Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет
имени первого Президента России Б. Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ
Школа бакалавриата

ОТЧЕТ

ПО ПРОЕКТНОЙ РАБОТЕ:

«Разработка образовательных материалов и проектов в сфере Data Science»

Дисциплина: «Проектный практикум»

Куратор: Ильинский А. Д. _____

Студент команды «Фермер»

Баталов К. Д. _____

Екатеринбург, 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Работа в рамках образовательного трека по Data Science представляет собой выполнение четырех лабораторных работ и двух итоговых заданий, направленных на формирование ключевых практических навыков в этой области.

Цель: освоение инструментов и методов Data Science через решение прикладных задач.

Задачи:

– изучение теоретических основ Data Science;

– получение практического навыка работы с функциями (лабораторная работа №1);

– получение практического навыка корреляционного анализа (лабораторная работа №2);

– получение практического навыка работы с линейными моделями (лабораторная работа №3);

– получение практического навыка работы с ансамблями и полносвязными нейронными сетями (лабораторная работа №4);

– создание предсказывающей модели-классификатора (итоговое задание №1);

– создание предсказывающей модели-регрессора (итоговое задание №2).

Актуальность обусловлена растущим спросом на специалистов в области Data Science. По данным Всемирного экономического форума сфера Data Science является одной из самых востребованных у работодателей [1]. Это делает критически важным получение не только теоретических знаний, но и практического опыта работы с реальными данными и алгоритмами.

Выполненные работы в рамках образовательного трека позволят обрести базовые практические навыки и создадут необходимый фундамент для дальнейшего развития в сфере.

Ожидаемые результаты:

– успешное выполнение всех лабораторных работ с применением изученных методов;

– создание рабочих моделей классификации и регрессии, демонстрирующих понимание ключевых этапов Data Science;

– формирование базы для дальнейшего углубленного изучения машинного обучения и анализа данных.

1 Выполнение лабораторных работ

1.1 Выполнение лабораторной работы №1

В рамках данной лабораторной работы были выполнены практические задания по освоению ключевых методов обработки данных с использованием библиотеки NumPy и базовых алгоритмов Python. Работа проводилась в два этапа: первый был посвящен матричным операциям с применением NumPy, второй - реализации фундаментальных алгоритмов обработки данных на чистом Python [2].

Основное внимание в первой части работы уделялось освоению возможностей библиотеки NumPy для эффективной работы с многомерными массивами. Была разработана функция `sum_prod(X, V)`, вычисляющая сумму произведений матриц на векторы. Особенностью реализации стало использование векторизованных операций через `np.dot()`, что обеспечило высокую производительность вычислений. Функция включает проверку корректности входных данных и протестирована для матриц различных размерностей (2×2, 3×3, 4×4).

Важным аспектом работы стала реализация функции бинаризации матрицы `binarize(M, threshold)`, которая преобразует исходную матрицу в бинарную форму относительно заданного порога. Алгоритм предусматривает обработку различных случаев, включая отрицательные и нулевые значения, и использует оптимизированные операции NumPy. Также были созданы функции `unique_rows()` и `unique_columns()` для поиска уникальных строк и столбцов матрицы соответственно, с применением эффективных методов библиотеки NumPy.

Во второй части работы реализован набор базовых алгоритмов для обработки данных. Среди них:

a) функция `vowel_counter()` для подсчета гласных букв в строке с учетом регистра;

b) алгоритм проверки уникальности символов `uniq_string()` с использованием множеств;

c) эффективная реализация подсчета единичных битов `count_bits()` через побитовые операции;

d) функция `magic()` для итеративного перемножения цифр числа;

e) вычисление среднеквадратического отклонения `mse()`;

f) алгоритм разложения на простые множители `prime_factors()` с форматированным выводом.

Все разработанные функции сопровождались комплексом модульных тестов, обеспечивающих проверку:

– стандартных случаев использования;

– граничных условий (пустые входные данные, минимальные/максимальные значения);

– некорректных входных данных.

Тестирование проводилось с использованием pytest, что позволило автоматизировать процесс проверки и гарантировать надежность реализаций.

В результате выполнения работы были получены или закреплены следующие практические навыки:

− работа с многомерными массивами в NumPy;

− реализация базовых алгоритмов обработки данных;

− написание модульных тестов;

− обработка различных типов входных данных.

1.2 Выполнение лабораторной работы №2

В ходе выполнения данной лабораторной работы были освоены ключевые методы обработки и анализа данных с использованием библиотеки Pandas,

а также проведен комплексный корреляционный анализ. Работа состояла из двух взаимосвязанных частей, каждая из которых была направлена на формирование конкретных практических навыков в области Data Science [3].

Первая часть работы была посвящена изучению функциональных возможностей библиотеки Pandas на примере анализа датасета "Titanic". Основное внимание уделялось следующим аспектам:

1) загрузка и первичный анализ данных:

a) освоена методика чтения CSV-файлов с помощью `pd.read_csv()`;

b) проведен предварительный анализ структуры данных через методы `head()`, `info()` и `describe()`;

c) изучены типы данных и общие характеристики набора данных.

2) обработка пропущенных значений:

a) реализована комплексная проверка наличия пропусков `isna().sum()`;

b) отработаны различные стратегии обработки пропущенных данных - заполнение медианными/средними значениями (`fillna()`), удаление некритичных строк с пропусками (`dropna()`);

c) проанализированы последствия применения разных подходов к очистке данных.

3) манипуляции с DataFrame:

a) освоены методы выборки данных (индексация, loc/iloc);

b) реализована фильтрация по сложным условиям;

c) проведена сортировка данных по нескольким критериям (`sort_values()`);

d) выполнена группировка и агрегация данных (`groupby()` с различными функциями агрегации).

Вторая часть работы была посвящена изучению методов корреляционного анализа на примере набора данных "Brain Size". В рамках этой части можно выделить следующие аспекты:

1) подготовка данных:

a) освоена загрузка данных с нестандартными разделителями (табуляция);

b) проведена предварительная обработка заголовков и структуры данных;

c) выполнена проверка целостности и согласованности данных.

2) корреляционный анализ:

a) рассчитана полная корреляционная матрица методом Пирсона;

b) проведен детальный анализ полученных корреляций: интерпретация диагональных значений (1.0), объяснение симметричности матрицы, выявление значимых взаимосвязей между переменными;

c) визуализированы наиболее интересные корреляционные пары.

3) интерпретация результатов:

a) сформулированы содержательные выводы о взаимосвязях между переменными;

b) оценена статистическая значимость обнаруженных корреляций;

c) определены направления для дальнейшего анализа.

В ходе выполнения работы были получены следующие результаты:

− практические навыки работы с реальными наборами данных;

− навыки работы с методами обработки и очистки данных;

− понимание принципов корреляционного анализа;

− опыт содержательной интерпретации результатов анализа.

1.3 Выполнение лабораторной работы №3

В ходе выполнения данной лабораторной работы были освоены основные принципы построения и обучения линейных моделей машинного обучения для задач регрессии и бинарной классификации с использованием градиентного спуска [4].

Первая часть работы была посвящена созданию базовой архитектуры линейной модели, реализации моделей регрессии и классификации, а также обучению разработанных моделей и проверке их работоспособности на реальных данных. Основное внимание уделялось следующим аспектам:

1) построение базовой линейной модели:

a) разработан универсальный класс для линейных моделей, обеспечивающий основу для последующей специализации;

b) реализована инициализация параметров модели: весов и смещения;

c) добавлен метод линейного предсказания на основе входных признаков.

2) реализация линейной регрессии:

a) создан подкласс, наследующий базовую модель и использующий в качестве функции потерь среднеквадратичную ошибку (MSE);

b) реализован алгоритм градиентного спуска для оптимизации параметров модели;

c) добавлены методы обучения и получения предсказаний.

3) реализация линейной классификации:

a) создан подкласс линейной модели для решения задачи бинарной классификации;

b) в качестве функции потерь использована бинарная кросс-энтропия;

c) добавлена сигмоидная функция для перевода линейного выхода в вероятность;

d) реализованы методы обучения и предсказания классов.

4) подготовка и предобработка данных:

a) использованы два набора данных: Student Performance Dataset (для регрессии) и German Credit Data (для классификации);

b) выполнена нормализация признаков и целевой переменной (в случае регрессии);

c) осуществлено разделение выборок на обучающую и тестовую части для корректной оценки качества моделей.

5) обучение моделей и анализ процесса:

a) реализован итеративный процесс обучения с помощью градиентного спуска;

b) произведено отслеживание функции потерь на протяжении эпох обучения;

c) выполнена визуализация процесса обучения с помощью графиков, демонстрирующих динамику снижения ошибки.

Вторая часть работы была посвящена экспериментам с базовыми моделями с целью увеличения их точности, главным образом за счет смены гиперпараметров. Удалось получить результат точности ROC-AUC = 0.79 для логистической регрессии с помощью подбора оптимальных гиперпараметров.

В ходе выполнения лабораторной работы были получены следующие результаты:

− приобретены практические навыки реализации базовых линейных моделей машинного обучения на языке программирования Python;

− освоены методы градиентной оптимизации и работа с функциями потерь;

− получен опыт предобработки данных и визуального анализа обучения моделей;

− выявлены направления дальнейшего улучшения моделей, включая подбор гиперпараметров, расширение признакового пространства и оценку качества на кросс-валидации.

