Live 2025-10-22

October 22, 2025

1 Clustering on Iris Data - Continued

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
X = sns.load_dataset('iris')
```

[3]:
```
y = X.species
X = X.drop('species',axis=1)
```

[4]:
```
X.head()
```

[4]:
```
   sepal_length  sepal_width  petal_length  petal_width
0           5.1          3.5           1.4          0.2
1           4.9          3.0           1.4          0.2
2           4.7          3.2           1.3          0.2
3           4.6          3.1           1.5          0.2
4           5.0          3.6           1.4          0.2
```

[5]:
```
y.head()
```

[5]:
```
0    setosa
1    setosa
2    setosa
3    setosa
4    setosa
Name: species, dtype: object
```

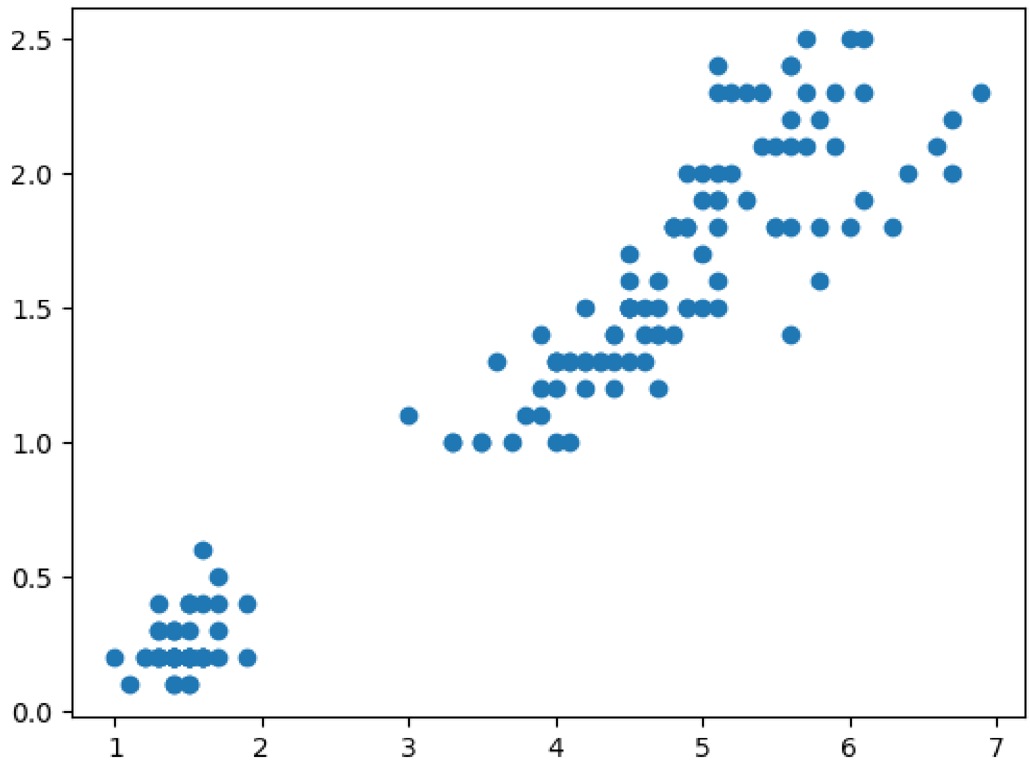
