МИНОБРНАУКИ РОССИИ

Санкт-Петербургский государственный

электротехнический университет

«ЛЭТИ» им. В.И. Ульянова (Ленина)

Кафедра Вычислительной техники

отчет

**по лабораторной работе №1**

по дисциплине «Машинное обучение»

Тема: «Классификация»

|  |  |  |
| --- | --- | --- |
| Студент гр. 2308 |  | Попов Н.А. |
| Преподаватель |  | Турсуков Н.О. |

Санкт-Петербург, 2024

СОДЕРЖАНИЕ

[ТЕОРИЯ 3](#_Toc178975993)

[ПРЕДОБРАБОТКА ДАННЫХ 5](#_Toc178975994)

[МОДЕЛЬ KNN 9](#_Toc178975995)

[МОДЕЛЬ SVM 10](#_Toc178975996)

[МОДЕЛЬ РЕШАЮЩИЙ ЛЕС 12](#_Toc178975997)

[ВЫВОД 14](#_Toc178975998)

# ТЕОРИЯ

Классификация — это один из ключевых процессов в машинном обучении, цель которого заключается в том, чтобы на основе обучающей выборки научить модель распознавать и присваивать классы новым данным. Классификация относится к категории задач обучения с учителем, где алгоритм работает с уже размеченными данными, и его задача — научиться предсказывать категориальную метку (класс) для новых, еще не встречавшихся данных.

**Метод k-ближайших соседей (kNN)**

Алгоритм k-ближайших соседей (k-Nearest Neighbors, kNN) — это интуитивно понятный метод классификации, где решение о принадлежности нового объекта к классу принимается на основе данных о его ближайших "соседях". Идея заключается в том, что объекты, которые находятся близко друг к другу в пространстве признаков, скорее всего, принадлежат одному и тому же классу. Чтобы классифицировать новый объект, алгоритм находит k ближайших объектов из обучающей выборки. Затем происходит "голосование", и класс, к которому принадлежит большинство этих соседей, присваивается новому объекту. Таким образом, алгоритм основывается на плотности распределения классов в пространстве признаков.

**Метод опорных векторов (SVM)**

Метод опорных векторов (Support Vector Machine, SVM) — это мощный инструмент для задач классификации, который работает на принципе поиска разделяющей гиперплоскости, максимизирующей расстояние между классами. В двумерном пространстве эта гиперплоскость может быть представлена прямой линией, разделяющей два класса, а в многомерном — гиперплоскостью более сложной формы. Алгоритм SVM стремится найти такую гиперплоскость, которая максимально отделяет классы друг от друга. Ключевую роль здесь играют опорные векторы — это точки данных, которые находятся ближе всего к гиперплоскости и непосредственно влияют на её положение и ориентацию. Таким образом, SVM обеспечивает надежное разделение классов, минимизируя ошибки на границе между ними.

**Решающее дерево**

Решающее дерево — это алгоритм, который строит древовидную модель на основе данных и условий, которые проверяются на каждом узле дерева. Каждый внутренний узел представляет собой проверку значения определенного признака (например, "больше или меньше заданного порога"), а листья дерева содержат предсказания классов. Дерево строится таким образом, чтобы на каждом этапе выбор признака и порогового значения максимально разделял данные на разные классы. Процесс принятия решения можно представить в виде последовательных шагов, каждый из которых делит данные на более однородные группы, пока не будет достигнуто предсказание в листе дерева.

# ПРЕДОБРАБОТКА ДАННЫХ

Перед обучением модели сделаем предобработку данных. Заполним пропуски в данных, удалим ненужные переменные и приведем целевую переменную к нужному виду: 0/1 – Год не урожайный/Год урожайный.

Далее проведем разведочный анализ: посмотрим что является численными переменными – рисунок 1. Построим матрицу корреляций - рисунок 2, и по зависимым признакам диаграмму рассеяния – рисунок 3. Построим гистограммы по зависимым переменным– рисунок 4. Построим для них графики «ящик с усами» - рисунок 5, и очистим для них выбросы.

