

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 R^2 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 r2_score, mean_squared_error

# 1. Загрузка данных
df = pd.read_csv('Fish.csv')

# 2. Подготовка признаков и целевой переменной
X = df[['Length1', 'Length2', 'Length3', 'Height', 'Width']]
y = df['Weight']

# 3. Обучение модели линейной регрессии
model = LinearRegression()
model.fit(X, y)

# 4. Предсказание и оценка качества
```

```

y_pred = model.predict(X)

# Расчет метрик
r2 = r2_score(y, y_pred)
rmse = np.sqrt(mean_squared_error(y, y_pred))

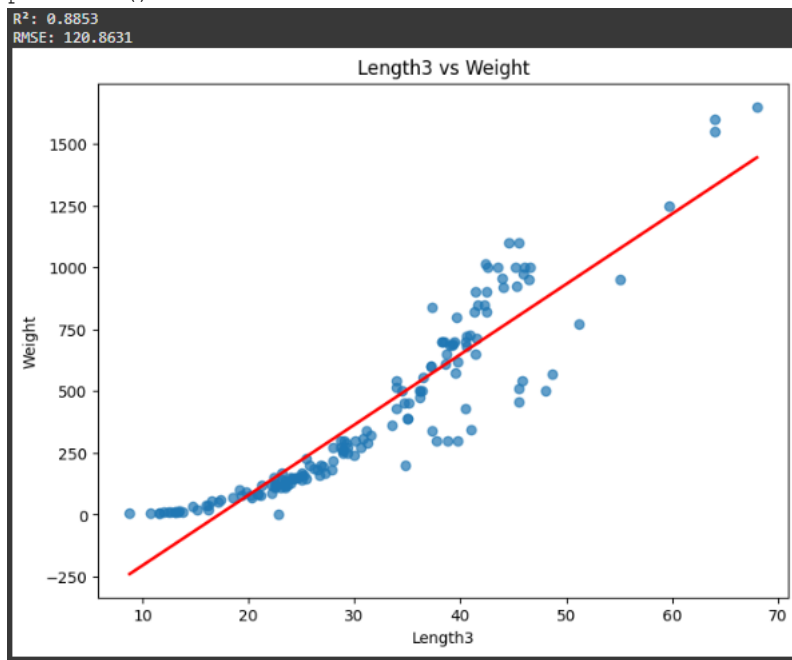
print(f"R²: {r2:.4f}")
print(f"RMSE: {rmse:.4f}")

# 5. Диаграмма рассеяния для Length3 и Weight с линией регрессии
plt.figure(figsize=(8, 6))
plt.scatter(df['Length3'], df['Weight'], alpha=0.7)
plt.xlabel('Length3')
plt.ylabel('Weight')
plt.title('Length3 vs Weight')

# Линия регрессии для Length3
x_line = pd.DataFrame({'Length3': np.linspace(df['Length3'].min(),
df['Length3'].max(), 100)})
model_length3 = LinearRegression()
model_length3.fit(df[['Length3']], y)
y_line = model_length3.predict(x_line)

plt.plot(x_line['Length3'], y_line, color='red', linewidth=2)
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.