|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Головной учебно-исследовательский и методический центр .

профессиональной реабилитации лиц с ограниченными .

возможностями здоровья (инвалидов) .

КАФЕДРА Системы обработки информации и управления

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ***

***НА ТЕМУ:***

**Система анализа алгоритмов машинного обучения для решения задач классификации c использованием Pandas**

Студент ИУ5Ц – 103Б А.М.Соловьева

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель ВКРБ Ю.А.Григорьев

(Подпись, дата) (И.О.Фамилия)

Нормоконтролер Ю.Н. Кротов

(Подпись, дата) (И.О.Фамилия)

2024 г.

АННОТАЦИЯ

Расчётно-пояснительная записка квалификационной работы бакалавра содержит 53 страницы. С приложениями объем составляет 73 страницы. Работа включает в себя 11 таблицы и 30 иллюстраций. В процессе выполнения было использовано 17 источников.

Объектом разработки является система анализа алгоритмов машинного обучения для решения задач классификации с использованием Pandas.

Квалификационная работа на тему «Система анализа алгоритмов машинного обучения для решения задач классификации с использованием Pandas» посвящена созданию системы, позволяющей анализировать алгоритмы классификации в машинном обучении.

Цель работы заключается в исследовании методов, моделей и библиотек машинного обучения для их дальнейшего анализа и сравнения, также предоставление пользователю результатов анализа алгоритмов машинного обучения для решения задач классификации с использованием Pandas.

В процессе выполнения квалификационной работы бакалавра было проведено исследование в выбранной предметной области, изучены подходящие методы и модели для решения задачи, определены критерии оценки качества разрабатываемой системы, а также разработано приложение в виде программного продукта.

Пояснительная записка содержит 3 приложения.

СОДЕРЖАНИЕ

[ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ 6](#_Toc169620619)

[ВВЕДЕНИЕ 7](#_Toc169620620)

[1 ПОСТАНОВКА ЗАДАЧ РАЗРАБОТКИ 9](#_Toc169620621)

[1.1.1 Постановка задачи разработки 9](#_Toc169620623)

[1.1.2 Описание предметной области 9](#_Toc169620624)

[1.1.3 Выбор критериев качества 10](#_Toc169620625)

[1.1.4 Анализ прототипов и аналогов 12](#_Toc169620626)

[1.1.5 Классификация в машинном обучении 18](#_Toc169620627)

[2 КОНСТРУКТОРСКО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ 29](#_Toc169620641)

[2.1.1 Выбор программных средств 29](#_Toc169620643)

[2.1.1.1 Выбор языка программирования 29](#_Toc169620644)

[2.1.1.2 Выбор фреймворка для разработки 30](#_Toc169620645)

[2.1.1.3 Выбор среды разработки 32](#_Toc169620646)

[2.1.1.4 Выбор технологии создания веб-приложения 32](#_Toc169620647)

[2.1.1.5 Выбор аппаратных средств 33](#_Toc169620648)

[2.2.1 Разработка системы 34](#_Toc169620650)

[2.2.1.1 Загрузка данных и анализ датасета 34](#_Toc169620651)

[2.2.1.2 Предварительная обработка данных 35](#_Toc169620652)

[2.2.1.3 Кодирование категориальных признаков 35](#_Toc169620653)

[2.2.1.4 Масштабирование данных 36](#_Toc169620654)

[2.2.1.5 Корреляционный анализ данных 38](#_Toc169620655)

[2.2.1.6 Формирование обучающей и тестовой выборки 40](#_Toc169620656)

[2.2.1.7 Обучение моделей 41](#_Toc169620657)

[2.2.1.8 Сравнение результатов моделей и вывод 42](#_Toc169620658)

[2.2.2 Разработка веб-приложения 45](#_Toc169620659)

[ЗАКЛЮЧЕНИЕ 53](#_Toc169620660)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 54](#_Toc169620661)

[ПРИЛОЖЕНИЕ A ГРАФИЧЕСКАЯ ЧАСТЬ 56](#_Toc169620662)

[A.1. Диаграммы сравнения оценки моделей 57](#_Toc169620663)

[A.2. Экранные формы 59](#_Toc169620664)

[ПРИЛОЖЕНИЕ B ТЕХНИЧЕСКОЕ ЗАДАНИЕ 64](#_Toc169620665)

[ПРИЛОЖЕНИЕ C ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ 70](#_Toc169620666)

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

1. **Система** – система анализа алгоритмов машинного обучения.
2. **Датасет** – набор данных.
3. **Python** – высокоуровневый язык программирования.
4. **Фреймворк –** программная платформа, определяющая структуру программной системы.
5. **Библиотека (пакет)** **–** набор кода для решения задач в определенной сфере деятельности.
6. **Pandas** – библиотека Python для работы с данными.
7. **DataFrame –** табличная структура данных библиотеки Pandas.
8. **EU** – набор данных **«**Безработица в Европейском Союзе».
9. **USA** – набор данных **«**Безработица в Америке, по штатам США».
10. **RU** – набор данных «Статистические данные о занятости и безработице среди населения по возрастным группам».

ВВЕДЕНИЕ

Сейчас объем данных, с которым мы имеем дело, постоянно растет. От области здравоохранения до финансов, от технологий до научных исследований, данные является ключевой ролью в понимании явлений, выявлении закономерностей и принятии обоснованных решений. В этом контексте машинное обучение превращается в важный инструмент, дающий возможность автоматизировать анализ данных, строить прогностические модели и принимать решения на основе данных.

Одним из самых распространенных типов задач в машинном обучении является классификация, которая представляет собой процесс присвоения категориальных меток на основе входных данных. Применение эффективных алгоритмов классификации позволяет решать множество прикладных задач, таких как диагностика заболеваний, обнаружения мошенничества и рекомендации продуктов

Данная дипломная работа предполагает анализ предметной области, изучение существующих алгоритмов машинного обучения и их оценивания. Система разработана для проведения анализа и оценки производительности различных моделей машинного обучения на примере данных, содержащих информацию о численности экономически активного населения, уровне занятости и безработицы по субъектам Российской Федерации.

Актуальность данной работы состоит в необходимости создания эффективных инструментов для анализа растущих объемов данных. Такая система способствует более осознанному и точному выбору алгоритмов классификации, улучшает процесс обработки данных и предоставляет широкие возможности для применения в различных отраслях.

Разработка такой системы будет полезна как для исследователей и специалистов по данным, так и для студентов и преподавателей, занимающихся изучением методов машинного обучения.

Система позволит упростить процесс анализа данных, ускорить обучение алгоритмов и повысить точность предсказаний, что в конце концов приведет к информированным решениям и глубокому пониманию анализируемых данных.

# ПОСТАНОВКА ЗАДАЧ РАЗРАБОТКИ

* 1. Общетехническое обоснование разработки
     1. Постановка задачи разработки

Разрабатываемая система предназначена для пользователей, желающих получить данные об анализе алгоритмов машинного обучения для решения задач классификации с использованием Pandas на примере предметной области «Статистические данные о занятости и безработице среди населения по возрастным группам».

Целью разработки является создание удобного инструмента для анализа и сравнения различных алгоритмов классификации, показ пользователю результатов анализа алгоритмов машинного обучения с использованием Pandas.

Для достижения заданной цели разработки важно выполнить следующие задачи:

* Исследование предметной области.
* Выбор алгоритмов машинного обучения.
* Обучение моделей.
* Оценивание качества каждой модели.
* Выбор модели с наилучшей производительностью.
* Вывод результатов анализа.
  + 1. Описание предметной области

Ключевая задача «Системы анализа алгоритмов машинного обучения с использованием Pandas» заключается в создании системы, который позволяет исследователям и аналитикам проводить анализ и сравнение различных алгоритмов машинного обучения с использованием библиотеки Pandas в языке программирования Python.

Предметной областью данной разработки является набор данных «Статические данные о занятости и безработице среди населения по возрастным группам», который предоставляет информацию о численности экономически активного населения, безработных, уровне безработицы и сопоставляет эти показатели между различными возрастными группами по субъектам РФ.

Набор данных включает в себя следующие показатели:

1. Численность экономически активного населения – всего;
2. Численность населения, занятого в экономике;
3. Численность безработных;
4. Численность экономической активности;
5. Уровень занятости;
6. Уровень безработицы;
7. Распределение безработных по возрастным группам по регионам РФ;
8. Численность безработных по возрастным группам по регионам РФ;
9. Распределение занятых в экономике по возрастным группам по регионам РФ;
10. Численность занятых в экономике по возрастным группам по регионам РФ.

Данные представлены за период 2001-2019 г.

* + 1. Выбор критериев качества

Каждый датасет имеет определенное число параметров, которые вычисляют его характеристики и пригодность для применения в обучении алгоритмов машинного обучения.

Давайте определим основные критерии качества набора данных, которые являются самым важным для решения поставленной задачи разработки:

* Полнота данных
* Точность данных
* Актуальность данных
* Понятность и интерпретируемость
* Доступность

**Полнота данных** имеет большое значение для обеспечения корректности и достоверности анализа данных, так как пропущенные значения могут искажать результаты и влиять на точность моделей машинного обучения. Не должно быть пропущенных значений, и данные должны быть достаточно подробными. Высокая полнота данных гарантирует надежность и точность анализа данных, что в свою очередь оказывается важным фактором для принятия решений на основе данных.

**Точность данных** необходима для соответствия данных реальным значениям и фактам. Это включает в себя проверку данных на наличие ошибок или неточностей. Точные данные позволяют делать более точные анализы, прогнозы и принимать более обоснованные решения на основе данных.

**Актуальность данных** важна, чтобы оценивать, насколько недавно были собраны или обновлены данные и насколько они релевантны для текущих потребностей анализа. Информация в наборе данных должна быть актуальной и соответствовать текущему состоянию проблемы или явления, которую мы хотим исследовать.

**Понятность и интерпретируемость** необходима для понимания и анализа данных пользователями без необходимости специальных навыков в области анализа данных. Данные должны быть легко понятны и интерпретируемы для аналитиков и исследователей. Это содержит хорошее описание переменных, их типов и значений.

**Доступность** важна для получения, использования данных в удобном и эффективном формате. Данные должны быть доступными для широкого круга пользователей, предпочтительно в открытом доступе или через официальные источники.

* + 1. Анализ прототипов и аналогов

Рассмотрим несколько популярных наборов данных, используемых для изучения безработицы с помощью машинного обучения:

1. «Безработица в Европейском Союзе» (**EU**), в котором собрана информация о безработице в странах Европейского Союза, разбитая по полу и возрасту. Временной интервал с января 1983 года до июля 2020 года. [1]

2. **«**Безработица в Америке, по штатам США» (**USA**), который отслеживает соответствующую статистику населения и уровень занятости в каждом штате США с 1976 года по декабрь 2022 г. [2]

3. «Статистические данные о занятости и безработице среди населения по возрастным группам» (**RU**) предоставляет информацию о численности рабочей силы и остального экономически активного населения, безработных, уровне безработицы в разных возрастных группах по субъектам РФ. Данные представлены за период 2001-2019 г. [3]

В таблице 1 указаны обозначения для каждого набора данных, которые будут использоваться при обучении алгоритмов машинного обучения методом взвешенной суммы. Для определения важности критериев мы будем применять метод базового критерия.

Таблица 1— Варианты для выбора набора данных

|  |  |
| --- | --- |
| Вариант | Обозначение |
| EU | В1 |
| USA | В2 |
| RU | В3 |

Критерии для сравнения перечисленных вариантов набора данных представлены в таблице 2.

Таблица 2 — Критерии для сравнения вариантов набора данных

|  |  |
| --- | --- |
| Критерий | Код критерия |
| Полнота данных | К1 |
| Точность данных | К2 |
| Актуальность данных | К3 |
| Понятность и интерпретируемость | К4 |
| Доступность | К5 |

Качественные аспекты аналогов были преобразованы в количественные критерии, которые представлены в таблицах 3-7.

Таблица 3 — Вербально-числовая шкала для критерия К1

(Полнота данных)

|  |  |  |  |
| --- | --- | --- | --- |
| Качественное описание критерия | Большая | Средняя | Малая |
| Балл | 3 | 2 | 1 |

Таблица 4 — Вербально-числовая шкала для критерия К2

(Точность данных)

|  |  |  |  |
| --- | --- | --- | --- |
| Качественное описание критерия | Высокая | Средняя | Низкая |
| Балл | 3 | 2 | 1 |

Таблица 5 — Вербально-числовая шкала для критерия К3

(Актуальность данных)

|  |  |  |  |
| --- | --- | --- | --- |
| Качественное описание критерия | Большая | Средняя | Малая |
| Балл | 3 | 2 | 1 |

Таблица 6 — Вербально-числовая шкала для критерия К4

(Понятность и интерпретируемость)

|  |  |  |  |
| --- | --- | --- | --- |
| Качественное описание критерия | Большая | Средняя | Малая |
| Балл | 3 | 2 | 1 |

Таблица 7 — Вербально-числовая шкала для критерия К5

(Доступность)

|  |  |  |  |
| --- | --- | --- | --- |
| Качественное описание критерия | Большая | Средняя | Малая |
| Балл | 3 | 2 | 1 |

Таблица 8 содержит определения количественных и качественных параметров рассматриваемых языков программирования.

Таблица 8 – количественные и качественные параметры

|  |  |  |  |
| --- | --- | --- | --- |
| Код критерия | Вариант набора данных | | |
| В1 | В2 | В3 |
| К1 | 1 | 2 | 3 |
| К2 | 2 | 2 | 1 |
| К3 | 3 | 2 | 1 |
| К4 | 1 | 1 | 2 |
| К5 | 1 | 1 | 1 |

Затем требуется оценить выбранные наборы данных на соответствие критериям Парето-оптимальности. Результаты анализа приведены в таблице 9.

Таблица 9 – Сравнение вариантов на Парето-оптимальность

|  |  |  |  |
| --- | --- | --- | --- |
| Вариант набора данных | Вариант набора данных | | |
|  | В1 | В2 | В3 |
| В1 | 0 | 0 | 0 |
| В2 | 0 | 0 | 0 |
| В3 | 0 | 0 | 0 |
| Результат сравнения | 0 | 0 | 0 |
| Парето-оптимальность варианта | Да | Да | Да |

Все рассматриваемые варианты являются Парето-оптимальными, поэтому продолжим сравнение между ними.

Для определения важности критериев используем метод базового критерия. Мы разделим все критерии на группы по уровню их значимости:

1. В первую группу включим критерий, который считаем менее значимым из рассматриваемых: 2;
2. Во вторую группу войдут критерии, которые мы считаем наиболее значимыми по сравнению с первой группой: 3, 5;
3. В третью группу включим критерии, которые рассматриваем как наиболее значимые: 1, 4;

Соответственно, имеем следующие исходные данные:

g = 3 – количество групп важностей локальных критериев сравнения наборов данных.

𝑛1 = 1; 𝑛2 = 2; 𝑛3 = 2 – количество локальных критериев, соответственно входящих в первую, вторую и третью группы.

𝑘1 = 1; 𝑘2 = 2; 𝑘3 = 4 – коэффициенты, отражающие уровень превосходства второй и третьей групп над критериями первой группы.

Составляем уравнение для нормализации локальных критериев:

Следовательно, согласно уравнению (1), получаем:

Получаем значение коэффициента важности:

Затем определяем коэффициенты важности для каждой из групп:

Для выравнивания значений локальных критериев используем соответствующие формулы в случае, когда нормализация осуществляется для критериев с принципом «чем выше, тем лучше»:

где

Если требуется привести критерии к нормальному виду с учетом условия «чем меньше, тем лучше»:

где ,

Коэффициент нормализации определяет, насколько параметр i варианта j соответствует идеальному значению, где .

Для каждого критерия нормализуем значения для всех вариантов.

В таблице 10 представлены итоговые значения и коэффициенты важности локальных критериев.

Согласно методу взвешенной суммы для каждого варианта необходимо вычислить сумму произведений коэффициента важности локального критерия и коэффициента нормализации (формула 4)

Таблица 10 — Нормализованные значения показателей вариантов сравнения

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Код критерия** | **Коэффициент важности локального**  **критерия (**𝜶𝒊**)** | **Нормированное значение локального**  **критерия** | | |
| 𝑘𝑖1 | 𝑘𝑖2 | 𝑘𝑖3 |
| К1 | 0,308 | 0,33 | 0,67 | 1 |
| К2 | 0,077 | 1 | 1 | 0,5 |
| К3 | 0,154 | 1 | 0,67 | 0,33 |
| К4 | 0,308 | 0,5 | 0,5 | 1 |
| К5 | 0,154 | 1 | 1 | 1 |
| 𝑛  𝑌𝑗 = ∑ 𝛼𝑖 ∙ 𝑘𝑖𝑗  𝑖 | | 0,64 | 0,69 | 0,86 |

Для определения наилучшего варианта используем формулу (5):

где m – количество вариантов сравнения.

Исходя из приобретенных результатов исследования, сделаем вывод, что самым оптимальным выбором для выполнения заданной задачи оказывается вариант В3 — датасет **RU**. Этот вывод основан на анализе собранных данных и учете их соответствия требованиям и целям данного исследования.

* + 1. Классификация в машинном обучении
  1. Понятие классификации в машинном обучении

Машинное обучение (Machine Learning) — это важная область искусственного интеллекта, фокусирующаяся на разработке алгоритмов и моделей, способных автоматически извлекать закономерности из данных и использовать их для принятия решений или прогнозирования будущих данных.

Основная цель машинного обучения заключается в том, чтобы обучить компьютерные системы не только определять шаблоны в данных, но и использовать их для прогнозирования и принятия решений. Это позволяет компьютерам улучшать свою производительность и адаптироваться к изменениям в окружающей среде, обеспечивая более эффективное решение задач различной сложности — от анализа больших объемов информации до создания интеллектуальных систем поддержки принятия решений.

Контролируемое обучение или обучение с учителем (Supervised learning) — это важная подкатегория машинного обучения, где модель обучается на наборе данных, где каждый пример сопоставлен с желаемым выходом. Это позволяет модели прогнозировать выходные данные для новых входных данных. Такой метод широко используется в задачах классификации, регрессии и обнаружения объектов [4].

Классификация в контексте машинного обучения является процессом разделения данных или объектов на заранее определенные классы или категории на основе их характеристик или атрибутов.

Модели классификации в машинном обучении обучаются на помеченных данных с целью прогнозировать класс или категорию для новых, неизвестных данных.

Основная задача классификационного подхода — создать модель, которая точно может определять метку или категорию для новых наблюдений на основе их характеристик [5].