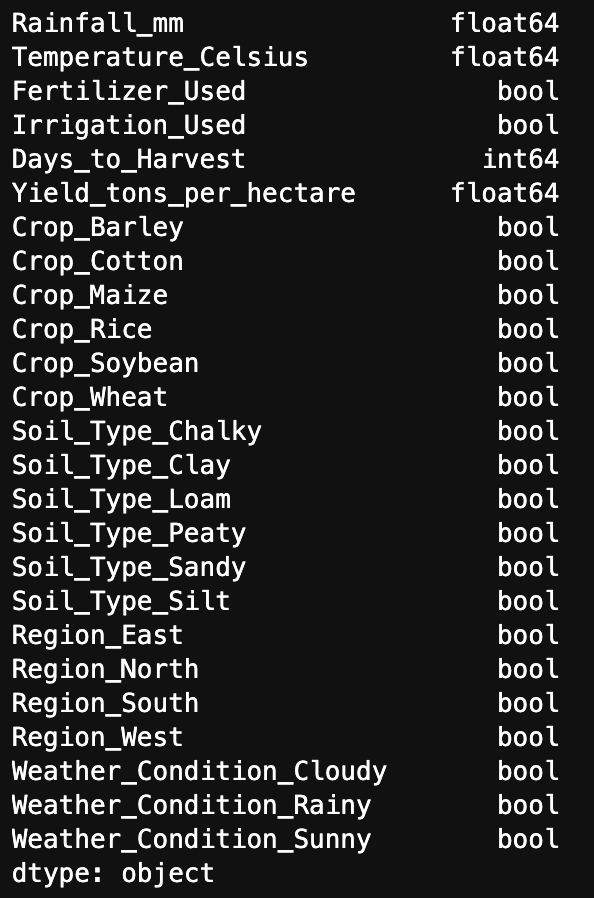


Рисунок 1 - Статистики по численным переменным.



