

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.

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:

1. загрузите данные;
2. обучите модель **линейной регрессии** на всех доступных признаках;
3. рассчитайте **MSE** и **R²**;
4. 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:

1. создайте новый бинарный целевой столбец на основе столбца quality;
2. обучите модель **логистической регрессии**;
3. рассчитайте **Accuracy**, **Precision** и **Recall** для класса "хорошее";
4. постройте **матрицу ошибок**.

Регрессия (Прогнозирование качества вина):

1. Загрузите данные

```
import os
import pandas as pd

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())
```

2. обучите модель **линейной регрессии** на всех доступных признаках

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

```

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)

```

3. рассчитайте MSE и R²;

```

# 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))

```

```

Среднеквадратичная ошибка (MSE): 0.569
Коэффициент детерминации (R²): 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

4. визуализируйте зависимость quality от alcohol с линией регрессии.

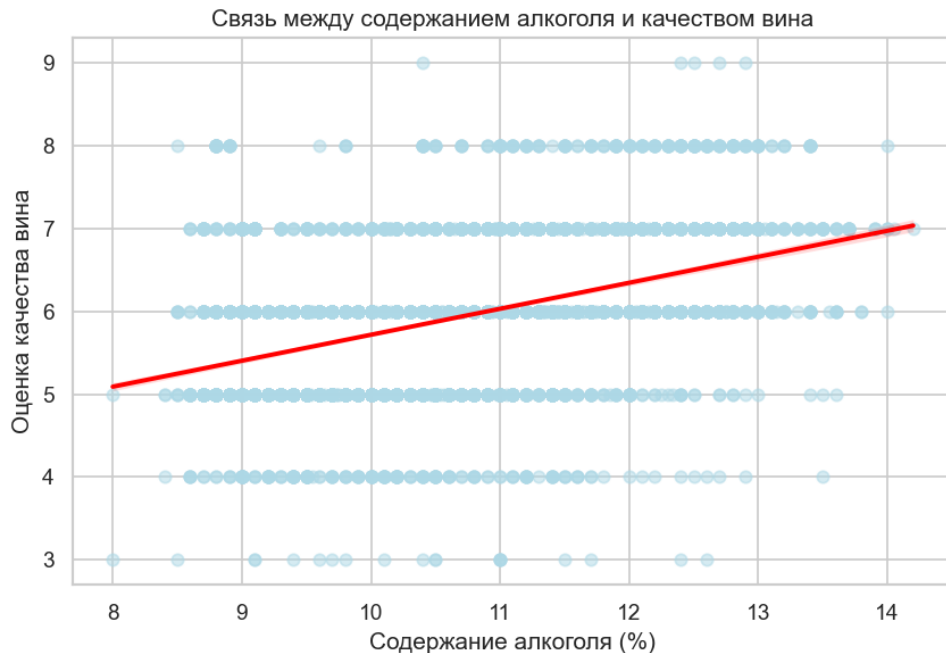
```

import matplotlib.pyplot as plt
import seaborn as sns

# 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()

```



Классификация (Определение "хорошего" вина)

- создайте новый бинарный целевой столбец на основе столбца quality;

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

	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

- обучите модель логистической регрессии

```
from sklearn.linear_model import LogisticRegression

# 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)
```

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

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, confusion_matrix, ConfusionMatrixDisplay

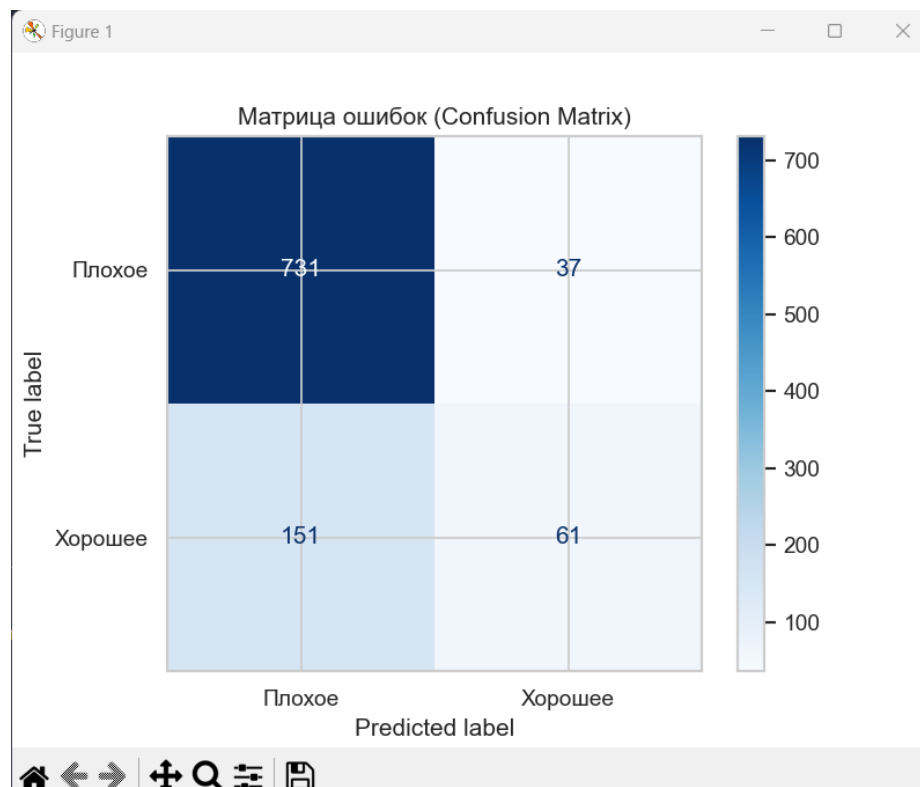
# Метрики качества классификации
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)

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}")
```

```
Метрики качества модели:
Accuracy (доля правильных предсказаний): 0.808
Precision (точность для класса 'хорошее'): 0.622
Recall (полнота для класса 'хорошее'): 0.288
```

5. постройте **матрицу ошибок**.

```
# 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()
```



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