X is now the matrix with the feature values

y is a series with the target value for each set of features

Let's start with unsupervised learning and not use y

[6]:
```
plt.scatter(X.petal_length,X.petal_width)
```

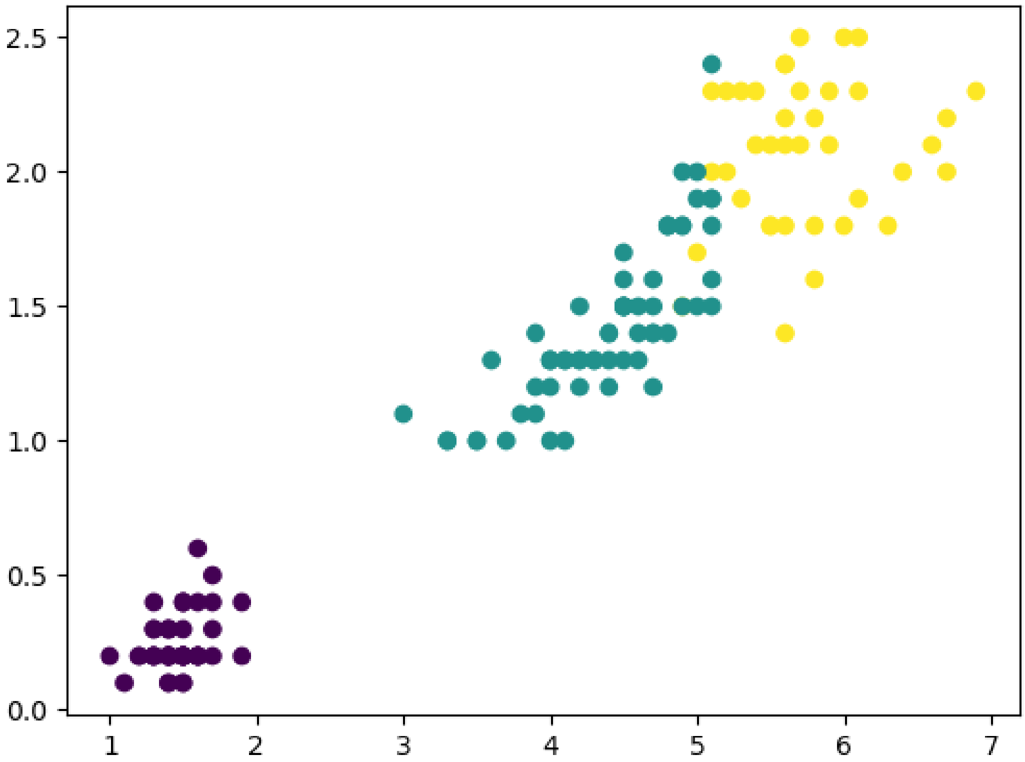
[6]:
```
<matplotlib.collections.PathCollection at 0x7f8edbb6dc60>
```



Let’s find some clusters with sklearn

[7]:
```
from sklearn.cluster import KMeans

clusterer = KMeans(
    n_clusters=3,
    init='random',
    random_state=42)

clusterer.fit(X)

y_pred = pd.Series(clusterer.predict(X))

plt.scatter(X.petal_length,X.petal_width,c=y_pred)
```

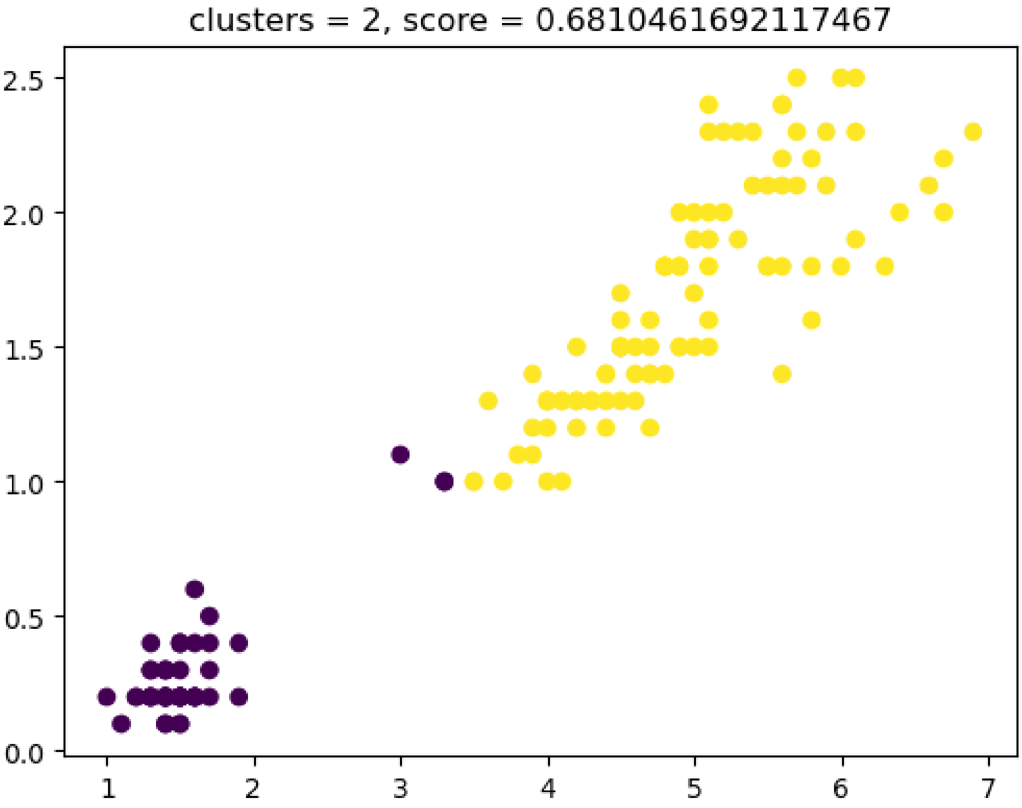
[7]:
```
<matplotlib.collections.PathCollection at 0x7f8ed272f0a0>
```

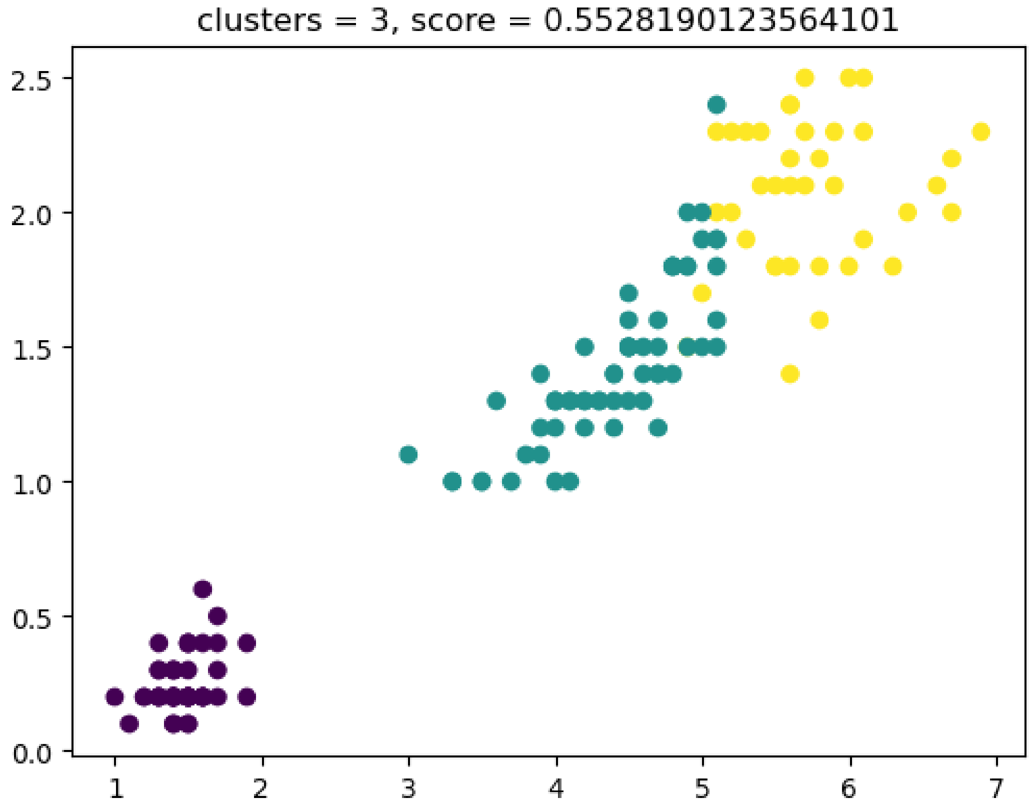