Рисунок 2 – Матрица корреляций для определения взаимосвязей

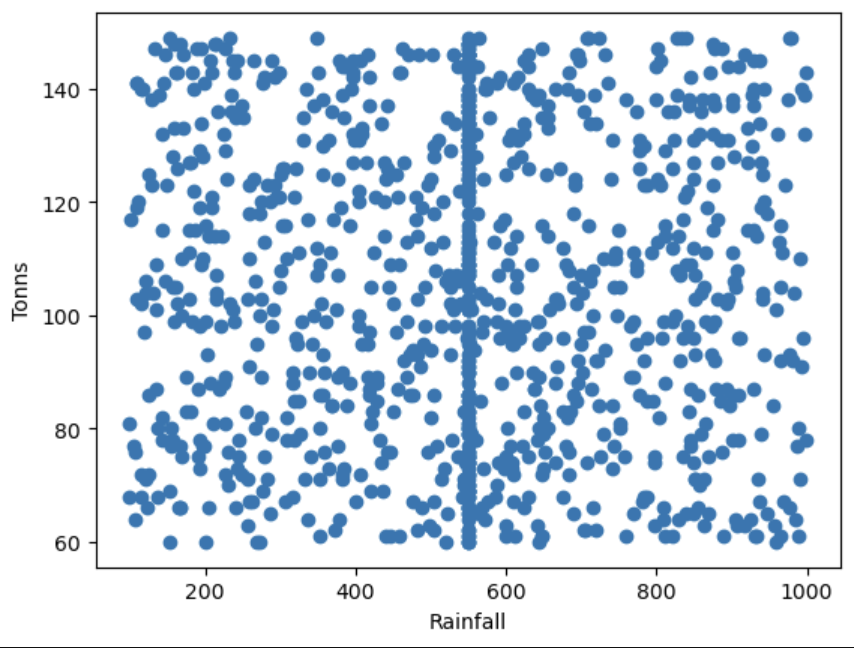


Рисунок 3 – Диаграмма рессеяния

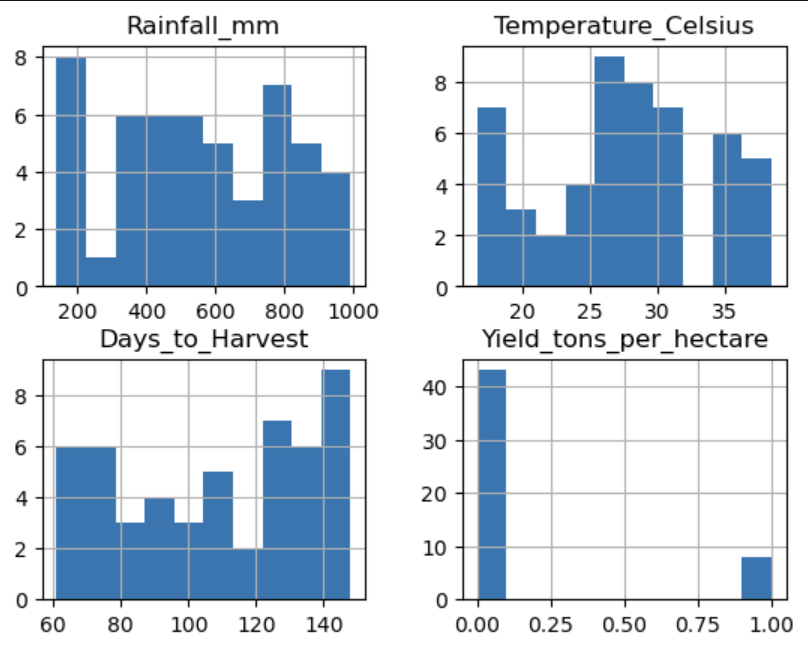


Рисунок 4 – Гистограммы.

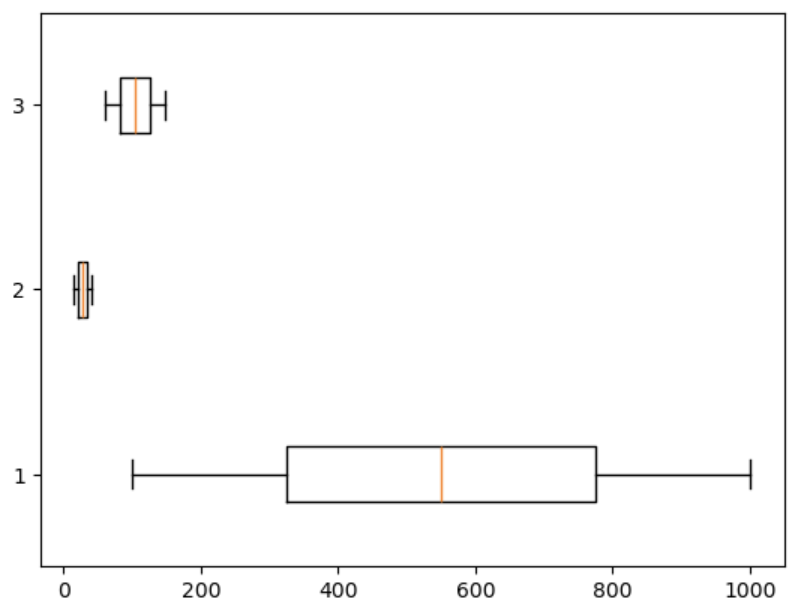


Рисунок 5 – «Ящики с усами».

Большинство алгоритмов машинного обучения не умеет работать с категориальными переменными, поэтому необходимо представить их в булевом виде при помощи метода OneHotEncoding– рисунок 6.



Рисунок 6 – Изменённые категориальные переменные.

Предварительно необходимо стандартизировать данные для нормальной работы модели. Для этого воспользуемся SimpleScaler из sklearn – Рисунок 7.

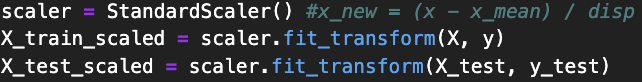


Рисунок 7 – Пример кода с методом fit\_transform

После всех преобразований можно приступать к обучению моделей.

# МОДЕЛЬ KNN

Для обучения модели KNN воспользуемся KNeighborsClassifier из библиотеки sklearn. С помощью RandomizedSearchCV подберем гиперпараметры модели. Обучим лучшую модель. Оценим качество модели на тестовой выборке: посчитаем метрики Precision, Recall, F1-score, ROC-AUC, а также построим ROC-AUC кривую – рисунок 8 для валидационной и тестовой выборок – рисунок 9.

