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

Рубежный контроль №2
по дисциплине
«Методы машинного обучения»

Выполнил:
Хотин П.Ю.
ИУ5-24М

Москва, 2020 год

Задание

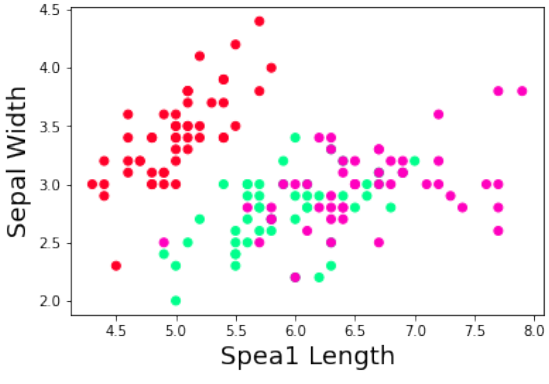
1. решить задачу кластеризации с использованием методов:

1) MeanShift
2) спектральная кластеризация
3) иерархическая кластеризация.

2. Оценить качество модели на основе подходящих метрик качества (не менее двух метрик, если это возможно).

3. Сделать выводы о качестве построенных моделей?

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from sklearn import preprocessing
from sklearn.datasets import load_iris

iris = load_iris()

X = iris.data[:, :2]
Y = iris.target

plt.scatter(X[:,0], X[:,1], c=Y, cmap='gist_rainbow')
plt.xlabel('Spea1 Length', fontsize=18)
plt.ylabel('Sepal Width', fontsize=18)

Text(0, 0.5, 'Sepal Width')
```



```
from sklearn.cluster import MeanShift

ms = MeanShift()
ms.fit(X)
```

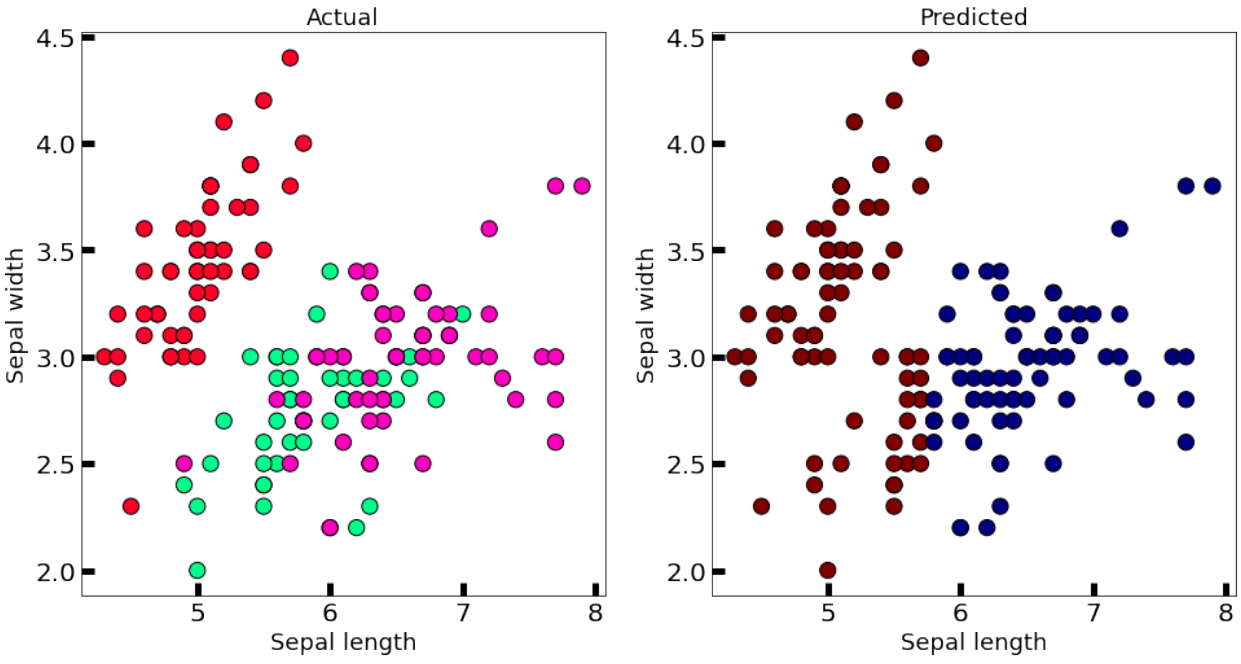
```
MeanShift(bandwidth=None, bin_seeding=False, cluster_all=True,
max_iter=300,
          min_bin_freq=1, n_jobs=None, seeds=None)
```