1.4 Выполнение лабораторной работы №4

В ходе выполнения данной лабораторной работы была решена задача повышения качества классификации кредитоспособности клиентов на основе набора данных German Credit Data [5]. Основной целью являлось достижение как можно более высокого значения метрики ROC-AUC на тестовой выборке. Работа представляла собой практическое применение современных моделей машинного обучения и методов ансамблирования. Оценка качества выполнения задания напрямую зависела от достигнутого значения ROC-AUC на тестовых данных.

Работа включала в себя несколько последовательных этапов, направленных на построение, улучшение и комбинирование моделей классификации.

Первая часть работы была посвящена подготовке данных и построению базовых моделей. Основное внимание уделялось следующим аспектам:

1) подготовка и анализ данных:

a) загружен исходный датасет из файла german.csv;

b) данные были разделены на обучающую и тестовую выборки в пропорции 80/20;

c) выполнена визуализация распределения классов, позволившая оценить баланс выборки;

d) применено масштабирование признаков с использованием StandardScaler для повышения стабильности обучения моделей.

2) обучение базовых моделей классификации:

a) были построены три независимые модели: Random Forest Classifier Gradient Boosting Classifier, Multilayer Perceptron (MLP);

b) модели обучались с использованием стандартных параметров, что позволило зафиксировать их начальную эффективность.

Вторая часть работы была направлена на улучшение качества предсказаний за счёт оптимизации и ансамблирования. В рамках этого этапа были выполнены следующие действия:

1) оптимизация гиперпараметров моделей:

a) для Random Forest увеличено количество деревьев и оптимизирована максимальная глубина;

b) для Gradient Boosting подобраны оптимальные значения learning_rate и количества итераций;

c) для MLP была изменена архитектура нейронной сети и параметры обучения, включая количество слоёв, количество нейронов и скорость обучения.

2) ансамблирование моделей:

a) Voting Ensemble (взвешенное голосование): использованы все три базовые модели, применено мягкое голосование (soft voting), учитывающее вероятности; заданы веса моделей (2:2:1), отражающие их относительную точность;

b) Stacking Ensemble: использованы те же базовые модели, предсказывающие на первом уровне; в качестве мета-классификатора использована Logistic Regression.

По итогу экспериментов наилучший показатель был получен при использовании Voting Ensemble, где значение ROC-AUC составило 0.7952.

В ходе выполнения лабораторной работы были получены следующие результаты:

– приобретены навыки построения и настройки нескольких моделей классификации;

– освоены методы предварительной обработки и масштабирования данных;

– реализованы и протестированы методы ансамблирования, продемонстрировавшие своё преимущество перед одиночными моделями.

2 Выполнение итогового задания №1

Итоговое задание №1 представляло собой создание модели машинного обучения для определения выживаемости пассажира на Титанике [6]. Работа включала в себя EDA и ресерч-анализ. Были построены графики, применены инструменты EDA, проведен анализ каждого графика. Рассмотрена корреляция колонок с таргетом. Проводился Feature Engineering, после которого рассматривалась корреляция новых колонок с таргетом. Были созданы простые модели. Проведены эксперименты с моделями машинного обучения/глубокого обучения, по одной из каждого семейства - линейные, деревья, модификации градиентного бустинга , нейронные сети.

2.1 Исследовательский анализ (EDA)

Исследовательский анализ данных был необходим для комплексного понимания структуры данных. Первым делам была выведена основная информация о датасете, в которой сразу удалось выявить пропущенные значения в столбцах: Age (19.8%), Cabin (77.1%), Embarked (0.2%).

Произведено разделение на числовые и категориальные признаки, после чего для числовых признаков были построены гистограммы распределения (приложение А) и боксплоты (приложение Б). По графикам обнаружены выбросы в Fare, SibSp.

Была также построена тепловая карта корреляций всех числовых признаков (рисунок 1).

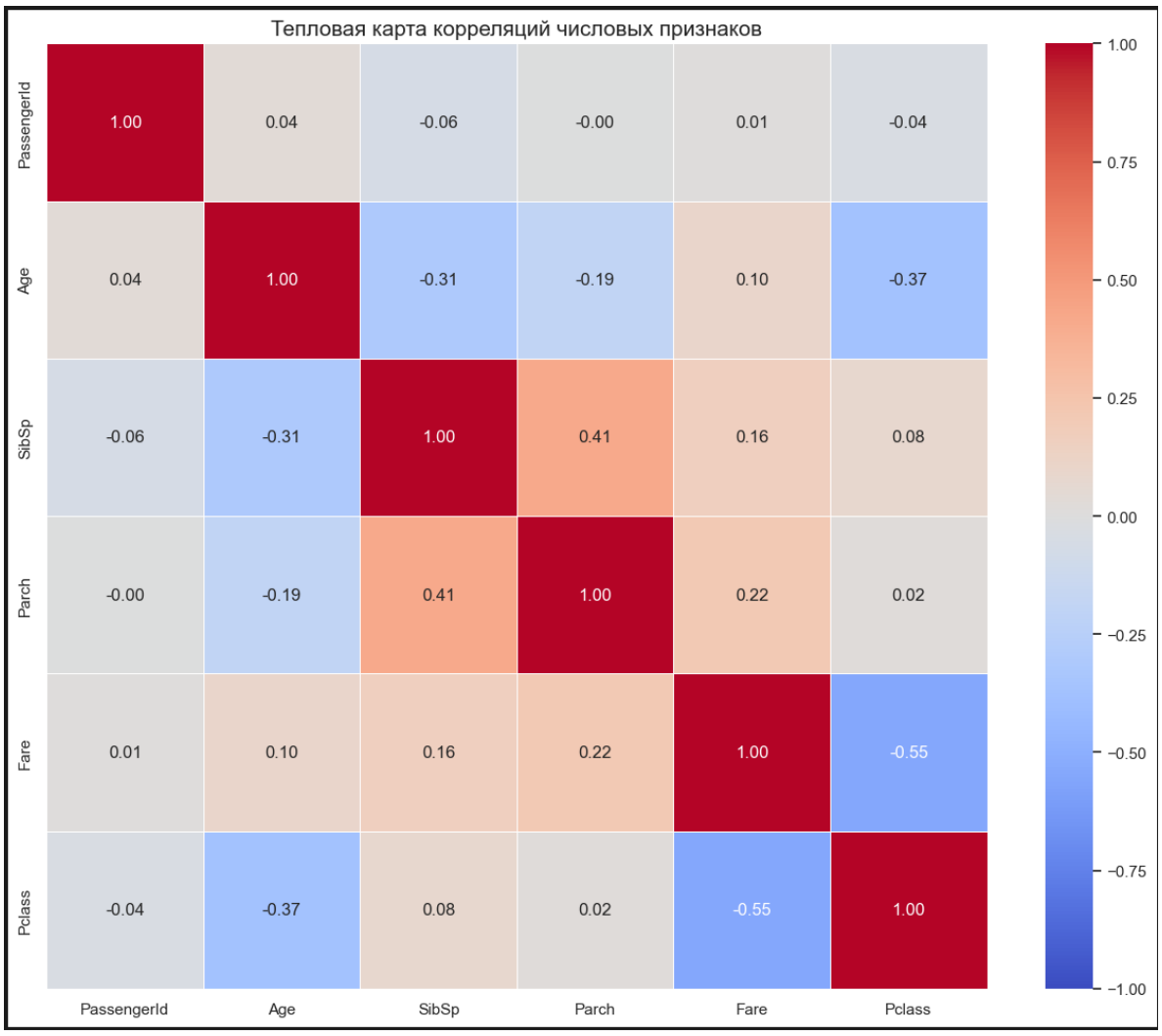


Рисунок 1 – Тепловая карта корреляций числовых признаков Titanic

На основе тепловой карты корреляций сделаны ключевые выводы:

– большинство коэффициентов корреляции находятся в диапазоне от -0.25 до 0.25;

– самая значимая положительная корреляция наблюдается между SibSp и Parch (0.41);

– самая значимая отрицательная корреляция наблюдается между Pclass и Fare (-0.55);

– отсутствие сильных корреляций с Fare: несмотря на интуитивные предположения, стоимость билета слабо коррелирует с другими числовыми признаками;

– наличие умеренных отрицательных корреляций нескольких переменных с Pclass.

Аналогичная тепловая карта была построена для оценки корреляций с таргетным признаком Survived (рисунок 2).

