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:

Обработка датасета

Студент ИУ5-61Б _____ Зобнин А. В.
(группа) (подпись, дата) (И.О. Фамилия)

Руководитель НИР _____ Гапанюк Ю. Е.
(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)

З А Д А Н И Е
на выполнение научно-исследовательской работы

по теме Обработка датасета

Студент группы ИУ5-61Б
Зобнин Александр Валерьевич

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:

Обработка датасета

Типовое исследование

Оформление научно-исследовательской работы:

Расчетно-пояснительная записка на 12 листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

Дата выдачи задания « 07 » февраля 2025 г.

Руководитель НИР _____ Гапанюк Ю. Е.
(подпись, дата) (И.О. Фамилия)

Студент _____ Зобнин А. В.
(подпись, дата) (И.О. Фамилия)

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

Содержание

Введение

Актуальность исследования

Качество вина — ключевой фактор, влияющий на его рыночную стоимость и потребительский спрос. Традиционные методы оценки вина основаны на работе экспертов-сомелье, что требует значительных временных и финансовых затрат. Однако современные технологии машинного обучения позволяют автоматизировать этот процесс, выявляя зависимости между химическими характеристиками вина и его качеством.

Цель исследования

Построить модель машинного обучения, способную классифицировать качество красного вина на основе его химических свойств. В ходе работы необходимо:

- Провести разведочный анализ данных (EDA) для выявления ключевых закономерностей.
- Выбрать оптимальные метрики оценки качества модели с учётом дисбаланса классов.
- Сравнить различные алгоритмы (включая ансамблевые методы) и выбрать наилучший.
- Разработать интерактивное веб-приложение для демонстрации работы модели.

Объект исследования

Датасет Red Wine Quality, содержащий 1143 образца португальского вина "Vinho Verde" с 11 химическими признаками и оценкой качества от 0 до 10 баллов.

Практическая значимость

Результаты исследования могут быть использованы для:

- Автоматизации контроля качества на производстве.
- Оптимизации рецептур вин на основе данных.
- Образовательных целей (демонстрация работы ML-моделей).

Работа демонстрирует полный цикл решения задачи машинного обучения: от анализа данных до внедрения в виде веб-приложения.

Постановка задачи

- Поиск и выбор набора данных для построения моделей машинного обучения. На основе выбранного набора данных студент должен построить модели машинного обучения для решения или задачи классификации, или задачи регрессии.
- Проведение разведочного анализа данных. Построение графиков, необходимых для понимания структуры данных. Анализ и заполнение пропусков в данных.
- Выбор признаков, подходящих для построения моделей. Кодирование категориальных признаков. Масштабирование данных. Формирование вспомогательных признаков, улучшающих качество моделей.
- Проведение корреляционного анализа данных. Формирование промежуточных выводов о возможности построения моделей машинного обучения. В зависимости от набора данных, порядок выполнения пунктов 2, 3, 4 может быть изменен.
- Выбор метрик для последующей оценки качества моделей. Необходимо выбрать не менее трех метрик и обосновать выбор.
- Выбор наиболее подходящих моделей для решения задачи классификации или регрессии. Необходимо использовать не менее пяти моделей, две из которых должны быть ансамблевыми.
- Формирование обучающей и тестовой выборок на основе исходного набора данных.
- Построение базового решения (baseline) для выбранных моделей без подбора гиперпараметров. Производится обучение моделей на основе обучающей выборки и оценка качества моделей на основе тестовой выборки.
- Подбор гиперпараметров для выбранных моделей. Рекомендуется использовать методы кросс-валидации. В зависимости от используемой библиотеки можно применять функцию GridSearchCV, использовать перебор параметров в цикле, или использовать другие методы.
- Повторение пункта 8 для найденных оптимальных значений гиперпараметров. Сравнение качества полученных моделей с качеством baseline-моделей.
- Формирование выводов о качестве построенных моделей на основе выбранных метрик. Результаты сравнения качества рекомендуется отобразить в виде графиков и сделать выводы в форме текстового описания. Рекомендуется построение графиков обучения и валидации, влияния значений гиперпарметров на качество моделей и т.д.
- Создание веб-приложение для демонстрации хотя бы одной модели машинного обучения. У пользователя должна быть возможность изменения хотя бы одного гиперпараметра модели, при изменении гиперпараметра модель должна перестраиваться в веб-интерфейсе.

