

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»

Выполнила:
Студентка 3 курса
Группы АС-65
Шлейхер А. С.
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 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.

Код программы:

```
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 выбор признаков
df = pd.read_csv("world_happiness_report.csv")

X = df[["GDP per capita", "Social support", "Healthy life expectancy"]]
y = df["Score"]

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

# MSE R2
y_pred = model.predict(X)

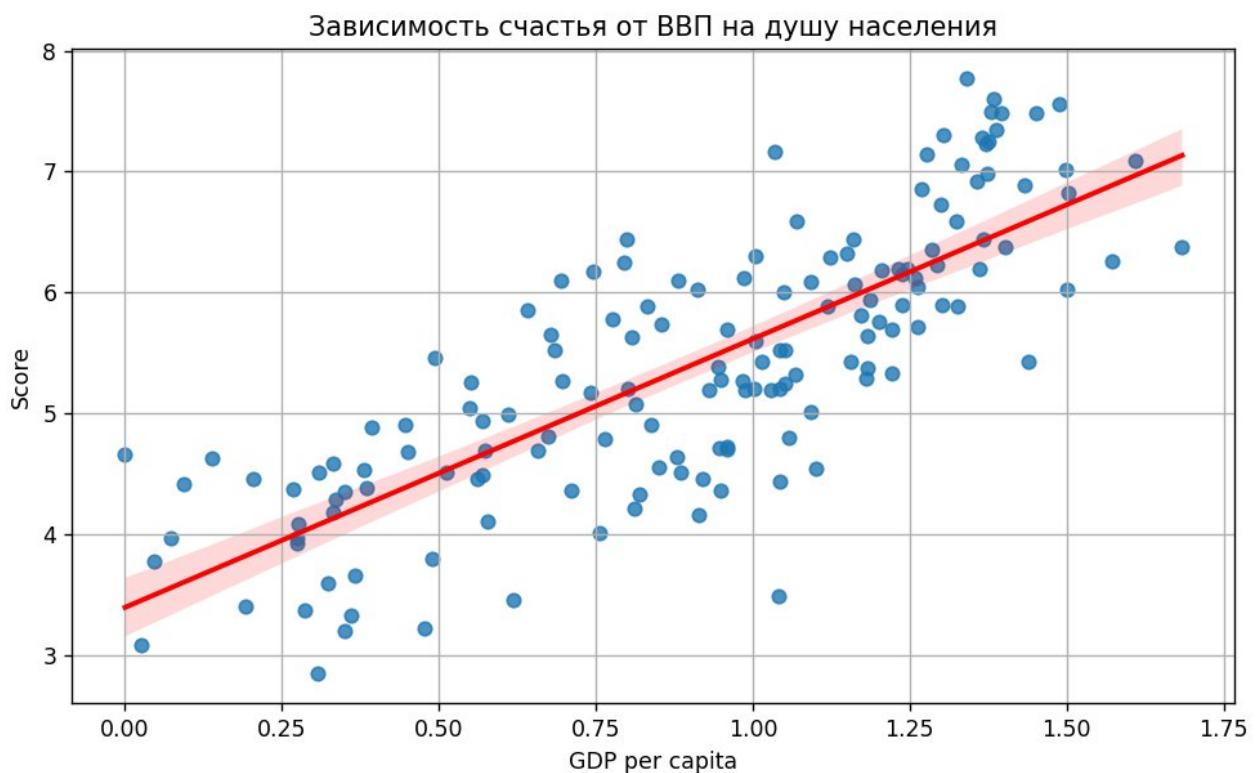
mse = mean_squared_error(y, y_pred)
r2 = r2_score(y, y_pred)

print(f"MSE: {mse:.4f}")
print(f"R2: {r2:.4f}")

# Зависимость Score от GDP per capita
plt.figure(figsize=(8, 5))
sns.regplot(x=df["GDP per capita"], y=df["Score"], line_kws={"color": "red"})
plt.title("Зависимость счастья от ВВП на душу населения")
plt.xlabel("GDP per capita")
plt.ylabel("Score")
plt.grid(True)
plt.tight_layout()
plt.show()
```

MSE: 0.3369

R²: 0.7263



- **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
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score,
confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
```

```
# Загрузка данных Обработка категориальных признаков
df = pd.read_csv("Telco-Customer-Churn.csv")
```

```
df = df.drop("customerID", axis=1)
```

```

cat_cols = df.select_dtypes(include="object").columns
for col in cat_cols:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])

# Обучение логистической регрессии
X = df.drop("Churn", axis=1)
y = df["Churn"]

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

model = LogisticRegression(max_iter=10000)
model.fit(X_train, y_train)

# Accuracy Precision Recall
y_pred = model.predict(X_test)

acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)

print(f"Accuracy: {acc:.3f}")
print(f"Precision (Yes): {prec:.3f}")
print(f"Recall (Yes): {rec:.3f}")

# Матрица ошибок
cm = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(5, 4))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", cbar=False,
            xticklabels=["No", "Yes"], yticklabels=["No", "Yes"])
plt.title("Матрица ошибок (Churn)")
plt.xlabel("Предсказано")
plt.ylabel("Фактическое значение")
plt.tight_layout()
plt.show()

```

```

Accuracy: 0.796
Precision (Yes): 0.642
Recall (Yes): 0.527

```

Матрица ошибок (Churn)



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