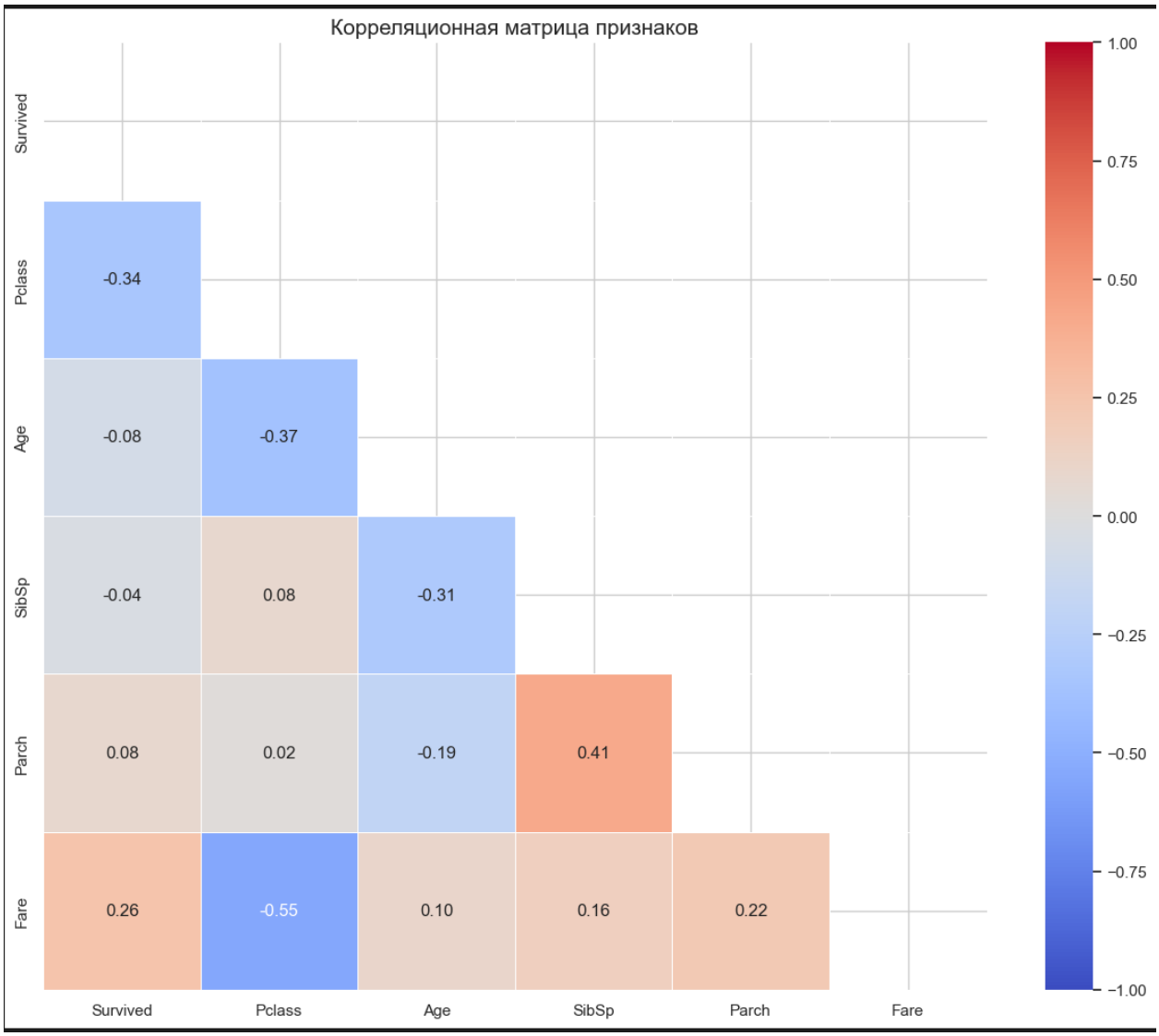


Рисунок 2 – Корреляции числовых признаков с Survived

Были сделаны выводы: умеренная отрицательная корреляция с классом билета (Pclass) (примерно -0.34) - пассажиры первого класса имели больше шансов выжить; слабая положительная корреляция с плата за проезд (Fare) (примерно 0.26) - более дорогие билеты связаны с большей выживаемостью.

2.2 Feature Engineering

В наборе данных Титаника были выполнены следующие шаги по созданию признаков для улучшения данных для моделирования:

1) обработка пропущенных значений:

a) Cabin (Каюта): был создан новый бинарный признак Has_Cabin, который указывает, есть ли у пассажира каюта, так как в столбце Cabin было более 50% пропущенных значений;

b) Age (Возраст): пропущенные значения в столбце Age были заполнены медианным возрастом пассажиров, сгруппированных по Pclass и Sex;

c) Embarked (Порт посадки): пропущенные значения в столбце Embarked были заполнены наиболее часто встречающимся портом посадки.

2) создание признаков:

FamilySize (Размер семьи): был создан новый признак FamilySize путем суммирования SibSp (братья/сестры и супруги на борту) и Parch (родители/дети на борту) и добавления 1 для учета самого пассажира;

3) преобразование категориальных признаков в числовые:

a) Sex (Пол): преобразован в числовые значения, где 'male' (мужчина) - 0, а 'female' (женщина) – 1;

b) Embarked (Порт посадки): преобразован в числовые значения с помощью маппинга: 'S' - 0, 'C' - 1, и 'Q' - 2.

Эти шаги были направлены на улучшение качества набора данных и его подготовку для использования в моделях машинного обучения. Была построена еще одна корреляционная матрица признаков (рисунок 3).

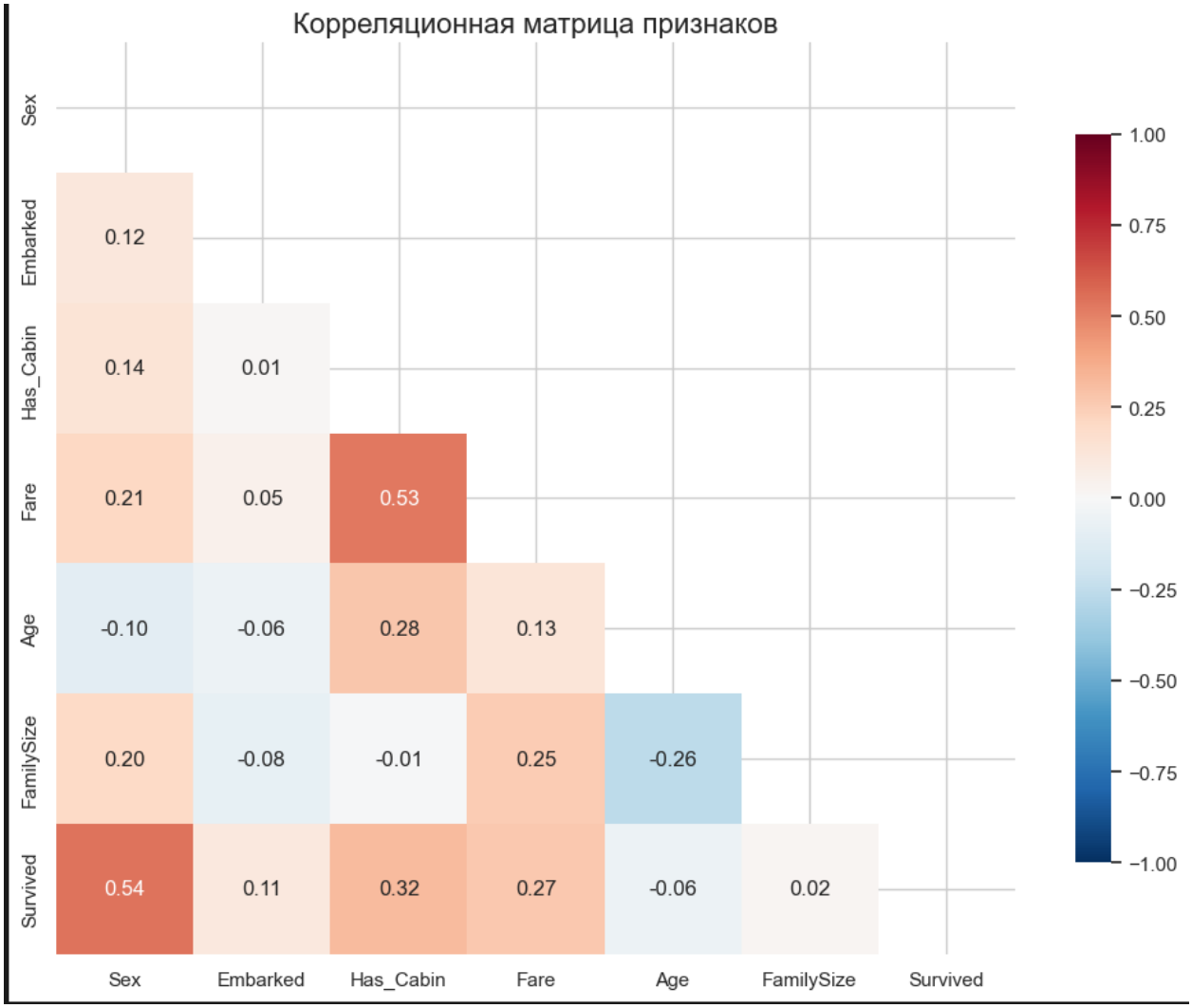


Рисунок 3 – Корреляционная матрица признаков Titanic

Здесь уже обнаружились значительные корреляции таргетного признака Survived с другими признаками.

2.3 Обучение моделей

В процессе обучения моделей на наборе данных Титаника были выполнены следующие шаги:

1) подготовка данных:

a) выделены признаки: Pclass, Sex, Age, Fare, Embarked, SibSp, Parch, Has_Cabin, FamilySize;

b) числовые признаки были масштабированы с использованием StandardScaler;

c) данные разделены на обучающую и тестовую выборки в соотношении 80/20.