Последовательность действий

1. Импорт библиотек

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.preprocessing import StandardScaler

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from xgboost import XGBClassifier
from sklearn.metrics import accuracy_score, f1_score, roc_auc_score

from sklearn.model_selection import GridSearchCV

import numpy as np
```

2. Загрузка и предварительный анализ данных

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality	Id
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5	0
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5	1
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5	2
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6	3
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5	4

Рисунок 1 – Датасет

3. Разведочный анализ данных (EDA)
    a. Распределение качества вина

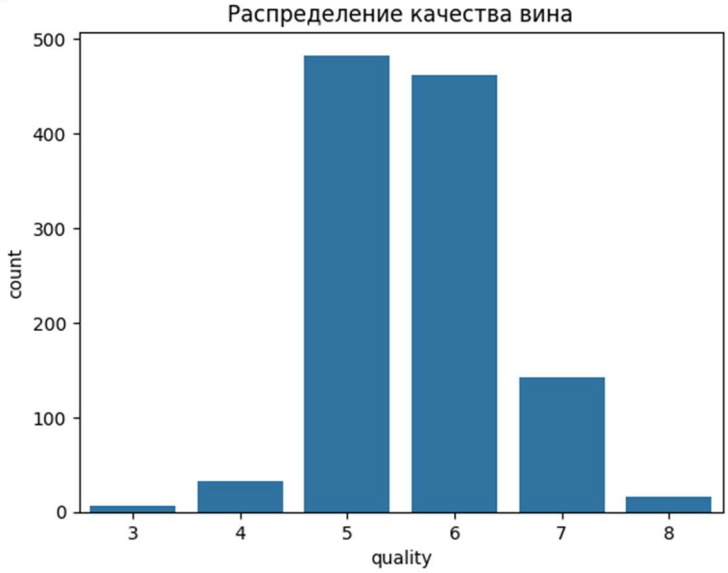

Рисунок 2 – Распределение качества вина

b. Гистограммы признаков

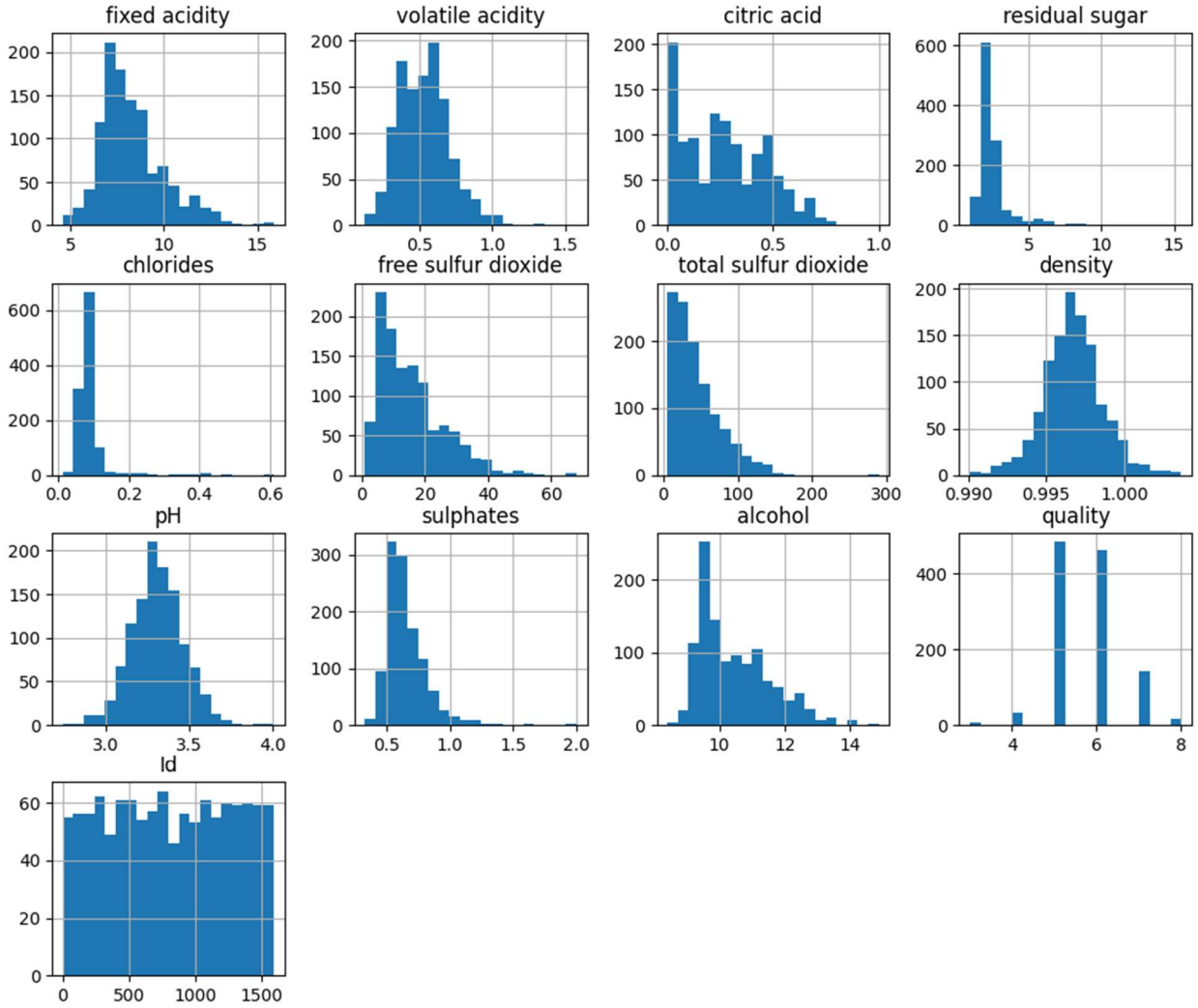

Рисунок 3 – Гистограммы признаков

c. Корреляционный анализ

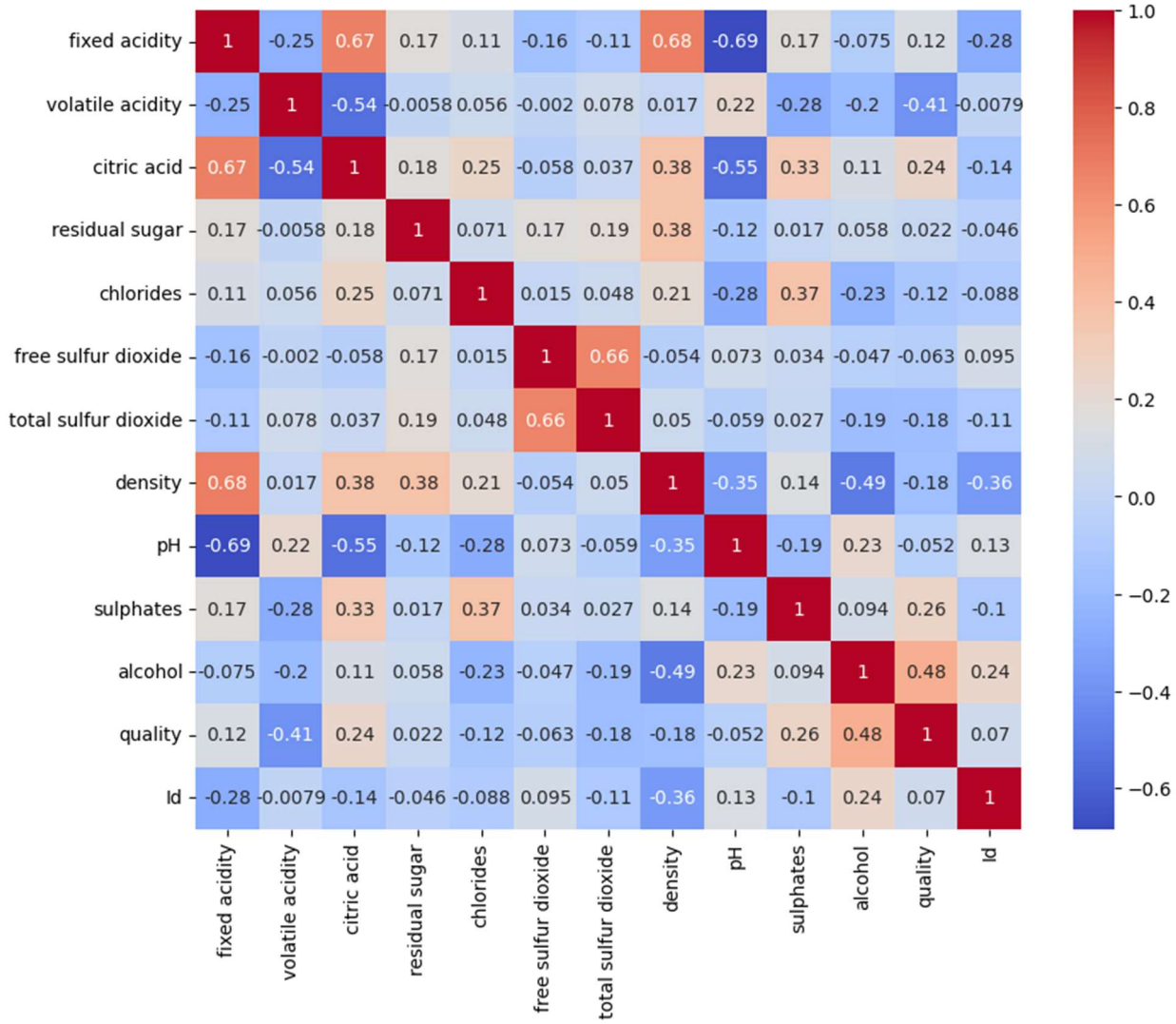

Рисунок 4 – Корреляционный анализ

4. Подготовка данных

