Московский государственный технический университет им. Н.Э. Баумана
Кафедра «Системы обработки информации и управления»

Рубежный контроль №2
по дисциплине
«Технологии машинного обучения»
на тему
«Технологии использования и оценки моделей машинного обучения»

Вариант 4

Выполнил:
студент группы ИУ5-61Б
Белоусов Е. А.

Москва — 2020 г.

[1]:
```
import numpy as np
import pandas as pd
import sklearn
```

[2]:
```
data = pd.read_csv('../data/toy_dataset.csv', index_col= 'Number')
```

[3]:
```
data.head()
```

[3]:

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

[4]:
```
data.shape
```

[4]: (150000, 5)

[5]:
```
data.isnull().sum()
```

[5]:
```
City       0
Gender     0
Age        0
Income     0
Illness    0
dtype: int64
```

Попробуем разделить имеющиеся данные на два кластера: больные и здоровые

[6]:
```
y = data['Illness'].apply(lambda x: 0 if x == 'No' else 1)
del data['Illness']
```

[7]:
```
data['Gender'] = data['Gender'].apply(lambda x: 1 if x == 'Male' else 0)
```

[8]:
```
data['City'].unique()
```

[8]:
```
array(['Dallas', 'New York City', 'Los Angeles', 'Mountain View',
       'Boston', 'Washington D.C.', 'San Diego', 'Austin'], dtype=object)
```

[9]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
city = le.fit_transform(data['City'])
data['City'] = city
data
```

[9]:

Number	City	Gender	Age	Income
1	2	1	41	40367.0
2	2	1	54	45084.0
3	2	1	42	52483.0
4	2	1	40	40941.0

```
5         2     1   46   50289.0
…       …     …   …       …
149996    0     1   48   93669.0
149997    0     1   25   96748.0
149998    0     1   26  111885.0
149999    0     1   25  111878.0
150000    0     0   37   87251.0

[150000 rows x 4 columns]
```

[10]:
```
from sklearn.metrics import adjusted_rand_score
from sklearn.metrics import adjusted_mutual_info_score
from sklearn.metrics import homogeneity_completeness_v_measure

def score(results, values):
    print('Adjusted Rand index = ', adjusted_rand_score(results, values))
    print('Adjusted Mutual Information = ', adjusted_mutual_info_score(results, values))
    print('Homogeneity, completeness, V-measure = ',
    ↪homogeneity_completeness_v_measure(results, values))
```

[11]:
```
from sklearn.cluster import KMeans, DBSCAN
kmean = KMeans(n_clusters=2, random_state=42)
result = kmean.fit_predict(data)
score(result, y)
```

```
Adjusted Rand index =  0.0006781463519216322
Adjusted Mutual Information =  -6.264690343856899e-06
Homogeneity, completeness, V-measure =  (1.497293656412718e-06,
2.8233395226531545e-06, 1.956828169375787e-06)
```

[16]:
```
dbscan = DBSCAN()
result = dbscan.fit_predict(data)
score(result, y)
```

```
Adjusted Rand index =  0.0
Adjusted Mutual Information =  -8.521706962347577e-15
Homogeneity, completeness, V-measure =  (1.0, -5.283920588925548e-15,
-1.0567841177851152e-14)
```

Получаем результаты в обоих алгоритмах равные нулю, следовательно, мы не можем разделить больных людей от здоровых на основе имеющихся в наборе данных.