```
centers = ms.cluster_centers_
centers
```

```
array([[6.22      , 2.892     ],
       [5.41142857, 3.03285714]])
```

```
new_labels = ms.labels_
# Plot the identified clusters and compare with the answers
fig, axes = plt.subplots(1, 2, figsize=(16,8))
axes[0].scatter(X[:, 0], X[:, 1], c=Y, cmap='gist_rainbow',
edgecolor='k', s=150)
axes[1].scatter(X[:, 0], X[:, 1], c=new_labels, cmap='jet',
edgecolor='k', s=150)
axes[0].set_xlabel('Sepal length', fontsize=18)
axes[0].set_ylabel('Sepal width', fontsize=18)
axes[1].set_xlabel('Sepal length', fontsize=18)
axes[1].set_ylabel('Sepal width', fontsize=18)
axes[0].tick_params(direction='in', length=10, width=5, colors='k',
labelsize=20)
axes[1].tick_params(direction='in', length=10, width=5, colors='k',
labelsize=20)
axes[0].set_title('Actual', fontsize=18)
axes[1].set_title('Predicted', fontsize=18)
```

```
Text(0.5, 1.0, 'Predicted')
```



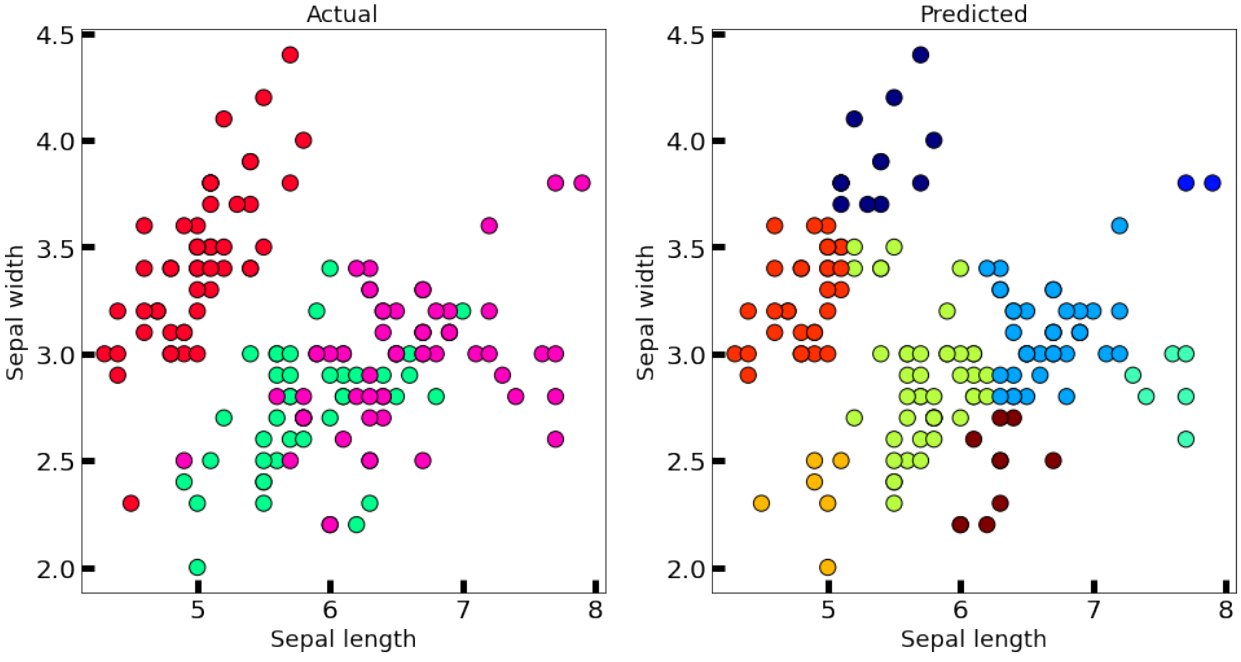

```
from                    sklearn.cluster import
SpectralClustering

sc = SpectralClustering()
sc.fit(X)
```

```
