Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ
Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №2

Выполнил
А.В. Горобец,
студент группы АС66
Проверил
А. А. Крощенко,
ст. преп. кафедры ИИТ,
« __ » _____2025 г.

Брест 2025

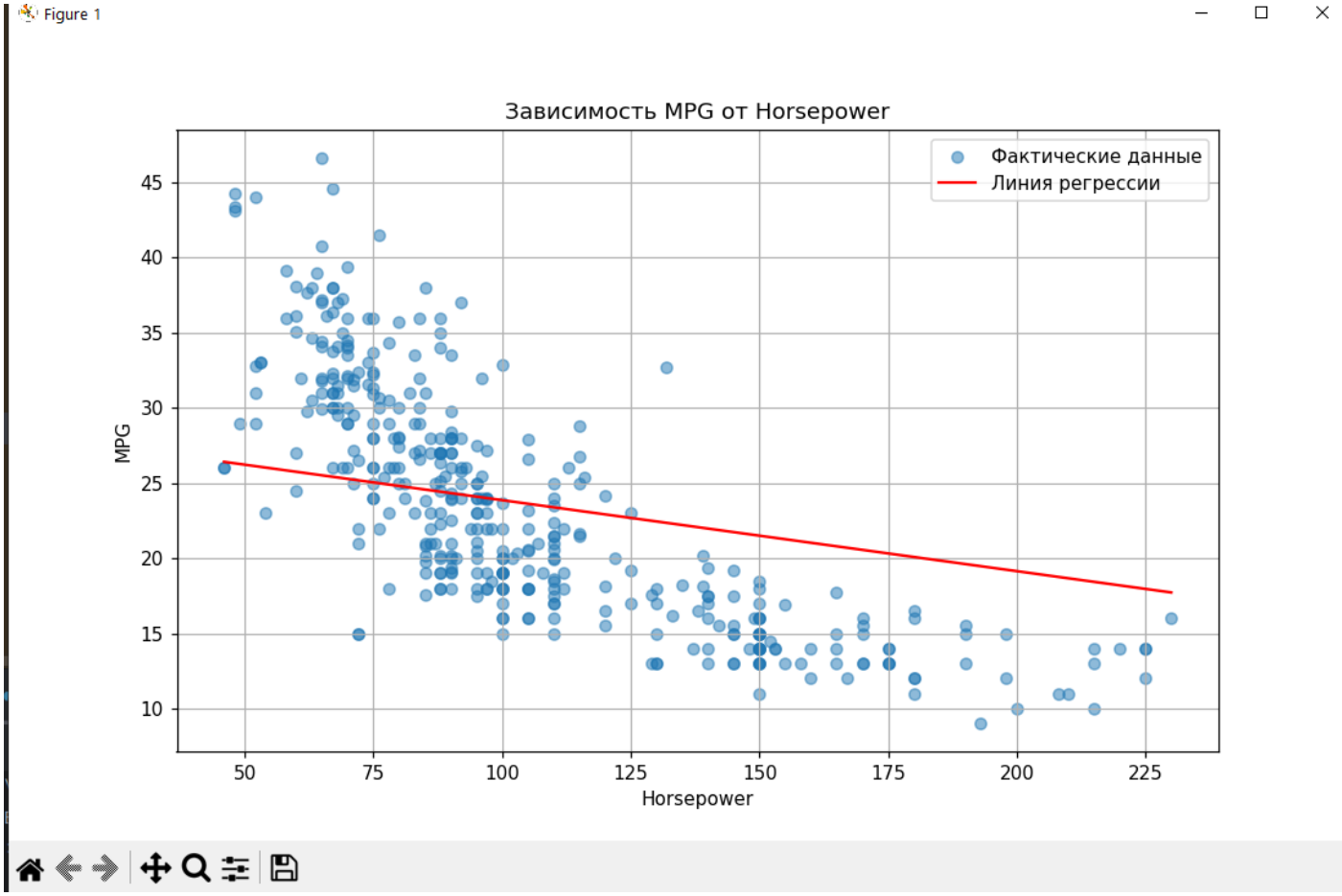
Цель работы: Изучить применение линейной и логистической регрессии для решения практических задач. Научиться обучать модели, оценивать их качество с помощью соответствующих метрик и интерпретировать результаты.

Вариант 3

Задание 1. Регрессия (Прогнозирование расхода топлива)

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

# 1. Загрузка данных
df = pd.read_csv('C:/Users/Anton/Downloads/auto-mpg.csv')  # Укажи путь к файлу, если он не в текущей папке

# 2. Обработка пропусков
df['horsepower'] = pd.to_numeric(df['horsepower'], errors='coerce')  # '?' → NaN
df.dropna(subset=['cylinders', 'horsepower', 'weight', 'mpg'], inplace=True)

# 3. Обработка категориальных признаков
df['origin'] = df['origin'].astype('category')
df = pd.get_dummies(df, columns=['origin'], drop_first=True)

# 4. Выбор признаков и целевой переменной
X = df[['cylinders', 'horsepower', 'weight']]
y = df['mpg']

# 5. Разделение на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 6. Обучение модели
model = LinearRegression()
model.fit(X_train, y_train)

# 7. Предсказание и метрики
y_pred = model.predict(X_test)
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}')

# 8. Визуализация зависимости mpg от horsepower
plt.figure(figsize=(10, 6))
plt.scatter(df['horsepower'], df['mpg'], alpha=0.5, label='Фактические данные')

