



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 с использованием Pandas**

Студент ИУ5Ц-1036  
(Группа)

А.М.Соловьева  
(Подпись, дата) (И.О.Фамилия)

Руководитель ВКР

Ю.А.Григорьев  
(Подпись, дата) (И.О.Фамилия)

Консультант

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 52 страницы. С приложениями объем составляет 53 страниц. Работа включает в себя 11 таблицы и 30 иллюстраций. В процессе выполнения было использовано 18 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 предметной области, рассмотрены наиболее применимые к выбранной задаче методы и модели, определены метрики оценки качества реализуемой системы, а также реализован программный продукт в виде приложения.

Пояснительная записка содержит 3 приложения.

# СОДЕРЖАНИЕ

АННОТАЦИЯ.....	2
СОДЕРЖАНИЕ .....	3
СПИСОК ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ.....	5
ВВЕДЕНИЕ.....	6
1 ПОСТАНОВКА ЗАДАЧ РАЗРАБОТКИ.....	7
1.1 Общетеchnическое обоснование разработки.....	7
1.1.1 Постановка задачи проектирования.....	7
1.1.2 Описание предметной области.....	8
1.1.3 Выбор критериев качества.....	9
1.1.4 Анализ прототипов и аналогов.....	11
2 ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ .....	18
2.1 Понятие классификации в машинном обучении.....	18
2.2 Описание алгоритмов классификации.....	20
2.3 Описание критериев сравнения моделей классификации.....	23
3 КОНСТРУКТОРО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ.....	27
3.1 Конструкторская часть.....	27
3.1.1 Выбор программных средств.....	27
3.1.1.1 Выбор языка программирования.....	27
3.1.1.2 Выбор фреймворка для разработки.....	28
3.1.1.3 Выбор среды для разработки.....	30
3.1.1.4 Выбор технологии создания веб-приложения.....	31
3.1.1.5 Выбор аппаратных средств.....	31
3.2 Технологическая часть.....	32
3.2.1 Разработка системы.....	32
3.2.1.1 Загрузка данных.....	32
3.2.1.2 Анализ датасета.....	32
3.2.1.3 Предварительная обработка данных.....	32
3.2.1.4 Кодирование категориальных признаков.....	33

3.2.1.5 Масштабирование данных.....	34
3.2.1.6 Корреляционный анализ данных.....	36
3.2.1.7 Формирование обучающей и тестовой выборок.....	38
3.2.1.8 Обучение моделей.....	38
3.2.1.9 Сравнение результатов моделей и вывод.....	40
3.2.2 Разработка веб-приложения.....	44
ЗАКЛЮЧЕНИЕ.....	51
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	52
ПРИЛОЖЕНИЕ А ГРАФИЧЕСКАЯ ЧАСТЬ.....	54

## СПИСОК ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ

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**

В современном мире объем данных, с которым мы имеем дело, постоянно растет. От области здравоохранения до финансов, от технологий до научных исследований, данные играют ключевую роль в понимании явлений, выявлении закономерностей и принятии обоснованных решений. В этом контексте машинное обучение становится более важным инструментом, позволяющим автоматизировать анализ данных, строить прогностические модели и принимать решения на основе данных.

Одним из наиболее распространенных типов задач в машинном обучении является классификация, которая включает предсказание 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 обосновывается необходимостью 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 областях.

# **1 ПОСТАНОВКА ЗАДАЧ РАЗРАБОТКИ**

## **1.1 Общетехническое обоснование разработки**

### **1.1.1 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, предоставление пользователю результатов анализа алгоритмов машинного обучения с использованием Pandas.

Для достижения поставленной цели разработки необходимо решить следующие задачи:

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

### 1.1.2 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 порегions РФ.

Данные представлены за период 2001-2019 г.

### 1.1.3 Выбор критериев качества

Каждый набор данных характеризуется некоторым количеством параметров, определяющим его качество и возможность дальнейшего использования этого датасета при обучении алгоритмов машинного обучения.

Выделим следующие наиболее значимые для решения поставленной задачи разработки критерии качества набора данных:

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

**Полнота данных** важна для обеспечения корректности и достоверности анализа данных, так как пропущенные значения могут исказить 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 является ключевым 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 определять, 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 включает в себя 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.

#### 1.1.4 Анализ прототипов и аналогов

Рассмотрим наиболее распространённые и известные наборы данных, используемые для анализа безработицы.

Набор данных «Безработица в Европейском Союзе» (**EU**). Данные содержат информацию по полу и возрасту в Европейском Союзе. Временной интервал с января 1983 года до июля 2020 года. [1]

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

Набор данных «Статистические данные о занятости и безработице среди населения по возрастным группам» (**RU**) предоставляет информацию о численности экономически активного населения, безработных, уровне безработицы и сопоставляет эти показатели между различными возрастными группами по субъектам РФ. Данные представлены за период 2001-2019 г. [3]

Сравним перечисленные варианты наборов данных для обучения алгоритмов машинного обучения методом взвешенной суммы. Показатели важности критериев определим, используя метод базового критерия. Обозначения для каждого из вариантов определены в таблице 1.

Таблица 1 — Варианты для выбора набора данных

Вариант	Обозначение
EU	B1
USA	B2
RU	B3

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

Таблица 2 — Критерии для сравнения вариантов набора данных

Критерий	Код критерия
Полнота данных	K1
Точность данных	K2
Актуальность данных	K3
Понятность и интерпретируемость	K4
Доступность	K5

Переведенные из качественных в количественные критерии сравнения аналогов представлены в таблицах 3-7.

Таблица 3 — Вербально-числовая шкала для критерия K1 (Полнота данных)

Качественное описание критерия	Большая	Средняя	Малая
Балл	3	2	1

Таблица 4 — Вербально-числовая шкала для критерия K2 (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

Количественные и качественные параметры рассматриваемых языков программирования определены в таблице 8.

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

Далее необходимо сравнить рассматриваемые варианты наборов данных на Парето-оптимальность. Результаты представлены в таблице 9.

Таблица 9 – Сравнение вариантов на Парето-оптимальность

Вариант набора данных	Вариант набора данных		
	B1	B2	B3
B1	0	0	0
B2	0	0	0
B3	0	0	0
Результат сравнения	0	0	0
Парето-оптимальность варианта	Да	Да	Да

Все варианты Парето-оптимальны, поэтому продолжим их сравнение.

Коэффициенты важности показателей сравнения (критериев) назначаем по методу базового критерия. Для этого разобьём все показатели на группы важности:

1) В первую группу включим показатель сравнения, который считаем наименее значимыми из набора: 2;

2) Во вторую группу включим показатель, который считаем более значимыми, по сравнению с первым: 3, 5;

3) В третью группу включаем показатели, которые считаем наиболее значимыми: 1, 4;

Тогда имеем следующие исходные данные:

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

$n_1 = 1; n_2 = 2; n_3 = 2$  – количество локальных критериев, которые соответственно входят в состав первой, второй и третьей группы.

$k_1 = 1$ ;  $k_2 = 2$ ;  $k_3 = 4$  – коэффициенты, которые показывают степень превосходства 2-ой, и 3-ей группы над критериями 1-ой группы.

Составим уравнение нормировки локальных критериев:

$$\sum_{i=1}^g n_i \cdot k_i \cdot \alpha = 1 \quad (1)$$

Тогда по формуле (1) имеем:

$$1 \cdot \alpha + 2 \cdot 2 \cdot \alpha + 2 \cdot 4 \cdot \alpha = 1$$

Получаем коэффициент важности:  $\alpha = 0,077$

Далее определяем коэффициенты важности для каждой из групп:

$$\alpha_1 = 0,077; \alpha_2 = 0,154; \alpha_3 = 0,308$$

Необходимо нормализовать значения локальных критериев, для этого применим формулы:

В случае, если нормализация критериев «чем больше, тем лучше»:

$$k_{ij} = \frac{X_{ij}}{X_i^+}, \quad (2)$$

где  $X_i^+ = \max_j X_{ij}$

Если нормализация критериев «чем больше, тем хуже»:

$$k_{ij} = \frac{X_i^-}{X_{ij}}, \quad (3)$$

где  $X_i^- = \min_j X_{ij}$ ,

$k_{ij}$  – коэффициент нормализации, определяет уровень соответствия  $i$ -го параметра  $j$ -го варианта наилучшему значению,  $0 < k_{ij} \leq 1$ .

Для всех вариантов по каждому критерию нормируем значения. Итоговые значения и коэффициенты важности локальных критериев представлены в таблице 10.

Согласно методу взвешенной суммы для каждого из вариантов необходимо вычислить сумму произведения коэффициента важности локального критерия и коэффициента нормализации (формула 4).

$$Y_j = \sum_i^n \alpha_i k_{ij} \quad (4)$$

Таблица 10 — Нормализованные значения показателей вариантов сравнения

Код критерия	Коэффициент важности локального критерия ( $\alpha_i$ )	Нормированное значение локального критерия		
		$k_{i1}$	$k_{i2}$	$k_{i3}$
K1	0,308	0,33	0,67	1
K2	0,077	1	1	0,5
K3	0,154	1	0,67	0,33
K4	0,308	0,5	0,5	1
K5	0,154	1	1	1



$Y_j = \sum_i^n \alpha_i \cdot k_{ij}$	0,64	0,69	0,86
--	------	------	------

Наилучший вариант находим по формуле (5):

$$Y_l = \max_{j \in m} Y_j, \quad (5)$$

где m – количество вариантов сравнения.

