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

Ракецкий П. П.

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 R² 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 (y)

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')

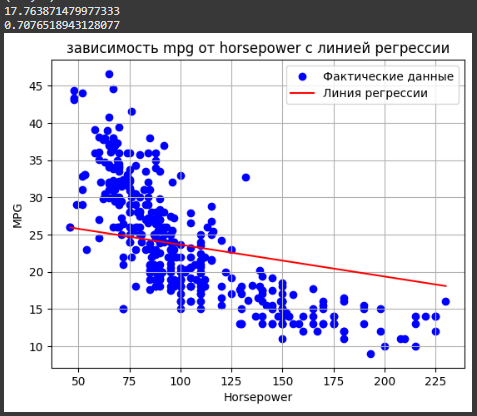
plt.xlabel('Horsepower')

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, confsion\_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 рассчет

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}")

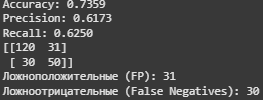
#6.матрица

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

print(matrix)

print(f"Ложноположительные (FP): {matrix[0][1]}")

print(f"Ложноотрицательные (False Negatives): {matrix[1][0]}")



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.