Алгоритм классификации, базирующийся на обучающих данных, представляет собой метод контролируемого обучения, который применяется для классификации новых наблюдений. В процессе классификации программа использует предоставленный набор данных или образцов, чтобы научиться различать новые наблюдения по различным категориям или группам. Эти категории могут быть представлены, например, числами 0 или 1, цветами «красный» или «синий», ответами «да» или «нет», или определением спама и того, что не является спамом и так далее. Классы могут быть описаны с использованием целей, меток или категорий. Алгоритм классификации использует помеченные входные данные, ибо это метод обучения с учителем, предполагающий наличие входной и соответствующей выходной информации.

На рисунке 1 изображены два класса: класс A и класс B. Эти классы обладают схожими и уникальными характеристиками, что отличает их от других классов [6].

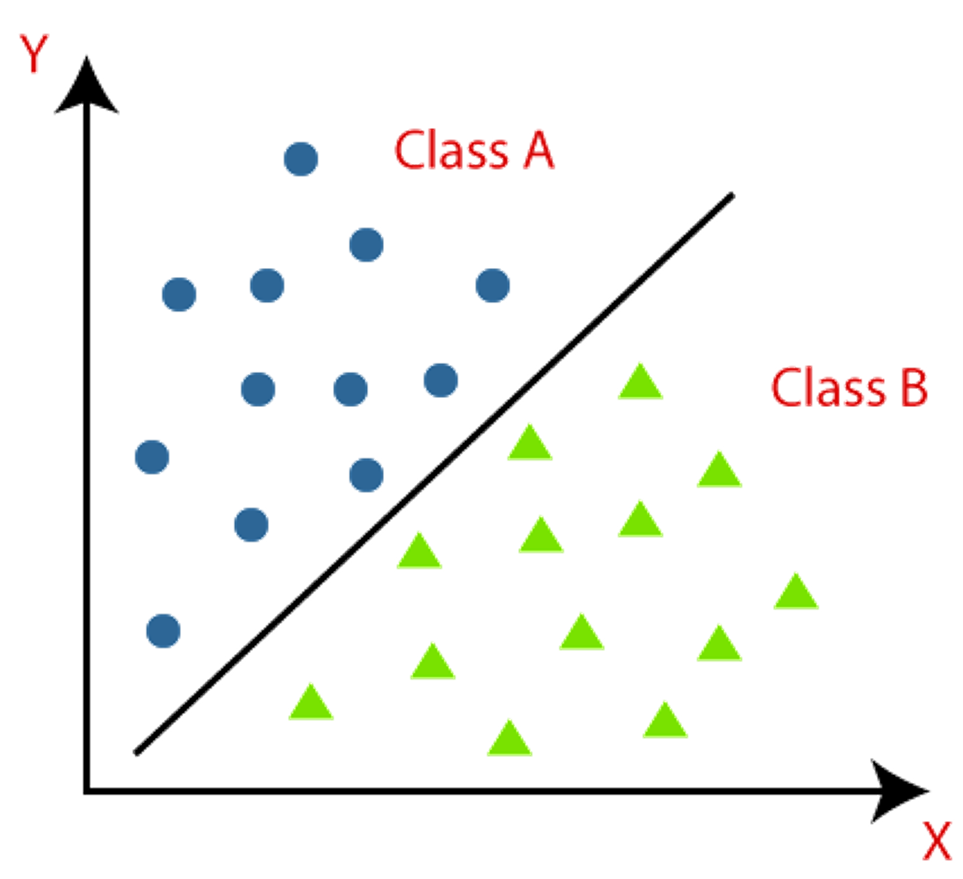


Рисунок 1—Классификация

Примером задачи классификации может служить борьба со спамом в электронной почте. В данном случае модель машинного обучения обучается узнавать особенности электронных сообщений, такие как определенные слова или фразы, отправители и другие аспекты, и классифицировать сообщения как «спам» или «не спам». Завершенная обученная модель автоматически фильтрует входящие сообщения, благодаря чему пользователь не сталкивается с нежелательной почтой.

* 1. Описание алгоритмов классификации

В области машинного обучения применяются разнообразные алгоритмы классификации, помогающие определить категориальные метки (классы) на основе предоставленных входных данных. На текущий момент существует несколько известных и часто используемых алгоритмов классификации, которые описаны ниже.

**Логистическая регрессия (Logistic Regression)**

Логистическая регрессия является методом контролируемого обучения, применяемого для задач бинарной классификации, где присутствуют два возможных класса. Основная идея заключается в том, чтобы вычислить вероятность принадлежности наблюдения к конкретному классу, используя логистическую функцию, изменяющую линейную комбинацию независимых переменных в вероятность от 0 до 1. Целью логистической регрессии является нахождение наилучшей модели, которая описывает отношение между зависимой переменной и набором независимых переменных.

Графически логистическая регрессия представляется сигмовидной кривой, которая плавно изменяется от 0 до 1 в зависимости от значений предикторов. Эта кривая наглядно демонстрирует, как вероятность отношения к классу меняется в зависимости от значений независимых переменных. Пример такой кривой показано на рисунке 2, иллюстрирующем особенности логистической регрессии в контексте бинарной классификации.

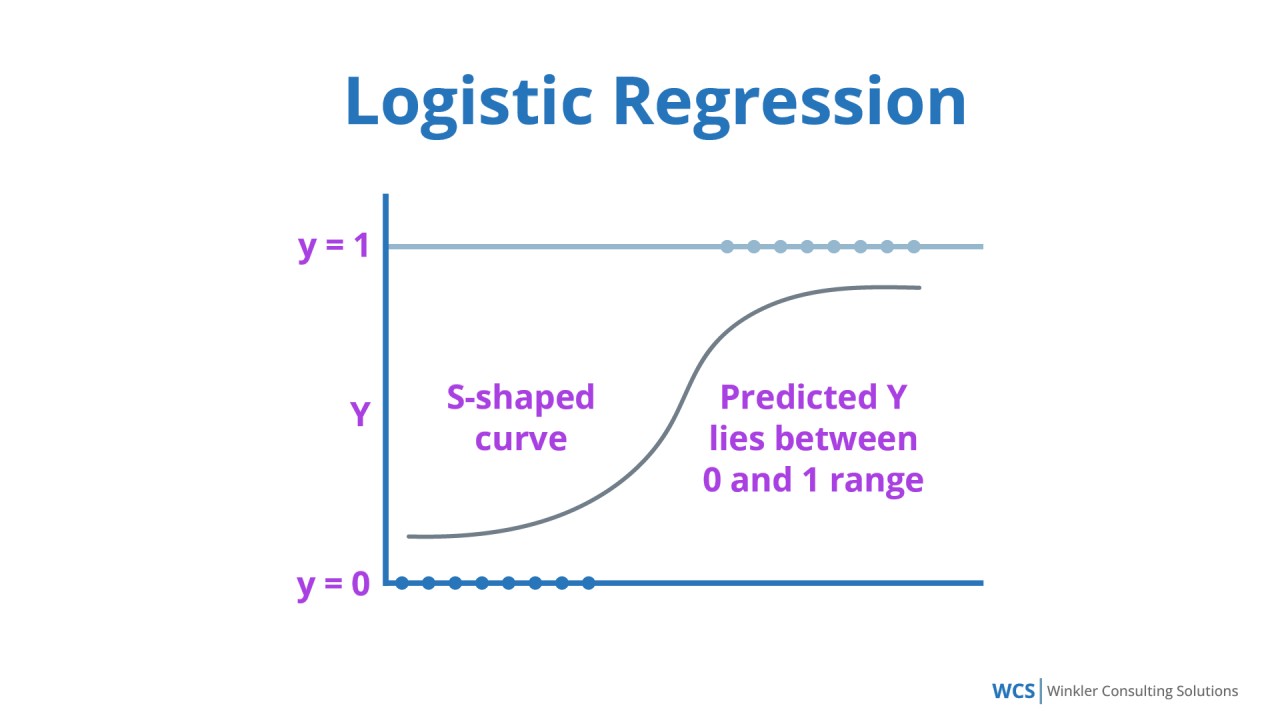


Рисунок 2 – Логистическая регрессия

**К-ближайших соседей (KNN)**

Алгоритм k-ближайших соседей (KNN) применяется для классификации новых данных на основе класса, к которому принадлежит большинство из k ближайших соседей в обучающем наборе данных. Обозреваемый метод основан на простом принципе подсчёта числа объектов каждого класса в окрестности, определяемой расстоянием до классифицируемого объекта. Классификация объекта происходит путём отнесения его к классу, который наиболее представлен в данной окрестности. Визуализация работы алгоритма К-Nearest Neighbors продемонстрирована на рисунке 3, иллюстрирующем процесс определения класса для нового объекта на основе его ближайших соседей в пространстве признаков.

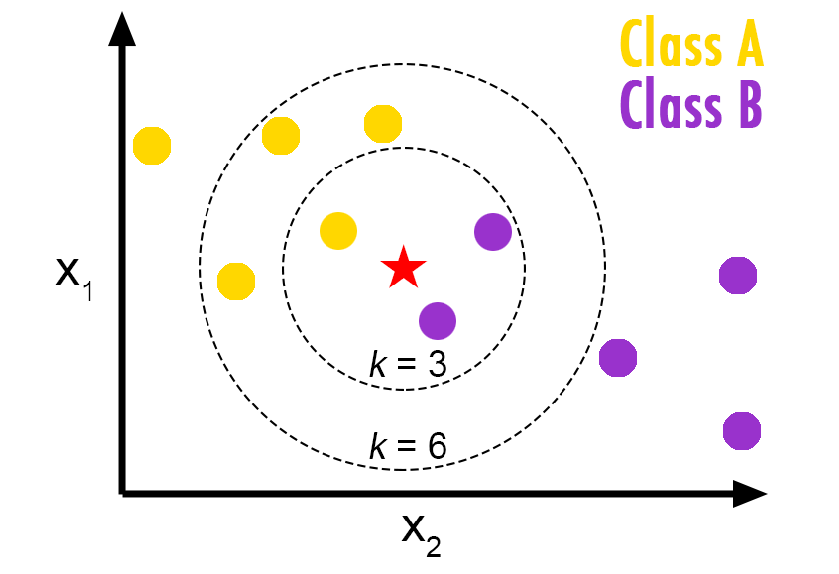


Рисунок 3 - KNN-ближайших соседей

В данном случае у нас есть два класса данных, обозначенные как A и B. Мы рассматриваем задачу предсказания класса для тестовой точки данных, изображающей звезду. При использовании значения k равного 3 (три ближайшие точки данных), алгоритм k-ближайших соседей прогнозирует класс B. Если же значение k равно 6, то прогноз будет отнесён к классу A.

**Случайный лес (Random Forest)**

Случайный лес (Randon Forest) – это ансамблевый метод, который строит множество деревьев решений и объединяет их результаты для улучшения точности и контроля переобучения. Его разработал Лео Брейман и Адель Катлер. Каждое дерево в случайном лесу обучается на различном подмножестве данных, а случайность, добавленная в процессе обучения, помогает улучшить общие показатели модели и ее обобщающие способности.

При обучении модели строятся множественные деревья решений, каждое из которых обучается на случайном подмножестве данных.

Случайность вносится в процесс обучения на двух уровнях:

1. Случайное подмножество данных. Для каждого дерева решений из исходного набора данных извлекается случайное подмножество определенного размера.
2. Случайное подмножество признаков. При построении каждого дерева из случайного подмножества данных случайно выбирается подмножество признаков.

После того, как все деревья решений построены с различной структурой и характеристиками, начинает решаться вопрос принадлежности нового объекта к классу. Итоговый прогноз модели определяется путём усреднения результатов всех деревьев, что помогает повысить точность и устойчивость предсказаний.

Размер каждой подвыборки всегда совпадает с исходным размером входных данных, но выборки часто осуществляются с заменами, что означает возможность повторного использования одних и тех же данных в разных подвыборках. Этот подход, известный как бутстреппинг, позволяет модели лучше справляться с вариативностью данных и снижает риск переобучения.

Визуализация работы алгоритма случайного леса, которая наглядно демонстрирует процесс создания и объединения результатов множества деревьев решений для получения точного и надежного прогноза, продемонстрирована на рисунке 4.

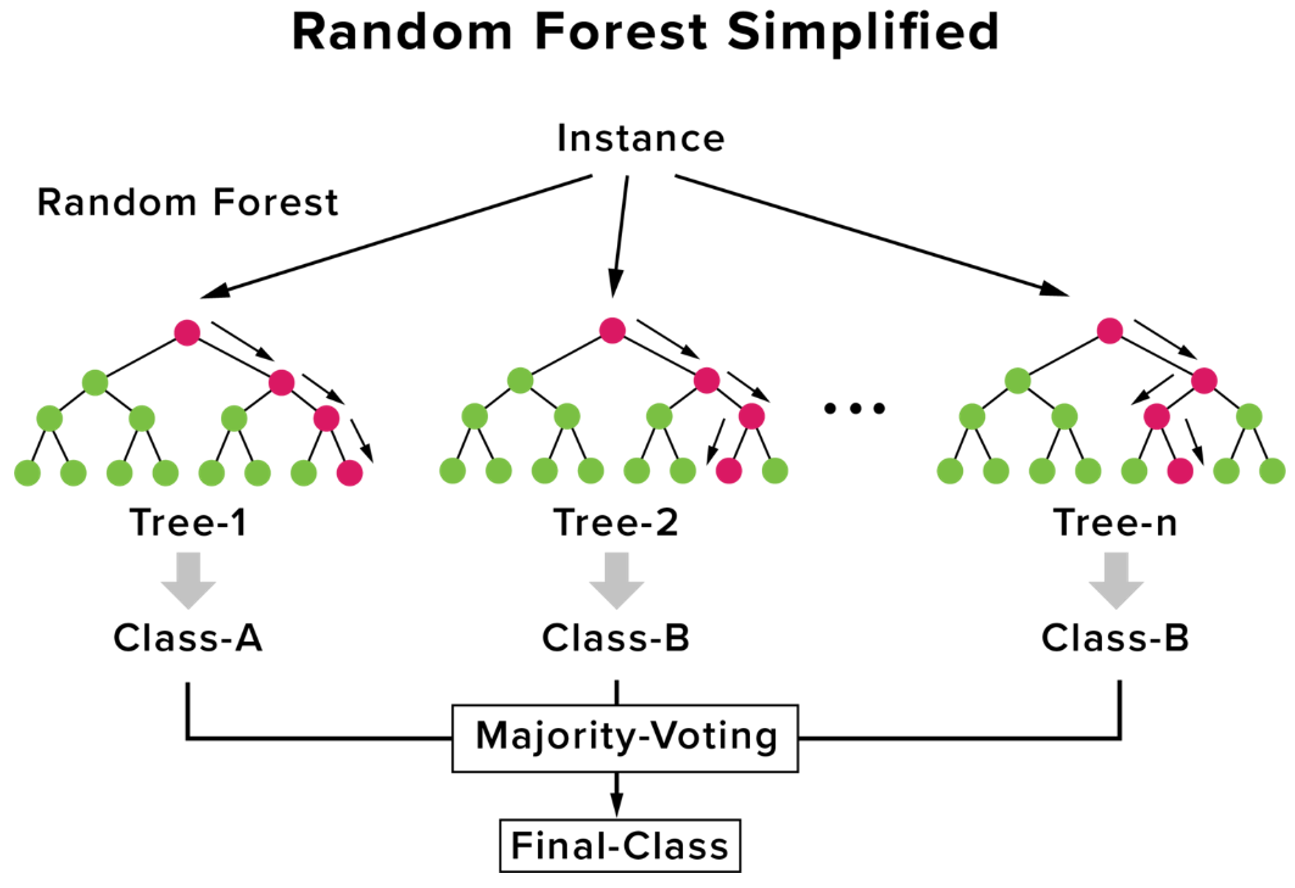


Рисунок 4 - Случайный лес

* 1. Описание критериев сравнения моделей классификации

Оценивание моделей классификации является ключевым аспектом в области машинного обучения, поскольку позволяет оценить эффективность модели в решении поставленной задачи. Основные критерии оценки моделей классификации включают метрики точности, полноты, точности и F1- меры и другие.

Следующим этапом является выбор метрик для оценки модели. Для задач классификации обычно выбирают стандартные метрики, такие как precision, recall, F1-score и accuracy [7].

Расчет одной из этих метрик включает использование матрицы ошибок. При расчёте каждой из этих метрик применяется матрица ошибок. Матрица ошибок позволяет наглядно оценить производительность классификационного алгоритма путем сопоставления ожидаемых и реальных значений целевой переменной.

Таблица 11 – Матрица ошибок

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Экспертная оценка | |
| Положительная | Отрицательная |
| Оценка системы | Положительная | TP | FP |
| Отрицательная | FN | TN |

TP — истинно-положительный (true positive), это случай, когда классификатор правильно распознает и относит объект к соответствующему классу или категории.

TN — истинно-отрицательный (true negative), это ситуация, когда классификатор корректно определяет, что объект не относится к соответствующему классу.

FP — ложно положительный (false positive), данная ситуация возникает, когда классификатор ошибочно относит объект к определенному классу, в то время как на самом деле он не принадлежит к нему.

FN — ложно отрицательный (false negative), это случай, когда классификатор некорректно доказывает, что объект не относится исследуемому классу, хотя на самом деле он относится к нему.

**Accuracy**

Доля правильных ответов – относительное количество корректно классифицированных объектов (процент или доля правильно классифицированных объектов).

Метрика accuracy показывает, насколько хорошо обучена модель и её функционирование в общих чертах. Только она не предоставляет детальной информации о том, как модель будет применяться к конкретной задаче.

Основной недостаток применения accuracy в роли главной метрики производительности состоит в том, что она становится неэффективной при наличии серьёзного дисбаланса классов. В таких ситуациях высокий процент правильных ответов может быть достигнут за счет преобладания одного из классов, поэтому для адекватной оценки модели необходимо применять и другие метрики.

**Precision**

Precision будет показывать часть правильно обнаруженных объектов заданного класса по отношению к общему числу объектов, принятых классификатором за объекты заданного класса. Часто применяется, когда важно учесть число ложноположительных предсказаний. В кое-каких случаях, предположим, при выявлении спама, ложное включение представляет собой более серьёзную ошибку, чем пропуск события.

**Recall**

Recall – это антипод precision, являющаяся ключевой метрикой в задачах классификации, оценивающая умение модели обнаруживать все положительные случаи в данных. Она показывает отношение правильно идентифицированных объектов к объему числу объектов данного класса. Определяет количество ложноотрицательных результатов в сравнении с истинно положительными. Это показатель, который подчеркивает важность минимизации ложноотрицательных результатов, особенно в областях, где жизненно важно выявить каждый положительный случай, таких как медицинская диагностика и безопасность.

Recall используется для оценки эффективности модели в выявлении редких событий или случаев, когда ошибки пропуска носят критическое значение. Например, в медицинской диагностике ложноотрицательные результаты могут означать неопределенность в определении заболевания, что может привести к неправильному лечению или недостаточному вмешательству. В таких случаях высокий recall важен для обеспечения надежности модели в выявлении всех возможных случаев заболевания, минимизируя пропущенные диагнозы.

