Министерство образования Республики Беларусь  
Учреждение образования

«Брестский государственный технический университет»

ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ  
Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №2

Выполнил

В.Д.Головкина,

студент группы АС66

Проверил

А. А. Крощенко,

ст. преп. кафедры ИИТ,

« \_\_ » \_\_\_\_\_\_\_\_\_\_2025 г.

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.

Вариант 1

• Регрессия (Прогнозирование стоимости жилья в Калифорнии)

1. California Housing

2. Предсказать медианную стоимость дома (median\_house\_value)

3. Задания:

§ загрузите данные и разделите их на обучающую и

тестовую выборки;

§ обучите модель линейной регрессии на обучающих

данных;

§ сделайте предсказания для тестовой выборки;

§ оцените качество модели, рассчитав метрики MSE (Mean

Squared Error) и R2 (Coefficient of Determination);

§ визуализируйте результат: постройте диаграмму

рассеяния для признака median\_income (медианный

доход) и целевой переменной, нанеся на неё линию

регрессии.

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.datasets import fetch\_california\_housing

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

# 1. Загрузка и подготовка данных

data = fetch\_california\_housing(as\_frame=True)

df = data.frame

X = df.drop(columns='MedHouseVal')

y = df['MedHouseVal']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# 2. Обучение модели

model = LinearRegression()

model.fit(X\_train, y\_train)

# 3. Предсказания

y\_pred = model.predict(X\_test)

# 4. Оценка качества

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print(f"MSE: {mse:.2f}")

print(f"R²: {r2:.2f}")

# 5. Визуализация зависимости от median\_income

plt.figure(figsize=(8, 6))

sns.scatterplot(x=X\_test['MedInc'], y=y\_test, label='Фактические значения')

# Сортировка данных

X\_test\_for\_plot = X\_test.sort\_values(by='MedInc')

y\_pred\_for\_plot = model.predict(X\_test\_for\_plot)

sns.lineplot(x=X\_test\_for\_plot['MedInc'], y=y\_pred\_for\_plot, color='red', label='Линия регрессии')

plt.xlabel('Median Income')

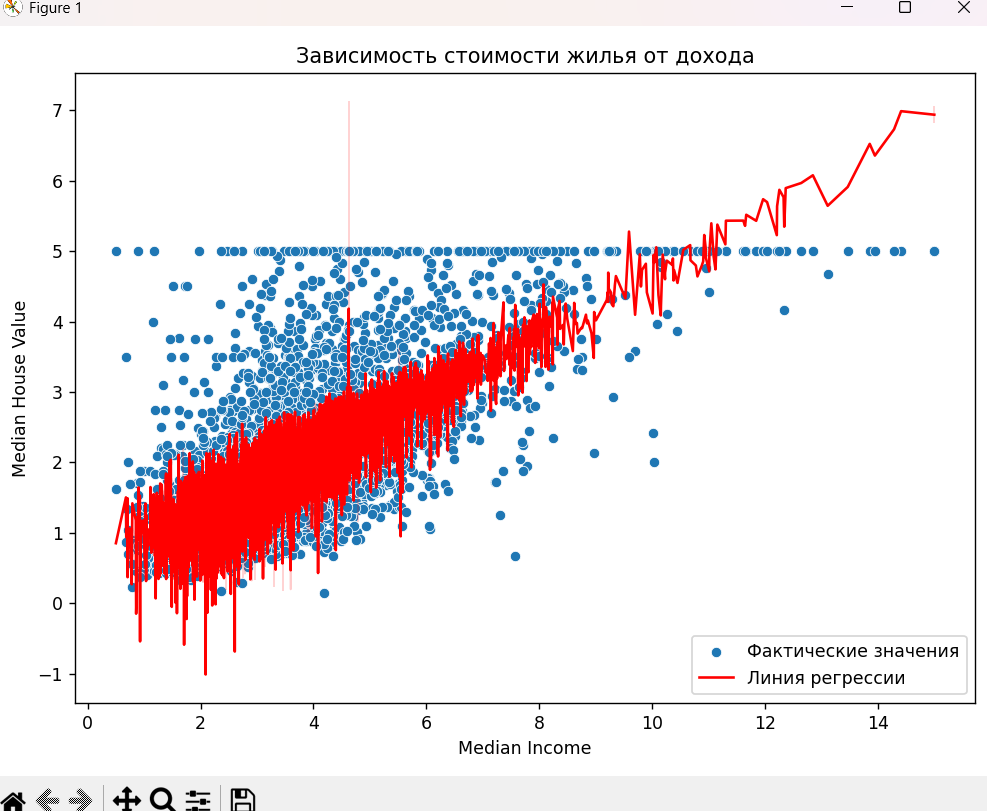
plt.ylabel('Median House Value')