SpectralClustering(affinity='rbf', assign_labels='kmeans', coef0=1,
degree=3,
                   eigen_solver=None, eigen_tol=0.0, gamma=1.0,
                   kernel_params=None, n_clusters=8,
n_components=None,
                   n_init=10, n_jobs=None, n_neighbors=10,
random_state=None)
```

```
new_labels = sc.labels_
# Plot the identified clusters and compare with the answers
fig, axes = plt.subplots(1, 2, figsize=(16,8))
axes[0].scatter(X[:, 0], X[:, 1], c=Y, cmap='gist_rainbow',
edgecolor='k', s=150)
axes[1].scatter(X[:, 0], X[:, 1], c=new_labels, cmap='jet',
```

```
edgecolor='k', s=150)
axes[0].set_xlabel('Sepal length', fontsize=18)
axes[0].set_ylabel('Sepal width', fontsize=18)
axes[1].set_xlabel('Sepal length', fontsize=18)
axes[1].set_ylabel('Sepal width', fontsize=18)
axes[0].tick_params(direction='in', length=10, width=5, colors='k',
labelsize=20)
axes[1].tick_params(direction='in', length=10, width=5, colors='k',
labelsize=20)
axes[0].set_title('Actual', fontsize=18)
axes[1].set_title('Predicted', fontsize=18)
```

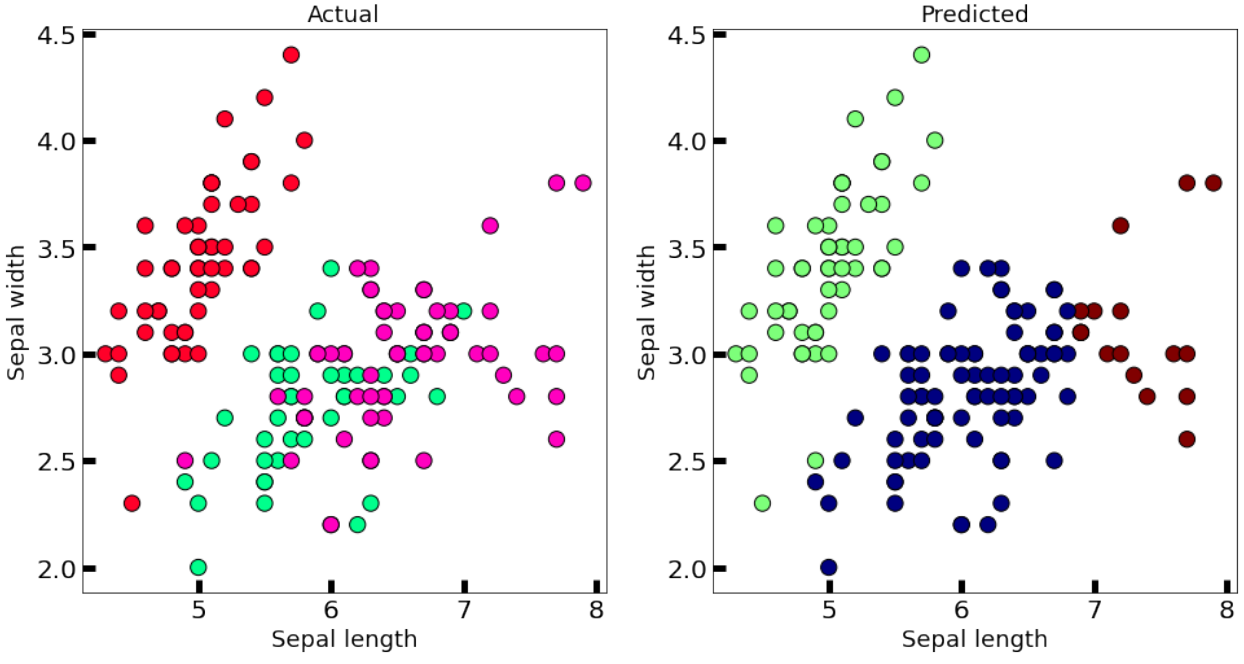
```
Text(0.5, 1.0, 'Predicted')
```




