класс бронирования. Это позволит авиакомпаниям и агентствам по продаже авиабилетов принимать более обоснованные решения о ценообразовании, а потребителям — получать более точные и полезные данные для выбора оптимального времени покупки билета.

Литература

1. Olga P. Sushko, Nickolay D. Koryagin «Modelling of Passenger Air Transportation Prices»–2022. — Режим доступа: URL: docviewer.yandex.ru
2. Vinicius Oliveira Lima and André Cunha « Predicting Flight Prices with Machine Learning «–2017. — Режим доступа: URL: https://achyutjoshi.github.io/btp/flightprices
3. Amanbir Singh and Gourav Ahuja « Airfare prediction using Machine Learning «–2022. — Режим доступа: URL: https://jetir.org/view?paper=JETIR2204660
4. Debajyoti Das and Paul Duan « Using Machine Learning to Predict Flight Prices «–2019. —Режим доступа: URL: https://www.researchgate.net/publication/364350711_Flight_Price_Prediction_Using_Machine_Learning_Techniques
5. Официальный сайт компании — Режим доступа: URL: https://www.aviasales.ru/?marker=15468.ydofav13322167936&etext=&yclid=13007950079453036543
6. Аллен Б. Дауни — «Think DSP. Цифровая обработка сигналов на Python» — Издательство «ДМК Пресс» — 2017 — Режим доступа: URL: https://avidreaders.ru/read-book/cifrovaya-obrabotka-signalov-na-yazyke-python.html
7. Документация API aviasales — Режим доступа: URL: https://travelpayouts-data-api.readthedocs.io/en/latest/
8. Элейн Энджел «Энциклопедия Парсинга. Настольная книга мастера и клиента»–2022 — Режим доступа: URL: https://royallib.com/read/endgel_eleyn/entsiklopediya_pirsinga_nastolnaya_kniga_mastera_i_klienta.html#0
9. Kyran Dale — «Data Visualization with Python and JavaScript» — 2016 — Режим доступа: URL: https://www.oreilly.com/library/view/data-visualization-with/9781491920565/
10. Ryan Mitchell — «Web Scraping with Python» — 2017 — Режим доступа: URL: https://www.oreilly.com/library/view/web-scraping-with/9781491910283/
11. Seppevan den Broucke, Bart Baesens — «Practical Web Scraping for Data Science» — 2018 — Режим доступа: URL: https://link.springer.com/book/10.1007/978–1–4842–3582–9
12. Jay M. Patel — «Getting Structured Data from the Internet: Running Web Crawlers/Scrapers on a Big Data Production Scale» — 2020 — Режим доступа: URL: https://www.oreilly.com/library/view/getting-structured-data/9781484265765/
13. Билл Любанович — «Простой Python. Современный стиль программирования» — 2021 — Режим доступа: URL: https://codelibs.ru/python/prostoj-python/
14. Дональд Р. Шихи — «Структуры данных в Python: начальный курс» — 2022 — Режим доступа: URL: https://codelibs.ru/python/strukturyi-dannyih-v-python-nachalnyij-kurs-768f8d1f/
15. Брантон, Куц — «Анализ данных в науке и технике» — 2021 — Режим доступа: URL: https://codelibs.ru/big-data/analiz-dannyih-v-nauke-i-tehnike/
16. Эрик Мэтиз — «Изучаем Python. Программирование игр, визуализация данных, веб-приложения» Режим доступа: URL: https://studylib.ru/doc/6336763/izuchaem-python-2020-e-rik-me-tiz
17. Билл Любанович — «Простой Python. Современный стиль программирования» — 2021 — Режим доступа: URL: https://coderbooks.ru/books/python/prostoj_python_lyubanovich_2021/
18. Уэс Маккини — «Python и анализ данных» — 2017 — Режим доступа: URL: https://coderbooks.ru/books/python/python_i_analiz_dannyh_makkinni_2015/
19. Saurabh Chandrakar, Dr. Nilesh Bhaskarrao Bahadure — «Python for Everyone» — 2023 — Режим доступа: URL: https://coderbooks.ru/books/python/python-for-everyone/
20. Джоэл Грасс — «Data Science. Наука о данных с нуля» — 2021 — Режим доступа: URL: https://codelibs.ru/big-data/data-science-nauka-o-dannyih-s-nulya-2-e-izdanie/

Analysis and forecasting of flight prices using machine learning tools

Chizhikova V. A., Bulgakov A. V.
Plekhanov Russian University of Economics, Lomonosov Moscow State University, Moscow Institute of Modern Academic Education

The article analyzes the possibility of using machine learning methods to conduct marketing research in the air travel market. Determining the optimal cost of air tickets and the time of their purchase can be difficult due to a large number of factors affecting prices, including seasonality, demand levels, competition between airlines, and other conditions. An attempt was made to take these factors into account when forecasting air ticket prices using machine learning methods. To forecast flight prices from the aviasales.ru website, data parsing was used as a tool for processing information that is displayed on the website. The article provides an example of data parsing and examines the results obtained, including studying the possibility of further data analysis and data visualization capabilities.

Keywords: tourism, big data, machine learning, parsing, forecasting.

Refences

1. Olga P. Sushko, Nickolay D. Koryagin «Modeling of Passenger Air Transportation Prices»–2022. — Access mode: URL: docviewer.yandex.ru
2. Vinicius Oliveira Lima and André Cunha «Predicting Flight Prices with Machine Learning»–2017. — Access mode: URL: https://achyutjoshi.github.io/btp/flightprices
3. Amanbir Singh and Gourav Ahuja «Airfare prediction using Machine Learning»–2022. — Access mode: URL: https://jetir.org/view?paper=JETIR2204660
4. Debajyoti Das and Paul Duan «Using Machine Learning to Predict Flight Prices»–2019. — Access mode: URL: https://www.researchgate.net/publication/364350711_Flight_Price_Prediction_Using_Machine_Learning_Techniques
5. Official website of the company — Access mode: URL: https://www.aviasales.ru/?marker=15468.ydofav13322167936&etext=&yclid=13007950079453036543
6. Allen B. Downey — «Think DSP. Digital Signal Processing in Python» — Publisher «DMK Press» — 2017 — Access mode: URL: https://avidreaders.ru/read-book/cifrovaya-obrabotka-signalov-na-yazyke-python.html
7. API documentation aviasales — Access mode: URL: https://travelpayouts-data-api.readthedocs.io/en/latest/
8. Elaine Engel «Encyclopedia of Parsing. Handbook of the Master and the Client» — 2022 — Access mode: URL: https://royallib.com/read/endgel_eleyn/entsiklopediya_pirsinga_nastolnaya_kniga_mastera_i_klienta.html#0
9. Kyran Dale — «Data Visualization with Python and JavaScript» — 2016 — Access mode: URL: https://www.oreilly.com/library/view/data-visualization-with/9781491920565/
10. Ryan Mitchell — «Web Scraping with Python» — 2017 — Access mode: URL: https://www.oreilly.com/library/view/web-scraping-with/9781491910283/
11. Seppe vanden Broucke, Bart Baesens — «Practical Web Scraping for Data Science» — 2018 — Access mode: URL: https://link.springer.com/book/10.1007/978–1–4842–3582–9
12. Jay M. Patel — «Getting Structured Data from the Internet: Running Web Crawlers/Scrapers on a Big Data Production Scale» — 2020 — Access mode: URL: https://www.oreilly.com/library/view/getting-structured-data/9781484265765/
13. Bill Lubanovich — «Simple Python. Modern programming style» — 2021 — Access mode: URL: https://codelibs.ru/python/prostoj-python/
14. Donald R. Sheehy — «Data structures in Python: a basic course» — 2022 — Access mode: URL: https://codelibs.ru/python/strukturyi-dannyih-v-python-nachalnyij-kurs-768f8d1f/
15. Brunton, Kutz — «Data analysis in science and technology» — 2021 — Access mode: URL: https://codelibs.ru/big-data/analiz-dannyih-v-nauke-i-tehnike/
16. Eric Matiz — «Learning Python. Game Programming, Data Visualization, Web Applications» Access mode: URL: https://studylib.ru/doc/6336763/izuchaem-python-2020-e-rik-me-tiz
17. Bill Lubanovich — «Simple Python. Modern Programming Style» — 2021 — Access mode: URL: https://coderbooks.ru/books/python/prostoj_python_lyubanovich_2021/
18. Wes McKinney — «Python and Data Analysis» — 2017 — Access mode: URL: https://coderbooks.ru/books/python/python_i_analiz_dannyh_makkinni_2015/
19. Saurabh Chandrakar, Dr. Nilesh Bhaskarrao Bahadure — «Python for Everyone» — 2023 — Access mode: URL: https://coderbooks.ru/books/python/python-for-everyone/
20. Joel Grass — «Data Science. Data Science from Scratch» — 2021 — Access mode: URL: https://codelibs.ru/big-data/data-science-nauka-o-dannyih-s-nulya-2-e-izdanie/

МАТЕМАТИЧЕСКИЕ, СТАТИСТИЧЕСКИЕ
И ИНСТРУМЕНТАЛЬНЫЕ МЕТОДЫ В ЭКОНОМИКЕ

Выявление оптимальных факторов для прогнозирования демографических показателей с помощью методов машинного обучения

Горбас Данила Андреевич
магистр, РЭУ им. Плеханова
E-mail: qorbas9@gmail.com

Булгаков Андрей Леонидович
канд. экономич. наук, доцент МГУ имени М. В. Ломоносова;
доцент РЭУ им. Г. В. Плеханова
профессор Московского института современного академического образования
E-mail: z3900207@mail.ru

Милютин Максим Александрович
студент, экономический факультет МГУ имени М. В. Ломоносова

Вопросы прогнозирования демографических показателей является актуальным для развития общества и экономики. Полученные на основании прогноза демографических показателей данные позволяют корректировать государственную политику в области занятости и развития экономики. Для прогнозирования демографических показателей используют разные методы. В статье предпринята попытка изучить вопрос прогнозирования демографических показателей с помощью методов машинного обучения, который можно использовать как мощный инструмент, позволяющий анализировать обширные объемы данных и выделять оптимальные факторы, влияющие на динамику населения.

Ключевые слова: машинное обучение, демографические показатели, прогнозирование, обработка данных, оптимизация факторов.

Использование искусственного интеллекта для анализа и прогнозирования данных

В последние десятилетия искусственный интеллект (ИИ) стал неотъемлемой частью передовых исследований в различных областях, включая анализ данных и прогнозирование. В контексте прогнозирования демографических показателей, ИИ предоставляет мощные алгоритмы и инструменты для обработки сложных многомерных данных, что позволяет исследователям выявлять неявные взаимосвязи и тренды [1–7].

Одним из ключевых преимуществ использования искусственного интеллекта в данном исследовании является его способность работать с обширными объемами информации. Алгоритмы машинного обучения способны обрабатывать данные о населении, ВВП, продолжительности жизни, безработице, образовании, а также информацию о здравоохранении, включая количество пациентов анатомо-курортных организаций и их общее количество [8–12]. Это позволяет создать комплексные модели, учитывающие множество влияющих факторов, что в свою очередь повышает точность и обобщающую способность прогнозов.

Еще одной значимой характеристикой искусственного интеллекта является его способность к автоматическому обучению на основе данных. Это позволяет моделям адаптироваться к изменениям в демографических трендах и динамике общественных про-

цессов, что является крайне важным аспектом в условиях постоянной эволюции социально-экономической среды.

В рамках данного исследования были применены различные алгоритмы машинного обучения, такие как регрессионный анализ, нейронные сети и методы кластеризации, для выделения оптимальных факторов, влияющих на демографические показатели. Подход, основанный на искусственном интеллекте, позволяет обеспечивает метод анализа данных, который не только облегчит понимание сложных взаимосвязей, но и предоставит возможность более точного прогнозирования демографического развития.

Ключевые факторы для прогнозирования демографических показателей с использованием методов машинного обучения

При проведении анализа и прогнозирования демографических показателей используется широкий спектр переменных, охватывающих не только основные демографические индикаторы, такие как продолжительность жизни и безработица, но и факторы, влияющие на социально-экономическое благополучие, такие как ВВП, образование, а также данные о здравоохранении в виде числа пациентов анатомо-курортных организаций и их общего количества.

Анализ такого комплекса данных позволит не только углубленно понять взаимосвязи между различными переменными, но и разработать более точные и адаптивные модели прогнозирования демографических показателей. Полученные результаты будут иметь практическое значение для государственных органов, социальных учреждений и предприятий, внедряющих стратегии развития с учетом изменяющихся демографических трендов.

Источники данных

Название датасета: Население, Площадь и Плотность населения

Ссылка на датасет: http://data.un.org/_Docs/SYB/CSV/SYB65_1_202209_Population,%20Surface%20Area%20and%20Density.csv

Количество наблюдений: По годам и странам

Этот датасет предоставляет информацию о населении, площади и плотности населения для различных стран. Данные включают информацию за разные годы, что позволяет провести анализ динамики изменений населения и его распределения по территории.

Название датасета: ВВП и ВВП на душу населения

Ссылка на датасет: https://data.un.org/_Docs/SYB/CSV/SYB65_230_202209_GDP%20and%20GDP%20Per%20Capita.csv

Количество наблюдений: По годам и странам

Этот датасет содержит информацию о ВВП (валовом внутреннем продукте) и ВВП на душу населения для различных стран. Анализ этих данных позволяет оценить экономическое развитие стран и сравнивать их уровень благосостояния.

Название датасета: Продолжительность жизни

Ссылка на датасет: https://data.un.org/_Docs/SYB/CSV/SYB65_246_202209_Population%20Growth,%20Fertility%20and%20Mortality%20Indicators.csv

Количество наблюдений: По годам и странам

Датасет предоставляет информацию о продолжительности жизни в различных странах. Эти данные важны для оценки здоровья нации и эффективности системы здравоохранения.

Название датасета: Трудовая сила и Безработица

Ссылка на датасет: https://data.un.org/_Docs/SYB/CSV/SYB65_329_202209_Labour%20Force%20and%20Unemployment.csv

Количество наблюдений: По годам и странам

Датасет предоставляет информацию о трудовой силе и уровне безработицы в различных странах. Анализ этих данных помогает понять динамику занятости и проблемы на рынке труда.

Название датасета: Образование

Ссылка на датасет: https://data.un.org/_Docs/SYB/CSV/SYB65_309_202209_Education.csv

Количество наблюдений: По годам и странам

Данные о образовании включают информацию о уровне грамотности, уровне образования населения и другие показатели. Анализ этих данных позволяет оценить образовательный уровень нации и его влияние на различные аспекты жизни.

Название датасета: Численность размещенных лиц в санаторно-курортных организациях (человек)

Ссылка на датасет: https://www.fedstat.ru/indicator/42102

Количество наблюдений: По годам и регионам Российской Федерации

Данный датасет, предоставленный Федеральной службой государственной статистики, содержит информацию о численности размещенных лиц в санаторно-курортных организациях. Он позволяет оценить популярность и загруженность данных организаций и связать ее с другими показателями, такими как нагрузка на систему здравоохранения и заболеваемость.

Название датасета: Число санаторно-курортных организаций

Ссылка на датасет: https://www.fedstat.ru/indicator/42106

Количество наблюдений: По годам и регионам Российской Федерации

Данный датасет, предоставленный Федеральной службой государственной статистики, содержит информацию о числе санаторно-курортных организаций. Он отражает общую сетевую инфраструктуру санаторно-курортного комплекса в различных регионах России и может быть связан с другими показателями, такими как нагрузка на систему здравоохранения и заболеваемость.

Настройка среды

```
# используйте этот код для доступа к данным на вашем облочном хранилище
from google.colab import drive
drive.mount('/content/drive')
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount(«/content/drive», force_remount=True).
# Dask — библиотека для параллельных вычислений и эффективной работы с данными, масштабируясь от небольших до больших датасетов.
# Поддерживает структуры данных, аналогичные Pandas и NumPy.
# Сайт: https://www.dask.org/
import dask.dataframe as dd

# операции с регулярными выражениями
#re — https://docs.python.org/3/library/re.html
import re

# предварительная обработка данных
# sklearn.preprocessing — https://scikit-learn.org/stable/modules/preprocessing.html
from sklearn.preprocessing import LabelEncoder

# расширенный анализ DataFrame, позволяющий экспортировать анализ данных в различные форматы, такие как html и json
# ydata_profiling — https://ydata-profiling.ydata.ai/docs/master/pages/getting_started/overview.html
#!pip install ydata-profiling
# from ydata_profiling import ProfileReport
```

```
# чтение и запись файлов CSV
# csv — https://docs.python.org/3/
library/csv.html
import csv

#блиотека для создания статистиче-
ских графиков
#seaborn — https://seaborn.pydata.
org/
import seaborn as sns

#Библиотека для машшиного обучения
#scikit-learn — https://scikit-
learn.org


import numpy as np
from sklearn.model_selection import
train_test_split
import seaborn as sns
#sns.set(style='white', font_
scale=2)
import matplotlib.pyplot as plt
from sklearn.linear_model import
LinearRegression
from sklearn.ensemble import
RandomForestRegressor
import pandas as pd

from sklearn.metrics import mean_
absolute_error, mean_squared_error,
r2_score
```

Импорт данных

Так как большинство поступающих данных будут поступать в формате, где первая строка отвечает за год, первый столбец содержит регион, а на пересечении находятся данные по интересующим нас показателям, то будет произведена трансформация данных. Трансформируем импортированные данные в структуру, где год и регион будут располагаться в соответствующих столбцах, наряду с названием исследуемого показателя. Такой формат позволит более эффективно и наглядно использовать и обрабатывать данные в дальшейнем.

Импорт данных из базы данных ООН

Импортируем данные Население, Площадь и Плотность населения.

```
# Настройка Dask DataFrame для дата-
сета Население, Площадь и Плотность
населения
dataPopulation = dd.read_
csv(«http://data.un.org/_Docs/SYB/
CSV/SYB65_1_202209_Population,%20
Surface%20Area%20and%20Density.
csv», header=1, thousands=',',
dtype={'Footnotes': 'object'})


dataPopulationProcessing = dataPo
pulation[dataPopulation[«Series»]
== «Population mid-year estimates
(millions)»]
dataPopulationProcessing = data
PopulationProcessing[dataPopul
ationProcessing[«Series»].str.
contains(«(millions)», case=False,
flags=re.I)]


dataPopulationProcessing = data
PopulationProcessing[[«Region/
Country/Area», «Unnamed: 1», «Year»,
«Value»]]
dataPopulationProcessing.columns
= ['Id', 'Region', 'Year',
'Population']


# Вывод информации о данных
dataPopulationProcessing.compute().
head()
/usr/local/lib/python3.10/dist-
packages/dask/dataframe/accessor.
py:96: UserWarning: This pattern is
interpreted as a regular expression,
and has match groups. To actually
get the groups, use str.extract.
out = getattr(getattr(obj, accessor,
obj), attr)(*args, **kwargs)
/usr/local/lib/python3.10/dist-
packages/dask/dataframe/accessor.
py:96: UserWarning: This pattern is
interpreted as a regular expression,
and has match groups. To actually
```

```
get the groups, use str.extract.
out = getattr(getattr(obj, accessor,
obj), attr)(*args, **kwargs)
```

	Id	Region	Year	GDP
0	1	Total, all countries or areas	1995	31247262.0
1	1	Total, all countries or areas	2005	47730924.0
2	1	Total, all countries or areas	2010	66461443.0
3	1	Total, all countries or areas	2015	75133208.0
4	1	Total, all countries or areas	2018	86357998.0

Импортируем данные о продолжительности жизни.

```
# Настройка Dask DataFrame для дата-
сета о продолжительности жизни
dataLifeExpectancy = dd.read_
csv(«https://data.un.org/_Docs/
SYB/CSV/SYB65_246_202209_
Population%20Growth,%20Fertility%20
and%20Mortality%20Indicators.
csv», header=1, thousands=',',
dtype={'Footnotes': 'object'})
```

```
dataLifeExpectancyProcessing = dataL
ifeExpectancy[dataLifeExpectancy[«Se
ries»] == «Life expectancy at birth
for both sexes (years)»]
dataLifeExpectancyProcessing = da
taLifeExpectancyProcessing[dataLi
feExpectancyProcessing[«Series»].
str.contains(«(years)», case=False,
flags=re.I)]
```

```
dataLifeExpectancyProcessing = data
LifeExpectancyProcessing[[«Region/
Country/Area», «Unnamed: 1», «Year»,
«Value»]]
dataLifeExpectancyProcessing.
columns = ['Id', 'Region', 'Year',
'LifeExpectancy']
```

```
# Вывод информации о данных
dataLifeExpectancyProcessing.
compute().head()
```

```
/usr/local/lib/python3.10/dist-
packages/dask/dataframe/accessor.
py:96: UserWarning: This pattern is
interpreted as a regular expression,
and has match groups. To actually
get the groups, use str.extract.
out = getattr(getattr(obj, accessor,
obj), attr)(*args, **kwargs)
/usr/local/lib/python3.10/dist-
packages/dask/dataframe/accessor.
py:96: UserWarning: This pattern is
interpreted as a regular expression,
and has match groups. To actually
get the groups, use str.extract.
out = getattr(getattr(obj, accessor,
obj), attr)(*args, **kwargs)
```

	Id	Region	Year	LifeExpectancy
4	1	Total, all countries or areas	2010	70.1
11	1	Total, all countries or areas	2015	71.8
18	1	Total, all countries or areas	2017	72.3
24	1	Total, all countries or areas	2022	71.7
30	2	Africa	2010	58.6

