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;

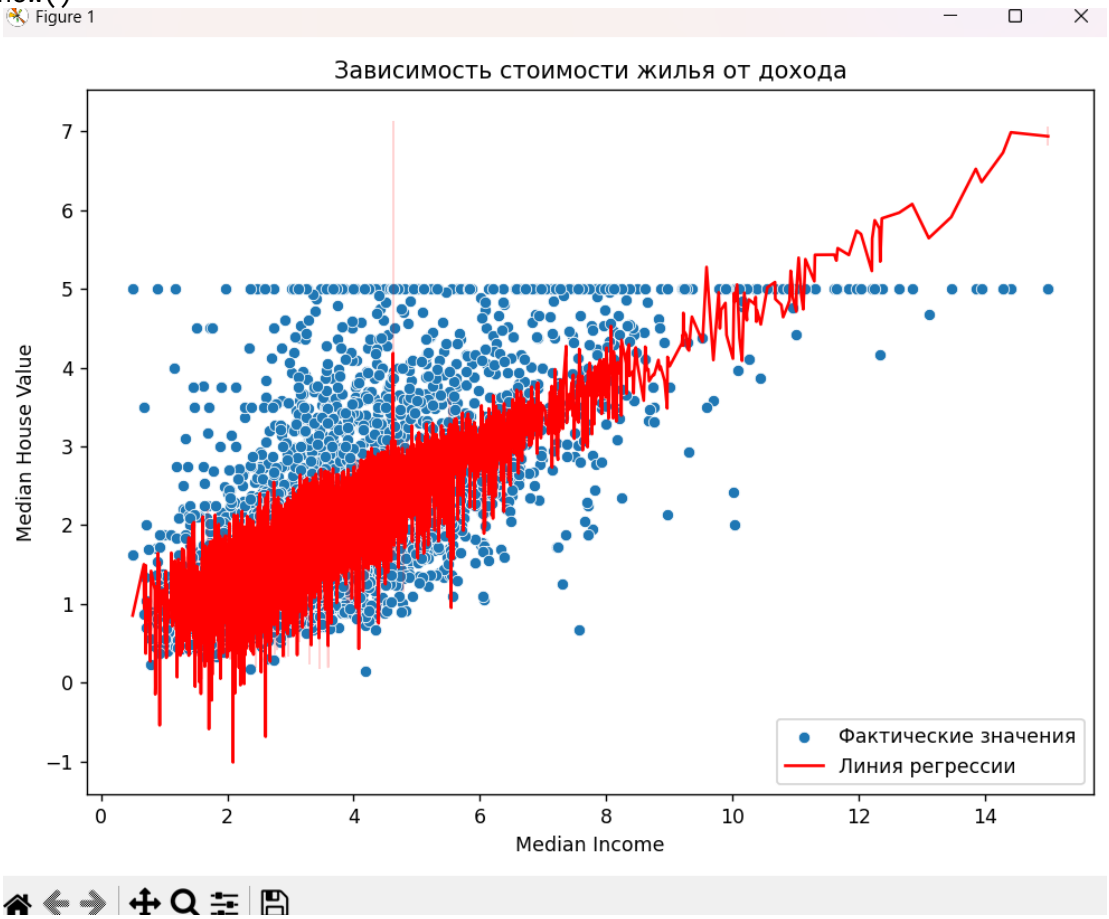
§ сделайте предсказания для тестовой выборки;

§ 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.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from pathlib import Path

local_file = Path(r"D:\y--6a\3 kurs\omo\2\california_housing.csv")

if not local_file.exists():
    raise FileNotFoundError(f" Файл не найден: {local_file}")

try:
    df = pd.read_csv(local_file)
    print(f" Данные успешно загружены из локального файла: {local_file}")
except Exception as e:
    raise RuntimeError(f"Ошибка при чтении файла: {e}")

target_column = "median_house_value"
if target_column not in df.columns:
    raise ValueError(f" Столбец '{target_column}' не найден в датасете.")