По полученным результатам можно сделать вывод о том, что наиболее подходящим вариантом набора данных для реализации задачи, поставленной в данной работе, является вариант В3 - датасет RU.

## **2 ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ**

### **2.1 Понятие классификации в машинном обучении**

Машинное обучение (Machine Learning) — это подраздел искусственного интеллекта, который занимается созданием алгоритмов и моделей, которые позволяют компьютерам извлекать закономерности из данных и принимать на их основе решения.

Основная идея машинного обучения заключается в том, чтобы обучить компьютер находить шаблоны и закономерности в данных, которые могут быть использованы для прогнозирования будущих данных или принятия решений.

Контролируемое обучение или обучение с учителем (Supervised learning) — это подкатегория машинного обучения. При контролируемом обучении машина обучается на наборе посеченных данных, что означает, что входные данные сопоставляются с желаемыми выходными данными. Затем машина учится прогнозировать выходные данные для новых входных данных. Обучение с учителем часто используется для таких задач, как классификация, регрессия и обнаружение объектов [4].

Классификация — это процесс категоризации данных или объектов в заранее определенные классы или категории на основе их характеристик или атрибутов.

Классификация машинного обучения — это тип метода обучения с учителем, при котором алгоритм обучается на помеченном наборе данных для прогнозирования класса или категории новых, невидимых данных.

Основная цель классификационного машинного обучения — построить модель, которая может точно присвоить метку или категорию новому наблюдению на основе его особенностей [5].

Алгоритм классификации, основанный на данных обучения, представляет

собой метод контролируемого обучения, используемый для классификации новых наблюдений. При классификации программа использует предоставленный набор данных или наблюдений, чтобы научиться классифицировать новые наблюдения по различным классам или группам. Например, 0 или 1, красный или синий, да или нет, спам или не спам и т. д. Для описания классов можно использовать цели, метки или категории. Алгоритм классификации использует помеченные входные данные, поскольку это метод обучения с учителем и включает входную и выходную информацию.

На рисунке 1 представлены два класса: класс А и класс В. Эти классы имеют функции, похожие друг на друга и отличающиеся от других классов [6].

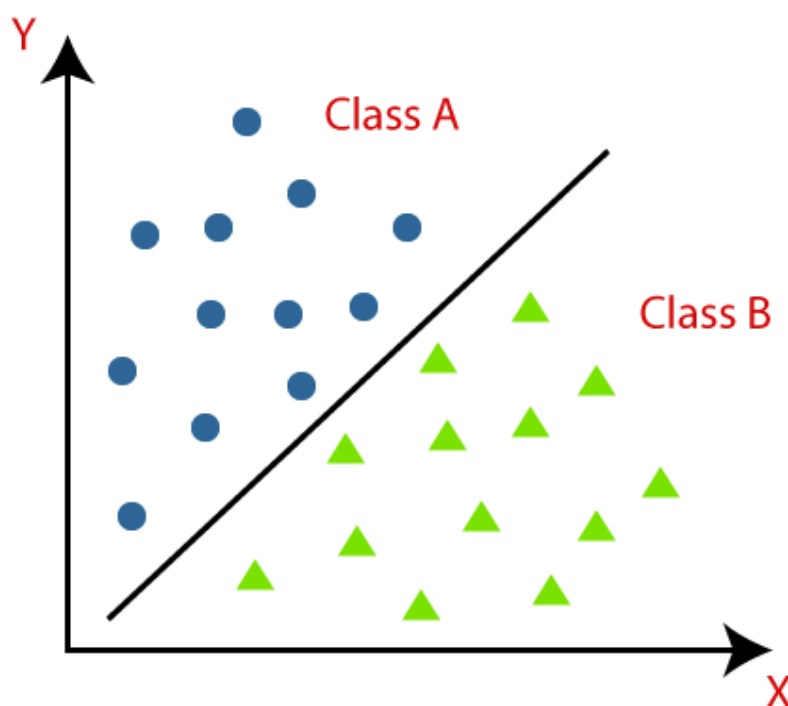


Рисунок 1 – Классификация

В качестве примера классификации можно использовать задачу борьбы со спамом в электронной почте. Модель машинного обучения учится находить характеристики писем, например, использование определенных слов или фраз, наличие конкретного отправителя и классифицировать как «спам» и «не спам». После обучения модель работает автоматически, пользователь не видит спам во входящих.

## **2.2 Описание алгоритмов классификации**

В машинном обучении используются разные алгоритмы классификации, которые позволяют определить категориальную метку (класс) на основе входных данных.

В настоящее время существует несколько популярных алгоритмов классификации.

### **Логистическая регрессия (Logistic Regression)**

Логистическая регрессия – это контролируемый алгоритм машинного обучения, который используется для бинарной классификации, где используется два класса. Она вычисляет вероятность принадлежности наблюдения к конкретному классу с использованием логистической функции. Этот метод основывается на линейной модели и часто применяется для задач, где необходимо предсказать вероятность события.

Цель логистической регрессии - найти наиболее подходящую взаимосвязь между зависимой переменной и набором независимых переменных. У него будет только два результата.

График логистической регрессии – кривая, которую описывает сигмовидная функция, принимающая значение на интервале от 0 до 1. Он представлен на рисунке 2.

# Logistic Regression

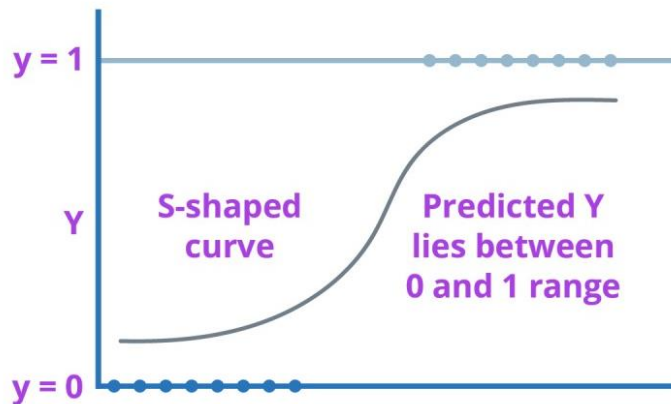


Рисунок 2 – Логистическая регрессия

## **К-ближайших соседей (KNN)**

KNN классифицирует новые данные на основе класса большинства среди  $k$ -ближайших соседей в обучающем наборе данных. Алгоритм основан на подсчете количества объектов каждого класса в сфере с центром в распознаваемом (классифицируемом) объекте. Классифицируемый объект относят к тому классу, объектов у которого больше всего в этой сфере. Визуализация алгоритма K-Nearest Neighbors представлена на рисунке 3.

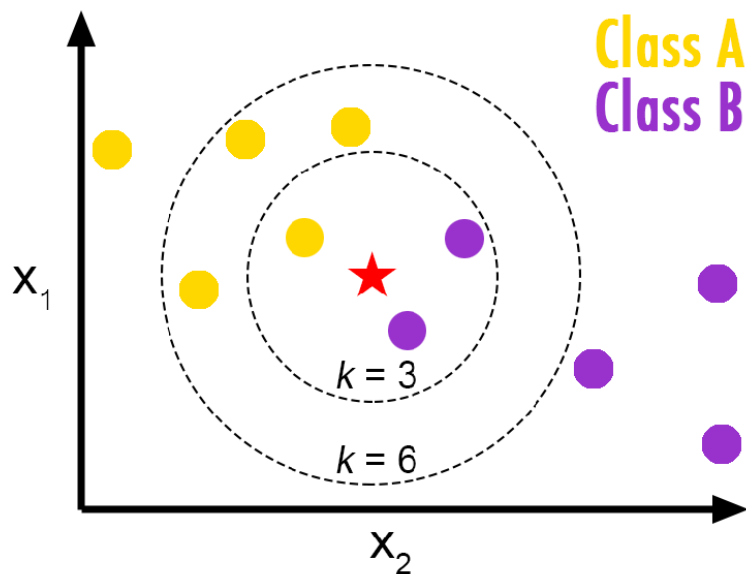


Рисунок 3 - KNN-ближайших соседей

В данном случае у нас есть точки данных класса А и В. Мы хотим предсказать, что представляет собой звезда (тестовая точка данных). Если мы рассмотрим значение  $k$ , равное 3 (3 ближайшие точки данных), мы получим прогноз класса В. Если же мы рассмотрим значение  $k$ , равное 6, мы получим прогноз класса А.

### Случайный лес (Random Forest)

Случайный лес – это ансамблевый метод, который строит множество деревьев решений и объединяет их результаты для улучшения точности и контроля переобучения. Каждое дерево в случайном лесу обучается на различном подмножестве данных, а случайность, добавленная во время обучения, помогает улучшить общие показатели модели и ее обобщающие способности.

Случайный лес сопоставляет несколько деревьев с различными подвыборками наборов данных, а затем использует среднее значение для повышения точности прогнозирования модели. Размер подвыборки всегда совпадает с исходным размером входных данных, но выборки часто рисуются с заменами. Визуализация алгоритма Случайный лес представлена на рисунке 4.