Таким образом, recall представляет собой критически важную метрику, которая дополняет другие метрики, такие как precision, для обеспечения комплексной оценки производительности модели классификации.

**F1-score**

Если требуется найти баланс между двумя целями: высокой точностью и высокой полнотой, то используется F1-score.

F1-оценка вычисляется, как среднее пропорционально среди точности и полноты. По сопоставлению с обычным арифметическим средним, пропорциональное среднее надёжнее к экстремальным значениям, что делает F1-меру предпочтительной для равномерной оценки моделей классификации.

F1-мера играет важную роль в количественной оценке качества моделей в разных сферах. Она предоставляет равновесное состояние между точностью и отзывчивостью, что позволяет более полно и объективно оценить производительность моделей.

# КОНСТРУКТОРСКО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

* 1. Конструкторская часть
     1. Выбор программных средств
        1. Выбор языка программирования

В качестве языка программирования для разработки был выбран язык Python.

**Python** – это универсальный язык программирования высокого уровня, который отличается своей динамической строгой типизацией и автоматическим управлением памятью.

Важной особенностью Python является его объектно-ориентированный характер. В Python всё – от простых переменных до сложных структур данных – являются объектами. Это позволяет использовать единый подход к работе с данными, упрощая разработку и поддержку кода.

Python также отличается уникальным синтаксисом, который использует отступы для выделения блоков кода. Синтаксис ядра языка минималистичен, за счёт чего на практике редко появляется важность обращаться к документации. В результате Python считается самым простым языком программирования, поэтому он самый популярный.

Python обладает богатой библиотекой для работы с данными, такими как NumPy, Pandas и Matplotlib, которые делают анализ данных более эффективным и удобным. Благодаря этому, специалисты в области аналитики и машинного обучения предпочитают использовать Python для своей работы. Кроме того, Python легко расширяем и поддерживает интеграцию с другими языками программирования, что делает его универсальным инструментом для разработки различных проектов [16].

Кроме того, Python активно поддерживается сообществом разработчиков и имеет обширную документацию, что упрощает процесс изучения и использования языка. Благодаря этому, даже специалисты со средним уровнем навыков могут быстро освоить Python и начать работу над проектами анализа данных и машинного обучения.

Все это делает Python оптимальным выбором для специалистов, занимающихся обработкой и анализом данных, за счёт его большой производительности, удобства использования и широким возможностям для разработки.

* + - 1. Выбор фреймворка для разработки

В процессе решения задач разработки были использованы следующие фреймворки и библиотеки:

**Pandas** – библиотека для анализа данных, построенный на языке программирования Python. Она включает в себя многофункциональные инструменты для операций с данными, включая сортировку, фильтрацию, очистку, удаление дубликатов, агрегирование, создание сводных таблиц и множество других операций. Являясь центром обширной экосистемы исследования данных, реализованной в среде языка Python, pandas хорошо сочетается с другими библиотеками для статистики, обработки естественного языка, машинного обучения, визуализации данных и многого другого [9].

Pandas построена на основе структуры данных, называемой DataFrame. **DataFrame** является таблицей, похожая на электронную таблицу Excel.

Библиотека pandas предлагает большой спектр методов по работе с этой таблицей, в частности, она позволяет выполнять SQL-подобные запросы и присоединения таблиц. В отличие от NumPy, который требует, чтобы все записи в массиве были одного и того же типа, в pandas каждый столбец может иметь отдельный тип, например, целые числа, даты, числа с плавающей точкой и строки. Еще одним превосходством библиотеки pandas значится её способность работать с различными форматами файлов и баз данных, например, с файлами SQL, Excel и CSV.

**NumPy** – библиотека с открытым исходным кодом для языка программирования Python, известная своими мощными возможностями. Её удобство включает в себя поддержку многомерных массивов (включая матрицы) и поддержку высокоуровневых математических функций, рассчитанный для работы с многомерными массивами. Библиотека NumPy предоставляет реализации вычислительных алгоритмов (в виде функций и операторов), оптимизированные для работы с многомерными массивами [17].

**Matplotlib** – библиотека для построения научных графиков в Python. Её функционал включает в себя функции для создания высококачественных визуализаций. Среди основных типов графиков, которые можно создавать с помощью Matplotlib, есть линейные диаграммы, гистограммы, диаграммы разброса и многие другие [10]. Matplotlib предоставляет разнообразные возможности для настройки внешнего вида графиков, что позволяет исследователям и разработчикам создавать информативные и эстетически привлекательные визуализации данных.

**Seaborn** – библиотека визуализации данных для языка программирования Python на основе библиотеки Matplotlib. Она предназначена для построения статических графиков, которые помогают визуально представить информацию из наборов данных [11]. Seaborn предлагает высокоуровневый интерфейс для создания разнообразных типов графиков, таких как гистограммы, ящики с усами (boxplots), тепловые карты и другие. Благодаря интеграции с Matplotlib, seaborn обеспечивает возможность дополнительной настройки внешнего вида графиков и улучшенного восприятия данных, что делает её популярным инструментом среди аналитиков и исследователей.

**Scikit-learn (sklearn)** – библиотека машинного обучения с открытым исходным кодом, которая поддерживает обучение с учителем и без учителя. Библиотека также предоставляет различные инструменты для подбора модели, предварительной обработки данных, выбора модели, оценки модели и многие другие утилиты [12]. Sklearn активно поддерживается сообществом разработчиков, имеет детализированную документацию и постоянно обновляется, включая в себя новые алгоритмы и функциональные возможности, что делает её одной из популярных и востребованных библиотек для машинного обучения в Python.

* + - 1. Выбор среды разработки

В качестве среды разработки выбран Jupyter Notebook.

**Jupyter Notebook** – интерактивная среда для запуска программного кода в браузере. Это отличное приспособление для разведочного анализа данных и широко используется специалистами по анализу данных [8].

Jupyter Notebook поддерживает множество языков программирования и позволяет легко интегрировать программный код, текст и графические элементы в одном документе. Она существует как веб-сервис, то есть доступна через интернет и позволяет передавать код другим разработчикам.

Чаще всего среду используют для Python, но она существует и для других

языков программирования. Jupyter Notebook поддерживает языки Ruby, R, MATLAB и другие. Часто это специализированные языки для задач, которые подразумевают быстрое написание и выполнение маленькой программы.

Разница между юпитер-ноутбуком и обычными инструментами разработки состоит в способности взаимодействия. С помощью этой программы можно выполнять отдельные фрагменты и блоки кода в любом порядке. Результаты работы моментально выводятся на экран рядом с кодом [15].

Основные сферы использования среды — big data и data science, машинное обучение, математическая статистика и аналитика. В этих направлениях используется Jupyter Notebook, чтобы выводить данные в том же месте интерфейса, где написан код.

* + - 1. Выбор технологии создания веб-приложения

Для создания веб-приложения был применеи фреймворк Streamlit. **Streamlit** – это платформа Python с открытым исходным кодом, специально разработанный для специалистов по данным и инженеров искусственного интеллекта и машинного обучения, работающих с Python, позволяющая создавать приложения с динамическими данными всего с помощью нескольких строк кода [13].

Этот фреймворк является одним из лучших для демонстрации проектов, связанных с анализом данных (data science). Он создает интерактивные веб-приложения благодаря нескольким строкам кода. Streamlit превращает скрипты данных в веб-приложения для совместного использования за считанные минуты [14].

Одной из ключевых особенностей Streamlit является его способность превращать скрипты данных в интерактивные веб-приложения за несколько минут. Этот фреймворк значительно упрощает процесс создания и развертывания проектов, предоставляя инструменты для быстрого взаимодействия с данными и визуализации результатов.

Каждое веб-приложение Streamlit – это скрипт Python, что обеспечивает удобство применения и объединения с существующими проектами. Он является могущим приспособлением для профессионалов, работающих в области анализа данных и машинного обучения, обеспечивая эффективное создание интерактивных приложений без необходимости в глубоких знаниях веб-разработки.

* + - 1. Выбор аппаратных средств

Для эффективного анализа алгоритмов машинного обучения и решения задач классификации с использованием библиотеки Pandas важно подобрать подходящие аппаратные средства.

Аппаратные компоненты компьютера являются совокупностью электрических, электронных и механических устройств, обеспечивающих его функционирование.

Локально исходный код системы запускался на ноутбуке со следующими характеристиками:

* Оперативная память – 8 ГБ
* Процессор AMD Ryzen 7 5700U with Radeon Graphics 1.80 GHz
* Тип системы – 64-разрядная операционная система, процессор x64
* Видеокарта AMD Radeon Graphics

Эти характеристики создают оптимальные условия для выполнения алгоритмов машинного обучения и обработки данных, использующих Pandas, обеспечивая быстродействие и эффективность анализа данных на локальном уровне.

* 1. Технологическая часть
     1. Разработка системы
        1. Загрузка данных и анализ датасета

В процессе работы с данными для машинного обучения, особенно при использовании библиотеки Pandas, ключевыми этапами являются загрузка данных и первоначальный анализ.

Данные для анализа были загружены из CSV файла c использованием библиотеки Pandas. Фрагмент кода загрузки данных приведен на рисунке 5.



Рисунок 5 - Загрузка датасета

После загрузки данных, был проведен анализ датасета. Изучили размер датасета, список столбцов, также убрали лишние столбцы, которые мешали дальнейшей разработке системы.

* + - 1. Предварительная обработка данных

Предварительная обработка данных является критическим этапом в подготовке данных для машинного обучения. В этом процессе осуществляется очистка, преобразование и нормализация данных для повышения качества и эффективности алгоритмов классификации.

После детального изучения датасета, была выполнена: замена названия колонок для улучшения читабельности, проверка и обработка пропущенных значений с помощью удаления строк и заменой пропущенных значений средним значением по столбцу, проверка дубликатов с помощью метода duplicated() и замена типов данных для кодирования категориальных признаков.

* + - 1. Кодирование категориальных признаков

Кодирование категориальных признаков – это процесс трансформации текстовых или категориальных значений признаков в цифровые, применяющая в моделях машинного обучения.

Целевая переменная – это переменная, которую модель пытается предсказать или моделировать в задаче обучения с учителем.

Взяли в качестве целевой переменной столбец «Год» и для решения задачи классификации выбраны два класса. Это 2001 и 2019 годы – рисунок 6.



Рисунок 6 - Целевая переменная

Используем метод Label Encoding для присваивания целочисленного значения категории «2001» и «2019». Теперь они преобразованы в целые числа 0 и 1 соответственно.

Фрагмент кода метода Label Encoding приведен на рисунке 7.

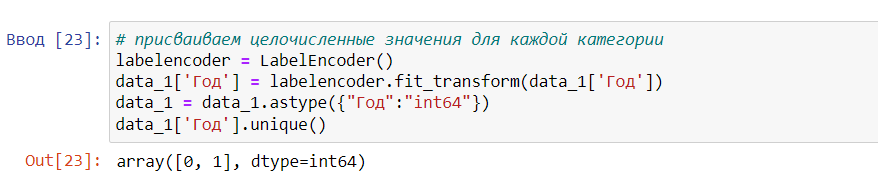


Рисунок 7 - Присваивание целочисленного значения

* + - 1. Масштабирование данных

Масштабирование данных – это этап изменения значений признаков в наборе данных таким образом, чтобы они находились в определенном диапазоне, обычно от 0 до 1 или со средним значением 0 и стандартным отклонением 1.

Был использован один из методов масштабирования – Min-Max масштабирование. Этот метод преобразует данные с помощью формулы (10) таким образом, чтобы они находились в заданном диапазоне, обычно от 0 до 1. Фрагмент кода метода приведен на рисунке 8.

Фрагмент кода Min-Max масштабирования приведен на рисунке 7.



Рисунок 8 - Преобразование значения признаков

Проверим, что масштабирование данных не повлияло на распределение данных с помощью рисунка 9. Это позволяет нам убедиться, что масштабирование не вносит нежелательные изменения в данные, что важно для корректной работы многих алгоритмов машинного обучения.

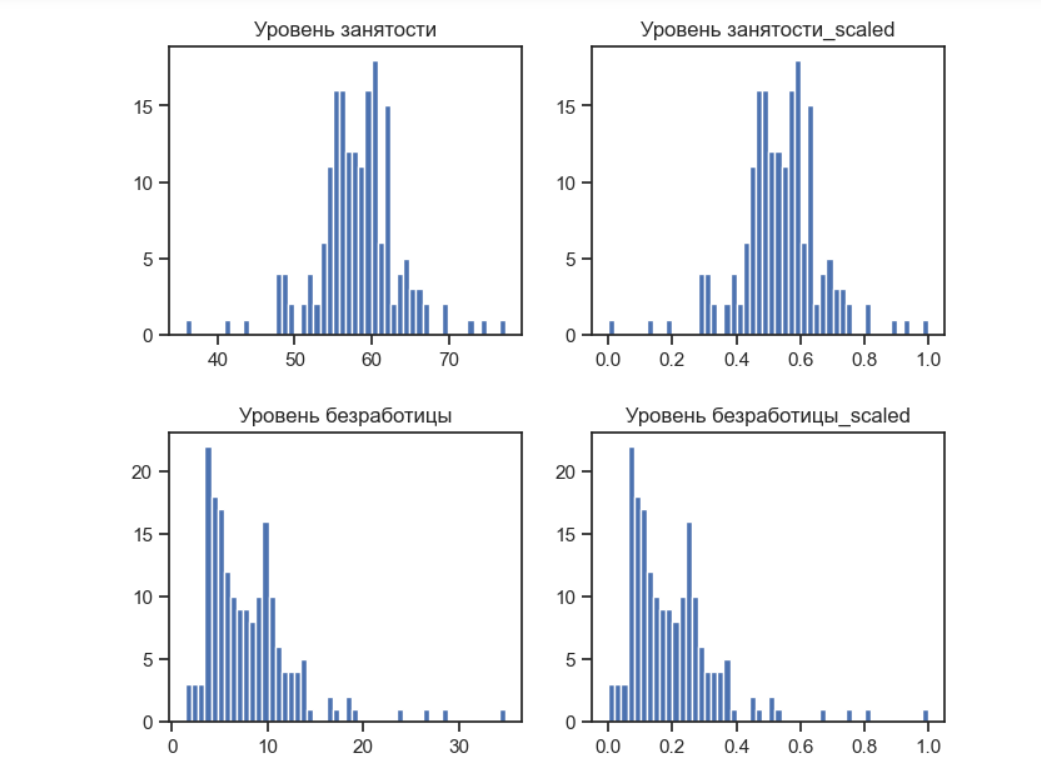


Рисунок 9 - Гистограммы

Масштабирование данных не повлияло на распределение данных.

* + - 1. Корреляционный анализ данных

Корреляционный анализ данных – статический метод, применяемый для оценки силы и направления взаимосвязи между двумя или более переменными. Он помогает определить, как изменения одной переменной могут быть связаны с изменениями другой.

Матрица корреляции – таблица, показывающая коэффициенты корреляции между множеством переменных. Она симметрична относительно главной диагонали, которая всегда содержит значения 1.

На рисунке 10 изображена матрица корреляции данных до масштабирования.

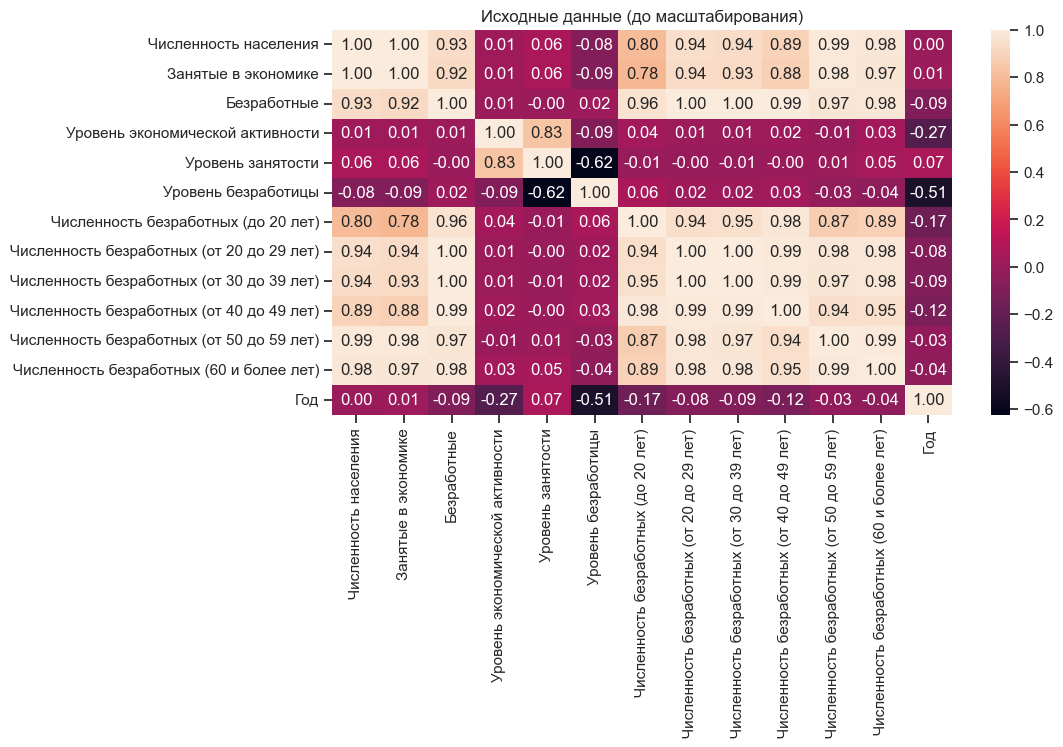


Рисунок 10 - Исходные данные

На рисунке 11 показана матрица корреляции масштабированных данных.

На основе корреляционных матриц сделаны следующие выводы:

Корреляционные матрицы для исходных и масштабированных данных совпадают.

Целевой признак классификации «Год» наиболее сильно коррелирует с «Уровнем занятости» (0.07), «Занятые в экономике» (0.01), «Численность населения» (0.00). Эти признаки обязательно следует оставить в модели классификации.

