

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Лабораторная работа №2
По дисциплине: «ОМО»
Тема: «Линейные модели для задач регрессии и классификации»

Выполнил:
Студент 3-го курса
Группы АС-65
Нестюк Н.С.
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.

Вариант 2

• Регрессия (Прогнозирование медицинских расходов)

1. Medical Cost Personal Datasets

2. Предсказать страховые выплаты (charges)

3. Задания:

- загрузите и обработайте категориальные признаки (например, sex, smoker);

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

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, r2_score

url = "https://raw.githubusercontent.com/stedy/Machine-Learning-with-R-datasets/master/insurance.csv"
data = pd.read_csv(url)

print("Размер данных:", data.shape)
print(data.head())

data_encoded = pd.get_dummies(data, drop_first=True)

X = data_encoded.drop("charges", axis=1)
y = data_encoded["charges"]

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

Размер данных: (1338, 7)								
	age	sex	bmi	children	smoker	region	charges	
0	19	female	27.900		0	yes	southwest	16884.92400
1	18	male	33.770		1	no	southeast	1725.55230
2	28	male	33.000		3	no	southeast	4449.46200
3	33	male	22.705		0	no	northwest	21984.47061
4	32	male	28.880		0	no	northwest	3866.85520

- обучите модель линейной регрессии для предсказания charges;

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

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

- рассчитайте MAE (Mean Absolute Error) и R²;

```
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
```

```
print(f"MAE (Средняя абсолютная ошибка): {mae:.2f}")
print(f"R2 (Коэффициент детерминации): {r2:.4f}")
```

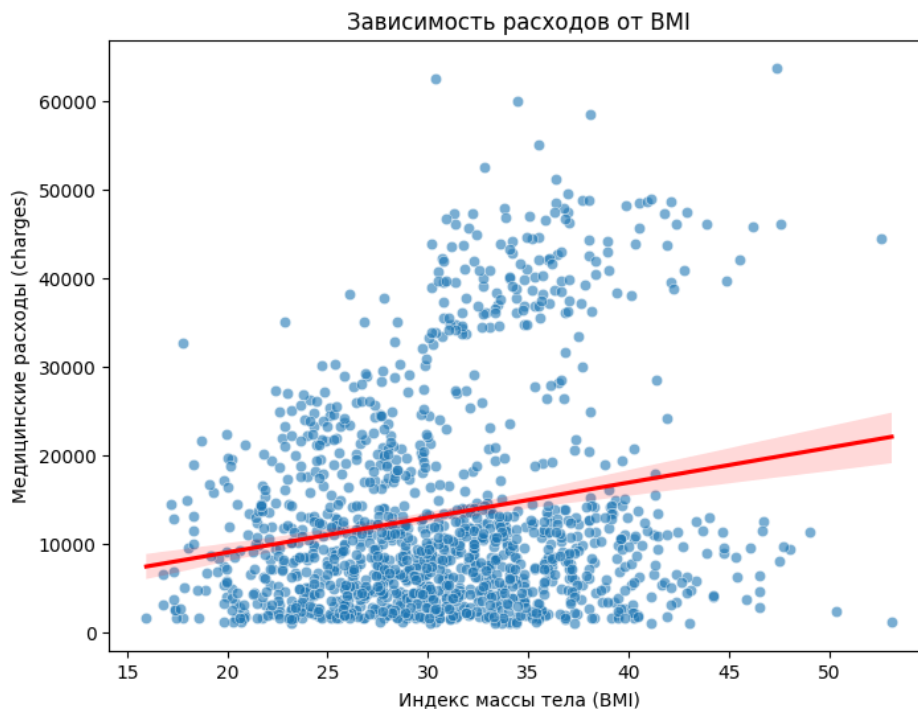
MAE (Средняя абсолютная ошибка): 4181.19
R2 (Коэффициент детерминации): 0.7836

- визуализируйте зависимость charges от bmi (индекс массы тела) с помощью диаграммы рассеяния и линии регрессии.

```
plt.figure(figsize=(8,6))
sns.scatterplot(x=data["bmi"], y=data["charges"], alpha=0.6)

sns.regplot(x=data["bmi"], y=data["charges"], scatter=False, color="red")

plt.xlabel("Индекс массы тела (BMI)")
plt.ylabel("Медицинские расходы (charges)")
plt.title("Зависимость расходов от BMI")
plt.show()
```



• Классификация (Диагностика заболеваний сердца)

1. Heart Disease UCI
2. Предсказать наличие у пациента болезни сердца (target)
3. Задания:

- загрузите данные и разделите их на обучающую и тестовую выборки;

```
url = "https://raw.githubusercontent.com/sharmaroshan/Heart-UCI-Dataset/master/heart.csv"
data = pd.read_csv(url)

print("Размер данных:", data.shape)
print(data.head())

X = data.drop("target", axis=1)
y = data["target"]

X_train, X_test, y_train, y_test = train_test_split(
```

```
X, y, test_size=0.2, random_state=42, stratify=y
)
```

Размер данных: (303, 14)

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	\
0	63	1	3	145	233	1	0	150	0	2.3	0	
1	37	1	2	130	250	0	1	187	0	3.5	0	
2	41	0	1	130	204	0	0	172	0	1.4	2	
3	56	1	1	120	236	0	1	178	0	0.8	2	
4	57	0	0	120	354	0	1	163	1	0.6	2	

	ca	thal	target
0	0	1	1
1	0	2	1
2	0	2	1
3	0	2	1
4	0	2	1

▪ обучите модель логистической регрессии;

```
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)
```

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

▪ оцените модель помощью Accuracy, Precision, Recall и F1-score;

```
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
```

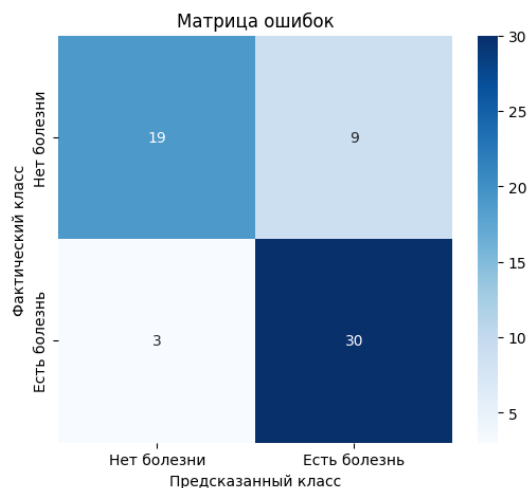
```
print(f"Accuracy: {acc:.4f}")
print(f"Precision: {prec:.4f}")
print(f"Recall: {rec:.4f}")
print(f"F1-score: {f1:.4f}")
```

```
Accuracy: 0.8033
Precision: 0.7692
Recall: 0.9091
F1-score: 0.8333
```

▪ постройте матрицу ошибок.

```
cm = confusion_matrix(y_test, y_pred)
```

```
plt.figure(figsize=(6,5))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
            xticklabels=["Нет болезни", "Есть болезнь"],
            yticklabels=["Нет болезни", "Есть болезнь"])
plt.xlabel("Предсказанный класс")
plt.ylabel("Фактический класс")
plt.title("Матрица ошибок")
plt.show()
```



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