

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №2

Специальность АС-66

Выполнила
А. А. Петrusевич,
студентка группы АС-66

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.

Код программы:

```
import pandas as pd
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

df = pd.read_csv("california_housing.csv")
df.dropna(inplace=True)
df["ocean_proximity"] =
df["ocean_proximity"].astype("category").cat.codes

y = df["median_house_value"]

X_single = df[["median_income"]]
X_train_s, X_test_s, y_train_s, y_test_s = train_test_split(X_single,
y, test_size=0.2, random_state=42)
model_single = LinearRegression()
model_single.fit(X_train_s, y_train_s)
y_pred_s = model_single.predict(X_test_s)
mse_s = mean_squared_error(y_test_s, y_pred_s)
r2_s = r2_score(y_test_s, y_pred_s)

X_multi = df.drop(columns=["median_house_value"])
X_train_m, X_test_m, y_train_m, y_test_m = train_test_split(X_multi,
y, test_size=0.2, random_state=42)
model_multi = LinearRegression()
model_multi.fit(X_train_m, y_train_m)
y_pred_m = model_multi.predict(X_test_m)
mse_m = mean_squared_error(y_test_m, y_pred_m)
r2_m = r2_score(y_test_m, y_pred_m)

print("Однопараметрическая модель:")
print(f" MSE: {mse_s:.2f}")
print(f" R2: {r2_s:.2f}")
print("\nМногопараметрическая модель:")
print(f" MSE: {mse_m:.2f}")
print(f" R2: {r2_m:.2f}")
```

```

plt.figure(figsize=(10, 6))
plt.scatter(X_test_s, y_test_s, alpha=0.3, label="Фактические
значения")
plt.plot(X_test_s, y_pred_s, color="red", label="Линия регрессии")
plt.xlabel("Median Income")
plt.ylabel("Median House Value")
plt.title("Однопараметрическая регрессия: Доход vs Стоимость жилья")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

```

убери комментарии Рисунки с результатами работы программы:

Однопараметрическая модель:

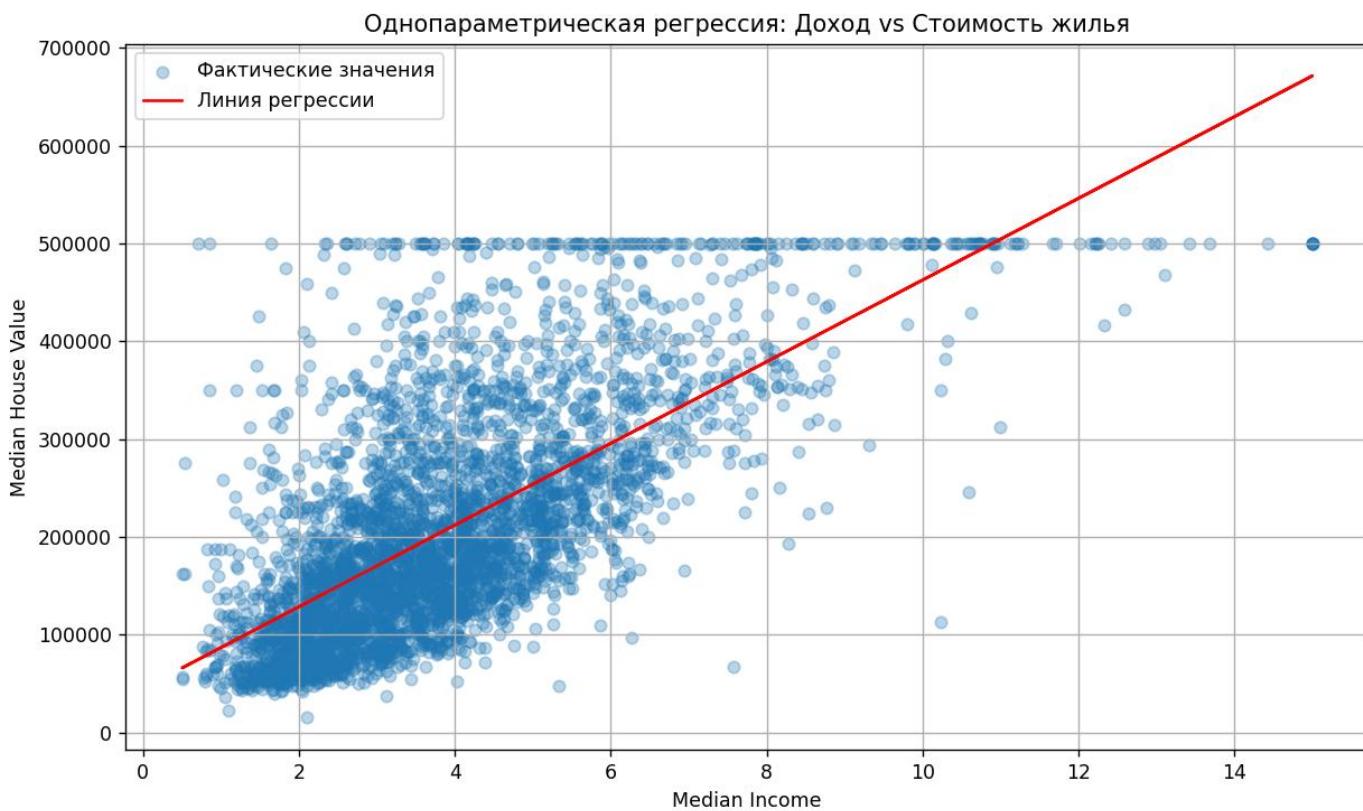
MSE: 7221011204.24

R²: 0.47

Многопараметрическая модель:

MSE: 4924108937.91

R²: 0.64



- 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).

Код программы:

```

import pandas as pd
from sklearn.model_selection import train_test_split

```

```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score,
recall_score, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv("titanic.csv")
df["Age"].fillna(df["Age"].median(), inplace=True)
df["Embarked"].fillna(df["Embarked"].mode()[0], inplace=True)
df["Sex"] = df["Sex"].map({"male": 0, "female": 1})
df["Embarked"] = df["Embarked"].map({"S": 0, "C": 1, "Q": 2})

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)

model = LogisticRegression(max_iter=200)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
print(f"Precision: {precision:.2f}")
print(f"Recall: {recall:.2f}")

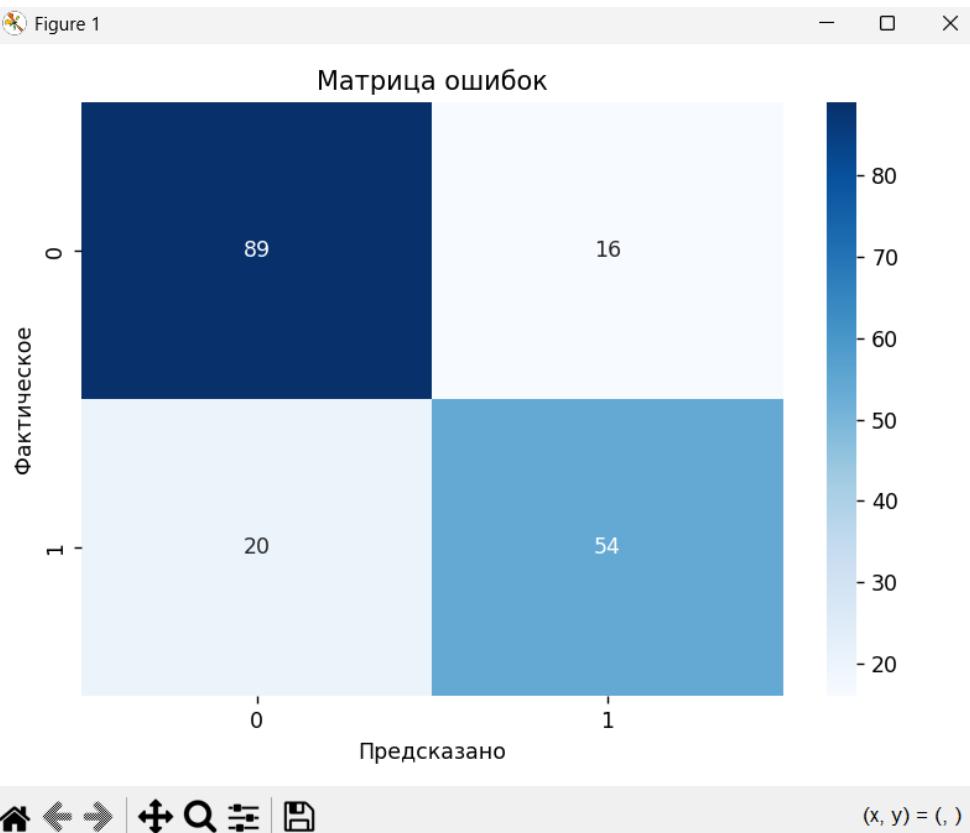
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues")
plt.title("Матрица ошибок")
plt.xlabel("Предсказано")
plt.ylabel("Фактическое")
plt.tight_layout()
plt.show()

```

Рисунки с результатами работы программы

Accuracy: 0.80
Precision: 0.77
Recall: 0.73

Figure 1



Вывод: научилась разрабатывать простые программы на Python с использованием библиотек **Pandas** для манипуляции и **Matplotlib** для визуализации, получил практический опыт работы с данными.