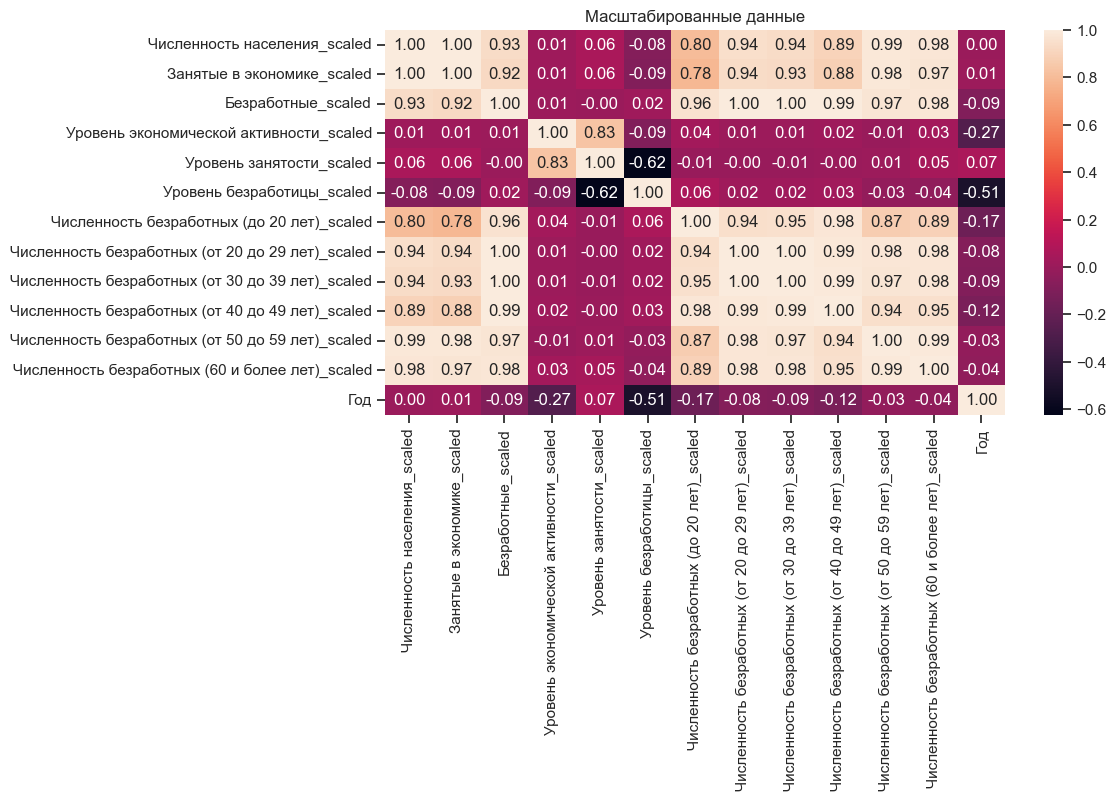


Рисунок 11 - Масштабированные данные

* + - 1. Формирование обучающей и тестовой выборки

Разделение на выборки для обучения и тестирования – это процесс разъединения доступного набора данных на обе части: одну для обучения модели (обучающую выборку) и другую для проверки её производительности (тестовую выборку). Этот процесс необходим для оценки обобщающей способности модели.

Деление выборки на обучающую и тестовую выборку приведено на рисунке 12.

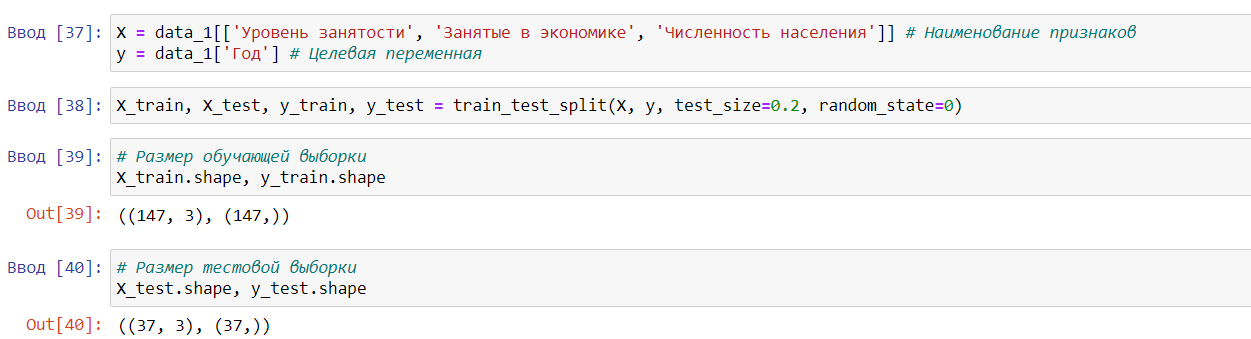


Рисунок 12 - Разделение данных

* + - 1. Обучение моделей

Для реализации алгоритмов классификации с использованием Pandas используется библиотека scikit-learn. Сначала нужно подготовить данные, затем выбрать алгоритмы классификации и обучить модели, используя методы из scikit-learn. После этого сравним модели с помощью метрик accuracy, precision, recall и F1-score.

Используем следующие алгоритмы классификации: логистическая регрессия, К-ближайших соседей (KNN), случайный лес.

Обучение модели логистической регрессии приведено на рисунке 13.

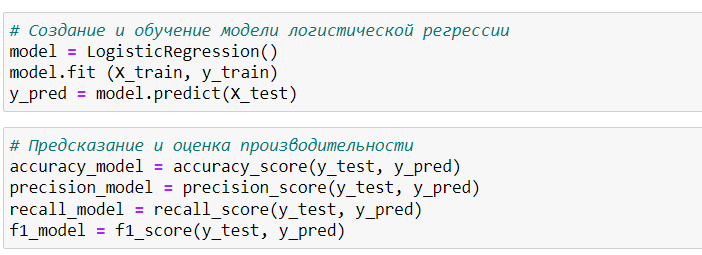


Рисунок 13 - Обучение логистической регрессии

Обучение модели К-ближайших соседей приведено на рисунке 14.

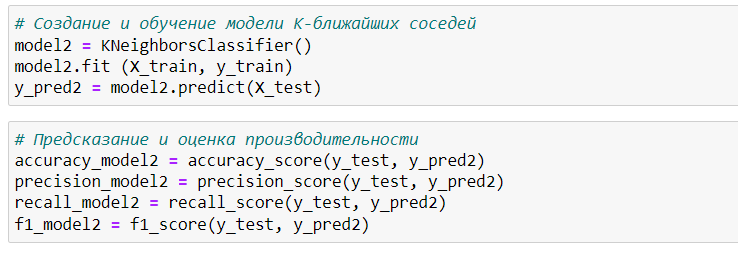


Рисунок 14 - Обучение K-ближайших соседей

Обучение модели случайный лес приведено на рисунке 15.

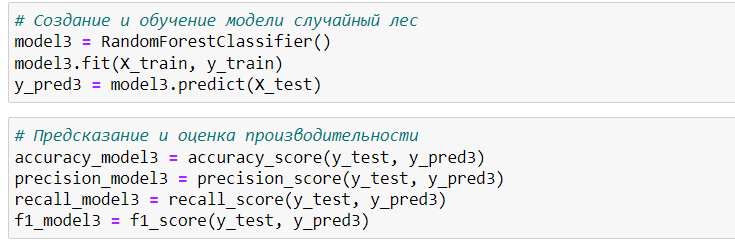


Рисунок 15 - Обучение случайного леса

* + - 1. Сравнение результатов моделей и вывод

Сделаем оценку модели с использованием указанных выше метрик.

Результаты оценки моделей отображены на рисунке 16.

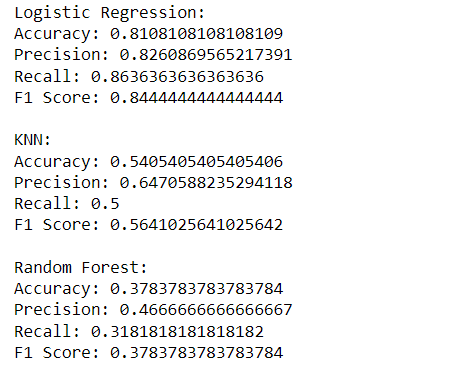


Рисунок 16 - Результаты оценки моделей

Диаграмма сравнения трех моделей с помощью метрики accuracy отображена на рисунке 17.

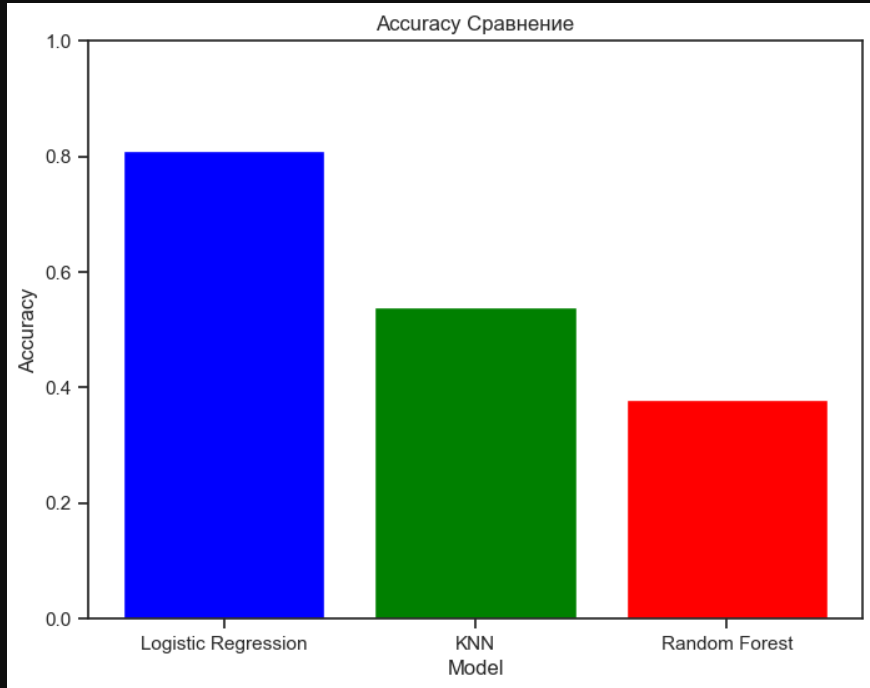


Рисунок 17 - Accuracy сравнение

Диаграмма сравнения трех моделей с помощью метрики precision отображена на рисунке 18.

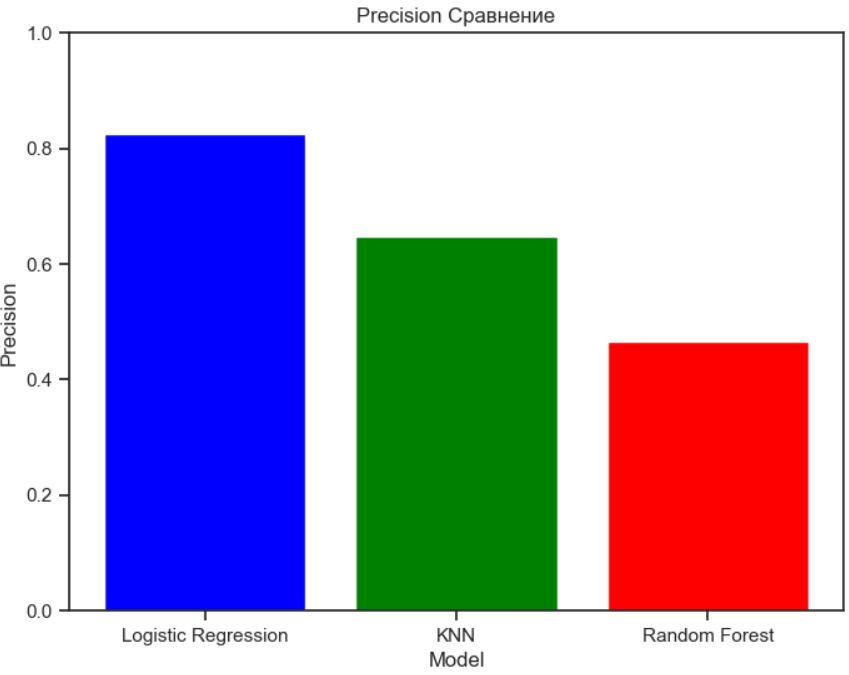


Рисунок 18 - Precision сравнение

Диаграмма сравнения трех моделей с помощью метрики recall отображена на рисунке 19.

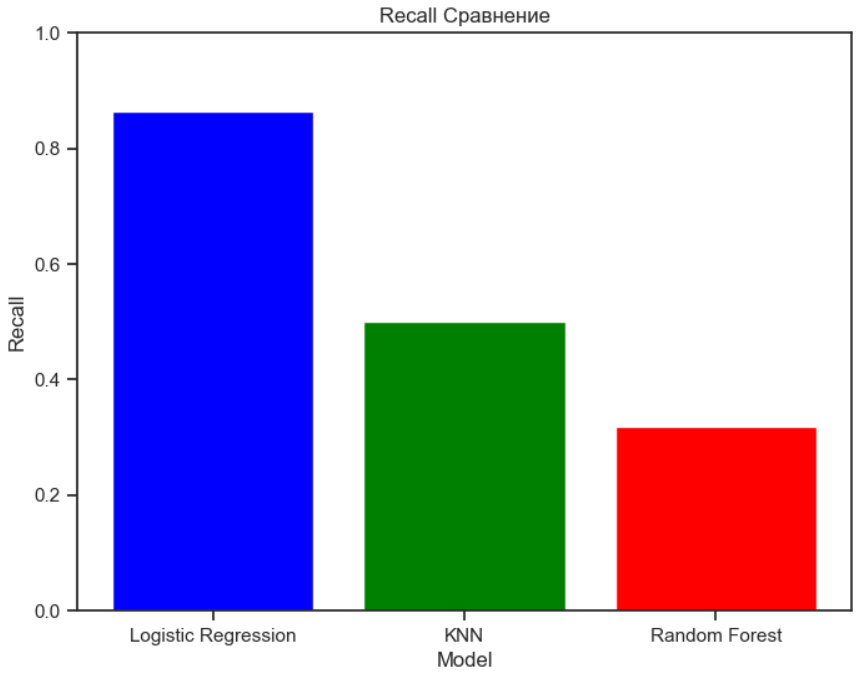


Рисунок 19 - Recall сравнение

Диаграмма сравнения трех моделей с помощью метрики f1-score отображена на рисунке 20.

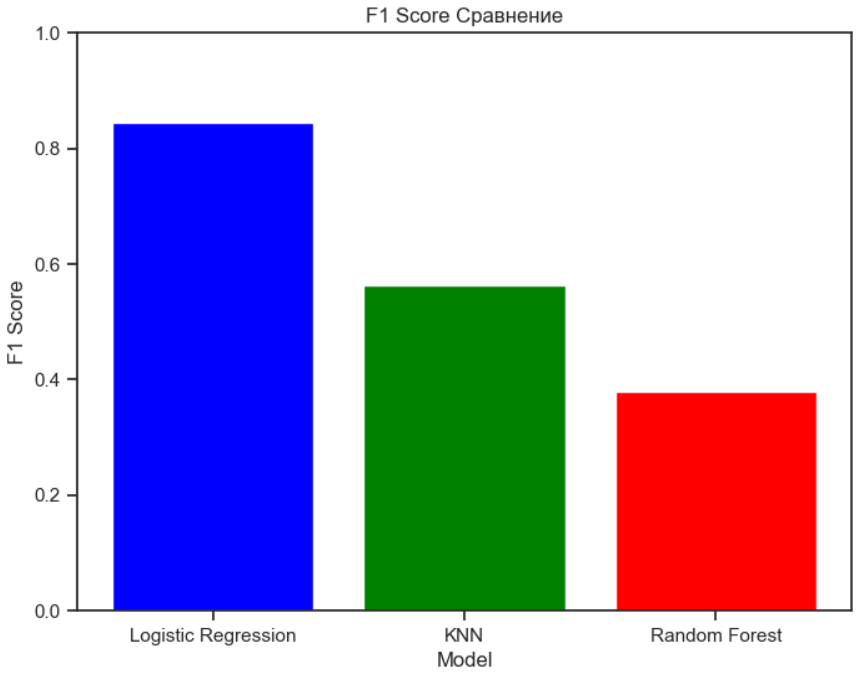


Рисунок 20 - F1 score сравнение

Из упомянутых диаграмм видно, что:

Логистическая регрессия показала наилучший результат среди всех моделей. Она имеет наивысшие показатели по всем метрикам: точность (0.8108), точность (0.8261), полнота (0.8636) и F1-мера (0.8444). Это указывает на хорошее общее качество модели, уравновешенное между правильными предсказаниями и пропущенными положительными случаями.

К-ближайших соседей (KNN) имеет средние результаты по сравнению с другими моделями. Точность (0.5405), точность (0.6471), полнота (0.5) и F1-мера (0.5641) значительно ниже, чем у логистической регрессии. Это указывает на то, что KNN менее эффективен для данной задачи.

Случайный лес показала наихудшие результаты. Точность (0.3783), точность (0.4667), полнота (0.3181) и F1-мера (0.3783) являются самыми низкими среди всех моделей. Это может указывать на проблемы с переобучением или недостаточной регулировкой параметров модели.

По результатам сравнения, логистическая регрессия является наилучшей моделью для данной задачи классификации. Она демонстрирует высокие показатели по всем ключевым метрикам, что делает ее предпочтительным выбором.

* + 1. Разработка веб-приложения

Согласно указанными задачами было создано веб-приложение для контакта пользователя с системой анализа алгоритмов машинного обучения, использующей Pandas. Для создания комфортного, эстетического, также ясного пользователю интерфейса была применена библиотека Streamlit.

При запуске веб-приложения на главной странице показывается название и описание системы «Система анализа алгоритмов машинного обучения для решения задач классификации с использованием Pandas» как демонстрируется на рисунке 21.

