Рубежный контроль № 2

Тема: Методы построения моделей машинного обучения.

Студентка:

Калашникова Анастасия

Группа:

ИУ5-64

Вариант:

4

Задание.

Для заданного набора данных построить модели классификации или регрессии (в зависимости от конкретной задачи, рассматриваемой в наборе данных). Для построения моделей использовать методы Линейная/логистическая регрессия и Градиентный бустинг. Оценить качество моделей на основе подходящих метрик качества (не менее двух метрик). Какие метрики качества использовались и почему? Какие выводы можно сделать о качестве построенных моделей? Для построения моделей необходимо выполнить требуемую предобработку данных: заполнение пропусков, кодирование категориальных признаков, и т.д.

Выполнение

Импорт библиотек

In [67]:

```
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import plot_confusion_matrix
from sklearn.metrics import precision_score, recall_score, f1_score, classification_report
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
```

Загрузка данных

Набор данных 4го варианта - https://www.kaggle.com/carlolepelaars/toy-dataset

In [4]:

```
data = pd.read_csv('toy_dataset.csv', sep=",")
```

In [5]:

```
# Первые 5 строк датасета
data.head()
```

Out[5]:

	Number	City	Gender	Age	Income	Illness
0	1	Dallas	Male	41	40367.0	No
1	2	Dallas	Male	54	45084.0	No
2	3	Dallas	Male	42	52483.0	No
3	4	Dallas	Male	40	40941.0	No
4	5	Dallas	Male	46	50289.0	No

In [6]:

```
# Размер датасета
data.shape
```

Out[6]:

```
(150000, 6)
```

In [7]:

```
# Проверим есть ли пропущенные значения
data.isnull().sum()
```

Out[7]:

```
Number     0
City       0
Gender     0
Age        0
Income     0
Illness    0
dtype: int64
```

После проверки на пустые значения видно, что нет ни одного пропуска в данных. Можно перейти к кодированию категориальных признаков.

In [30]:

```
# Кодировка гендерного признака методом LabelEncoder
le = LabelEncoder()
data.loc[:,'Gender'] = le.fit_transform(data['Gender'])
data['Gender'].head()
```

Out[30]:

```
0    1
1    1
2    1
3    1
4    1
Name: Gender, dtype: int64
```

In [32]:

```
data.loc[:,'Illness'] = le.fit_transform(data['Illness'])
data['Illness'].head()
```

Out[32]:

```
0    0
1    0
2    0
3    0
4    0
Name: Illness, dtype: int64
```

In [22]:

```
cat_enc_c = data['City']
```

In [23]:

```
# Уникальные значения признака City
cat_enc_c.unique()
```

Out[23]:

```
array(['Dallas', 'New York City', 'Los Angeles', 'Mountain View',
       'Boston', 'Washington D.C.', 'San Diego', 'Austin'], dtype=object)
```

In [28]:

```
# Быстрый способ one-hot кодирования
one_hot = pd.get_dummies(cat_enc_c)
one_hot.head()
```

Out[28]:

	Austin	Boston	Dallas	Los Angeles	Mountain View	New York City	San Diego	Washington D.C.
0	0	0	1	0	0	0	0	0
1	0	0	1	0	0	0	0	0
2	0	0	1	0	0	0	0	0
3	0	0	1	0	0	0	0	0
4	0	0	1	0	0	0	0	0

In [33]:

```
data = data.join(one_hot)
```

```
data.drop(columns = 'City', inplace = True)
```

In [34]:

```
data.head()
```

Out[34]:

	Number	Gender	Age	Income	Illness	Austin	Boston	Dallas	Los Angeles	Mountain View	New York City	San Diego	Washington D.C.
0	1	1	41	40367.0	0	0	0	1	0	0	0	0	0
1	2	1	54	45084.0	0	0	0	1	0	0	0	0	0
2	3	1	42	52483.0	0	0	0	1	0	0	0	0	0
3	4	1	40	40941.0	0	0	0	1	0	0	0	0	0
4	5	1	46	50289.0	0	0	0	1	0	0	0	0	0

In [37]:

```
# Разделим выборку на обучающую и тестовую
X = data.drop('Illness', axis = 1)
Y = data['Illness']
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, random_state=1)
```

Модели

Линейная регрессия

In [77]:

```
model_log = LinearRegression().fit(X_train, Y_train)
Y_pred = model_log.predict(X_test)

# print('Коэффициенты b1: \n', reg.coef_)

print(f'Среднеквадратичная ошибка: {mean_squared_error(Y_test, Y_pred)}')
# The coefficient of determination: 1 is perfect prediction
print(f'Средняя абсолютная ошибка: {mean_absolute_error(Y_test, Y_pred)}')
print(f'Median absolute error: {median_absolute_error(Y_test, Y_pred)}')
print(f'R2 score: {r2_score(Y_test, Y_pred)}')
```

```
Среднеквадратичная ошибка: 0.07442103075914841
Средняя абсолютная ошибка: 0.1487809370260357
Median absolute error: 0.08151593804050852
R2 score: -0.00025909576016536207
```

Градиентный бустинг

In [79]:

```
model_boost = GradientBoostingRegressor(random_state=1)
model_boost.fit(X_train, Y_train)
Y_pred_boost = model_boost.predict(X_test)

print(f'Среднеквадратичная ошибка: {mean_squared_error(Y_test, Y_pred_boost)}')
# The coefficient of determination: 1 is perfect prediction
print(f'Средняя абсолютная ошибка: {mean_absolute_error(Y_test, Y_pred_boost)}')
print(f'Median absolute error: {median_absolute_error(Y_test, Y_pred_boost)}')
print(f'R2 score: {r2_score(Y_test, Y_pred_boost)}')
```

```
Среднеквадратичная ошибка: 0.07444516455788218
Средняя абсолютная ошибка: 0.14871425440542638
Median absolute error: 0.08084219121816509
R2 score: -0.0005834671300919414
```

Вывод

Для оценки линейной регрессии и градиентного бустинга были использованы метрики MSE, MAE, Median absolute error и R2-мера. Первые три метрики оказались примерно схожи для данной выборки, а отрицательная мера R2 показывает, что в выборке нет связности.

In [ ]: