



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 _____ РАДИОТЕХНИЧЕСКИЙ _____

КАФЕДРА _____ СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ _____

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

НА ТЕМУ:

**Исследование применения моделей машинного
обучения для прогнозирования качества вина по
физико-химическим характеристикам**

Студент _____
_____ (Группа)

(Подпись, дата) **К.А. Ильина**
(И.О.Фамилия)

Руководитель

(Подпись, дата) **Ю.Е. Гапанюк**
(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)
« 07 » февраля 2024 г.

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

по теме Исследование применения моделей машинного обучения для прогнозирования качества вина по физико-химическим характеристикам

Студент группы РТ5-61Б

Ильина Ксения Андреевна
(Фамилия, имя, отчество)

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., 100% к нед.

Техническое задание Разработать модель машинного обучения для прогнозирования качества вина на основе его физико-химических характеристик. Использовать датасет Wine Quality с возможностью объединения данных по красному и белому вину. Выполнить разведочный анализ данных, подготовить признаки, обучить не менее пяти моделей, включая ансамблевые. Выбрать метрики оценки (MAE, MSE, R²) и провести подбор гиперпараметров с использованием GridSearchCV. Сравнить качество baseline и оптимизированных моделей. Реализовать интерактивное веб-приложение с помощью Streamlit, позволяющее пользователю изменять параметры вина и гиперпараметры модели, а также получать прогноз качества.

Оформление научно-исследовательской работы:

Расчетно-пояснительная записка на 21 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

К.А. Ильина
(Подпись, дата) (И.О.Фамилия)

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

Содержание

1. Введение.....	3
2. Основная часть	4
Постановка задачи.....	4
Подготовка данных	4
Обучение моделей.....	10
Подбор гиперпараметров	12
Анализ результатов.....	15
Разработка веб-приложения.....	16
3. Заключение	21
4. Список использованных источников	22
Электронные ресурсы:.....	22
Литература:	22

Введение

В рамках курсовой работы была выполнена задача прогнозирования качества вина на основе его химического состава. Целью проекта являлось построение модели машинного обучения, способной предсказывать числовую оценку качества вина (от 3 до 8) на основании параметров, таких как кислотность, содержание алкоголя, сахара и других веществ.

Работа проводилась в соответствии с методикой типового исследования, включающей этапы:

- Подготовки и анализа данных
- Обучения нескольких моделей машинного обучения
- Подбора гиперпараметров
- Оценки качества моделей
- Создания интерактивного веб-интерфейса для демонстрации результатов

Проект имеет практическое значение, так как может быть использован в винодельческой промышленности для улучшения технологических процессов и контроля качества продукции.

Основная часть

Постановка задачи

Задача заключалась в решении проблемы регрессии — предсказании числового значения качества вина (**quality**) на основе его химического состава.

Для решения задачи были выбраны следующие этапы:

- Загрузка и объединение датасетов **winequality-red.csv** и **winequality-white.csv**
- Предобработка данных: кодирование категориальных признаков, масштабирование числовых признаков
- Обучение пяти моделей машинного обучения, из них две ансамблевые
- Подбор гиперпараметров с помощью **GridSearchCV**
- Сравнение качества моделей
- Создание веб-приложения для демонстрации лучшей модели

Подготовка данных

Для работы был использован набор данных Wine Quality, содержащий информацию о красном и белом вине. Данные были загружены, объединены и преобразованы:

- Категориальный признак **type** (красное/белое вино) закодирован через One-Hot Encoding
- Все числовые признаки масштабированы с помощью **StandardScaler**
- Выполнена проверка на наличие пропусков
- Построены графики распределения целевой переменной и корреляционной матрицы

Эти действия позволили подготовить данные для последующего обучения моделей.

Код:

```
# Считываем данные
```

```
red_wine = pd.read_csv('winequality-red.csv', sep=';')
```

```

white_wine = pd.read_csv('winequality-white.csv', sep=';')

# Добавляем столбец типа вина
red_wine['type'] = 'red'
white_wine['type'] = 'white'

# Объединяем датасеты
data = pd.concat([red_wine, white_wine], axis=0).reset_index(drop=True)

# Создаем новые признаки
data['alcohol_to_sugar'] = data['alcohol'] / (data['residual sugar'] + 1e-6)
data['acid_balance'] = data['citric acid'] / (data['volatile acidity'] + 1e-6)
data['total_acidity'] = data['fixed acidity'] + data['volatile acidity'] + data['citric acid']

# Проверяем новые признаки
print(data[['alcohol_to_sugar', 'acid_balance', 'total_acidity']].describe())

# Кодировем категориальный признак 'type'
data = pd.get_dummies(data, columns=['type'], drop_first=True) # создаст
'type_white' = 1, 'type_red' = 0

# Проверяем
print("Форма датасета:", data.shape)
data.head()

```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality	alcohol_to_sugar	acid_balance	total_acidity	type_white
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5	4.947366	0.000000	8.10	False
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5	3.769229	0.000000	8.68	False
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5	4.260868	0.052632	8.60	False
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6	5.157892	1.999993	12.04	False
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5	4.947366	0.000000	8.10	False

```

# Описательная статистика
print(data.describe())