```
sc2 =
SpectralClustering(n_clusters=3)
sc2.fit(X)
```

```
SpectralClustering(affinity='rbf', assign_labels='kmeans', coef0=1,
degree=3,
                   eigen_solver=None, eigen_tol=0.0, gamma=1.0,
                   kernel_params=None, n_clusters=3,
```

```
n_components=None,
                    n_init=10, n_jobs=None, n_neighbors=10,
random_state=None)

new_labels = sc2.labels_
# Plot the identified clusters and compare with the answers
fig, axes = plt.subplots(1, 2, figsize=(16,8))
axes[0].scatter(X[:, 0], X[:, 1], c=Y, cmap='gist_rainbow',
edgecolor='k', s=150)
axes[1].scatter(X[:, 0], X[:, 1], c=new_labels, cmap='jet',
edgecolor='k', s=150)
axes[0].set_xlabel('Sepal length', fontsize=18)
axes[0].set_ylabel('Sepal width', fontsize=18)
axes[1].set_xlabel('Sepal length', fontsize=18)
axes[1].set_ylabel('Sepal width', fontsize=18)
axes[0].tick_params(direction='in', length=10, width=5, colors='k',
labelsize=20)
axes[1].tick_params(direction='in', length=10, width=5, colors='k',
labelsize=20)
axes[0].set_title('Actual', fontsize=18)
axes[1].set_title('Predicted', fontsize=18)