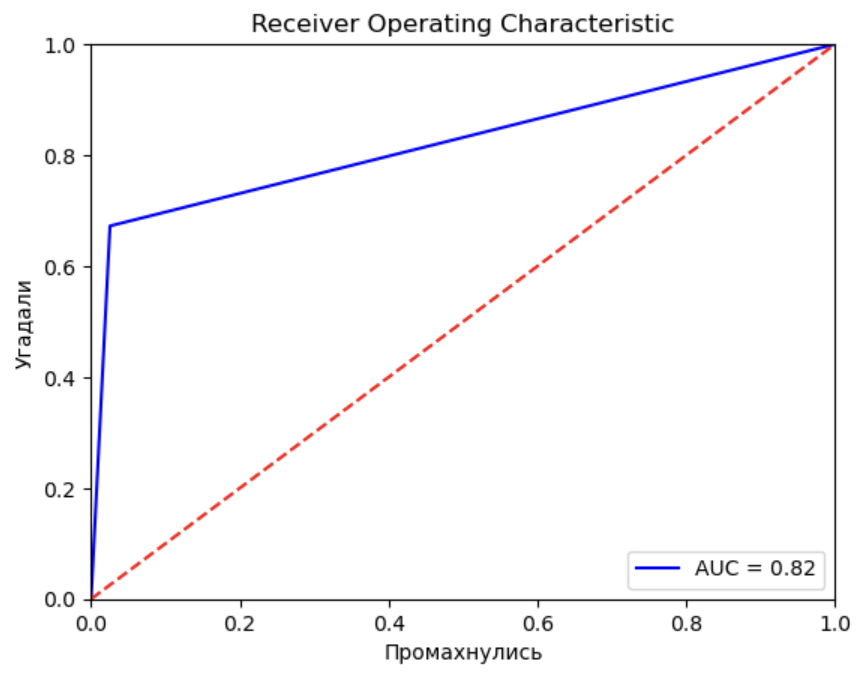


Рисунок 8 – ROC-AUC кривая для валидационной выборки.

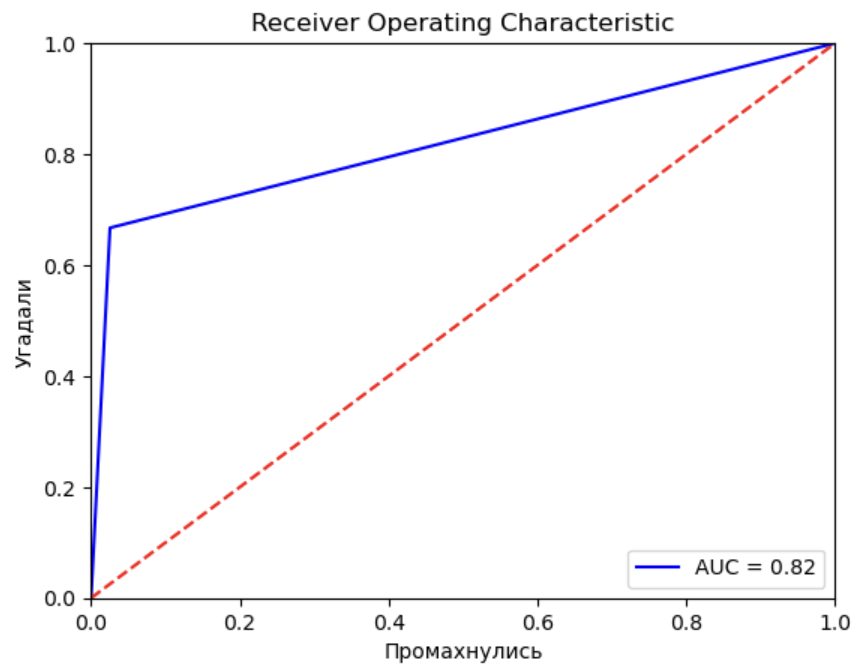


Рисунок 9 – ROC-AUC кривая для тестовой выборки.

# МОДЕЛЬ SVM

Для обучения модели SVM воспользуемся классом SVC библиотеки sklearn.

Подберем гиперпарметры с помощью GridSearchCV. Оценим качество модели на валидационной и тестовой выборках, а также построим ROC-AUC кривую – рисунок 10 для валидационной, и тестовой выборок – рисунок 11.

