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

|  |  |
| --- | --- |
| ФАКУЛЬТЕТ | ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ |
| КАФЕДРА | СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ |

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ***

***НА ТЕМУ:***

|  |
| --- |
|  |
| ***Разработка и оценка моделей*** |
| ***Машинного обучения*** |
|  |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИУ5-61Б |  |  |  | А.В. Филенко |
|  | (группа) |  | (подпись, дата) |  | (И.О. Фамилия) |
| Руководитель НИР |  |  |  |  | Ю.Е. Гапанюк |
|  |  |  | (подпись, дата) |  | (И.О. Фамилия) |
|  |  |  |  |  |  |

*2025 г.*

**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)**

|  |  |  |
| --- | --- | --- |
| Утверждаю | | |
|  | | |
| Заведующий кафедрой | | ИУ5 |
|  | | (индекс) |
|  | В.И. Терехов | |
|  | (И.О. Фамилия) | |
| (подпись) |  | |
|  | (дата) | |

**ЗАДАНИЕ**

**на выполнение научно-исследовательской работы**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| по теме | | \_Разработка и оценка моделей методов машинного обучения | | | | | | | | | | | | | |
|  | | | |  | | | | | | | | | | | |
|  | | | |  | | | | | | | | | | | |
| Студент группы | | | | ИУ5-61Б | | | |  | | | | | | | |
|  | | | | Филенко Александр Владимирович | | | | | | | | | | | |
| Направленность НИР (учебная, исследовательская, практическая, производственная, др.) | | | | | | | | | | | | | | | |
| ИССЛЕДОВАТЕЛЬСКАЯ | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
| Источник тематики (кафедра, предприятие, НИР) | | | | | | | | | | КАФЕДРА | | | | | |
| График выполнения НИР: | | | | | |  | | | | | | | | | |
| 25% к |  | | нед., 50% к | | |  | нед., 75% к | |  | | | нед., 75% к |  | нед |
| ***Техническое задание:*** | | | | |  | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
| ***Оформление научно-исследовательской работы:*** | | | | | | | | | | |  | | | | |
| Расчетно-пояснительная записка на \_\_\_\_\_ листах формата А4. | | | | | | | | | | | | | | | |
| Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.) | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | |

Дата выдачи задания « 07 » \_февраля\_\_\_ 2025 г.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Руководитель НИР** |  |  |  | Ю.Е. Гапанюк |
|  |  | (подпись, дата) |  | (И.О. Фамилия) |
| **Студент** |  |  |  | А.В. Филенко |
|  |  | (подпись, дата) |  | (И.О. Фамилия) |

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

**Оглавление**

[**Введение** 4](#_Toc199698306)

[**Постановка задачи** 4](#_Toc199698309)

[**Подбор и подготовка данных** 5](#_Toc199698311)

[**Исследовательский анализ данных (EDA)** 7](#_Toc199698312)

[**Обработка и преобразование признаков** 10](#_Toc199698313)

[**Построение и сравнение моделей** 11](#_Toc199698314)

[**Настройка гиперпараметров** 12](#_Toc199698315)

[**Заключение** 14](#_Toc199698316)

[**Список использованных источников** 15](#_Toc199698317)

**Введение**

Машинное обучение сегодня занимает важное место в сфере анализа данных и активно используется для решения различных задач, таких как классификация, регрессия и кластеризация. В данной работе рассматривается задача бинарной классификации на основе классического набора данных — Titanic, загруженного непосредственно из репозитория datasciencedojo. Необходимо определить, выжил ли пассажир, используя такие признаки, как возраст, пол, класс обслуживания и другие характеристики, представленные в исходном датасете.

Цель исследования — выполнить полный цикл анализа данных и построения моделей машинного обучения. Я провожу детальную предварительную обработку данных, включающую обработку пропущенных значений (медианой по группам, заполнением 'Unknown', модой), создание новых признаков (например, Title, FamilySize, IsAlone) и кодирование категориальных переменных с использованием LabelEncoder. Выполнен обширный исследовательский анализ данных с визуализациями распределений признаков, целевой переменной, корреляционной матрицы и взаимосвязей между признаками. Обучено несколько моделей бинарной классификации, включая Логистическую Регрессию, Дерево Решений, Случайный Лес, Градиентный Бустинг и SVM. Для каждой модели проведен подбор гиперпараметров с использованием GridSearchCV. Результаты моделей до и после оптимизации сравниваются по ключевым метрикам: Accuracy, Precision, Recall, F1-score и ROC AUC. Также реализована визуализация сравнения метрик, анализ важности признаков для лучшей модели (Случайный Лес) и построение матрицы ошибок.