```
data['quality_class'] = pd.cut(data['quality'], bins=[0, 4, 6, 10], labels=[0, 1, 2])
X = data.drop(['quality', 'quality_class'], axis=1)
y = data['quality_class']

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

5. Выбор метрик

Для задачи классификации с дисбалансом классов выберем:

- Accuracy (общая точность): одна из самых популярных метрик
- F1-score (среднее гармоническое precision и recall): подходит для задач с дисбалансом классов
- ROC-AUC (площадь под ROC-кривой, для многоклассовой классификации): для сравнения моделей на вероятностях, устойчива к дисбалансу

6. Разделение данных

```
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42, stratify=y)
```

7. Выбор и обучение моделей (baseline)

```
models = {
    'Logistic Regression': LogisticRegression(),
    'SVM': SVC(probability=True),
    'Random Forest': RandomForestClassifier(),
    'Gradient Boosting': GradientBoostingClassifier(),
    'XGBoost': XGBClassifier()
}

results = {}
for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    results[name] = {
        'Accuracy': accuracy_score(y_test, y_pred),
        'F1-score': f1_score(y_test, y_pred, average='weighted'),
        'ROC-AUC': roc_auc_score(y_test, model.predict_proba(X_test), multi_class='ovr')
    }

results_df = pd.DataFrame(results).T
print(results_df)
```

	Accuracy	F1-score	ROC-AUC
Logistic Regression	0.834061	0.808235	0.803709
SVM	0.851528	0.816625	0.795464
Random Forest	0.903930	0.884047	0.842532
Gradient Boosting	0.877729	0.862444	0.775337
XGBoost	0.886463	0.870817	0.787009

Рисунок 5 – Первоначальная оценка моделей

8. Подбор гиперпараметров

```
param_grids = {
    'Logistic Regression': {
        'C': [0.1, 1],
        'penalty': ['l2']
    },
    'SVM': {
        'C': [0.1, 1, 10],
        'kernel': ['linear', 'rbf', 'poly'],
        'gamma': ['scale', 'auto']
    },
    'Random Forest': {
        'n_estimators': [75, 80, 85, 90, 95, 100, 105, 110, 115, 120],
        'max_depth': [10, 15, 20, 25]
    },
    'Gradient Boosting': {
        'n_estimators': [50, 100, 200],
        'learning_rate': [0.01, 0.1, 0.2],
        'max_depth': [3, 5, 7]
    },
    'XGBoost': {
```

```
        'n_estimators': [50, 100, 200],
        'learning_rate': [0.01, 0.1, 0.2],
        'max_depth': [3, 5, 7],
        'subsample': [0.8, 1.0],
        'colsample_bytree': [0.8, 1.0]
    }
}

best_models = {}
for name, model in models.items():
    print(f"Подбор параметров для модели: {name}")
    grid_search = GridSearchCV(
        estimator=model,
        param_grid=param_grids[name],
        cv=5,
        scoring='accuracy',
        n_jobs=-1,
        verbose=1
    )
    grid_search.fit(X_train, y_train)
    best_models[name] = grid_search.best_estimator_
    print(f"Лучшие параметры для {name}: {grid_search.best_params_}\n")
```