Text(0.5, 1.0, 'Predicted')
```

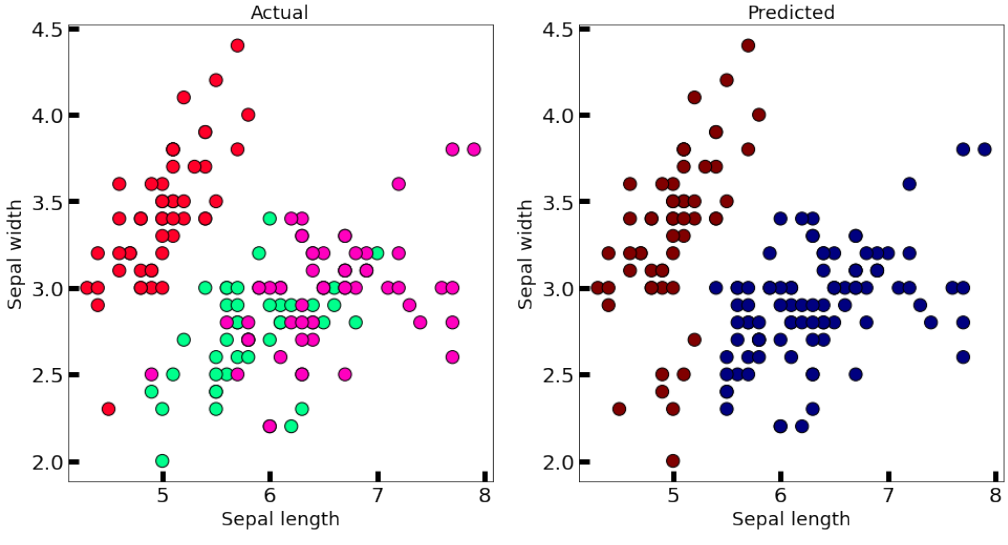


```
from sklearn.cluster import AgglomerativeClustering

ag = AgglomerativeClustering()
ag.fit(X)
```

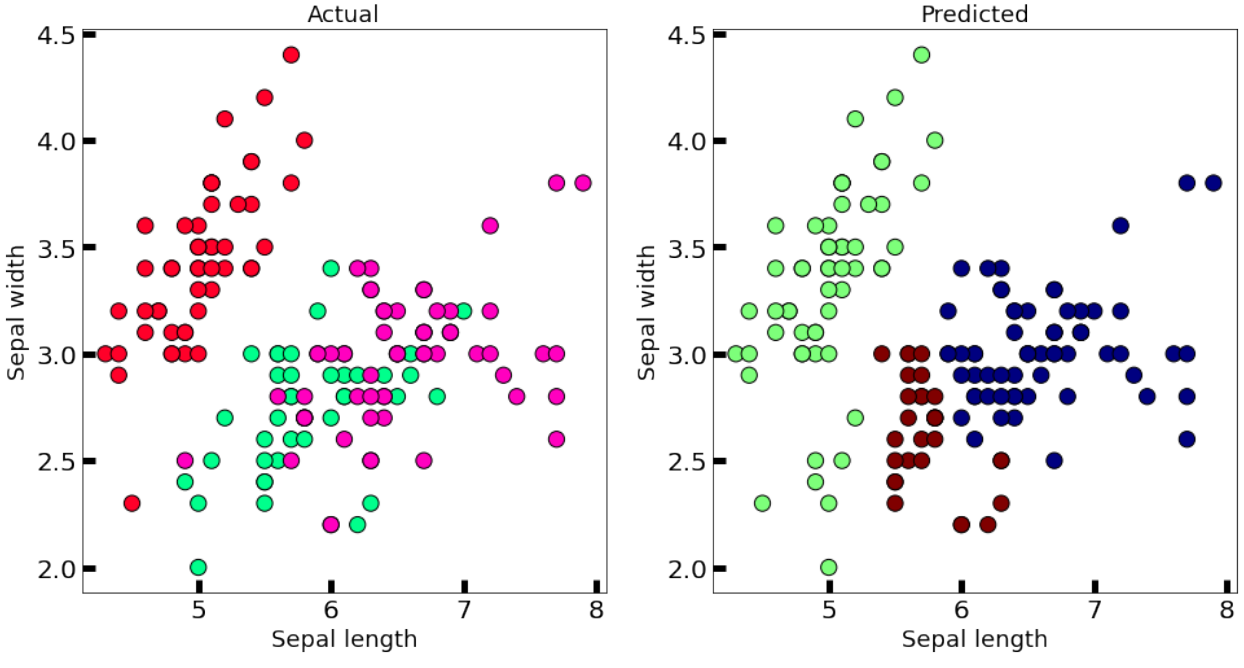
```
AgglomerativeClustering(affinity='euclidean',
compute_full_tree='auto',
                        connectivity=None, distance_threshold=None,
                        linkage='ward', memory=None, n_clusters=2)
```

```
new_labels = ag.labels_
# Plot the identified clusters and compare with the answers
fig, axes = plt.subplots(1, 2, figsize=(16,8))
axes[0].scatter(X[:, 0], X[:, 1], c=Y, cmap='gist_rainbow',
edgecolor='k', s=150)
axes[1].scatter(X[:, 0], X[:, 1], c=new_labels, cmap='jet',
edgecolor='k', s=150)
axes[0].set_xlabel('Sepal length', fontsize=18)
axes[0].set_ylabel('Sepal width', fontsize=18)
axes[1].set_xlabel('Sepal length', fontsize=18)
axes[1].set_ylabel('Sepal width', fontsize=18)
axes[0].tick_params(direction='in', length=10, width=5, colors='k',
labelsize=20)
axes[1].tick_params(direction='in', length=10, width=5, colors='k',
labelsize=20)
axes[0].set_title('Actual', fontsize=18)
axes[1].set_title('Predicted', fontsize=18)
```