**Постановка задачи**

В рамках данной работы рассматривается задача предсказания выживаемости пассажиров легендарного лайнера «Титаник» на основе исторических данных. Исходный датасет, содержащий информацию о пассажирах, был загружен из открытого источника. Каждый объект в выборке описывается рядом признаков, включая класс обслуживания (Pclass), пол (Sex), возраст (Age), количество братьев/сестер и супругов (SibSp), количество родителей и детей (Parch), стоимость билета (Fare), пункт отправления (Embarked), а также уникальные идентификаторы (PassengerId, Name, Ticket, Cabin). Целевая переменная — Survived, принимающая значение 1 (пассажир выжил) или 0 (не выжил), что формализует задачу как бинарную классификацию.Для решения задачи выполнен полный цикл работ по анализу данных и построению моделей:

* Проведен разведочный анализ данных (EDA), включающий проверку размеров датасета, просмотр первых строк, анализ типов данных и статистическое описание. Выполнен подробный анализ пропущенных значений с их визуализацией (тепловая карта), а также анализ распределения целевой переменной. Исследованы распределения и взаимосвязи числовых признаков (Age, Fare, SibSp, Parch) и категориальных признаков (Pclass, Sex, Embarked) с использованием гистограмм, корреляционной матрицы, countplot'ов и boxplot'ов.
* Выполнена предварительная обработка данных:
* Пропущенные значения в признаке Age заполнены медианными значениями, рассчитанными индивидуально для каждой комбинации Sex и Pclass.
* Пропущенные значения в признаке Cabin заменены на 'Unknown'.
* Пропущенные значения в признаке Embarked заполнены наиболее часто встречающимся значением (модой).
* Созданы дополнительные признаки: Title (извлеченный из Name), FamilySize (сумма SibSp и Parch + 1) и IsAlone (бинарный признак, указывающий, путешествует ли пассажир один).
* Категориальные признаки (Sex, Embarked, Title) закодированы числовыми метками с использованием LabelEncoder.
* Выбран набор признаков (Pclass, Sex, Age, SibSp, Parch, Fare, Embarked, Title, FamilySize, IsAlone) для построения моделей.
* Выборка разделена на обучающую (80%) и тестовую (20%) подвыборки с использованием train\_test\_split и random\_state=42.
* Признаки масштабированы с использованием StandardScaler, обученного на обучающей выборке и примененного к обеим выборкам.
* Построено пять моделей машинного обучения: Логистическая Регрессия (LogisticRegression), Дерево Решений (DecisionTreeClassifier), Случайный Лес (RandomForestClassifier), Градиентный Бустинг (GradientBoostingClassifier) и Метод опорных векторов (SVC).
* Проведена оценка качества базовых версий моделей по метрикам: Accuracy, Precision, Recall, F1-score и ROC AUC.
* Выполнена настройка гиперпараметров для всех пяти моделей с использованием GridSearchCV и кросс-валидации (cv=5), с целевой метрикой accuracy.
* Проведено сравнение результатов моделей до и после оптимизации гиперпараметров, а также визуализация этого сравнения для ключевых метрик (Accuracy, F1-score, ROC AUC) с помощью столбчатых диаграмм. Проанализирована важность признаков для лучшей модели (Случайный Лес после оптимизации) и построена матрица ошибок для этой модели.

Реализация всех этапов осуществлена с использованием инструментов Python и библиотек: pandas для обработки данных, matplotlib и seaborn для визуализации, а также sklearn для предобработки, разделения данных, построения моделей, подбора гиперпараметров и оценки качества.

**Подбор и подготовка данных**

В качестве исходных данных используется классический набор Titanic, загруженный непосредственно по URL. Этот датасет содержит информацию о пассажирах, включая как числовые, так и категориальные признаки. Основные переменные, представленные в данных, включают PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin и Embarked.Этапы подготовки данных для анализа и построения моделей:

1. Обработка пропущенных значений:

* В признаке Age пропущенные значения были заполнены медианным возрастом, рассчитанным индивидуально для каждой комбинации пола (Sex) и класса обслуживания (Pclass). Такой подход позволяет учесть возможную зависимость возраста от этих характеристик.
* В признаке Embarked пропущенные значения были заполнены наиболее часто встречающимся значением (модой), определенной по всему датасету.
* В признаке Cabin, содержащем значительное количество пропусков, они были заменены специальной строковой меткой 'Unknown' для сохранения информации о наличии/отсутствии данных.

1. Создание дополнительных признаков:

* Из признака Name был извлечен титул пассажира (Title).
* На основе признаков SibSp (количество братьев/сестер и супругов) и Parch (количество родителей и детей) был создан признак размера семьи (FamilySize = SibSp + Parch + 1).
* На основе размера семьи был создан бинарный признак IsAlone, указывающий, путешествует ли пассажир в одиночку ( FamilySize == 1).

1. Кодирование категориальных признаков:

* Признаки Sex, Embarked и новосозданный признак Title были преобразованы в числовой формат с использованием LabelEncoder.

1. Отбор признаков:

* Для построения моделей был выбран следующий набор признаков: Pclass, Sex, Age, SibSp, Parch, Fare, Embarked, Title, FamilySize, IsAlone. Таким образом, признаки PassengerId, Name, Ticket и Cabin в их исходном виде не использовались в качестве входных данных для моделей.

1. Разделение данных:

* Подготовленный набор данных был разделен на обучающую и тестовую выборки в соотношении 80/20 для последующего обучения и независимой оценки моделей.