Импортируем данные о безработице.

```
# Настройка Dask DataFrame для дата-
сета по безработице
dataUnemployment = dd.read_
csv(«https://data.un.org/_Docs/
SYB/CSV/SYB65_329_202209_Labour%20
Force%20and%20Unemployment.csv»,
header=1, thousands=',')
```

```
# Создаем Dask DataFrame для мужчин
df_male = dataUnemployment[[«Region/
Country/Area», «Unnamed: 1», «Year»,
«Series», «Value»]]
df_male.columns = ['Id', 'Region',
'Year', 'Series', 'Value']
df_male = df_male[df_male['Series'].
str.contains('Male')]
df_male = df_male.groupby(['Id',
'Region', 'Year']).agg({
'Value': 'sum'
}).reset_index()
```

```
# Создаем Dask DataFrame для женщин
df_female =
dataUnemployment[[«Region/Country/
Area», «Unnamed: 1», «Year»,
«Series», «Value»]]
df_female.columns = ['Id', 'Region',
'Year', 'Series', 'Value']
df_female = df_female[df_
female['Series'].str.
contains('Female')]
df_female = df_female.groupby(['Id',
'Region', 'Year']).agg({
'Value': 'sum'
}).reset_index()
```

```
# Объединяем данные
dataUnemploymentProcessing =
dd.merge(df_male, df_female,
on=['Id', 'Region', 'Year'])

# Переименовываем столбцы и объеди-
няем суммарные значения
dataUnemploymentProcessing.
columns = ['Id', 'Region', 'Year',
'Unemployment rate', 'Labour force
participation']

# Вывод информации о данных
dataUnemploymentProcessing.
compute().head()
```

	Id	Region	Year	Unemployment rate	Labour force participation
0	1	Total, all countries or areas	2005	82.7	56.7
1	1	Total, all countries or areas	2010	81.3	54.9
2	1	Total, all countries or areas	2015	79.8	53.7
3	1	Total, all countries or areas	2022	77.7	52.7
4	2	Africa	2005	81.4	63.1

Импортируем данные об образовании.

```
# Настройка Dask DataFrame для дата-
сета по образованию
dataEducation = dd.read_
csv(«https://data.un.org/_Docs/SYB/
CSV/SYB65_309_202209_Education.csv»,
header=1, thousands=',')

# Создаем Dask DataFrame для мужчин
df_male = dataEducation[[«Region/
Country/Area», «Unnamed: 1», «Year»,
«Series», «Value»]]
df_male.columns = ['Id', 'Region',
'Year', 'Series', 'Value']
df_male = df_male[df_male['Series'].
str.contains('male')]
df_male = df_male.groupby(['Id',
'Region', 'Year']).agg({
'Value': 'sum'
}).reset_index()
```

```
# Создаем Dask DataFrame для женщин
df_female = dataEducation[[«Region/
Country/Area», «Unnamed: 1», «Year»,
«Series», «Value»]]
df_female.columns = ['Id', 'Region',
'Year', 'Series', 'Value']
df_female = df_female[df_
female['Series'].str.
contains('female')]
df_female = df_female.groupby(['Id',
'Region', 'Year']).agg({
'Value': 'sum'
}).reset_index()

# Объединяем данные
dataEducationProcessing =
dd.merge(df_male, df_female,
on=['Id', 'Region', 'Year'])
```

```
# Переименовываем столбцы и объединяем суммарные значения
dataEducationProcessing.columns = ['Id', 'Region', 'Year', 'Gross enrollment ratio — Primary', 'Gross enrollment ratio — Secondary']

# Вывод информации о данных
dataEducationProcessing.compute().head()
```

	Id	Region	Year	Gross enrollment ratio - Primary	Gross enrollment ratio - Secondary
0	1	Total, all countries or areas	2005	431.8	210.3
1	1	Total, all countries or areas	2010	467.0	229.8
2	1	Total, all countries or areas	2015	488.9	243.4
3	1	Total, all countries or areas	2020	495.3	245.6
4	15	Northern Africa	2005	436.4	216.2

Импорт данных о санаторно-курортных организациях

Импортируем данные о численность размещенных лиц в санаторно-курортных организациях. Источник: https://www.fedstat.ru/indicator/42102

```
dataSanatPeopleRaw = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/colab_data/sanatorium_peaple_2002_2022.csv', thousands=' ', sep=';', decimal=',')
dataSanatPeople = dataSanatPeopleRaw.melt(id_vars='Регион', var_name='Год', value_name='Размещений в санаториях')
dataSanatPeople = dd.from_pandas(dataSanatPeople, npartitions=1)
dataSanatPeople.compute().head()
```

	Регион	Год	Размещений в санаториях
0	Российская Федерация	2002	4953271.0
1	Центральный федеральный округ	2002	1044953.0
2	Белгородская область	2002	12217.0
3	Брянская область	2002	40215.0
4	Владимирская область	2002	32967.0

Импортируем данные о количестве санаторно-курортных организаций. Источник: https://www.fedstat.ru/indicator/42106

```
dataSanatCountRaw = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/colab_data/sanatorium_count.csv', thousands=' ', sep=';', decimal=',')
dataSanatCount = dataSanatCountRaw.melt(id_vars='Регион', var_name='Год', value_name='Кол-во санаториев')
dataSanatCount = dd.from_pandas(dataSanatCount, npartitions=1)
dataSanatCount.compute().head()
```

	Регион	Год	Кол-во санаториев
0	Российская Федерация	2002	2347.0
1	Центральный федеральный округ	2002	458.0
2	Белгородская область	2002	8.0
3	Брянская область	2002	20.0
4	Владимирская область	2002	26.0

Объединение полученных наборов данных

Анализ различий названий регионов

Теперь необходимо выполнить слияние полученных наборов данных по годам и регионам. Однако если внимательно разобрать полученные данные (или попробовать провести слияние в автоматическом режиме), то видно, что множество регионов из первого набора данных не находят пары во втором. Это вызвано наличием лишних данных по другим странам, а также по регионам РФ. Чтобы решить эту проблему, необходимо избавиться от ненужных данных.

```
# Удаление строк с регионом, отличным от «Российская Федерация»
dataSanatPeople = dataSanatPeople.loc[dataSanatPeople['Регион'] == 'Российская Федерация']

# Вывод результата
dataSanatPeople.compute().head()
```

	Регион	Год	Размещений в санаториях
0	Российская Федерация	2002	4953271.0
106	Российская Федерация	2003	4961015.0
212	Российская Федерация	2004	5472792.0
318	Российская Федерация	2005	5941198.0
424	Российская Федерация	2006	6084758.0

```
# Удаление строк с регионом, отличным от «Российская Федерация»
dataSanatCount = dataSanatCount.loc[dataSanatCount['Регион'] == 'Российская Федерация']

# Вывод результата
dataSanatCount.compute().head()
```

	Регион	Год	Кол-во санаториев
0	Российская Федерация	2002	2347.0
108	Российская Федерация	2003	2259.0
216	Российская Федерация	2004	2233.0
324	Российская Федерация	2005	2173.0
432	Российская Федерация	2006	2148.0

Также в наборах данных полученных из ООН присутствует статистика не только по РФ, необходимо удалить излишние данные и привести наборы данных к единому виду.

```
dataEducationProcessing = dataEducationProcessing.loc[dataEducationProcessing['Region'] == 'Russian Federation']
dataUnemploymentProcessing = dataUnemploymentProcessing.loc[dataUnemploymentProcessing['Region'] == 'Russian Federation']
dataLifeExpectancyProcessing = dataLifeExpectancyProcessing.loc[dataLifeExpectancyProcessing['Region'] == 'Russian Federation']
dataGDPProcessing = dataGDPProcessing.loc[dataGDPProcessing['Region'] == 'Russian Federation']
dataPopulationProcessing = dataPopulationProcessing.loc[dataPopulationProcessing['Region'] == 'Russian Federation']
```

Объединение модифицированных данных

Теперь выполним слияние данных полученных наборов данных. Начнем с данных о санаториях.

```
logicData = dd.merge(dataSanatPeople, dataSanatCount, on=['Регион', 'Год'])
logicData.compute().head()
```

	Регион	Год	Размещений в санаториях	Кол-во санаториев
0	Российская Федерация	2002	4953271.0	2347.0
1	Российская Федерация	2003	4961015.0	2259.0
2	Российская Федерация	2004	5472792.0	2233.0
3	Российская Федерация	2005	5941198.0	2173.0
4	Российская Федерация	2006	6084758.0	2148.0

Далее выполним слияние данных ОНН, предварительно заменив на русский названия регионов и удалив избыточные столбцы.

```
dataPopulationProcessing['Region'] =
dataPopulationProcessing['Region'].
replace('Russian Federation', 'Рос-
сийская Федерация')
dataGDPProcessing['Region'] =
dataGDPProcessing['Region'].
replace('Russian Federation', 'Рос-
сийская Федерация')
dataLifeExpectancyProcessing['Re
gion'] = dataLifeExpectancyProce
ssing['Region'].replace('Russian
Federation', 'Российская Федерация')
dataUnemploymentProcessing['Region']
= dataUnemploymentProcessing['Regi
on'].replace('Russian Federation',
'Российская Федерация')
dataEducationProcessing['Region'] =
dataEducationProcessing['Region'].
replace('Russian Federation', 'Рос-
сийская Федерация')

dataPopulationProcessing =
dataPopulationProcessing.
rename(columns={'Region': 'Регион'})
dataGDPProcessing =
dataGDPProcessing.
rename(columns={'Region': 'Регион'})
dataLifeExpectancyProcessing =
dataLifeExpectancyProcessing.
rename(columns={'Region': 'Регион'})
dataUnemploymentProcessing =
dataUnemploymentProcessing.
rename(columns={'Region': 'Регион'})
dataEducationProcessing =
dataEducationProcessing.
rename(columns={'Region': 'Регион'})

dataPopulationProcessing =
dataPopulationProcessing.
rename(columns={'Year': 'Год'})
dataGDPProcessing =
dataGDPProcessing.
rename(columns={'Year': 'Год'})
dataLifeExpectancyProcessing =
dataLifeExpectancyProcessing.
rename(columns={'Year': 'Год'})
dataUnemploymentProcessing =
dataUnemploymentProcessing.
rename(columns={'Year': 'Год'})
dataEducationProcessing =
dataEducationProcessing.
rename(columns={'Year': 'Год'})

dataPopulationProcessing =
dataPopulationProcessing.
drop(columns='Id')
dataGDPProcessing =
dataGDPProcessing.drop(columns='Id')
dataLifeExpectancyProcessing =
dataLifeExpectancyProcessing.
drop(columns='Id')
dataUnemploymentProcessing =
dataUnemploymentProcessing.
drop(columns='Id')
dataEducationProcessing =
dataEducationProcessing.
drop(columns='Id')

logicData['Год'] = logicData['Год'].
astype(int)
logicData.info()
<class 'dask.dataframe.core.
```

```
DataFrame'>
Columns: 4 entries, Регион to Кол-во
санаториев
dtypes: object(1), float64(2),
int64(1)

# Используем merge для объединения
Dask DataFrames по столбцам
logicData = dd.merge(logicData,
dataPopulationProcessing, on=['Реги-
он', 'Год'], how='outer')
logicData = dd.merge(logicData,
dataGDPProcessing, on=['Регион',
'Год'], how='outer')
logicData = dd.merge(logicData,
dataLifeExpectancyProcessing,
on=['Регион', 'Год'], how='outer')
logicData = dd.merge(logicData,
dataUnemploymentProcessing, on=['Ре-
гион', 'Год'], how='outer')
logicData = dd.merge(logicData,
dataEducationProcessing, on=['Реги-
он', 'Год'], how='outer')

# Вывод информации о данных
logicData.compute()
/usr/local/lib/python3.10/dist-
packages/dask/dataframe/accessor.
py:96: UserWarning: This pattern is
interpreted as a regular expression,
and has match groups. To actually
get the groups, use str.extract.
out = getattr(getattr(obj, accessor,
obj), attr)(*args, **kwargs)
/usr/local/lib/python3.10/dist-
packages/dask/dataframe/accessor.
py:96: UserWarning: This pattern is
interpreted as a regular expression,
and has match groups. To actually
get the groups, use str.extract.
out = getattr(getattr(obj, accessor,
obj), attr)(*args, **kwargs)
/usr/local/lib/python3.10/dist-
packages/dask/dataframe/accessor.
py:96: UserWarning: This pattern is
interpreted as a regular expression,
and has match groups. To actually
get the groups, use str.extract.
out = getattr(getattr(obj, accessor,
obj), attr)(*args, **kwargs)
```

	Регион	Год	Размещений в санаториях	Кол-во санаториев	Population	GDP	LifeExpectancy	Unemployment rate	Labour force participation	Gross enrollment ratio - Primary	Gross enrollment ratio - Secondary
0	Российская Федерация	2002	4953271.0	2347.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	Российская Федерация	2003	4961015.0	2259.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	Российская Федерация	2004	5472792.0	2233.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	Российская Федерация	2005	5941198.0	2173.0	NaN	771495.0	NaN	75.0	62.0	541.6	268.6
4	Российская Федерация	2006	6084758.0	2148.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	Российская Федерация	2007	6071425.0	2118.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
6	Российская Федерация	2008	6356495.0	2147.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7	Российская Федерация	2009	5774527.0	1997.0	NaN	NaN	NaN	NaN	NaN	532.4	262.0
8	Российская Федерация	2010	5674233.0	1945.0	143.24	1539845.0	69.4	78.2	62.7	NaN	NaN
9	Российская Федерация	2011	5732863.0	1959.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
10	Российская Федерация	2012	5750682.0	1905.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
11	Российская Федерация	2013	5682543.0	1840.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN

12	Российская Федерация	2014	6087366.0	1905.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
13	Российская Федерация	2015	6100583.0	1878.0	144.67	1363482.0	72.1	77.0	60.8	622.1	304.7
14	Российская Федерация	2016	6440175.0	1832.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
15	Российская Федерация	2017	5959408.0	1809.0	NaN	NaN	73.4	NaN	NaN	NaN	NaN
16	Российская Федерация	2018	6415018.0	1755.0	NaN	1657328.0	NaN	NaN	NaN	NaN	NaN
17	Российская Федерация	2019	6704444.0	1777.0	NaN	1687450.0	NaN	NaN	NaN	636.9	312.3
18	Российская Федерация	2020	4044485.0	1752.0	145.62	1483498.0	NaN	NaN	NaN	NaN	NaN
19	Российская Федерация	2021	5992352.0	1768.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
20	Российская Федерация	2022	6561834.0	1742.0	144.71	NaN	70.1	74.2	59.0	NaN	NaN
21	Российская Федерация	1995	NaN	NaN	NaN	402295.0	NaN	NaN	NaN	NaN	NaN

```
new_column_names = {
'Population': 'Население',
'GDP': 'ВВП',
'LifeExpectancy': 'Продолжительность жизни',
'Unemployment rate': 'Уровень безработицы',
'Labour force participation': 'Участие в рабочей силе',
'Gross enrollment ratio — Primary': 'Валовая степень вовлеченности — Начальное образование',
'Gross enrollment ratio — Secondary': 'Валовая степень вовлеченности — Среднее образование'
}
```

```
# Переименование столбцов
logicData = logicData.rename(columns=new_column_names)
```

Очистка данных

Далее выполним очистку итоговых данных.

Описание и статический анализ данных

В первую очередь посмотрим на типы данных полученного датасета, количество пустых значений и статистические данные. Просмотрим информацию о типах данных.

```
Регион                                                    object
Год                                                        int64
Размещений в санаториях                                  float64
Кол-во санаториев                                        float64
Население                                                float64
ВВП                                                      float64
Продолжительность жизни                                  float64
Уровень безработицы                                      float64
Участие в рабочей силе                                   float64
Валовая степень вовлеченности — Начальное образование    float64
Валовая степень вовлеченности — Среднее образование      float64
dtype: object
```

Получим статистические показатели набора данных.

```
logicData.describe().compute()
```

	Год	Размещений в санаториях	Кол-во санаториев	Население	ВВП	Продолжительность жизни	Уровень безработицы	Участие в рабочей силе	Валовая степень вовлеченности - Начальное образование	Валовая степень вовлеченности - Среднее образование
count	22.000000	2.100000e+01	21.000000	4.000000	7.000000e+00	4.000000	4.00000	4.000000	4.00000	4.00000
mean	2011.227273	5.845784e+06	1966.142857	144.560000	1.272199e+06	71.250000	76.10000	61.125000	583.25000	286.90000
std	7.057126	6.127450e+05	190.351329	0.983294	4.920721e+05	1.833939	1.82939	1.619413	53.87668	25.27779
min	1995.000000	4.044485e+06	1742.000000	143.240000	4.022950e+05	69.400000	74.20000	59.000000	532.40000	262.00000
25%	2006.250000	5.682543e+06	1809.000000	144.312500	1.067488e+06	69.925000	74.80000	60.350000	539.30000	266.95000
50%	2011.500000	5.959408e+06	1905.000000	144.690000	1.483498e+06	71.100000	76.00000	61.400000	581.85000	286.65000
75%	2016.750000	6.100583e+06	2147.000000	144.937500	1.598586e+06	72.425000	77.30000	62.175000	625.80000	306.60000
max	2022.000000	6.704444e+06	2347.000000	145.620000	1.687450e+06	73.400000	78.20000	62.700000	636.90000	312.30000

```
logicData.shape
```

Трансформация данных и экстраполяция недостающих данных

Набор данных по-прежнему содержит отсутствующие значения в нескольких показателях. Для заполнения оставшихся пустых значений используем метод экстраполяции. Данная процедура основывается на доступных значениях данных показателей для конкретных регионов в различные периоды времени. Рассчитываем ближайший год для отсутствующего значения, а затем применяем среднее значение этого показателя для выбранного региона и найденного года.

Приведенный код демонстрирует этот процесс, где функция fill_missing_values принимает строку данных и индикатор показателя, и, если значение отсутствует, выполняет экстраполяцию на основе ближайшего года и среднего значения для соответствующего показателя.

```
logicData = logicData.compute()
def fill_missing_values(row,
indicator):
region = row['Регион']
year = row['Год']

if np.isnan(row[indicator]):
years = mergedDataRaw_filtered_
grouped.loc[mergedDataRaw_filtered_
grouped['Регион'] == region, 'Год']
nearest_year = years.iloc[(years -
year).abs().argsort()].min()

mean_value = mergedDataRaw_filtered_
grouped.loc[(mergedDataRaw_
filtered_grouped['Регион'] ==
region) & (mergedDataRaw_filtered_
grouped['Год'] == nearest_year),
indicator].values[0]

return mean_value
else:
return row[indicator]

mergedDataRaw_filtered_grouped
= logicData.groupby(['Регион',
'Год']).mean().reset_index()

logicData['Размещений в санатори-
ях'] = logicData.apply(lambda row:
fill_missing_values(row, 'Размещений
в санаториях'), axis=1)
logicData['Кол-во санаториев'] =
logicData.apply(lambda row: fill_
missing_values(row, 'Кол-во санато-
риев'), axis=1)
logicData['Население'] = logicData.
apply(lambda row: fill_missing_
values(row, 'Население'), axis=1)
logicData['ВВП'] = logicData.
apply(lambda row: fill_missing_
values(row, 'ВВП'), axis=1)
logicData['Продолжительность жизни']
= logicData.apply(lambda row: fill_
missing_values(row, 'Продолжитель-
ность жизни'), axis=1)
logicData['Уровень безработицы'] =
logicData.apply(lambda row: fill_
```

```
missing_values(row, 'Уровень безработицы'), axis=1)
logicData['Участие в рабочей силе'] = logicData.apply(lambda row: fill_missing_values(row, 'Участие в рабочей силе'), axis=1)
logicData['Валовая степень вовлеченности — Начальное образование'] = logicData.apply(lambda row: fill_missing_values(row, 'Валовая степень вовлеченности — Начальное образование'), axis=1)
logicData['Валовая степень вовлеченности — Среднее образование'] = logicData.apply(lambda row: fill_missing_values(row, 'Валовая степень вовлеченности — Среднее образование'), axis=1)
# logicData = dd.from_pandas(logicData, npartitions=1)

logicData
```

	Регион	Год	Размещений в санаториях	Кол-во санаториев	Население	ВВП	Продолжительность жизни	Уровень безработицы	Участие в рабочей силе	Валовая степень вовлеченности – Начальное образование	Валовая степень вовлеченности – Среднее образование
0	Российская Федерация	2002	4953271.0	2347.0	144.71	402295.0	70.1	75.0	62.0	541.6	268.6
1	Российская Федерация	2003	4961015.0	2259.0	144.71	402295.0	70.1	75.0	62.0	541.6	268.6
2	Российская Федерация	2004	5472792.0	2233.0	144.71	402295.0	70.1	75.0	62.0	541.6	268.6
3	Российская Федерация	2005	5941198.0	2173.0	144.71	771495.0	70.1	75.0	62.0	541.6	268.6
4	Российская Федерация	2006	6084758.0	2148.0	144.71	402295.0	70.1	75.0	62.0	541.6	268.6
5	Российская Федерация	2007	6071425.0	2118.0	144.71	402295.0	70.1	75.0	62.0	541.6	268.6
6	Российская Федерация	2008	6356495.0	2147.0	144.71	402295.0	70.1	75.0	62.0	541.6	268.6
7	Российская Федерация	2009	5774527.0	1997.0	144.71	402295.0	70.1	75.0	62.0	532.4	262.0
8	Российская Федерация	2010	5674233.0	1945.0	143.24	1539845.0	69.4	78.2	62.7	541.6	268.6
9	Российская Федерация	2011	5732863.0	1959.0	144.71	402295.0	70.1	75.0	62.0	541.6	268.6
10	Российская Федерация	2012	5750682.0	1905.0	144.71	402295.0	70.1	75.0	62.0	541.6	268.6
11	Российская Федерация	2013	5682543.0	1840.0	144.71	402295.0	70.1	75.0	62.0	541.6	268.6
12	Российская Федерация	2014	6087366.0	1905.0	144.71	402295.0	70.1	75.0	62.0	541.6	268.6
13	Российская Федерация	2015	6100583.0	1878.0	144.67	1363482.0	72.1	77.0	60.8	622.1	304.7
14	Российская Федерация	2016	6440175.0	1832.0	144.71	402295.0	70.1	75.0	62.0	541.6	268.6
15	Российская Федерация	2017	5959408.0	1809.0	144.71	402295.0	73.4	75.0	62.0	541.6	268.6
16	Российская Федерация	2018	6415018.0	1755.0	144.71	1657328.0	70.1	75.0	62.0	541.6	268.6
17	Российская Федерация	2019	6704444.0	1777.0	144.71	1687450.0	70.1	75.0	62.0	636.9	312.3
18	Российская Федерация	2020	4044485.0	1752.0	145.62	1483498.0	70.1	75.0	62.0	541.6	268.6
19	Российская Федерация	2021	5992352.0	1768.0	144.71	402295.0	70.1	75.0	62.0	541.6	268.6
20	Российская Федерация	2022	6561834.0	1742.0	144.71	402295.0	70.1	74.2	59.0	541.6	268.6
21	Российская Федерация	1995	5941198.0	2173.0	144.71	402295.0	70.1	75.0	62.0	541.6	268.6

```
logicData = dd.from_pandas(logicData, npartitions=1)
```

Исследование полученного набора данных

Визуализируем полученные данные для оценки зависимостей и корреляций.