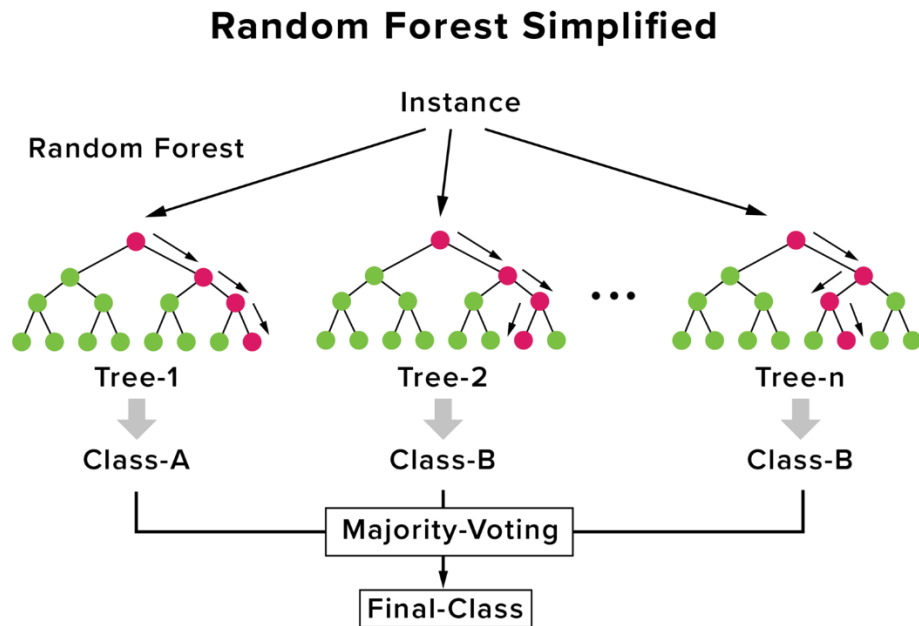


Рисунок 4 - Случайный лес

## 2.3 Описание критериев сравнения моделей классификации

Оценка моделей классификации играет важную роль в машинном обучении, так как помогает определить, насколько хорошо модель справляется с задачей. Основные критерии оценки моделей классификации включают метрики точности, полноты, точности и F1- меры и другие.

Далее необходимо выбрать метрики оценки модели. Были выбраны стандартные метрики для задач классификации: precision, recall, F1-score и ассигасу [7]. В вычислении каждой из этих метрик используется матрица ошибок. С помощью матрицы ошибок можно визуализировать эффективность алгоритма классификации, сравнивая ожидаемое значение целевой переменной с ее реальным значением.

Таблица 11 – Матрица ошибок

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

TP — истинно положительный (true positive), классификатор правильно отнёс объект к рассматриваемому классу.

TN — истинно отрицательный (true negative), классификатор верно утверждает, что объект не принадлежит к рассматриваемому классу.

FP — ложноположительный (false positive), классификатор неверно отнёс объект к рассматриваемому классу.

FN — ложноотрицательный (false negative), классификатор неверно утверждает, что объект не принадлежит к рассматриваемому классу.

### Accuracy

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

Доля правильных ответов – относительное количество корректно классифицированных объектов (процент или доля правильно классифицированных объектов). Метрика ассурасу может сразу сказать, правильно ли обучается модель и как она может работать в целом. Однако она не дает подробной информации о её применении к проблеме. Проблема с использованием ассурасу в качестве основной метрики производительности заключается в том, что она не работает, когда имеется серьезный дисбаланс классов. Чтобы объективно оценить модель, нужно использовать и другие метрики.



## Precision

$$Precision = \frac{TP}{TP+FP} \quad (7)$$

Precision будет показывать часть правильно распознанных объектов заданного класса по отношению к общему числу объектов, принятых классификатором за объекты заданного класса. Активно используется, когда важно фиксировать количество FP предсказаний. В некоторых задачах, например, при обнаружении спама, ложное срабатывание является худшей ошибкой, чем ложноотрицательное.

## Recall

$$Recall = \frac{TP}{TP+FN} \quad (8)$$

Recall – это противоположность precision. Эта метрика будет показывать отношение правильно распознанных объектов к общему числу объектов данного класса. Измеряет ложноотрицательные результаты по сравнению с истинными положительными. Ложноотрицательные результаты особенно важны для задач обнаружения болезней и других прогнозах, связанных с безопасностью.

## F1-score

$$F1 = 2 * \frac{precision*recall}{precision+recall} \quad (9)$$

Если необходимо сбалансировать две цели: высокую точность и высокую отзывчивость, то стоит использовать F1-score. F1-score рассчитывается как среднее гармоническое значение precision и recall. Хотя мы использовать и

простое среднее из двух оценок, гармонические средние более устойчивы к выбросам. Таким образом, оценка F1 представляет собой сбалансированный показатель, который надлежащим образом количественно определяет правильность моделей во многих областях.

### 3 КОНСТРУКТОРСКО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

#### 3.1 Конструкторская часть

##### 3.1.1 Выбор программных средств

###### 3.1.1.1 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. [9]

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 возникает необходимость 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 [17].

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. В целом, Python является идеальным выбором для специалистов, занимающихся обработкой и анализом данных, благодаря своей производительности, удобству использования и широким возможностям для разработки разнообразных проектов.

### 3.1.1.2 Выбор фреймворка для разработки

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

**Pandas** – библиотека для анализа данных, основанная на языке программирования Python. Библиотека pandas представляет собой набор инструментов для операций с данными: сортировки, фильтрации, очистки, удаления дубликатов, агрегирования, создания сводных таблиц и т. д. 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 [10].

Pandas построена на основе структуры данных, называемой DataFrame. DataFrame библиотеки pandas представляет собой таблицу, похожую 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 преимуществом библиотеки **pandas** является 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. Её функциональные возможности включают в себя поддержку многомерных массивов (включая матрицы) и поддержку высокоуровневых математических функций, предназначенных для работы с многомерными массивами. Библиотека NumPy предоставляет реализации вычислительных алгоритмов (в виде функций и операторов), оптимизированные для работы с многомерными массивами [18].

**Matplotlib** – библиотека для построения научных графиков в Python. Она включает функции для создания высококачественных визуализаций, таких как линейная диаграмма, гистограмма, диаграмма разброса и т.д. [11].

**Seaborn** – библиотека визуализации данных для языка программирования Python, основанная на библиотеке Matplotlib. Она предназначена для создания статических графиков, которые помогают визуально представить информацию из наборов данных [12].

**Scikit-learn** – 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 [13].

### 3.1.1.3 Выбор среды разработки

В качестве среды разработки выбран Jupyter Notebook.

**Jupyter Notebook** – интерактивная среда для запуска программного кода в браузере. Это отличный инструмент 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 изображения. 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 заключается в возможности взаимодействия. С помощью этой программы можно выполнять отдельные фрагменты и блоки кода в любом порядке. Результаты работы моментально выводятся на экран рядом с кодом [16].

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

### 3.1.1.4 Выбор технологии создания веб-приложения

Для создания веб-приложения было использован 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 [14].

Этот фреймворк является одним из лучших для демонстрации проектов, связанных с анализом данных (data science). Он позволяет создавать интерактивные веб-приложения благодаря нескольким строкам кода. Streamlit превращает скрипты данных в веб-приложения для совместного использования за считанные минуты [15].

В сущности, каждое веб-приложение Streamlit – это скрипт Python.

### 3.1.1.5 Выбор аппаратных средств

Аппаратные компоненты компьютера представляют собой совокупность электрических, электронных и механических устройств, обеспечивающих его функционирование.

Локально исходный код системы запускался на ноутбуке со следующими характеристиками:

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

## **3.2 Технологическая часть**

### **3.2.1 Разработка системы**

#### **3.2.1.1 Загрузка данных**

Данные для анализа были загружены из CSV файла с использованием библиотеки Pandas. Фрагмент кода загрузки данных приведен на рисунке 4.

```
Ввод [2]: data = pd.read_csv('data.csv')
```

Рисунок 5 - Загрузка датасета

#### **3.2.1.2 Анализ датасета**

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

#### **3.2.1.3 Предварительная обработка данных**

После детального изучения датасета, была выполнена замена названия колонок для улучшения читабельности, проверка и обработка пропущенных значений, проверка дубликатов и обработка типов данных.