1. Масштабирование признаков:

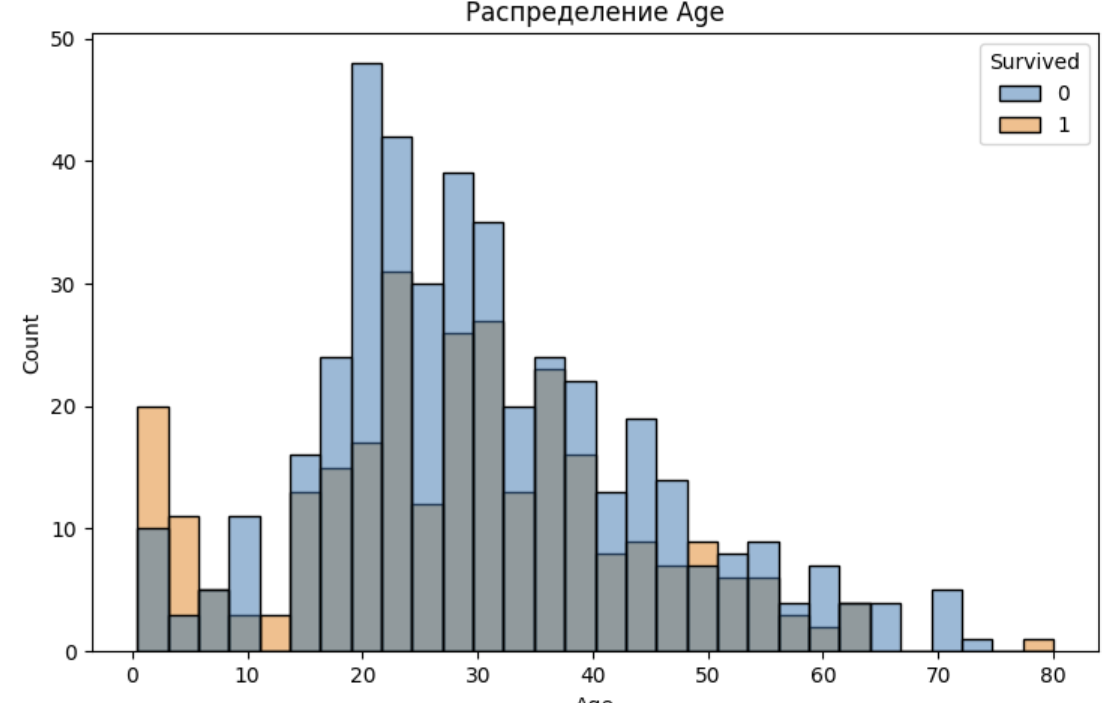
* Для числовых признаков в отобранном наборе данных было выполнено масштабирование с использованием StandardScaler. Этот шаг необходим для алгоритмов, чувствительных к масштабу признаков (например, SVM, Логистическая регрессия), и был применен отдельно к обучающей и тестовой выборкам (обучение на обучающей, трансформация обеих).

Все преобразования были реализованы с использованием библиотек pandas для манипуляций с данными, numpy для числовых операций, matplotlib и seaborn для визуализации на этапе EDA, и scikit-learn для предобработки данных, разделения выборки и масштабирования. На выходе получен очищенный и подготовленный набор данных, готовый к обучению моделей.

**Исследовательский анализ данных (EDA)**

Разведочный анализ данных (EDA) являлся ключевым этапом для понимания структуры датасета, выявления распределений признаков, обнаружения пропущенных значений и формирования гипотез относительно факторов, влияющих на выживаемость. Этот анализ позволил принять обоснованные решения на последующих этапах обработки и отбора признаков.

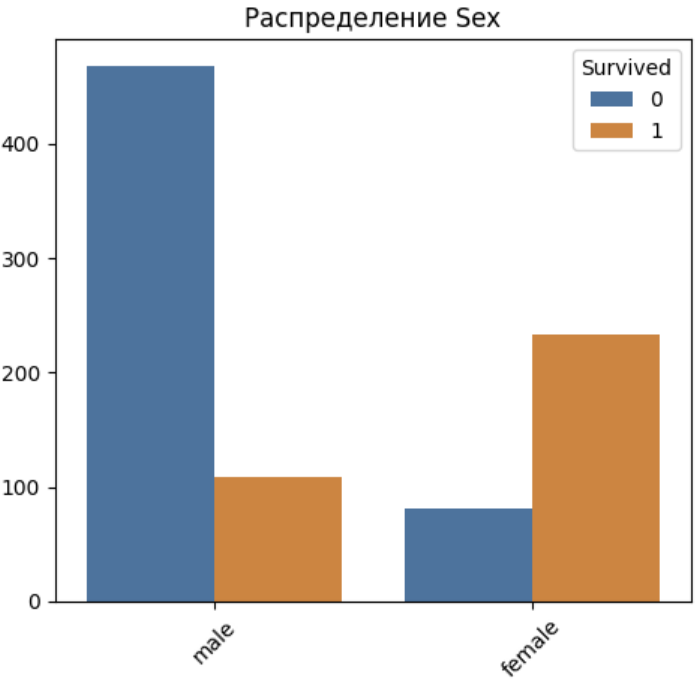
Для начала была изучена структура переменной **Age (возраст)**.



**Рисунок 1 – Распределение возраста пассажиров**

На графике видно, что наибольшая доля пассажиров находится в возрасте 20–35 лет. Этот интервал следует рассматривать как основной при оценке выживаемости.