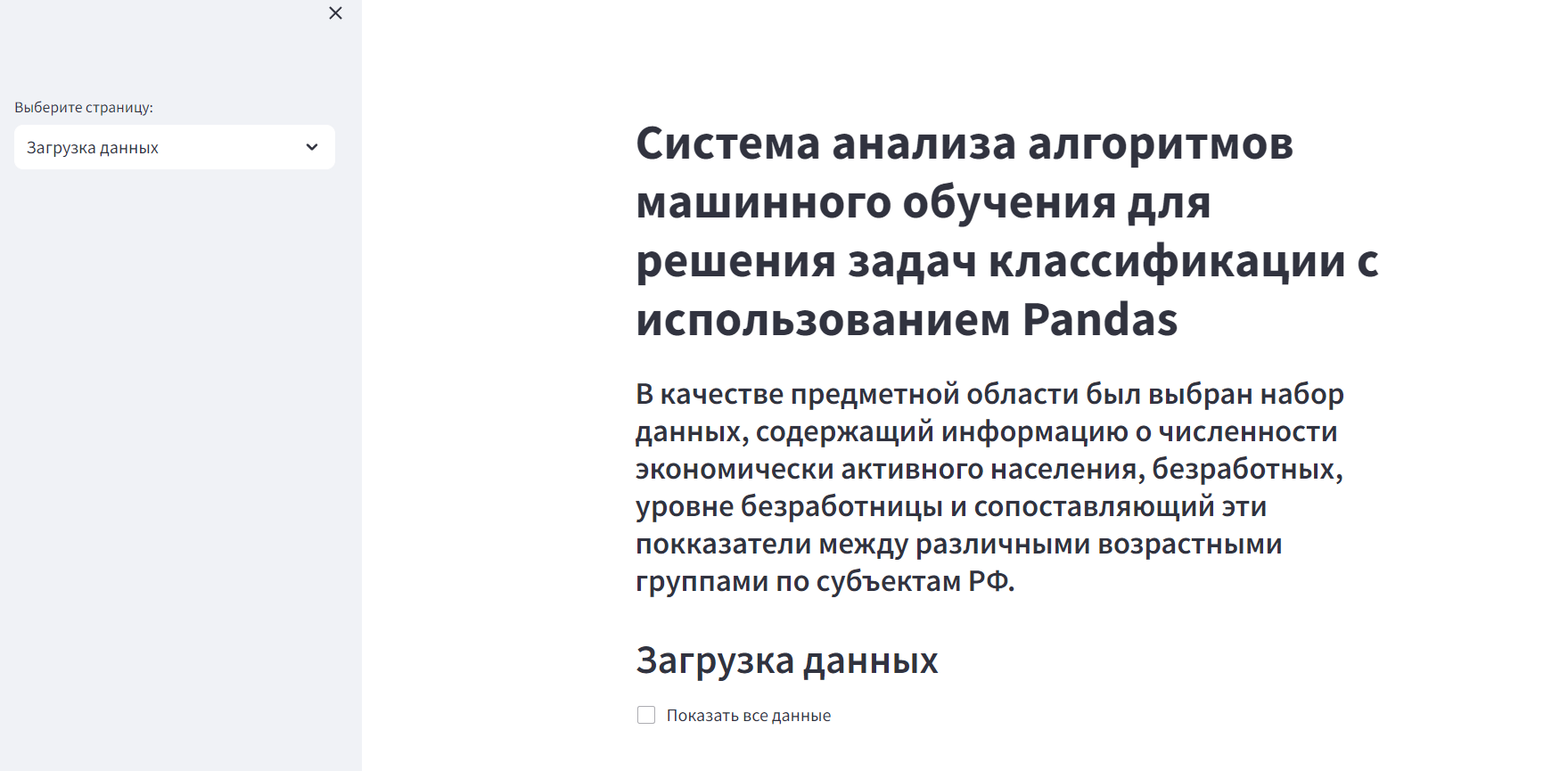


Рисунок 21 - Страница системы

Для перехода между страницами веб-приложения показывается выпадающий список «Выберите страницу» на боковой панели, что иллюстрируется на рисунке 22.

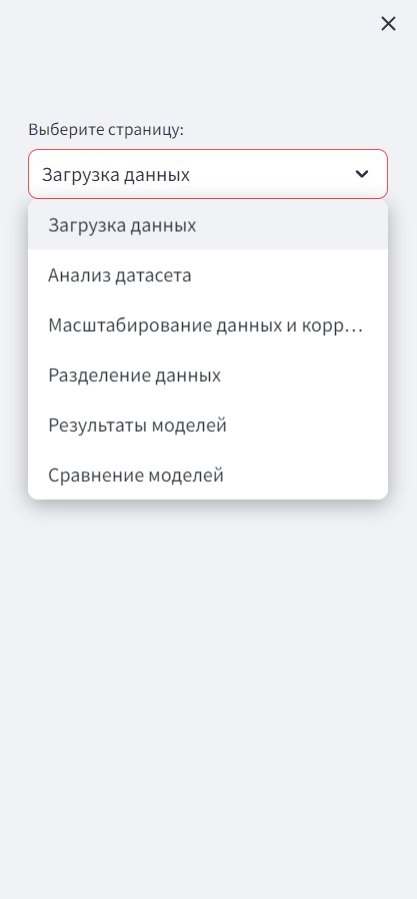


Рисунок 22 – Навигационное меню для перемещения между страницами

Страница «Загрузка данных» показана на рисунке 23. В ней отображены данные датасета.

На этой странице веб-приложения есть элемент графического интерфейса, который позволяет пользователям отмечать или снимать отметку с флажка. Этот элемент используется для управления отображения данных в приложении.

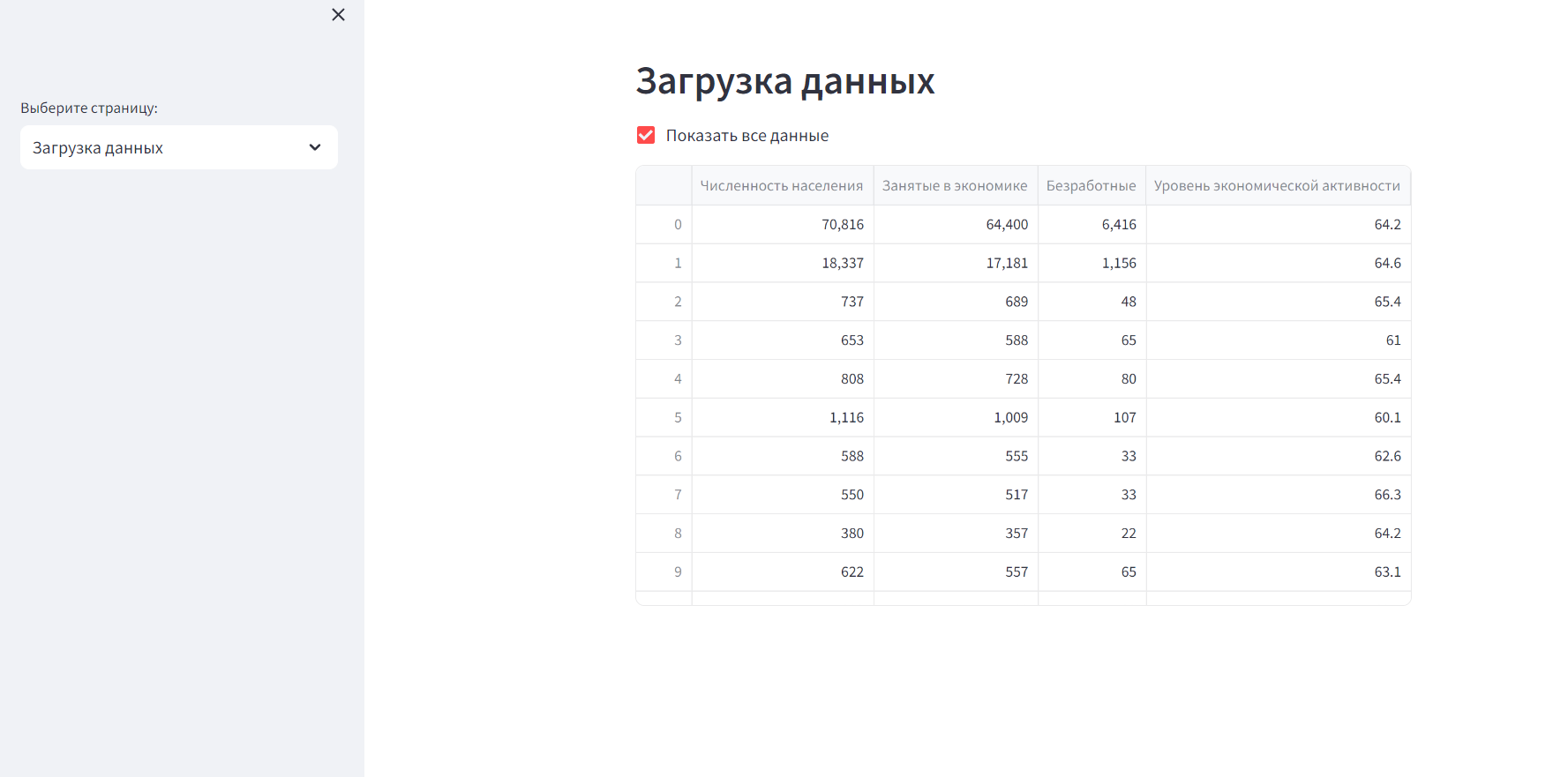


Рисунок 23 - Страница «Загрузка данных»

При выборе опции «Анализ датасета» из выпадающего списка на боковой панели показывается страница, продемонстрированный на рисунке 24.

Он содержит данные, описывающие датасет. Это отображение первых 5 значений, размер и список столбцов датасета.

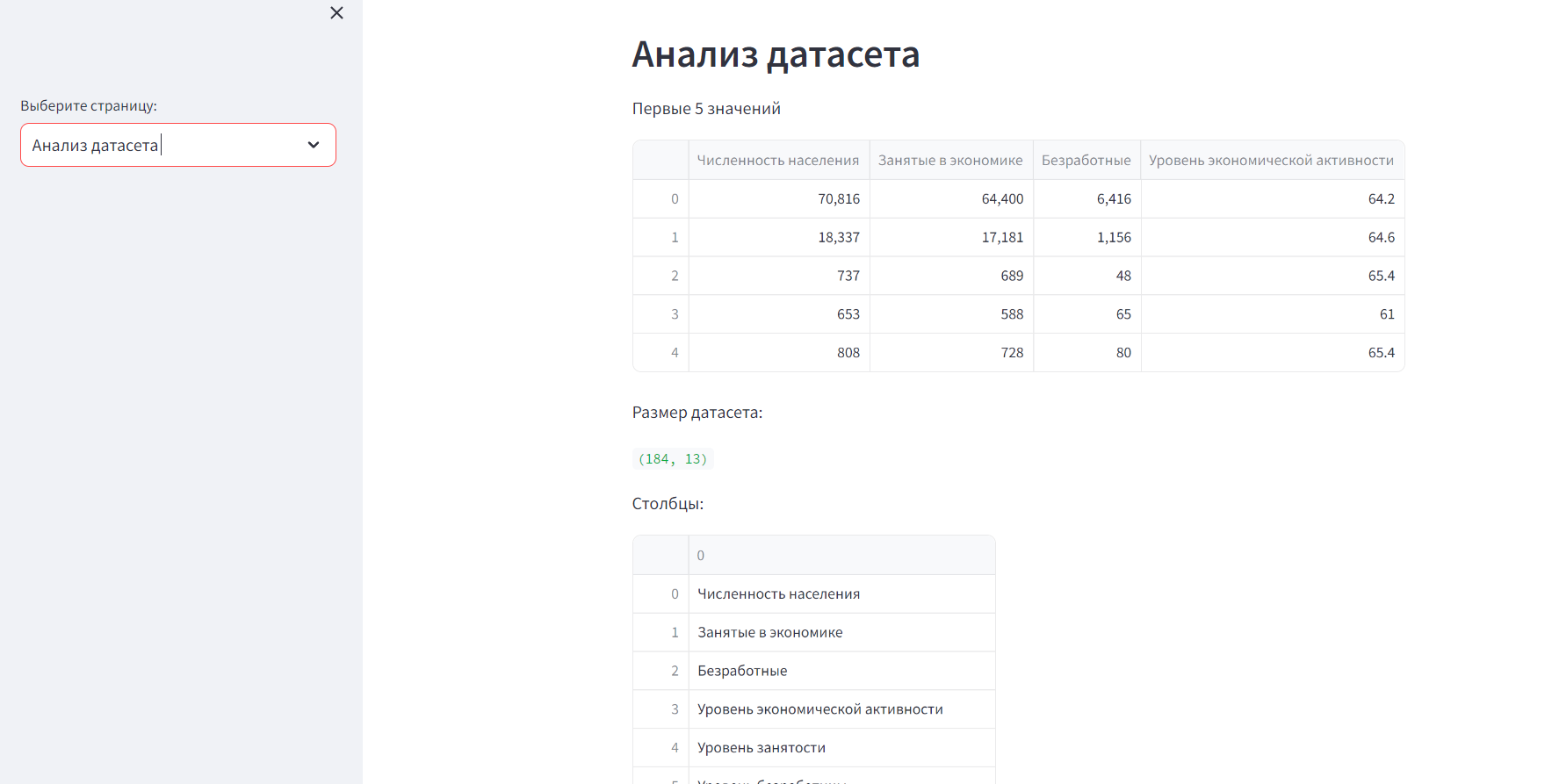


Рисунок 24 - Страница «Анализ датасета»

Переход на страницу «Масштабирование данных и корреляционный анализ» показывает интерфейс с разделами «Масштабирование данных» и «Корреляционный анализ», как на рисунке 25.

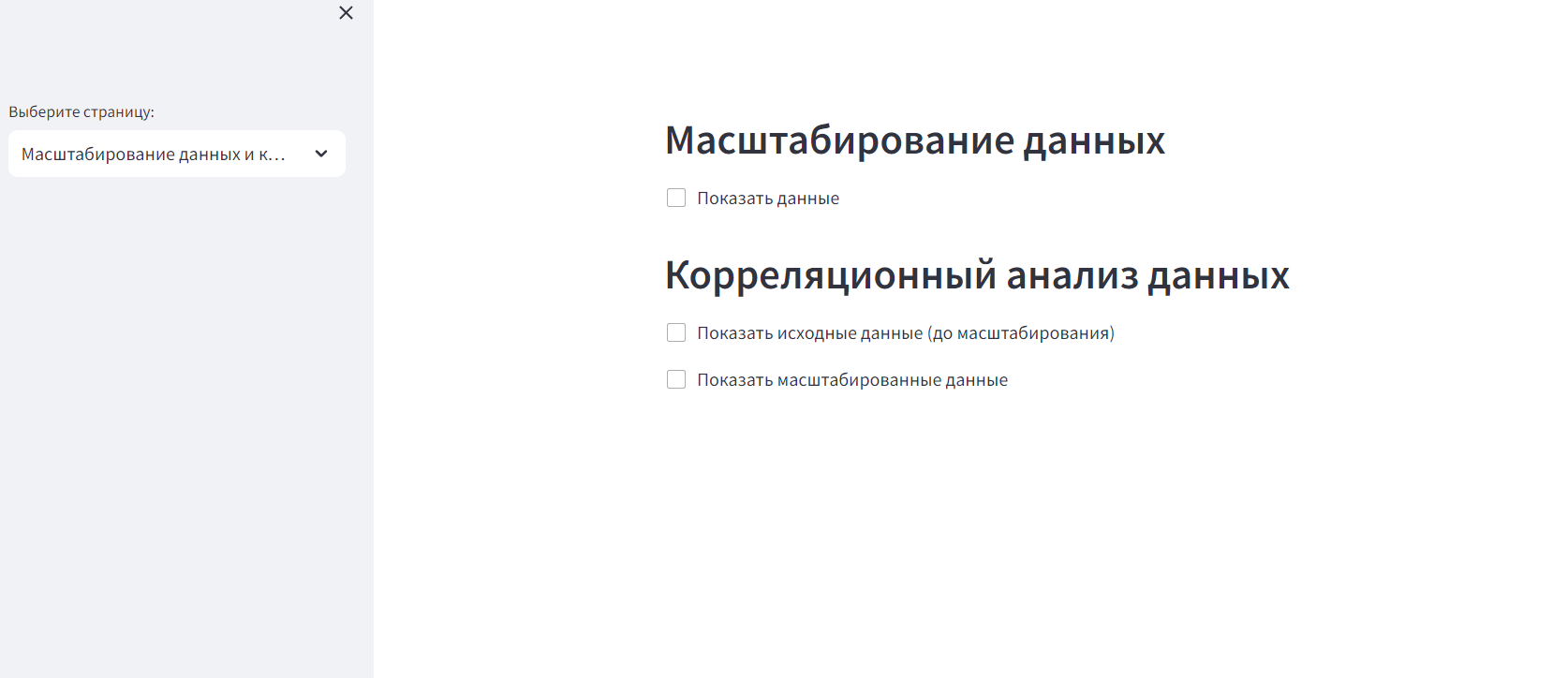


Рисунок 25 - Страница «Масштабирование и корреляция»

На этой странице есть флажки «Показать данные» в разделе «Масштабирование данных» и флажки «Показать исходные данные (до масштабирования)», «Показать масштабированные данные» в разделе «Корреляционный анализ».

При нажатии на флажок «Показать данные» в разделе «Масштабирование данных» на странице появляются графики распределения данных до и после масштабирования. Это продемонстрировано на рисунке 26.



Рисунок 26 - Раздел «Масштабирование данных»

При нажатии на флажки «Показать исходные данные (до масштабирования)» и «Показать масштабированные данные» в разделе «Корреляционный анализ» на странице появляются корреляционные матрицы до и после масштабирования данных. Это показано на рисунке 27.



Рисунок 27 - Раздел «Корреляционный анализ»

На странице «Разделение данных» человек видит интерфейс, отображающий размеры обучающей и тестовой выборок, как демонстрировано на рисунке 28.

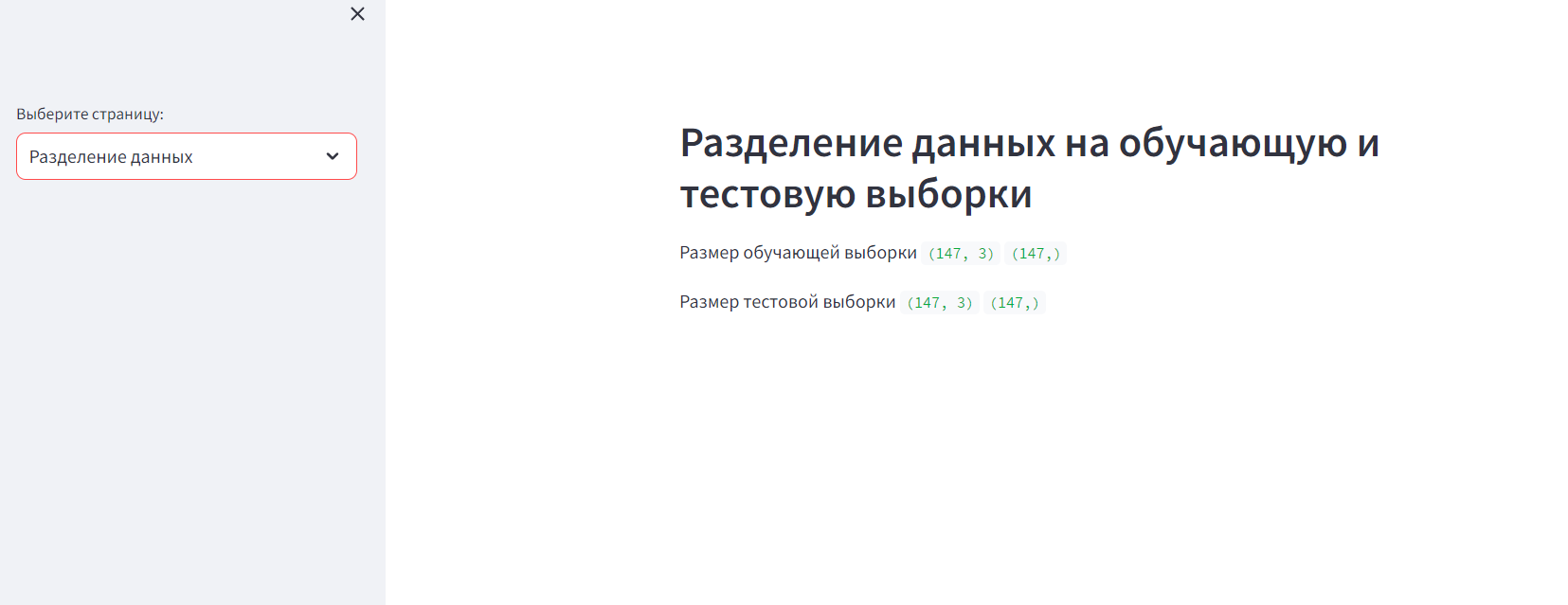


Рисунок 28 - Страница «Разделение данных»

На странице «Оценка моделей» человек видит интерфейс, показанный на рисунке 29.

