

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №2

Специальность АС-66

Выполнила
Е. С. Неруш,
студент группы АС-66

Проверил
А. А. Крощенко,
ст. преп. кафедры ИИТ,
«___» _____ 2025 г.

Брест 2025

Цель работы: Получить практические навыки работы с данными с использованием библиотек **Pandas** для манипуляции и **Matplotlib** для визуализации. Научиться выполнять основные шаги предварительной обработки данных, такие как очистка, нормализация и работа с различными типами признаков.

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 10

- Регрессия (Прогнозирование счастья в странах)

1. World Happiness Report

2. Предсказать оценку счастья (Score)

3. Задания:

- § загрузите данные. В качестве признаков используйте GDP per capita, Social support, Healthy life expectancy;

- § обучите модель линейной регрессии;

- § рассчитайте MSE и R2;

- § визуализируйте зависимость Score от GDP per capita с линией регрессии.

- Классификация (Прогнозирование оттока клиентов)

1. Telco Customer Churn

2. Предсказать, уйдёт ли клиент (Churn = 'Yes')

3. Задания:

- § загрузите данные, обработайте категориальные признаки;

- § обучите модель логистической регрессии;

- § рассчитайте Accuracy, Precision и Recall для класса 'Yes';

- § постройте матрицу ошибок. Код программы:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.linear_model import LinearRegression, LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score, accuracy_score,
precision_score, recall_score, confusion_matrix

print("Регрессия: World Happiness Report")

# Загрузка данных
happiness_df =
pd.read_csv("E:/Projects/ml_as66/reports/Nerush/lab2/src/world_happiness_report.csv")
print(f"Загружено строк: {len(happiness_df)}")
```

```

# Выбор признаков и целевой переменной
reg_features = ['GDP per capita', 'Social support', 'Healthy life expectancy']
reg_target = 'Score'

# Удаление строк с пропущенными значениями
happiness_df = happiness_df.dropna(subset=reg_features + [reg_target])
print(f"После удаления пропусков осталось строк: {len(happiness_df)}")

# Разделение на обучающую и тестовую выборки
X_reg = happiness_df[reg_features]
y_reg = happiness_df[reg_target]
X_reg_train, X_reg_test, y_reg_train, y_reg_test = train_test_split(X_reg,
y_reg, test_size=0.2, random_state=42)

# Обучение модели линейной регрессии
reg_model = LinearRegression()
reg_model.fit(X_reg_train, y_reg_train)
print("Модель линейной регрессии обучена.")

# Предсказания и метрики
y_reg_pred = reg_model.predict(X_reg_test)
mse = mean_squared_error(y_reg_test, y_reg_pred)
r2 = r2_score(y_reg_test, y_reg_pred)

print(f"MSE (среднеквадратичная ошибка): {mse:.3f}")
print(f"R2 (коэффициент детерминации): {r2:.3f}")
print(f"Модель объясняет {r2*100:.1f}% дисперсии оценки счастья.")

```