```

```
plt.show()
```

	fixed acidity	volatile acidity	citric acid	residual sugar \
count	6497.000000	6497.000000	6497.000000	6497.000000
mean	7.215307	0.339666	0.318633	5.443235
std	1.296434	0.164636	0.145318	4.757804
min	3.800000	0.080000	0.000000	0.600000
25%	6.400000	0.230000	0.250000	1.800000
50%	7.000000	0.290000	0.310000	3.000000
75%	7.700000	0.400000	0.390000	8.100000
max	15.900000	1.580000	1.660000	65.800000

	chlorides	free sulfur dioxide	total sulfur dioxide	density \
count	6497.000000	6497.000000	6497.000000	6497.000000
mean	0.056034	30.525319	115.744574	0.994697
std	0.035034	17.749400	56.521855	0.002999
min	0.009000	1.000000	6.000000	0.987110
25%	0.038000	17.000000	77.000000	0.992340
50%	0.047000	29.000000	118.000000	0.994890
75%	0.065000	41.000000	156.000000	0.996990
max	0.611000	289.000000	440.000000	1.038980

	pH	sulphates	alcohol	quality	alcohol_to_sugar \
count	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000
mean	3.218501	0.531268	10.491801	5.818378	3.935654
std	0.160787	0.148806	1.192712	0.873255	2.966427
min	2.720000	0.220000	8.000000	3.000000	0.177812
25%	3.110000	0.430000	9.500000	5.000000	1.230769
50%	3.210000	0.510000	10.300000	6.000000	3.531249
75%	3.320000	0.600000	11.300000	6.000000	5.882349
max	4.010000	2.000000	14.900000	9.000000	17.666637

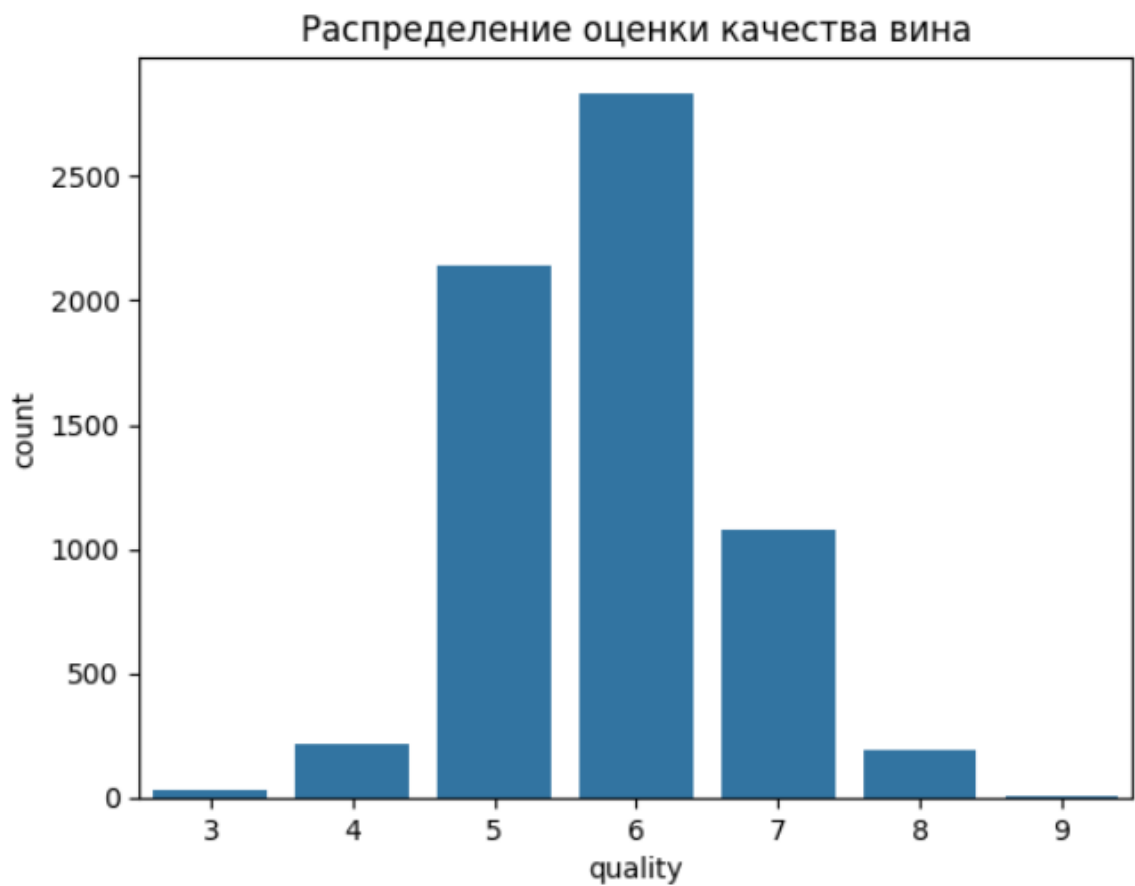
	acid_balance	total_acidity
count	6497.000000	6497.000000
mean	1.193659	7.873606
std	0.735549	1.388024
min	0.000000	4.130000
25%	0.717947	7.020000
50%	1.133330	7.600000
75%	1.599994	8.380000
max	8.299959	17.045000

```
# Распределение качества вина
```

