

Chapter 4 - Clustering Models

Part 1 - K - means method

Setting up for clustering analysis

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

import matplotlib.pyplot as plt

import sklearn
from sklearn.preprocessing import scale
import sklearn.metrics as sm
from sklearn import datasets
```

```
In [2]: from sklearn.cluster import KMeans
from mpl_toolkits.mplot3d import Axes3D
from sklearn import datasets
```

```
In [4]: %matplotlib inline
plt.figure(figsize=(7,4))
```

```
Out[4]: <Figure size 504x288 with 0 Axes>

<Figure size 504x288 with 0 Axes>
```

```
In [5]: iris = datasets.load_iris()

x = scale(iris.data)
y = pd.DataFrame(iris.target)
variable_names = iris.feature_names
x[0:10]
```

```
Out[5]: array([[ -0.90068117,  1.01900435, -1.34022653, -1.3154443 ],
 [ -1.14301691, -0.13197948, -1.34022653, -1.3154443 ],
 [ -1.38535265,  0.32841405, -1.39706395, -1.3154443 ],
 [ -1.50652052,  0.09821729, -1.2833891 , -1.3154443 ],
 [ -1.02184904,  1.24920112, -1.34022653, -1.3154443 ],
 [ -0.53717756,  1.93979142, -1.16971425, -1.05217993],
 [ -1.50652052,  0.78880759, -1.34022653, -1.18381211],
 [ -1.02184904,  0.78880759, -1.2833891 , -1.3154443 ],
 [ -1.74885626, -0.36217625, -1.34022653, -1.3154443 ],
 [ -1.14301691,  0.09821729, -1.2833891 , -1.44707648]])
```

Building and running you model

```
In [6]: clustering = KMeans(n_clusters=3, random_state=5)

clustering.fit(x)
```

```
Out[6]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
              n_clusters=3, n_init=10, n_jobs=None, precompute_distances='auto',
              random_state=5, tol=0.0001, verbose=0)
```

Plotting your model outputs

```
In [7]: iris_df = pd.DataFrame(iris.data)
iris_df.columns = ['Sepal_Length', 'Sepal_Width', 'Petal_Length', 'Petal_Width']
y.columns = ['Targets']
```

```
In [8]: color_theme = np.array(['darkgray', 'lightsalmon', 'powderblue'])

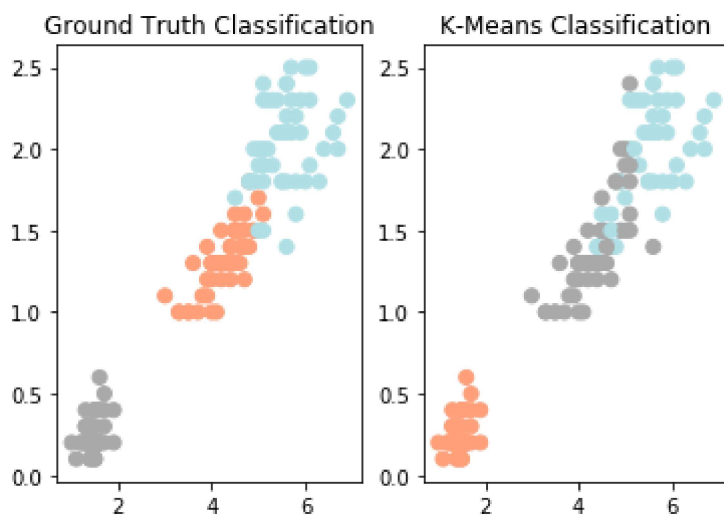
plt.subplot(1,2,1)

plt.scatter(x=iris_df.Petal_Length, y=iris_df.Petal_Width, c=color_theme[iris.targets])
plt.title('Ground Truth Classification')

plt.subplot(1,2,2)

plt.scatter(x=iris_df.Petal_Length, y=iris_df.Petal_Width, c=color_theme[clustering.labels_])
plt.title('K-Means Classification')
```

```
Out[8]: Text(0.5, 1.0, 'K-Means Classification')
```



```
In [11]: relabel = np.choose(clustering.labels_, [2,0,1]).astype(np.int64)

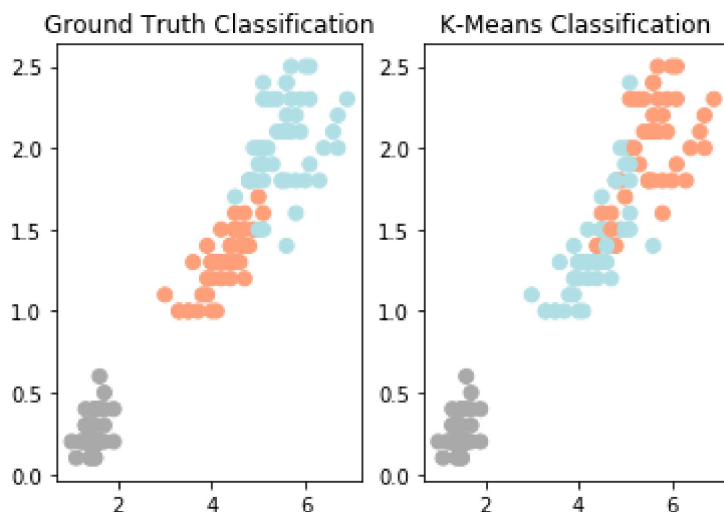
plt.subplot(1,2,1)

plt.scatter(x=iris_df.Petal_Length, y=iris_df.Petal_Width, c=color_theme[iris.target])
plt.title('Ground Truth Classification')

plt.subplot(1,2,2)

plt.scatter(x=iris_df.Petal_Length, y=iris_df.Petal_Width, c=color_theme[relabel])
plt.title('K-Means Classification')
```

Out[11]: Text(0.5, 1.0, 'K-Means Classification')



Evaluate your clustering results

```
In [ ]: print(classification_report(y, relabel))
#It doesn't work this line of code. It said that classification_report is not def
```