9. Сравнение моделей с оптимальными гиперпараметрами с baseline

```
Результаты после подбора гиперпараметров:
                     Accuracy  F1-score   ROC-AUC
Logistic Regression  0.838428  0.809073  0.792483
SVM                  0.877729  0.861198  0.798737
Random Forest        0.908297  0.889094  0.864533
Gradient Boosting    0.899563  0.882744  0.803974
XGBoost              0.895197  0.878750  0.759518

Сравнение с baseline:
                     Baseline                          Optimized            \
                     Accuracy  F1-score   ROC-AUC  Accuracy  F1-score
Logistic Regression  0.834061  0.808235  0.803709  0.838428  0.809073
SVM                  0.851528  0.816625  0.795464  0.877729  0.861198
Random Forest        0.903930  0.884047  0.842532  0.908297  0.889094
Gradient Boosting    0.877729  0.862444  0.775337  0.899563  0.882744
XGBoost              0.886463  0.870817  0.787009  0.895197  0.878750


                      ROC-AUC
Logistic Regression  0.792483
SVM                  0.798737
Random Forest        0.864533
Gradient Boosting    0.803974
XGBoost              0.759518
```

Рисунок 6 – Оценка оптимальных моделей

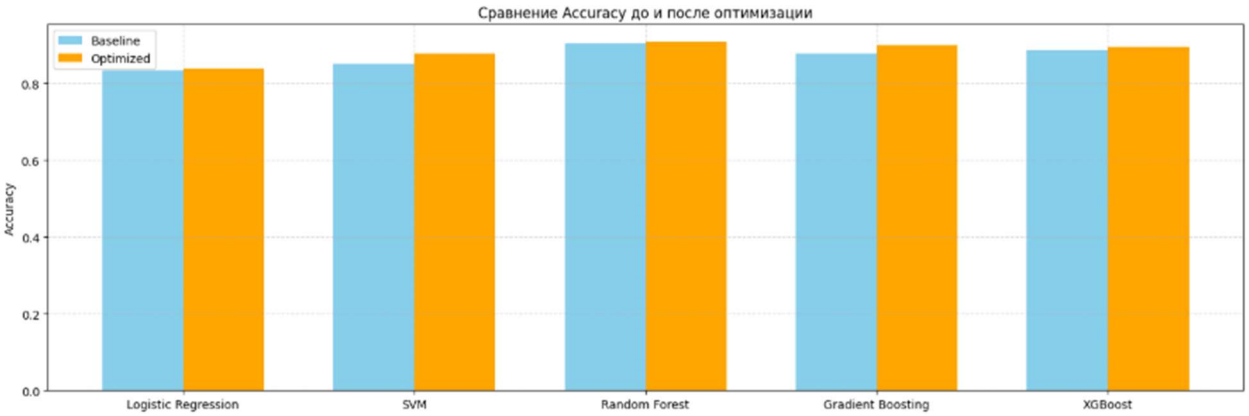

Рисунок 7 – Сравнение Accuracy до и после оптимизации

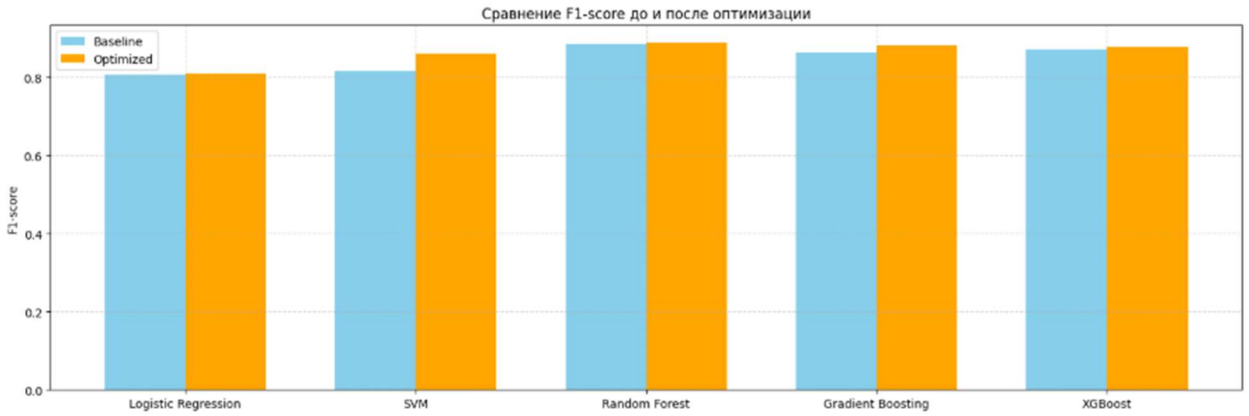

Рисунок 8 – Сравнение F1-score до и после оптимизации

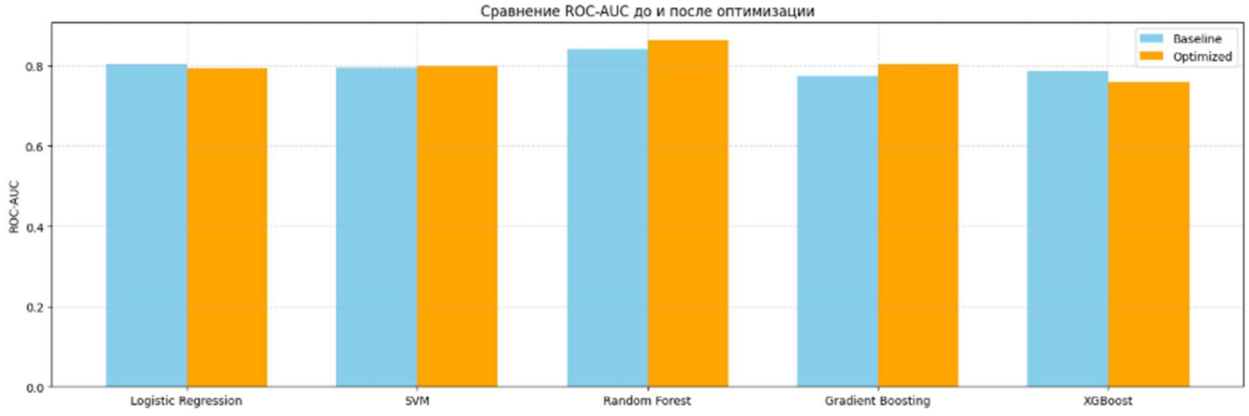

Рисунок 9 – Сравнение ROC-AUC до и после оптимизации

10. Влияние гиперпараметров на качество (для Random Forest)