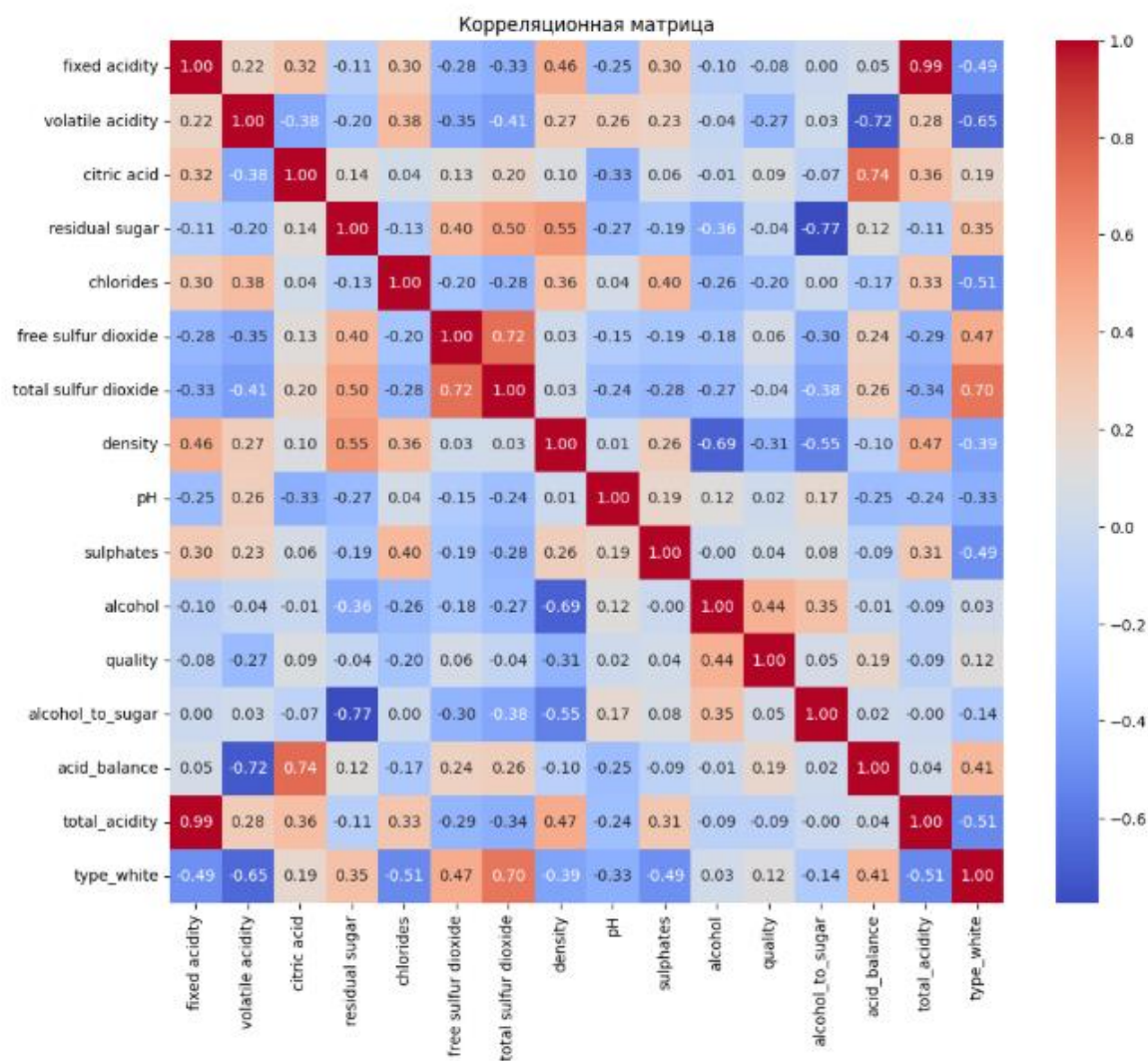
```
sns.countplot(x='quality', data=data)
```

```
plt.title('Распределение оценки качества вина')
```

```
plt.show()
```



```
# Корреляционная матрица
corr_matrix = data.corr()
plt.figure(figsize=(12, 10))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Корреляционная матрица')
plt.show()
```