### 3.2.1.4 Кодирование категориальных признаков

Кодирование категориальных признаков – это процесс преобразования текстовых или категориальных значений признаков в числовые для использования 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 5.

```
Ввод [19]: data['Год'].unique()
Out[19]: array([2001, 2004, 2007, 2009, 2012, 2013, 2015, 2017, 2019], dtype=object)
```

Для решения задачи классификации выберем два класса. Это год 2001 и 2019

```
Ввод [20]: data[data['Год'].isin([2001, 2019])]
```

Out[20]:

Введенные в экономику	Безработные	Уровень экономической активности	Уровень занятости	Уровень безработицы	Численность безработных (до 20 лет)	Численность безработных (от 20 до 29 лет)	Численность безработных (от 30 до 39 лет)	Численность безработных (от 40 до 49 лет)	Численность безработных (от 50 до 59 лет)	Численность безработных (60 и более лет)	Год
64400	6416	64.2	58.4	9.1	526.1	1969.7	1591.2	1552.7	622.4	154.000000	2001
17181	1156	64.6	60.5	6.3	91.3	344.5	284.4	282.1	120.2	33.500000	2001
689	48	65.4	61.1	6.5	2.9	21.0	8.3	9.4	4.3	2.400000	2001
588	65	61.0	54.9	10.0	3.9	21.7	18.3	15.4	5.4	0.700000	2001
728	80	65.4	58.9	9.9	10.6	26.1	17.2	20.1	4.0	2.500000	2001
...	...	...	...	...	...	...	...	...	...	...	...
385	21	63.3	59.9	5.4	0.2	7.9	6.4	4.1	3.3	5.143452	2019
81	3	72.4	69.1	4.6	0.1	0.5	1.2	0.4	1.1	0.700000	2019
260	14	68.7	65.2	5.2	0.3	3.9	2.8	2.4	1.7	3.100000	2019
72	4	59.7	56.0	6.2	0.2	0.9	1.3	1.3	0.8	0.300000	2019
30	1	80.5	77.4	3.8	0.0	0.6	0.4	0.1	0.1	5.143452	2019

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 6.

```
Ввод [23]: # присваиваем целочисленные значения для каждой категории
labelencoder = LabelEncoder()
data_1['Год'] = labelencoder.fit_transform(data_1['Год'])
data_1 = data_1.astype({"Год": "int64"})
data_1['Год'].unique()

Out[23]: array([0, 1], dtype=int64)
```

Рисунок 7 - Присваивание целочисленного значения

### 3.2.1.5 Масштабирование данных

Масштабирование данных – это процесс преобразования 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 7.

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (10)$$

Фрагмент кода Min-Max масштабирования приведен на рисунке 7.

```
Ввод [29]: # Преобразование значения признаков таким образом,  
# чтобы они находились в диапазоне от 0 до 1  
sc1 = MinMaxScaler()  
sc1_data = sc1.fit_transform(data_1[scale_cols])
```

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.

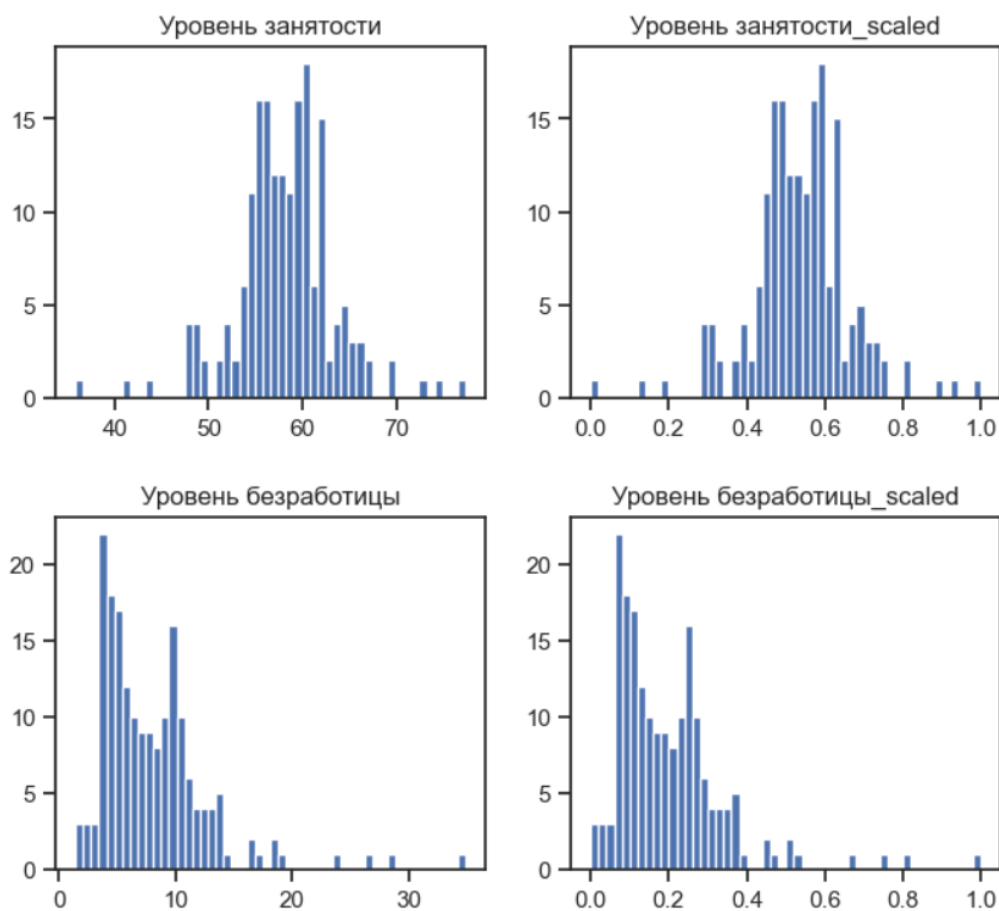


Рисунок 9 - Гистограммы

Масштабирование данных не повлияло на распределение данных.

### 3.2.1.6 Корреляционный анализ данных

Корреляционный анализ данных – статический метод, который используется 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.

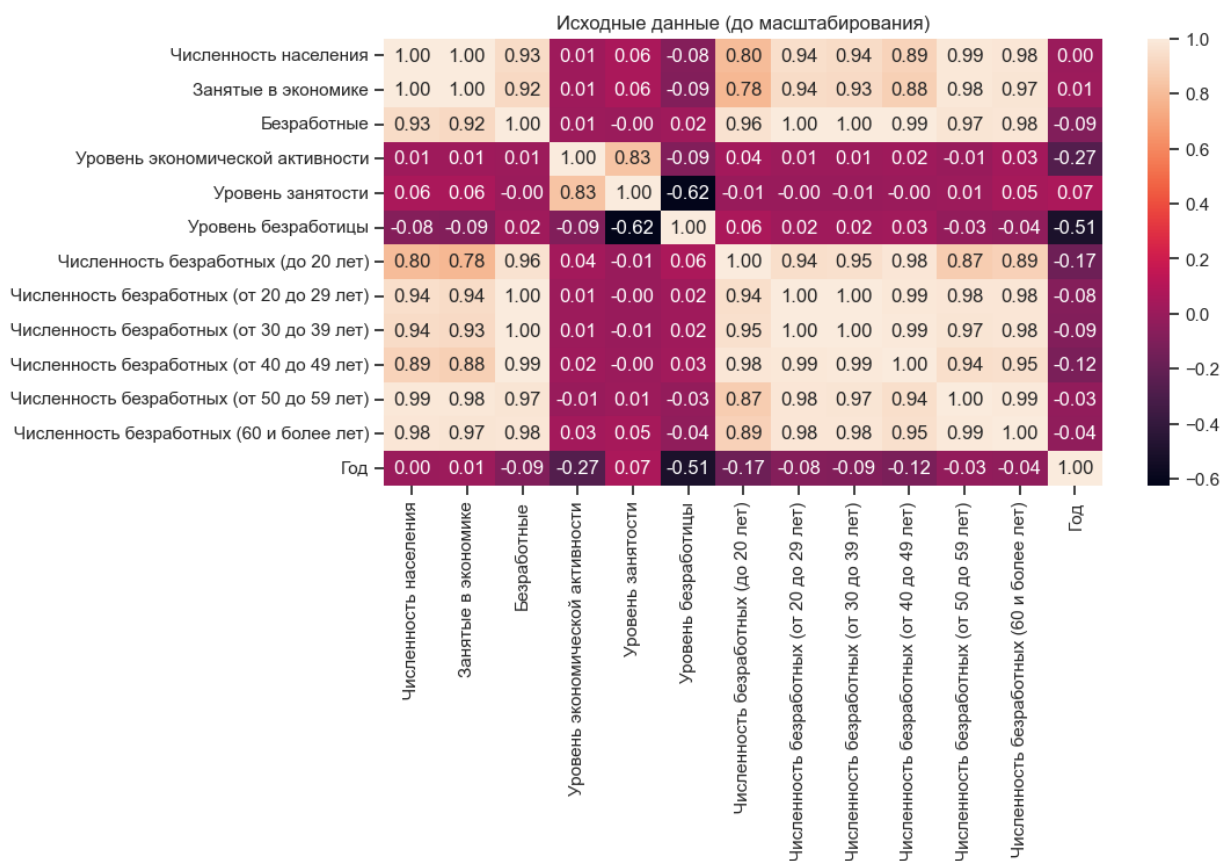


Рисунок 10 - Исходные данные

На рисунке 11 изображена матрица корреляции масштабированных данных.

