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
Ярмак К. А.
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 12

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 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


ds_fileName = "datasets/medical_cost_personal_dataset.csv"
df = pd.read_csv(ds_fileName)


df_encoded = pd.get_dummies(df, drop_first=True)
print(df_encoded.head())
```

	age	bmi	children	charges	sex_male	smoker_yes	region_northwest	region_southeast	region_southwest
0	19	27.900	0	16884.92400	False	True	False	False	True
1	18	33.770	1	1725.55230	True	False	False	True	False
2	28	33.000	3	4449.46200	True	False	False	True	False
3	33	22.705	0	21984.47061	True	False	True	False	False
4	32	28.880	0	3866.85520	True	False	True	False	False

- обучите модель линейной регрессии для предсказания charges;
- рассчитайте MAE (Mean Absolute Error) и R^2;

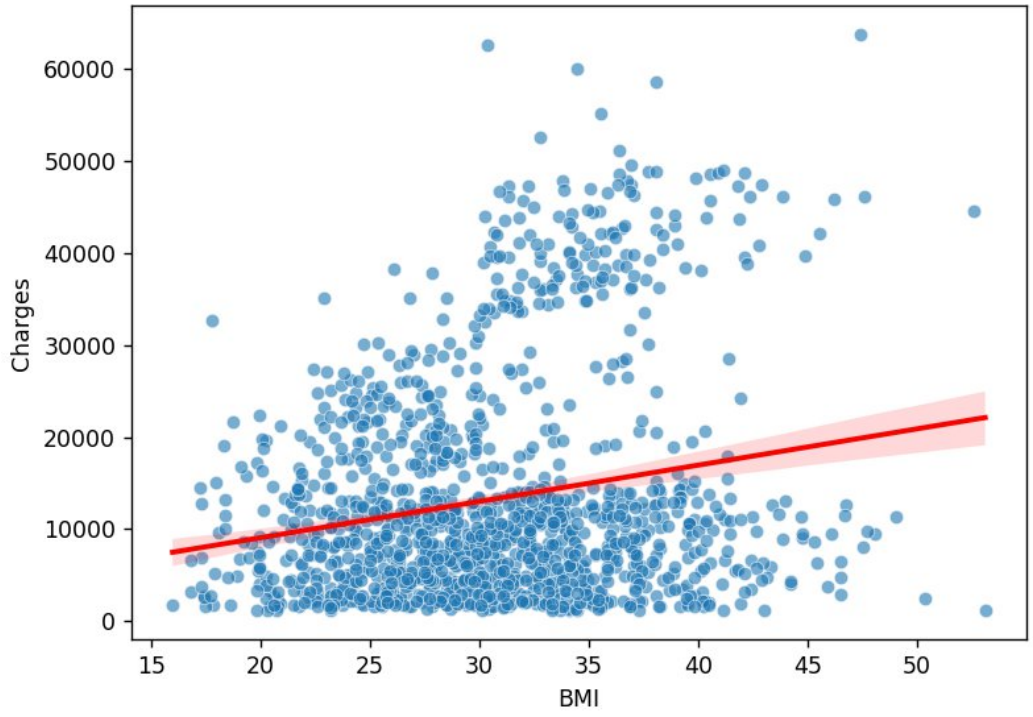
```
X = df_encoded.drop("charges", axis=1)
y = df_encoded["charges"]


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:.3f}")
```

```
MAE: 4181.19
R²: 0.784
```

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

```
plt.figure(figsize=(7,5))
sns.scatterplot(x=df["bmi"], y=df["charges"], alpha=0.6)
sns.regplot(x=df["bmi"], y=df["charges"], scatter=False, color="red")
plt.xlabel("BMI")
plt.ylabel("Charges")
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;
    - обучите модель логистической регрессии;

```
from matplotlib import pyplot as plt
import pandas as pd
import seaborn
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score,
f1_score, confusion_matrix, ConfusionMatrixDisplay


ds_fileName = "datasets/heart_disease_uci.csv"
df = pd.read_csv(ds_fileName)

df['target'] = df['num'].apply(lambda x: 1 if x > 0 else 0)
df.drop(columns=['id', 'dataset', 'num'], inplace=True)
df.replace({'TRUE': 1, 'FALSE': 0}, inplace=True)

df_encoded = pd.get_dummies(df, drop_first=True)
df_encoded.dropna(inplace=True)

X = df_encoded.drop('target', axis=1)
y = df_encoded['target']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

model = LogisticRegression(max_iter=100000)
model.fit(X_train, y_train)

print(f"Train size: {X_train.shape[0]}")
print(f"Test size: {X_test.shape[0]}")
```

```
Train size: 246
Test size: 62
```

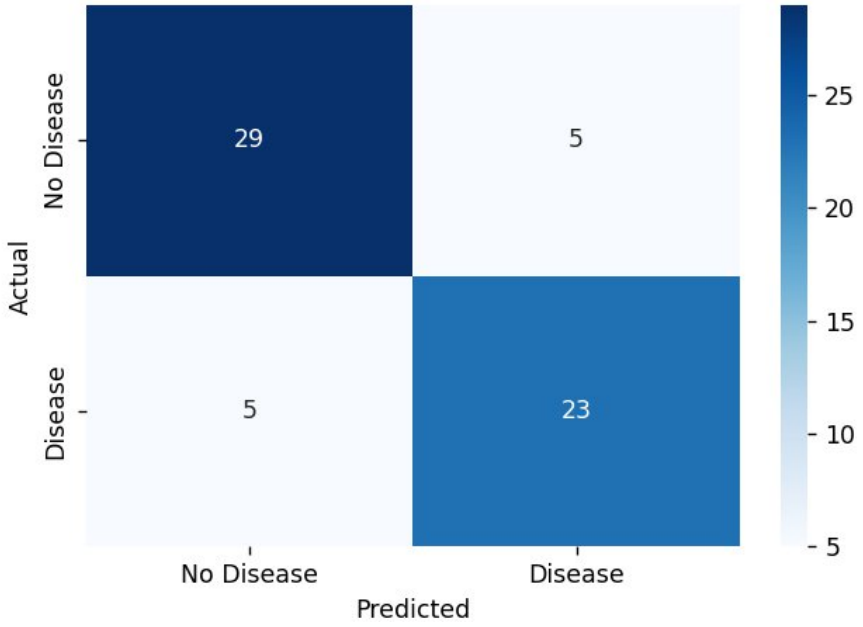
- оцените модель с помощью Accuracy, Precision, Recall и F1-score;

```
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:.3f}")
print(f"Precision: {precision:.3f}")
print(f"Recall:    {recall:.3f}")
print(f"F1-score:  {f1:.3f}")
```

```
Test size: 62
Accuracy:  0.839
Precision: 0.821
Recall:    0.821
F1-score:  0.821
```

- постройте матрицу ошибок.

```
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(6,4))
seaborn.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=["No
Disease", "Disease"], yticklabels=["No Disease", "Disease"])
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```



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