

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:
Студентка 3 курса
Группы АС-65
Зинчук М.С.
Проверил:
Крощенко А. А.

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.

Ход работы

Общее задание: выполнить задания по варианту (регрессия и классификация), построить все требуемые визуализации и рассчитать метрики, написать отчет, создать пул-реквест в репозиторий с кодом решения и отчетом в формате pdf.

Вариант 6

- **Регрессия (Прогнозирование цен на жильё в Бостоне)**

1. Boston Housing

2. Предсказать медианную стоимость дома (MEDV)

3. **Задания:**

- загрузите данные. Используйте все признаки для обучения;
- обучите модель **линейной регрессии**;
- рассчитайте **MSE** и **R²**;
- визуализируйте зависимость MEDV от RM (среднее количество комнат) с линией регрессии.

- **Классификация (Диагностика рака груди)**

1. Breast Cancer Wisconsin

2. Предсказать, является ли опухоль злокачественной (malignant) или доброкачественной (benign)

3. **Задания:**

- загрузите данные и выполните стандартизацию признаков;
- обучите модель **логистической регрессии**;
- рассчитайте **Accuracy, Precision** и **Recall**;
- постройте **матрицу ошибок** и объясните значение каждого из её квадрантов в контексте задачи.

Код программы (1):

```
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

df = pd.read_csv('BostonHousing.csv')

X = df.drop(['MEDV', 'CAT. MEDV'], axis=1)
y = df['MEDV']

model = LinearRegression()
model.fit(X, y)
```

```

y_pred = model.predict(X)
mse = mean_squared_error(y, y_pred)
r2 = r2_score(y, y_pred)

print(f'MSE: {mse:.2f}')
print(f'R2: {r2:.2f}')

plt.figure(figsize=(10, 6))
plt.scatter(df['RM'], df['MEDV'], alpha=0.6, color='blue', s=30)

rm_model = LinearRegression()
rm_model.fit(df[['RM']], y)
rm_pred = rm_model.predict(df[['RM']])

plt.plot(df['RM'], rm_pred, color='red', linewidth=2, label='Линия регрессии')
plt.xlabel('Среднее количество комнат (RM)', fontsize=12)
plt.ylabel('Медианная стоимость дома (MEDV)', fontsize=12)
plt.title('Зависимость стоимости дома от количества комнат', fontsize=14)
plt.grid(True, alpha=0.3)
plt.legend()
plt.tight_layout()
plt.show()

```

График:



Код программы (2):

```

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

data = pd.read_csv('breast_cancer.csv')

X = data.drop(['id', 'diagnosis'], axis=1)
y = data['diagnosis'].map({'M': 1, 'B': 0})

X = X.fillna(0)

```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
scaler = StandardScaler()
```

```
X_train = scaler.fit_transform(X_train)
```

```
X_test = scaler.transform(X_test)
```

```
model = LogisticRegression()
```

```
model.fit(X_train, y_train)
```

```
predictions = model.predict(X_test)
```

```
print("Accuracy:", accuracy_score(y_test, predictions))
```

```
print("Precision:", precision_score(y_test, predictions))
```

```
print("Recall:", recall_score(y_test, predictions))
```

```
print("\nConfusion Matrix:")
```

```
print(confusion_matrix(y_test, predictions))
```

```
Accuracy: 0.9736842105263158
```

```
Precision: 0.9761904761904762
```

```
Recall: 0.9534883720930233
```

```
Confusion Matrix:
```

```
[[70  1]
```

```
 [ 2 41]]
```

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