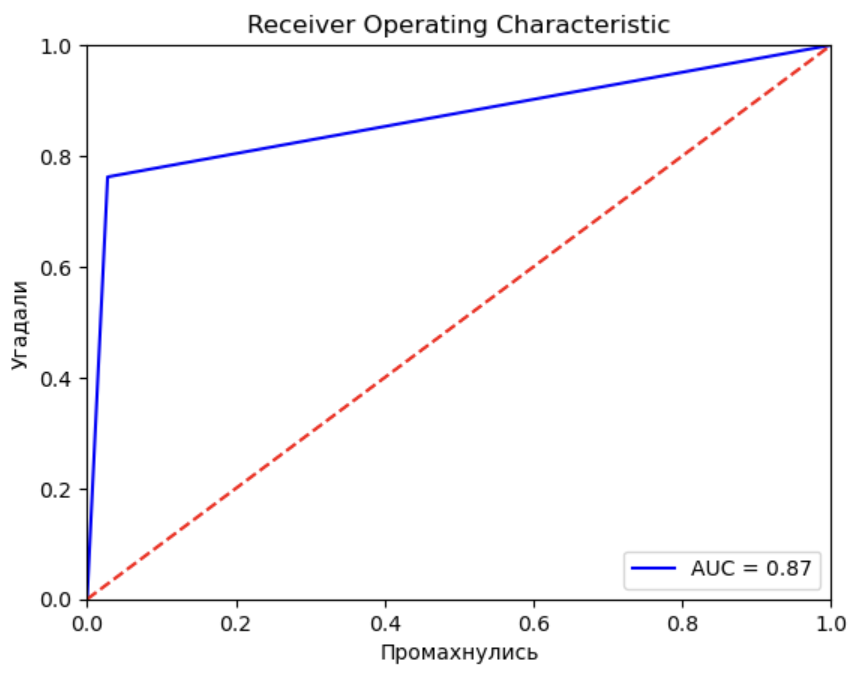


Рисунок 10 – ROC-AUC кривая для валидационной выборки.

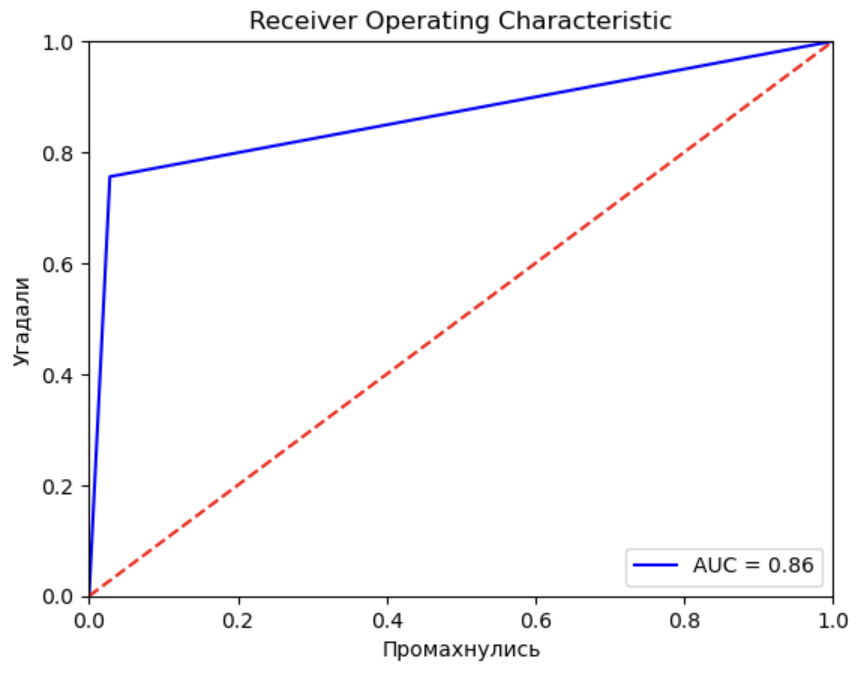


Рисунок 11 – ROC-AUC кривая для тестовой выборки.

# МОДЕЛЬ СЛУЧАЙНЫЙ ЛЕС

Для обучения модели решающего дерева воспользуемся классом RandomForestClassifier библиотеки sklearn. С помощью RandomizedSearchCV подберем гиперпараметры модели. Обучим лучшую модель. Оценим качество модели на тестовой выборке, а также построим ROC-AUC кривую для тестовой выборки – рисунок 12.