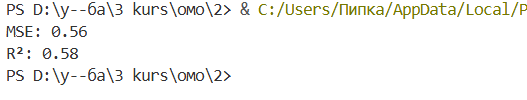
plt.title('Зависимость стоимости жилья от дохода')

plt.legend()

plt.tight\_layout()

plt.show()





Классификация (Прогнозирование выживаемости на

"Титанике")

1. Titanic

2. Предсказать, выжил ли пассажир (Survived)

3. Задания:

§ загрузите и предварительно обработайте данные

(заполните пропуски, преобразуйте категории в числа);

§ обучите модель логистической регрессии;

§ оцените качество модели,

рассчитав Accuracy, Precision и Recall;

§ постройте и проанализируйте матрицу

ошибок (confusion matrix).

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import LabelEncoder, StandardScaler # Добавлен StandardScaler

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, confusion\_matrix

# 1. Загрузка данных из Google Drive (используем прямую ссылку)

url = 'https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv'  # Прямая ссылка на .csv

df = pd.read\_csv(url)

# 2. Предобработка данных

df['Age'] = df['Age'].fillna(df['Age'].median())

df['Embarked'] = df['Embarked'].fillna(df['Embarked'].mode()[0])

for col in ['Sex', 'Embarked']:

    df[col] = LabelEncoder().fit\_transform(df[col])

# 3. Выбор признаков и разделение

features = ['Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare', 'Embarked']

X = df[features]

y = df['Survived']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# 4. Масштабирование данных

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# 5. Обучение модели

model = LogisticRegression(max\_iter=1000)

model.fit(X\_train, y\_train)

# 6. Предсказания и метрики

y\_pred = model.predict(X\_test)

print(f"Accuracy: {accuracy\_score(y\_test, y\_pred):.2f}")

print(f"Precision: {precision\_score(y\_test, y\_pred):.2f}")

print(f"Recall: {recall\_score(y\_test, y\_pred):.2f}")

# 7. Матрица ошибок

cm = confusion\_matrix(y\_test, y\_pred)

plt.figure(figsize=(6, 4))

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')

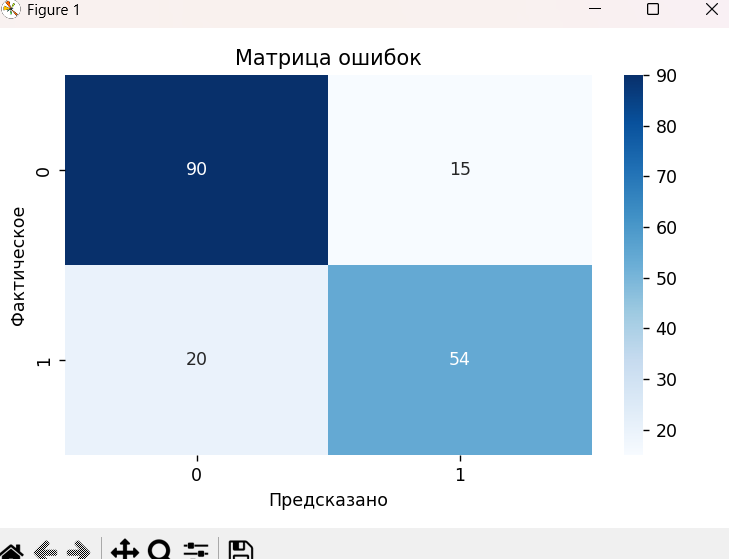
plt.xlabel('Предсказано')

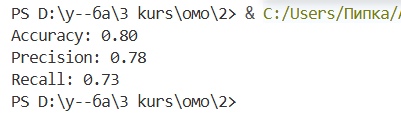
plt.ylabel('Фактическое')

plt.title('Матрица ошибок')

plt.tight\_layout()

plt.show()





Вывод: я изучила применение линейной и логистической регрессии для

решения практических задач. Научилась обучать модели, оценивать их

качество с помощью соответствующих метрик и интерпретировать

результаты.