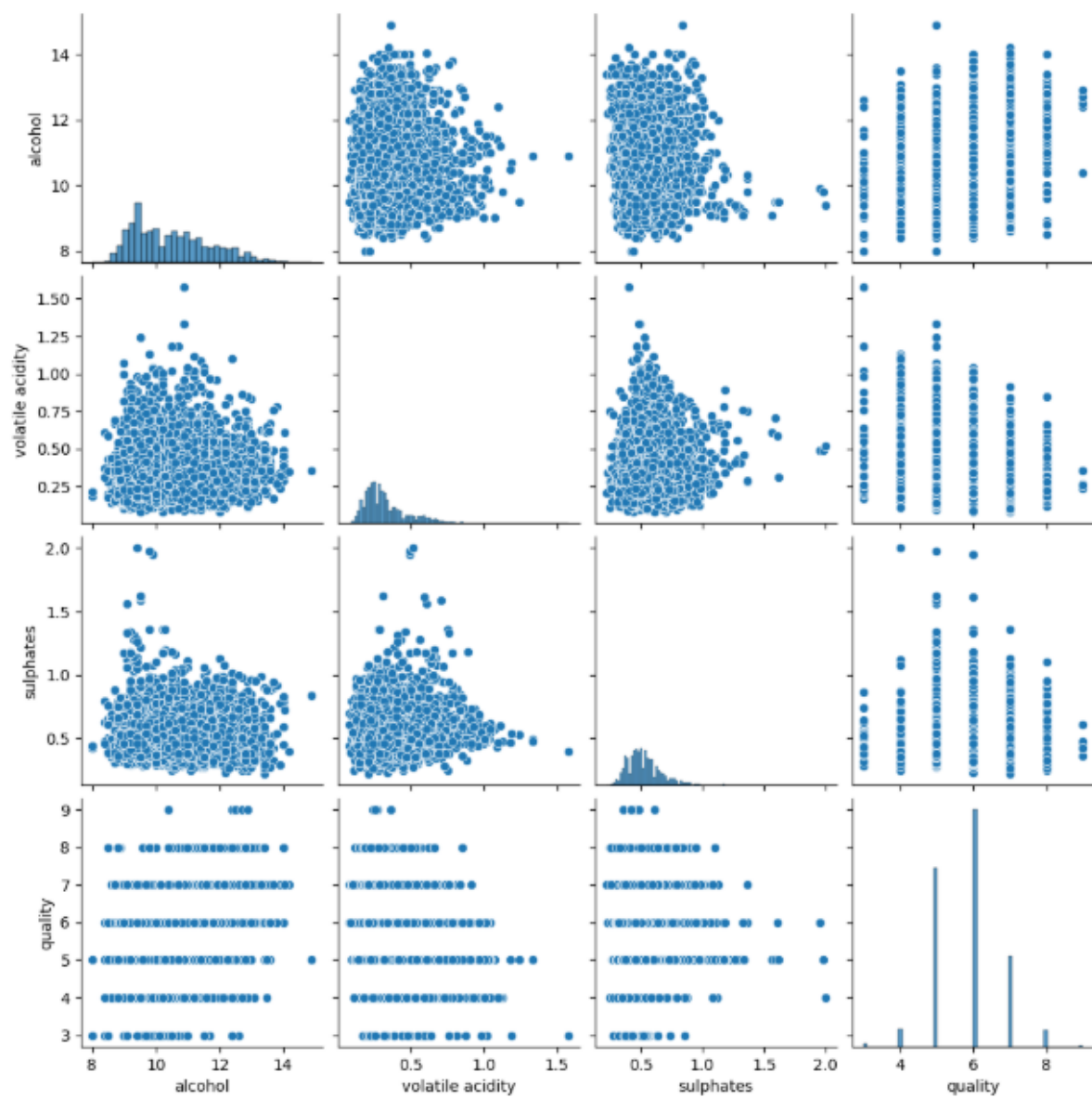


alcohol - самый сильный положительный фактор: чем выше содержание алкоголя, тем выше качество вина

volatile acidity - самый сильный отрицательный фактор: высокая летучая кислотность ухудшает вкус и

качество.

```
sns.pairplot(data[['alcohol', 'volatile acidity', 'sulphates', 'quality']])  
plt.show()
```



```
# Проверка на пропуски
print("Пропуски:\n", data.isnull().sum())

# Разделение на признаки и целевую переменную
X = data.drop('quality', axis=1)
y = data['quality']
```

```
Пропуски:
fixed acidity      0
volatile acidity   0
citric acid        0
residual sugar     0
chlorides          0
free sulfur dioxide 0
total sulfur dioxide 0
density           0
pH                0
sulphates         0
alcohol           0
quality           0
alcohol_to_sugar   0
acid_balance       0
total_acidity      0
type_white         0
dtype: int64
```

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

# Масштабирование данных
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Обучение моделей

Было обучено 5 моделей:

- Linear Regression
- Decision Tree Regressor
- Random Forest Regressor (ансамблевая)
- Gradient Boosting Regressor (ансамблевая)
- SVR (Support Vector Regression)

Для каждой модели рассчитаны метрики:

- MAE (Mean Absolute Error)
- MSE (Mean Squared Error)
- R^2 (R-squared)
- MAE — показывает среднюю ошибку, легко интерпретируется.
- MSE — учитывает большие ошибки сильнее, помогает выявить "провалы".
- R^2 — позволяет оценить, насколько модель лучше среднего значения .

Эти метрики дополняют друг друга: MAE — простота, MSE — чувствительность к большим ошибкам, R^2 — объяснение дисперсии. Все вместе позволяют объективно оценить качество модели в задаче регрессии.

Наиболее успешной моделью оказался Random Forest , показавший на тестовой выборке $MAE = 0.436$.

Код:

```
# Словарь моделей
```

```
models = {  
    "Linear Regression": LinearRegression(),  
    "Decision Tree": DecisionTreeRegressor(random_state=42),  
    "Random Forest": RandomForestRegressor(random_state=42),  
    "Gradient Boosting": GradientBoostingRegressor(random_state=42),  
    "SVR": SVR()  
}
```

```
results = { }
```

```
for name, model in models.items():  
    model.fit(X_train_scaled, y_train)  
    y_pred = model.predict(X_test_scaled)  
    results[name] = {  
        "MAE": mean_absolute_error(y_test, y_pred),  
        "MSE": mean_squared_error(y_test, y_pred),  
        "R2": r2_score(y_test, y_pred)  
    }
```

```
# Вывод результатов
```

```
baseline_results_df = pd.DataFrame(results).T  
baseline_results_df.sort_values(by="MAE")
```

]:	MAE	MSE	R2
Random Forest	0.437608	0.372752	0.495289
Decision Tree	0.504615	0.712308	0.035528
SVR	0.511569	0.457604	0.380399
Gradient Boosting	0.533472	0.461830	0.374678
Linear Regression	0.561619	0.536924	0.272999

Подбор гиперпараметров

Для улучшения качества модели Random Forest был выполнен подбор гиперпараметров с использованием **GridSearchCV**. Были исследованы следующие параметры:

- **n_estimators** (количество деревьев): [50, 100, 200]
- **max_depth** (максимальная глубина дерева): [None, 5, 10, 15]
- **min_samples_split**: [2, 4]
- **min_samples_leaf**: [1, 2]
- **max_features**: ['sqrt', 'log2']