Корреляционная матрица

Корреляционная матрица (Heatmap) является мощным инструментом для визуализации взаимосвязей между различными показателями в наборе данных.

Каждая ячейка в тепловой карте представляет собой значение корреляции между двумя соответствующими показателями. Значение корреляции может находиться в диапазоне от −1 до 1, где−1 указывает на полностью обратную корреляцию, 1 — на полностью прямую корреляцию, а 0 — на отсутствие корреляции.

Чем ярче цвет ячейки в тепловой карте, тем более сильная и положительная корреляция между соответствующими показателями. Напротив, чем темнее цвет, тем более сильная и отрицательная корреляция. Если цвет ячейки близок к белому, то это указывает на незначительную или отсутствующую корреляцию между показателями (рис. 1).

```
# Выбор необходимых столбцов для анализа корреляции
columns = [
'Размещений в санаториях',
'Кол-во санаториев',
'Население',
'ВВП',
'Продолжительность жизни',
'Уровень безработицы',
'Участие в рабочей силе',
'Валовая степень вовлеченности — Начальное образование',
'Валовая степень вовлеченности — Среднее образование'
]
correlation = logicData[columns].corr()

# Построение корреляционной матрицы в виде тепловой карты
sns.heatmap(correlation, annot=True, cmap='coolwarm')
plt.title('Корреляционная матрица')
plt.show()
```

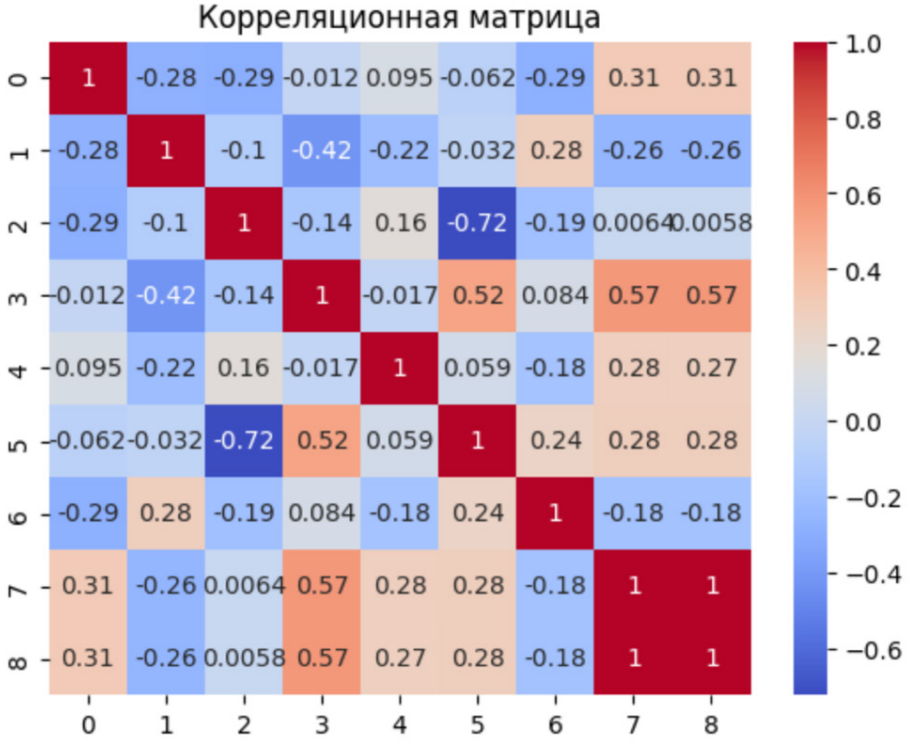

Рис. 1. Корреляционная матрица взаимосвязей между различными показателями в наборе данных

Матрица рассеяния

Часто бывает полезно изучить матрицу рассеяния, чтобы получить представление о взаимосвязях между переменными. Для некоторых переменных уже есть представление о том, как они должны соотноситься друг с другом, и несоответствие этому представлению, может означать проблему с данными. Проверка связей является хорошей проверкой на вменяемость, чтобы убедиться, что собранный набор данных имеет смысл.

Здесь видна значительная корреляция между показателями. Для ее подробного изучение используется матрица рассеяния. На графике (рис. 2) каждая точка представляет собой отдельное наблюдение, где одна координата соответствует значению одного показателя, а другая координата — значению другого показателя.

```
scatterplot = pd.plotting.scatter_
matrix(logicData.compute().
loc[:, columns], figsize=(13, 11),
diagonal='kde')
#plt.tight_layout() # Распределить
графики равномерно на поле


# Изменение названий показателей
n = len(columns)
for i in range(n):
for j in range(n):
ax = scatterplot[i, j]
ax.yaxis.label.set_rotation(0) # По-
ворот названия по вертикали
ax.yaxis.labelpad = 50 # Увеличение
пространства между названием и гра-
фиком
ax.xaxis.label.set_rotation(90) #
Поворот названия по горизонтали
ax.xaxis.labelpad = 20 # Увеличение
пространства между названием и гра-
фиком

plt.show()
```

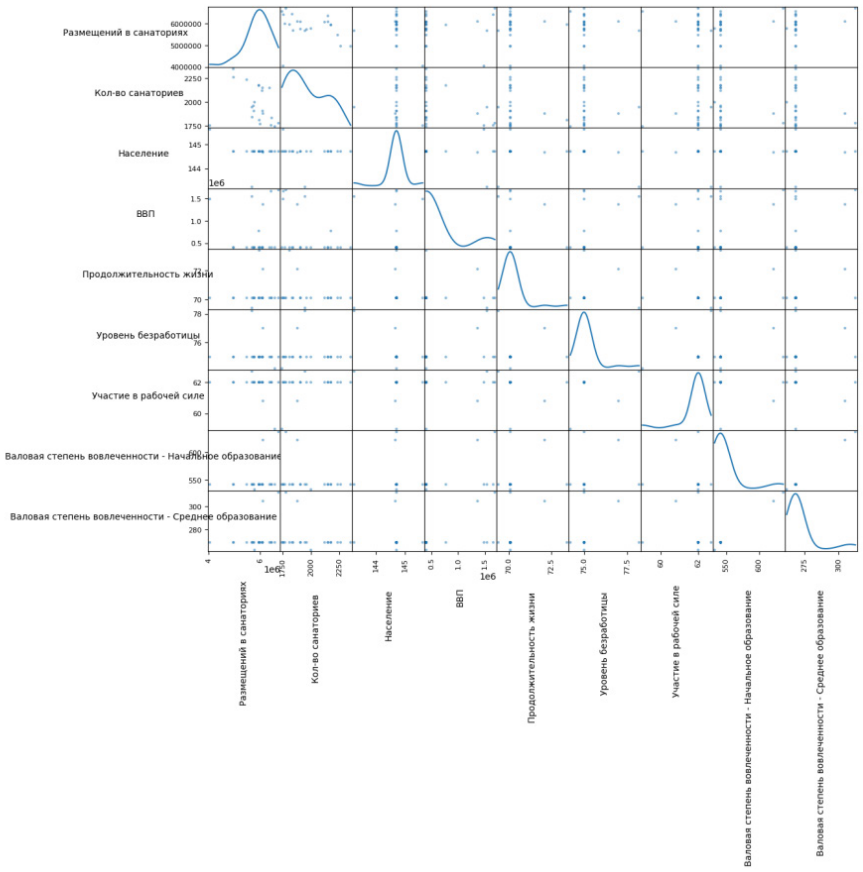

Рис. 2. Матрица рассеяния.

Построение моделей Машиного обучения на полученных данных

Исследование проводится с целью оптимизации параметров модели для прогнозирования демографических показателей. Процесс оптимизации включает в себя поиск оптимальных комбинаций параметров модели, а также определение ключевых факторов, используемых для предсказания.

Необходимо осуществить подготовку данных, предварительно обработанных и готовых для моделирования. Затем выбирается целевой демографический показатель и определяются параметры модели, такие как гиперпараметры алгоритмов. Следующим шагом будет систематический перебор комбинаций параметров с использованием методов кросс-валидации для оценки производительности.

После этого будет проводиться оценка каждой модели с использованием различных метрик, таких как точность, полнота, F1-мера и другие. Будет проанализировано влияние различных факторов на предсказываемый демографический показатель с выделением наиболее важных. Завершен процесс будет оптимизацией модели в целях достижения наилучшей производительности.

Этот подход позволяет построить эффективную модель машинного обучения, способную точно предсказывать демографические показатели, а также выявит важные факторы, влияющие на прогноз демографических показателей.

```
import itertools
from sklearn.ensemble import
RandomForestRegressor
from sklearn.metrics import mean_
absolute_error, mean_squared_error,
r2_score

# Список всех показателей
columns = [
'Год',
'Размещений в санаториях',
'Кол-во санаториев',
'Население',
'ВВП',
'Продолжительность жизни',
'Уровень безработицы',
'Участие в рабочей силе',
'Валовая степень вовлеченности — На-
чальное образование',
'Валовая степень вовлеченности —
Среднее образование'
]

# Создание всех возможных комбинаций
показателей для предсказания
prediction_combinations = []
for r in range(1, len(columns) + 1):
prediction_combinations.
extend(itertools.
combinations(columns, r))

# Создание всех возможных комбинаций
показателей по которым идет предска-
зание
target_combinations = []
for r in range(1, len(columns) + 1):
target_combinations.extend(itertools.
combinations(columns, r))

# Создание словаря для хранения ре-
зультатов моделей
results = {}

# Итерация по всем комбинациям пока-
зателей
for prediction_features in
prediction_combinations:
for target_features in target_
combinations:
# Пропуск комбинации, если показате-
ли предсказания и показатели целевых
значений пересекаются
if set(prediction_features).
intersection(set(target_features)):
continue

# Пропуск комбинации, если показате-
ли предсказания и показатели целевых
значений пересекаются
if set(prediction_features).
intersection(set(target_features)):
continue
```

```
# Пропуск комбинации, если количество признаков меньше 3
if len(prediction_features) > 3 or len(target_features) < 5:
continue

# Выбор соответствующих столбцов из набора данных
selected_columns = list(prediction_features) + list(target_features)
data = logicData.loc[:, selected_columns]

# Преобразование категориальных признаков в числовые с помощью LabelEncoder, если необходимо

# Разделение данных на обучающую и тестовую выборки
X = data[list(prediction_features)]
y = data[list(target_features)]
X_train, X_test, y_train, y_test = train_test_split(X.compute(), y.compute(), test_size=0.2, random_state=42)

#print(X_train)

# Создание и обучение модели случайного леса
model = RandomForestRegressor()
model.fit(X_train, y_train)

# Прогнозирование на тестовой выборке
y_pred = model.predict(X_test)

# Вычисление метрик оценки модели
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

#print({'MAE': mae, 'MSE': mse, 'R^2': r2})

# Сохранение результатов в словаре
results[(prediction_features, target_features)] = {'MAE': mae, 'MSE': mse, 'R^2': r2}

# Сортировка результатов по коэффициенту детерминации (R^2) в порядке убывания
sorted_results = sorted(results.items(), key=lambda x: x[1]['R^2'], reverse=False)

# Вывод результатов
for combination, metrics in sorted_results:
if metrics['R^2'] < 0:
continue
prediction_features, target_features = combination
print(f»Показатели предсказания: {prediction_features}»)
print(f»Показатели целевых значений: {target_features}»)
print(f»MAE: {metrics['MAE']}»)
print(f»MSE: {metrics['MSE']}»)
print(f»R^2: {metrics['R^2']}»)
print()
```

Выходные данные были обрезаны до нескольких последних строк (5000).

R^2: 0.03793322095101448

Показатели предсказания: ('Уровень безработицы', 'Валовая степень вовлеченности — Среднее образование')

Показатели целевых значений: ('Год', 'Размещений в санаториях', 'Кол-во санаториев', 'Население', 'ВВП', 'Участие в рабочей силе', 'Валовая степень вовлеченности — Начальное образование')

MAE: 114473.15888904207

MSE: 82038998608.7286

R^2: 0.038143687851828326

Показатели предсказания: ('Продолжительность жизни', 'Участие в рабочей силе',

'Валовая степень вовлеченности — Среднее образование')

Показатели целевых значений: ('Год', 'Кол-во санаториев', 'Население', 'ВВП', 'Уровень безработицы', 'Валовая степень вовлеченности — Начальное образование')

MAE: 51219.90627384959

MSE: 33943958104.468643

R^2: 0.03823791179228072

Показатели предсказания: ('Участие в рабочей силе', 'Валовая степень вовлеченности — Начальное образование')

Показатели целевых значений: ('Год', 'Размещений в санаториях', 'Кол-во санаториев', 'Население', 'ВВП', 'Валовая степень вовлеченности — Среднее образование')

MAE: 133736.4559472756

MSE: 93247933697.29176

R^2: 0.03832238277206237

Вывод

Необходимо отметить, что несмотря на систематический анализ и поиск оптимальных комбинаций показателей для предсказания демографических характеристик, не были обнаружены такие комбинаций, которые бы существенно улучшили точность прогнозирования. Возможные причины этого могут включать в себя сложность взаимодействия между различными факторами, недостаточное количество данных для устойчивого обучения, а также влияние внешних факторов, не учтенных в данном исследовании.

Несмотря на отсутствие выявленных оптимальных комбинаций, процесс исследования позволил более глубоко понять влияние различных параметров на демографические показатели. Важно отметить, что в контексте сложных социоэкономических процессов часто требуется учет множества факторов и их динамических взаимодействий.

Данное исследование выявило необходимость дальнейших исследований и уточнения методологии с учетом возможных ограничений данных. Перспективы включают расширение объема данных, углубленный анализ влияния временных и географических изменений, а также применение более сложных моделей машинного обучения.

Полученные результаты позволяют констатировать факт эффективности применения методов машинного обучения для прогнозирования демографических процессов.

Литература

1. Pedersen, U. T. (2023). Python Pandas vs. Dask DataFrames: A Comparative Analysis. Towards AI. // Электронный ресурс, July 17, 2023 // URL: https://towardsai.net/p/l/python-pandas-vs-dask-dataframes-a-comparative-analysis
2. McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. O'Reilly Media, Inc.
3. Pandas Development Team. (2022). Pandas: Powerful data structures for data analysis, time series, and statistics. // Электронный ресурс, 2022 // URL: https://pandas.pydata.org/
4. McKinney, W., & others. (2010). Data structures for statistical computing in Python. In Proceedings of the 9th Python in Science Conference (pp. 51–56). // Электронный ресурс, June 28, 2010 // URL: https://proceedings.scipy.org/articles/Majora-92bf1922–00a
5. Pandas Development Team. (2022). Pandas: Powerful data structures for data analysis, time series, and statistics. Retrieved from https://pandas.pydata.org/
6. McKinney, W. (2011). pandas: a foundational Python library for data analysis and statistics. Python for High Performance and Scientific Computing, 14, 1–9. // Электронный ресурс, January, 2011 // URL: https://www.researchgate.net/publication/265194455_pandas_a_Foundational_Python_Library_for_Data_Analysis_and_Statistics
7. Dask Development Team. (2022). Dask: Parallel Computing with Task Scheduling. // Электронный ресурс, 2022 // URL: https://dask.org/
8. Pandas Development Team. (2022). Pandas: Powerful data structures for data analysis, time

series, and statistics. // Электронный ресурс, 2022 // URL: https://pandas.pydata.org/
9. Adnan, K., Akbar, R. (2019). An analytical study of information extraction from unstructured and multidimensional big data. Journal of Big Data 6(1), 91. // Электронный ресурс, 2019 // URL: https://journalofbigdata.springeropen.com/articles/10.1186/s40537–019–0254–8
10. Rocklin, M. (2015). Dask: Parallel computation with blocked algorithms and task scheduling. In: Python in Science Conference. pp. 126–132. // Электронный ресурс, January, 2015 // URL: https://www.researchgate.net/publication/328778461_Dask_Parallel_Computation_with_Blocked_algorithms_and_Task_Scheduling
11. Salvador, S., Chan, P. (2004). Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In: 16th IEEE International Conference on Tools with Artificial Intelligence. pp. 576–584. // Электронный ресурс, January 10, 2005 // URL: https://ieeexplore.ieee.org/document/1374239
12. Zambelli, A. (2016). A data-driven approach to estimating the number of clusters in hierarchical clustering. F1000Research 5. // Электронный ресурс, December 1, 2016 // URL: https://f1000research.com/articles/5–2809

Identifying Optimal Factors for Predicting Demographic Indicators Using Machine Learning Methods

Gorbas D. A., Bulgakov A. L., Milyutin M. A.
Lomonosov Moscow State University; Plekhanov Russian University of Economics, Moscow Institute of Modern Academic Education

The issues of forecasting demographic indicators are relevant for the development of society and the economy. The data obtained on the basis of the forecast of demographic indicators allow to adjust the state policy in the field of employment and economic development. Different methods are used to forecast demographic indicators. The article attempts to study the issue of forecasting demographic indicators using machine learning methods, which can be used as a powerful tool for analyzing large amounts of data and identifying optimal factors affecting population dynamics.

Keywords: machine learning, demographic indicators, forecasting, data processing, factor optimization.

References

1. Pedersen, U. T. (2023). Python Pandas vs. Dask DataFrames: A Comparative Analysis. Towards AI. // Electronic resource, July 17, 2023 // URL: https://towardsai.net/p/l/python-pandas-vs-dask-dataframes-a-comparative-analysis
2. McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. O'Reilly Media, Inc.
3. Pandas Development Team. (2022). Pandas: Powerful data structures for data analysis, time series, and statistics. // Electronic resource, 2022 // URL: https://pandas.pydata.org/
4. McKinney, W., & others. (2010). Data structures for statistical computing in Python. In Proceedings of the 9th Python in Science Conference (pp. 51–56). // Electronic resource, June 28, 2010 // URL: https://proceedings.scipy.org/articles/Majora-92bf1922–00a
5. Pandas Development Team. (2022). Pandas: Powerful data structures for data analysis, time series, and statistics. Retrieved from https://pandas.pydata.org/
6. McKinney, W. (2011). pandas: a fundamental Python library for data analysis and statistics. Python for High Performance and Scientific Computing, 14, 1–9. // Electronic resource, January, 2011 // URL: https://www.researchgate.net/publication/265194455_pandas_a_Foundational_Python_Library_for_Data_Analysis_and_Statistics
7. Dask Development Team. (2022). Dask: Parallel Computing with Task Scheduling. // Electronic resource, 2022 // URL: https://dask.org/
8. Pandas Development Team. (2022). Pandas: Powerful data structures for data analysis, time series, and statistics. // Electronic resource, 2022 // URL: https://pandas.pydata.org/
9. Adnan, K., Akbar, R. (2019). An analytical study of information extraction from unstructured and multidimensional big data. Journal of Big Data 6(1), 91. // Electronic resource, 2019 // URL: https://journalofbigdata.springeropen.com/articles/10.1186/s40537–019–0254–8
10. Rocklin, M. (2015). Dask: Parallel computation with blocked algorithms and task scheduling. In: Python in Science Conference. pp. 126–132. // Electronic resource, January, 2015 // URL: https://www.researchgate.net/publication/328778461_Dask_Parallel_Computation_with_Blocked_algorithms_and_Task_Scheduling
11. Salvador, S., Chan, P. (2004). Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In: 16th IEEE International Conference on Tools with Artificial Intelligence. pp. 576–584. // Electronic resource, January 10, 2005 // URL: https://ieeexplore.ieee.org/document/1374239
12. Zambelli, A. (2016). A data-driven approach to estimating the number of clusters in hierarchical clustering. F1000Research 5. // Electronic resource, December 1, 2016 // URL: https://f1000research.com/articles/5–2809

Сравнительный анализ моделей прогнозирования кредитоспособности

Демидов Алексей Дмитриевич
Магистр, факультет Высшая школа кибертехнологий, математики и статистики, РЭУ имени Г. В. Плеханова.
E-mail: demidov.ad@bk.ru

Алешина Анна Валентиновна
К.э.н., доцент МГУ имени М. В. Ломоносова
E-mail: bulgakoval@my.msu.ru

Милютин Максим Александрович
Студент, Экономический факультет МГУ имени М. В. Ломоносова
E-mail: milyutinma@my.msu.ru

Процесс оценки кредитоспособности заявителей является ключевым элементом деятельности банковских учреждений. От точности этого процесса зависит общая эффективность кредитной политики, уровень рисков и прибыльность организации. В настоящее время с расширением доступных данных и прогрессом в аналитике появилась возможность использовать современные методы машинного обучения для детального и точного прогнозирования финансового поведения заявителей.
В работе рассматривается набор данных, отражающий поведение потенциальных клиентов с учетом современных исследований рынка ипотечного кредитования в Российской Федерации (Далингер, 2021), который был подробно проанализирован. Выявлены факторы, оказывающие наибольшее влияние на вероятность получения кредита, а также оценено качество различных алгоритмов классификации, таких как логистическая регрессия, опорная векторная машина (SVM), XGBoost и случайный лес. Результаты позволили сравнить точность моделей, а корреляционная карта и визуализация позволили выделить ключевые показатели, способствующие принятию решений.
На основе полученных результатов сделаны выводы о преимуществах той или иной модели, а также определены рекомендации по повышению эффективности кредитной политики и управления рисками в банковской сфере.