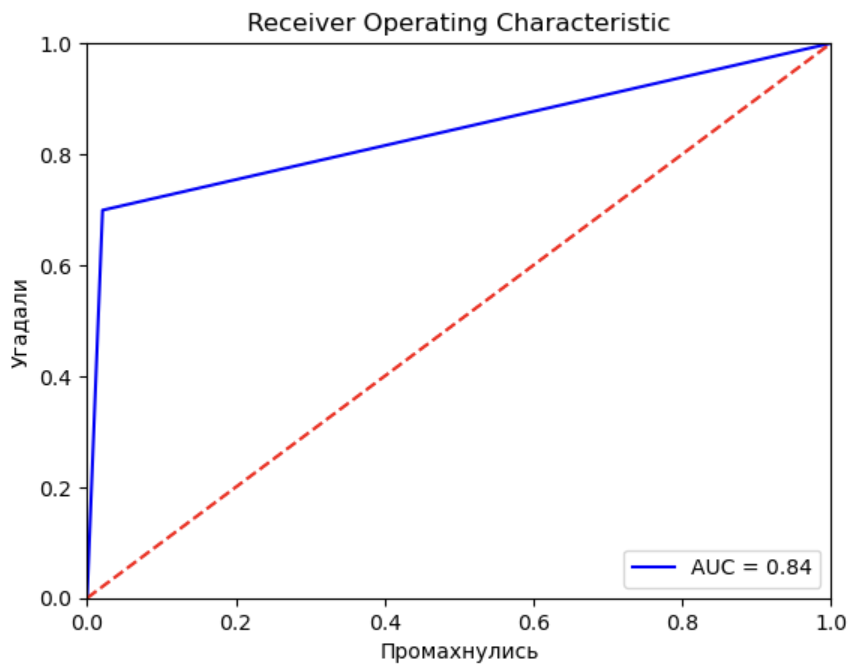


Рисунок 12 - ROC-AUC кривая для тестовой выборки.

Визуализируем feature importance – важность переменных для предсказания дерева. Feature importance представлен на рисунке 13.

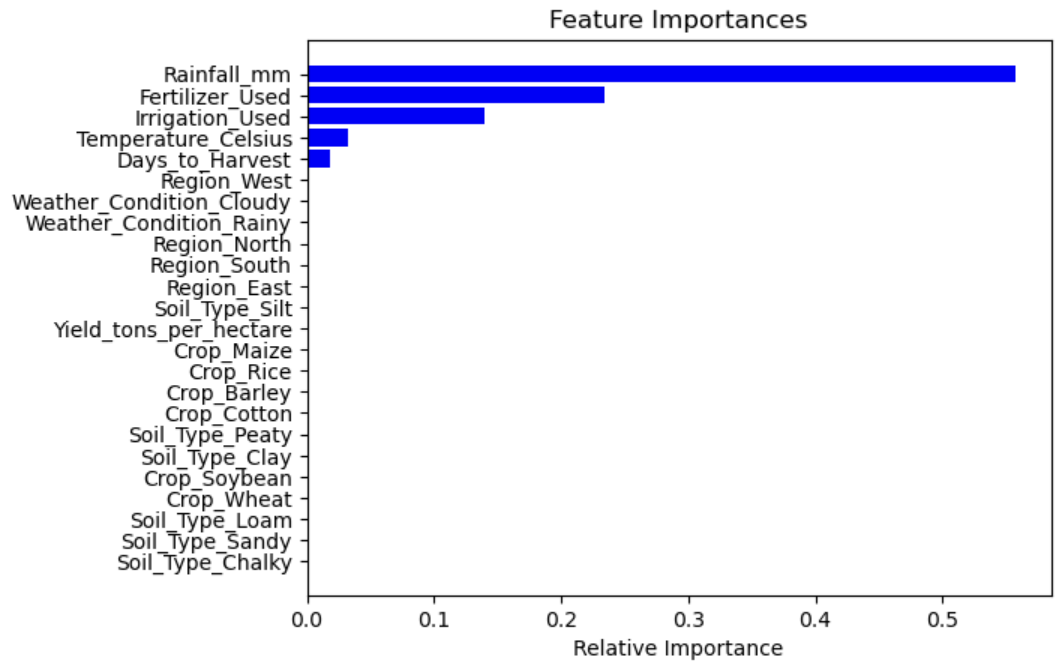


Рисунок 13 - Важность переменных для предсказания дерева.

# ВЫВОД

В ходе лабораторной работы был проведен сравнительный анализ эффективности трех широко используемых алгоритмов классификации: K ближайших соседей (k-Nearest Neighbors, KNN), метода опорных векторов (Support Vector Machine, SVM) и решающего дерева (Decision Tree). Каждый из данных алгоритмов был протестирован на одном и том же наборе данных с целью выявления их производительности и сравнительных характеристик. Для оценки эффективности применялись такие метрики, как **precision**, **recall**, **F1-score** и **ROC-AUC**, которые продемонстрировали высокие значения для всех методов, что свидетельствует о высокой точности классификации каждого алгоритма.