# Линия регрессии при фиксированных средних значениях других признаков
hp_range = np.linspace(df['horsepower'].min(), df['horsepower'].max(), 100)
mean_cyl = df['cylinders'].mean()
mean_weight = df['weight'].mean()
X_line = pd.DataFrame({'cylinders': mean_cyl, 'horsepower': hp_range, 'weight': mean_weight})
y_line = model.predict(X_line)
```

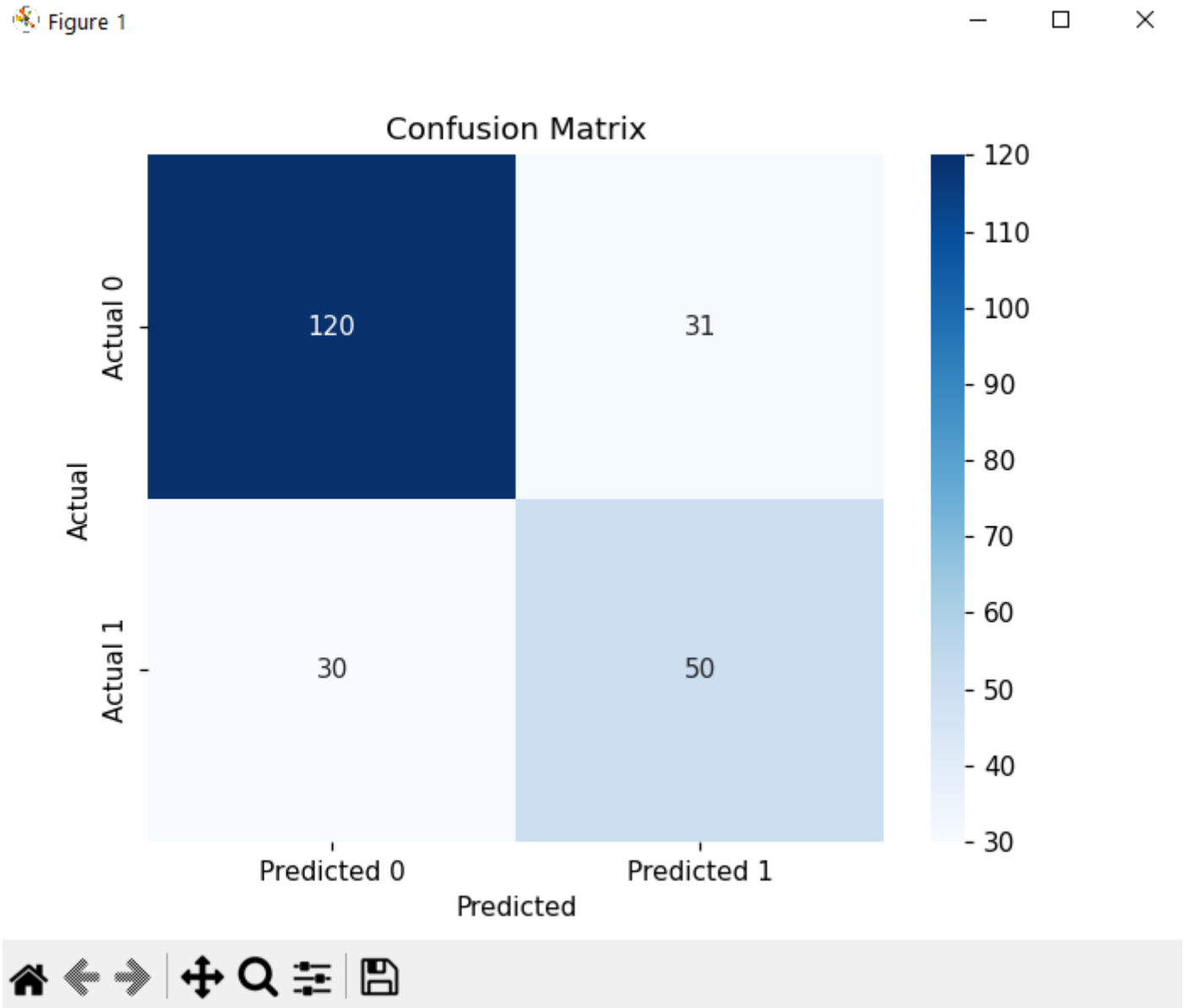
```
plt.plot(hp_range, y_line, color='red', label='Линия регрессии')
plt.xlabel('Horsepower')
plt.ylabel('MPG')
plt.title('Зависимость MPG от Horsepower')
plt.legend()
plt.grid(True)
plt.show()
```
MSE: 17.68
R^2: 0.65



Задание 2. Классификация (Диагностика диабета)

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
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

# 1. Загрузка данных
df = pd.read_csv("C:/Users/Anton/Downloads/pima-indians-diabetes.csv", comment="#",
header=None)
df.columns = [
    "Pregnancies", "Glucose", "BloodPressure", "SkinThickness", "Insulin",
    "BMI", "DiabetesPedigreeFunction", "Age", "Outcome"
```

```
]

# 2. Разделение признаков и целевой переменной
X = df.drop("Outcome", axis=1)
y = df["Outcome"]

# 3. Стандартизация признаков
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# 4. Разделение на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3, ran-
dom_state=42)

# 5. Обучение модели логистической регрессии
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)

# 6. Предсказания и метрики
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}")

# 7. Матрица ошибок
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=["Predicted 0", "Pre-
dicted 1"], yticklabels=["Actual 0", "Actual 1"])
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()

# 8. Выводы
fp = cm[0][1]
fn = cm[1][0]
print(f"Ложноположительных: {fp}")
print(f"Ложноотрицательных: {fn}")
Accuracy: 0.74
Precision: 0.62
Recall: 0.62
Ложноположительных: 31
Ложноотрицательных: 30
```



Вывод: я изучил применение линейной и логистической регрессии для решения практических задач. Научился обучать модели, оценивать их качество с помощью соответствующих метрик и интерпретировать результаты.