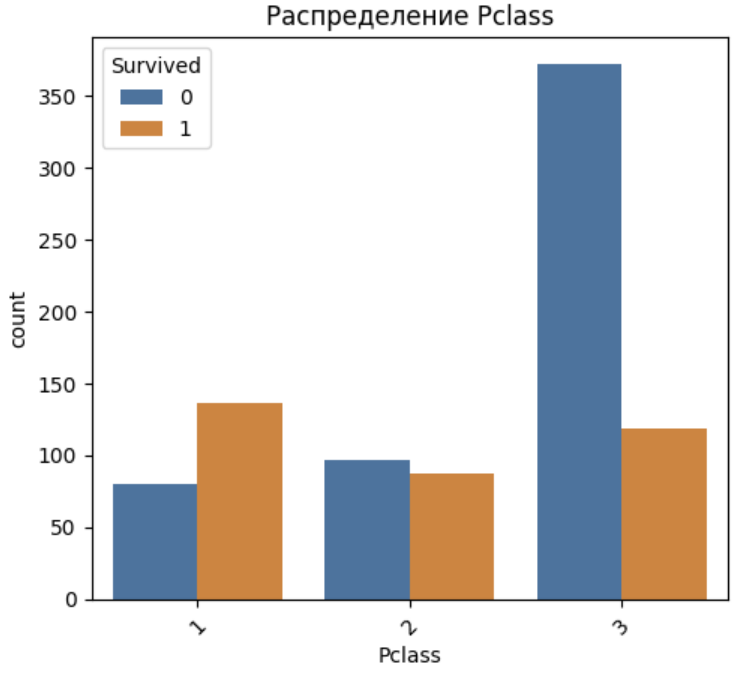
Далее был проанализирован пол пассажира как фактор выживания.



**Рисунок 2 – Выживаемость в зависимости от пола пассажира**

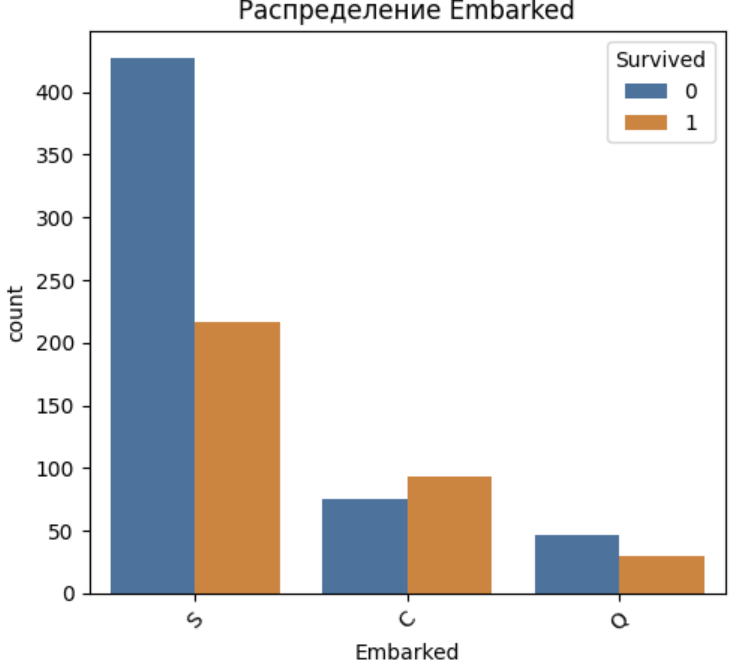
Очевидно, что женщины имели значительно больший шанс на выживание, что соответствует историческим данным: при эвакуации приоритет отдавался женщинам и детям.

Был построен график зависимости выживания от класса билета (**Pclass**). Он продемонстрировал, что зависимость выживаемости напрямую связана с классом.