2) обучение моделей:

a) использовались следующие модели: Логистическая регрессия, Дерево решений, Случайный лес, Градиентный бустинг, XGBoost, Нейронная сеть (MLP);

b) лучшая модель по тестовой выборке: Нейронная сеть с точностью accuracy = 0.8268.

3) оценка моделей:

a) нейронная сеть: точность на тестовой выборке 0.8268, средняя точность при кросс-валидации (5 фолдов) составила 0.8182 ± 0.0131.

b) ансамбли: VotingClassifier показал точность на тестовой выборке 0.8492, а StackingClassifier — 0.8324; при кросс-валидации VotingClassifier имел среднюю точность 0.8305 ± 0.0207, а StackingClassifier — 0.8316 ± 0.0258.

Лучшая классическая модель для предсказания на тестовой выборке - нейронная сеть MLP с результатом accuracy на test = 0.8268 и средней accuracy 0.8182 ± 0.0131 при кросс-валидации (5 фолдов).

Базовые модели (логистическая регрессия, деревья, случайный лес, бустинг, XGBoost) показали среднюю точность на тестовой выборке в диапазоне 0.78–0.82).

Кросс-валидация (5 фолдов) показала, что средняя accuracy чуть ниже, чем на одном test-сплите, что ожидаемо и говорит о том, что модель не переобучается под конкретное разбиение. Разброс accuracy по фолдам связан с небольшим размером датасета и неравномерным распределением классов.

Ансамбли (Voting, Stacking) дают ощутимый прирост точности по сравнению с одиночными моделями. VotingClassifier accuracy на test = 0.8492, а при кросс-валидации (5 фолдов) VotingClassifier средний accuracy на test = 0.8305 ± 0.0207. StackingClassifier Accuracy на test = 0.8324, при кросс-валидации (5 фолдов) StackingClassifier средний accuracy = 0.8316 ± 0.0258.

Это подтверждает, что комбинация разных подходов позволяет чуть лучше обобщать паттерны в данных.

Нейронная сеть (MLP) иногда не успевает сойтись за 1000 эпох, но даже в этом случае её качество сопоставимо с классическими моделями.

3 Выполнение итогового задания №2

Итоговое задание №2 представляло собой создание модели машинного обучения для определения популярности трека в Spotify [7]. Работа включала в себя EDA и ресерч-анализ. Были построены графики, применены инструменты EDA, проведен анализ каждого графика. Рассмотрена корреляция колонок с таргетом. Проводился Feature Engineering, после которого рассматривалась корреляция новых колонок с таргетом. Были созданы простые модели регрессии и проведена оценка точности моделей на представленном датасете.

3.1 Исследовательский анализ (EDA)

Как и в прошлом задании, исследовательский анализ данных необходим для комплексного понимания структуры данных. При первичном осмотре удалось выявить 1 строку с пустыми значениями в некоторых столбцах, ее пришлось удалить. Далее признаки были разделены на числовые и категориальные.

Для числовых признаков построены гистограммы (приложение В) и боксплоты (приложение Г). Также создана тепловая карта корреляций всех числовых признаков (рисунок 4).

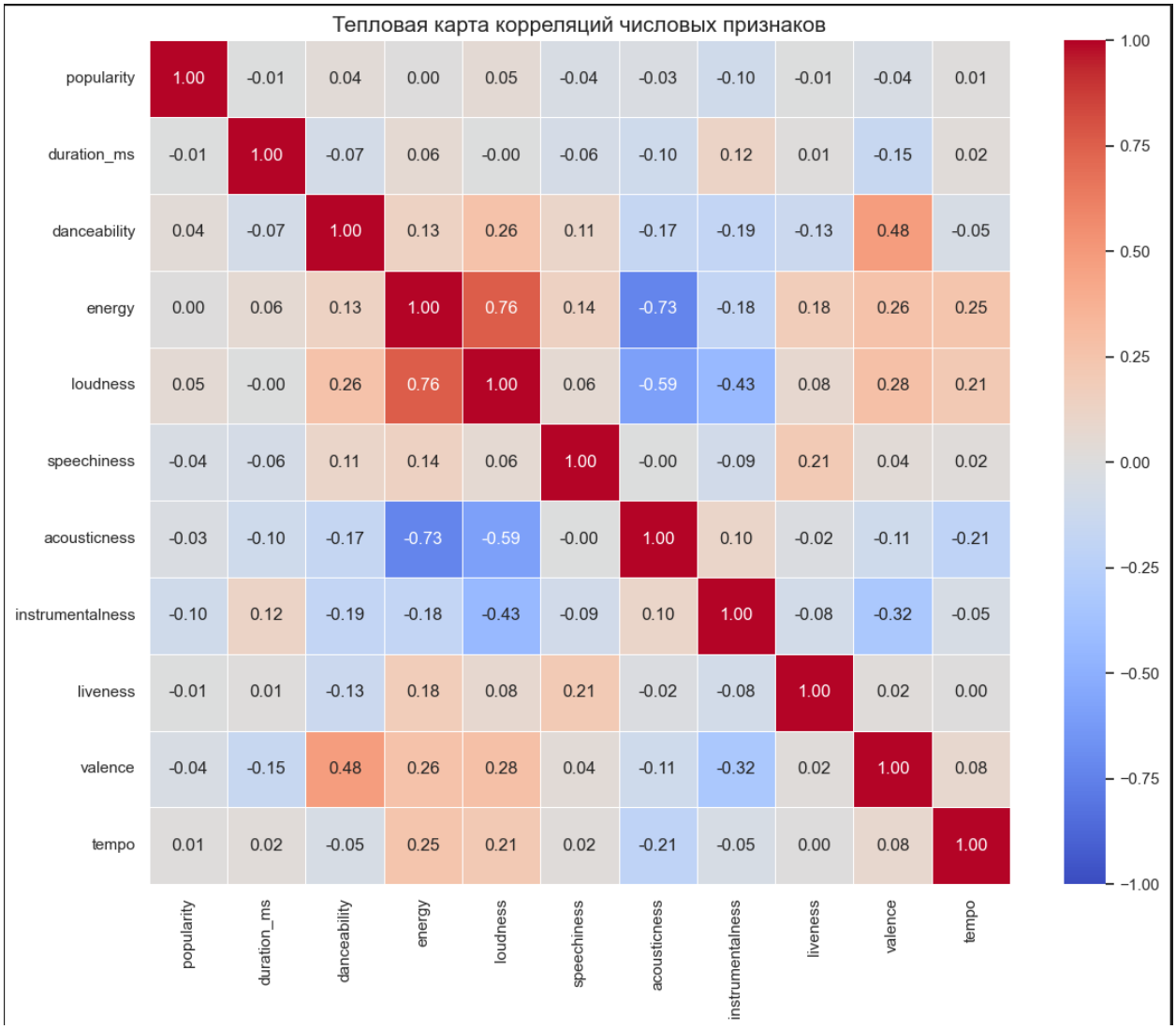


Рисунок 4 – Тепловая карта корреляций числовых признаков Spotify

По тепловой карте был сделан вывод о том, что Popularity (целевая переменная очень слабо коррелирует со всеми числовыми признаками (< 0.1)

Для категориальных признаков построены боксплоты (приложение Д) и отдельно для жанров и артистов созданы гистограммы (рисунок 5, рисунок 6).