```

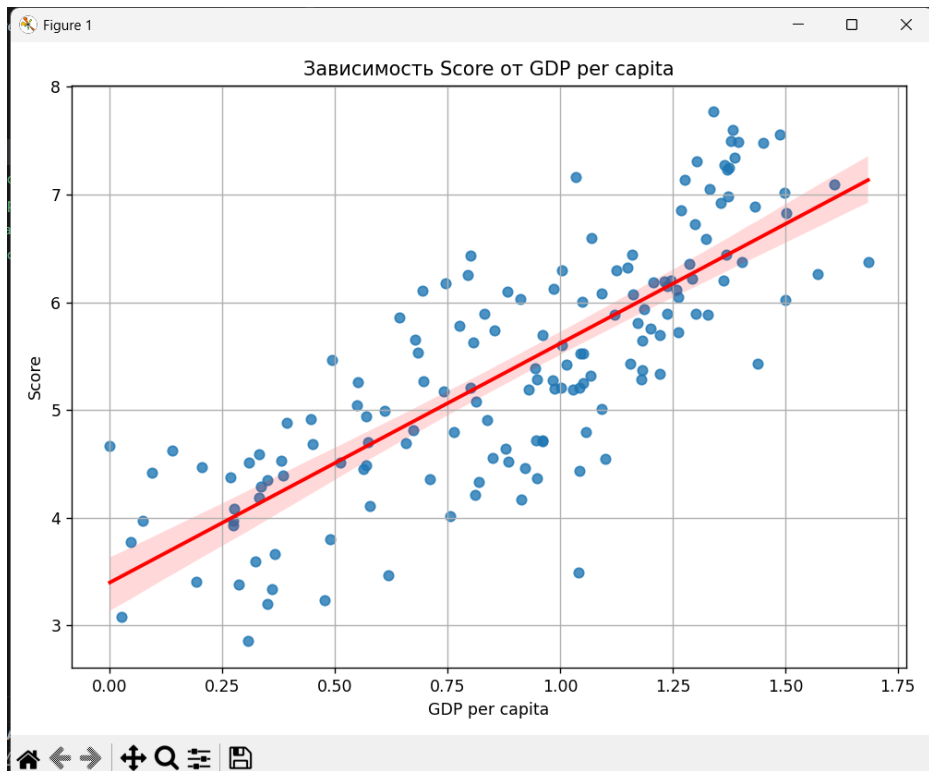
Регрессия: World Happiness Report
Загружено строк: 156
После удаления пропусков осталось строк: 156
Модель линейной регрессии обучена.
MSE (среднеквадратичная ошибка): 0.464
R2 (коэффициент детерминации): 0.554
Модель объясняет 55.4% дисперсии оценки счастья.

```

```

# Визуализация зависимости Score от GDP per capita
plt.figure(figsize=(8, 6))
sns.regplot(x='GDP per capita', y='Score', data=happiness_df,
line_kws={'color': 'red'})
plt.title('Зависимость Score от GDP per capita')
plt.xlabel('GDP per capita')
plt.ylabel('Score')
plt.grid(True)
plt.tight_layout()
plt.show()

```



```
print("\nКлассификация: Telco Customer Churn")

# Загрузка данных
churn_df = pd.read_csv('E:/Projects/ml_as66/reports/Nerush/lab2/src/Telco-
Customer-Churn.csv')
print(f"Загружено строк: {len(churn_df)}")

# Удаление строк с пропущенными значениями
churn_df = churn_df.dropna()
print(f"После удаления пропусков осталось строк: {len(churn_df)}")

# Преобразование целевой переменной
churn_df['Churn'] = churn_df['Churn'].map({'Yes': 1, 'No': 0})
print("Целевая переменная преобразована в бинарный формат.")

# Обработка категориальных признаков
X_clf = churn_df.drop(columns=['customerID', 'Churn'])
X_clf = pd.get_dummies(X_clf, drop_first=True)
y_clf = churn_df['Churn']
print(f"После кодирования категориальных признаков: {X_clf.shape[1]}
признаков.")

# Разделение на обучающую и тестовую выборки
X_clf_train, X_clf_test, y_clf_train, y_clf_test = train_test_split(X_clf,
y_clf, test_size=0.2, random_state=42)

# Обучение модели логистической регрессии
clf_model = LogisticRegression(max_iter=1000)
clf_model.fit(X_clf_train, y_clf_train)
print("Модель логистической регрессии обучена.")

