

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:  
Студент 3 курса  
Группы АС-65  
Нестюк Н. 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.

## Вариант 2

### Ход работы:

Регрессия (Прогнозирование медицинских расходов)

1. Medical Cost Personal Datasets

2. Предсказать страховые выплаты (charges)

3. Задания:

- загрузите и обработайте категориальные признаки (например, sex, smoker);
- обучите модель линейной регрессии для предсказания charges;
- рассчитайте MAE (Mean Absolute Error) и R2;
- визуализируйте зависимость charges от bmi (индекс массы тела) с помощью диаграммы рассеяния и линии регрессии.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, LogisticRegression
from sklearn.metrics import mean_absolute_error, r2_score, accuracy_score,
precision_score, recall_score, f1_score, confusion_matrix
from sklearn.preprocessing import LabelEncoder
import matplotlib.pyplot as plt
from google.colab import files
uploaded = files.upload()

df_medical = pd.read_csv('medical_cost_personal_dataset.csv')
df_medical['sex'] = df_medical['sex'].map({'male': 0, 'female': 1})
df_medical['smoker'] = df_medical['smoker'].map({'no': 0, 'yes': 1})
df_medical['region'] = LabelEncoder().fit_transform(df_medical['region'])

X = df_medical.drop('charges', axis=1)
y = df_medical['charges']
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)

mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"MAE: {mae:.2f}")
print(f"R2: {r2:.4f}")

plt.figure(figsize=(12, 4))

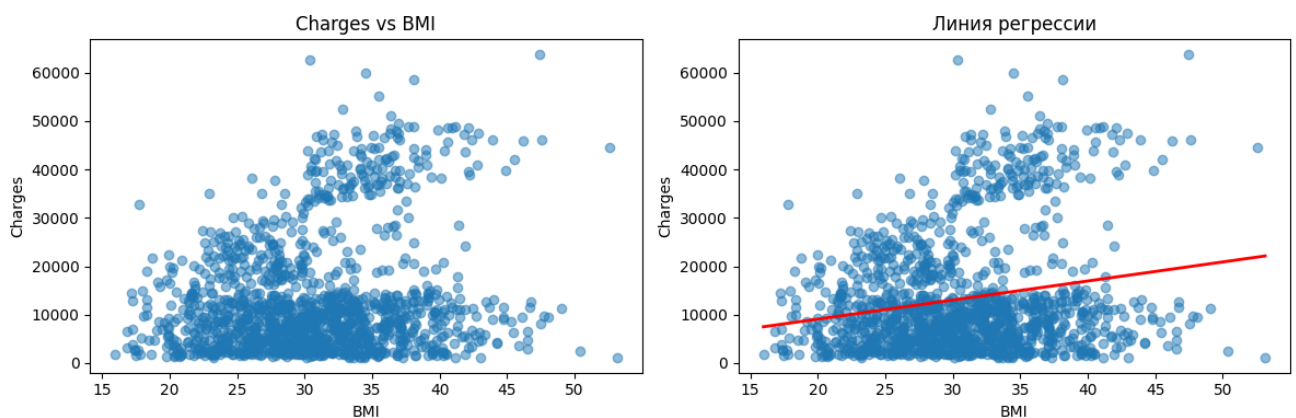
plt.subplot(1, 2, 1)
plt.scatter(df_medical['bmi'], df_medical['charges'], alpha=0.5)
plt.xlabel('BMI')
plt.ylabel('Charges')
plt.title('Charges vs BMI')
```

```
plt.subplot(1, 2, 2)
bmi_model = LinearRegression()
bmi_model.fit(df_medical[['bmi']], df_medical['charges'])
bmi_range = [[df_medical['bmi'].min()], [df_medical['bmi'].max()]]
bmi_pred = bmi_model.predict(bmi_range)

plt.scatter(df_medical['bmi'], df_medical['charges'], alpha=0.5)
plt.plot([df_medical['bmi'].min(), df_medical['bmi'].max()], bmi_pred, color='red',
linewidth=2)
plt.xlabel('BMI')
plt.ylabel('Charges')
plt.title('Линия регрессии')

plt.tight_layout()
plt.show()
```

**MAE: 4186.51**  
**R2: 0.7833**



## 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;

- обучите модель логистической регрессии;
- оцените модель с

помощью Accuracy, Precision, Recall и F1-score;

- постройте матрицу ошибок.

```
from google.colab import files
uploaded = files.upload()

df_medical = pd.read_csv('heart_disease_uci.csv')
df_medical['target'] = (df_medical['num'] > 0).astype(int)

features = ['age', 'trestbps', 'chol', 'thalch', 'oldpeak']
df_clean = df_medical[features + ['target']].dropna()

X = df_clean[features]
y = df_clean['target']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)
```

```

model = LogisticRegression(max_iter=1000)
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)
f1 = f1_score(y_test, y_pred)

print(f"Accuracy: {accuracy:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
print(f"F1-score: {f1:.4f}")

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

print("\n [ [TN, FP] \n [FN, TP] ]")

plt.figure(figsize=(5, 4))
plt.imshow(cm, cmap='Blues')
for i in range(2):
    for j in range(2):
        plt.text(j, i, str(cm[i, j]), ha='center', va='center', fontsize=20)
plt.xticks([0, 1], )
plt.yticks([0, 1], )
plt.xlabel
plt.ylabel
plt.title('Матрица ошибок')
plt.show()

```

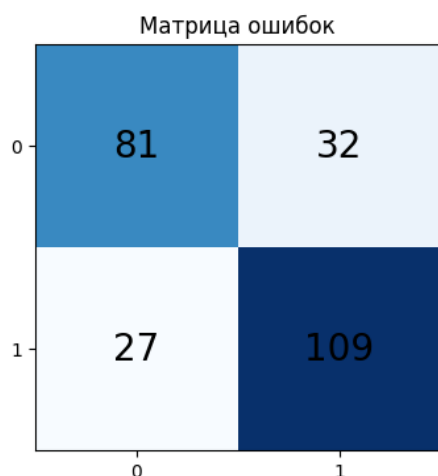
```

Accuracy: 0.7631
Precision: 0.7730
Recall: 0.8015
F1-score: 0.7870

Матрица ошибок:
[[ 81  32]
 [ 27 109]]

[ [TN, FP]
 [FN, TP] ]

```



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