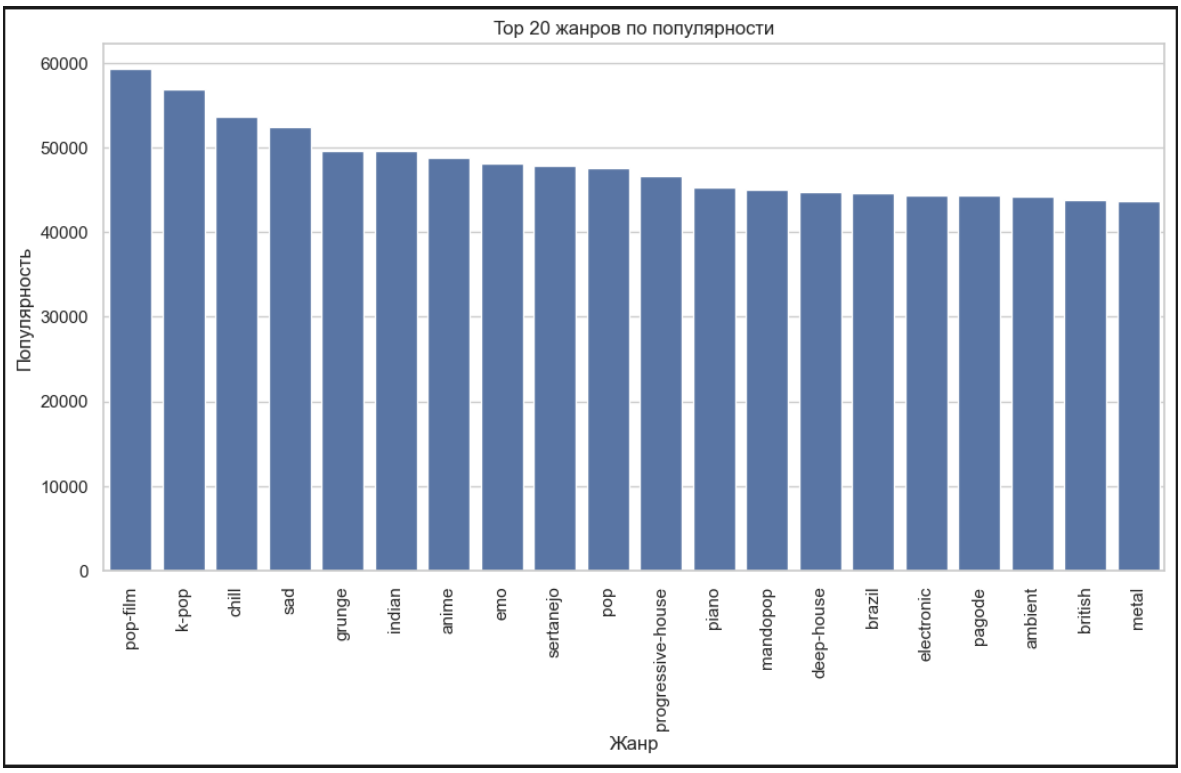


Рисунок 5 – Жанры по популярности

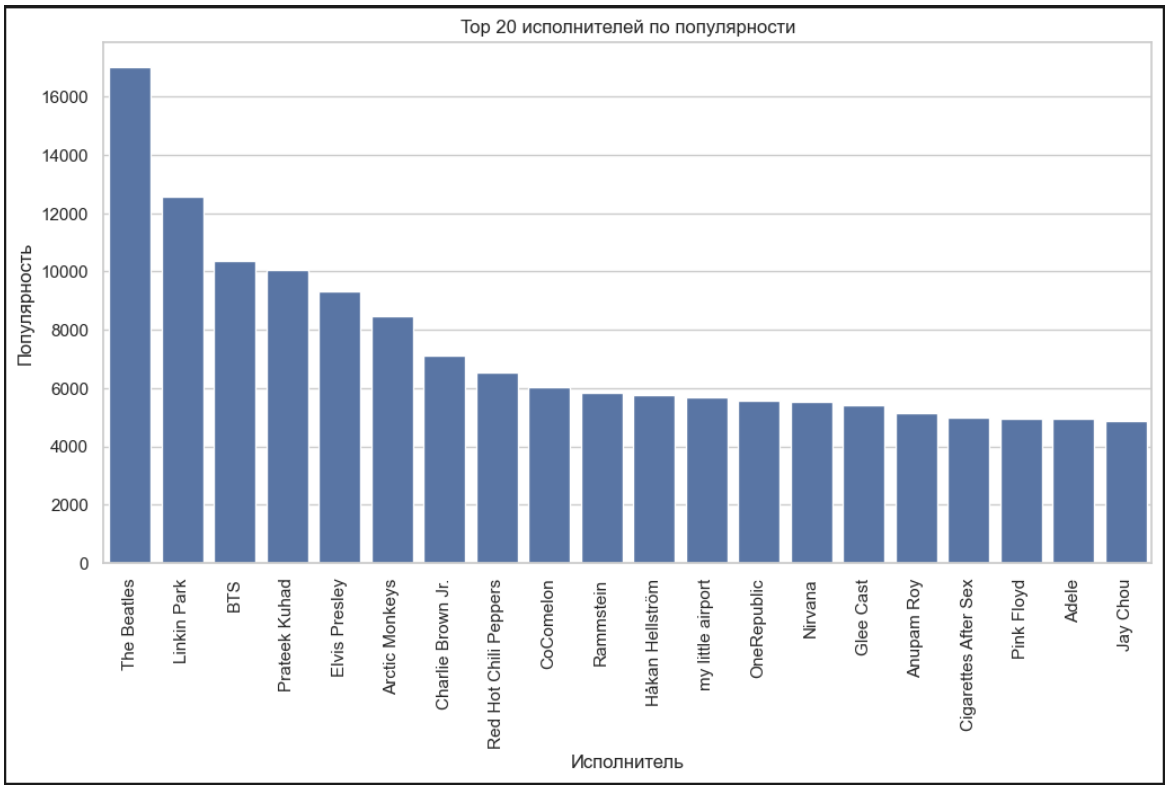


Рисунок 6 – Артисты(исполнители) по популярности

Большинство категориальных признаков (mode, explicit, key, time_signature) оказывали слабое влияние на популярность — различия медиан минимальны. Жанр (track_genre) — признак, где различия в популярности

между категориями выражены явно. В Artists (исполнители) различия тоже выражены явно, но с этим столбцом сложно работать, поскольку есть совместные песни, а также много аномалий.

3.2 Feature Engineering

Жанры были сгруппированы в более широкие категории, такие как электронная музыка, поп, рок, хип-хоп, джаз/блюз, классическая музыка, латино, мировая музыка, регги, настроенческая музыка, специфические жанры и кантри. Это помогает уменьшить размерность признаков, связанных с жанрами, и охватывает более широкие музыкальные стили.

Темп был разделен на категории: медленный, средний, быстрый и очень быстрый. Эта категоризация основана на наблюдаемых пиках в распределении темпа, что позволяет более структурированно анализировать эффекты, связанные с темпом.

Был создан бинарный признак is_extreme_tempo, указывающий, находится ли темп трека в верхних или нижних 10% распределения темпа. Это может помочь выявить треки с необычными характеристиками темпа.

Был введен новый признак features_balance, измеряющий баланс между энергией, танцевальностью и валентностью. Этот признак отражает гармонию между этими атрибутами, что может коррелировать с популярностью трека.

Логарифмические преобразования были применены к speechiness и instrumentalness для обработки скошенных распределений и стабилизации дисперсии. Категориальные признаки, такие как key, time_signature, genre_category, tempo_bin и duration_category, были закодированы методом one-hot для преобразования их в формат, подходящий для моделей машинного обучения.

Выбросы в duration_min, loudness и tempo были удалены с использованием метода IQR, чтобы гарантировать, что экстремальные значения не искажают обучение модели.

Были применены различные методы масштабирования:

– StandardScaler для нормально распределенных признаков, таких как энергия, танцевальность и валентность;

– RobustScaler для признаков с выбросами, таких как громкость, темп и длительность;

– MinMaxScaler для признаков, которые нужно привести к диапазону [0,1], таких как speechiness_log и instrumentalness_log.

Для оценки преобразований был построен график корреляции новых признаков с популярностью треков (рисунок 7).

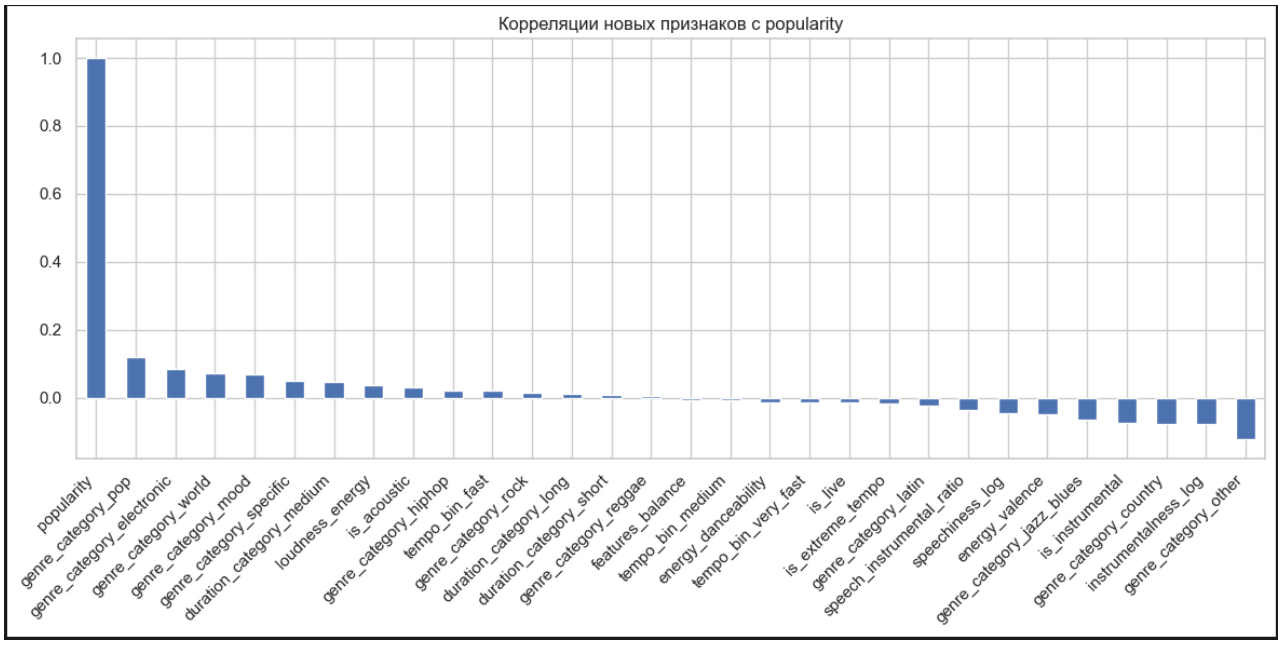


Рисунок 7 – Корреляция новых признаков с popularity

Корреляции с популярностью треков по-прежнему слабые, но заметна корреляция с некоторыми жанрами (например, корреляция с категорией поп).