Ключевые слова: скоринговые системы кредитования.

Введение

Эффективное прогнозирование кредитоспособности клиентов является одной из ключевых задач банковских учреждений, поскольку не только снижает риски, но и повышает доходность кредитной политики. В первой части исследования был проведен детальный анализ данных, включая изучение их структуры, устранение аномалий, балансировку классов и подготовку признаков для построения моделей машинного обучения. Эти шаги заложили прочную основу для применения современных алгоритмов и оценки их эффективности при решении задач кредитного скоринга [1–4].

Процесс принятия решений о выдаче кредита традиционно основывался на классических статистических методах и экспертной оценке. Однако растущий объем данных, их высокая сложность и наличие нелинейных зависимостей требуют более современных подходов. Алгоритмы машинного обучения, такие как ансамблевые методы (например, Random Forest и XGBoost), позволяют обрабатывать большие наборы данных, выявлять скрытые закономерности и создавать модели, способные с высокой точностью прогнозировать вероятность принятия кредитного предложения.

Настоящая работа продолжает исследование, сосредотачиваясь на сравнительном анализе производительности различных моделей машинного обучения. Рассматриваются такие популярные алгоритмы, как логистическая ре-

грессия, машина опорных векторов (SVM), случайный лес и градиентный бустинг. Основной целью является определение наиболее точной и устойчивой модели для прогнозирования финансового поведения клиентов. Для достижения этой цели проведена всесторонняя оценка моделей с использованием метрик точности, F1-меры, матриц ошибок и перекрестной проверки.

Полученные результаты позволяют не только выбрать наиболее подходящий метод для решения задач кредитного скоринга, но и сформировать рекомендации для внедрения этих моделей в практику банковского дела. Настоящая часть исследования демонстрирует значимость современных алгоритмов машинного обучения для повышения эффективности работы финансовых организаций в условиях сложных и изменчивых рыночных реалий.

Обучение модели и оценка модели

Обучение модели

- Обучение модели с различными алгоритмами классификации
- Сравнение производительности и точности различных алгоритмов
- Вычисление баллов перекрестной проверки и их сравнение

Импорт всех библиотек, необходимых для обучения модели.

```
from sklearn.linear_model import LogisticRegression #логистическая регрессия
from sklearn import svm #машина опорных векторов
from sklearn.ensemble import RandomForestClassifier #метод случайного леса
from sklearn.neighbors import KNeighborsClassifier #KNN
from sklearn.naive_bayes import GaussianNB #Naive bayes
from sklearn.tree import DecisionTreeClassifier #дерево решений
from sklearn.model_selection import train_test_split #разделение данных обучения и тестирования
from sklearn import metrics #мера точности
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.preprocessing import StandardScaler
```

Логистическая регрессия

Блок с отступами.

```
lr= LogisticRegression()
lr.fit(X_train_s, Y_train)
Y_pred_lr = lr.predict(X_test_s)
print(«Проверка точности логистической регрессии: «, metrics.accuracy_score(Y_test, Y_pred_lr))
print(«Точность обучения логистической регрессии: «, lr.score(X_train_s, Y_train))
```

```
Проверка точности логистической регрессии:  0.7920255183413079
Точность обучения логистической регрессии:  0.7964178288214383
```

SVM (машина опорных векторов)

```
svm=svm.SVC(kernel='linear')

svm.fit(X_train_s, Y_train)
Y_pred_svm = svm.predict(X_test_s)

print(«Проверка точности SVM: «, metrics.accuracy_score(Y_test, Y_pred_svm))

print(«Точность обучения SVM: «, svm.score(X_train_s, Y_train))
```

```
Проверка точности SVM:
0.794896331738437
Точность обучения SVM:
0.7972381733661471
```

Древо решений

```
dt= DecisionTreeClassifier()

dt.fit(X_train_s, Y_train)
Y_pred_dt = dt.predict(X_test_s)

print(«Проверка точности Decision Tree:
«, metrics.accuracy_score(Y_test, Y_
pred_dt))

print(«Точность обучения Decision Tree:
«, dt.score(X_train_s, Y_train))
```

```
Проверка точности Decision Tree:
0.7891547049441786
Точность обучения Decision Tree:  1.0
```

К-ближайших соседей (KNN)

```
knn= KNeighborsClassifier(n_neighbors=5)

knn.fit(X_train_s, Y_train)
Y_pred_knn = knn.predict(X_test_s)

print(«Проверка точности KNN: «,
metrics.accuracy_score(Y_test, Y_pred_
knn))

print(«Точность обучения KNN: «, knn.
score(X_train_s, Y_train))
```

```
Проверка точности KNN:
0.7333333333333333
Точность обучения KNN:
0.8188405797101449
```

```
l = list(range(1,25,2))
for i in l:
  knn1= KNeighborsClassifier(n_
neighbors=i)
  knn1.fit(X_train_s, Y_train)
  Y_pred = knn1.predict(X_test_s)
  accuracy = metrics.accuracy_score(Y_
test, Y_pred)
  train_acc = knn1.score(X_train_s, Y_
train)
  print(f»Для K={i} точность теста
{accuracy}:».format(i, accuracy))
  print(f»Для K={i} точность обучения
{accuracy}:».format(i, train_acc))
  print()
```

```
   Для K=1 точность теста
0.7087719298245614:
Для K=1 точность обучения
0.7087719298245614:

Для K=3 точность теста
0.727591706539075:
Для K=3 точность обучения
0.727591706539075:

Для K=5 точность теста
0.7333333333333333:
Для K=5 точность обучения
0.7333333333333333:

Для K=7 точность теста
0.7314194577352472:
Для K=7 точность обучения
0.7314194577352472:

Для K=9 точность теста
0.7323763955342902:
Для K=9 точность обучения
0.7323763955342902:

Для K=11 точность теста
0.7355661881977671:
Для K=11 точность обучения
0.7355661881977671:

Для K=13 точность теста
0.732695374800638:
Для K=13 точность обучения
0.732695374800638:

Для K=15 точность теста
0.7349282296650718:
Для K=15 точность обучения
0.7349282296650718:

Для K=17 точность теста
0.7298245614035088:
Для K=17 точность обучения
0.7298245614035088:
```

```
Для K=19 точность теста
0.7330143540669857:
Для K=19 точность обучения
0.7330143540669857:

Для K=21 точность теста
0.7282296650717703:
Для K=21 точность обучения
0.7282296650717703:

Для K=23 точность теста
0.7291866028708134:
Для K=23 точность обучения
0.7291866028708134:
```

Наивный байесовский классификатор

```
nb= GaussianNB()

nb.fit(X_train_s, Y_train)
Y_pred_nb = nb.predict(X_test_s)

print(«Проверка точности Naive
Bayes: «, metrics.accuracy_score(Y_
test, Y_pred_knn))

print(«Точность обучения Naive
Bayes: «, nb.score(X_train_s, Y_
train))
Проверка точности Naive Bayes:
0.7333333333333333
Точность обучения Naive Bayes:
0.7083675143560295
```

Случайный лес

```
rf = RandomForestClassifier(n_
estimators=200)

rf.fit(X_train, Y_train)
Y_pred_rf = rf.predict(X_test)

print(«Проверка точности RF: «,
metrics.accuracy_score(Y_test, Y_
pred_rf))

print(«Точность обучения RF: «,
rf.score(X_train, Y_train))
Проверка точности RF:
0.8373205741626795
Точность обучения RF:  1.0
```

Градиентный бустинг

```
from sklearn.ensemble import
GradientBoostingClassifier
gb = GradientBoostingClassifier(rand
om_state=5)

gb.fit(X_train_s, Y_train.squeeze().
values)
y_train_preds = gb.predict(X_
train_s)
y_test_preds = gb.predict(X_test_s)

print('Проверка точности GB: ',
metrics.accuracy_score(Y_test, y_
test_preds))
print('Точность обучения GB:',
metrics.accuracy_score(y_train_
preds, Y_train))
Проверка точности GB:
0.835725677830941
Точность обучения GB:
0.8538419469510528
```

XG Boost

```
import xgboost
xgb = xgboost.XGBClassifier(n_
estimators=80, learning_rate=0.1,
gamma=0, subsample=0.75,
                            colsample_
bytree=1, max_depth=5)
xgb.fit(X_train_s, Y_train.
squeeze().values)

y_train_preds = xgb.predict(X_
train_s)
y_test_preds = xgb.predict(X_test_s)
print(xgb.score(X_test_s, Y_test))
print(xgb.score(X_train_s, Y_train))
0.8465709728867623
0.8796828001093793
print('Проверка точности XGB: ',
```

```
metrics.accuracy_score(Y_test, y_
test_preds))
print('Точность обучения XGB: ',
metrics.accuracy_score(y_train_
preds, Y_train))
Проверка точности XGB:
0.8465709728867623
Точность обучения XGB:
0.8796828001093793
print(classification_report(Y_test,
y_test_preds))
print(confusion_matrix(Y_test, y_
test_preds))
              precision    recall
f1-score   support

           0       0.87      0.83
0.85      1615
           1       0.82      0.87
0.85      1520

    accuracy
0.85      3135
   macro avg       0.85      0.85
0.85      3135
weighted avg       0.85      0.85
0.85      3135

[[1335  280]
[2011319]]
```

Точность XG Boost очень высокая без переобучения.

Матрица ошибок

Матрица ошибок — это метод обобщения производительности алгоритма классификации.

Вычисление матрицы путаницы может дать вам лучшее представление о том, что делает ваша модель классификации правильно и какие типы ошибок она допускает.

Точность Precision Recall and F1 Score?

Accuracy является мерой общей эффективности модели классификации и представляет собой отношение правильно предсказанного наблюдения к общему количеству наблюдений.

Precision — это отношение правильно спрогнозированных положительных наблюдений к общему количеству спрогнозированных положительных наблюдений.

Recall (Sensitivity) — это отношение правильно предсказанных положительных наблюдений ко всем наблюдениям в классе.

F1 score — представляет собой средневзвешенное значение Accuracy и Precision. Таким образом, эта оценка учитывает как ложноположительные, так и ложноотрицательные результаты.

Матрица ошибок XG Boost

```
plt.figure(figsize=(7,5))
sns.heatmap(confusion_matrix(Y_test, y_
test_preds), annot=True, cmap=»OrRd_r»,
          fmt=»d», cbar=True,
xticklabels=['Yes','No'],
yticklabels=['Yes','No'],
          annot_kws={«fontsize»:15})
plt.show() (см. рис. 1)
```

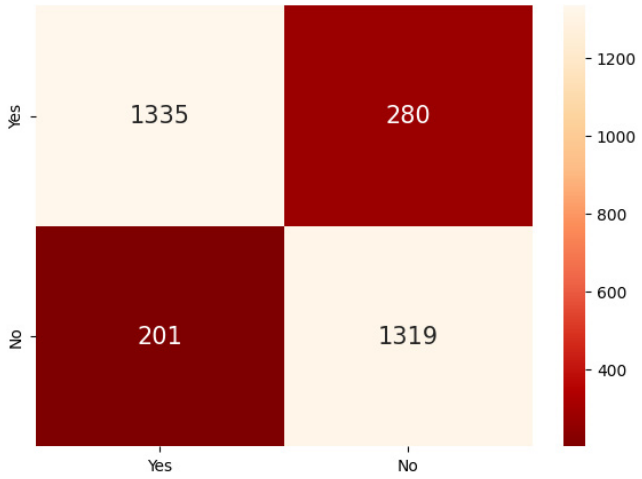

Рис. 1. Матрица ошибок для алгоритма XGBoost

Важность показателей для XGBoost

```
# используя случайный лес здесь,
```

```
чтобы получить важность показателей
plt.figure(figsize=(8,6))
importances= xgb.feature_importances_
feature_importances=
pd.Series(importances, index=X_train.
columns).sort_values(ascending=False)
sns.barplot(x=feature_
importances[:15], y=feature_
importances.index[:15],
palette=»rocket»)
sns.despine()
plt.xlabel(«Важность показателя»)
plt.ylabel(«Показатель»)
plt.show() (см. рис. 2)
```

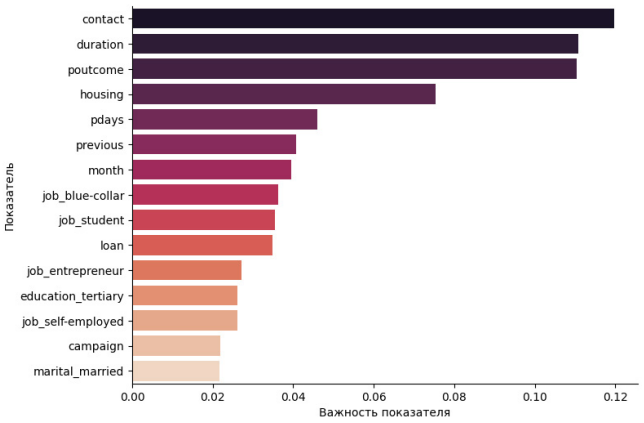

Рис. 2. Важность показателей для модели XGBoost по анализу данных

Случайный лес

```
rf1 =RandomForestClassifier(random_
state=0, n_estimators=200, max_
features=25, max_depth=10, min_
samples_leaf=50)
rf1.fit(X_train_s, Y_train.
squeeze().values)

#рассчитать и выведем для оценки
по 15 основным показателям
y_train_preds = rf1.predict(X_
train_s)
y_test_preds = rf1.predict(X_test_s)
print(rf1.score(X_test_s, Y_test))
print(rf1.score(X_train_s, Y_train))
0.8185007974481658
0.831555920153131
print(classification_report(Y_test,
y_test_preds))
print(confusion_matrix(Y_test, y_
test_preds))
```

```
              precision    recall
f1-score   support

           0       0.83      0.82
0.82      1615
           1       0.81      0.82
0.81      1520

    accuracy
0.82      3135
   macro avg       0.82      0.82
0.82      3135
weighted avg       0.82      0.82
0.82      3135

[[1321  294]
[2751245]]
```

Матрица ошибок для Случайного леса

```
plt.figure(figsize=(7,5))
sns.heatmap(confusion_matrix(Y_
test, y_test_preds), annot=True,
cmap=»OrRd_r»,
            ФИНАНСЫ ФИНАНСЫ fmt=»d»,
cbar=True, xticklabels=['Yes','No'],
yticklabels=['Yes','No'],
            annot_kws={«fontsize»:15})
plt.show() (см. рис. 3)
```

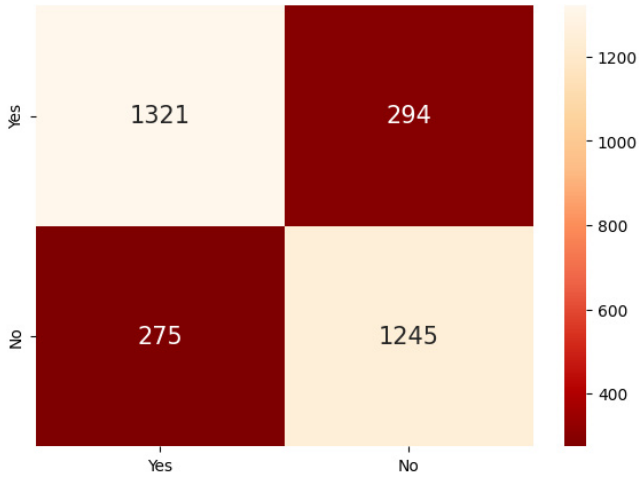

Рис. 3. Матрица ошибок для модели «Случайный лес»

Важность показателей для Случайного леса

```
plt.figure(figsize=(8,6))
importances= rf1.feature_importances_
feature_importances=
pd.Series(importances, index=X_train.
columns).sort_values(ascending=False)
sns.barplot(x=feature_
importances[0:10], y=feature_
importances.index[0:10],
palette=»rocket»)
sns.despine()
plt.xlabel(«Важность показателя»)
plt.ylabel(«Показатель»)
plt.show() (см. рис. 4)
```

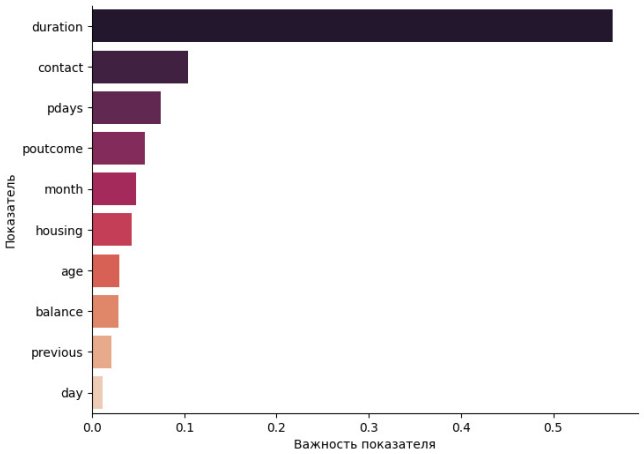

Рис. 4. Важность показателей для модели «Случайный лес» на основе обучающих данных

Вывод

Самые важные показатели для XGBoost и метода Случайного леса — это:
1) Duration
2) contact
3) poutcomes

Точность перекресной проверки

```
## XG Boost
gs_svm_scores = cross_val_score(xgb,
X=X_train_s, y=Y_train, cv=5,
scoring='accuracy', n_jobs= -1)
print('Точность перекрестной:
{0:.1f}%'.format(np.mean(gs_svm_
scores)*100))
Точность перекрестной: 84.6%
## Случайный лес
gs_svm_scores = cross_val_score(rf1,
X=X_train_s, y=Y_train, cv=5,
scoring='accuracy', n_jobs= -1)
print('Точность перекрестной:
{0:.1f}%'.format(np.mean(gs_svm_
scores)*100))
Точность перекрестной: 81.8%
```

Вывод

Точность перекрестной проверки выше у XG Boost.

Результаты сравнения моделей показали, что ансамблевые методы, такие как XGBoost и Random Forest, продемонстрировали наивысшую точность и F1-Score. Это связано с их способностью эффективно обрабатывать сложные нелинейные зависимости и комбинировать предсказания множества слабых моделей для достижения более высокого качества результата.

Почему XGBoost лучше линейных моделей? Нелинейные зависимости: Линейные модели, такие как логистическая регрессия и SVM, предполагают наличие линейных зависимостей между признаками и целевой переменной, что ограничивает их эффективность при сложных паттернах в данных. XGBoost, в свою очередь, способен выявлять как линейные, так и нелинейные связи. Работа с признаками: Ансамблевые методы автоматически учитывают важность признаков и взаимодействие между ними, что делает их более гибкими при наличии большого количества переменных. Устойчивость к выбросам: XGBoost и Random Forest менее чувствительны к выбросам благодаря использованию деревьев решений и механизмов, сглаживающих влияние аномалий. Время обучения и сложность Несмотря на высокую точность, ансамблевые методы требуют больше времени на обучение по сравнению с линейными моделями и KNN.

Например, XGBoost обучается в среднем 10 секунд, в то время как логистическая регрессия требует менее 2 секунд. Это связано с построением большого количества деревьев и их оптимизацией. Однако увеличение времени обучения оправдывается улучшением качества предсказаний.

Сравнение с деревьями решений Простое дерево решений продемонстрировало хорошую скорость обучения, но значительно уступило в точности из-за переобучения на обучающих данных. В ансамблевых методах, таких как Random Forest и XGBoost, переобучение контролируется за счёт усреднения предсказаний множества деревьев.

Таким образом, ансамблевые модели, особенно XGBoost, оказались наиболее эффективными для данной задачи, обеспечивая баланс между точностью, устойчивостью и возможностью выявления сложных зависимостей в данных, что согласуется с выводами о применении современных подходов к кредитованию малого и среднего бизнеса (Предет, 2024).

Пороговые значения для параметров

```
Balance
#создадим один dataframe df3
df3 = pd.DataFrame()
df3['balance']=df['balance']

df3['deposit']=df['deposit']
df3['balance_quantile'] =
pd.qcut(df3['balance'], q=25,
labels=False, duplicates ='drop')

#сгруппируем по 'balance_buckets' и
найдем средний результат
mean_deposit = df3.groupby(['balance_
quantile'])['deposit'].mean()

#plot
plt.plot(mean_deposit.index, mean_
deposit.values)
plt.title('Средний % кредита в зави-
симости от величены баланса')
plt.xlabel('величина баланса')
plt.ylabel('средний %')
plt.show() (см. рис. 5)
```

Рис. 5. Визуализация точности моделей на основе перекрестной проверки

```
df3['balance_group'] =
pd.qcut(df3['balance'], q=25,
precision=0, duplicates ='drop')
mean_deposit = df3.groupby(['balance_
group'])['deposit'].mean()
mean_deposit.sort_
values(ascending=False).head(10)
balance_group
(2879.0, 3586.0]      0.610048
(3586.0, 4721.0]      0.582339
(7102.0, 37127.0]     0.574163
(2360.0, 2879.0]      0.568345
(1938.0, 2360.0]      0.568019
(4721.0, 7102.0]      0.565947
(1610.0, 1938.0]      0.565947
(948.0, 1133.0]       0.529833
(1133.0, 1337.0]      0.516827
(805.0, 948.0]        0.497608
Name: deposit, dtype: float64
for i in range(0,20):
  mean=df3[df3['balance_
quantile']==i]['deposit'].mean()
  print(i,»—», df3[df3['balance_
quantile']==i]['balance_group'].
values[0],» Mean prob», mean)
0 — (-1.0, 2.0]  Mean prob
0.38416075650118203
1 — (2.0, 32.0]  Mean prob
0.30392156862745096
```

```
2 — (32.0, 76.0]  Mean prob
0.4028436018957346
3 — (76.0, 121.0]  Mean prob
0.41204819277108434
4 — (121.0, 169.0]  Mean prob
0.46335697399527187
5 — (169.0, 222.0]  Mean prob
0.38954869358669836
6 — (222.0, 280.0]  Mean prob
0.4819277108433735
7 — (280.0, 336.0]  Mean prob
0.46265060240963857
8 — (336.0, 408.0]  Mean prob
0.4819277108433735
9 — (408.0, 488.0]  Mean prob
0.47494033412887826
10 — (488.0, 577.0]  Mean prob
0.47129186602870815
11 — (577.0, 679.0]  Mean prob
0.4714285714285714
12 — (679.0, 805.0]  Mean prob
0.4650602409638554
13 — (805.0, 948.0]  Mean prob
0.49760765550239233
14 — (948.0, 1133.0]  Mean prob
0.5298329355608592
15 — (1133.0, 1337.0]  Mean prob
0.5168269230769231
16 — (1337.0, 1610.0]  Mean prob
0.49403341288782815
17 — (1610.0, 1938.0]  Mean prob
0.565947242206235
18 — (1938.0, 2360.0]  Mean prob
0.568019093078759
19 — (2360.0, 2879.0]  Mean prob
0.5683453237410072
```

Вывод

Баланс выше 2700 имеет больше шансов на то, что они будут списаны на кредит.

