Московский Государственный технический университет имени Н. Э. Баумана

Рулежный контроль по курсу: «Технология машинного обучения»

Работу выполнил студент группы ИУ5-63

Федорова Антонина_____

Работу проверил:

Гапанюк Ю.Е._____

Москва 2019

Задание:

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:

1. Adjusted Rand index
2. Adjusted Mutual Information
3. Homogeneity, completeness, V-measure
4. Коэффициент силуэта

Сделате выводы о том, какой алгоритм осуществляет более качественную кластеризацию на Вашем наборе данных.

Текст программы с примерами выполнения программы:

```
from google.colab import drive
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
from sklearn.cluster import KMeans
from sklearn.metrics import adjusted_rand_score
from sklearn.metrics import adjusted_mutual_info_score
from sklearn.metrics import homogeneity_completeness_v_measure
from sklearn.metrics import silhouette_score
from sklearn.cluster import DBSCAN
from sklearn.cluster import MeanShift
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.cluster import Birch
from sklearn.model_selection import GridSearchCV, KFold
#Монтирую гугл диск, чтобы взять оттуда датасет
drive.mount("/content/gdrive", force_remount=True)
#Загружаю данные с гугл диска
data = pd.read_csv('/content/gdrive/My Drive/mushrooms.csv', sep=«,")
data.head()
```

	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	stalk-shape	stalk-root	stalk-surface-above-ring	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color	n
0	p	x	s	n	t	p	f	c	n	k	e	e	s	s	w	w	p	w	
1	e	x	s	y	t	a	f	c	b	k	e	c	s	s	w	w	p	w	
2	e	b	s	w	t	l	f	c	b	n	e	c	s	s	w	w	p	w	
3	p	x	y	w	t	p	f	c	n	n	e	e	s	s	w	w	p	w	
4	e	x	s	g	f	n	f	w	b	k	t	e	s	s	w	w	p	w	

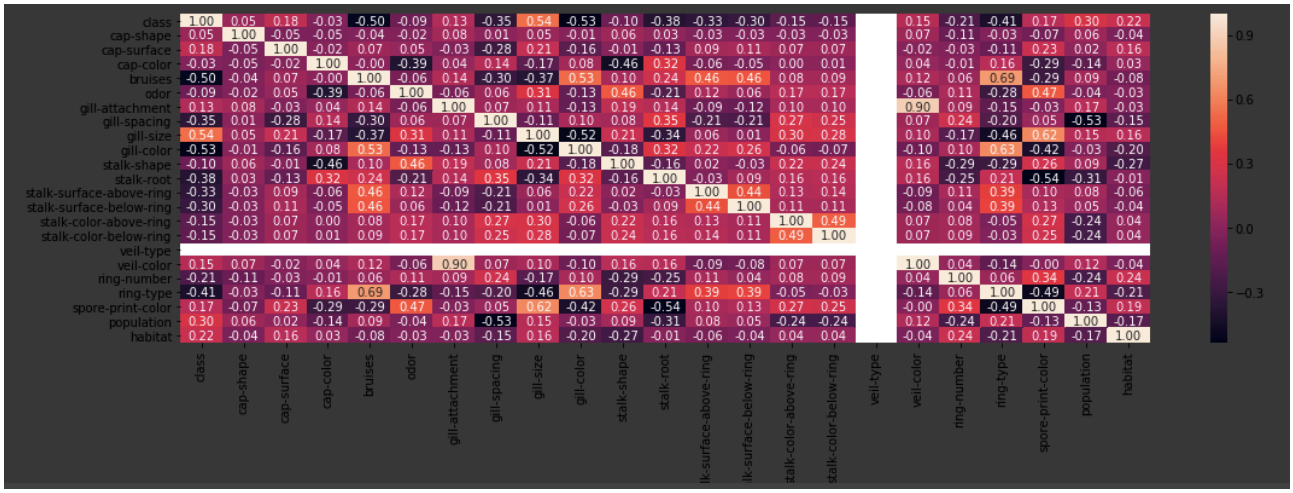
```
data.size
```

186852

```
le = LabelEncoder()
cols_x = ['cap-shape', 'cap-surface', 'cap-color', 'bruises',
          'odor', 'gill-attachment', 'gill-spacing', 'gill-size',
          'gill-color', 'stalk-shape', 'stalk-root', 'stalk-surface-above-ring',
          'stalk-surface-below-ring', 'stalk-color-above-ring',
          'stalk-color-below-ring', 'veil-type', 'veil-color', 'ring-number',
          'ring-type', 'spore-print-color', 'population', 'habitat']
col_y = ‘class'
for i in cols_x:
  data[i] = le.fit_transform(data[[i]])
data['class'] = le.fit_transform(data[[‘class']])
data.head()
```