```
Text(0.5, 1.0, 'Predicted')
```



```
ag2 = AgglomerativeClustering(n_clusters=3)
ag2.fit(X)

new_labels = ag2.labels_
# Plot the identified clusters and compare with the answers
fig, axes = plt.subplots(1, 2, figsize=(16,8))
axes[0].scatter(X[:, 0], X[:, 1], c=Y, cmap='gist_rainbow',
edgecolor='k', s=150)
axes[1].scatter(X[:, 0], X[:, 1], c=new_labels, cmap='jet',
edgecolor='k', s=150)
axes[0].set_xlabel('Sepal length', fontsize=18)
axes[0].set_ylabel('Sepal width', fontsize=18)
axes[1].set_xlabel('Sepal length', fontsize=18)
axes[1].set_ylabel('Sepal width', fontsize=18)
axes[0].tick_params(direction='in', length=10, width=5, colors='k',
labelsize=20)
axes[1].tick_params(direction='in', length=10, width=5, colors='k',
labelsize=20)
axes[0].set_title('Actual', fontsize=18)
axes[1].set_title('Predicted', fontsize=18)

Text(0.5, 1.0, 'Predicted')
```



Метрики

```
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

def count_metrics(name, method):
    tmp = method.fit_predict(X)
    print("Dataset: " + name)
    print("ARI: "+ str(adjusted_rand_score(Y, tmp)))
    print("AMI: "+  str(adjusted_mutual_info_score(Y, tmp)))
    h, c, v = homogeneity_completeness_v_measure(Y, tmp)
    print("HCVm: Homogeneity - " + str(h) + "\nCompleteness - " +
str(c) + "\nV-measure - "+str(v))
    print("SL: " + str(silhouette_score(X, tmp)))
    print("==================================")

count_metrics("MeanShift", MeanShift())
count_metrics("Spectral default", SpectralClustering())
count_metrics("Spectral 3", SpectralClustering(n_clusters=3))
count_metrics("Agglomerative def", AgglomerativeClustering())
count_metrics("Agglomerative 3",
AgglomerativeClustering(n_clusters=3))
```

```
Dataset: MeanShift
ARI: 0.3944401908806803
AMI: 0.4317743582900882
HCVm: Homogeneity - 0.355574438925241
Completeness - 0.5636444355672562
V-measure - 0.43606057162569084
SL: 0.4644681851183547
==================================
Dataset: Spectral default
ARI: 0.3103895058381067
AMI: 0.4078924556814485
HCVm: Homogeneity - 0.5607839300056536
Completeness - 0.34798815614173934
```

```
V-measure - 0.4294721828965487
SL: 0.36454192316615136
==================================
Dataset: Spectral 3
ARI: 0.5529473055759424
AMI: 0.6353736832348081
HCVm: Homogeneity - 0.595146173209358
Completeness - 0.6928341566599039
V-measure - 0.6402855500855817
SL: 0.4131437626307253
==================================
Dataset: Agglomerative def
ARI: 0.5114270772970757
AMI: 0.5875852748543551
HCVm: Homogeneity - 0.4730196835308308
Completeness - 0.7865303025387341
V-measure - 0.590757522780414
SL: 0.47767996898758924
==================================
Dataset: Agglomerative 3
ARI: 0.5112126489117526
AMI: 0.5240179186847511
HCVm: Homogeneity - 0.5190720845536648
Completeness - 0.5414839345877656
V-measure - 0.5300412040588491
SL: 0.3653346819163389
==================================
```

Были использованы метрики кластеризации ARI (так как известны истинные метки), AMI, HSV, коэфф. силуэта).

По результатам можно сказать, что лучше всего справились спектральная классификация и иерархическая, так как ARI у них ближе к 1. Но значение коэффицентов все равно небольшие, поэтому сказать, что модель получилась хорошего качества, нельзя.