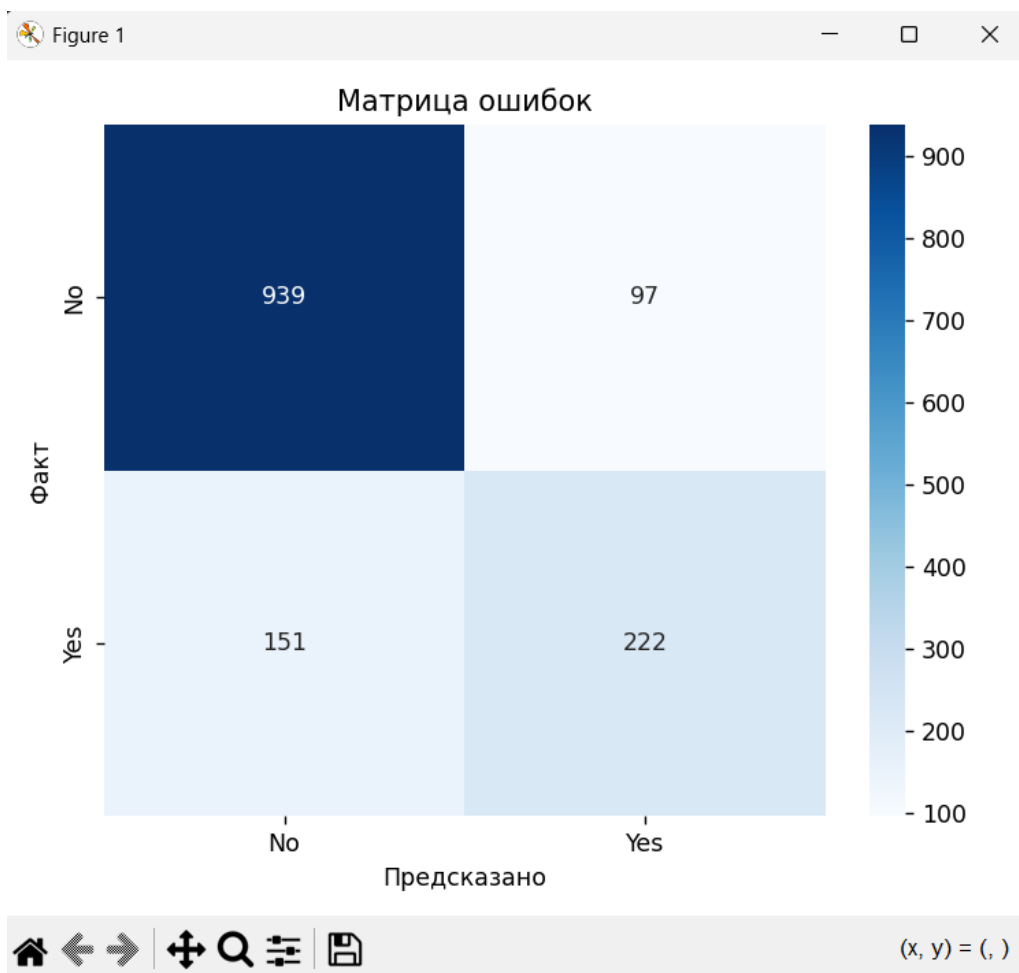
# Предсказания и метрики
y_clf_pred = clf_model.predict(X_clf_test)
accuracy = accuracy_score(y_clf_test, y_clf_pred)
```

```
precision = precision_score(y_clf_test, y_clf_pred)
recall = recall_score(y_clf_test, y_clf_pred)

print(f"Accuracy (доля правильных предсказаний): {accuracy:.3f}")
print(f"Precision (точность для класса 'Yes'): {precision:.3f}")
print(f"Recall (полнота для класса 'Yes'): {recall:.3f}")
```

```
Классификация: Telco Customer Churn
Загружено строк: 7043
После удаления пропусков осталось строк: 7043
Целевая переменная преобразована в бинарный формат.
После кодирования категориальных признаков: 6559 признаков.
Модель логистической регрессии обучена.
Accuracy (доля правильных предсказаний): 0.824
Precision (точность для класса 'Yes'): 0.696
Recall (полнота для класса 'Yes'): 0.595
□
```

```
# Матрица ошибок
conf_matrix = confusion_matrix(y_clf_test, y_clf_pred)
plt.figure(figsize=(6, 5))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['No',
'Yes'], yticklabels=['No', 'Yes'])
plt.title('Матрица ошибок')
plt.xlabel('Предсказано')
plt.ylabel('Факт')
plt.tight_layout()
plt.show()
```



Вывод: Научился применять линейную и логистическую регрессию для решения задач регрессии и классификации. Получил практический опыт загрузки и предобработки данных, построения моделей с использованием библиотек Pandas, Scikit-learn и визуализации результатов с помощью Matplotlib и Seaborn. Освоил расчёт метрик качества моделей и интерпретацию полученных результатов.