На этой странице есть ещё один выпадающий список «Выберите модель для оценки», в котором отображены названия разных моделей для оценки. При нажатии на модель «Logistic Regression» в выпадающем списке справа отображаются результаты оценки этой модели. При нажатии на модель «KNN» отображаются результаты оценки KNN-модели. При нажатии на модель «Random Forest» отображаются результаты оценки этой модели.

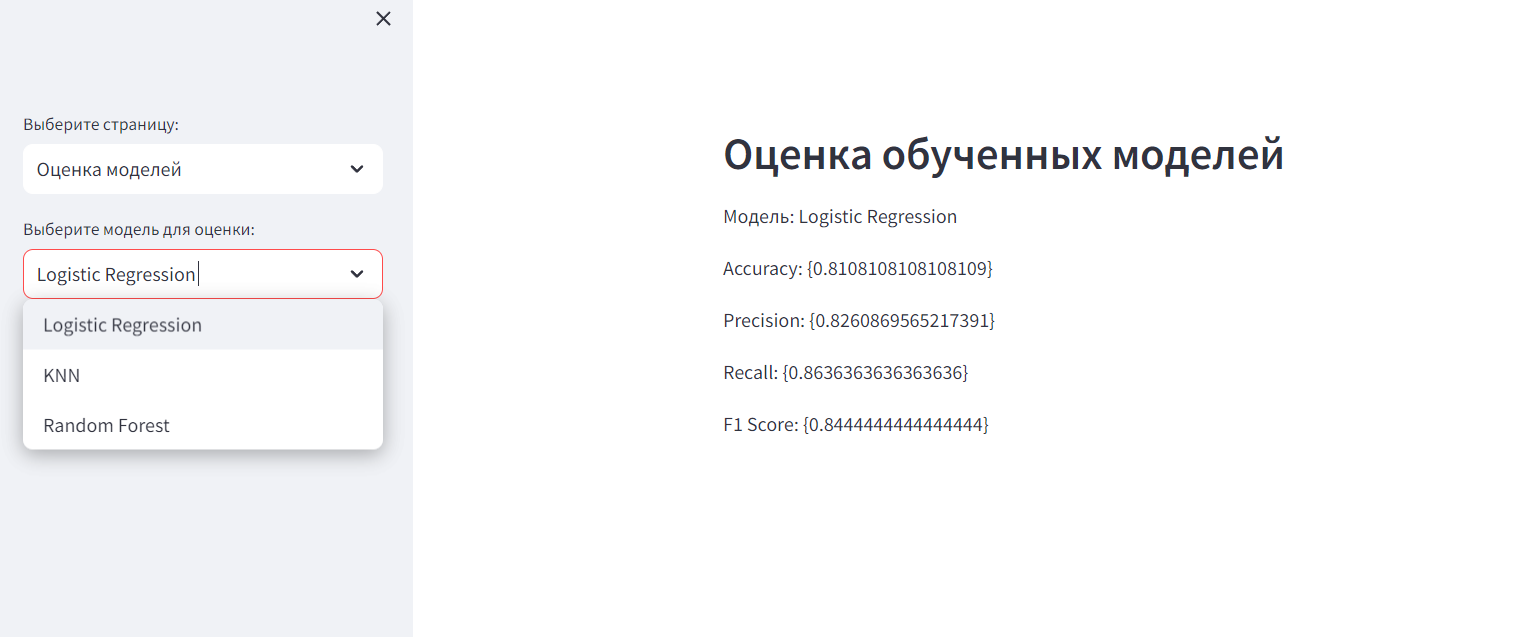


Рисунок 29 - Страница «Оценка моделей»

На странице «Сравнение моделей», человек видит интерфейс, продемонстрированный на рисунке 30.

На этой странице показываются графики, отображающие различия в оценках трех моделей.

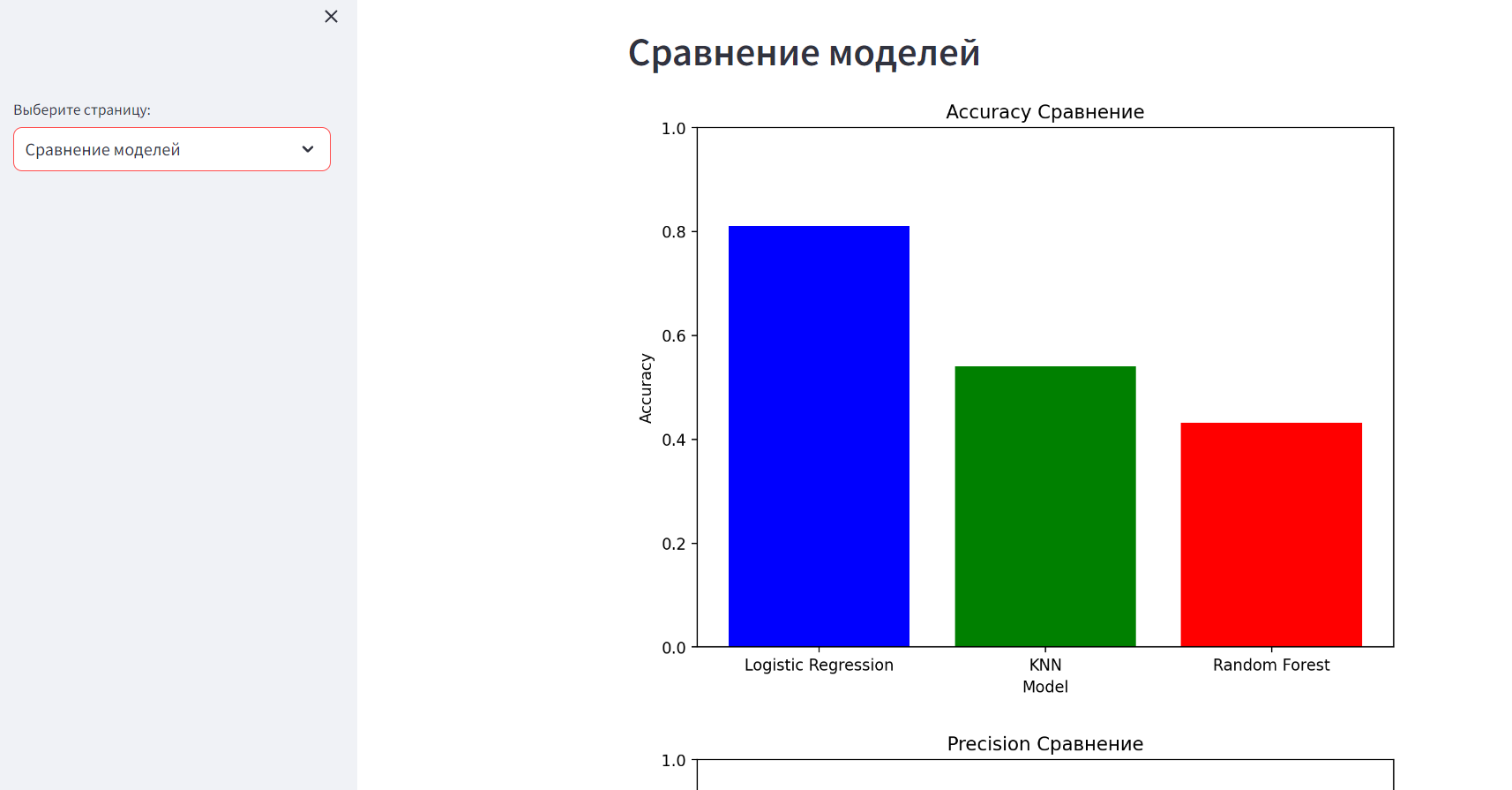


Рисунок 30 - Страница «Сравнение моделей»

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы бакалавра была разработана система, содержащая данные об анализе алгоритмов машинного обучения для решения задач классификации с использованием Pandas на примере предметной области «Статистические данные о занятости и безработице среди населения по возрастным группам».

Для этого были сделаны следующие пункты:

1. загрузка данных в DataFrame Pandas.
2. предварительную обработку данных, обработка пропущенных значений, кодирование категориальных признаков и масштабирование числовых признаков;
3. разделение данных на обучающий набор и тестовой набор данных с помощью scikit-learn;
4. выбор алгоритмов машинного обучения для классификации, таких как логическая регрессия, метод опорных векторов, случайный лес;
5. обучение моделей;
6. оценивание каждой модели на тестовом наборе;
7. выбирать модель с наилучшей производительностью на основе оценок;
8. вывод результатов анализа, включая графики и метрики для каждой модели.

При разработке системы была выполнена главная цель – предоставление пользователю результатов анализа алгоритмов машинного обучения для решения задач классификации с использованием Pandas.

Созданная система может быть усовершенствована в дальнейшем. Может быть улучшены модели машинного обучения и интерфейс веб-приложения системы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Безработица в Европейском Союзе [Электронный ресурс] // kaggle.com URL.: <https://www.kaggle.com/datasets/gpreda/unemployment-in-european-union/data> (дата обращения 25.02.2024).
2. Безработица в Америке [Электронный ресурс] // kaggle.com URL.: <https://www.kaggle.com/datasets/justin2028/unemployment-in-america-per-us-state> (дата обращения 25.02.2024).
3. Статистические данные о занятости и безработице среди населения по возрастным группам [Электронный ресурс] // data.rcsi.science URL.: <https://data.rcsi.science/data-catalog/datasets/156/> (дата обращения: 25.02.2024).
4. Supervised and Unsupervised learning [Электронный ресурс] // geeksforgeeks.org URL.: <https://www.geeksforgeeks.org/supervised-unsupervised-learning/> (дата обращения 10.04.2024).
5. Getting started with Classification [Электронный ресурс] // geeksforgeeks.org URL.: <https://www.geeksforgeeks.org/getting-started-with-classification/> (дата обращения 10.04.2024).
6. Classification Algorithm in Machine Learning [Электронный ресурс] // javatpoint.com URL.: <https://www.javatpoint.com/classification-algorithm-in-machine-learning> (дата обращения: 15.04.2024).
7. Basic metrics for classification tasks in machine learning [Электронный ресурс] // webiomed.ai URL.: <https://webiomed.ai/blog/osnovnye-metriki-zadach-klassifikatsii-v-mashinnom-obuchenii/> (дата обращения: 04.05.2024).
8. Код. Что такое jupyter-ноутбук и зачем он нужен. [Электронный ресурс] – URL.: <https://thecode.media/jupyter/> (Дата обращения – 04.05.2024).
9. Пасхавер Борис Pandas в действии. – СПб.: Питер, 2023. – 512 с.
10. Мюллер А., Гвидо С. Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными. / М.: Вильямс, 2017. – 480 с.
11. Официальный сайт Seaborn [Электронный ресурс] // seaborn.pydata.org URL.: <https://seaborn.pydata.org/> (дата обращения: 20.05.2024)
12. Документация Scikit-learn [Электронный ресурс] // scikit-learn.org URL.: [https://scikit-learn.ru/getting\_started/](https://scikit-learn.ru/getting_started/%20) (дата обращения: 20.05.2024).
13. Документация Streamlit [Электронный ресурс] // docs.streamlit.io URL.: <https://docs.streamlit.io/> (дата обращения 20.05.2024)
14. Репозиторий курса «Машинное обучение» бакалавриат, 6 семестр [Электронный ресурс] // github.com URL.: <https://github.com/ugapanyuk/courses_current/wiki/COURSE_TMO_SPRING_2023/> (дата обращения: 12.03.2024).
15. Мухамедиев Р.И., Амиргалиев Е.Н. Введение в машинное обучение. Учебник. – М.: УМО РУМС, 2022. – 252 с.
16. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных /. – М.: ДМК Пресс, 2015. – 400 с.
17. Beasley D. Python. A detailed guide. – Пер. с англ. – СПб.: Символ-Плюс, 2018.- 864 с.

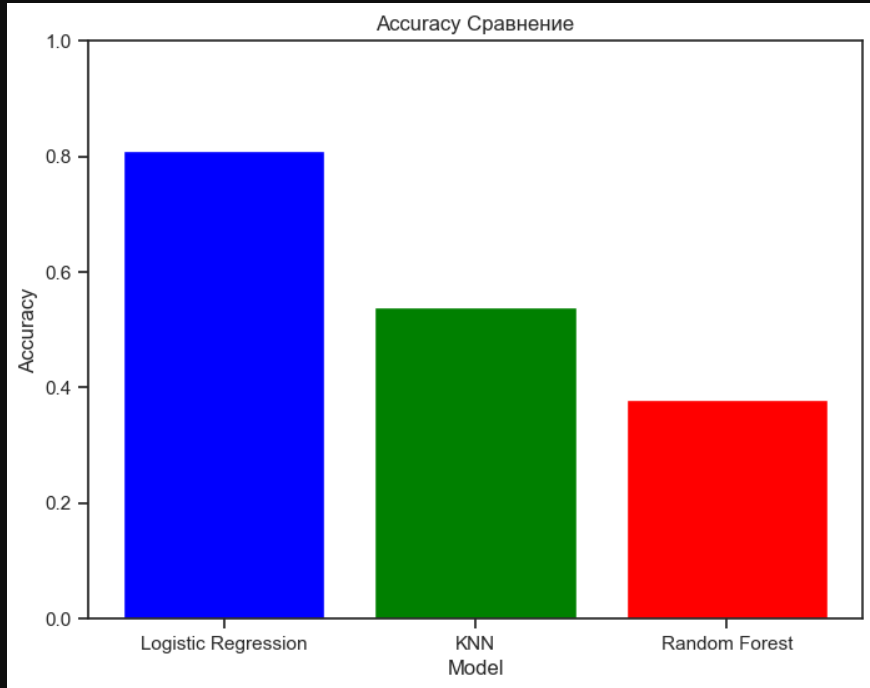
ПРИЛОЖЕНИЕ A ГРАФИЧЕСКАЯ ЧАСТЬ

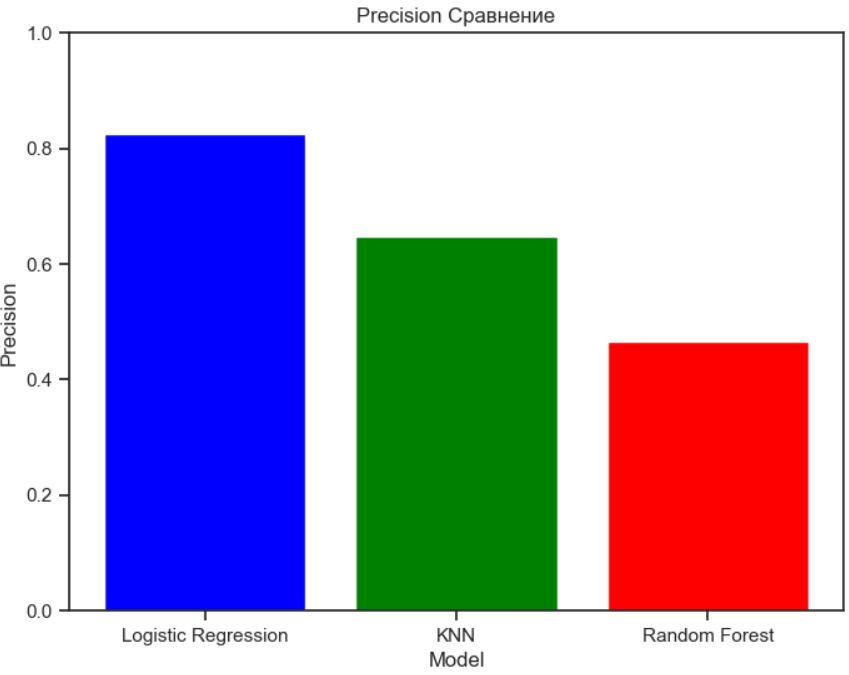
В графическую часть выпускной квалификационной работы входят:

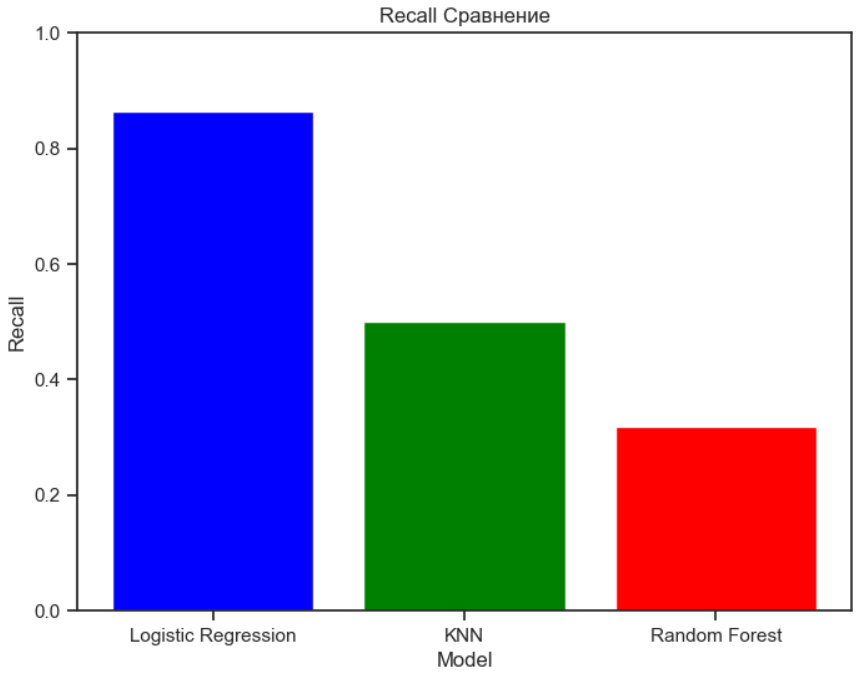
А.1. Диаграмма сравнения моделей

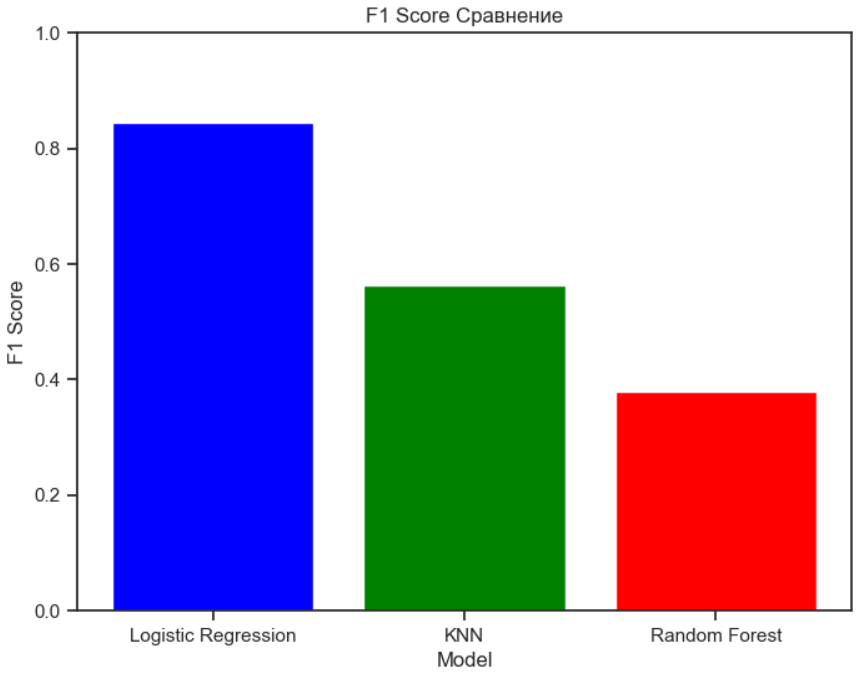
А.2. Экранные формы

A.1. Диаграммы сравнения оценки моделей

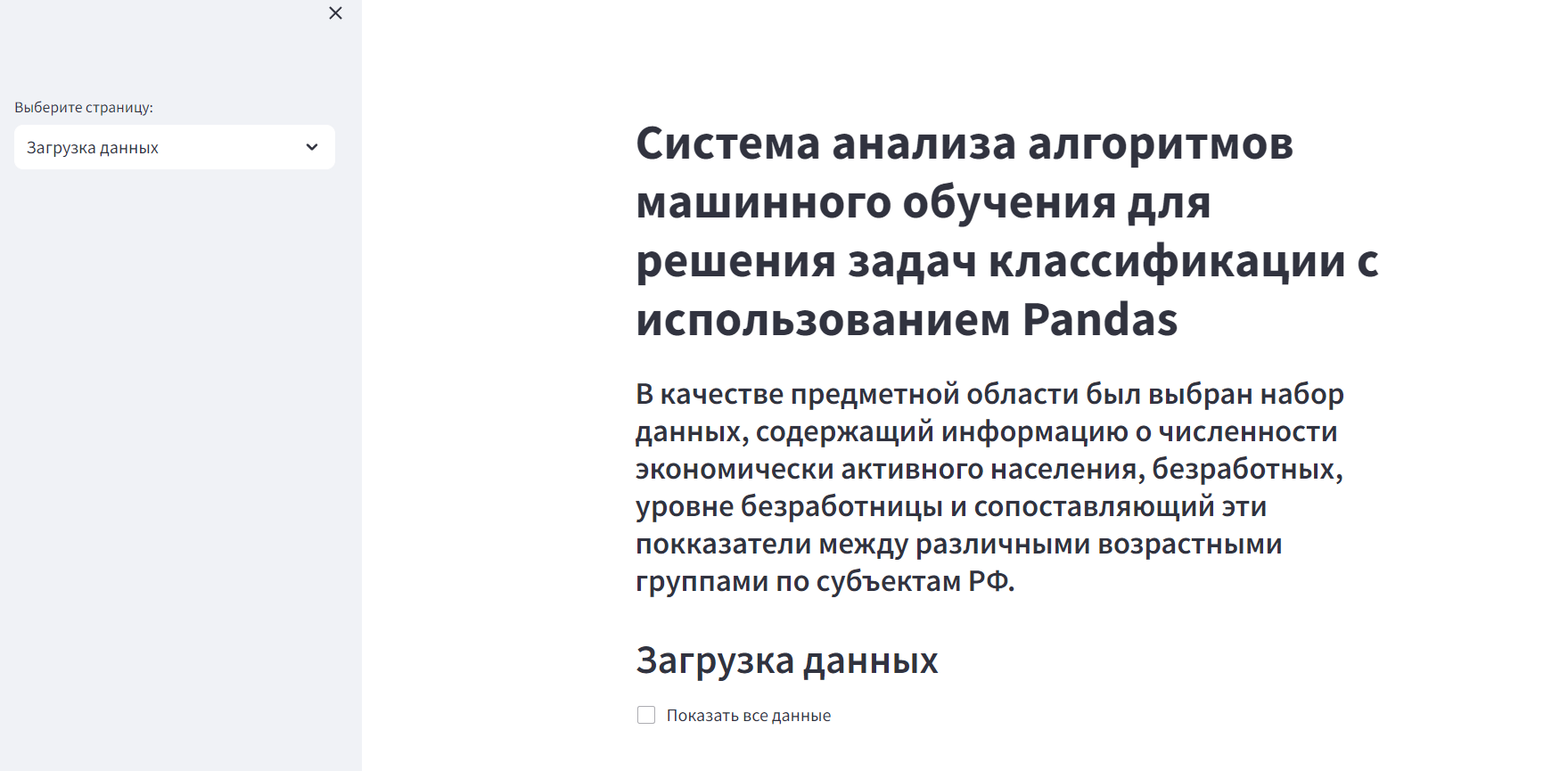


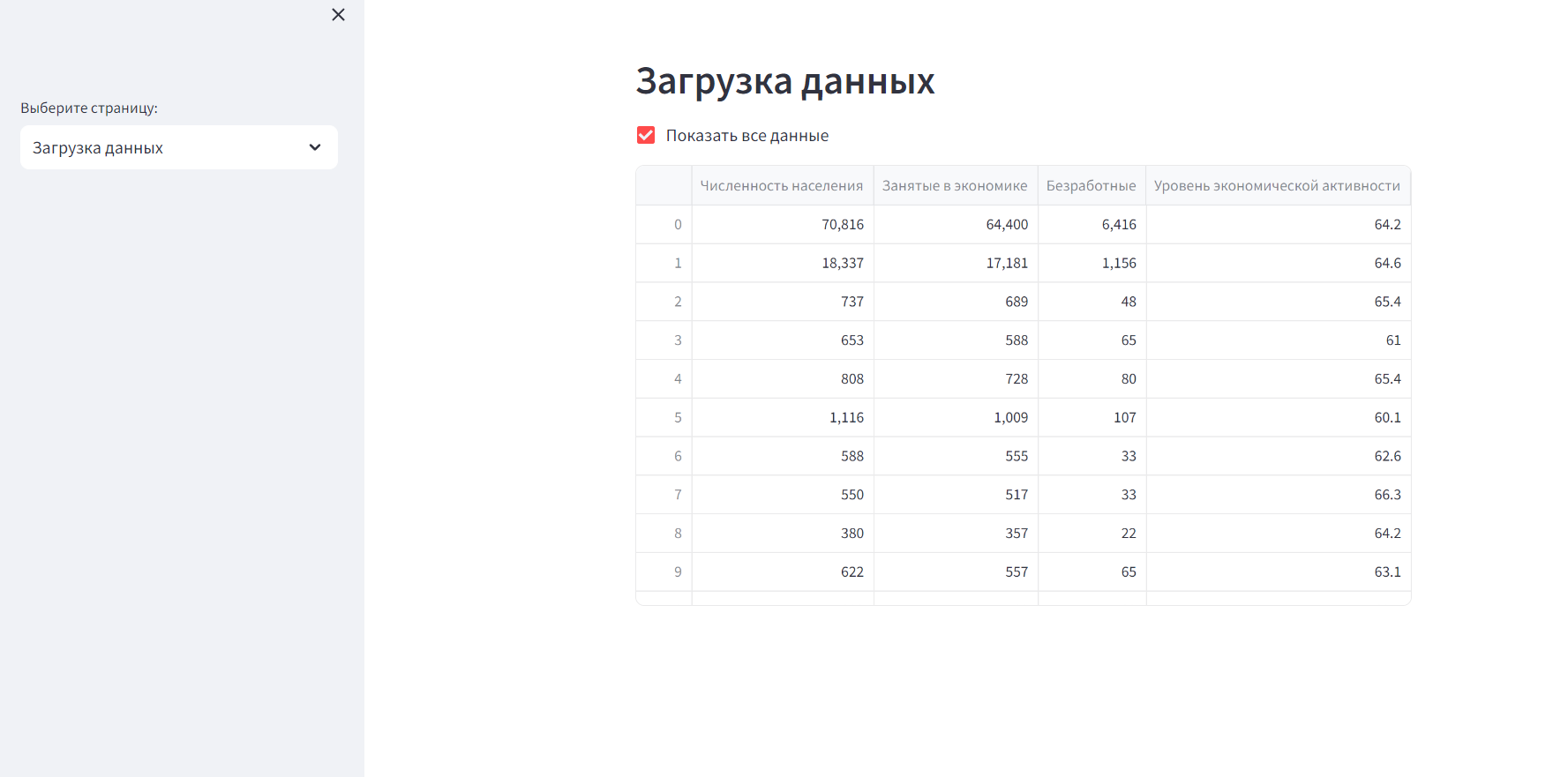


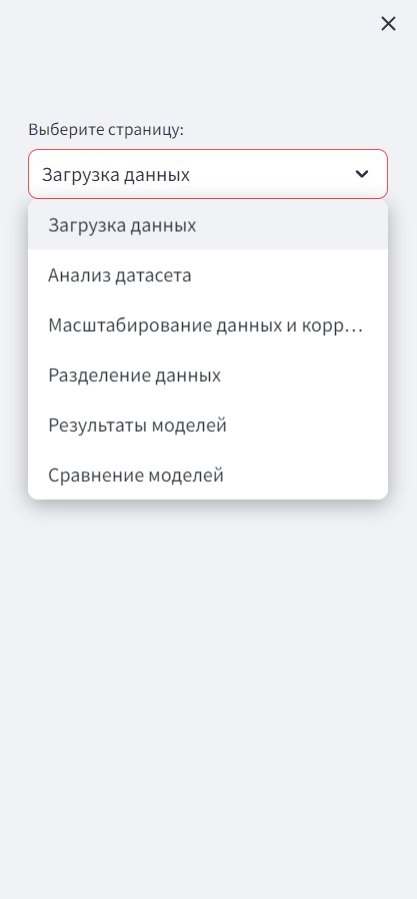


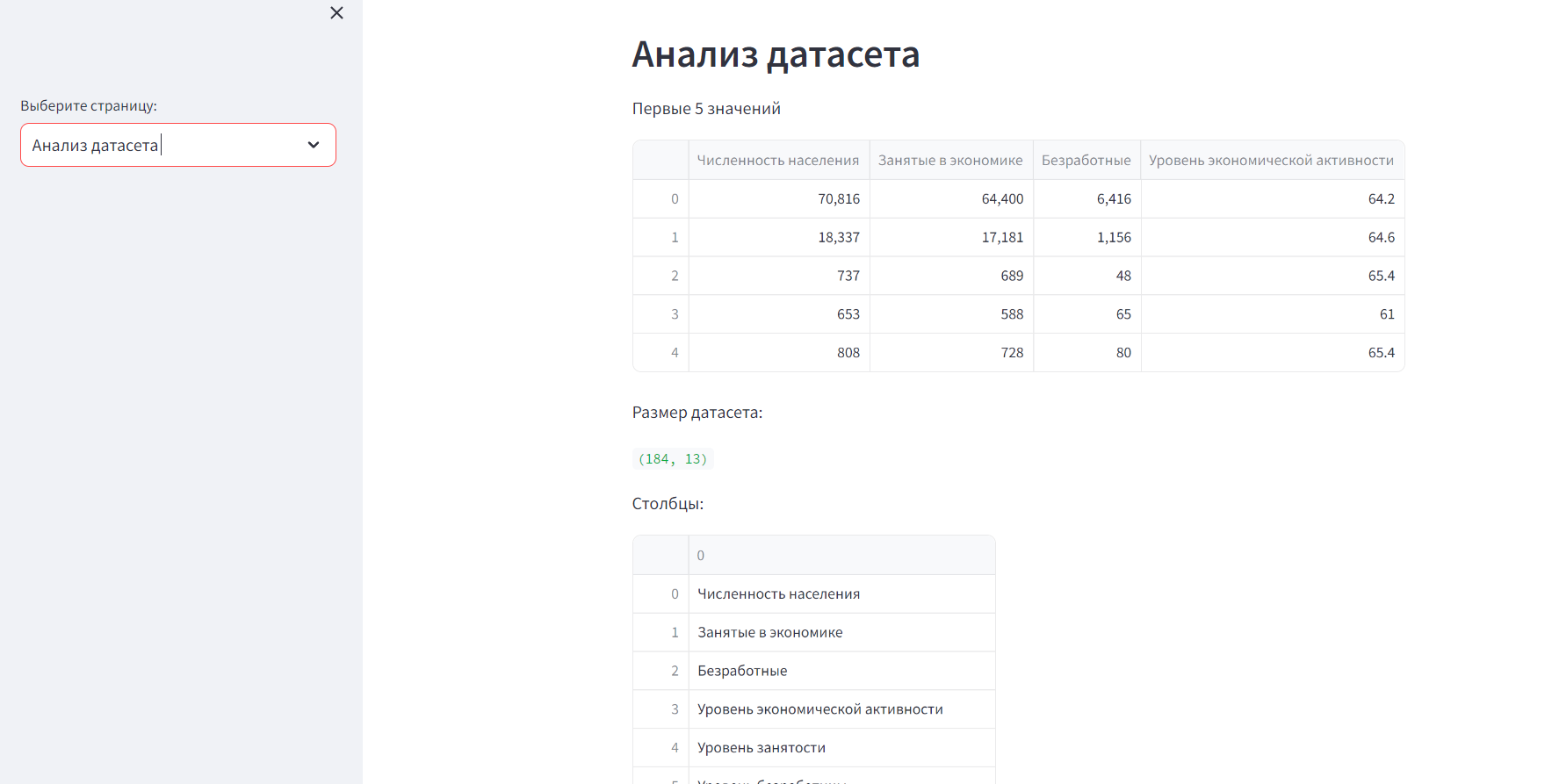


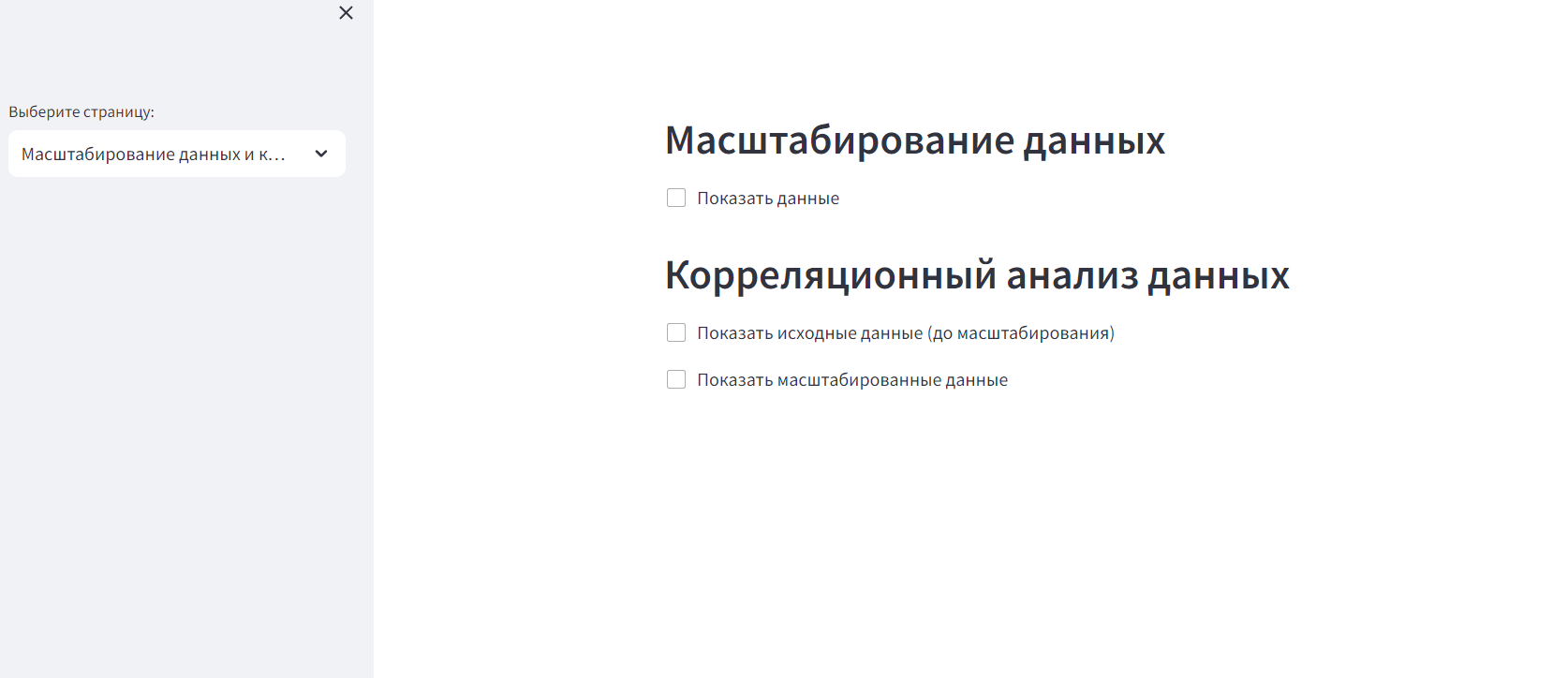
A.2. Экранные формы





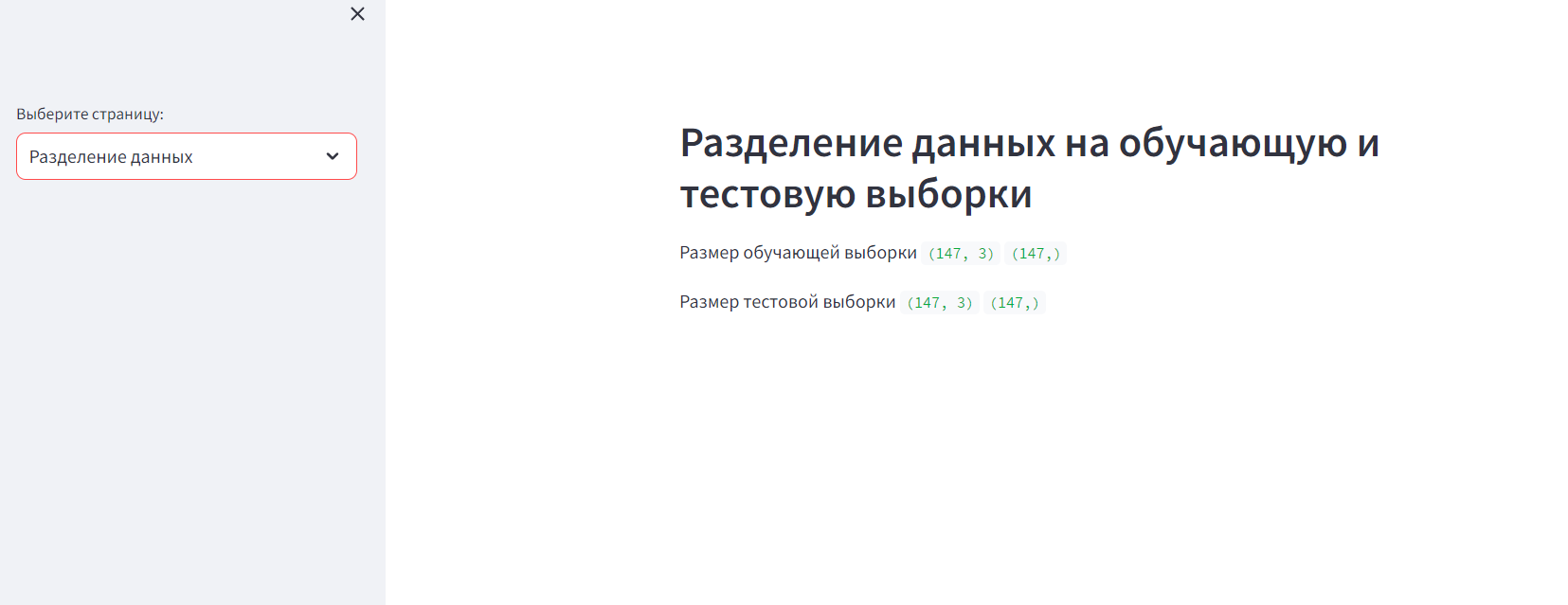


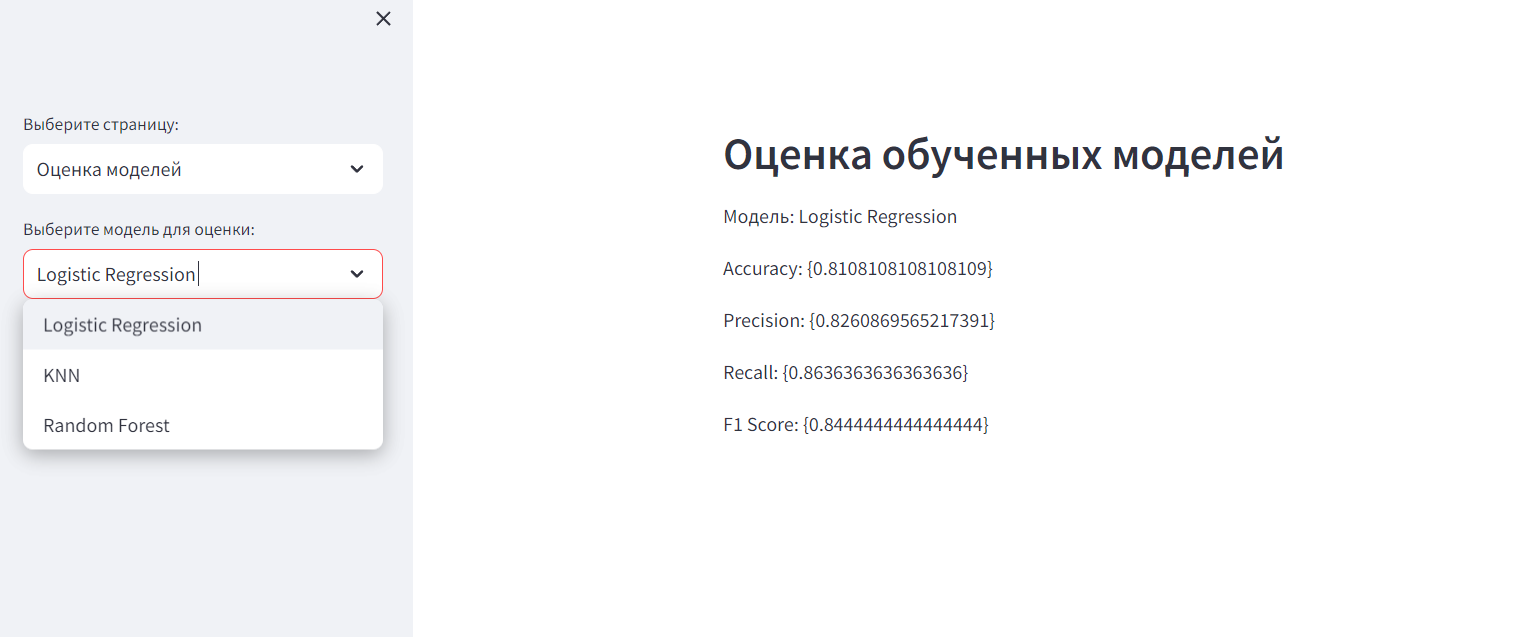


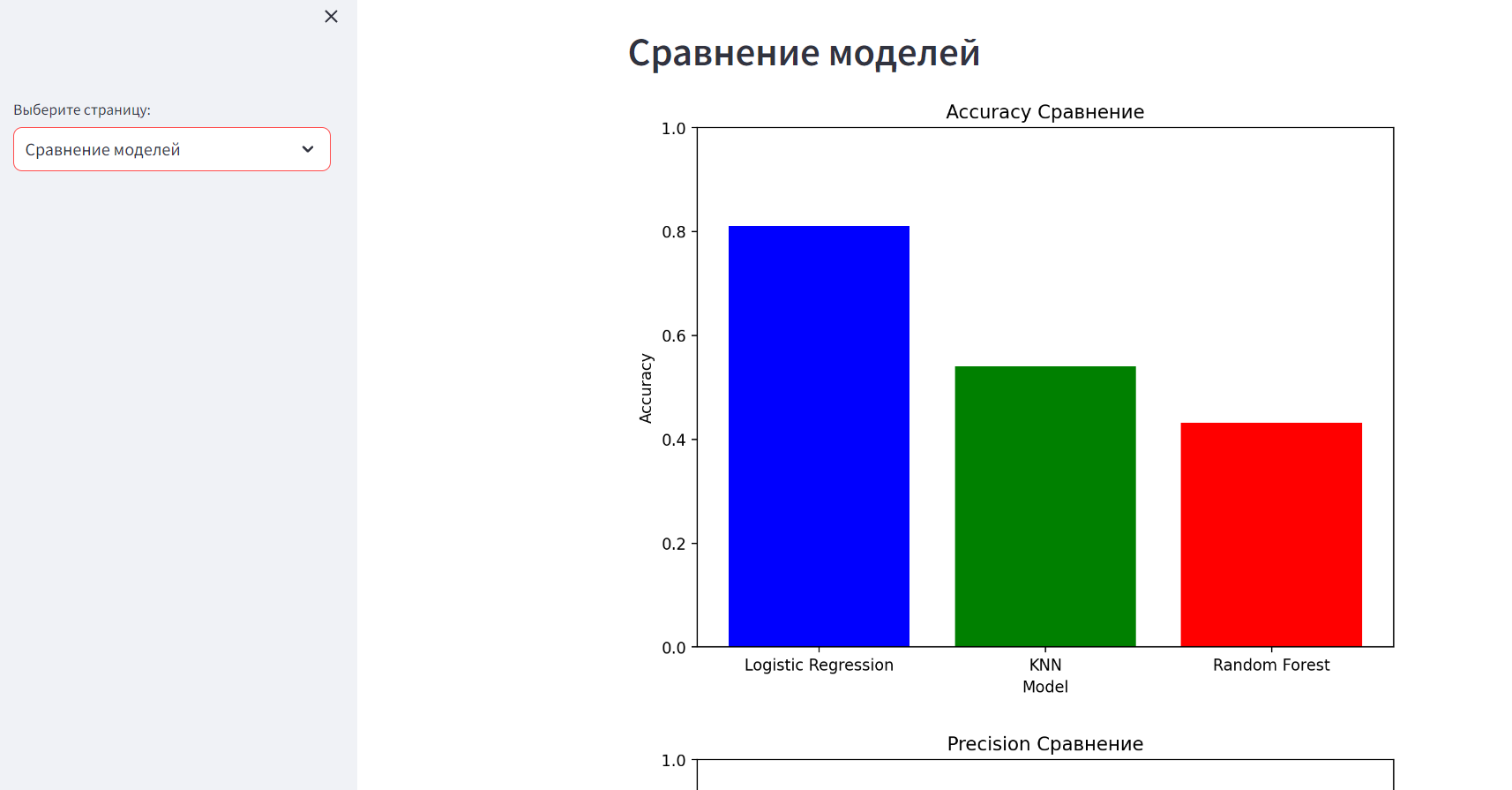












ПРИЛОЖЕНИЕ B ТЕХНИЧЕСКОЕ ЗАДАНИЕ

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Н.Э. Баумана

Кафедра «Системы обработки информации и управления»

|  |  |  |
| --- | --- | --- |
| Утверждаю  Заведующий кафедрой ИУ-5 |  | Согласовано  Научный руководитель |
| \_\_\_\_\_\_\_\_\_ В.И.Терехов  "\_\_"\_\_\_\_\_\_\_\_\_\_2024 г. |  | Ю.А. Григорьев "\_\_"\_\_\_\_\_\_\_\_\_\_2024 г. |

**Система анализа алгоритмов машинного обучения для решения задач классификации с использованием Pandas**

Техническое задание

(вид документа)

писчая бумага

(вид носителя)

6

(количество листов)

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| Соловьева Александра Михайловна |  |
| "\_\_"\_\_\_\_\_\_\_\_\_ 2024 г. |  |

Москва - 2024

**1. Наименование**

Анализ алгоритмов машинного обучения для решения задач классификации с использованием Pandas на примере предметной области “Статистические данные о занятости и безработице среди населения по возрастным группам”.

**2. Основания для разработки**

Основанием для разработки является задание на выпускную квалификационную работу, подписанное руководителем выпускной работы и утверждённое заведующим кафедрой ИУ5 МГТУ им. Н.Э. Баумана 12 декабря 2023 года.

**3. Назначение разработки**

Разрабатываемая система предназначена для пользователей, желающих получить данные об анализе алгоритмов машинного обучения для решения задач классификации с использованием Pandas на примере предметной области “Статистические данные о занятости и безработице среди населения по возрастным группам”.

Целью является предоставление пользователю результатов анализа алгоритмов машинного обучения для решения задач классификации с использованием Pandas.

**4. Требования к программе или программному изделию**

4.1 Требования к функциональным характеристикам:

4.1.1 Требования к составу выполняемых функций

Система должна выполнять следующие функции:

4.1.1.1 импорт библиотек

4.1.1.2 загрузка данных в DataFrame Pandas

4.1.1.3 показ основных характеристик датасета, удаление лишних столбцов

4.1.1.4 замена названия колонок, обработка пропущенных значений, проверка дубликатов и замена типов данных

4.1.1.5 кодирование категориальных признаков, масштабирование данных, и корреляционный анализ данных;

4.1.1.6 разделение данных на обучающий набор и тестовой набор;

4.1.1.7 выбор алгоритмов машинного обучения для решения задачи классификации

4.1.1.8 обучение модели

4.1.1.9 оценивать качество каждой модели

4.1.1.10 выбор модели с наилучшей производительностью

4.1.1.11 вывод результатов анализа.

4.1.2 Требования к входным данным

Входными данными для анализа алгоритмов машинного обучения для решения задач классификации с использованием Pandas должен быть датасет, записанный в файл формата .csv, кодировка UTF-8.

4.1.3 Требования к выходным данным

Выходными данными должны быть данные, показывающие результаты анализа алгоритмов машинного обучения для решения задач классификации с использованием Pandas на примере предметной области “Статистические данные о занятости и безработице среди населения по возрастным группам”.

4.2 Требования к надёжности

Система должна надежно и устойчиво функционировать, при вводе некорректных данных выдавать сообщение об ошибке.

4.3 Условия эксплуатации

Пользователь должен быть знаком с основами работы в среде Jupyter Notebook.

4.4 Требования к составу и параметрам технических средств

Компьютер с процессором AMD ryzen 5 c частотой 3.4 ГГц и выше, и ОП не менее 4 ГБ.

4.5 Требования к информационной и программной совместимости

Операционная система Windows 11, приложение Anaconda3, приложение Jupyter Notebook, предустановленные библиотеки в Python: pandas, numpy, seaborn, matplotlib, scikit-learn.

**5. Требования к программной документации**

Для представления заказчику разрабатываются следующие документы:

1. Техническое задание.

2. Программа и методика испытаний.

3. Расчётно-пояснительная записка.

4. Графический материал по работе в формате презентации.

**6. Технико-экономические показатели**

Требования к данному разделу не предъявляются.

**7. Этапы разработки**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Наименование этапов выпускной квалификационной работы** | **Сроки выполнения этапов** | | **Отметка о выполнении** | |
| **план** | **факт** | **Должность** | **подпись, ФИО** |
|  | Формулирование проблемы, цели и задач работы | *15.01.2024*  *Планируемая*  *дата* |  | Руководитель ВКР | Григорьев Ю.А. |
|  | Подготовка рабочих материалов | *15.01.2024*  *Планируемая*  *дата* |  | Руководитель ВКР | Григорьев Ю.А. |