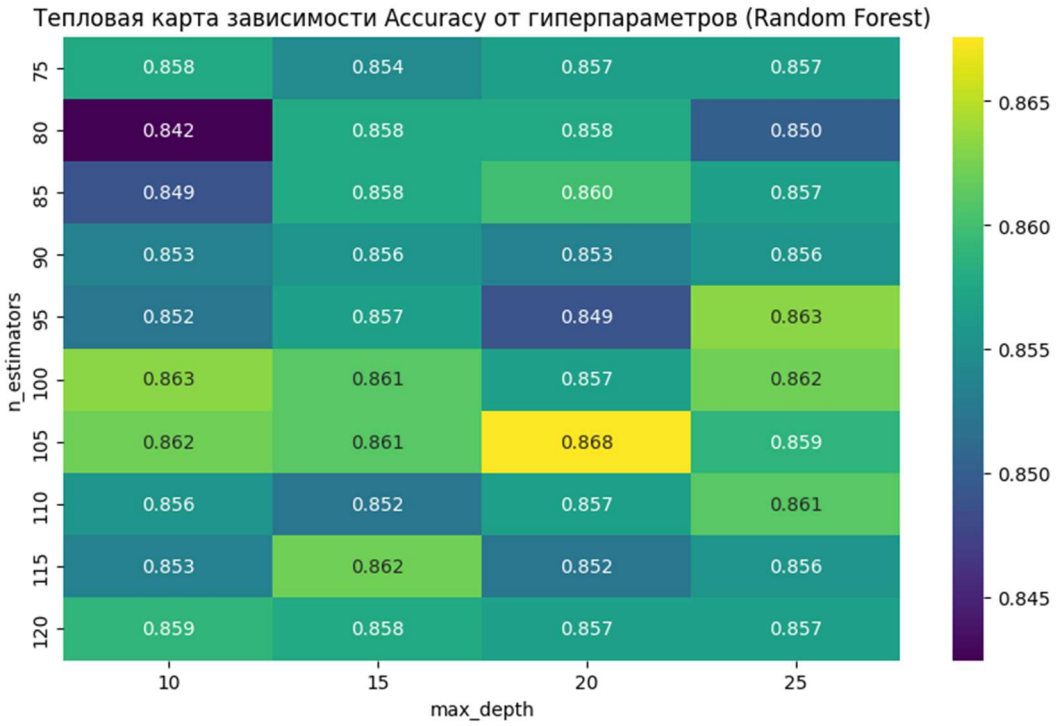

Рисунок 10 – Тепловая карта зависимости Accuracy от гиперпараметров

11. Веб-приложение

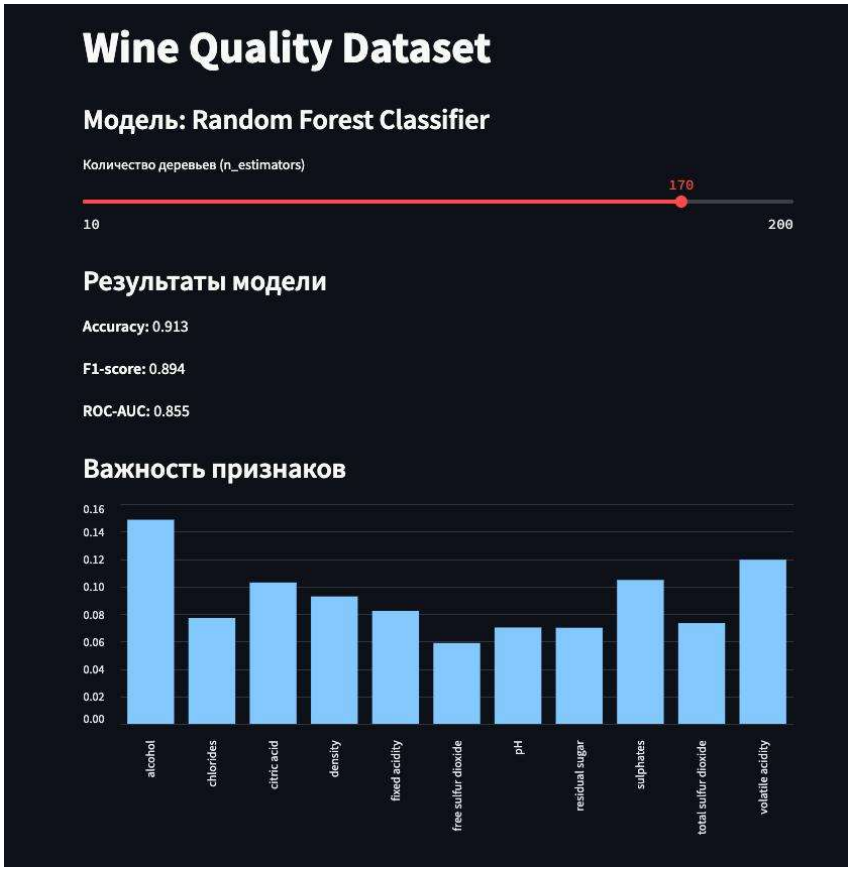

Рисунок 11 – Веб-приложение для демонстрации результатов Random Forest

Заключение

- Лучшая модель: Random Forest (F1-score = 0.889).
- Ансамблевые методы (Random Forest, Gradient Boosting, XGBoost) показали себя лучше других.
- Logistic Regression слабо реагирует на подбор параметров

Список источников

1. Документация Pandas – [https://pandas.pydata.org/]
2. Документация Matplotlib – [https://matplotlib.org/]
3. Документация Sklearn – [https://scikit-learn.org/stable/]
4. Документация XGBoost – [https://xgboost.readthedocs.io/en/release_3.0.0/]
5. Документация Numpy – [https://numpy.org/]
6. Документация Seaborn – [https://seaborn.pydata.org/]
7. Датасет Red Wine Quality – [https://www.kaggle.com/datasets/uciml/red-wine-quality-cortez-et-al-2009]
8. Метрика Accuracy – [https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall]
9. Метрика F1-score – [https://www.geeksforgeeks.org/f1-score-in-machine-learning/]
10. Метрика ROC-AUC – [https://habr.com/ru/companies/otus/articles/809147/]