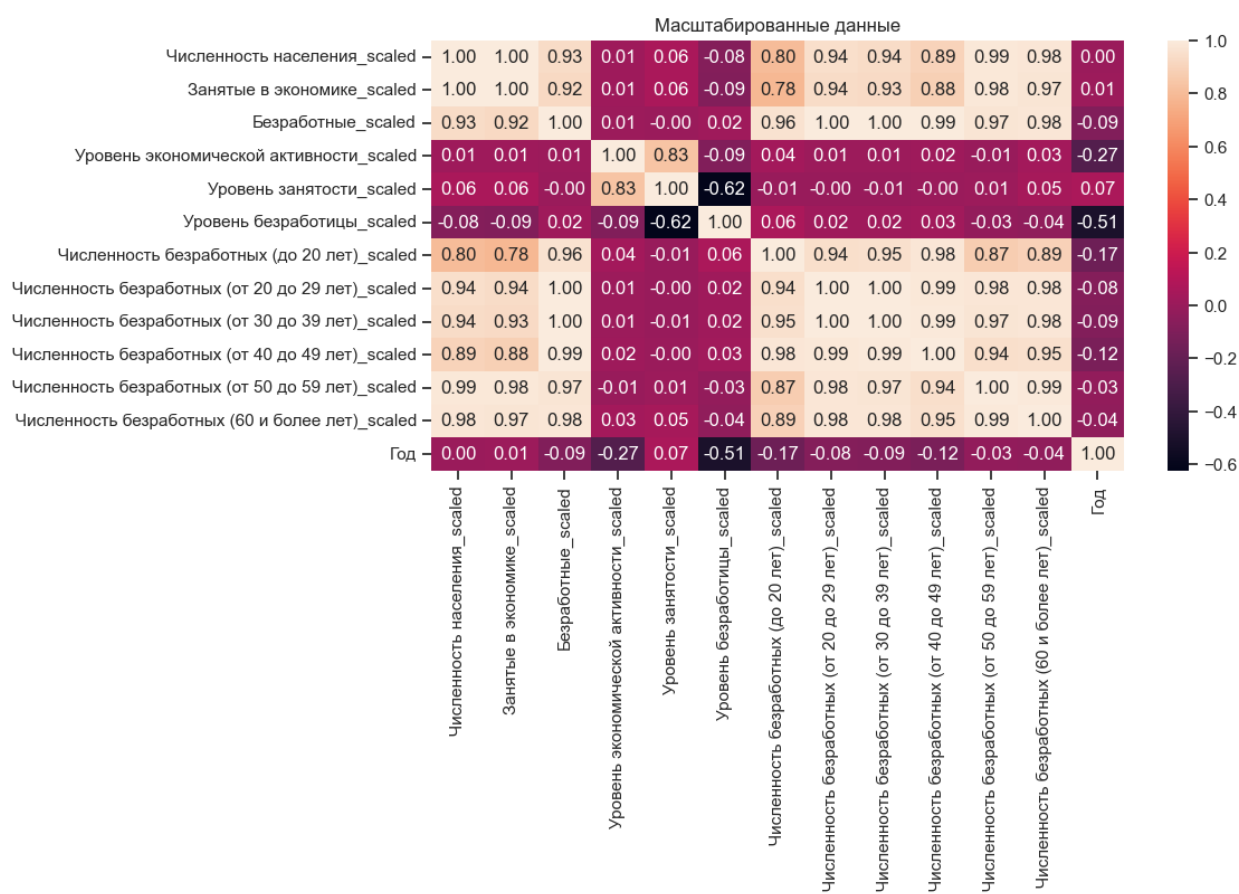


Рисунок 11 - Масштабированные данные

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.

### 3.2.1.7 Формирование обучающей и тестовой выборок на основе исходного набора данных

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

Разделение данных на обучающую и тестовую выборку приведено на рисунке 12.

```
Ввод [37]: x = data_1[['Уровень занятости', 'Занятые в экономике', 'Численность населения']] # Наименование признаков
           y = data_1['Год'] # Целевая переменная

Ввод [38]: x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

Ввод [39]: # Размер обучающей выборки
           x_train.shape, y_train.shape

Out[39]: ((147, 3), (147,))

Ввод [40]: # Размер тестовой выборки
           x_test.shape, y_test.shape

Out[40]: ((37, 3), (37,))
```

Рисунок 12 - Разделение данных

### 3.2.1.8 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.

```
# Создание и обучение модели логистической регрессии
model = LogisticRegression()
model.fit (X_train, y_train)
y_pred = model.predict(X_test)

# Предсказание и оценка производительности
accuracy_model = accuracy_score(y_test, y_pred)
precision_model = precision_score(y_test, y_pred)
recall_model = recall_score(y_test, y_pred)
f1_model = f1_score(y_test, y_pred)
```

Рисунок 13 - Обучение логистической регрессии

Обучение модели К-ближайших соседей приведено на рисунке 14.

```
# Создание и обучение модели К-ближайших соседей
model2 = KNeighborsClassifier()
model2.fit (X_train, y_train)
y_pred2 = model2.predict(X_test)

# Предсказание и оценка производительности
accuracy_model2 = accuracy_score(y_test, y_pred2)
precision_model2 = precision_score(y_test, y_pred2)
recall_model2 = recall_score(y_test, y_pred2)
f1_model2 = f1_score(y_test, y_pred2)
```

Рисунок 14 - Обучение К-ближайших соседей

Обучение модели случайный лес приведено на рисунке 15.

```
# Создание и обучение модели случайный лес
model3 = RandomForestClassifier()
model3.fit(X_train, y_train)
y_pred3 = model3.predict(X_test)

# Предсказание и оценка производительности
accuracy_model3 = accuracy_score(y_test, y_pred3)
precision_model3 = precision_score(y_test, y_pred3)
recall_model3 = recall_score(y_test, y_pred3)
f1_model3 = f1_score(y_test, y_pred3)
```

Рисунок 15 - Обучение случайного леса

### 3.2.1.9 Сравнение результатов моделей и вывод

Оценим полученные модели с помощью приведенных выше метрик. Результаты оценки моделей приведены на рисунке 16.

```
Logistic Regression:
Accuracy: 0.8108108108108109
Precision: 0.8260869565217391
Recall: 0.8636363636363636
F1 Score: 0.8444444444444444

KNN:
Accuracy: 0.5405405405405406
Precision: 0.6470588235294118
Recall: 0.5
F1 Score: 0.5641025641025642

Random Forest:
Accuracy: 0.3783783783783784
Precision: 0.4666666666666667
Recall: 0.3181818181818182
F1 Score: 0.3783783783783784
```