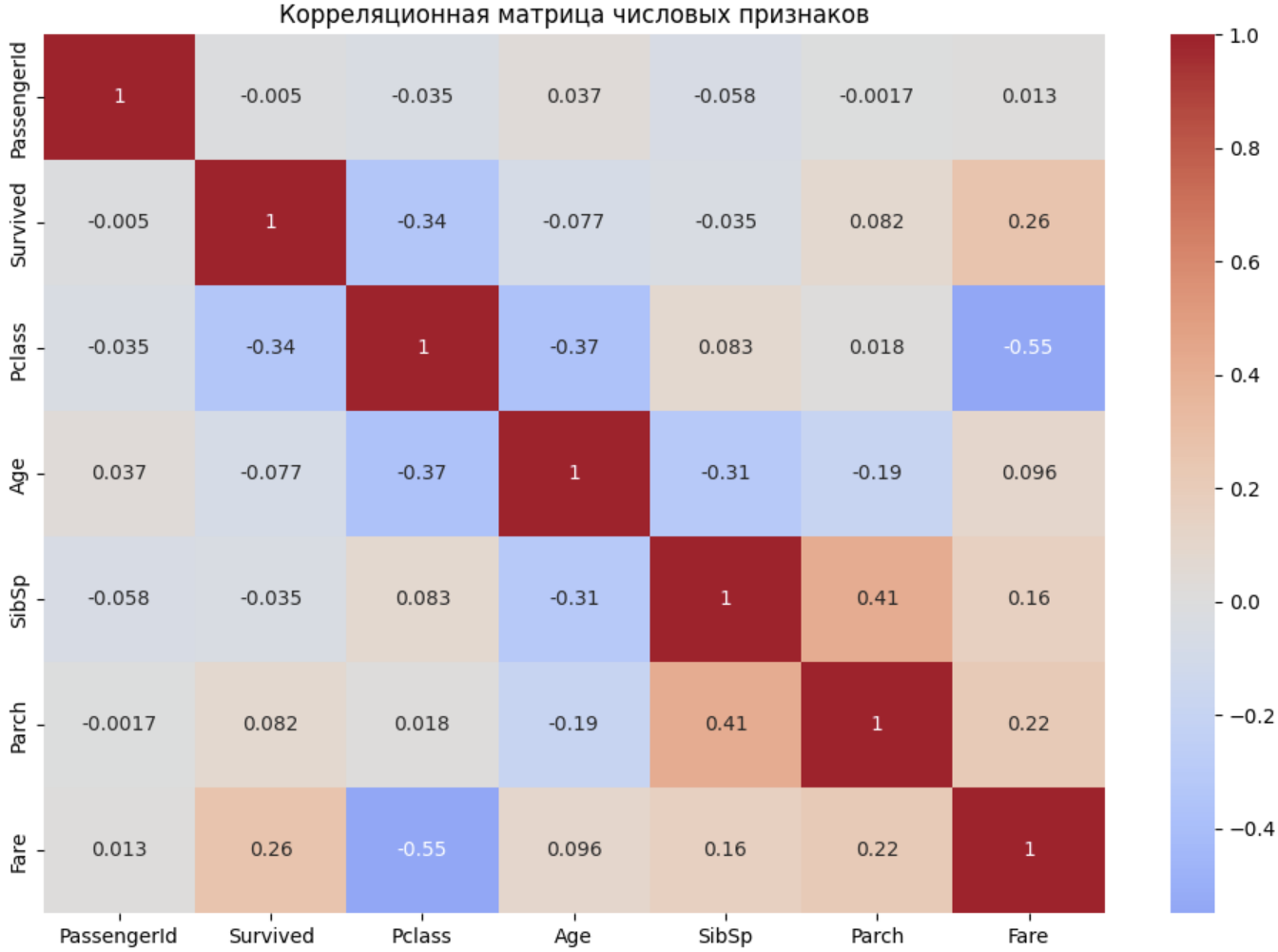
**Рисунок 3 – Зависимость выживания от класса билета**

Был построен график зависимости выживания от порта посадки (**Embarked**). Он продемонстрировал, что зависимость выживаемости также была связана и с портом посадки.



**Рисунок 4 – Зависимость выживания от порта посадки**

Для всестороннего анализа взаимосвязей между всеми числовыми признаками датасета, включая целевую переменную, была построена тепловая карта корреляций.



**Рисунок 5 – Корреляционная матрица признаков**

**Сильные корреляции (абсолютное значение > 0.7):**

* Между представленными на графике исходными числовыми признаками сильных корреляций не наблюдается.

**Умеренные корреляции (абсолютное значение от 0.3 до 0.7):**

* Pclass и Fare: -0.55 (Умеренная отрицательная) - Чем выше класс обслуживания (меньше числовое значение Pclass), тем выше стоимость билета.
* Pclass и Age: -0.37 (Умеренная отрицательная) - Пассажиры в более высоких классах обслуживания в среднем несколько старше.
* Pclass и Survived: -0.34 (Умеренная отрицательная) - Чем выше класс обслуживания, тем выше вероятность выживания (Survived=1).
* Age и SibSp: -0.31 (Умеренная отрицательная) - С возрастом количество братьев/сестер/супругов на борту имеет тенденцию к уменьшению (меньше вероятность путешествовать с большим количеством SibSp).
* SibSp и Parch: 0.41 (Умеренная положительная) - Пассажиры, путешествующие с братьями/сестрами/супругами, чаще также путешествуют с родителями/детьми.

**Слабые корреляции (абсолютное значение < 0.3):**

* Survived и Fare: 0.26 (Слабая положительная) - Некоторая слабая тенденция: более высокая стоимость билета немного связана с большей вероятностью выживания.
* Parch и Fare: 0.22 (Слабая положительная).
* SibSp и Fare: 0.16 (Слабая положительная).
* Age и Fare: 0.096 (Слабая положительная).
* Parch и Survived: 0.082 (Слабая положительная) - Количество родителей/детей на борту очень слабо связано с выживаемостью в этой матрице.
* Все корреляции с Passengerid очень близки к нулю, как и ожидалось, поскольку это просто идентификатор.
* Большинство остальных пар признаков также имеют слабые или очень слабые корреляции (значения близкие к нулю).

**Обработка и преобразование признаков**

Для повышения информативности моделей и подготовки данных к использованию алгоритмами машинного обучения, требующими числового формата и масштабирования, были выполнены следующие шаги по обработке и преобразованию признаков:

1. **Инженерия признаков:**

* На основе признаков SibSp и Parch был создан новый признак **FamilySize**, представляющий суммарное количество членов семьи пассажира на борту (включая самого пассажира).
* Создан бинарный признак **IsAlone**, который принимает значение 1, если пассажир путешествует в одиночку (FamilySize == 1), и 0 в противном случае.
* Из признака Name был извлечен **титул** пассажира (Title), отражающий социальный статус или обращение.

1. **Кодирование категориальных признаков:**

* Категориальные признаки Sex, Embarked и новосозданный признак Title были преобразованы в числовой формат с использованием **LabelEncoder**.

1. **Отбор признаков:**

* Для построения моделей был отобран следующий набор признаков, включающий как исходные, так и созданные: Pclass, Sex, Age, SibSp, Parch, Fare, Embarked, Title, FamilySize, IsAlone.

