

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
Вавдийчик Н.Д.
Проверил:
Крощенко А. А.

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.

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 1

- **Регрессия (Прогнозирование стоимости жилья в Калифорнии)**
 1. California Housing
 2. Предсказать медианную стоимость дома (median_house_value)
 3. **Задания:**
 - загрузите данные и разделите их на обучающую и тестовую выборки;
 - обучите модель **линейной регрессии** на обучающих данных;
 - сделайте предсказания для тестовой выборки;
 - оцените качество модели, рассчитав метрики **MSE** (Mean Squared Error) и **R²** (Coefficient of Determination);
 - визуализируйте результат: постройте диаграмму рассеяния для признака median_income (медианный доход) и целевой переменной, нанеся на неё линию регрессии.
- **Классификация (Прогнозирование выживаемости на "Титанике")**
 1. Titanic
 2. Предсказать, выжил ли пассажир (Survived)
 3. **Задания:**
 - загрузите и предварительно обработайте данные (заполните пропуски, преобразуйте категории в числа);
 - обучите модель **логистической регрессии**;
 - оцените качество модели, рассчитав **Accuracy, Precision и Recall**;
- постройте и проанализируйте **матрицу ошибок** (confusion matrix).

Код программы (1):

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

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

X = df[['median_income']]
y = df['median_house_value']

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

model = LinearRegression()
model.fit(X_train, y_train)

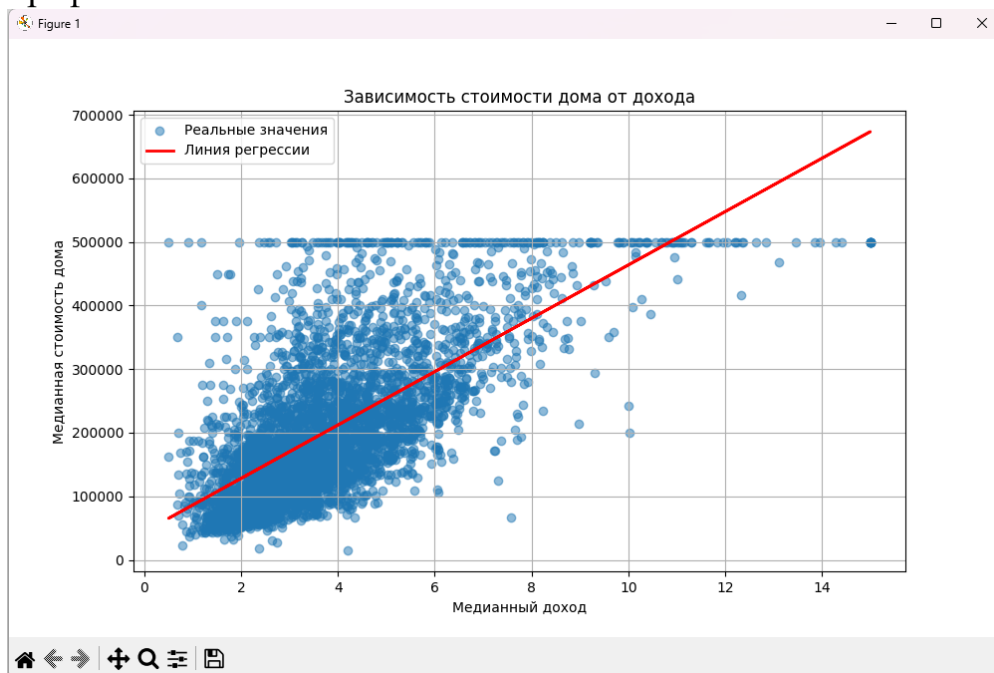
y_pred = model.predict(X_test)

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}')

plt.figure(figsize=(10, 6))
plt.scatter(X_test, y_test, alpha=0.5, label='Реальные значения')
plt.plot(X_test, y_pred, color='red', linewidth=2, label='Линия регрессии')
plt.xlabel('Медианный доход')
plt.ylabel('Медианная стоимость дома')
plt.title('Зависимость стоимости дома от дохода')
plt.legend()
plt.grid(True)
plt.show()
```

График:



Код программы (2):

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, confusion_matrix
import warnings
warnings.filterwarnings('ignore')

df = pd.read_csv('Titanic-Dataset (1).csv')
df['Age'].fillna(df['Age'].mean(), inplace=True)

df['Sex'] = df['Sex'].map({'male': 0, 'female': 1})

df['Embarked'].fillna(df['Embarked'].mode()[0], inplace=True)

df['Embarked'] = df['Embarked'].map({'S': 0, 'C': 1, 'Q': 2})

features = ['Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare', 'Embarked']
X = df[features]
y = df['Survived']

X['Fare'].fillna(X['Fare'].mean(), inplace=True)

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

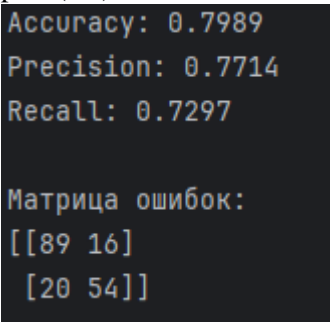
model = LogisticRegression()
model.fit(X_train, y_train)

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}")

cm = confusion_matrix(y_test, y_pred)
print("\nМатрица ошибок:")
print(cm)
```



```
Accuracy: 0.7989
Precision: 0.7714
Recall: 0.7297

Матрица ошибок:
[[89 16]
 [20 54]]
```

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