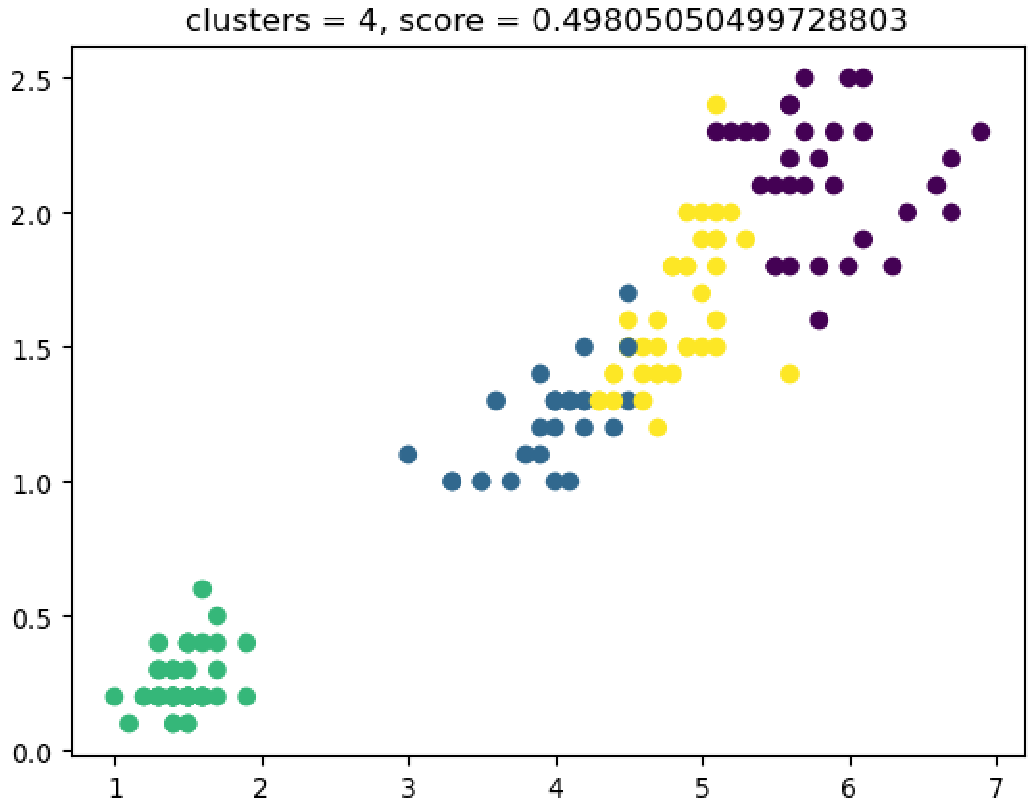
Let's try some metrics

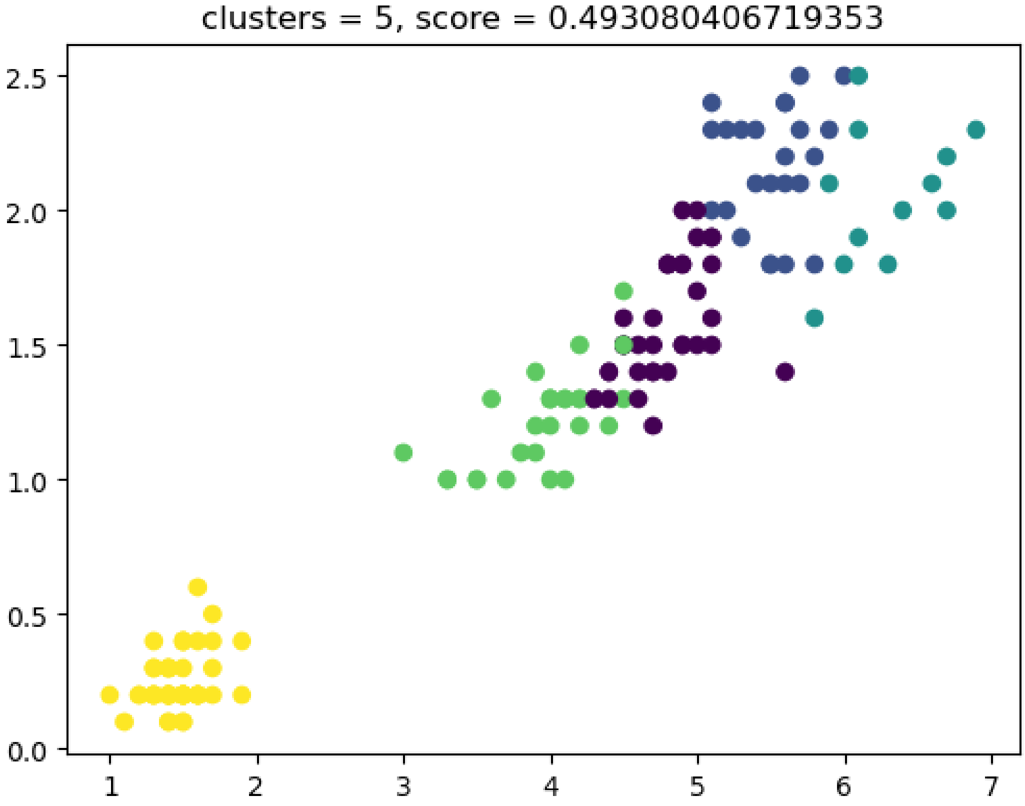
[8]:
```
from sklearn import metrics
from sklearn.metrics import pairwise_distances
```

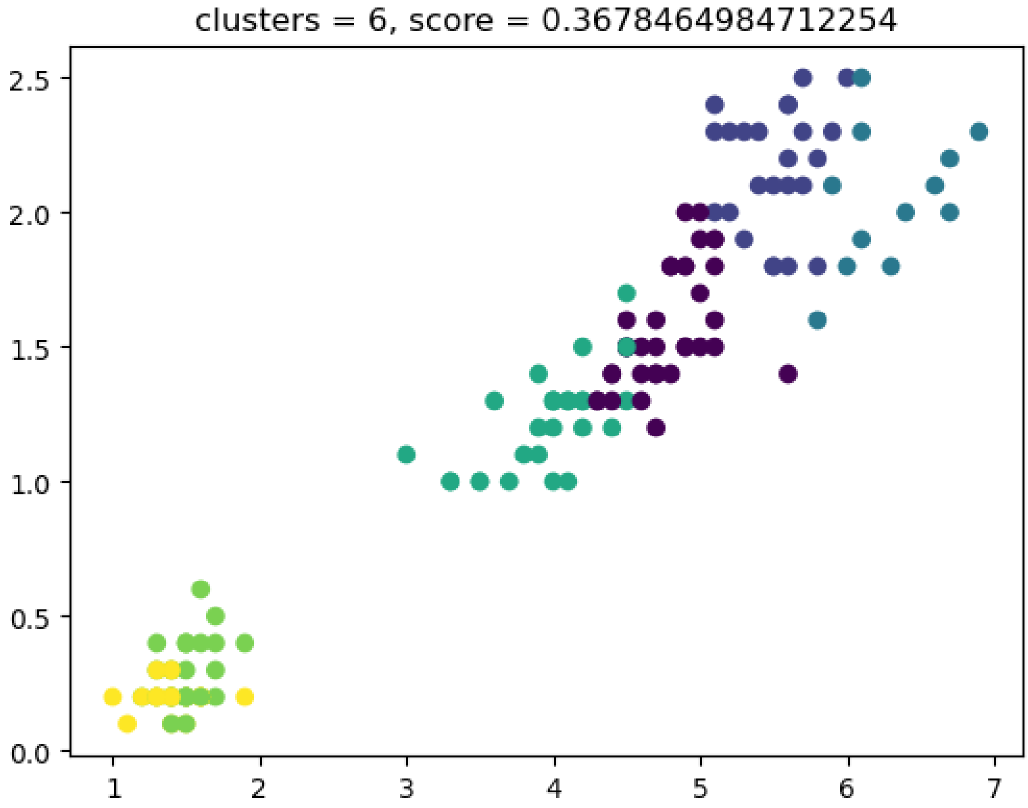
[9]:
```
for nC in range(2,11):
    clusterer = KMeans(
    n_clusters=nC,
    init='random',
    random_state=42)
    clusterer.fit(X)
