Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Лабораторная работа №2
По дисциплине: «Основы машинного обучения»
Тема**: «Линейные модели**
**для задач регрессии и классификации»**

Выполнил:
Студент 3-го курса
Группы АС-65
Кисель М. С.
Проверил:
Крощенко А.А.

Брест 2025

**Цель работы:** Изучить применение **линейной** и **логистической регрессии** для решения практических задач. Научиться обучать модели, оценивать их качество с помощью соответствующих метрик и интерпретировать результаты.

## Ход работы

**Общее задание:** выполнить задания по варианту (регрессия и классификация), построить все требуемые визуализации и рассчитать метрики, **написать отчет, создать пул-реквест в репозиторий с кодом решения и отчетом в формате pdf**.

**Вариант 8**

**• Регрессия (Прогнозирование качества вина)**

1. Wine Quality

2. Предсказать оценку качества вина (quality) как непрерывную величину

3. **Задания:**

§ загрузите данные;

§ обучите модель **линейной регрессии** на всех доступных признаках;

§ рассчитайте **MSE** и **$R^2$**;

§ визуализируйте зависимость quality от alcohol с линией регрессии.

**• Классификация (Определение "хорошего" вина)**

1. Wine Quality

2. Классифицировать вино как "хорошее" (quality >= 7) или "плохое" (quality < 7)

3. **Задания:**

§ создайте новый бинарный целевой столбец на основе столбца quality;

§ обучите модель **логистической регрессии**;

§ рассчитайте **Accuracy**, **Precision** и **Recall** для класса "хорошее";

§ постройте **матрицу ошибок**.

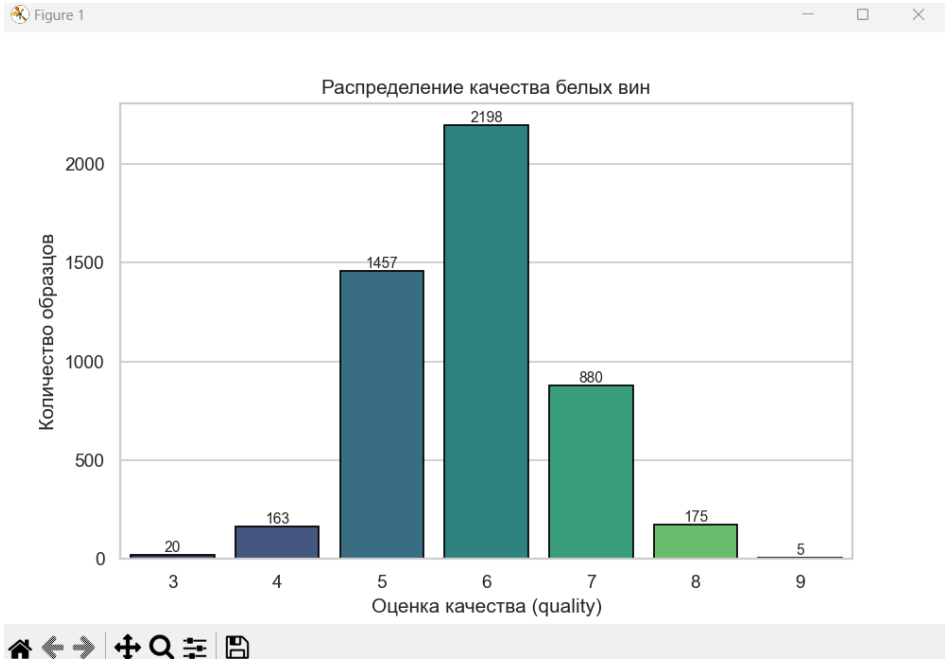
Код программы:

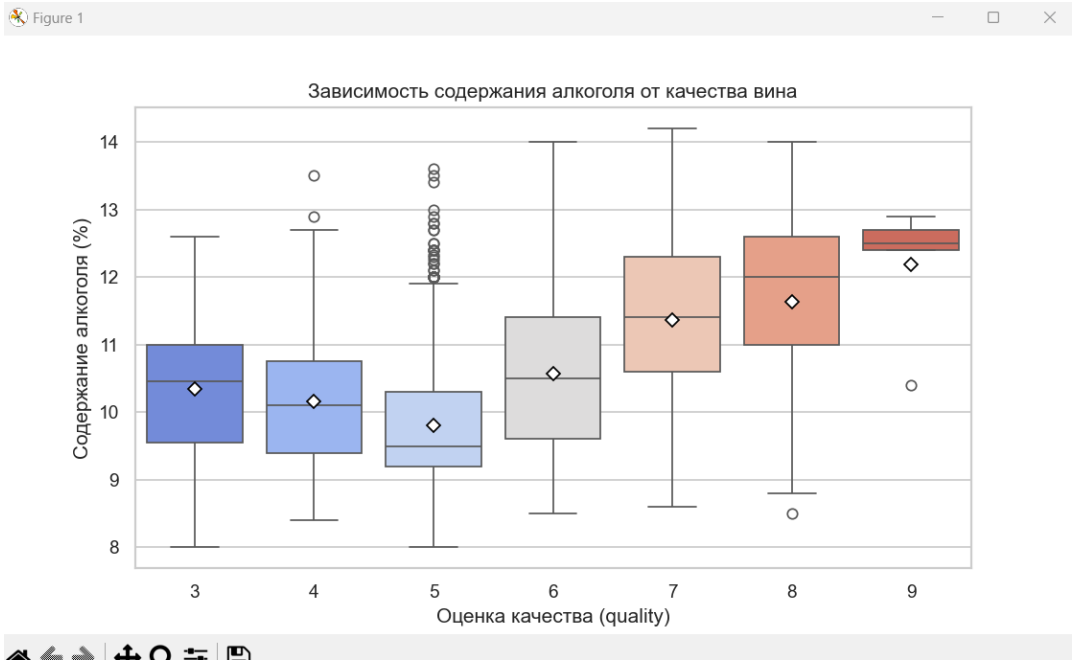
```
import os
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, confusion_matrix, ConfusionMatrixDisplay

sns.set(style="whitegrid")

os.chdir("d:/Универ/ОМО/ОМО2025/Лаба2/")
df = pd.read_csv("winequality-white.csv", sep=';')

print("Первые 5 строк:")
print(df.head())
print("\nИнформация о данных:")
print(df.info())
```

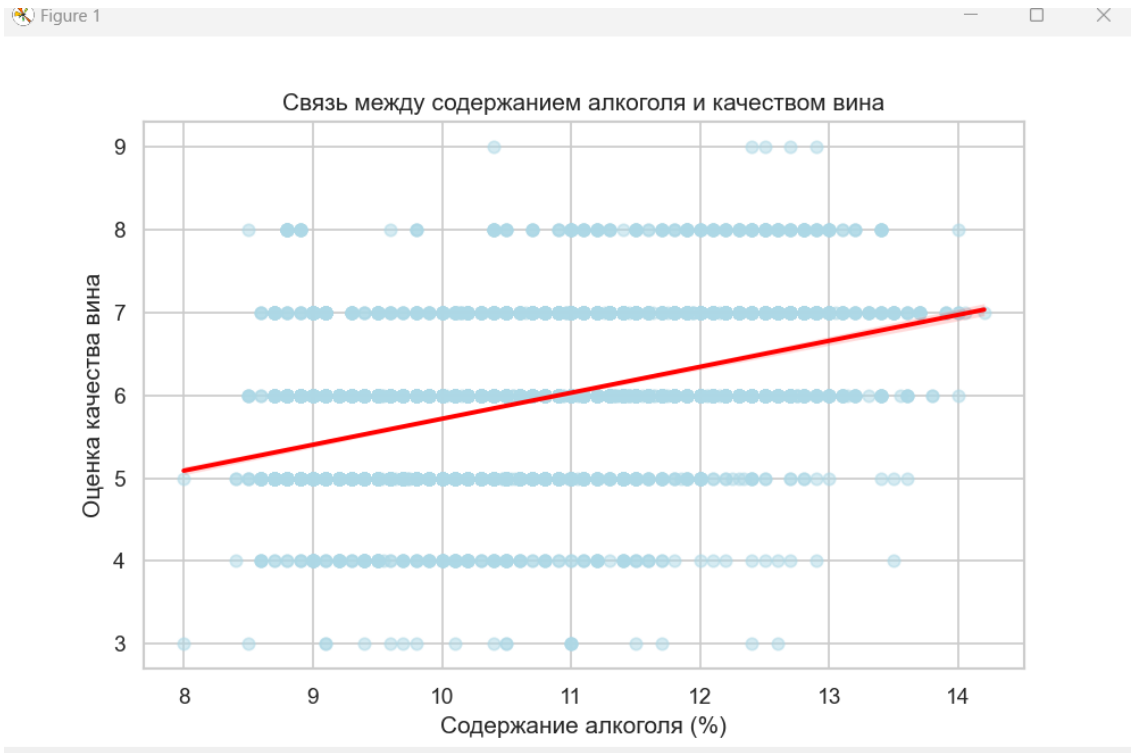
```
print("\nСтатистика по числовым признакам:")
print(df.describe())
print("\nПропуски по столбцам:")
print(df.isnull().sum())

# COUNTPLOT — количество наблюдений в каждой ка тегории quality
plt.figure(figsize=(8, 5))
quality_order = sorted(df['quality'].unique())
ax = sns.countplot(x='quality', data=df, palette='viridis', order=quality_order, edgecolor='black')
plt.title('Распределение качества белых вин')
plt.xlabel('Оценка качества (quality)')
plt.ylabel('Количество образцов')

# Подписи над столбиками (значения count)
for p in ax.patches:
    height = p.get_height()
    ax.annotate(f'{int(height)}', (p.get_x() + p.get_width() / 2, height),
                ha='center', va='bottom', fontsize=9)
plt.show()

# BOXPLOT — распределение alcohol в каждой категории quality
plt.figure(figsize=(9, 5))
ax2 = sns.boxplot(x='quality', y='alcohol', data=df, palette='coolwarm', order=quality_order,
                                showmeans=True,  meanprops={"marker":"D",  "markeredgecolor":"black",
"markerfacecolor":"white"})
plt.title('Зависимость содержания алкоголя от качества вина')
plt.xlabel('Оценка качества (quality)')
plt.ylabel('Содержание алкоголя (%)')
plt.show()

plt.figure(figsize=(8, 5))

# Диаграмма рассеяния + линия регрессии
sns.regplot(
    x='alcohol',
    y='quality',
    data=df,
    scatter_kws={'alpha': 0.5, 'color': 'lightblue'},
    line_kws={'color': 'red'}
)

plt.title('Связь между содержанием алкоголя и качеством вина')
plt.xlabel('Содержание алкоголя (%)')
plt.ylabel('Оценка качества вина')
plt.show()

# Разделяем признаки (X) и целевую переменную (y)
X = df.drop('quality', axis=1)
y = df['quality']

# Разделяем выборку на обучающую и тестовую (80% / 20%)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Создаём и обучаем модель
```

