Московский Государственный Технический Университет
им. Н.Э. Баумана

Рубежный контроль №2

по курсу
Технологии Машинного Обучения

Выполнила:
Костян Алина
ИУ5-53

Проверил:
Гапанюк Ю.Е.

Москва, 2019

Задание

Данный вариант выполняется на основе материалов лекции.

Необходимо решить задачу кластеризации на основе любого выбранного Вами датасета.

Кластеризуйте данные с помощью трех различных алгоритмов кластеризации. Алгоритмы выбираются произвольным образом, рекомендуется использовать алгоритмы из лекции.

Сравните качество кластеризации для трех алгоритмов с помощью следующих метрик качества кластеризации:

- Adjusted Rand index
- Adjusted Mutual Information
- Homogeneity, completeness, V-measure
- Коэффициент силуэта
  Сделате выводы о том, какой алгоритм осуществляет более качественную кластеризацию на Вашем наборе данных

Код и результаты выполнения

1. Подключим библиотеки:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans, MiniBatchKMeans
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.impute import SimpleImputer
from sklearn.impute import MissingIndicator
%matplotlib inline
sns.set(style="ticks")
```

2. Подготовим данные

```
data.isnull().sum()
```

```
Serial No.           0
GRE Score            0
TOEFL Score          0
University Rating    0
SOP                  0
LOR                  0
CGPA                 0
Research             0
Chance of Admit      0
dtype: int64
```

Разделим выборку на тренировочную и тестовую

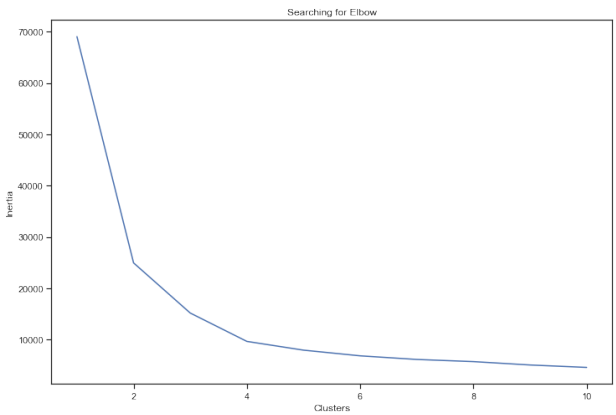
```
X_train, X_test, y_train, y_test = train_test_split(
    x, y, test_size=0.2, random_state=42)
```

3. Кластеризация

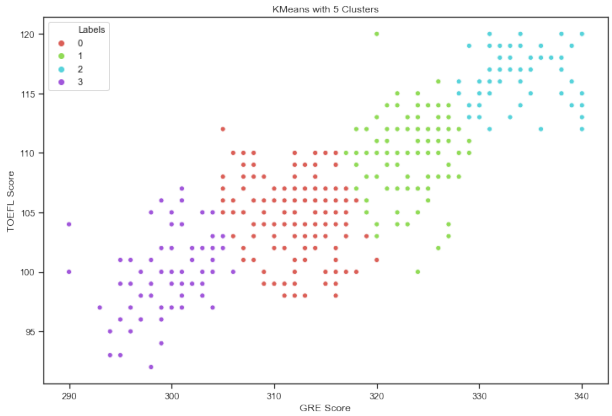
K-means

```
target=data['University Rating']
X = data.drop(['University Rating'], axis=1)
```

```
from sklearn.cluster import KMeans

clusters = []

for i in range(1, 11):
    km = KMeans(n_clusters=i).fit(X)
    clusters.append(km.inertia_)

fig, ax = plt.subplots(figsize=(12, 8))
sns.lineplot(x=list(range(1, 11)), y=clusters, ax=ax)
ax.set_title('Searching for Elbow')
ax.set_xlabel('Clusters')
ax.set_ylabel('Inertia')

plt.show()
```

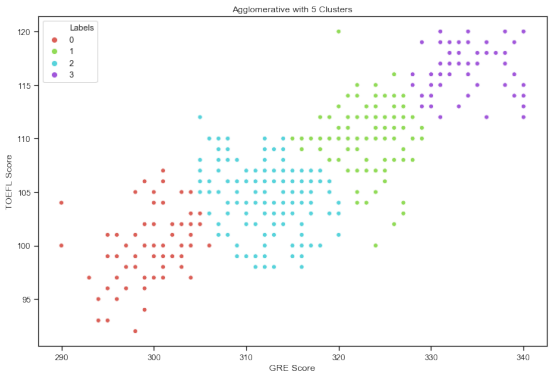


```
km5 = KMeans(n_clusters=4).fit(X)