	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	stalk-shape	stalk-root	stalk-surface-above-ring	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring
0	1	5	2	4	1	6	1	0	1	4	0	3	2	2	7	7
1	0	5	2	9	1	0	1	0	0	4	0	2	2	2	7	7
2	0	0	2	8	1	3	1	0	0	5	0	2	2	2	7	7
3	1	5	3	8	1	6	1	0	1	5	0	3	2	2	7	7
4	0	5	2	3	0	5	1	1	0	4	1	3	2	2	7	7

```
plt.figure(figsize = (18,5))
sns.heatmap(data.corr(method='pearson'), annot=True, fmt='.2f', square=False)
```




```
cols_x2 = ['bruises', 'gill-spacing', 'gill-size', 'gill-color',  'stalk-root',
          'stalk-surface-above-ring', 'stalk-surface-below-ring', 'ring-type',
          'population', ]
X = data[cols_x2]
Y = data[col_y]
```

Mean Shift

```
[ ]  temp_cluster_ms = MeanShift().fit_predict(X)
```

```
[ ]  ari = adjusted_rand_score(Y, temp_cluster_ms)
     ami = adjusted_mutual_info_score(Y, temp_cluster_ms)
     h, c, v = homogeneity_completeness_v_measure(Y, temp_cluster_ms)
     sl = silhouette_score(X, temp_cluster_ms)
     print('''ARI: {0},
     AMI:{1},
     Homogeneity:{2},
     Completeness: {3},
     V-measure: {4},
     Silhouette: {5}'''.format(ari, ami, h, c, v, sl))
```

```
:/local/lib/python3.6/dist-packages/sklearn/metrics/cluster/supervi
ıtureWarning)
: 0.3069603675011273,
:0.2885523801640643,
ɔgeneity:0.39129684154183636,
ɔleteness: 0.2886585628469996,
easure: 0.33223115279908827,
ıouette: 0.4531091565478523
```

DBSCAN

```
[ ]  temp_cluster_db = DBSCAN(eps=0.99).fit_predict(X)
```

```
[ ]  ari = adjusted_rand_score(Y, temp_cluster_db)
     ami = adjusted_mutual_info_score(Y, temp_cluster_db)
     h, c, v = homogeneity_completeness_v_measure(Y, temp_cluster_db)
     sl = silhouette_score(X, temp_cluster_db)
     print('''ARI: {0},
     AMI:{1},
     Homogeneity:{2},
     Completeness: {3},
     V-measure: {4},
     Silhouette:{5}'''.format(ari, ami, h, c, v, sl))
```

```
ocal/lib/python3.6/dist-packages/sklearn/metrics/cluster/supervised
reWarning)
.045137012766120144,
15358923643298394,
neity:0.9728493007154457,
teness: 0.15573378177437797,
ure: 0.26848798825290127,
ette:0.9998172438350283
```

Birch

```
[19]  temp_cluster_br = Birch().fit_predict(X)
```

```
[20]  ari = adjusted_rand_score(Y, temp_cluster_br)
      ami = adjusted_mutual_info_score(Y, temp_cluster_br)
      h, c, v = homogeneity_completeness_v_measure(Y, temp_cluster_br)
      sl = silhouette_score(X, temp_cluster_br)
      print('''ARI: {0},
      AMI:{1},
      Homogeneity:{2},
      Completeness: {3},
      V-measure: {4},
      Silhouette: {5}'''.format(ari, ami, h, c, v, sl))
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/metrics/cluste
  FutureWarning)
ARI: 0.30438499877111097,
AMI:0.269229255154706,
Homogeneity:0.361660376699665,
Completeness: 0.26932602881101514,
V-measure: 0.30873740601743555,
Silhouette: 0.462697384729021
```

```
[33] n_range_br = np.array(np.arange(0.01,1,0.1))
     tuned_parameters_br = [{'threshold': n_range_br}]
     tuned_parameters_br
```

```
[{'threshold': array([0.01, 0.11, 0.21, 0.31, 0.41, 0.51, 0.61, 0.71, 0.81, 0.91])}]
```

```
[40] br_gs = GridSearchCV(Birch(), tuned_parameters_br, cv=KFold(n_splits=5), scoring='adjusted_mutual_info_score')
     br_gs.fit(data[cols_x2], data[col_y])
     br_gs.best_params_
```

```
[45] temp_cluster_br_gs = br_gs.best_estimator_.fit_predict(X)
```

```
[44] ari = adjusted_rand_score(Y, temp_cluster_br_gs)
     ami = adjusted_mutual_info_score(Y, temp_cluster_br_gs)
     h, c, v = homogeneity_completeness_v_measure(Y, temp_cluster_br_gs)
     sl = silhouette_score(X, temp_cluster_br_gs)
     print('''ARI: {0},
     AMI:{1},
     Homogeneity:{2},
     Completeness: {3},
     V-measure: {4},
     Silhouette: {5}'''.format(ari, ami, h, c, v, sl))
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/metrics/cluster/supervised.py:746: FutureWarning: The behavio
  FutureWarning)
ARI: 0.13627703099355284,
AMI:0.1765824873001168,
Homogeneity:0.20769636588400608,
Completeness: 0.1767070717384534,
V-measure: 0.1909525932082184,
Silhouette: 0.2787015966914067
```