Возраст

```
df3['age']=df['age']
df3['age_quantile'] =
pd.qcut(df3['age'], q=25,
labels=False, duplicates ='drop')

#сгруппируем по 'age_quantile' и
найдем средний результат
mean_deposit = df3.groupby(['age_
quantile'])['deposit'].mean()

#plot
plt.plot(mean_deposit.index, mean_
deposit.values)
plt.title('Средний % кредита в зави-
симости от возраста')
plt.xlabel('возраст')
plt.ylabel('Средний %')
plt.show() (см. рис. 6)
```

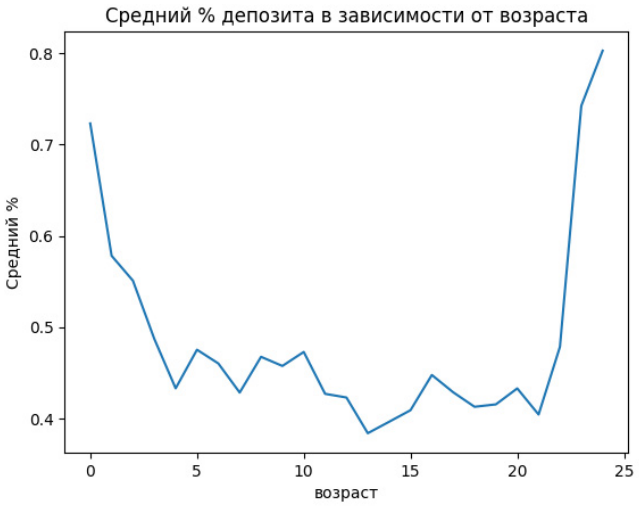

Рис. 6. Средний процент вероятности получения кредита в зависимости от уровня баланса клиента

```
df3['age_group'] =
pd.qcut(df3['age'], q=25,
precision=0, duplicates ='drop')
mean_deposit = df3.groupby(['age_
group'])['deposit'].mean()
mean_deposit.sort_
values(ascending=False).head(10)
```

```
age_group
(65.0, 95.0]    0.803030
(59.0, 65.0]    0.742547
(17.0, 25.0]    0.723256
(25.0, 27.0]    0.578231
(27.0, 29.0]    0.550877
(29.0, 30.0]    0.487119
```

```
(57.0, 59.0]    0.478395
(31.0, 32.0]    0.475225
(36.0, 37.0]    0.472779
(34.0, 35.0]    0.467442
Name: deposit, dtype: float64
for i in range(0,23):
  mean=df3[df3['age_quantile']==i]
['deposit'].mean()
  print(i,»—», df3[df3['age_
quantile']==i]['age_group'].
values[0],» Mean prob», mean)
0 — (17.0, 25.0]  Mean prob
0.7232558139534884
1 — (25.0, 27.0]  Mean prob
0.5782312925170068
2 — (27.0, 29.0]  Mean prob
0.5508771929824562
3 — (29.0, 30.0]  Mean prob
0.48711943793911006
4 — (30.0, 31.0]  Mean prob
0.4329004329004329
5 — (31.0, 32.0]  Mean prob
0.4752252252252252
6 — (32.0, 33.0]  Mean prob
0.4601366742596811
7 — (33.0, 34.0]  Mean prob
0.428246013667426
8 — (34.0, 35.0]  Mean prob
0.46744186046511627
9 — (35.0, 36.0]  Mean prob
0.45742092457420924
10 — (36.0, 37.0]  Mean prob
0.47277936962750716
11 — (37.0, 38.0]  Mean prob
0.42686567164179107
12 — (38.0, 39.0]  Mean prob
0.4228395061728395
13 — (39.0, 41.0]  Mean prob
0.3836805555555556
14 — (41.0, 42.0]  Mean prob
0.39622641509433965
15 — (42.0, 44.0]  Mean prob
0.4088888888888889
16 — (44.0, 46.0]  Mean prob
0.4475374732334047
17 — (46.0, 48.0]  Mean prob
0.42857142857142855
18 — (48.0, 50.0]  Mean prob
0.41265822784810124
19 — (50.0, 52.0]  Mean prob
0.41530054644808745
20 — (52.0, 54.0]  Mean prob
0.4327956989247312
21 — (54.0, 57.0]  Mean prob
0.40417457305502846
22 — (57.0, 59.0]  Mean prob
0.4783950617283951
```

Вывод

Люди в возрасте от 17 до 32 и от 57 до 95 имеют больше всего шансов взять кредит.

Duration

```
df3['duration']=df['duration']
df3['duration_quantile'] =
pd.qcut(df3['duration'], q=25,
labels=False, duplicates ='drop')

mean_deposit = df3.
groupby(['duration_quantile'])
['deposit'].mean()

plt.plot(mean_deposit.index, mean_
deposit.values)
plt.title('Средний % в зависимости
от Duration')
plt.xlabel('Duration')
plt.ylabel('Средний %')
plt.show() (см. рис. 7)
```

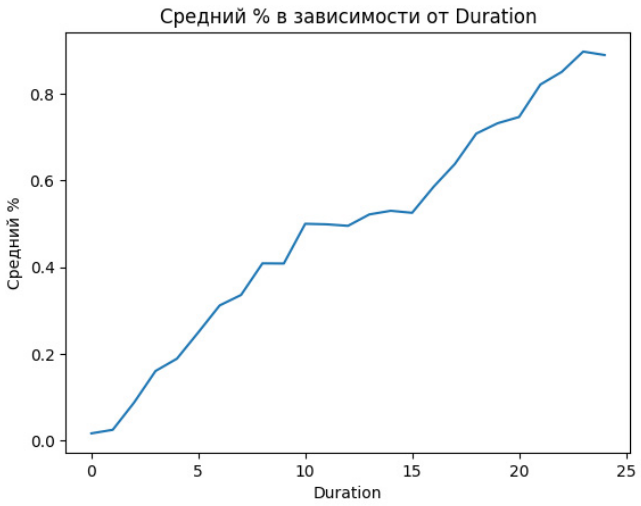

Рис. 7. Соотношение времени обучения и качества предсказаний для ключевых моделей

```
df3['duration_group'] =
pd.qcut(df3['duration'], q=25,
precision=0, duplicates ='drop')
mean_deposit = df3.
groupby(['duration_group'])
['deposit'].mean()
print(mean_deposit.sort_
values(ascending=False).head(10))
duration_group
(910.0, 1141.0]     0.897375
(1141.0, 2775.0]    0.889423
(763.0, 910.0]      0.850602
(660.0, 763.0]      0.821429
(580.0, 660.0]      0.746411
(510.0, 580.0]      0.732057
(448.0, 510.0]      0.708134
(397.0, 448.0]      0.638095
(355.0, 397.0]      0.585132
(293.0, 323.0]      0.529976
Name: deposit, dtype: float64
for i in range(0,25):
  mean=df3[df3['duration_
quantile']==i]['deposit'].mean()
  print(i,»—», df3[df3['duration_
quantile']==i]['duration_group'].
values[0],» Mean prob», mean)
0 — (1.0, 45.0]  Mean prob
0.016317016317016316
1 — (45.0, 68.0]  Mean prob
0.02457002457002457
2 — (68.0, 86.0]  Mean prob
0.08767772511848342
3 — (86.0, 102.0]  Mean prob
0.16028708133971292
4 — (102.0, 119.0]  Mean prob
0.18867924528301888
5 — (119.0, 135.0]  Mean prob
0.24939467312348668
6 — (135.0, 151.0]  Mean prob
0.3115942028985507
7 — (151.0, 166.0]  Mean prob
0.33573141486810554
8 — (166.0, 185.0]  Mean prob
0.408675799086758
9 — (185.0, 205.0]  Mean prob
0.4083129584352078
10 — (205.0, 225.0]  Mean prob 0.5
11 — (225.0, 245.0]  Mean prob
0.49876543209876545
12 — (245.0, 266.0]  Mean prob
0.495260663507109
13 — (266.0, 293.0]  Mean prob
0.5216346153846154
14 — (293.0, 323.0]  Mean prob
0.5299760191846523
15 — (323.0, 355.0]  Mean prob
0.5253012048192771
16 — (355.0, 397.0]  Mean prob
0.5851318944844125
17 — (397.0, 448.0]  Mean prob
0.638095238095238
18 — (448.0, 510.0]  Mean prob
0.7081339712918661
19 — (510.0, 580.0]  Mean prob
0.7320574162679426
20 — (580.0, 660.0]  Mean prob
0.7464114832535885
21 — (660.0, 763.0]  Mean prob
0.8214285714285714
22 — (763.0, 910.0]  Mean prob
0.8506024096385543
23 — (910.0, 1141.0]  Mean prob
0.8973747016706444
24 — (1141.0, 2775.0]  Mean prob
0.8894230769230769
```

Вывод

Показатель duration имеет ключевое значение. Если разговор продлится больше 720 секунд, то вероятность того, что человек возьмет кредит 80%.

Выводы

Создание модели

```
#импорт библиотек
from sklearn.base import
BaseEstimator, TransformerMixin
from sklearn.preprocessing import
OneHotEncoder, MinMaxScaler,
LabelEncoder, OrdinalEncoder
from sklearn.impute import
SimpleImputer
from sklearn.pipeline import
Pipeline
from sklearn.compose import
ColumnTransformer
import xgboost
df1
     age           job  marital
education default  balance housing
loan  \
0       59       admin.  married
secondary      no     2343     yes
no
1       56       admin.  married
secondary      no       45      no
no
2       41   technician  married
secondary      no     1270     yes
no
3       55     services  married
secondary      no     2476     yes
no
4       54       admin.  married
tertiary      no      184      no
no
…    …           …      …        …
…      …     …  …
11157   33  blue-collar   single
primary      no        1     yes
no
11158   39     services  married
secondary      no      733      no
no
11159   32   technician   single
secondary      no       29      no
no
11160   43   technician  married
secondary      no        0      no
yes
11161   34   technician  married
secondary      no        0      no
no

      contact  day month  duration
campaign  pdays  previous poutcome
\
0       unknown    5   may      1042
1     -1         0  unknown
1       unknown    5   may      1467
1     -1         0  unknown
2       unknown    5   may      1389
1     -1         0  unknown
3       unknown    5   may       579
1     -1         0  unknown
4       unknown    5   may       673
2     -1         0  unknown
…          …  …   …        …       …
…       …      …
11157  cellular   20   apr       257
1     -1         0  unknown
11158   unknown   16   jun        83
4     -1         0  unknown
11159  cellular   19   aug       156
2     -1         0  unknown
11160  cellular    8   may         9
2    172         5  failure
11161  cellular    9   jul       628
1     -1         0  unknown

    deposit
0         yes
1         yes
2         yes
3         yes
4         yes
…       …
11157      no
11158      no
11159      no
11160      no
```

```
11161      no

[10449 rows x 17 columns]
df1=df1.reset_index()
df1.drop('index', axis=1,
inplace=True)
df1
     age           job  marital
education default  balance housing
loan  \
0       59       admin.  married
secondary      no     2343     yes
no
1       56       admin.  married
secondary      no       45      no
no
2       41   technician  married
secondary      no     1270     yes
no
3       55     services  married
secondary      no     2476     yes
no
4       54       admin.  married
tertiary      no      184      no
no
…    …           …      …        …
…      …     …  …
10444   33  blue-collar   single
primary      no        1     yes
no
10445   39     services  married
secondary      no      733      no
no
10446   32   technician   single
secondary      no       29      no
no
10447   43   technician  married
secondary      no        0      no
yes
10448   34   technician  married
secondary      no        0      no
no

      contact  day month  duration
campaign  pdays  previous poutcome
\
0       unknown    5   may      1042
1     -1         0  unknown
1       unknown    5   may      1467
1     -1         0  unknown
2       unknown    5   may      1389
1     -1         0  unknown
3       unknown    5   may       579
1     -1         0  unknown
4       unknown    5   may       673
2     -1         0  unknown
…          …  …    …        …        …
…       …      …
10444  cellular   20   apr       257
1     -1         0  unknown
10445   unknown   16   jun        83
4     -1         0  unknown
10446  cellular   19   aug       156
2     -1         0  unknown
10447  cellular    8   may         9
2    172         5  failure
10448  cellular    9   jul       628
1     -1         0  unknown

    deposit
0         yes
1         yes
2         yes
3         yes
4         yes
…       …
10444      no
10445      no
10446      no
10447      no
10448      no

[10449 rows x 17 columns]
dic = {«yes»:1,»no»:0}
lst = [«loan»,»default»,»housing»]
for i in lst:
  df1[i] = df1[i].map(dic)
dic = {«yes»:1,»no»:0}
df1[«deposit»] = df1[«deposit»].
map(dic)
df1
     age           job  marital
education  default  balance  housing
loan  \
0       59       admin.  married
secondary        0     2343        1
```

```
0
1       56       admin.  married
secondary        0       45        0
0
2       41   technician  married
secondary        0     1270        1
0
3       55     services  married
secondary        0     2476        1
0
4       54       admin.  married
tertiary         0      184        0
0
…     …           …       …         …
…       …       …    …
10444   33  blue-collar   single
primary          0        1        1
0
10445   39     services  married
secondary        0      733        0
0
10446   32   technician   single
secondary        0       29        0
0
10447   43   technician  married
secondary        0        0        0
1
10448   34   technician  married
secondary        0        0        0
0

      contact  day month  duration
campaign  pdays  previous poutcome  \
0       unknown    5   may      1042
1    -1         0  unknown
1       unknown    5   may      1467
1    -1         0  unknown
2       unknown    5   may      1389
1    -1         0  unknown
3       unknown    5   may       579
1    -1         0  unknown
4       unknown    5   may       673
2    -1         0  unknown
…         …  …   …       …       …    …
…      …
10444  cellular   20   apr       257
1    -1         0  unknown
10445   unknown   16   jun        83
4    -1         0  unknown
10446  cellular   19   aug       156
2    -1         0  unknown
10447  cellular    8   may         9
2   172         5  failure
10448  cellular    9   jul       628
1    -1         0  unknown

     deposit
0           1
1           1
2           1
3           1
4           1
…       …
10444       0
10445       0
10446       0
10447       0
10448       0

[10449 rows x 17 columns]
```

```
contact_list = df1['contact'].
unique().tolist()
poutcome_list = ['success','unknown',
'other','failure']
month_list = df1['month'].unique().
tolist()
month_list = list(reversed(month_
list))
print(contact_list)
print(poutcome_list)
print(month_list)
```

```
['unknown', 'cellular', 'telephone']
['success', 'unknown', 'other',
'failure']
['sep', 'apr', 'mar', 'feb', 'jan',
'dec', 'nov', 'oct', 'aug', 'jul',
'jun', 'may']
```

```
contact_label=list(range(0,3,1))
poutcome_label=list(range(0,4,1))
month_label=list(range(0,12,1))
dic_contact=dict(zip(contact_list,
contact_label))
dic_poutcome=dict(zip(poutcome_list,
poutcome_label))
dic_month=dict(zip(month_list,
month_label))
```

```
print(dic_contact)
print(dic_poutcome)
print(dic_month)
```

```
{'unknown': 0, 'cellular': 1,
'telephone': 2}
{'success': 0, 'unknown': 1,
'other': 2, 'failure': 3}
{'sep': 0, 'apr': 1, 'mar': 2,
'feb': 3, 'jan': 4, 'dec': 5, 'nov':
6, 'oct': 7, 'aug': 8, 'jul': 9,
'jun': 10, 'may': 11}
```

```
#маппинг словарей с параметрами
df1[«contact»] = df1[«contact»].
map(dic_contact)
df1[«poutcome»] = df1[«poutcome»].
map(dic_poutcome)
df1[«month»] = df1[«month»].map(dic_
month)
```

```
#X и Y разделение
X = df1.drop('deposit', axis=1)
Y = df1['deposit']
```

```
#StratifiedShuffleSplit
sss = StratifiedShuffleSplit(n_
splits=1, test_size=0.3, random_
state=1)
for train_index, test_index in sss.
split(X, Y):
  train_df = df1.loc[train_index]
  test_df = df1.loc[test_index]
```

```
#тренировка и тестирование на наборе
данных
X_train = train_df.drop(«deposit»,
axis=1)
Y_train = train_df['deposit']

X_test = test_df.drop(«deposit»,
axis=1)
Y_test = test_df['deposit']
```

```
X_train
```

```
    age            job   marital
education  default  balance  housing
\
5461   28    blue-collar   married
primary        0      674        1
4220   36    blue-collar   married
secondary        0      324        1
5530   56     technician  divorced
secondary        0     1480        1
4249   31      housemaid    single
primary        0    26965        0
9514   30    blue-collar   married
secondary        0      177        1
…    …              …        …        …
…      …      …
8100   34    blue-collar   married
primary        0      425        1
4223   27     technician    single
tertiary        0    11862        0
343    26  self-employed    single
secondary        0      551        0
4449   41    blue-collar   married
secondary        0     5517        1
2983   76        retired   married
primary        0     2223        0

    loan  contact  day  month
duration  campaign  pdays  previous
poutcome
5461     0        2   14     11
921         4     -1         0
1
4220     1        1   16      9
830         1     -1         0
1
5530     1        1    5      3
576         1     -1         0
1
4249     0        1   21      1
654         2     -1         0
1
9514     0        1    9      1
62          2    332         3
3
…    …       …  …    …        …
…    …        …        …
8100     0        1   16      9
1389         7     -1         0
1
4223     0        1   25      6
285         2     97         7
0
343      0        1    8      9
531         1     -1         0
1
4449     0        1   10      9
584         1     -1         0
```

```
1
2983     0         2     4       3
429          1     -1          0
1
[7314 rows x 16 columns]
trf1 = ColumnTransformer([
  ('ohe', OneHotEncoder(sparse=False,
handle_unknown='ignore'),[1,2,3])
], remainder='passthrough')
trf2 = ColumnTransformer(
  transformers=[('scaler',
StandardScaler(), [0,-1])],
  remainder='passthrough'
)
trf3 = xgboost.XGBClassifier(n_
estimators=80, learning_rate=0.1,
gamma=0, subsample=0.75,
                          colsample_
bytree=1, max_depth=5)
pipe = Pipeline([
  ('trf1', trf1),
  ('trf2', trf2),
  ('trf3', trf3)
])
#щаги pipe
pipe.steps
[('trf1',
ColumnTransformer(remainder='passthr
ough',

transformers=[('ohe',

OneHotEncoder(handle_
unknown='ignore',

sparse=False),
                                  [1,
2, 3])])),
('trf2',
ColumnTransformer(remainder='passthr
ough',

transformers=[('scaler',
StandardScaler(), [0, -1])])),
('trf3',
XGBClassifier(base_score=None,
booster=None, callbacks=None,
              colsample_bylevel=None,
colsample_bynode=None, colsample_
bytree=1,
              device=None, early_
stopping_rounds=None, enable_
categorical=False,
              eval_metric=None,
feature_types=None, gamma=0, grow_
policy=None,
              importance_type=None,
interaction_constraints=None,
              learning_rate=0.1, max_
bin=None, max_cat_threshold=None,
              max_cat_to_onehot=None,
max_delta_step=None, max_depth=5,
              max_leaves=None, min_
child_weight=None, missing=nan,
              monotone_
constraints=None, multi_
strategy=None, n_estimators=80,
              n_jobs=None, num_
parallel_tree=None, random_
state=None, …))]
#тренировка модели на 70% от набора
данных
pipe.fit(X_train, Y_train)
Pipeline(steps=[('trf1',
                 ColumnTransformer(rem
ainder='passthrough',

transformers=[('ohe',

OneHotEncoder(handle_
unknown='ignore',

sparse=False),

[1, 2, 3])])),
              ('trf2',
               ColumnTransformer(rem
ainder='passthrough',

transformers=[('scaler',
StandardScaler(),

[0, -1])])),
              ('trf3',
               XGBClassifier(base_
score=None, booster=None,
```

```
callbacks=None,

colsample_bylevel=None, colsampl…
                              feature_
types=None, gamma=0, grow_
policy=None,

importance_type=None,

interaction_constraints=None,
learning_rate=0.1,
                              max_
bin=None, max_cat_threshold=None,
                              max_
cat_to_onehot=None, max_delta_
step=None,
                              max_
depth=5, max_leaves=None,
                              min_
child_weight=None, missing=nan,

monotone_constraints=None, multi_
strategy=None,
                              n_
estimators=80, n_jobs=None,
                              num_
parallel_tree=None, random_
state=None, …))])
#предсказание по тестовому набору
данных
y_pred = pipe.predict(X_test)
y_pred
array([1, 0, 0, …, 0, 1, 0])
#отчет по классификации
print(classification_report(Y_test,
y_pred))
              precision    recall
f1-score   support

           0       0.88      0.82
0.85      1615
           1       0.82      0.88
0.85      1520

    accuracy
0.85      3135
   macro avg       0.85      0.85
0.85      3135
weighted avg       0.85      0.85
0.85      3135
```

```
#оценка точности модели
from sklearn.metrics import
accuracy_score
accuracy_score(Y_test, y_pred)
0.850079748165869
```

Заключение

Сравнение различных моделей машинного обучения показало, что алгоритм XGBoost имеет наилучшие показатели с точностью 84,6% на тестовых данных и минимальным риском переобучения. Это делает его наиболее эффективным инструментом для прогнозирования результатов рекламных кампаний. Кроме того, было выявлено, что наибольшую активность в принятии кредитных предложений проявляют возрастные группы от 17 до 32 лет и старше 57 лет. Важно отметить, что качественная предварительная обработка данных, включая удаление выбросов и масштабирование признаков, сыграла ключевую роль в повышении точности моделей. Полученные результаты и идеи могут быть использованы для оптимизации маркетинговых стратегий, более эффективного таргетинга клиентов и повышения общей эффективности кредитных кампаний.

