

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 курса  
Группы АС-65  
Дацкевич М. Г.  
Проверил:  
Крощенко А. А.

Брест 2025

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

## Вариант 4

Регрессия (Определение веса рыбы)

1. Fish Market

2. Предсказать вес рыбы (Weight)

3. Задания:

- \* загрузите данные. В качестве признаков используйте Length1, Length2, Length3, Height, Width;
- \* обучите модель линейной регрессии;
- \* оцените качество, рассчитав R<sup>2</sup> и RMSE (Root Mean Squared Error);
- \* постройте диаграмму рассеяния для Length3 и Weight с линией регрессии.
- Классификация (Прогнозирование отклика на банковское предложение)
  - 1. Bank Marketing UCI
  - 2. Предсказать, подпишется ли клиент на срочный вклад (у)
- 3. Задания:
  - \* загрузите данные, преобразуйте категориальные признаки;
  - \* обучите модель логистической регрессии;
  - \* рассчитайте Accuracy, Precision и Recall для класса "yes";
  - \* постройте матрицу ошибок.

Fish Market:

```
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

# Загрузка данных
df = pd.read_csv('auto-mpg.csv')
print(df.shape)

# обработка нулей
df['horsepower'] = pd.to_numeric(df['horsepower'], errors='coerce')
df = df.dropna(subset=['cylinders', 'horsepower', 'weight', 'mpg'])
print(df.shape)

# обучение
X = df[['cylinders', 'horsepower', 'weight']]
y = df['mpg']
```

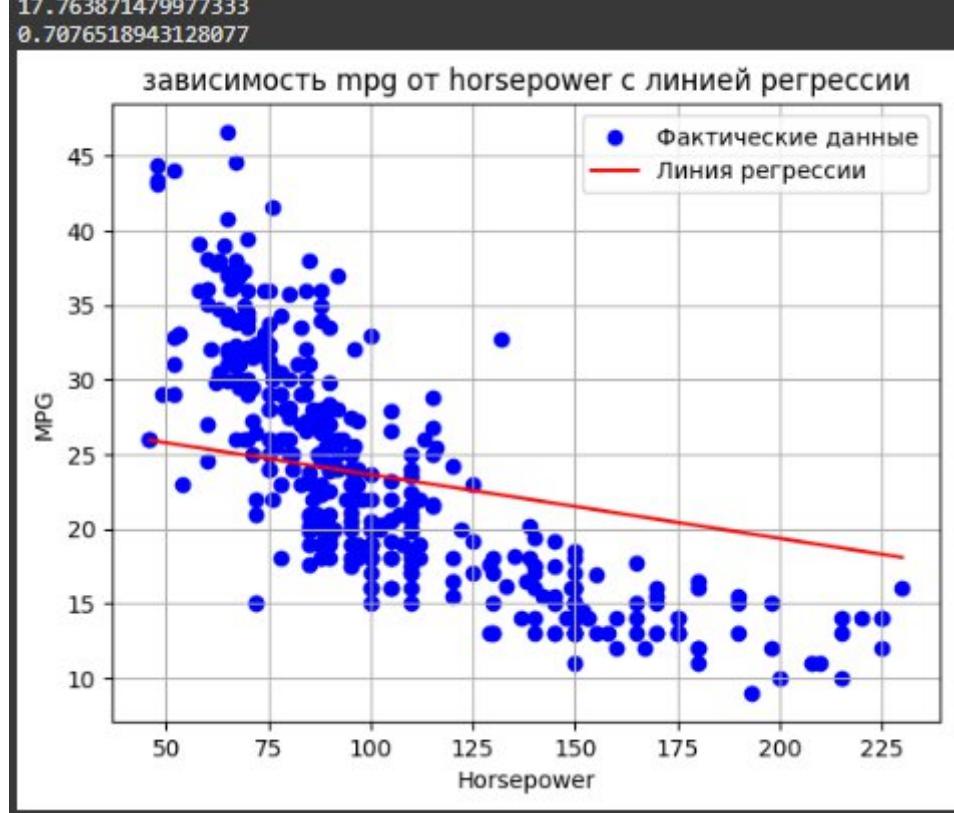
```

model=LinearRegression().fit(X,y)
y_m = model.predict(X)

#4. признаки
mse=mean_squared_error(y,y_m)
print(mse)
r2=r2_score(y,y_m)
print(r2)

#5. визуализация
plt.scatter(df['horsepower'], df['mpg'], color='blue', label='Фактические данные')
mean_cylinders = np.mean(df['cylinders'])
mean_weight = np.mean(df['weight'])
sorted_hp = np.sort(df['horsepower'])
#значения
hp_line = pd.DataFrame({
    'cylinders': mean_cylinders,
    'horsepower': sorted_hp,
    'weight': mean_weight
})
plt.plot(sorted_hp, model.predict(hp_line), color='red', label='Линия регрессии')
plt.title('зависимость mpg от horsepower с линией регрессии')
plt.xlabel('Horsepower')
plt.ylabel('MPG')
plt.legend()
plt.grid(True)
plt.show()

```



### Bank Marketing UCI:

```

import pandas as pd
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, ConfusionMatrixDisplay
import matplotlib.pyplot as plt

```

```

#1. Загрузка
df = pd.read_csv("pima-indians-diabetes.csv", skiprows=9, header=None)
df.columns = [
    'Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness',
    'Insulin', 'BMI', 'DiabetesPedigree', 'Age', 'Outcome'
]
print(df.head())
print("Размер данных:", df.shape)

#2. разделение признаков и целевой
X = df.drop('Outcome', axis=1)
y = df['Outcome']

#3. Стандартизация признаков
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_scaled = pd.DataFrame(X_scaled, columns=X.columns)
print(X_scaled)

#4. разделение на обучающую и тестовую выборку и обучение
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3, random_state=42)
model = LogisticRegression()
model.fit(X_train, y_train)

#5. Предсказание и расчет
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:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")

#6. матрица
matrix = confusion_matrix(y_test, y_pred)
print(matrix)
print(f"Ложноположительные (FP): {matrix[0][1]}")
print(f"Ложноотрицательные (False Negatives): {matrix[1][0]}")
Accuracy: 0.7359
Precision: 0.6173
Recall: 0.6250
[[120  31]
 [ 30  50]]
Ложноположительные (FP): 31
Ложноотрицательные (False Negatives): 30

```

Вывод: изучил применение линейной и логистической регрессии для решения практических задач и научился обучать модели, оценил их качество с помощью соответствующих метрик и интерпретировал результаты.