Рисунок 16 - Результаты оценки моделей

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



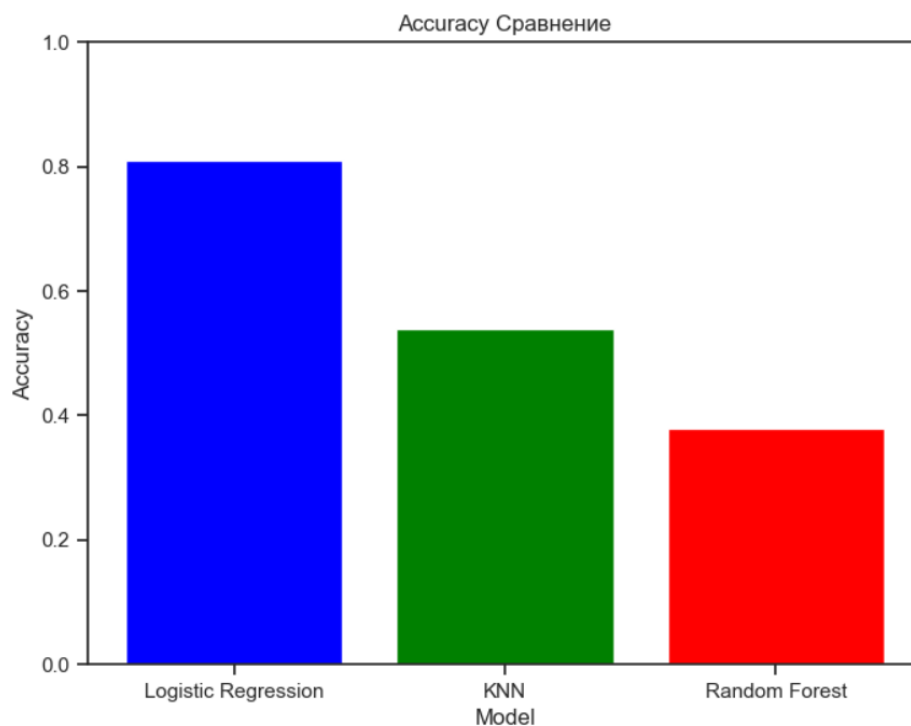


Рисунок 17 - Ассурасу сравнение

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

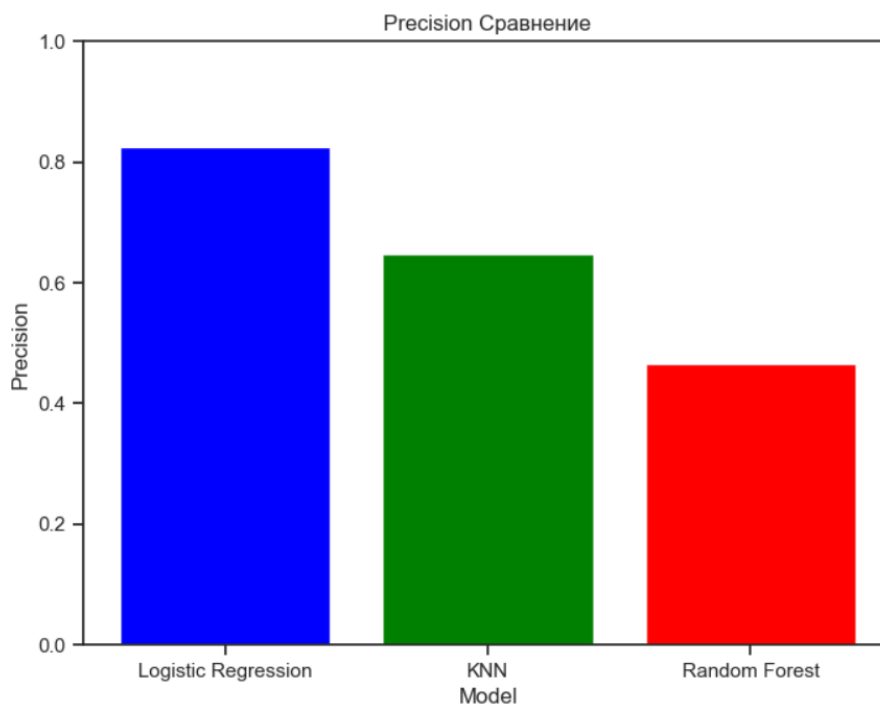


Рисунок 18 - Precision сравнение

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

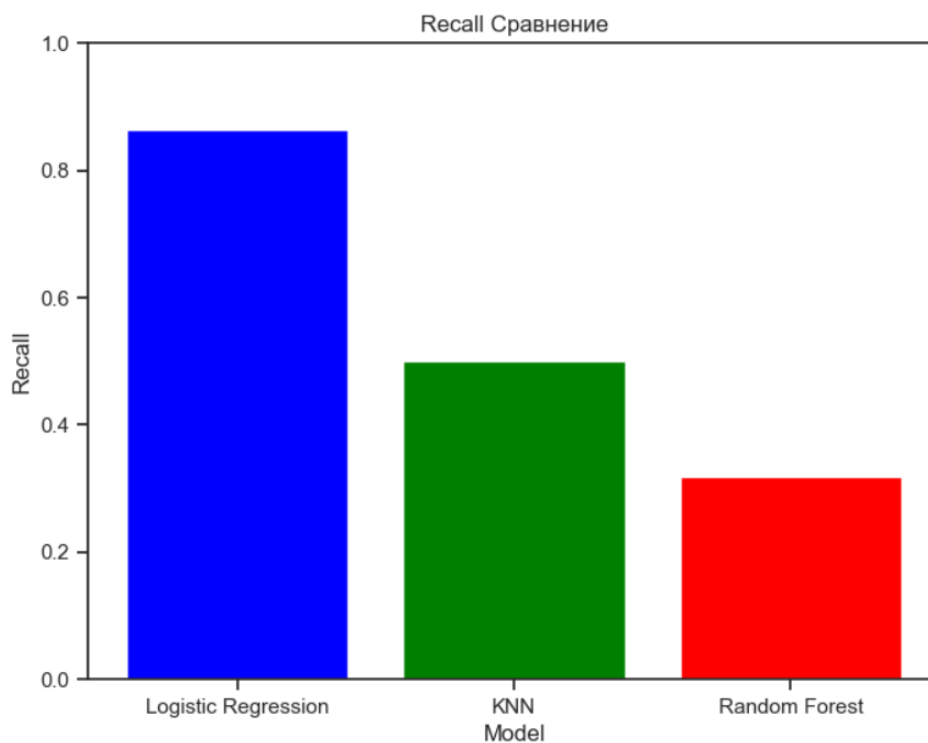


Рисунок 19 - Recall сравнение

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

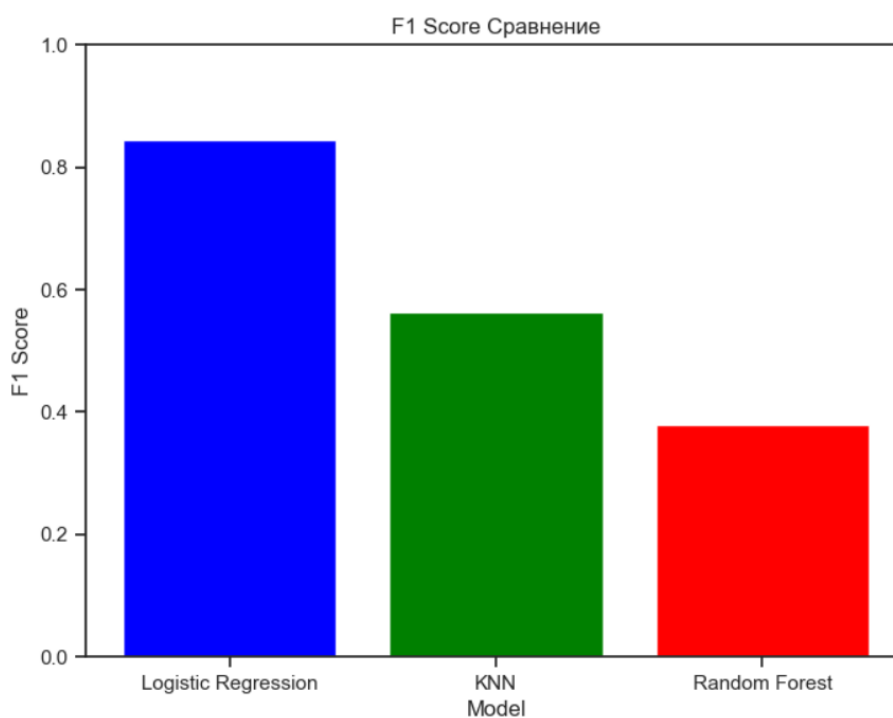


Рисунок 20 - F1 score сравнение

Из приведенных 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 настройкой 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.

### 3.2.2 Разработка веб-приложения

В соответствии с поставленными задачами разработки было разработано веб-приложение взаимодействия пользователя с системой анализа алгоритмов машинного обучения с использованием Pandas. Для удобного, красивого и понятного пользователю интерфейса была использована библиотека Streamlit.

При запуске веб-приложения отображается начальная страница «Система анализа алгоритмов машинного обучения для решения задач классификации с использованием Pandas» на рисунке 21. В ней отображаются название и описание системы.

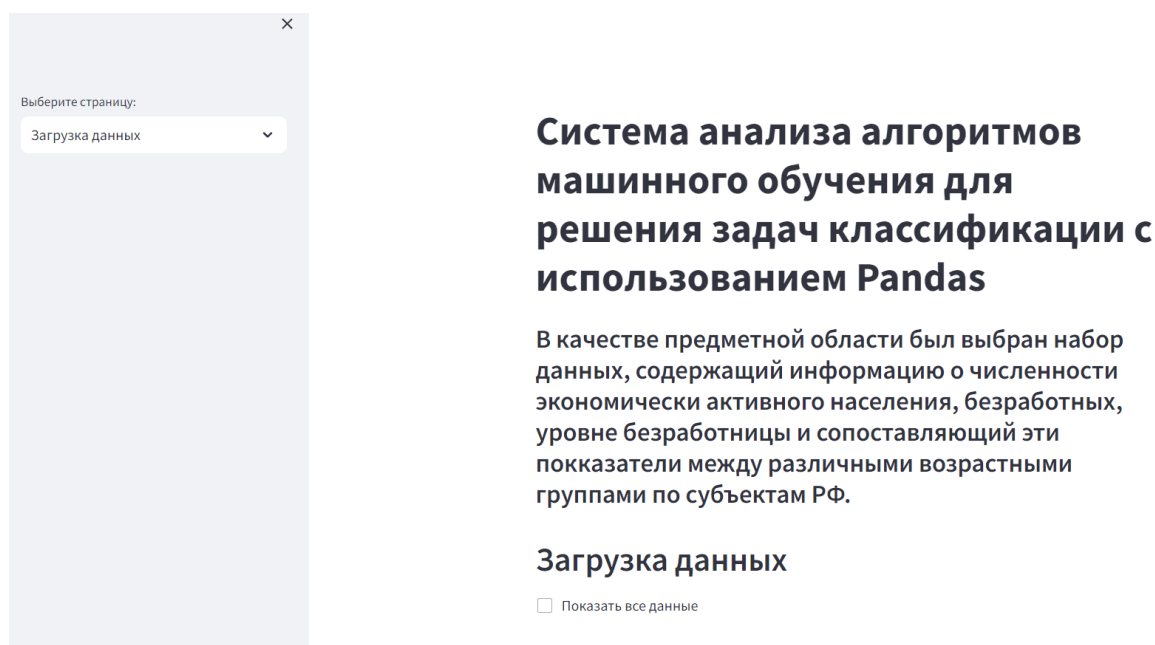


Рисунок 21 - Страница системы

Переход на другие страницы веб-приложения производится с помощью нажатия на нужный вариант в выпадающем списке «Выберите страницу», который находится на боковой панели. Это продемонстрировано на рисунке 22.