3.3 Обучение моделей

В процессе обучения моделей на наборе данных Spotify были выполнены следующие шаги:

1) подготовка данных:

a) выделен ключевой признак popularity;

b) выделены все остальные признаки для предсказания;

c) данные разделены на обучающую и тестовую выборки в соотношении 80/20.

2) обучение моделей:

a) использовались следующие модели: Линейная регрессия, Градиентный бустинг, XGBoost, KNN, Случайный лес, MLP;

b) лучшая модель по тестовой выборке: Случайный лес со средней квадратической ошибкой RMSE = 16.5319.

После кросс-валидации из 5 фолдов модель Случайного леса показала среднее RMSE = 16.4202 ± 0.1344. Для наглядности был построен график предсказанных и настоящих значений популярности (рисунок 8).

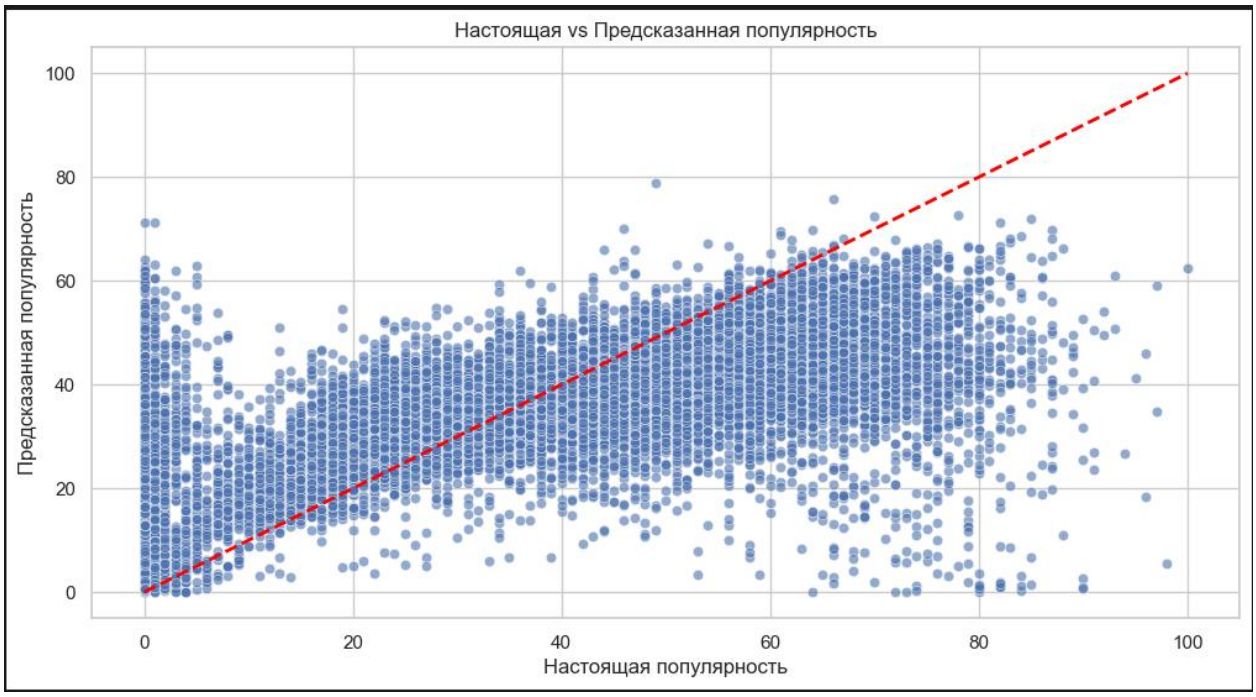


Рисунок 8 – График настоящие vs предсказанные значения

Наглядно видно, что модель не идеальна - в области маленьких значений наибольшие проблемы. Но в целом, удалось добиться приемлемых результатов предсказания.

ЗАКЛЮЧЕНИЕ

В ходе выполнения проектной работы были разработаны и протестированы модели машинного обучения для решения задач классификации (прогнозирование выживаемости пассажиров Титаника) и регрессии (предсказание популярности треков в Spotify). Результаты работы демонстрируют соответствие поставленным целям, однако выявлены аспекты, требующие доработки и оптимизации.

Оценка соответствия требованиям:

1) лабораторные работы:

a) освоены ключевые методы обработки данных (NumPy, Pandas), корреляционного анализа, построения линейных моделей и ансамблирования;

b) реализованные модели показали удовлетворительные результаты (например, ROC-AUC = 0.7952 для Voting Ensemble в лабораторной работе №4), что соответствует базовым требованиям к обучению;

c) однако в некоторых случаях (например, нейронные сети) наблюдалась нестабильность обучения, что указывает на необходимость более тщательного подбора гиперпараметров.

2) итоговые задания:

a) для задачи классификации (Титаник) наилучшая модель (MLP) достигла accuracy = 0.8268, а ансамбли (VotingClassifier) улучшили результат до 0.8492, это свидетельствует о корректной обработке данных и эффективном комбинировании моделей;

b) в задаче регрессии (Spotify) лучшая модель (Random Forest) показала RMSE = 16.53, что приемлемо, но указывает на ограниченную предсказательную силу из-за слабых корреляций признаков с целевой переменной.

Оценка качества и выявленные проблемы:

1) качество данных: в данных Spotify слабая корреляция признаков с популярностью ограничила точность моделей;

2) моделирование:

a) ансамблирование дало прирост точности, но требует больших вычислительных ресурсов;

b) нейронные сети показали нестабильность, особенно при недостаточном количестве эпох обучения.

3) интерпретируемость: линейные модели и деревья обеспечивают прозрачность решений, в отличие от MLP и ансамблей, что важно для задач, требующих объяснимости.

Проект успешно выполнен: все поставленные задачи решены, а модели демонстрируют достаточную работоспособность. Для улучшения результатов требуется еще больше экспериментировать с готовыми моделями и тщательнее работать с предоставленными данными в датасетах.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Высшая школа экономики: «Data Science». [Электронный ресурс]. URL : https://studyonline.hse.ru/data-science (дата обращения : 27.05.2025).

2. Репозиторий GitHub «lab1». [Электронный ресурс]. URL: https://github.com/vvoyage/lab1 (дата обращения : 27.05.2025).

3. Репозиторий GitHub «lab2». [Электронный ресурс]. URL: https://github.com/vvoyage/lab2 (дата обращения : 27.05.2025).

4. Репозиторий GitHub «lab3». [Электронный ресурс]. URL: https://github.com/vvoyage/lab3 (дата обращения : 27.05.2025).

5. Репозиторий GitHub «lab4». [Электронный ресурс]. URL: https://github.com/vvoyage/lab4 (дата обращения : 27.05.2025).

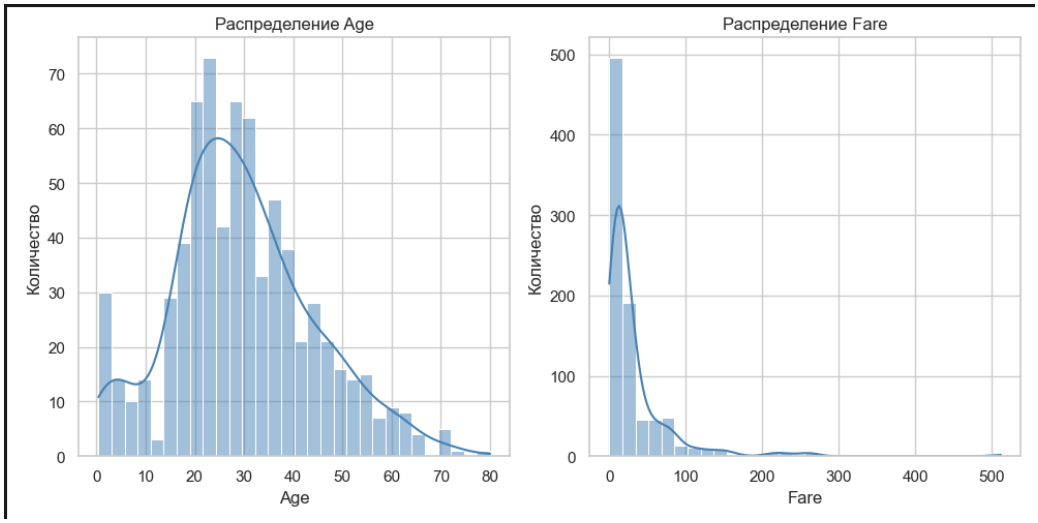
6. Репозиторий GitHub «titanic_contest». [Электронный ресурс]. URL: https://github.com/vvoyage/titanic_contest (дата обращения : 27.05.2025).