|  | Разработка программы и методики испытания | *21.03.2024*  *Планируемая дата* |  | Руководитель ВКР | Григорьев Ю.А. |
|  | Разработка второй части РПЗ «Исследовательская часть» | *10.04.2024*  *Планируемая*  *дата* |  | Руководитель ВКР | Григорьев Ю.А. |
|  | Защита макета программы  (АСОиУ) | *12.05.2024*  *Планируемая дата* |  | Нормоконтролер | Кротов Ю.Н. |

**8. Порядок контроля и приёмки**

Приём и контроль программного изделия осуществляется в соответствие с документом «Программа и методика испытаний».

ПРИЛОЖЕНИЕ C ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Н.Э. Баумана

Кафедра «Системы обработки информации и управления»

|  |  |  |
| --- | --- | --- |
| Утверждаю  Заведующий кафедрой ИУ-5 |  | Согласовано  научный руководитель |
| \_\_\_\_\_\_\_\_\_ В.И.Терехов  "\_\_"\_\_\_\_\_\_\_\_\_\_2024 г. |  | Ю.А. Григорьев "\_\_"\_\_\_\_\_\_\_\_\_\_2024 г. |

**Система анализа алгоритмов машинного обучения для решения задач классификации с использованием Pandas**

Программа и методика испытаний

(вид документа)

писчая бумага

(вид носителя)

5

(количество листов)

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| Соловьева Александра Михайловна |  |
| "\_\_"\_\_\_\_\_\_\_\_\_ 2024 г. |  |

Москва - 2024

Аннотация

В данном документе описываются последовательность и методы проведения испытаний при тестировании программного изделия, состав и структура технических и программных средств, необходимых для проведения испытаний, а также приводятся требования к предъявляемой документации, характеристикам программы применительно к условиям эксплуатации и требования к информационной и программной совместимости. Описывается тестовый пример и реакция системы на него.

1. Объект испытаний

Объектом испытания является система анализа алгоритмов машинного обучения для решения задач классификации с использованием Pandas на примере предметной области “Статические данные о занятости и безработице среди населения по возрастным группам” – далее Система.

1. Цель испытаний

Цель испытания состоит в проверке работоспособности Системы и соответствия выполняемых функций требованиям документа «Техническое задание».

1. Состав предъявляемой документации

На испытания программного продукта предъявляются следующие документы:

1. Техническое задание.

2. Программа и методика испытаний.

1. Технические требования

* 1. **Требования к программной документации**

Комплектность программной документации должна удовлетворять разделу данного документа "Состав предъявляемой документации".

* 1. **Требования к техническим характеристикам**
     1. **Требования к составу аппаратного обеспечения**

Система должна выполняться на компьютере со следующими характеристиками:

* Компьютер с процессором ADM ryzen 5 с частотой 3.4 ГГц и выше, и ОП не менее 4 ГБ;
* Наличие доступа к сети Интернет.
  + 1. **Требование к составу программного обеспечения**

Операционная система Windows 11, приложение Anaconda3, приложение Jupyter Notebook, предустановленные библиотеки в Python: pandas, numpy, seaborn, matplotlib, scikit-learn.

1. Порядок проведения испытаний

Испытания данного программного продукта будут проводиться в следующем порядке:

1) Запуск системы.

2) Тестирование функционала системы.

1. **Методы** испытаний

|  |  |  |  |
| --- | --- | --- | --- |
| № | Действие | Результат | № п. ТЗ |
| 1. | Запустить Jupyter Notebook и открыть файл ВКРБ.ipynb | Система запущена |  |
| 2. | Импортировать библиотеки | Библиотеки импортировались в Систему | 4.1.1.1 |
| 3. | Загрузить данные в DataFrame Pandas | Данные из датасета загрузились в систему | 4.1.1.2 |
| 4. | Показать основные характеристики датасета, удалить лишние столбцы | Показаны основные характеристики датасета, выполнено удаление лишних столбцов. | 4.1.1.3 |
| 5. | Заменить названия колонок, обработать пропущенных значения, проверить дубликаты, заменить типы данных | Изменились названия колонок, обработаны пропущенные значения, проверили дубликаты и заменили типы данных | 4.1.1.4 |
| 6. | Кодировать категориальные признаки, масштабировать данные и выполнить корреляционный анализ данных | Выполнено кодирование категориальных признаков, масштабирование данных и сделан корреляционный анализ данных | 4.1.1.5 |
| 7. | Разделить данные на обучающий и тестовой набор | Разделили данные на обучающий и тестовой набор | 4.1.1.6 |
| 8. | Выбрать алгоритмы машинного обучения для решения задачи классификации | Выбраны алгоритмы машинного обучения | 4.1.1.7 |
| 9. | Обучить модели | Обучили модели | 4.1.1.8 |
| 10. | Оценивать качество каждой модели | Показана оценка каждой модели с использованием метрики качества | 4.1.1.9 |
| 11. | Выбрать модель с наилучшей производительностью | Показана модель с наилучшей производительностью | 4.1.1.10 |
| 12. | Вывести результат анализа | Показан результат анализа в виде графика | 4.1.1.11 |

1. **Результат** испытаний

Основой испытаний является демонстрация работы системы анализа алгоритмов машинного обучения.

Испытание считается пройденным успешно, если в процессе демонстрации все действия прошли успешно и результат соответствовал ожидаемому с учетом используемых данных.