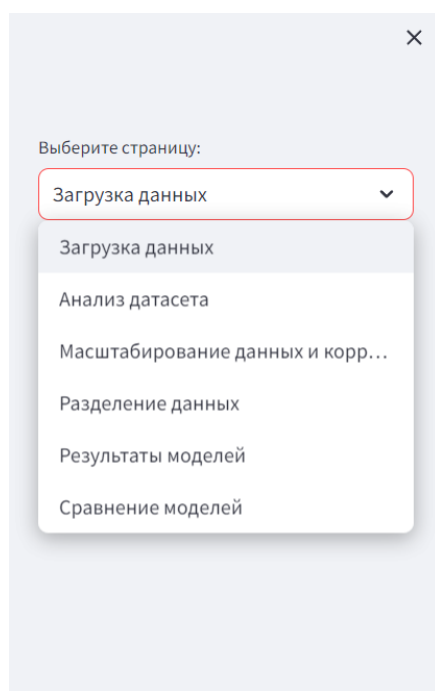


Рисунок 22 - Выпадающий список для перехода по страницам

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

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

Выберите страницу:

Загрузка данных

### Загрузка данных

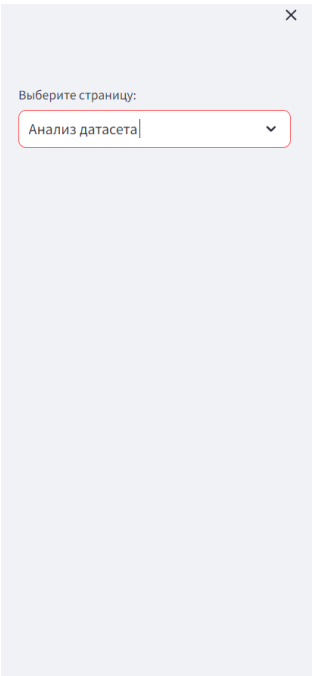
☒ Показать все данные

	Численность населения	Занятые в экономике	Безработные	Уровень экономической активности
0	70,816	64,400	6,416	64.2
1	18,337	17,181	1,156	64.6
2	737	689	48	65.4
3	653	588	65	61
4	808	728	80	65.4
5	1,116	1,009	107	60.1
6	588	555	33	62.6
7	550	517	33	66.3
8	380	357	22	64.2
9	622	557	65	63.1

Рисунок 23 - Страница «Загрузка данных»

При переходе на страницу «Анализ датасета» путём выбора в выпадающем списке на боковой панели соответствующего варианта пользователю отображается визуальный интерфейс, продемонстрированный на рисунке 24.

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



### Анализ датасета

Первые 5 значений

	Численность населения	Занятые в экономике	Безработные	Уровень экономической активности
0	70,816	64,400	6,416	64.2
1	18,337	17,181	1,156	64.6
2	737	689	48	65.4
3	653	588	65	61
4	808	728	80	65.4

Размер датасета:

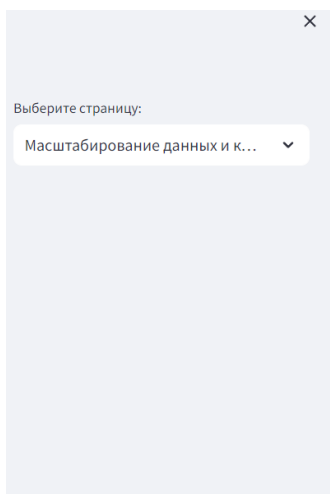
(184, 13)

Столбцы:

0	Численность населения
1	Занятые в экономике
2	Безработные
3	Уровень экономической активности
4	Уровень занятости
5	Уровень безработицы

Рисунок 24 - Страница «Анализ датасета»

При переходе на страницу «Масштабирование данных и корреляционный анализ» путём выбора в выпадающем списке на боковой панели соответствующего варианта пользователю отображается визуальный интерфейс с разделами «Масштабирование данных» и «Корреляционный анализ», продемонстрированный на рисунке 25.



## Масштабирование данных

☐ Показать данные

## Корреляционный анализ данных

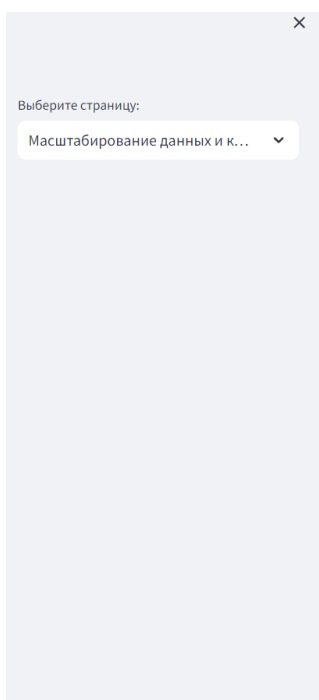
☐ Показать исходные данные (до масштабирования)

☐ Показать масштабированные данные

Рисунок 25 - Страница «Масштабирование и корреляция»

На этой странице находятся 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 представлено на рисунке 26.



## Масштабирование данных

☒ Показать данные

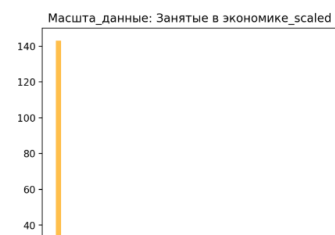
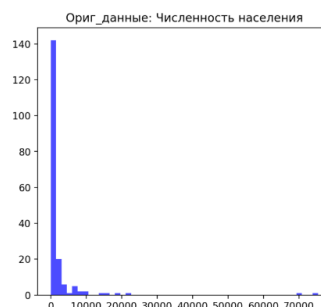


Рисунок 26 - Раздел «Масштабирование данных»

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 представлено на рисунке 27.

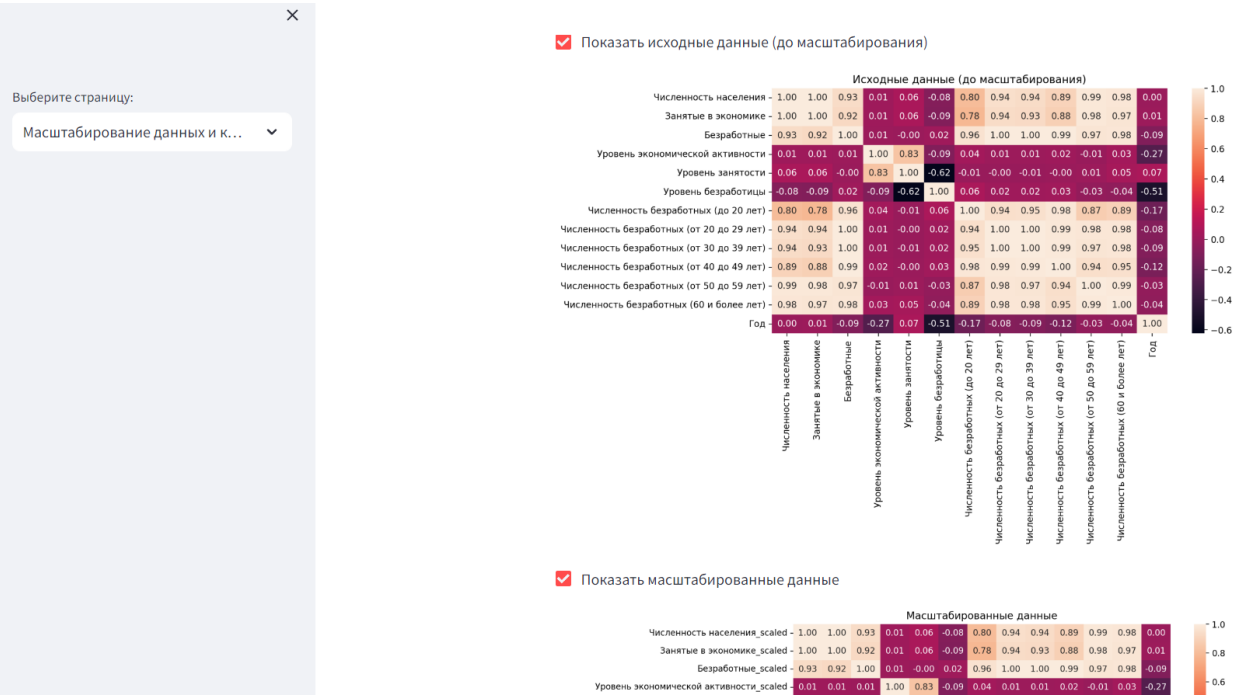


Рисунок 27 - Раздел «Корреляционный анализ»

При переходе на страницу «Разделение данных» путём выбора в выпадающем списке на боковой панели соответствующего варианта пользователю отображается визуальный интерфейс, продемонстрированный на рисунке 28. В ней отображаются размеры обучающей и тестовой выборки.