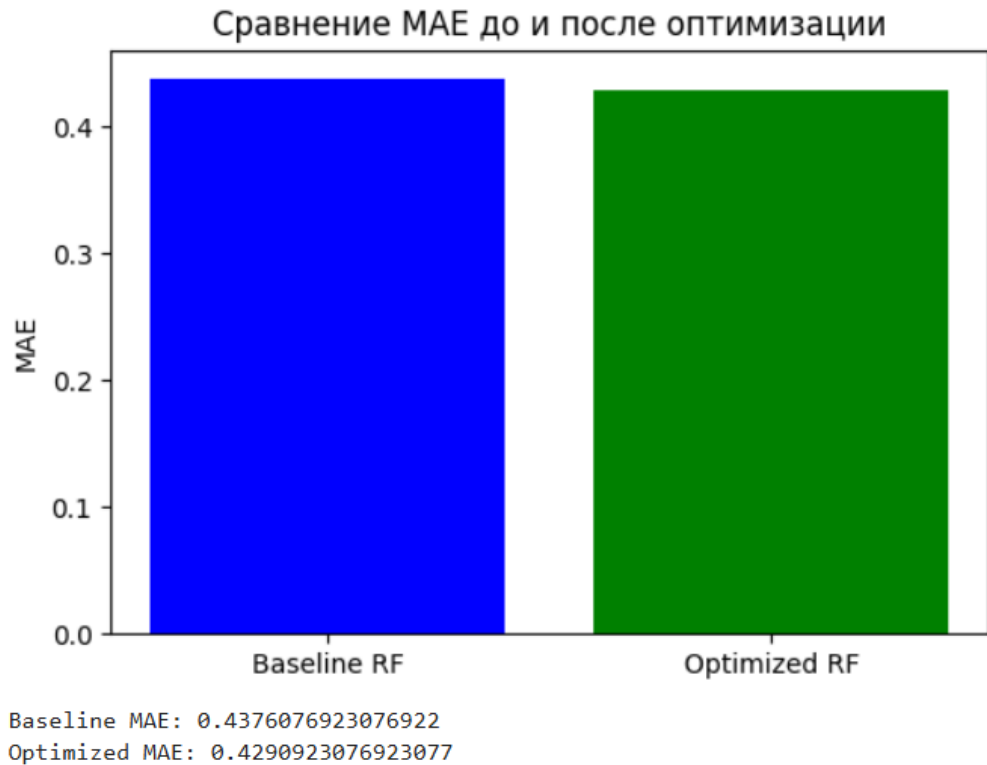
Лучшая комбинация:

```
{
  'bootstrap': True,
  'ccp_alpha': 0.0,
  'criterion': 'squared_error',
  'max_depth': None,
  'max_features': 'sqrt',
  'min_samples_leaf': 1,
  'min_samples_split': 2,
  'n_estimators': 200,
  'random_state': 42
}
```

После оптимизации качество модели немного улучшилось:

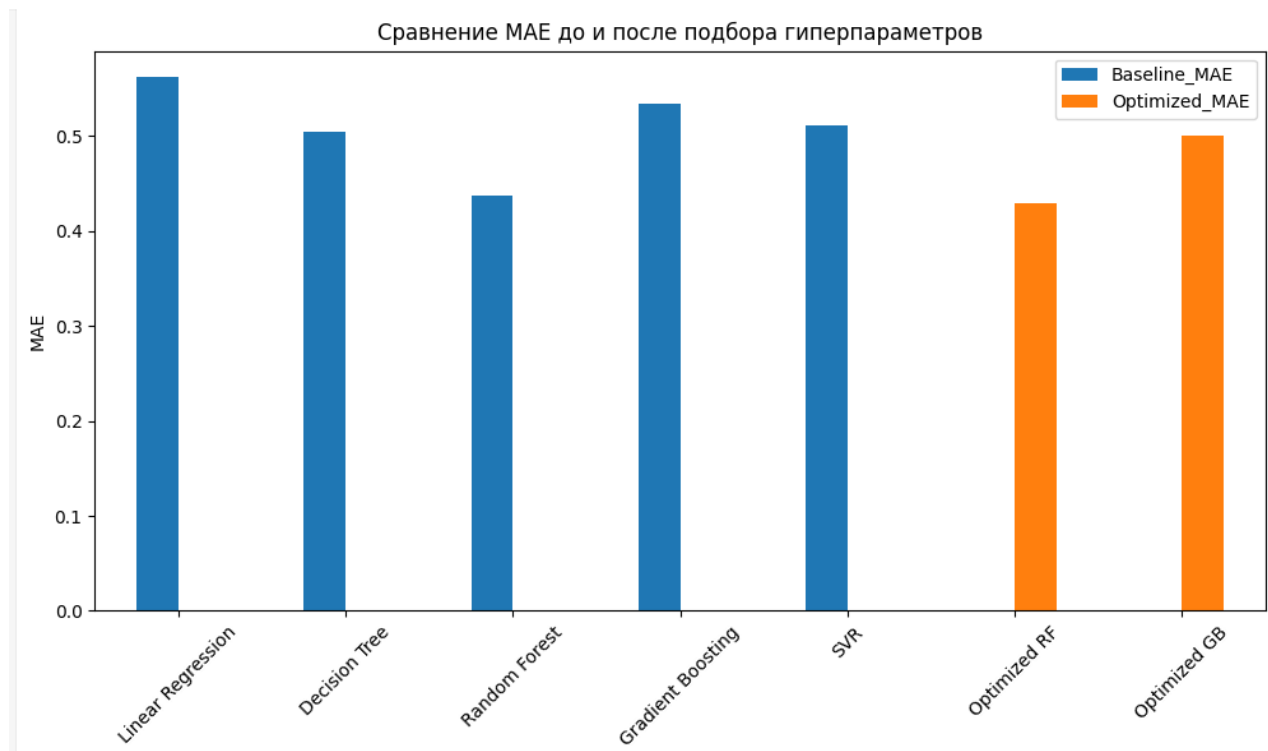
- Baseline MAE: 0.436

- Optimized MAE: 0.428



Сравним MAE после подбора гиперпараметров:

```
comparison_df = pd.concat([baseline_results_df, optimized_results_df], axis=1)
comparison_df.columns = ['Baseline_MAE', 'Baseline_MSE', 'Baseline_R2',
                        'Optimized_MAE', 'Optimized_MSE', 'Optimized_R2']
comparison_df[['Baseline_MAE', 'Optimized_MAE']].plot(kind='bar', figsize=(10,
6))
plt.title("Сравнение MAE до и после подбора гиперпараметров")
plt.ylabel("MAE")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



```
explainer = shap.TreeExplainer(best_rf)
```

```
shap_values = explainer.shap_values(X_test_scaled)
```

```
# Проверка данных
```

```
print("SHAP values shape:", np.array(shap_values).shape) # Должно быть  
(n_samples, n_features)
```

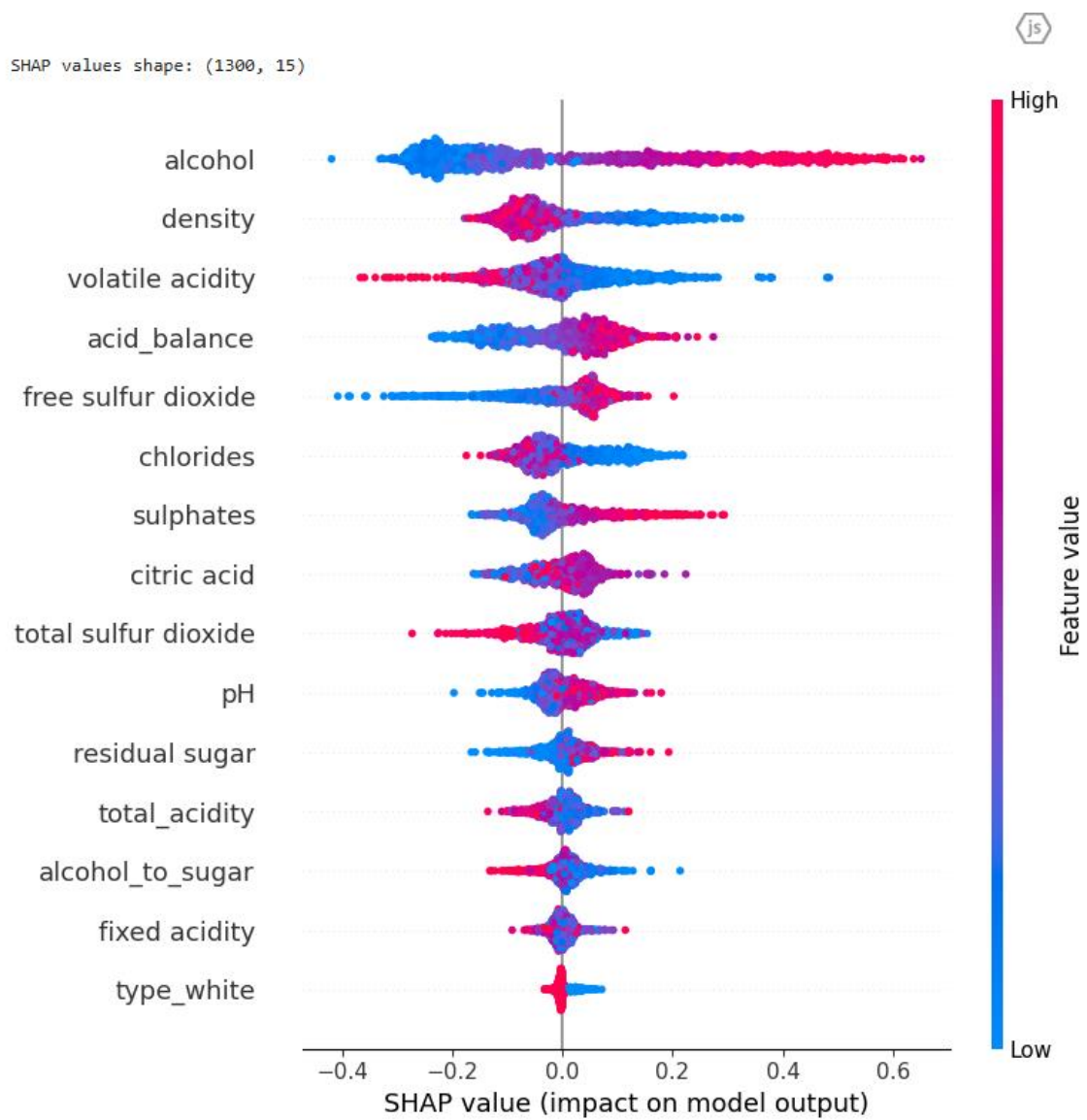
```
# Визуализация
```