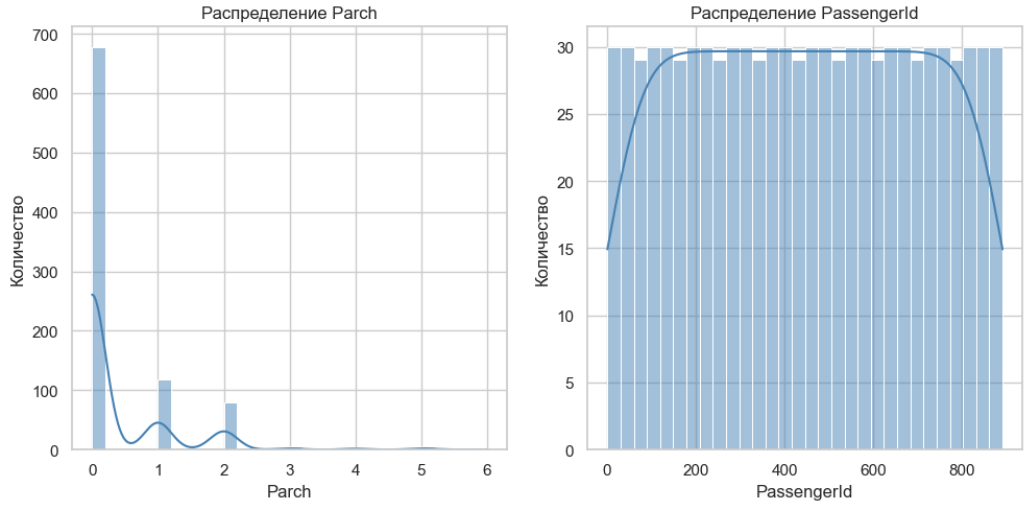
7. Репозиторий GitHub «spotify_contest». [Электронный ресурс]. URL: https://github.com/vvoyage/spotify_contest (дата обращения : 27.05.2025).

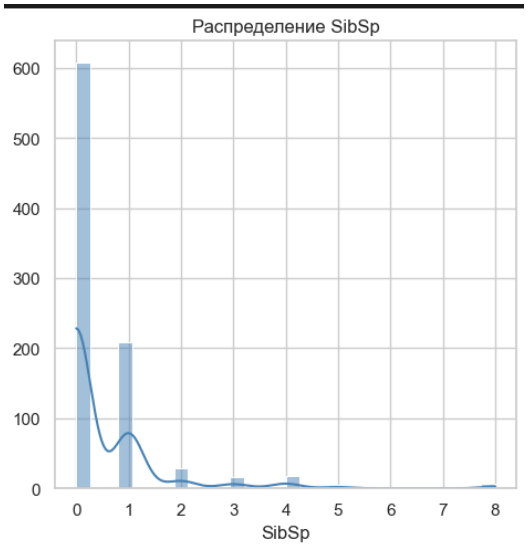
ПРИЛОЖЕНИЕ А

(справочное)

Графики распределения числовых данных в итоговом задании №1



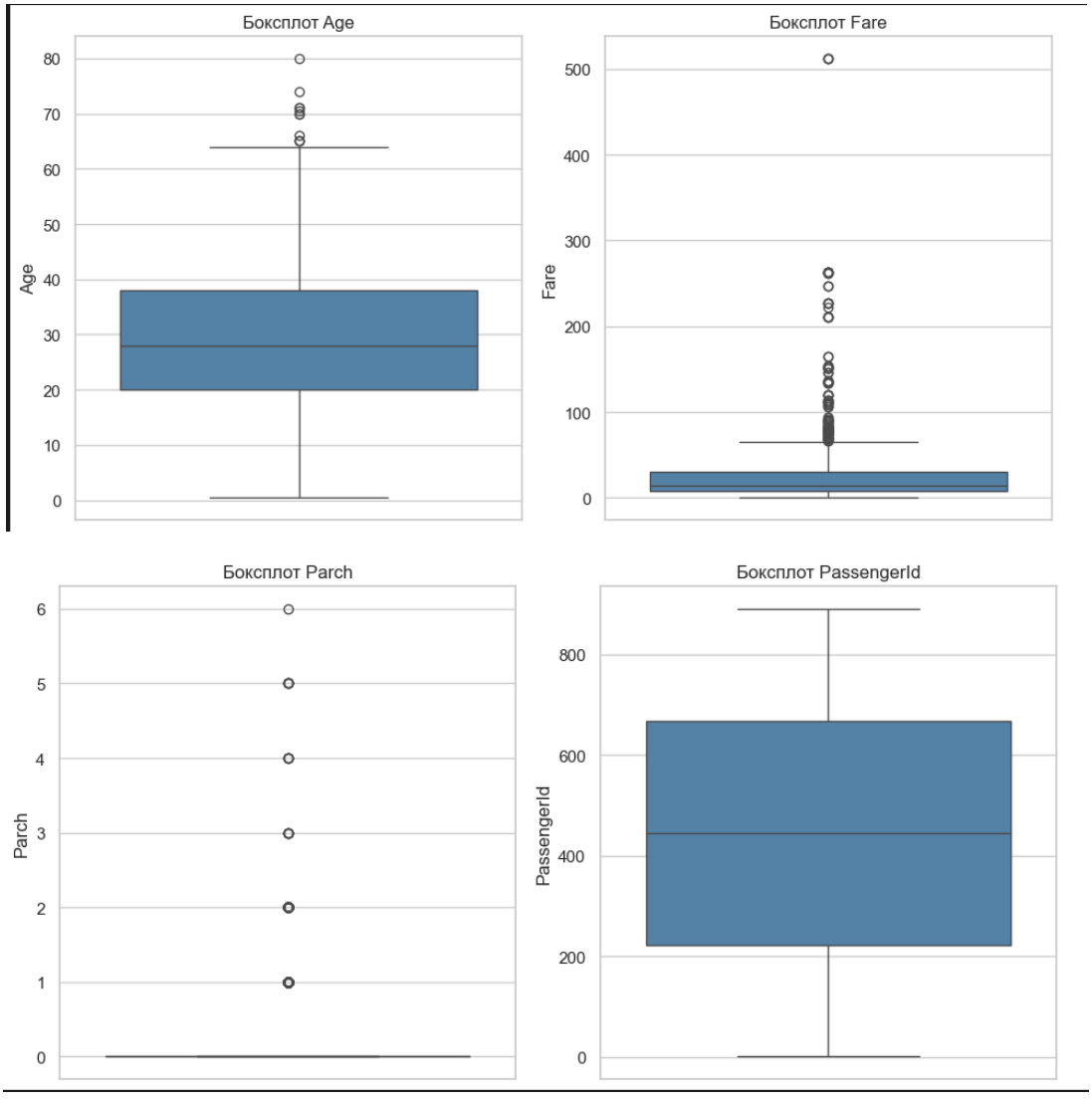


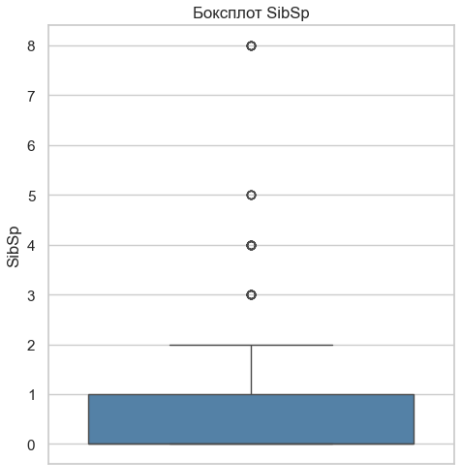


ПРИЛОЖЕНИЕ Б

(справочное)