# Уд нечисловых призн
df_numeric = df.select_dtypes(include=["number"])

df_filled = df_numeric.fillna(df_numeric.median(numeric_only=True))

X = df_filled.drop(columns=target_column)
y = df_filled[target_column]
```

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

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

y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"\n MSE: {mse:.2f}")
print(f" R²: {r2:.2f}")

if "median_income" in X_test.columns:
    plt.figure(figsize=(8, 6))
    sns.scatterplot(x=X_test["median_income"], y=y_test, label="Фактические значения")

    plot_data = pd.DataFrame({
        "median_income": X_test["median_income"],
        "Predicted": y_pred
    }).sort_values("median_income")

    sns.lineplot(x=plot_data["median_income"], y=plot_data["Predicted"], color="red", la-
bel="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()
```

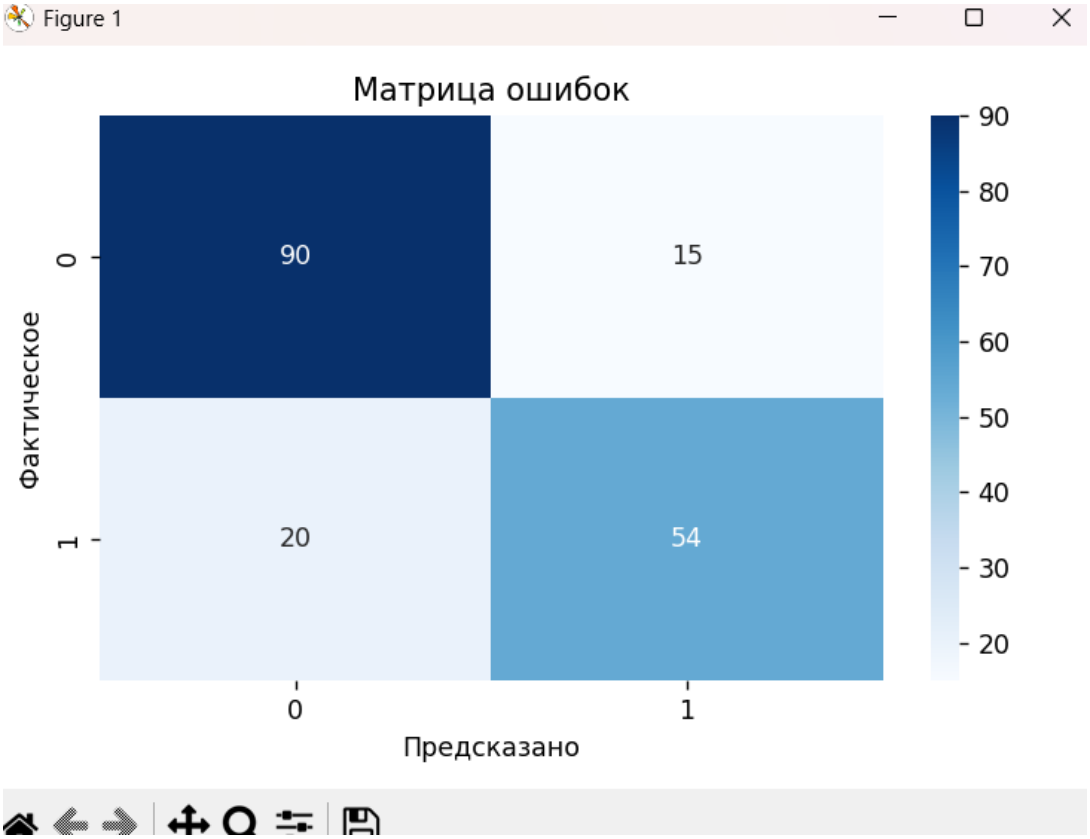


```
Данные успешно загружен

MSE: 5059928371.17
R²: 0.61
PS C:\Users\Пипка>
```

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
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, confusion_ma-
trix
from pathlib import Path

file_path = Path(r"D:\y--6a\3 kurs\omo\2\Titanic-Dataset.csv")
if not file_path.exists():
    raise FileNotFoundError(f" Файл не найден: {file_path}")

df = pd.read_csv(file_path)
print(f" Данные успешно загружены из: {file_path}")

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])

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)

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

model = LogisticRegression(max_iter=1000)
```

```
model.fit(X_train, y_train)

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}")

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()
```




```
PS D:\y--6a\3 kurs\omo\2> & C:/Users/Пипка/
Accuracy: 0.80
Precision: 0.78
Recall: 0.73
PS D:\y--6a\3 kurs\omo\2>
```

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.