    y_pred = pd.Series(clusterer.predict(X))
    score = metrics.silhouette_score(X,y_pred)

    plt.scatter(X.petal_length,X.petal_width,c=y_pred)
    plt.title(f"clusters = {nC}, score = {score}")
    plt.show()
```

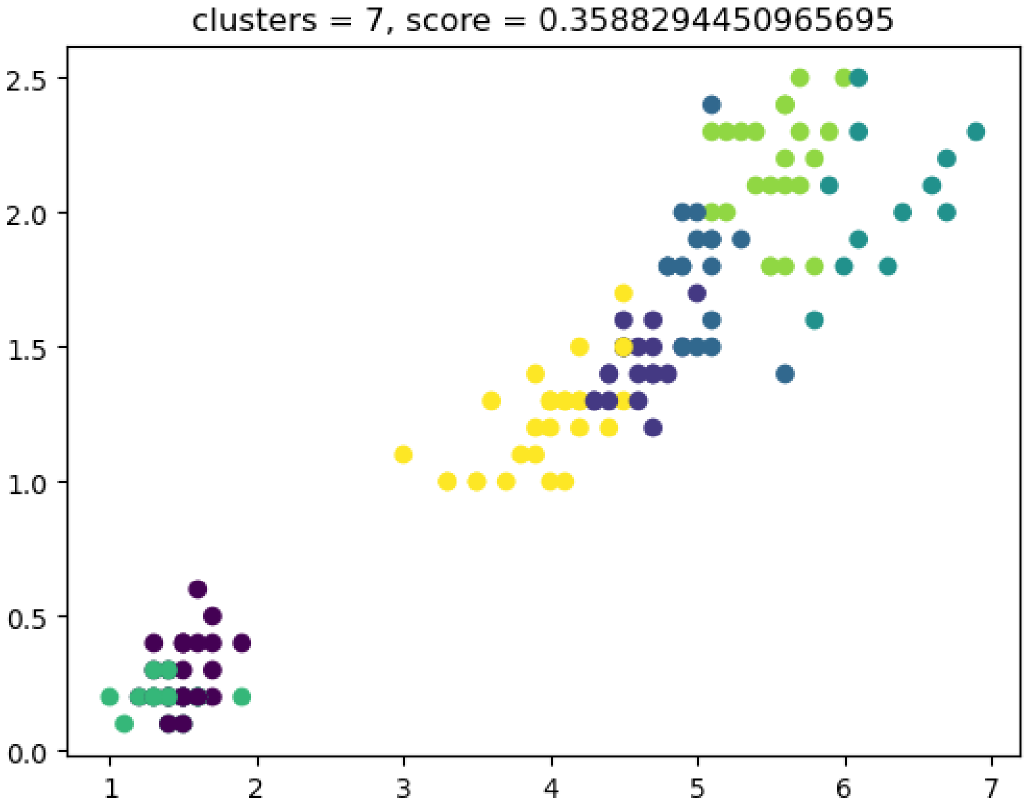

clusters = 2, score = 0.6810461692117467
2.5
2.0
1.5
1.0
0.5
0.0
1
2
3
4
5
6
7

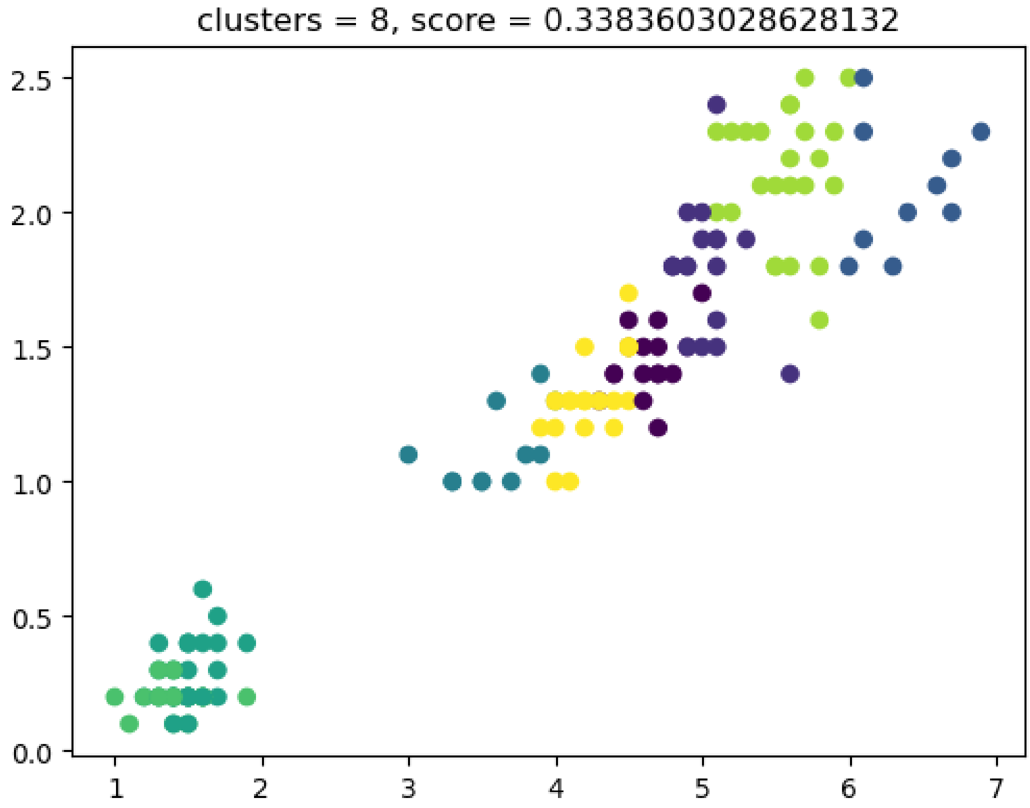