Боксплоты числовых данных в итоговом задании №1

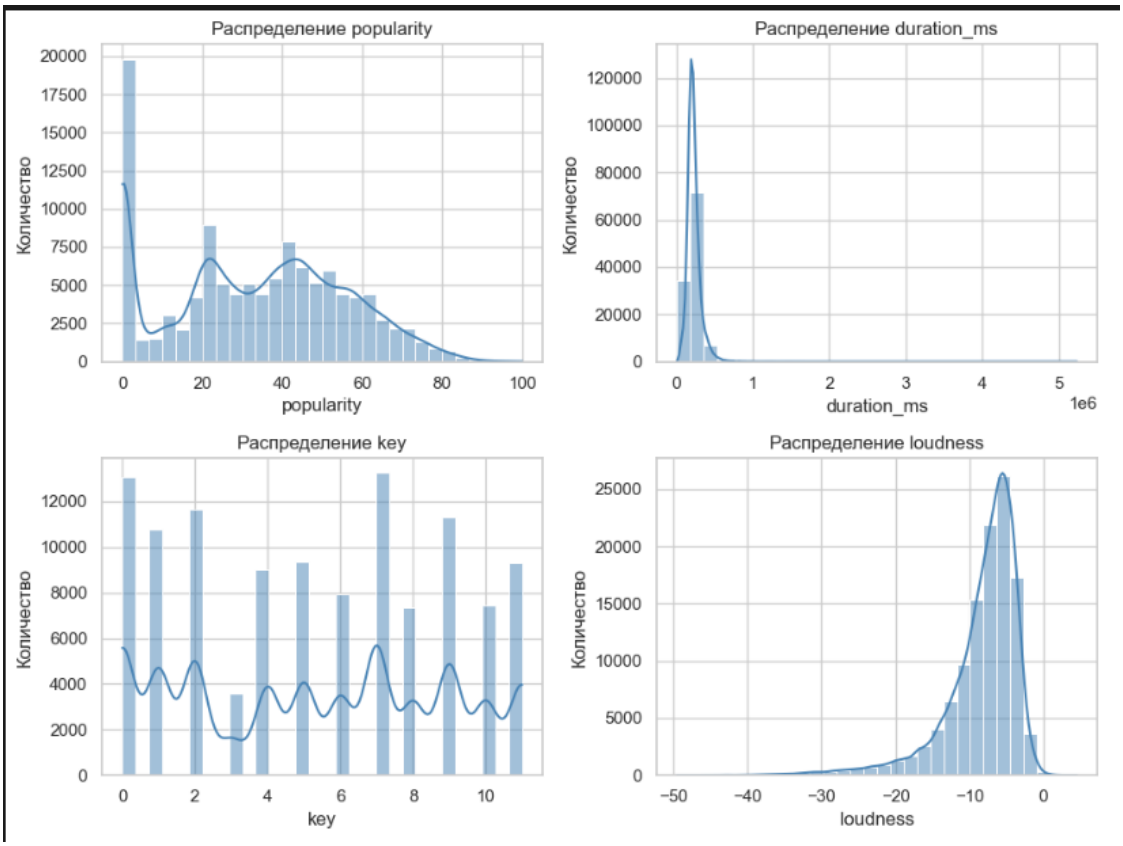


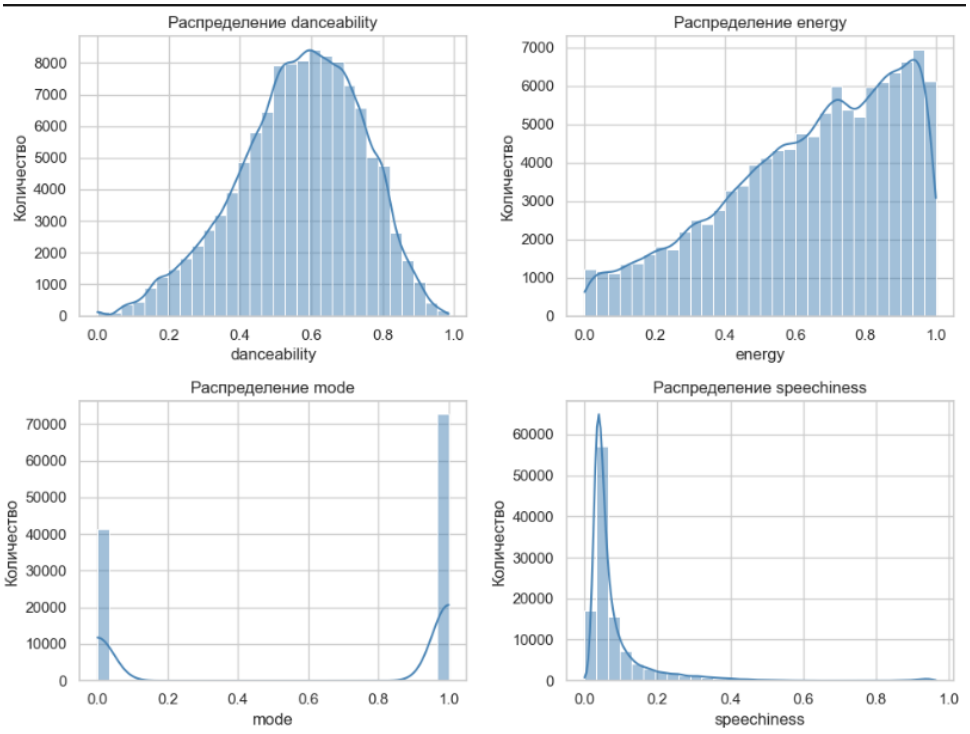


ПРИЛОЖЕНИЕ В

(справочное)

Графики распределения числовых данных в итоговом задании №2

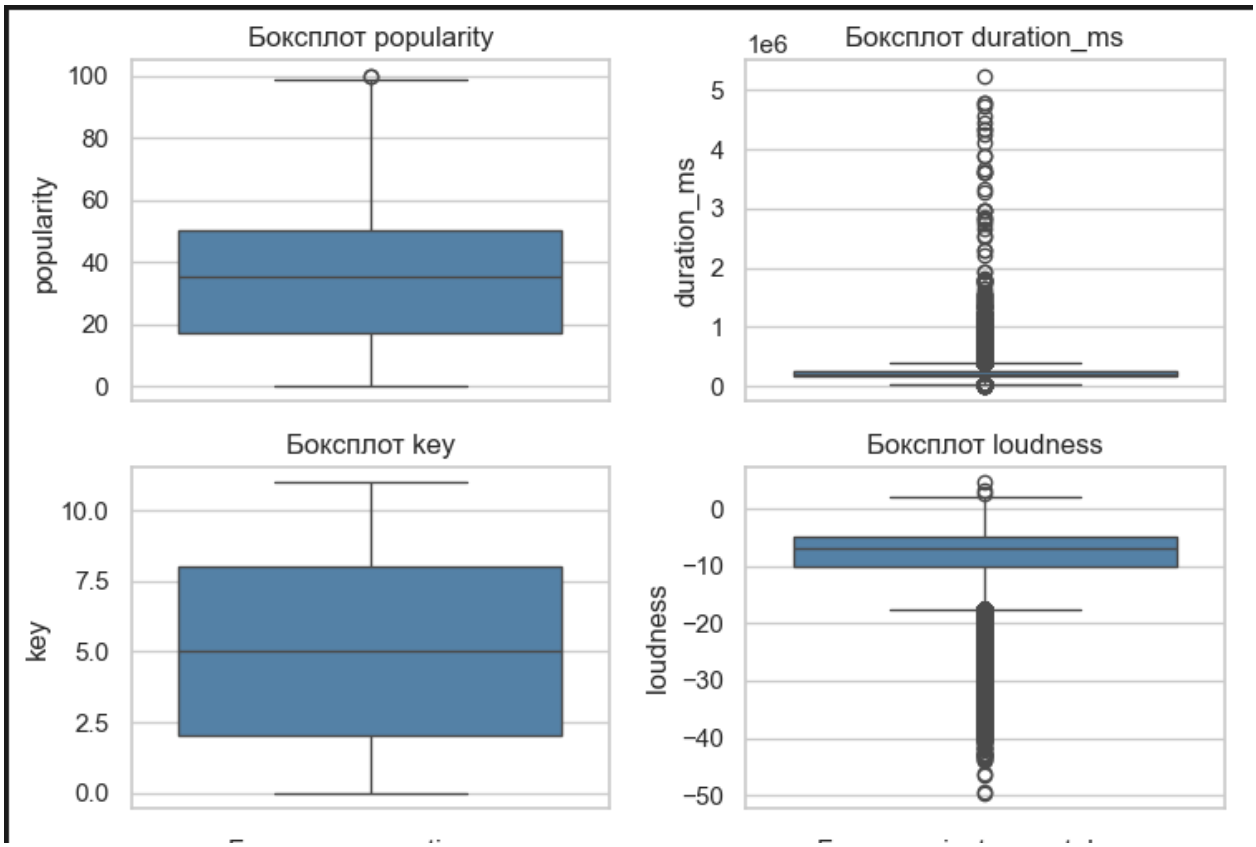


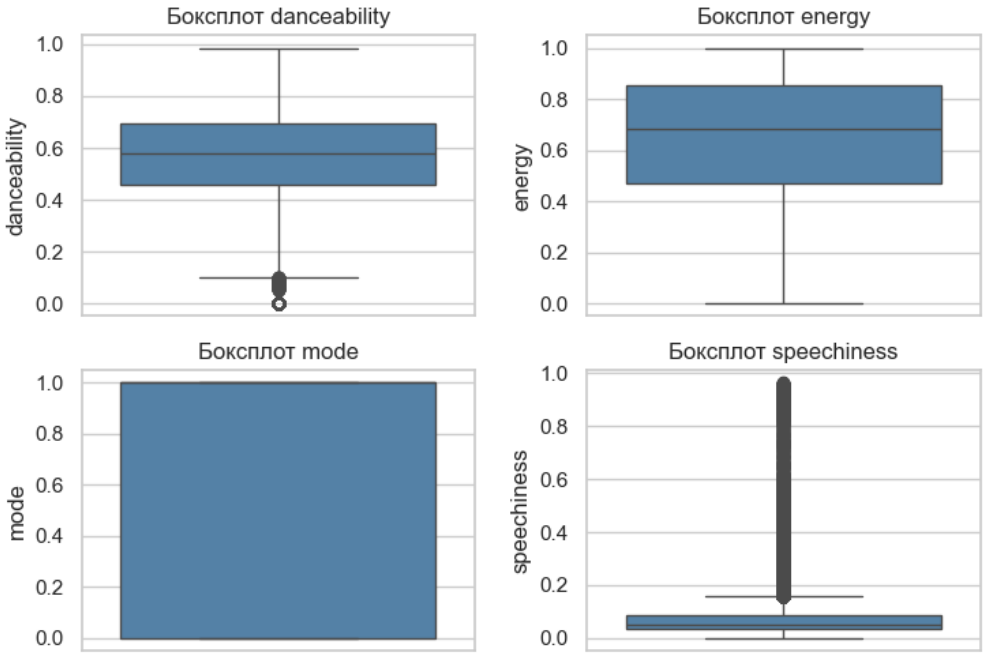


ПРИЛОЖЕНИЕ Г

(справочное)

Боксплоты числовых данных в итоговом задании №2





ПРИЛОЖЕНИЕ Д

(справочное)

Боксплоты категориальных данных в итоговом задании №2