Кроме того, было обнаружено, что клиенты с высоким балансом счета с большей вероятностью принимают кредитные предложения, поскольку их финансовая стабильность позволяет им уверенно принимать решения о новых обязательствах. Интересно, что возраст также играет важную роль: наиболее активны клиенты в возрасте от 17 до 32 лет, а также старше 57 лет. Молодые клиенты обычно заинтересованы в дополнительных средствах для удовлетворения текущих потребностей, в то время как старшая возрастная группа может использовать кредиты для реализации накопленных планов или улучшения качества своей жизни.

Таким образом, сочетание этих факторов позволяет нам строить более точные и эффективные модели для прогнозирования поведения

клиентов и оптимизации кредитных кампаний, что в конечном итоге может повысить прибыльность и успешность банковских предложений.

Список литературы:

1) Далингер К. Е., Айграшева А. А. Анализ текущего состояния льготного ипотечного кредитования в Российской Федерации // Вопросы студенческой науки, выпуск № 10 (62), октябрь 2021 г. // URL: https://sciff.ru/wp-content/uploads/2021/11/Sciff_10_62.pdf (дата обращение 5 июля 2023 года)

2) Мыцык И. В., Кузнецова Е. В. Современные проблемы ипотечного кредитования в России и анализ перспектив их решения // Международный научный журнал «Инновационная наука» № 5–1 / 2023 // URL: https://aeterna-ufa.ru/sbornik/IN-2023–05–1.pdf (дата обращение 5 июля 2023 года)

3) Предет Р. В. Анализ банковского кредитования субъектов малого и среднего бизнеса // Форум молодых ученых, 3(79) 2023 // URL: https://www.forum-nauka.ru/_files/ugd/b06fdc_ded5cc7854e341879805bc7e3d415403.pdf?index=true (дата обращение 5 июля 2023 года)

4) Мишина М. Ю., Зверев А. В. Статистический анализ состояния ипотечного кредитования в Российской Федерации // Статистический анализ социально-экономического развития субъектов Российской Федерации // Сборник научных трудов по материалам VIII Международной научно-практической конференции. Брянск, 2021 // Издательство: Федеральное государственное бюджетное образовательное учреждение высшего образования «Брянский государственный инженерно-технологический университет» (Брянск)

Comparative analysis of creditworthiness forecasting models

Demidov A. D., Aleshina A. V., Milyutin M. A.
Plekhanov Russian University of Economics, Lomonosov Moscow State University

The process of assessing the creditworthiness of applicants is a key element of the activities of banking institutions. The overall effectiveness of the credit policy, the level of risks and the profitability of the organization depend on the accuracy of this process. Currently, with the expansion of available data and progress in analytics, it has become possible to use modern machine learning methods to predict the financial behavior of applicants in detail and accurately.

The paper considers a data set reflecting the behavior of potential clients taking into account modern research on the mortgage lending market in the Russian Federation (Dalinger, 2021), which was analyzed in detail. The factors that have the greatest impact on the likelihood of obtaining a loan were identified, and the quality of various classification algorithms, such as logistic regression, support vector machine (SVM), XGBoost and random forest, was assessed. The results made it possible to compare the accuracy of the models, and the correlation map and visualization made it possible to identify key indicators that contribute to decision making.

Based on the results obtained, conclusions were made about the advantages of a particular model, and recommendations were determined to improve the effectiveness of credit policy and risk management in the banking sector.

Keywords: credit scoring systems.

References

1. Dalinger K. E., Aigrasheva A. A. Analysis of the current state of preferential mortgage lending in the Russian Federation // Issues of student science, issue No. 10 (62), October 2021. // URL: https://sciff.ru/wp-content/uploads/2021/11/Sciff_10_62.pdf (date accessed July 5, 2023)
2. Mytsyk I. V., Kuznetsova E. V. Modern problems of mortgage lending in Russia and analysis of prospects for their solution // International scientific journal «Innovation Science» No. 5–1 / 2023 // URL: https://aeterna-ufa.ru/sbornik/IN-2023–05–1.pdf (date accessed July 5, 2023)
3. Predet R. V. Analysis of bank lending to small and medium-sized businesses // Forum of young scientists, 3 (79) 2023 // URL: https://www.forum-nauka.ru/_files/ugd/b06fdc_ded5cc7854e341879805bc7e3d415403.pdf?index=true (date accessed July 5, 2023)
4. Mishina M. Yu., Zverev A. V. Statistical analysis of the state of mortgage lending in the Russian Federation // Statistical analysis of the socio-economic development of the constituent entities of the Russian Federation // Collection of scientific papers based on the materials of the VIII International Scientific and Practical Conference. Bryansk, 2021 // Publisher: Federal State Budgetary Educational Institution of Higher Education «Bryansk State University of Engineering and Technology» (Bryansk)

Формирование инвестиционного портфеля частным инвестором в современной России

Гревцев Михаил Эдуардович
Студент Московский государственный университет имени М. В. Ломоносова
E-mail: blessedlypretty@gmail.com

В статье рассматриваются подходы к формированию инвестиционного портфеля для российских частных инвесторов. Особое внимание уделено критериям отбора акций с использованием преимущественно фундаментального анализа с указанием конкретных значений ключевых финансовых показателей и мультипликаторов. Дополнительно рассмотрена эволюция портфельных теорий, отмечен факт устаревания моделей Марковица и CAPM, но подчеркнута важность усовершенствованных моделей (Фамы-Френча и Кархарта). Подробно расписаны критерии отбора облигаций в портфель в зависимости от макроэкономических ожиданий, дюрации, финансового положения эмитента, прозрачности компании, репутации и кредитного рейтинга. Дополнительно рассмотрено использование ИИС нового типа, который дает возможность получать сразу несколько налоговых вычетов, что серьезно влияет на общую доходность портфеля. В статье уделяется внимание немаловажным индивидуальным аспектам инвестирования, таким как цели, риск-профиль, горизонт инвестирования и возраст. Приведены исследования, отражающие зависимость горизонта инвестирования и разброса доходности финансовых инструментов, то есть изменение их риска. В заключении предложен пошаговый алгоритм для составления инвестором портфеля финансовых инструментов, учитывающий цели, срок инвестирования, отношение к риску и возможности использования налоговых вычетов.

Ключевые слова: портфельные инвестиции, фундаментальный анализ, финансовые показатели и мультипликаторы, портфельные теории, выбор акций, выбор облигаций.

Введение

Портфельные инвестиции представляют собой стратегию распределения денежных средств между различными финансовыми инструментами. Такие инвестиции позволяют поддерживать желаемое соотношение риска и доходности. Важно отметить, что портфельные инвестиции обладают высокой ликвидностью и не требуют больших вложений.

В России интерес к такому виду инвестирования стремительно растет в последние годы. Это достигается благодаря повышению финансовой грамотности населения и простоте приобретения финансовых инструментов.

Для начала инвестирования в портфельные активы необходимо получить доступ к бирже. Это можно сделать, заключив договор с брокером для использования торгового терминала или вложив средства через управляющую компанию, такую как ПИФы, ETF, ИДУ и хедж-фонды. Использование биржей специального посредника, центрального контрагента, исключает риск неисполнения обязательств по биржевым операциям купли-продажи.

При формировании портфеля инвестору важно определить свои цели и сроки инвестирования, учитывая свой возраст, отношение к риску и размер имеющегося капитала.

Критерии отбора акций в инвестиционный портфель

Отбор акций в портфель мы не будем проводить с помощью коэффициентов риска и доходности, так как они изменяются во времени, зависят от методики расчета и под риском понимают не только отрицательное, но и положительное изменение цен актива. Более того, они скорее подходят для оценки уже готового портфеля, а не для формирования нового. Аналогично технический анализ рассматривать не будем, ведь он нужен для спекуляций, а не для реальных инвестиций.

Актуальным будет использование новостного анализа, под которым понимается мониторинг новостных ресурсов (в том числе изучение санкций), просмотр аналитики и отчетностей компаний. К прогнозам стоит относиться скептически, но можно использовать их как дополнение к собственному анализу.

Фундаментальный анализ является наиболее полезным при отборе акций в портфель. Среди методов анализа выделяют сравнение по финансовым показателям и мультипликаторам, о чем мы подробно поговорим далее, и DCF (дисконтирование будущих денежных потоков компании для определения ее стоимости) [1], [2]. Мы рассмотрим самые основные показатели и мультипликаторы. Стоит понимать, что у каждой отрасли есть дополнительные коэффициенты, свойственные именно ей.

Рыночные коэффициенты показывают, как инвесторы в данный момент оценивают компанию. Выделяют следующие показатели: 1) P/E — отношение капитализации компании к чистой прибыли. Значения от 0 до 5 говорят о ее недооцененности. Более 20 — о переоцененности. 2) PEG — отношение показателя P/E к ожидаемому росту прибыли в последующие годы (в %). Оптимальные значения — меньше одного и меньше среднеотраслевого. 3) P/S — отношение капитализации к выручке. Используется для ритейла. Значение меньше 1 говорит о недооцененности компании. Больше 2 — о переоцененности. 4) Показатель P/BV больше применим именно для анализа банков. От отражает отношение капитализации компании к капиталу. Если значение показателя превышает 1.3, то компания переоценена. Значения меньше 1 говорят о недооцененности. 5) EV/EBITDA, где EV есть сумма капитализации и чистого долга компании. Значения больше 5 или среднеотраслевого значения говорят о переоцененности компании.

Дивидендные коэффициенты стоит принимать во внимание только в том случае, если инвестор желает получать пассивный доход. Выделяют следующие показатели: 1) EPS — чистая прибыль на одну акцию. Важно, чтобы динамика была положительной. 2) Под DPS понимается размер дивидендов на одну акцию. 3) Dividend Yield — отношение дивидендов на акцию к ее стоимости. 4) DPR представляет собой отношение DPS к EPS. То есть показывает, какая доля прибыли компании направляется на выплату акционерам. Слишком высокие значения — негативный фактор, так как средства не реинвестируются. В долгосрочной перспективе могут появиться проблемы с поддержанием EPS. 5) DSI — индекс дивидендной стабильности. Принимает значения от 0 до 1. Чем больше, тем лучше.

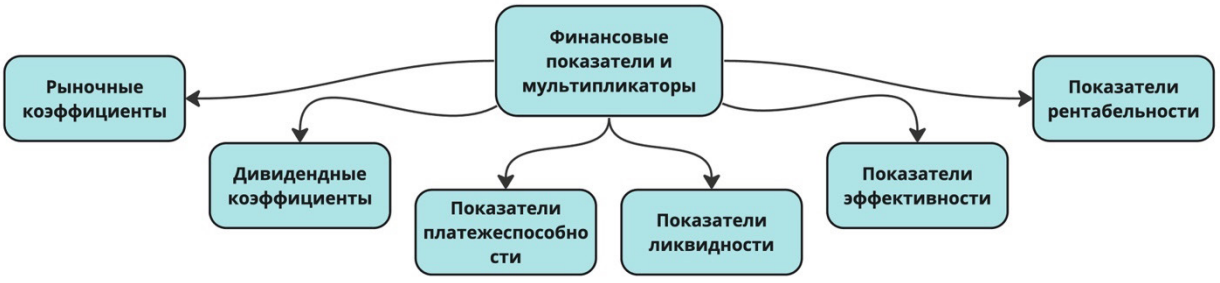

Рис. 1. Виды финансовых показателей и мультипликаторов (составлено автором).

Показатели платежеспособности показывают, насколько компания способна выполнять свои финансовые обязательства. 1) Debt/Eq — отношение долга к капиталу. Оптимальные значения 0,5–0,7. Отдельно стоит посмотреть на сами долги. Если они преимущественно краткосрочные, то это плохо. 2) Net Debt/EBITDA — Отношение долга к EBITDA. Если показатель растет, то это указывает на то, что долговая нагрузка растет быстрее прибыли, что усложняет выплату дохода акционерам. 3) EBITDA/ % расходы — Оптимальные значения — больше двух. Аналогично предыдущему пункт, если показатель со временем уменьшается, это плохой знак для платежеспособности компании.

Показатели ликвидности оценивают, способна ли компания покрывать текущими активами краткосрочные обязательства. Рассмотрим два показателя — Quick Ratio и Current Ratio. Первый представляет собой отношение оборотных активов за вычетом запасов к краткосрочным обязательствам. Желательно, чтобы показатель принимал значения более 0,5. Однако слишком большие значения указывают на неэффективность компании. Второй показатель похож на первый и является отношением оборотных активов к краткосрочным обязательствам. Оптимальные значения составляют от 1 до 2.

Показатели эффективности показывают, насколько эффективна компания доходит от выручки к прибыли. Они наиболее простые для восприятия. Среди них выделяют Gross margin, Operating margin и Net Income Margin (ROS). Они представляют собой отношение валовой прибыли, операционной прибыли и чистой прибыли к выручке соответственно. Желательно, чтобы показатели не снижались со временем и были выше, чем среднеотраслевые значения.

Показатели рентабельности: 1) ROE представляет собой отношение чистой прибыли к капиталу компании, но делать выводы на его основе нельзя, так как мы ничего не знаем про долги компании. Следует сравнивать с показателем ROA, отражающий отношение чистой прибыли к активам компании. 2) ROI есть отношение чистой прибыли к сумме капитала и долгосрочных обязательств (по сути, отражает рентабельность инвестиций). У данного показателя есть два основных аналога: ROIC и ROCE. Отличие заключается лишь в том, что вместо чистой прибыли используется операционная прибыль и EBIT соответственно.

Для проведения сравнительного анализа необходимо следовать определенным правилам. Необходимо сравнивать компании по финансовым показателям только внутри одной отрасли и сразу по совокупности показателей, причем обращая внимание на их динамику. Такие показатели как P/S, EV/S, EV/EBITDA, Debt/EBITDA нерелевантны для банков.

Отдельно стоит затронуть портфельные теории. Их существенным упрощением является предположение, что активы не приносят текущую доходность. В целом, теории сводятся к выбору актива на основе его риска и доходности. Дополнительно стоит отметить, что существует рыночный риск и риск отдельного актива. Последний можно устранить путем использования диверсификации, то есть приобретения большого количества активов. Первой моделью, которая заложила основу для дальнейших исследований, была модель Марковица. Ее идея заключается к формированию портфеля, который бы давал максимальную доходность при заданном риске [3]. Позднее Джеймс Тобин дополнил модель, добавив в нее безрисковый актив. Он расширяет инвестиционные возможности индивида [4]. Оптимальный портфель формируется на основе прошлых данных. На практике же будущая доходность актива может сильно отличаться от исторической. Модель хорошо подходит именно для тех активов, риск и доходность которых со временем не меняется. Важно отметить, что под риском понимаются не только отрицательные изменения доходности, но и положительные, хотя для инвестора это благоприятно, что является еще одним минусом. Более того, модель не принимает во внимание Макро- и микро- факторы. Формирование портфелей на основе данной модели становится всё менее актуальным.

Модель CAPM является более современной. Разработана она была независимо Уильямом Шарпом [5], Джоном Линтнером [6] и Яном Моссином [7]. Строится модель на основе теории Марковица. Используется для описания взаимосвязи между ожидаемой доходности актива и его рыночного риска. Основное уравнение данной модели:

$$r_i = r_f + \beta_i(r_m - r_f) \quad (1)$$

Где r_i — ожидаемая доходность актива (портфеля). r_f — безрисковая доходность. r_m — рыночная доходность, β_i — коэффициент бета актива (портфеля).

Согласно модели CAPM вознаграждается только рыночный риск, так как именно он не подлежит диверсификации.

Исследования показывают, что далеко не всегда требуемая доходность, полученная из данной модели, совпадает с фактической. Было выведено следующее уравнение изменения коэффициента бета [8], [9]:

$$\beta_{Blume;i} = 0.66\beta_i + 0.34 \quad (2)$$

Уравнение показывает, что, используя исторические значения коэффициента, мы недооцениваем прогнозируемую доходность (при маленьких значениях коэффициента), а при больших — переоцениваем.

Проблема модели CAPM заключается в том, что учитывается лишь один фактор, влияющий на определение требуемой доходности. Более того, коэффициент бета постоянно меняется, рыночный портфель не наблюдаем и используется линейная зависимость от рыночной премии за риск [10].

Для устранения этих недостатков была придумана Модель Фамы-Френч, в которой для определения требуемой доходности учитывается уже три фактора. Ее уравнение выглядит так [11]:

$$r_i = r_f + \beta_i * (r_m - r_f) + \beta_{SMB;i} * r_{SMB} + \beta_{HML;i} * r_{HML} + \alpha_i \quad (3)$$

Первый фактор такой же, как и в старой модели. SMB представляет собой разницу в доходностях компаний с малой и большой капитализацией, а HML — разницу доходностей компаний с высокими и маленькими значениями BV/P [11].

Модель Кархарта является расширением модели Фамы-Френча и использует четыре фактора для вычисления требуемой доходности. Основное уравнение [12]:

$$r_i = r_f + \beta_i * (r_m - r_f) + \beta_{SMB;i} * r_{SMB} + \beta_{HML;i} * r_{HML} + \beta_{MOM;i} * r_{MOM} \quad (4)$$

Где под r_{MOM} понимается разность доходностей акций, показавших высокую доходность за последние 12 месяцев и низкую.

Дополнительно важно отметить необходимость проводить регулярную ребалансировку портфеля, особенно в текущих условиях. Наиболее актуально это делать при изменении ожидаемой доходности или риска актива и изменении структуры портфеля (пропорции активов).

Критерии отбора облигаций в портфель

Для начала рассмотрим, какие преимущества облигаций можно выделить по сравнению с депозитами. Облигации обладают большей ликвидностью, что означает возможность легко и быстро их конвертировать в деньги. Выбор облигаций существенно шире, что позволяет инвестору легко подобрать финансовый инструмент с наиболее подходящим сроком погашения. Облигации обычно предоставляют возможность получения более высокой доходности, чем депозиты. Однако у облигаций есть и свои недостатки. Для успешного инвестирования инвестору необходимы определенные финансовые знания. По облигациям не предусмотрено страхование, как во вкладах. Необлагаемая налоговая база отсутствует (устраняется с помощью ИИС).

Отдельно поговорим про замещающие облигации. Их особенность заключается в том, что номинал и купоны привязаны к иностранной валюте, но торгуются они в националь-

ной валюте и все выплаты по ним инвестор получает в рублях. Поскольку это российский актив, риска блокировки из-за санкций нет. Из минусов можно выделить лишь высокую стоимость, однако это устраняется путем покупки соответствующих фондов, специализирующихся на таких облигациях. Покупать этот инструмент стоит в том случае, если вы ожидаете обесценение национальной валюты. Альтернативным вариантом служат российские облигации, номинированные в иностранной валюте. В случае приостановки торгов соответствующей валютой на бирже эмитент может произвести расчеты в рублях. Важно отметить, что при покупке любых облигаций следует дополнительно учитывать кредитный риск, процентный, ликвидности и рыночный.

Если инвестор планирует реализовать облигацию до погашения, то важно понимать, по какой цене будет торговаться облигация в случае изменения процентной ставки. Для этого нужно использовать дюрацию. Лучше всего подходит именно модифицированная дюрация, которая показывает, на сколько процентов изменится цена облигации при изменении требуемой доходности на 100 б. п. Облигации с большей дюрацией сильнее реагируют на изменение ставки, поэтому выгоднее приобретать fixed income облигации с большой дюрацией в случае, если ожидается снижение процентных ставок. Иначе — флоатеры, линкеры, fixed income облигации с минимальной дюрацией и краткосрочные. При этом дюрация тоже меняется при изменении доходности, но это можно устранить с помощью выпуклости. Важно осознавать, что, даже несмотря на учет выпуклости, этот метод представляет собой упрощенную оценку того, как изменения процентных ставок влияют на стоимость облигации [13].

При формировании портфеля стоит обращать внимание на финансовое положение компании, которое должно быть устойчивым. Все дело в том, что по облигациям обычно нет залога в виде имущества, обеспечением служат лишь доходы компании. Необходимо учитывать коэффициенты NET DEBT / EBITDA и EBITDA / % расходы, которые мы ранее подробно описывали. Желательно, чтобы компания показывала положительную чистую прибыль хотя бы за последние 3 года, а размер активов и выручки составлял не менее 30–35 млрд рублей. Немаловажным фактором является прозрачность компании и репутация. Так, стоит выбирать компании, у которых отсутствуют просроченные платежи, есть хотя бы локальный кредитный рейтинг, против которых нет крупных судебных исков. Компания должна существовать не менее пяти лет и иметь отчетность по МСФО минимум за три года. Желательно, чтобы она была одним из лидеров в своем сегменте.

Как ИИС может увеличить доходность инвестиционного портфеля

В 2024 году появился ИИС-3. Мы будем рассматривать только его, так как прошлые версии индивидуальных инвестиционных счетов неактуальны. Благодаря ИИС инвестор может получить сразу два налоговых вычета — 13% от суммы пополнения в год, но не более 52000 рублей в год. (если вы платите НДФЛ по ставке 15%, то не более 60000 рублей в год). Данный вычет можно получить только в том случае, если вы являетесь плательщиком НДФЛ. Второй вычет позволяет не платить налог с доходов на ИИС, если они не превышают 30 миллионов рублей. Иначе будет взиматься налог с суммы превышения по стандартным правилам. Необходимый минимальный срок владения для получения вычета составляет 5 лет. Далее он будет постепенно увеличиваться до 10 лет. В случае вывода денег раньше срока инвестору придется вернуть налоговые вычета и выплатить пени. Исключением является лишь вывод денег для дорогостоящего лечения. В этом случае вычеты сохраняются [14].