```
shap.summary_plot(shap_values, X_test_scaled, feature_names=X.columns)
```

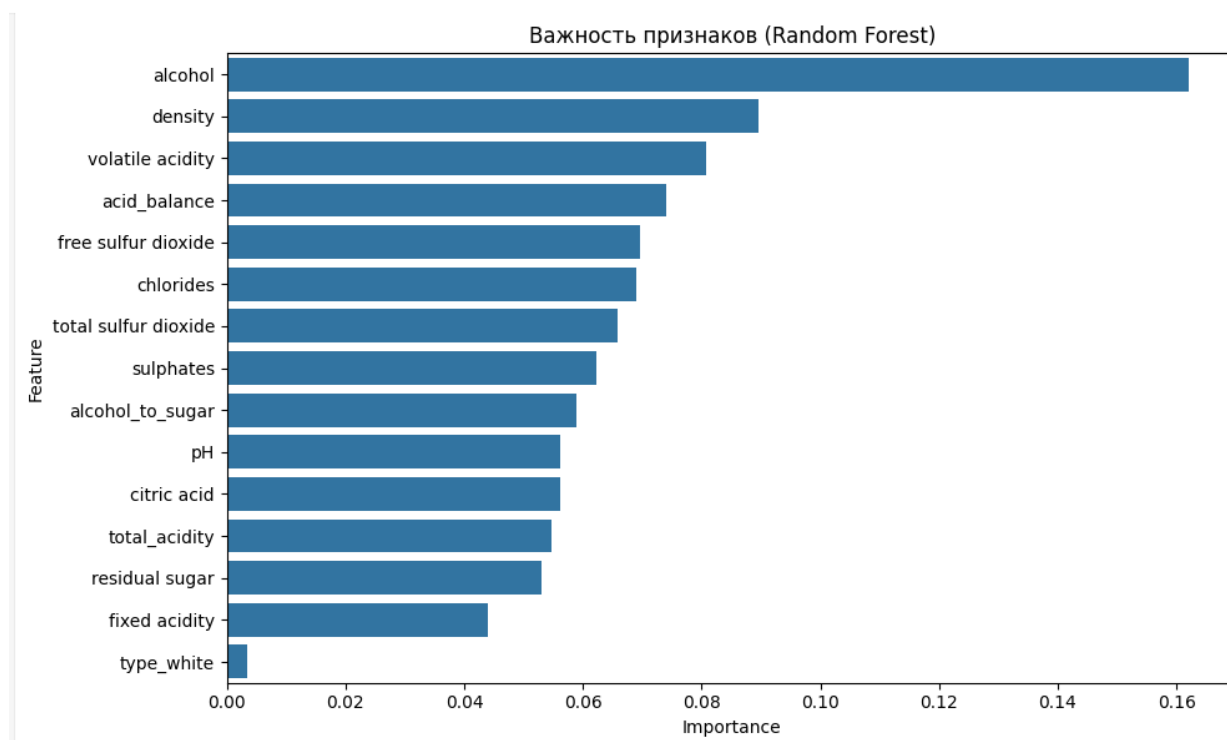
```
# Анализ отдельных предсказаний
```

```
sample_idx = 0 # Можно изменить на любой индекс из тестовой выборки
```

```
shap.force_plot(explainer.expected_value, shap_values[sample_idx, :],  
                X_test_scaled[sample_idx, :], feature_names=X.columns)
```



Анализ результатов



Анализ важности признаков показал, что наиболее значимыми факторами, влияющими на качество вина, являются:

- Содержание алкоголя (**alcohol**)
- Летучая кислотность (**volatile acidity**)
- Сульфаты (**sulphates**)

Графики важности признаков и сравнения MAE до и после оптимизации помогли наглядно продемонстрировать эффективность подбора гиперпараметров.

Разработка веб-приложения

Для демонстрации результатов была разработана интерактивная система с использованием фреймворка Streamlit. Она позволяет пользователю:



- Вводить параметры вина
- Менять гиперпараметры модели (**n_estimators**, **max_depth**)
- Получать прогноз качества вина в реальном времени


Интерфейс разделён на группы:

- Очень важные параметры
- Умеренно важные
- Менее важные

Каждый элемент содержит всплывающие подсказки, поясняющие влияние параметра на качество вина.

```
import streamlit as st
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

```
#  Установка конфигурации страницы — должно быть первой командой!
st.set_page_config(page_title=" Предсказание качества вина", layout="wide")
```

```
#  Функция загрузки и предобработки данных
```

```
@st.cache_data
```

```
def load_and_preprocess():
```

```
    red_wine = pd.read_csv('winequality-red.csv', sep=';')
```

```
    white_wine = pd.read_csv('winequality-white.csv', sep=';')
```

```
    red_wine['type'] = 'red'
```

```
    white_wine['type'] = 'white'
```

```
    data = pd.concat([red_wine, white_wine], axis=0).reset_index(drop=True)
```

```
    data = pd.get_dummies(data, columns=['type'], drop_first=True)
```

```
    X = data.drop('quality', axis=1)
```

```
    y = data['quality']
```

```
    scaler = StandardScaler()
```

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

```
return X_scaled, y, X.columns, scaler
```

```
# 🔄 Выполняем один раз — до запуска интерфейса
```

```
X, y, feature_names, scaler = load_and_preprocess()
```

```
# 🎨 Интерфейс Streamlit
```

```
st.title("🍷 Предсказание качества вина")
```

```
st.markdown("Выберите параметры вина для оценки его качества.")
```

```
# ⚙️ Настройка гиперпараметров модели
```

```
with st.sidebar:
```

```
    st.header("⚙️ Настройки модели")
```

```
    n_estimators = st.slider("Количество деревьев", 10, 200, 100, step=10)
```

```
    max_depth = st.slider("Максимальная глубина деревьев", 1, 20, 5)
```

```
# 🌲 Создаём модель
```

```
model = RandomForestRegressor(n_estimators=n_estimators,  
                             max_depth=max_depth, random_state=42)
```

```
# 📊 Разделение выборки
```