1. **Масштабирование признаков:**

* Весь набор отобранных признаков (X) был стандартизирован с использованием **StandardScaler**. Этот шаг был применен ко всем признакам в наборе (как исходным числовым, так и закодированным/производным), что важно для алгоритмов, чувствительных к диапазону значений признаков, таких как Логистическая Регрессия и SVM. Масштабирование выполнялось отдельно для обучающей и тестовой выборок (обучение скалера на тренировочных данных и применение трансформации к обеим выборкам).

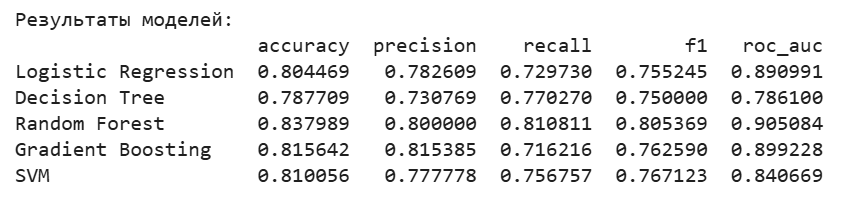
Все эти преобразования были реализованы с использованием соответствующих функций из библиотек pandas и scikit-learn, что позволило получить подготовленный набор данных, готовый для этапа обучения моделей.

**Построение и сравнение моделей**

Были построены и обучены следующие модели:

1. Логистическая регрессия (LogisticRegression)
2. Решающее дерево (DecisionTreeClassifier)
3. Случайный лес (RandomForestClassifier)
4. Градиентный бустинг (GradientBoostingClassifier)
5. Метод опорных векторов (SVC)

Все модели были обучены на одной и той же выборке (80% обучающая, 20% тестовая), и поначалу использовались базовые параметры без настройки.

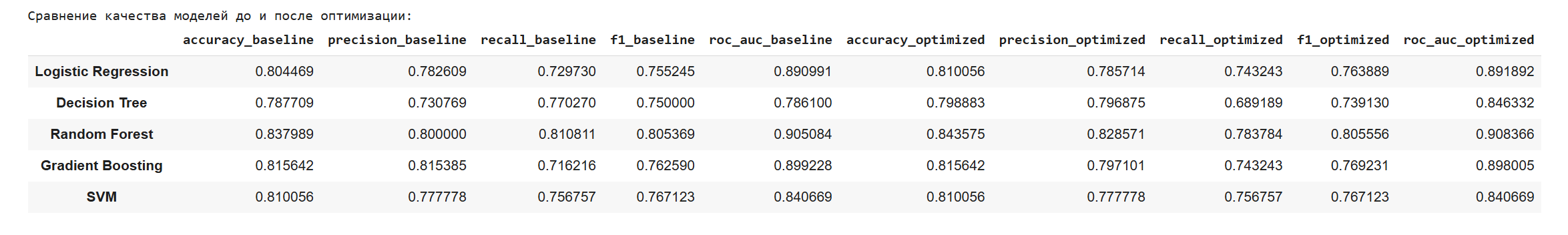


**Рисунок 6 – Результаты базовых моделей**

**Настройка гиперпараметров**

В целом, оптимизация гиперпараметров привела к улучшению большинства метрик для большинства моделей, хотя степень улучшения варьируется.

**Изменения по моделям:**

* **Logistic Regression:**
* accuracy: Улучшилась с 0.8045 до 0.8101 (рост на ~0.0056).
* precision: Улучшилась с 0.7826 до 0.7857 (рост на ~0.0031).
* recall: Улучшилась с 0.7297 до 0.7432 (рост на ~0.0135).
* f1: Улучшилась с 0.7552 до 0.7639 (рост на ~0.0087).
* roc\_auc: Улучшилась незначительно с 0.8910 до 0.8919 (рост на ~0.0009).
* *Вывод:* Логистическая регрессия показала небольшое, но последовательное улучшение по всем метрикам после оптимизации.
* **Decision Tree:**
* accuracy: Улучшилась с 0.7877 до 0.7989 (рост на ~0.0112).
* precision: Значительно улучшилась с 0.7308 до 0.7969 (рост на ~0.0661).
* recall: Ухудшилась с 0.7703 до 0.6892 (снижение на ~0.0811).
* f1: Улучшилась с 0.7500 до 0.7391 (незначительное снижение на ~0.0109, несмотря на рост точности).
* roc\_auc: Значительно улучшилась с 0.7861 до 0.8463 (рост на ~0.0602).
* *Вывод:* Оптимизация дерева решений привела к заметному росту Precision и ROC AUC, а также небольшому росту Accuracy, но ценой снижения Recall.
* **Random Forest:**
* accuracy: Улучшилась с 0.8380 до 0.8436 (рост на ~0.0056).
* precision: Улучшилась с 0.8000 до 0.8286 (рост на ~0.0286).
* recall: Ухудшилась с 0.8108 до 0.7838 (снижение на ~0.0270).
* f1: Практически не изменилась (с 0.8054 до 0.8056, рост на ~0.0002).
* roc\_auc: Улучшилась с 0.9051 до 0.9084 (рост на ~0.0033).
* *Вывод:* Random Forest показал небольшое улучшение Accuracy и ROC AUC, заметный рост Precision, но небольшое снижение Recall, что привело к минимальному изменению F1-меры.
* **Gradient Boosting:**
* accuracy: Не изменилась (0.8156).
* precision: Незначительно снизилась с 0.8154 до 0.7971 (снижение на ~0.0183).
* recall: Улучшилась с 0.7162 до 0.7432 (рост на ~0.0270).
* f1: Улучшилась с 0.7626 до 0.7692 (рост на ~0.0066).
* roc\_auc: Незначительно снизилась с 0.8992 до 0.8980 (снижение на ~0.0012).
* *Вывод:* Градиентный бустинг показал улучшение Recall и F1-меры, но незначительное снижение Precision и ROC AUC при неизменной Accuracy.
* **SVM:**
* По всем метрикам (accuracy, precision, recall, f1, roc\_auc) значения **не изменились** после оптимизации гиперпараметров.
* *Вывод:* В данном случае подбор гиперпараметров не привел к изменению метрик качества модели SVM на тестовой выборке.