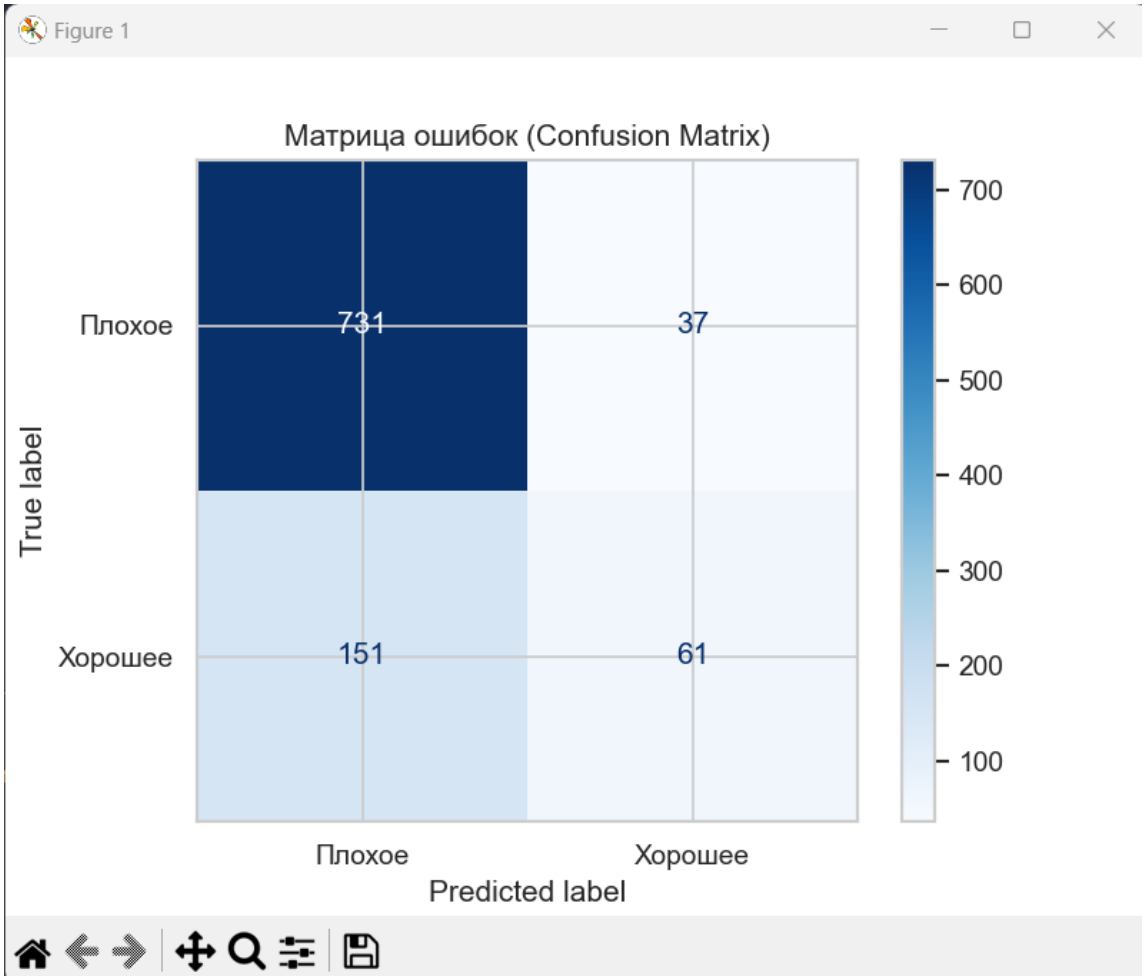
```
model = LinearRegression()
model.fit(X_train, y_train)

# Делаем предсказания на тестовой выборке
y_pred = model.predict(X_test)

# Вычисляем метрики качества модели
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Среднеквадратичная ошибка (MSE):", round(mse, 3))
print("Коэффициент детерминации (R²):", round(r2, 3))

# Смотрим, какие признаки сильнее влияют на качество
coefficients = pd.DataFrame({'Признак': X.columns, 'Коэффициент': model.coef_})
print("\nВлияние признаков на качество вина:")
print(coefficients.sort_values(by='Коэффициент', ascending=False))

# Создаём новый бинарный столбец: 1 — хорошее, 0 — плохое
df['good'] = (df['quality'] >= 7).astype(int)
print(df[['quality', 'good']].head(10))

# Разделяем признаки (X) и целевую переменную (y)
X = df.drop(['quality', 'good'], axis=1)
y = df['good']

# Разделяем выборку на обучающую и тестовую (80% / 20%)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

# Обучаем модель логистической регрессии
log_model = LogisticRegression(max_iter=1000)
log_model.fit(X_train, y_train)

# Предсказания на тестовых данных
y_pred = log_model.predict(X_test)

# Метрики качества классификации
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)

print("\nМетрики качества модели:")
print(f"Accuracy (доля правильных предсказаний): {acc:.3f}")
print(f"Precision (точность для класса 'хорошее'): {prec:.3f}")
print(f"Recall (полнота для класса 'хорошее'): {rec:.3f}")

# Матрица ошибок
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=["Плохое", "Хорошее"])
disp.plot(cmap="Blues")
plt.title("Матрица ошибок (Confusion Matrix)")
plt.show()
```