Новый ИИС теперь можно пополнять на любую сумму, ограничений нет. Однако покупать можно только ценные бумаги российских эмитентов (операции с валютой разрешены) [14].

Влияние риск-профиля и целей инвестора на структуру портфеля

При составлении инвестиционного портфеля важно учитывать личное отношение инвестора к риску. Первым элементом риск-профиля, согласно статье [15], является уровень риска, который требуется для достижения поставленных финансовых целей. Это предполагает, что инвестор уже определил целевой уровень доходности, на основе которого можно точно рассчитать необходимый риск, используя подходы, предложенные Марковицем.

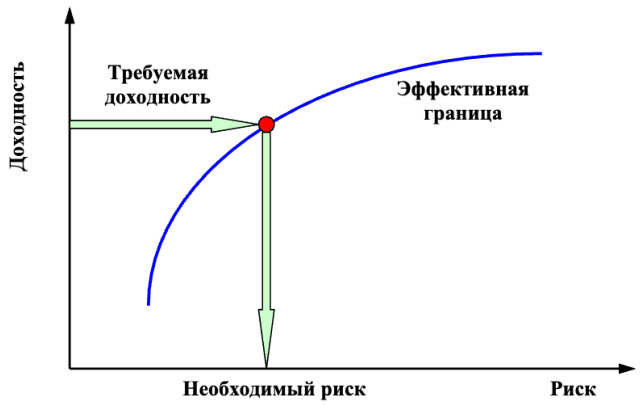

Рис. 2. Определение необходимого риска [15].

2) «Финансовая возможность принять риск, или способность к риску» [15]. Этот компонент показывает, насколько у инвестора есть возможность покрывать инвестиционные потери из других источников, если реализуется негативный сценарий. 3) Толерантность к риску. Она формируется исходя из врожденных особенностей человека и его жизненного опыта. На толерантность к риску влияет стрессоустойчивость, финансовые возможности, цели и многое другое. Для определения отношения к риску инвестор может пройти различные тесты, в том числе у некоторых брокеров. На их основе можно определить примерную комфортную долю акций в портфеле.

Влияние горизонта инвестирования на разброс в доходностях финансовых инструментов

Среднегодовая эффективная доходность зависит от срока инвестирования. Важно отметить, что в рамках одного года возможны значительные колебания, однако в долгосрочной перспективе рынки склонны расти. Было проведено исследование на рынке США, в рамках которого проводился анализ доходности акций и облигаций, посчитанной по разным временным периодам. В результате исследования было установлено, что спред в доходности уменьшается при увеличении горизонта инвестирования, что говорит о снижении риска при долгосрочных вложениях. Более того, средняя доходность растет с увеличением срока инвестирования. [16]:

Автор также провел исследование на российском рынке, что наиболее актуально для наших инвесторов. Важно понимать, что фондовый рынок в России появился недавно, поэтому автором было принято рассмотреть сроки инвестирования от 1 до 60 месяцев. Результаты оказались аналогичны тем, что получились при анализе рынка США.

Вариация годовой доходности за период	73 однолетних периода		69 пятилетних периода		64 десятилетних периода		54 двадцатилетних периода	
	Акции	Облигации	Акции	Облигации	Акции	Облигации	Акции	Облигации
Максимальная	54,0	40,4	24,1	21,6	20,1	15,6	17,7	11,1
Средняя	11,2	5,3	10,3	4,5	10,1	4,6	10,5	4,9
Минимальная	- 43,3	- 9,2	- 12,5	- 2,1	- 0,9	0,1	3,1	0,7

Рис. 3. Доходности в зависимости от временного горизонта [16].

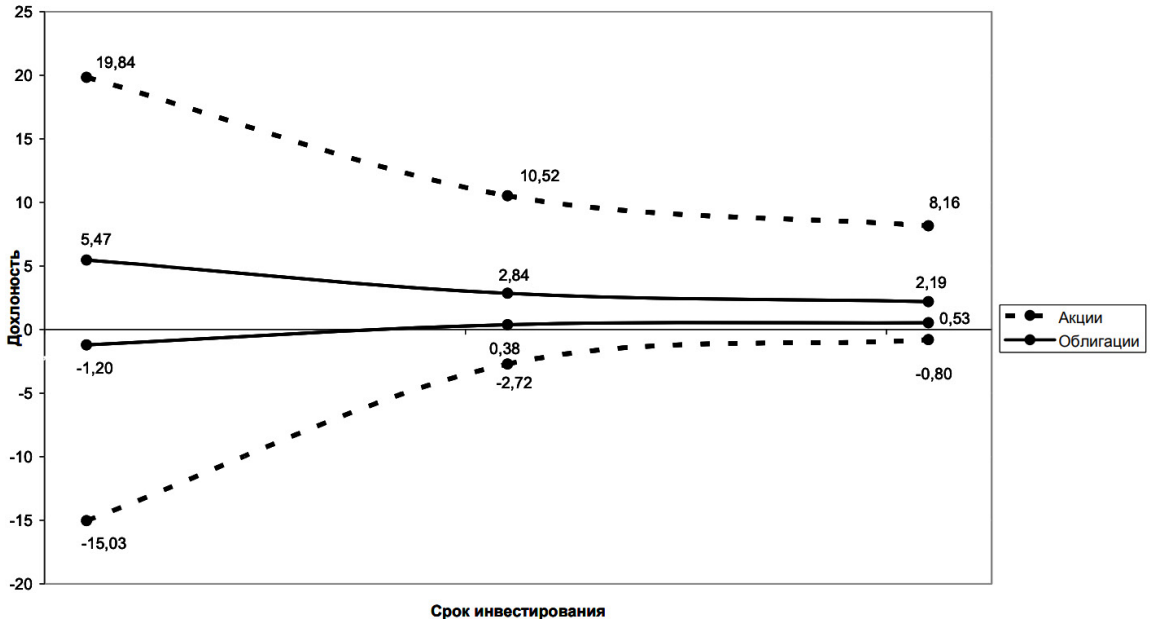

Рис. 4. Зависимость доходности от срока инвестирования [16].

Таким образом, в долгосрочной перспективе акции становятся менее рисковым инструментом, при этом приносят высокую доходность, что оправдывает увеличение их доли в портфеле (чем дольше срок, тем больше доля акций).

Возраст инвестора также играет важную роль в определении структуры портфеля. Это объясняется тем, что с возрастом уменьшается потенциальный временной горизонт инвестирования. Поэтому по мере приближения к пенсионному возрасту рекомендуется снижать долю высокорисковых инвестиционных инструментов.

Основные классификации инвестиционных портфелей

Инвестиционные портфели классифицируются по различным критериям, включая уровень риска, степень вовлеченности в управление и горизонт инвестирования. По размеру принимаемого риска выделяют: 1) консервативный портфель считается самым надежным, поскольку риск минимален. Включает в себя депозиты, ОФЗ, надежные корпоративные облигации, драгоценные металлы, недвижимость и иногда акции — голубые фишки. 2) В умеренном портфеле риск и доходность средние. Он состоит из корпоративных облигаций, ОФЗ, паевых фондов, акций и иногда производных финансовых инструментов. 3) Агрессивный портфель сопряжен с наибольшим риском и доходностью. В него включаются производные финансовые инструменты, акции рисковых компаний, стартапы, высокодоходные облигации. 4) Сбалансированный портфель является комбинацией трех предыдущих в примерно равных долях. Его риск и доходность средние [17]. 5) Вечнозеленый портфель нужен для того, чтобы приносить постоянную положительную доходность, преимущественно в основном за счет активов с минимальным риском. В него включаются облигации и недвижимость, а остальная часть портфеля состоит из акций для роста стоимости. Это позволяет перекрывать риски потери стоимости за счет доходов от безрисковых активов

По степени вовлеченности выделяют активный портфель, который требует постоянного контроля и управления, так как использует активные стратегии, производные финансовые инструменты и так далее, и пассивный, который не требует постоянного участия инвестора. В него часто включаются индексные

фонды и облигации, которые почти не требуют ребалансировки.

По горизонту инвестирования выделяют долгосрочные портфели (от 10 лет), среднесрочные (от 3 до 10 лет) и краткосрочные (до 3 лет).

Каков алгоритм составлении инвестиционного портфеля

1) Инвестор должен определиться с целью инвестирования и срок, исходя из чего стоит рассчитать необходимую итоговую прибыль (не забывая про инфляцию). 2) Пройти тест на определения отношения к риску, исходя из чего выбрать наиболее подходящий вид инвестиционного портфеля. 3) Далее необходимо просмотреть условия разных брокеров и открыть счет у наиболее подходящего вам. Если ваша цель долгосрочная, будет выгодно использовать ИИС. 4) На данном этапе необходимо определиться со структурой портфеля. При выборе акций следует ориентироваться на фундаментальный анализ, изучение новостей и макроэкономических факторов. Если инвестор имеет возможность, полезно рассчитать предполагаемую доходность портфеля с использованием моделей Фамы-Френча или Кархарта, описанных ранее. Однако важно помнить, что такие расчеты являются теоретическими и реальная доходность может значительно отличаться от прогноза. Как именно выбирать облигации для портфеля мы уже подробно описывали ранее. С помощью полученной теоретической доходности стоит вычислить необходимую сумму ежемесячного пополнения для достижения цели. 5) Следует пользоваться всеми доступными налоговыми льготами.

Литература

1. Портал Управляющая компания «Доходъ» // Электронный ресурс // URL: https://www.dohod.ru/ (дата обращения 01.08.2024)
2. Что такое мультипликаторы // Портал Тинькофф Журнал, 25.04.2022 // Электронный ресурс // URL: https://journal.tinkoff.ru/multilplicator/ (дата обращения 01.08.2024)
3. Markowitz, Harry. «Portfolio Selection.» The Journal of Finance, vol. 7, no. 1, 1952, pp. 77–91.
4. Tobin, James. «Liquidity preference as behavior towards risk.» The review of economic studies 25.2 (1958): 65–86.
5. Sharpe, William F. «Capital asset prices: A theory of market equilibrium under conditions of risk.» The journal of finance 19.3 (1964): 425–442.
6. Lintner, John. «The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets.» Stochastic optimization models in finance. Academic Press, 1975. 131–155.
7. Mossin, Jan. «Equilibrium in a capital asset market.» Econometrica: Journal of the econometric society (1966): 768–783.
8. Blume, Marshall E. «Portfolio theory: a step toward its practical application.» The Journal of Business 43.2 (1970): 152–173.
9. Blume, Marshall E., and Irwin Friend. «A new look at the capital asset pricing model.» The journal of finance 28.1 (1973): 19–33.
10. Roll, Richard. «A critique of the asset pricing theory's tests Part I: On past and potential testability of the theory.» Journal of financial economics 4.2 (1977): 129–176.
11. Fama, Eugene F., and Kenneth R. French. «Common risk factors in the returns on stocks and bonds.» Journal of financial economics 33.1 (1993): 3–56.
12. Carhart, Mark M. «On persistence in mutual fund performance.» The Journal of finance 52.1 (1997): 57–82.
13. Шалыганов, К. Ю. Вопросы управления процентным риском портфеля облигаций // К. Ю. Шалыганов // Научный альманах Центрального Черноземья. — 2022. — № 1–8. — С. 43–50.
14. Как будет работать ИИС нового типа // Тинькофф Инвестиции, 16.02.2024 // Электронный ресурс // URL: https://www.tinkoff.ru/finance/blog/new-iis/ (дата обращения 05.08.2024)

15. Якушин Д. И., Юдин С. В. Risk profiling и его место в системе финансового консультирования // Научно-методический электронный журнал «Концепт». — 2018. — № 12 (декабрь). — С. 204–215.
16. Берзон, Николай Иосифович. «Зависимость риска и доходности активов от временного горизонта инвестирования.» Университетское управление: практика и анализ 3 (2008): 65–72.
17. Что такое инвестиционный портфель? // Альфа-Банк, 21.01.2022 // Электронный ресурс // URL: https://alfabank.ru/help/articles/investments/chto-takoe-investicionnyj-portfel/ (дата обращения 11.08.2024)

Formation of an investment portfolio by a private investor in modern Russia

Grevtsev M. E.
Lomonosov Moscow State University

This article explores approaches to forming investment portfolios for private investors in Russia. Special attention is given to the criteria for selecting stocks, primarily using fundamental analysis, including specific values for key financial indicators and multiples. The evolution of portfolio theories is also discussed, noting the obsolescence of the Markowitz and CAPM models, while emphasizing the importance of advanced models (Fama-French and Carhart). Detailed criteria for selecting bonds are outlined, considering macroeconomic expectations, duration, issuer's financial condition, company transparency, reputation, and credit rating. Additionally, the use of the new type of Individual Investment Account, which offers multiple tax deductions, significantly impacting portfolio returns, is examined. The article addresses important individual aspects of investing, such as goals, risk profile, investment horizon, and age. Research is presented that reflects the relationship between investment horizon and the variability of returns on financial instruments, indicating changes in their risk levels. Finally, a step-by-step algorithm for constructing an investment portfolio is proposed, taking into account goals, investment period, risk tolerance, and the potential use of tax deductions.

Keywords: portfolio investments, fundamental analysis, financial indicators and multiples, portfolio theories, stock selection, bond selection.

References

1. Portal of the Management Company «Dohod» // Electronic resource // URL: https://www.dohod.ru/ (accessed 01.08.2024)
2. What Are Multiples // Tinkoff Journal, 25.04.2022 // Electronic resource // URL: https://journal.tinkoff.ru/multilplicator/ (accessed 01.08.2024)
3. Markowitz, Harry. «Portfolio Selection.» The Journal of Finance, vol. 7, no. 1, 1952, pp. 77–91.
4. Tobin, James. «Liquidity Preference as Behavior Towards Risk.» The Review of Economic Studies 25.2 (1958): 65–86.
5. Sharpe, William F. «Capital Asset Prices: A Theory of Market Equilibrium Under Conditions of Risk.» The Journal of Finance 19.3 (1964): 425–442.
6. Lintner, John. «The Valuation of Risk Assets and the Selection of Risky Investments in Stock Portfolios and Capital Budgets.» Stochastic Optimization Models in Finance. Academic Press, 1975. 131–155.
7. Mossin, Jan. «Equilibrium in a Capital Asset Market.» Econometrica: Journal of the Econometric Society (1966): 768–783.
8. Blume, Marshall E. «Portfolio Theory: A Step Toward Its Practical Application.» The Journal of Business 43.2 (1970): 152–173.
9. Blume, Marshall E., and Irwin Friend. «A New Look at the Capital Asset Pricing Model.» The Journal of Finance 28.1 (1973): 19–33.
10. Roll, Richard. «A Critique of the Asset Pricing Theory's Tests Part I: On Past and Potential Testability of the Theory.» Journal of Financial Economics 4.2 (1977): 129–176.
11. Fama, Eugene F., and Kenneth R. French. «Common Risk Factors in the Returns on Stocks and Bonds.» Journal of Financial Economics 33.1 (1993): 3–56.
12. Carhart, Mark M. «On Persistence in Mutual Fund Performance.» The Journal of Finance 52.1 (1997): 57–82.
13. Shalyganov, K. Yu. «Questions of Managing Interest Rate Risk in a Bond Portfolio.» Scientific Almanac of the Central Black Earth Region, vol. 1–8, 2022, pp. 43–50.
14. «How the New Type of IIA Will Work.» Tinkoff Investments, 16 Feb. 2024 // Electronic resource // URL: https://www.tinkoff.ru/finance/blog/new-iis/ (accessed 05.08.2024)
15. Yakushin, D. I., and Yudin, S. V. «Risk Profiling and Its Place in the Financial Advisory System.» Scientific and Methodological Electronic Journal «Concept,» Dec. 2018, no. 12, pp. 204–215.
16. Berzon, Nikolai Iosifovich. «The Dependence of Risk and Return on the Investment Time Horizon.» University Management: Practice and Analysis, vol. 3, 2008, pp. 65–72.
17. 17. What Is an Investment Portfolio? // Alfa-Bank, 21 Jan. 2022 // Electronic resource // URL: https://alfabank.ru/help/articles/investments/chto-takoe-investicionnyj-portfel/ (accessed 11.08.2024)

Тенденции развития зарубежного гражданского авиастроения в современных условиях

Кадетов Андрей Валерьевич
Студент экономического факультета МГУ
имени М. В. Ломоносова
E-mail: kadetovandre@gmail.com

Многие страны для достижения экономического роста и развития вкладывают огромные ресурсы в авиастроительную отрасль. Не все страны способны на создание гражданских самолётов, а уж те более на создание конкурентоспособных самолётов. Благодаря развитию авиастроения государства могут использовать все преимущества глобальной кооперации для достижения экономических и политических выгод, так как отрасль является технологичной, инновационной и капиталоёмкой, а также требует много денежных вложений и наличие качественного человеческого капитала в виде высококвалифицированных ученых и инженеров. В статье рассматриваются основные современные тенденции в развитии зарубежного гражданского авиастроения, дополнительно рассматриваются тренды в развитии пассажирских перевозок. Особое внимание уделяется анализу восстановления отрасли после шока, вызванного пандемией коронавирусной инфекции.

Ключевые слова: авиастроение, пассажирские перевозки, коронавирусная инфекция, Airbus, Boeing.

Прежде чем говорить непосредственно про авиастроение и авиастроительные компании, необходимо посмотреть на картину шире и узнать, что в целом происходит в области гражданской авиации, какие можно наблюдать главные тренды и с чем приходится сталкиваться отрасли в кризисные моменты. Для этого мы рассмотрим различные статистические данные, доступные в открытом доступе — прежде всего это базы данных Международной организации гражданской авиации (ICAO) и Международной ассоциации воздушного транспорта (IATA), а в дополнение к этому данные Всемирного банка и официальные ресурсы авиапроизводителей.

Для начала скажем, что одним из главных критериев, характеризующих состояние отрасли в целом, является показатель ежегодного пассажиропотока. Для того, чтобы детально изучить этот аспект, необходимо построить график и проследить за динамикой пассажирских перевозок, осуществляемых воздушным транспортом. Необходимые данные будем брать из двух источников: во-первых, это база данных Всемирного банка, во-вторых, для более детального и современного обзора дополним данными Международной организации гражданской авиации:

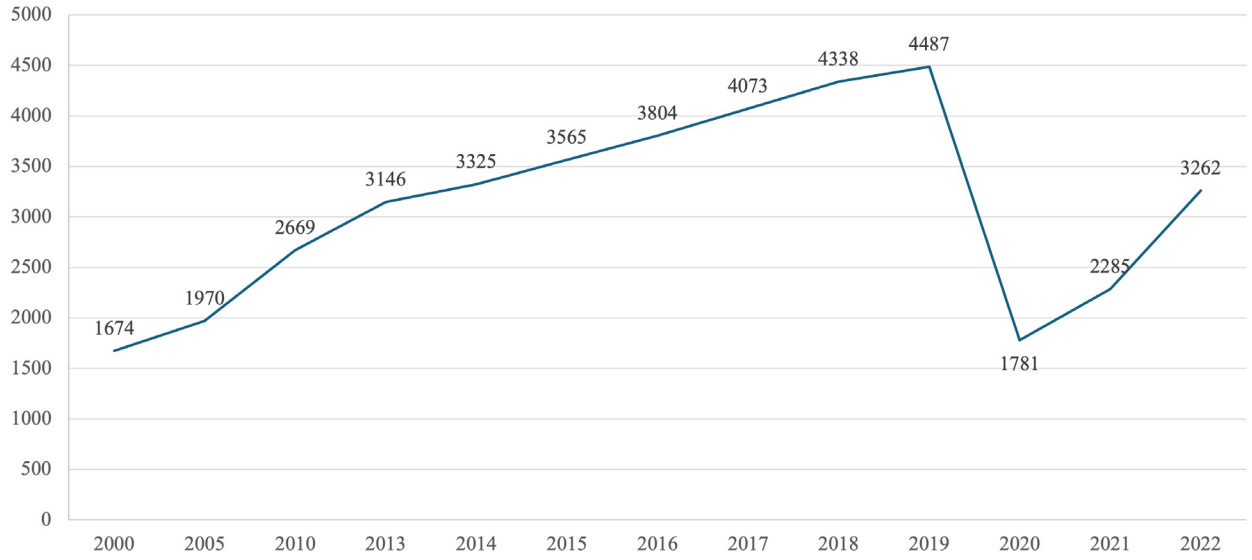

Рис 1. Количество пассажиров, перевезённых воздушным транспортом (млн чел.).
Источник: построено автором по данным [1] и [2]

На данном графике (см. рис. 1) мы можем оценить динамику пассажиропотока начиная с 2000 года. Как мы видим, количество пассажиров, перевозимых воздушным транспортом, постоянно росло на протяжении 20 лет, вплоть до 2019 года. Если в 2000 году количество пассажиров равнялось 1,674 млрд человек, то уже в 2019 году это количество выросло до 4,49 млрд человек, то есть рост более чем в 2,5 раза. Однозначного объяснения того, почему мы наблюдаем такой рост, нет — это могут быть как демографические факторы, ведь за 20 лет по данным Всемирного банка население планеты выросло с 6,16 млрд человек в 2000 году до 7,78 млрд человек в 2019 году [3], так и факторы, связанные с ростом и расширением глобализации, усилением взаимозависимости стран в контексте развития национальных экономик. Всё это тем или иным способом способствует развитию воздушных перемещений не только внутри отдельно взятых стран, но и между ними, таким образом суммарное ежегодное количество пассажиров постоянно растёт. Однако уже в 2020 году мы видим резкое снижение показателя до 1,78 млрд перевезенных пассажиров. Конечно же, причиной такого падения является кризис, связанный с пандемией COVID-19. Коронавирус привёл к тому, что на уровне национальных экономик по всему миру вводились ограничения на перемещения. Соответственно, из-за локдаунов людям было затруднительно путешествовать внутри страны, а на международном уровне были выставлены ограничения и барьеры, которые существенно снизили пассажиропоток — введение обязательных сертификатов о вакцинации, запрет на полёты и многое другое. Постепенное снятие ограничений позволило довольно быстро восстановить количество перевезенных пассажиров — уже в 2021 году показатель вырос в 1,3 раза, до 2,29 млрд человек, а в 2022 году показатель восстановился до уровня 2014 года — было перевезено 3,26 млрд человек. Несмотря на то, что кроме коронавируса были и другие ограничивающие факторы, например, эскалация кризиса на Украине в 2022 и на Ближнем Востоке в 2023 году, Международная организация гражданской авиации прогнозирует уже в 2024 году восстановление пассажиропотока до уровня 2019 года [4].