clusters = 3, score = 0.5528190123564101
0.0
0.5
1.0
1.5
2.0
2.5
1
2
3
4
5
6
7

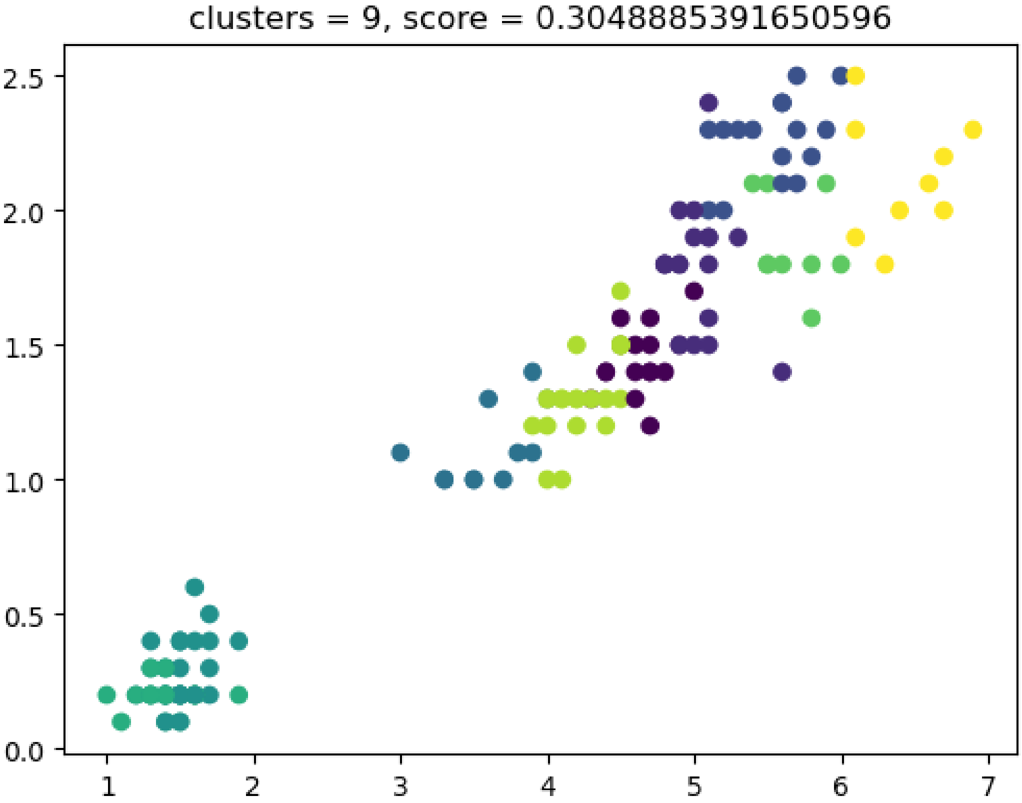

clusters = 4, score = 0.49805050499728803
2.5
2.0
1.5
1.0
0.5
0.0
1
2
3
4
5
6
7

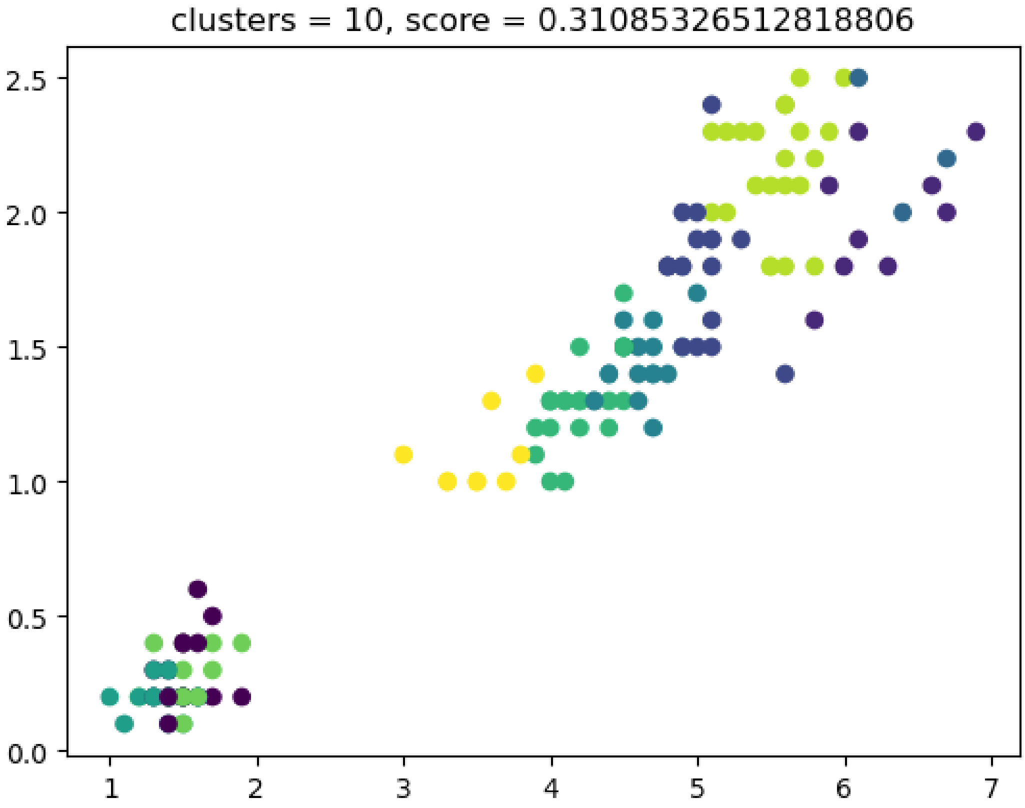

clusters = 5, score = 0.493080406719353
0.0
0.5
1.0
1.5
2.0
2.5
1
2
3
4
5
6
7


clusters = 6, score = 0.3678464984712254
2.5
2.0
1.5
1.0
0.5
0.0
1
2
3
4
5
6
7


clusters = 7, score = 0.3588294450965695
2.5
2.0
1.5
1.0
0.5
0.0
1
2
3
4
5
6
7


clusters = 8, score = 0.3383603028628132
2.5
2.0
1.5
1.0
0.5
0.0
1
2
3
4
5
6
7


clusters = 9, score = 0.3048885391650596
2.5
2.0
1.5
1.0
0.5
0.0
1
2
3
4
5
6
7


clusters = 10, score = 0.31085326512818806
0.0
0.5
1.0
1.5
2.0
2.5
1
2
3
4
5
6
7

```
[ ]:
```