X['Labels'] = km5.labels_
plt.figure(figsize=(12, 8))
sns.scatterplot(X['GRE Score'], X['TOEFL Score'], hue=X['Labels'],
                palette=sns.color_palette('hls', 4))
plt.title('KMeans with 5 Clusters')
plt.show()
```



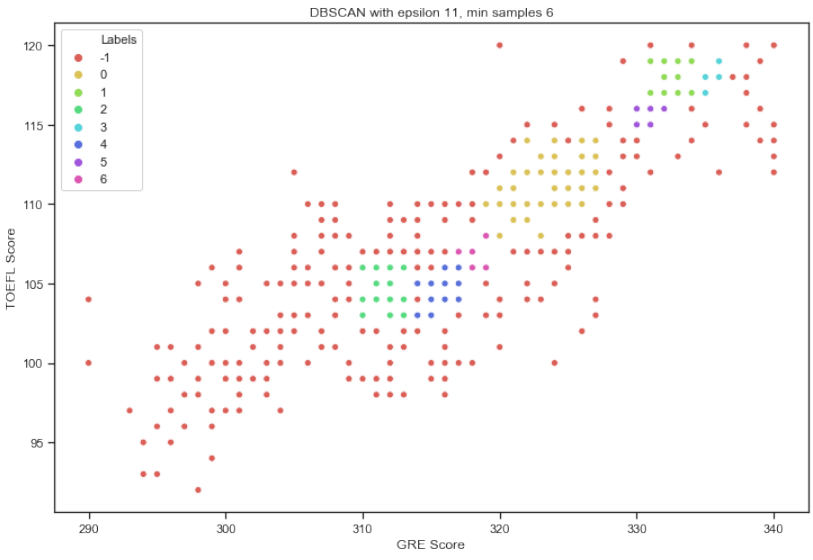
Иерархическая кластеризация

```
from sklearn.cluster import AgglomerativeClustering

agglom = AgglomerativeClustering(n_clusters=4, linkage='average').fit(X)

X['Labels'] = agglom.labels_
plt.figure(figsize=(12, 8))
sns.scatterplot(X['GRE Score'], X['TOEFL Score'], hue=X['Labels'],
                palette=sns.color_palette('hls', 4))
plt.title('Agglomerative with 5 Clusters')
plt.show()
```



DBSCAN

```
from sklearn.cluster import DBSCAN

db = DBSCAN(eps=2, min_samples=10).fit(X)

X['Labels'] = db.labels_
plt.figure(figsize=(12, 8))
sns.scatterplot(X['GRE Score'], X['TOEFL Score'], hue=X['Labels'],
                palette=sns.color_palette('hls', np.unique(db.labels_).shape[0]))
plt.title('DBSCAN with epsilon 11, min samples 6')
plt.show()
```



4. Метрики качества

Метрики

```
from sklearn import metrics
import pandas as pd
from sklearn.cluster import KMeans, AgglomerativeClustering, AffinityPropagation, SpectralClustering

algorithms = []
algorithms.append(KMeans(n_clusters=4, random_state=1))
algorithms.append(DBSCAN(eps=2, min_samples=10))
algorithms.append(AgglomerativeClustering(n_clusters=4))

y=target
data = []
for algo in algorithms:
    algo.fit(X)
    data.append(({
        'ARI': metrics.adjusted_rand_score(y, algo.labels_),
        'AMI': metrics.adjusted_mutual_info_score(y, algo.labels_),
        'Homogenity': metrics.homogeneity_score(y, algo.labels_),
        'Completeness': metrics.completeness_score(y, algo.labels_),
        'V-measure': metrics.v_measure_score(y, algo.labels_),
        'Silhouette': metrics.silhouette_score(X, algo.labels_)}))

results = pd.DataFrame(data=data, columns=['ARI', 'AMI', 'Homogenity',
                                           'Completeness', 'V-measure',
                                           'Silhouette'],
                       index=['K-means', 'DBSCAN', 'Agglomerative'])

results
```

	ARI	AMI	Homogenity	Completeness	V-measure	Silhouette
K-means	0.167845	0.218413	0.226443	0.254430	0.239622	0.443709
DBSCAN	0.002346	0.082551	0.099482	0.140120	0.116355	-0.170763
Agglomerative	0.144356	0.205386	0.213553	0.238365	0.225278	0.413737

Вывод

По данным полученным выше (ARI, AMI, Homogenity, Completeness, V-measure, Silhouette):

- Наиболее качественную кластеризацию осуществляет метод K-means
- Самый худший результат у алгоритма DBSCAN