**Рисунок 7 – Сравнение моделей**

Наиболее заметный рост метрик, в частности Precision и ROC AUC, продемонстрировало **Дерево Решений**. **Логистическая регрессия** показала стабильный небольшой рост по всем метрикам. **Random Forest** улучшил Accuracy, Precision и ROC AUC, но немного потерял в Recall. **Градиентный Бустинг** улучшил Recall и F1-меру. **SVM** не показал изменений после оптимизации. В целом, процесс оптимизации гиперпараметров оказался полезным для большинства моделей, позволив улучшить их производительность по тем или иным метрикам, что подчеркивает важность этого этапа в цикле машинного обучения.

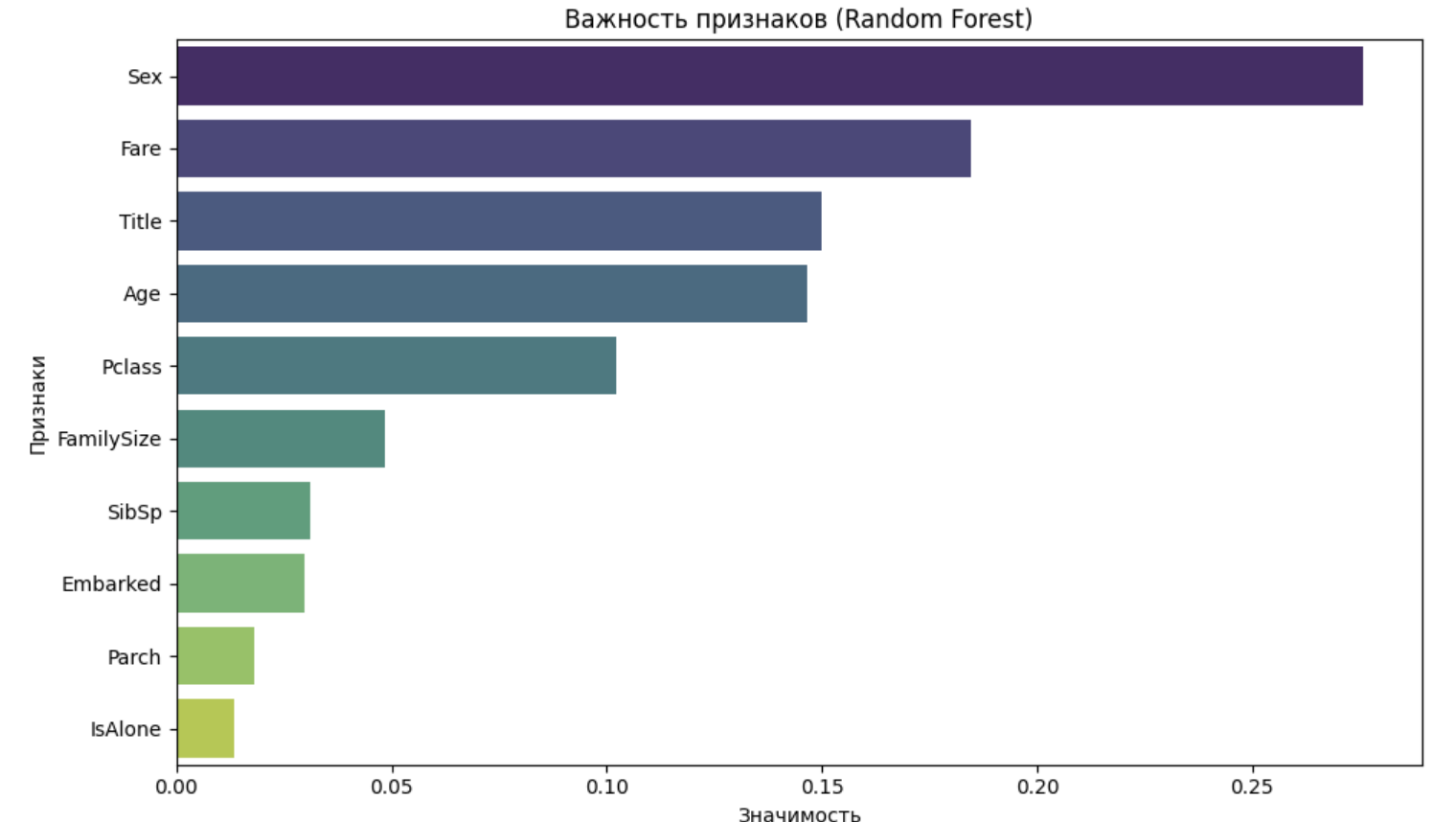
**Заключение**

В рамках настоящей работы был успешно выполнен полный цикл решения задачи бинарной классификации — предсказания выживаемости пассажиров «Титаника». Проведенный анализ данных и построение моделей подтвердили важность системного подхода, включающего несколько ключевых этапов:

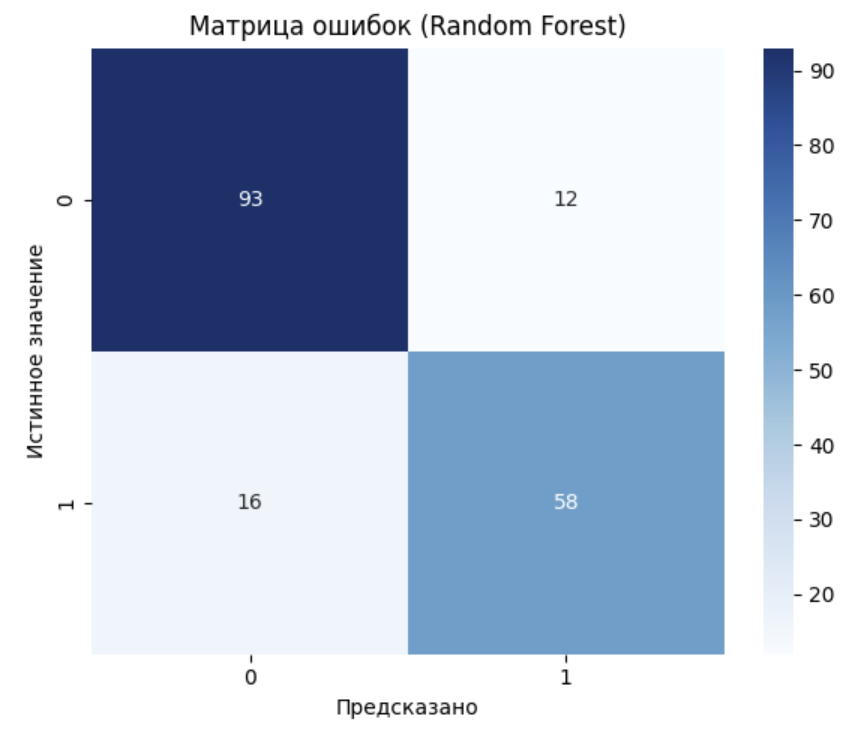
1. **Разведочный анализ данных (EDA):** Детальное изучение датасета позволило выявить структуру данных, распределения признаков, наличие пропусков и, что особенно важно, ключевые факторы, влияющие на выживаемость (такие как пол, класс обслуживания и возраст). Визуализации наглядно продемонстрировали эти зависимости.
2. **Предварительная обработка и инженерия признаков:** Этап подготовки данных включил эффективную обработку пропущенных значений (с учетом контекста признаков Age, Embarked, Cabin), создание новых информативных признаков (Title, FamilySize, IsAlone) и корректное кодирование категориальных переменных (Sex, Embarked, Title) с использованием LabelEncoder.
3. **Масштабирование признаков:** Применение StandardScaler ко всему набору признаков перед обучением моделей обеспечило их правильную работу, особенно для алгоритмов, чувствительных к масштабу данных (например, Логистическая Регрессия, SVM).
4. **Построение, сравнение и оптимизация моделей:** Были построены и оценены пять различных моделей машинного обучения. Сравнение их производительности на независимой тестовой выборке по ключевым метрикам (Accuracy, Precision, Recall, F1-score, ROC AUC) позволило выявить наиболее перспективные. Дальнейшая настройка гиперпараметров с использованием GridSearchCV продемонстрировала потенциал для улучшения качества моделей, что было подтверждено сравнительным анализом метрик до и после оптимизации.

* Как показал сравнительный анализ, оптимизация привела к улучшению метрик у большинства моделей (Логистическая Регрессия, Дерево Решений, Случайный Лес, Градиентный Бустинг), хотя SVM в данном случае не показал значимых изменений.

**Наилучшая модель:** По совокупности метрик после оптимизации, модель **Random Forest Classifier** продемонстрировала высокую производительность. Она показала лучшую метрику **Accuracy (0.8436)** и **ROC AUC (0.9084)** среди всех моделей. Метрика **F1-score составила 0.8056**, что является сбалансированным показателем, отражающим хорошее сочетание Precision (0.8286) и Recall (0.7838). Эти результаты делают Random Forest наиболее подходящей моделью для решения задачи предсказания выживаемости на данном наборе данных.



**Рисунок 8 – Важность признаков (Random Forest)**

****

**Рисунок 9 – Матрица ошибок (Random Forest)**

**Список использованных источников**

1.Kaggle: Titanic - Machine Learning from Disaster (https://www.kaggle.com/competitions/titanic)

2.Géron, Aurélien. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow — O'Reilly, 2019.

3.Документация Scikit-learn — https://scikit-learn.org/

4.Материалы курса "Машинное обучение", OpenAI, Stepik, Coursera

5.Python Software Foundation — https://www.python.org/

6.Визуализация и EDA: https://seaborn.pydata.org/, https://matplotlib.org/