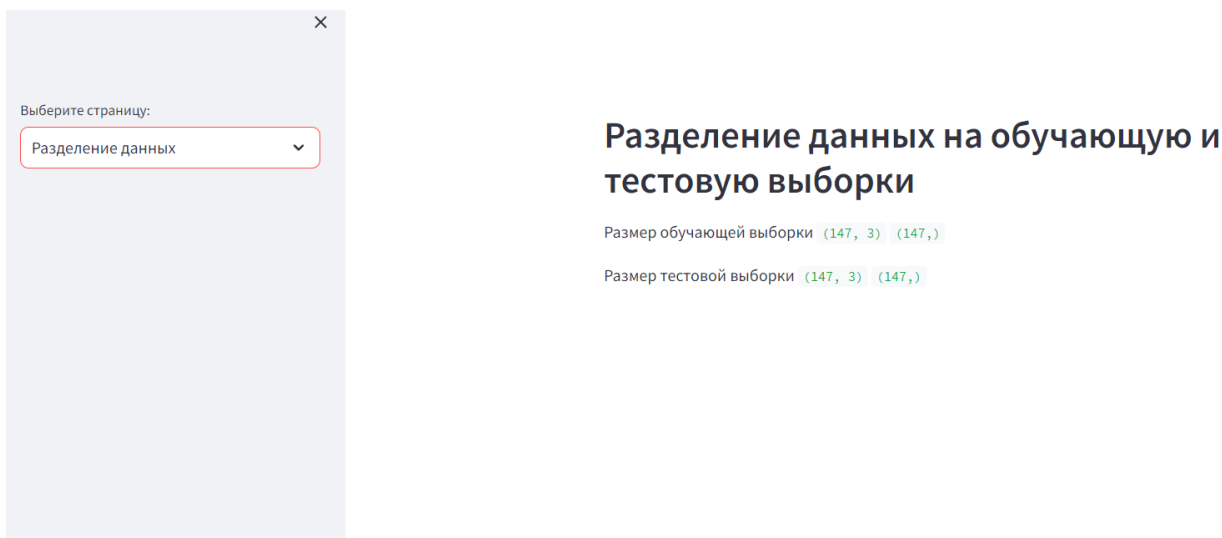


Рисунок 28 - Страница «Разделение данных»

При переходе на страницу «Оценка моделей» путём выбора в выпадающем списке на боковой панели соответствующего варианта пользователю отображается визуальный интерфейс, продемонстрированный 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 выбранной модели. Также происходит и с другими моделями: KNN, Random Forest.

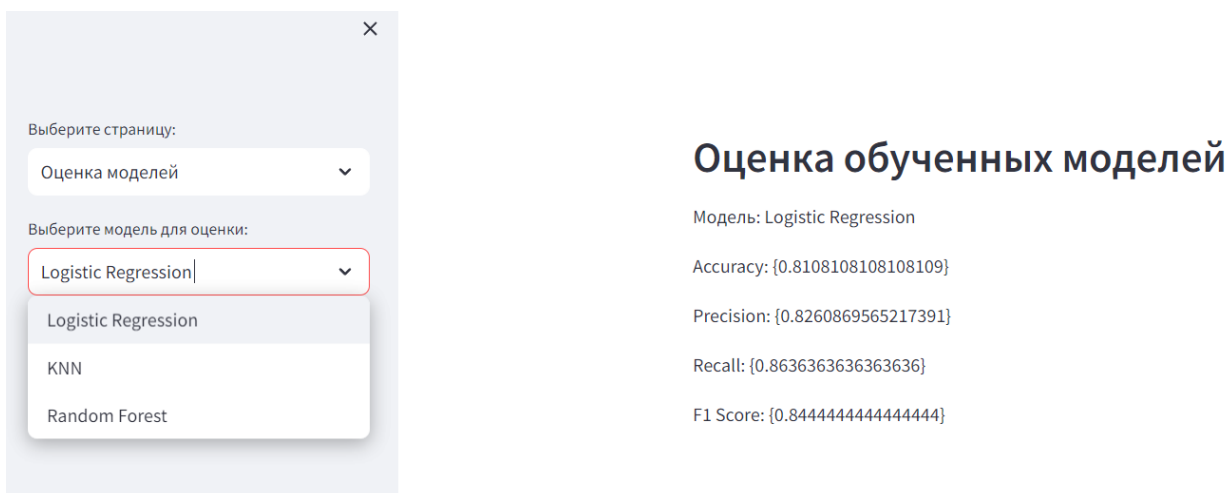


Рисунок 29 - Страница «Оценка моделей»

После перехода на страницу «Сравнение моделей», путём выбора в выпадающем списке на боковой панели соответствующего варианта пользователю отображается интерфейс, продемонстрированный на рисунке 30.

На этой странице показываются графики, отображающие разницу оценок между тремя моделями.

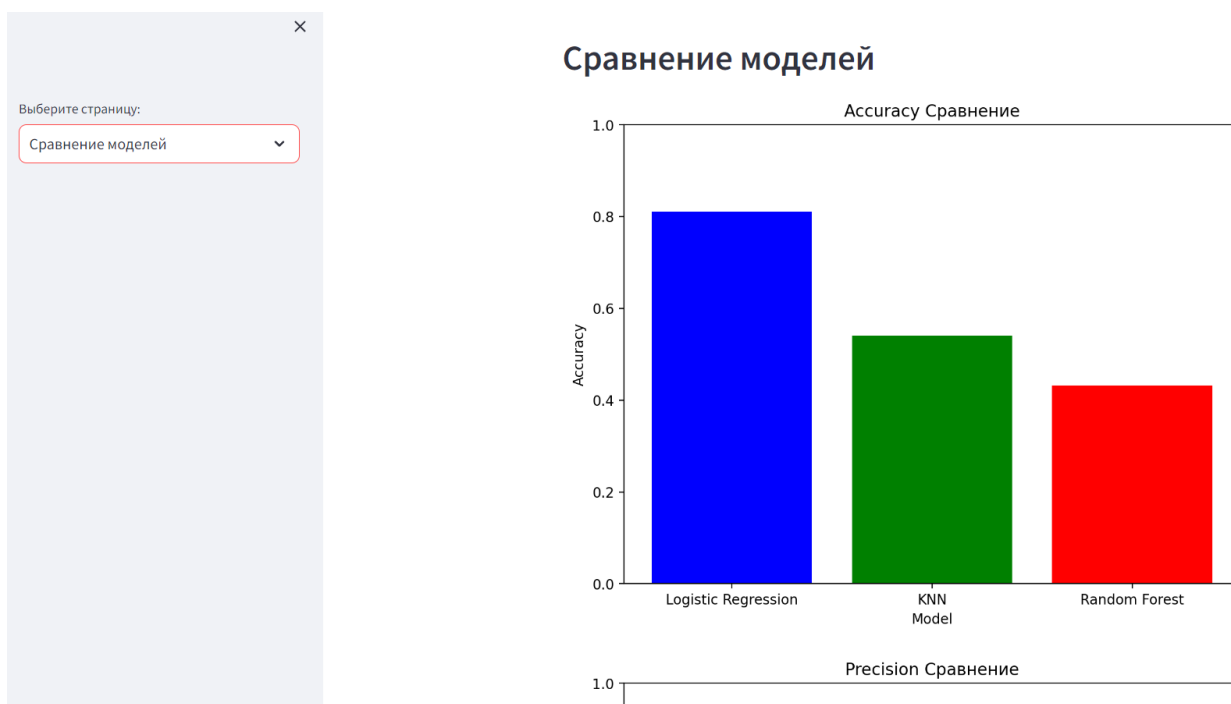


Рисунок 30 - Страница «Сравнение моделей»

## 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 выполнены 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 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.

Полученная система может быть доработана 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. Основные метрики задач классификации в машинном обучении [Электронный ресурс] // <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. Документация Python [Электронный ресурс] // python.org URL: <https://www.python.org/doc> (дата обращения: 20.05.2024).
10. Пасхавер Борис Pandas в действии. – СПб.: Питер, 2023. – 512 с.

11. Андреас Мюллер, Сара Гвидо Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными. / М.: Вильямс, 2017. – 480 с.
12. Официальный сайт Seaborn [Электронный ресурс] // seaborn.pydata.org URL.: <https://seaborn.pydata.org/> (дата обращения: 20.05.2024)
13. Документация Scikit-learn [Электронный ресурс] // scikit-learn.org URL.: [https://scikit-learn.ru/getting\\_started/](https://scikit-learn.ru/getting_started/) (дата обращения: 20.05.2024).
14. Документация Streamlit [Электронный ресурс] // docs.streamlit.io URL.: <https://docs.streamlit.io/> (дата обращения 20.05.2024)
15. Репозиторий курса “Технологии машинного обучения”, бакалавриат, 6 семестр [Электронный ресурс] // [https://github.com/ugapanyuk/courses\\_current/wiki/COURSE\\_TMO\\_SPRING\\_2023/](https://github.com/ugapanyuk/courses_current/wiki/COURSE_TMO_SPRING_2023/) (дата обращения: 12.03.2024).
16. Мухамедиев Р.И., Амиргалиев Е.Н. Введение в машинное обучение. Учебник. – М.: УМО РУМС, 2022. – 252 с.
17. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных /. – М.: ДМК Пресс, 2015. – 400 с.
18. Бизли Д. Python. Подробный справочник. – Пер. с англ. – СПб.: Символ-Плюс, 2018.- 864 с.

## **ПРИЛОЖЕНИЕ А ГРАФИЧЕСКАЯ ЧАСТЬ**

В графическую часть выпускной квалификационной работы входят:

- A.1. Цель работы
- A.2. Задачи
- A.3. Набор данных
- A.4. Средства разработки
- A.5. Обучение и оценка моделей
- A.6. Сравнение моделей
- A.7. Интерфейс приложения.
- A.8. Заключение