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

```
# 🖋️ Обучаем модель
```

```
model.fit(X_train, y_train)
```

```
# 🖋️ Пользовательский ввод с группировкой по важности
```

```
st.subheader("📊 Выберите параметры вина")
```

```
# 🔥 Очень важные параметры
```

```
st.markdown("#### 🔥 Очень важные параметры")
```

```
col1, col2 = st.columns(2)
```

```
with col1:
```

```
    alcohol = st.slider("Alcohol (%) — чем выше, тем лучше", 8.0, 14.9, 10.0,  
    help="Самый важный фактор качества вина")
```

```
with col2:
```

```
    volatile_acidity = st.slider("Volatile Acidity — чем ниже, тем лучше", 0.1, 1.6,  
    0.5, help="Летучая кислотность — нежелательна")
```

```
with col2:
```

```
    sulphates = st.slider("Sulphates — чем выше, тем лучше", 0.3, 2.0, 0.6,  
    help="Положительно влияет на вкус и стабильность")
```

```
# ⚖️ Умеренно важные
```

```
st.markdown("#### ⚖️ Умеренно важные параметры")
```

```
col1, col2 = st.columns(2)
```

```
with col1:
```

```
    citric_acid = st.slider("Citric Acid", 0.0, 1.0, 0.3)
```

```
    chlorides = st.slider("Chlorides — чем ниже, тем лучше", 0.01, 0.6, 0.05)
```

```
with col2:
```

```
    pH = st.slider("pH — оптимально ~3.1–3.3", 2.7, 4.0, 3.2)
```

```
    density = st.slider("Density", 0.99, 1.005, 0.995)
```

```
# 📏 Менее важные
```

```
st.markdown("#### 📏 Менее важные параметры")
```

```
col1, col2 = st.columns(2)
```

```
with col1:
```

```
fixed_acidity = st.slider("Fixed Acidity", 4.0, 16.0, 7.0)
residual_sugar = st.slider("Residual Sugar", 0.9, 15.5, 6.0)
free_sulfur_dioxide = st.slider("Free Sulfur Dioxide", 1, 72, 30)
with col2:
    total_sulfur_dioxide = st.slider("Total Sulfur Dioxide", 6, 289, 100)
    type_red = st.checkbox("Красное вино")
```

```
# 📊 Подготовка данных для предсказания
```

```
input_data = pd.DataFrame({
    'fixed acidity': [fixed_acidity],
    'volatile acidity': [volatile_acidity],
    'citric acid': [citric_acid],
    'residual sugar': [residual_sugar],
    'chlorides': [chlorides],
    'free sulfur dioxide': [free_sulfur_dioxide],
    'total sulfur dioxide': [total_sulfur_dioxide],
    'density': [density],
    'pH': [pH],
    'sulphates': [sulphates],
    'alcohol': [alcohol],
    'type_red': [1 if type_red else 0]
}, columns=feature_names)
```

```
# 🔍 Масштабируем введённые данные
```

```
input_data_scaled = scaler.transform(input_data)
```

```
# 🧠 Предсказание
```

```
if st.button("👉 Предсказать"):
    prediction = model.predict(input_data_scaled)[0]
```

```
st.success(f"**Предсказанное качество вина: {round(prediction, 2)} из 8**")
st.info(f"Текущие параметры модели: n_estimators={n_estimators},
max_depth={max_depth}")
```

Настройки модели

Количество деревьев

100

Максимальная глубина деревьев

5

Предсказание качества вина

Выберите параметры вина для оценки его качества.

Выберите параметры вина

Очень важные параметры

Alcohol (%) — чем выше, тем лучше

10.00

Volatile Acidity — чем ниже, тем лучше

0.50

Sulphates — чем выше, тем лучше

0.60

Умеренно важные параметры

Citric Acid

0.30

pH — оптимально ~3.1-3.3

3.20

Chlorides — чем ниже, тем лучше

0.05

Density

0.99

Менее важные параметры

Fixed Acidity

7.00

Total Sulfur Dioxide

100

Residual Sugar

6.00

Free Sulfur Dioxide

30

☐ Красное вино

Предсказать

Предсказанное качество вина: 5.48 из 8

Текущие параметры модели: n_estimators=100, max_depth=5

Заключение

Курсовая работа успешно реализовала задачу прогнозирования качества вина на основе его химического состава.

Основные результаты:

- Выполнен полный цикл ML-исследования: от загрузки данных до построения и сравнения моделей
- Обучены 5 моделей, из них 2 ансамблевые

- Подобраны гиперпараметры для Random Forest, что позволило улучшить качество модели
- Создано веб-приложение, позволяющее изменять параметры вина и гиперпараметры модели

Список использованных источников

Электронные ресурсы:

1. Датасет Wine Quality: <https://archive.ics.uci.edu/ml/datasets/wine+quality>
2. Документация Scikit-learn: <https://scikit-learn.org/stable/>
3. Streamlit Documentation: <https://docs.streamlit.io/>
4. GitHub репозиторий с примерами курсовых работ: https://github.com/ugapanyuk/courses_current
5. Библиотека Seaborn: <https://seaborn.pydata.org/>

Литература:

6. Python для анализа данных / Wes McKinney
7. Прикладной анализ данных с помощью Scikit-learn / Sarah Guido, Andreas C. Müller
8. Machine Learning с примерами на Python / Sebastian Raschka