Далее посмотрим на то, как пассажиропоток распределяется по разным регионам. Для изучения регионального аспекта мы также обратимся к данным ICAO и IATA, которые предоставляют данные по шести регионам мира.

Таблица 1. Распределение количества пассажиров по регионам мира за 2021 и 2022 года. Источник: построено автором по данным [2].

Год	2021		2022		
Регион	Количество пассажиров	Доля	Количество пассажиров	Доля	Годовой прирост
Азиатско-тихоокеанский регион	744,8 млн.	32,6%	870,3 млн.	26,7%	16,9%
Северная Америка	691,1 млн.	30,3%	919,7 млн.	28,2%	33,1%
Европа	513,2 млн.	22,5%	937,7 млн.	28,7%	82,7%
Латинская Америка	187,7 млн.	8,2%	281,4 млн.	8,6%	49,9%
Ближний Восток	100,5 млн.	4,4%	178,1 млн.	5,5%	77,2%
Африка	46,4 млн.	2,0%	75,1 млн.	2,3%	61,9%
Весь мир	**2283,8** млн.	**100%**	**3262,3** млн.	**100%**	**42,9%**

При рассмотрении регионального распределения пассажиропотока (таблица 1) мы можем увидеть, что в 2021 году лидером по количеству пассажиров был Азиатско-Тихоокеанский регион (АТР) — за год было перевезено воздушным транспортом практически 745 млн человек, что составляет 32,6% от мирового пассажиропотока. Далее за АТР следовал регион Северная Америка с долей 30,3%, затем Европа (22,5%), Латинская Америка и Карибский бассейн (8,2%), Ближний Восток (4,4%) и Африка (2%). Но уже в 2022 году ситуация кардинально изменилась — наибольший прирост пассажиропотока можно было наблюдать в европейском регионе (прирост количества пассажиров 82,7% год к году). Соответственно, Европа вышла в лидеры и заняла первое место среди регионов с долей в 28,7% от мирового пассажиропотока. Далее существенный прирост наблюдался в таких регионах как Ближний Восток, Африка и Латинская Америка. Однако тройка регионов лидеров осталась неизменной, за исключением их распределения внутри группы (первое место за Европой, далее Северная Аме

Отдельно посмотрим на коэффициент загрузки по регионам за 2021 и 2022 года (таблица 2). Этот показатель позволяет нам оценить эффективность работы авиакомпаний, представленных в том или ином регионе. Высоким считается показатель 80 и более процентов. Как мы можем увидеть, буквально за один год ситуация кардинально меняется, во всех регионах мира эффективность компаний резко возрастает, однако общие тренды никуда не уходят. Во-первых, мы вновь наблюдаем за восстановлением

Таблица 2. Коэффициент загрузки в разных регионах за 2021 и 2022 года. Источник: построено автором по данным [2]

	Ближний Восток	Африка	АТР	Европа	Северная Америка	Латинская Америка
Коэффициент загрузки пассажирами в 2021 году	55%	61%	63%	69%	73%	77%
Коэффициент загрузки пассажирами в 2022 году	74%	71%	71%	81%	83%	81%

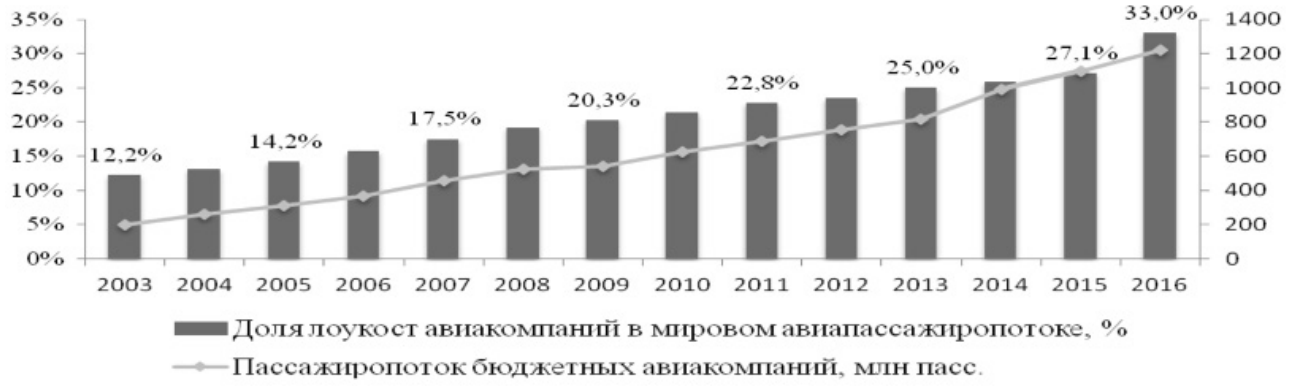

Рис 2. Доля лоукост-компаний в глобальном авиапассажиропотоке в 2003–2016 гг. Источник: [5]

отрасли после пандемии, буквально за один год данный показатель существенно вырос во всех регионах, а во-вторых, по-прежнему главная тройка регионов с самыми эффективными авиакомпаниями — это Северная Америка, Латинская Америка и Европа. В 2022 году их показатели превысили 80%, что также позволяет говорить о том, что авиакомпании достаточно эффективно используют самолёты в своей коммерческой деятельности.

Отдельно хочется отметить растущую роль бюджетных авиакомпаний в развитие воздушных перевозок в целом. Как отмечают Матвеева и Мальцев [5], низкобюджетные авиакомпании или лоукостеры (low-cost) всё активнее развиваются, темпы роста таких компаний высокие, а само их количество постоянно увеличивается:

Так на данном графике (см. рис. 2) мы можем заметить, что за всё время наблюдений (с 2003 года по 2016 год) пассажиропоток бюджетных авиакомпаний значительно вырос — примерно в 6,5 раз. Если в 2003 году пассажиропоток составлял более 190 млн пассажиров, то уже к 2016 году увеличился до значения 1,21 млрд пассажиров. В то же время сама по себе значимость лоукостеров в мировой авиации увеличилась более чем в 2 раза — если в 2003 году доля в мировом пассажиропотоке составляла 12%, то в 2016 — уже более 30% [5]. Такие высокие темпы роста достигаются благодаря тому, что low-cost компании максимально эффективно используют располагаемые ресурсы, проводят грамотную оптимизацию затрат, а также постоянно увеличивают масштабы своей деятельности. Всё это позволяют придать импульс развитию отрасли, причём бюджетные авиакомпании эффективно развиваются не только на отдельных внутренних рынках, но и на глобальных, тем самым мировой пассажиропоток постоянно растёт.

После того как мы изучили основные тенденции в развитии пассажирских авиаперевозок, самое время перейти к рассмотрению трендов в авиастроении. В рамках данной работы мы сфокусируемся на рассмотрении именно гражданских самолётов. Но перед этим необходимо прояснить классификацию различных самолётов. Существуют две ключевые характеристики воздушных судов, позволяющих их классифицировать — это дальность полёта, а также вместительность. Припадчев в своей работе приводит следующую классификацию, акцентируя внимание на показателе дальности полёта гражданского самолёта: «ближнемагистральные самолёты с дальностью полёта от 1000 до 2500 километров, среднемагистральные с дальностью полёта от 2500 до 6000 километров и дальнемагистральные, у которых

дальность полёта свыше 6000 километров» [6]. При этом существует и другая, более детальная классификация самолётов, представленная, например, российской компанией ОАК (Объединенная авиастроительная корпорация) [7]: сначала происходит разделение самолётов на две большие категории по дальности полёта, а именно региональные и магистральные самолёты, затем каждая категория имеет свою подкатегорию, различающаяся вместительностью — региональные самолёты бывают турбовинтовыми (РТ) (от 30 до 60 пассажиров на борту) и реактивными (РР) (от 30 до 110 пассажиров), в то время как магистральные бывают узкофюзеляжными (УФ) (от 110 до 200 пассажиров) и широкофюзеляжными (ШФ) (от 200 до 350+ человек).

Теперь после изучения классификации гражданских судов следует посмотреть на мировой парк самолётов. Разные компании предлагают разную аналитику по этому вопросу, мы обратимся к документу компании Avolon, которая предоставляет услуги по лизингу самолётов

Данное распределение парка воздушных судов по категориям в Таблице 3 позволяет нам сказать, что сегмент магистральных самолётов самый популярный в мире — на их долю приходится суммарно практически 83% от всего мирового парка. При этом самыми популярными и самолётами являются именно узкофюзеляжные — 67% от всего парка. Интересно, что с точки зрения прогнозирования, именно на узкофюзеляжные самолёты авиапроизводители будут делать ставку, так как к 2042 году их количество увеличится практически в 2 раза до 34 тыс. экземпляров, а доля такого типа самолётов в мировом парке вырастет до 73,2%. Такую тенденцию достаточно просто объяснить — так как спрос на эффективное использование самолётов постоянно растет для того, чтобы максимально сократить затраты при производстве, а потом и эксплуатации самолётов, производители вместо того, чтобы каждый раз создавать новые модели самолётов скорее фокусируются на постоянной модификации существующих моделей, выпуская принципиально новые модели крайне редко. Лучшим образом для этого подходят узкофюзеляжные самолёты, так как авиастроители способны предложить огромное количество разных моделей, которые удовлетворяют практически любой спрос и являются универсальными в применении — захватывается бОльшее число маршрутов, а затраты на эксплуатацию снижаются.

Анализ распределения мирового парка гражданских самолётов позволяет нам сделать вывод об особой значимости сегмента магистральных самолётов. Так как на данный момент внушительная часть воздушных судов в мировом парке приходится именно на узкофюзеляжные и широкофюзеляжные самолёты, причём их доля со временем будет значительно увеличиваться, то имеет смысл сфокусироваться на том, какие производители представлены на международном рынке.

Таблица 3. Мировой парк пассажирских самолётов по категориям за 2022 год и прогноз на 2042 год. Источник: построено автором по данным [8]

	Фактическое значение на 2022 год	Доля в 2022 году	Прогнозное значение на 2042 год	Доля в 2042 году
Широкофюзеляжные	3800	15,7%	7480	16,0%
Узкофюзеляжные	16220	67,1%	34320	73,2%
Региональные реактивные	2320	9,6%	2580	5,5%
Региональные турбовинтовые	1840	7,6%	2500	5,3%
Всего	**24180**	**100%**	**46880**	**100%**

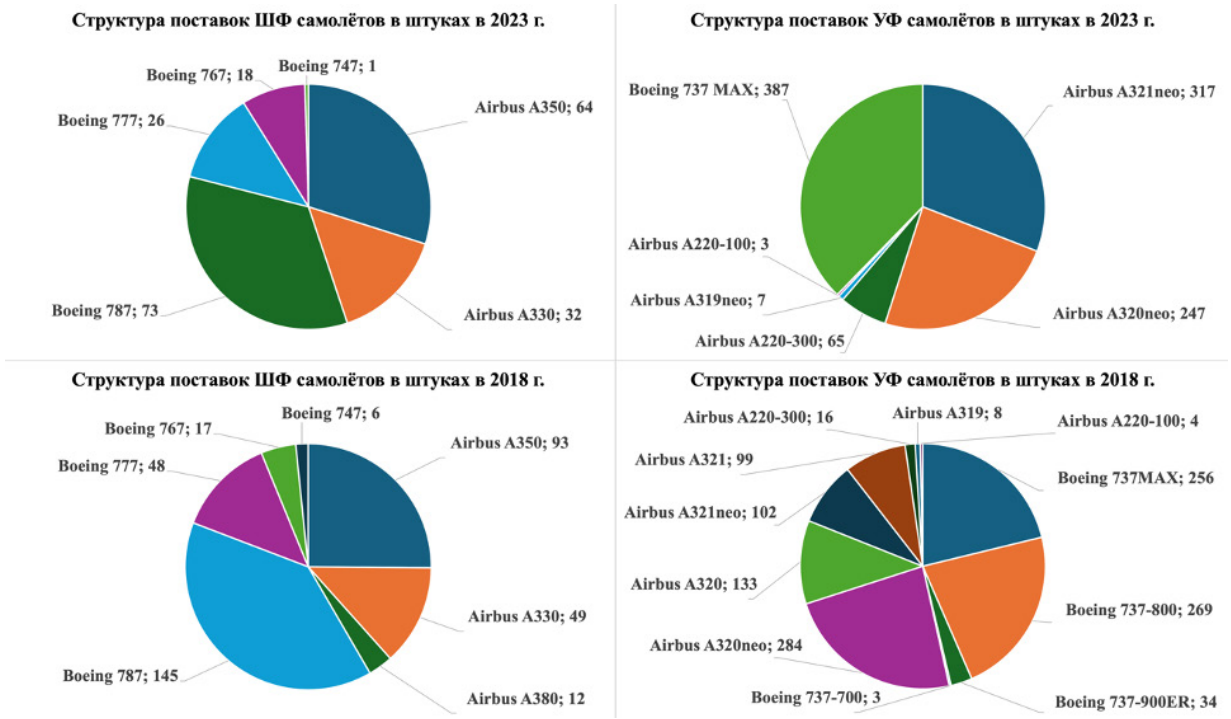

Рис. 3. Структура поставок широкофюзеляжных и узкофюзеляжных самолётов по моделям компаний Airbus и Boeing за 2018 и 2023 года. Источник: построено автором по данным [9] и [10]

По данным МАЦ 2.0 [11] на данный момент ключевыми игроками в сегменте магистральных самолётов являются только две компании — европейская Airbus и американская Boeing. Именно они фактически делят между собой этот рынок, создавая дуополию. При этом самолёты других производителей также учитываются в мировом парке, однако с точки зрения новых поставок никак не влияют на ситуацию на рынке — например, ИЛ-96, являющийся российским широкофюзеляжным самолётом, изредка собирается в единичных экземплярах и то не ежегодно. На Рисунке 3 можно увидеть распределение широкофюзеляжных и узкофюзеляжных самолётов по моделям в 2018 и 2023 годах. В сегменте широкофюзеляжных самолётов в 2018 году лидирует Boeing — американская компания поставила 216 самолётов против 154 у Airbus. Также отметим, что у компании Boeing с точки зрения предложения на одну модель больше. В 2023 году ситуация с поставками выглядит заметно хуже — из-за пандемии резко снизились портфели заказов, потребители решили отложить спрос, поэтому поставки новых самолётов тоже снизились до 118 и 96 самолётов у Boeing и Airbus соответственно. Несмотря на снижение поставок почти в 2 раза, распределение осталось прежним — доля Boeing на рынке больше, а европейская компания предлагает на 2 модели меньше в сравнении с американским конкурентом. В сегменте узкофюзеляжных самолётов ситуация заметно отличается — в 2018 году больше всего на рынок поставила самолётов компания Airbus, а именно 646, в то время как компания Boeing поставила лишь 562, что по-прежнему можно считать отличным показателем. Но, как и в случае с ШФ самолётами, пандемия COVID-19 нашла отражение в результатах деятельности обеих компаний. В 2023 году обе компании поставили на рынок меньше самолётов нежели в 2018 году. Однако стоит отметить, что Airbus лучше чувствует себя после кризиса и поставила в 2023 году 639 самолётов, что сопоставимо с допандемийным уровнем 2018 года. Обе компании сфокусировались на поставках самолётов нового поколения. При этом европейская компания поставила 5 моделей самолётов, в то время как американская компания,

сфокусировавшись исключительно на новом поколении 737 MAX, поставила на рынок всего 387 самолётов, что на 175 экземпляров меньше через 5 годами ранее.

Подводя итоги, следует сказать, что текущие тенденции развития гражданского авиастроения позволяют нам сделать вывод о восстановлении отрасли после шока, вызванного пандемией COVID-19. Воздушные перевозки постепенно восстанавливаются до уровня 2019 года и далее будут расти. Восстановление можно наблюдать как в отдельных регионах, так и в целом по миру. В ближайшие годы прогнозируется существенный рост мирового парка воздушных судов, особенно в сегменте магистральных самолётов — быстрее всего будет расти сегмент узкофюзеляжных самолётов, соответственно, их доля будет увеличиваться. Компании, специализирующиеся на магистральных самолётах, являются самыми крупными и важными, и на данный момент рынок фактически поделён между двумя производителями, Airbus и Boeing, причём европейская компания лучше всего восстанавливается после экономического спада 2020 года.

Литература

1. Данные портала Всемирного банка [Электронный ресурс] // URL: https://data.worldbank.org/indicator/IS.AIR.PSGR (дата обращения: 20.08.2024)
2. Данные портала Международной организации гражданской авиации [Электронный ресурс]//URL:https://www.icao.int/sustainability/WorldofAirTransport/Documents/ARC_2022_Tables_final_12032024.pdf (дата обращения: 20.08.2024)
3. Данные портала Всемирного банка [Электронный ресурс] // URL: https://data.worldbank.org/indicator/SP.POP.TOTL (дата обращения: 20.08.2024)
4. Passenger air traffic surpasses pre-pandemic levels [Электронный ресурс] // Материалы Международной организации гражданской авиации от 27.02.2024 // URL: https://www.icao.int/Newsroom/Pages/Passenger-air-traffic-surpasses-pre-pandemic-levels.aspx (дата обращения: 20.08.2024)
5. Матвеева А. В., Мальцев А. А. Лоукостеры как вектор динамичного развития мирового рынка авиаперевозок //Российский внешнеэкономический вестник. — 2017. — № . 8. — С. 80–91.
6. Припадчев А. Д. Определение оптимального парка воздушных судов. — 2009.
7. Данные портала Объединенной авиастроительной корпорации [Электронный ресурс] // URL: https://www.uacrussia.ru/ru/investors/presentations/?year=2019 (дата обращения: 20.08.2024)
8. Avolon World Fleet Forecast 2023–2042 [Электронный ресурс] // Материалы Avolon от 13.06.2023 // URL: https://avolon.aero/news/world-fleet-forecast-2023–2042 (дата обращения: 20.08.2024)
9. Данные портала Airbus [Электронный ресурс] // URL: https://www.airbus.com/en/products-services/commercial-aircraft/orders-and-deliveries (дата обращения: 20.08.2024)
10. Данные портала Boeing [Электронный ресурс] // URL: https://www.boeing.com/commercial#orders-deliveries (дата обращения: 20.08.2024)
11. Направления развития мирового рынка гражданской авиатехники в 2018 году [Электронный ресурс] // Материалы Межотраслевого аналитического центра (МАЦ 2.0) от 27.06.2019 // URL: https://iac2.ru/ru/analytics/ (дата обращения: 20.08.2024)

Trends in the development of foreign civil aircraft industry in modern conditions

Kadetov A. V.
Lomonosov Moscow State University

Many countries invest enormous resources in the aircraft industry to achieve economic growth and development. However, not all countries are capable of creating civil aircraft, not to mention competitive ones. The development of the aircraft industry enables states to leverage the advantages of global cooperation to attain economic and political benefits, as the industry is highly technological, innovative, and capital-intensive. It also requires significant financial investments and high-quality human capital in the form of highly qualified scientists and engineers. This article examines the main contemporary trends in the development of foreign civil aircraft industry, with an additional focus on trends in the

development of passenger transportation. Special attention is paid to analyzing the industry's recovery following the shock caused by the COVID-19 pandemic.

Keywords: aircraft industry, passenger transportation, COVID-19, Airbus, Boeing.

References

Data from the World Bank portal [Electronic resource] // URL: https://data.worldbank.org/indicator/IS.AIR.PSGR (date of access: 20.08.2024)

12. Data from the International Civil Aviation Organization portal [Electronic resource] // URL: https://www.icao.int/sustainability/WorldofAirTransport/Documents/ARC_2022_Tables_final_12032024.pdf (date of access: 20.08.2024)
13. Data from the World Bank portal [Electronic resource] // URL: https://data.worldbank.org/indicator/SP.POP.TOTL (date of access: 20.08.2024)
14. Passenger air traffic surpasses pre-pandemic levels [Electronic resource] // Materials of the International Civil Aviation Organization dated 27.02.2024 // URL: https://www.icao.int/Newsroom/Pages/Passenger-air-traffic-surpasses-pre-pandemic-levels.aspx (date accessed: 20.08.2024)
15. Matveeva A. V., Maltsev A. A. Low-cost airlines as a vector of dynamic development of the global air transportation market // Russian Foreign Economic Bulletin. — 2017. — No. 8. — P. 80–91.
16. Pripadchev A. D. Determination of the optimal aircraft fleet. — 2009.
17. Data from the United Aircraft Corporation portal [Electronic resource] // URL: https://www.uacrussia.ru/ru/investors/presentations/?year=2019 (date of access: 20.08.2024)
18. Avolon World Fleet Forecast 2023–2042 [Electronic resource] // Avolon materials from 13.06.2023 // URL: https://avolon.aero/news/world-fleet-forecast-2023–2042 (date of access: 20.08.2024)
19. Data from the Airbus portal [Electronic resource] // URL: https://www.airbus.com/en/products-services/commercial-aircraft/orders-and-deliveries (date of access: 20.08.2024)
20. Data from the Boeing portal [Electronic resource] // URL: https://www.boeing.com/commercial#orders-deliveries (date of access: 20.08.2024)
21. Directions for the development of the global civil aviation market in 2018 [Electronic resource] // Materials of the Inter-Industry Analytical Center (IAC 2.0) dated 27.06.2019 // URL: https://iac2.ru/ru/analytics/ (date of access: 20.08.2024)