Графики и матрица после выполнения программы:






## Консольный вывод:

Первые 5 строк:

| | fixed acidity | volatile acidity | citric acid | residual sugar | chlorides | free sulfur dioxide | total sulfur dioxide | density | pH | sulphates | alcohol | quality |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 7.0 | 0.27 | 0.36 | 20.7 | 0.045 | 45.0 | 170.0 | 1.0010 | 3.00 | 0.45 | 8.8 | 6 |
| 1 | 6.3 | 0.30 | 0.34 | 1.6 | 0.049 | 14.0 | 132.0 | 0.9940 | 3.30 | 0.49 | 9.5 | 6 |
| 2 | 8.1 | 0.28 | 0.40 | 6.9 | 0.050 | 30.0 | 97.0 | 0.9951 | 3.26 | 0.44 | 10.1 | 6 |
| 3 | 7.2 | 0.23 | 0.32 | 8.5 | 0.058 | 47.0 | 186.0 | 0.9956 | 3.19 | 0.40 | 9.9 | 6 |
| 4 | 7.2 | 0.23 | 0.32 | 8.5 | 0.058 | 47.0 | 186.0 | 0.9956 | 3.19 | 0.40 | 9.9 | 6 |

Информация о данных:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 4898 entries, 0 to 4897

Data columns (total 12 columns):

| # | Column | Non-Null Count | Dtype |
|---|---|---|---|
| 0 | fixed acidity | 4898 non-null | float64 |
| 1 | volatile acidity | 4898 non-null | float64 |
| 2 | citric acid | 4898 non-null | float64 |
| 3 | residual sugar | 4898 non-null | float64 |
| 4 | chlorides | 4898 non-null | float64 |
| 5 | free sulfur dioxide | 4898 non-null | float64 |
| 6 | total sulfur dioxide | 4898 non-null | float64 |
| 7 | density | 4898 non-null | float64 |
| 8 | pH | 4898 non-null | float64 |
| 9 | sulphates | 4898 non-null | float64 |

```
 10  alcohol              4898 non-null   float64
 11  quality              4898 non-null   int64
dtypes: float64(11), int64(1)
memory usage: 459.3 KB
None
```

Статистика по числовым признакам:

| | fixed acidity | volatile acidity | citric acid | residual sugar | chlorides | free sulfur dioxide | total sulfur dioxide | density | pH | sulphates | alcohol | quality |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| count | 4898.000000 | 4898.000000 | 4898.000000 | 4898.000000 | 4898.000000 | 4898.000000 | 4898.000000 | 4898.000000 | 4898.000000 | 4898.000000 | 4898.000000 | 4898.000000 |
| mean | 6.854788 | 0.278241 | 0.334192 | 6.391415 | 0.045772 | 35.308085 | 138.360657 | 0.994027 | 3.188267 | 0.489847 | 10.514267 | 5.877909 |
| std | 0.843868 | 0.100795 | 0.121020 | 5.072058 | 0.021848 | 17.007137 | 42.498065 | 0.002991 | 0.151001 | 0.114126 | 1.230621 | 0.885639 |
| min | 3.800000 | 0.080000 | 0.000000 | 0.600000 | 0.009000 | 2.000000 | 9.000000 | 0.987110 | 2.720000 | 0.220000 | 8.000000 | 3.000000 |
| 25% | 6.300000 | 0.210000 | 0.270000 | 1.700000 | 0.036000 | 23.000000 | 108.000000 | 0.991723 | 3.090000 | 0.410000 | 9.500000 | 5.000000 |
| 50% | 6.800000 | 0.260000 | 0.320000 | 5.200000 | 0.043000 | 34.000000 | 134.000000 | 0.993740 | 3.180000 | 0.470000 | 10.400000 | 6.000000 |
| 75% | 7.300000 | 0.320000 | 0.390000 | 9.900000 | 0.050000 | 46.000000 | 167.000000 | 0.996100 | 3.280000 | 0.550000 | 11.400000 | 6.000000 |
| max | 14.200000 | 1.100000 | 1.660000 | 65.800000 | 0.346000 | 289.000000 | 440.000000 | 1.038980 | 3.820000 | 1.080000 | 14.200000 | 9.000000 |

Пропуски по столбцам:

```
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
dtype: int64
```

Среднеквадратичная ошибка (MSE): 0.569

Коэффициент детерминации ($R^2$): 0.265

Влияние признаков на качество вина:

| | Признак | Коэффициент |
|---|---|---|
| 9 | sulphates | 0.649073 |
| 8 | pH | 0.600700 |
| 10 | alcohol | 0.229009 |
| 3 | residual sugar | 0.071240 |
| 0 | fixed acidity | 0.045907 |
| 5 | free sulfur dioxide | 0.005119 |
| 6 | total sulfur dioxide | -0.000242 |
| 4 | chlorides | -0.026475 |
| 2 | citric acid | -0.061303 |
| 1 | volatile acidity | -1.914884 |
| 7 | density | -124.264125 |

| | quality | good |
|---|---|---|
| 0 | 6 | 0 |
| 1 | 6 | 0 |
| 2 | 6 | 0 |
| 3 | 6 | 0 |
| 4 | 6 | 0 |
| 5 | 6 | 0 |
| 6 | 6 | 0 |
| 7 | 6 | 0 |
| 8 | 6 | 0 |
| 9 | 6 | 0 |

Метрики качества модели:

Accuracy (доля правильных предсказаний): 0.808

Precision (точность для класса 'хорошее'): 0.622

Recall (полнота для класса 'хорошее'): 0.288

Вывод: Линейная регрессия применяется в случае числовых значений(качество вина), а логистическая регрессия – когда нужна классификация по принципу да/нет(хорошее/плохое вино)