

LAB 8

Write a python program to perform K-Means clustering on Breast Cancer Data

```
In [1]: # import all mandatory libraries
from sklearn.datasets import load_breast_cancer
from sklearn.cluster import KMeans
import numpy as np
from scipy.stats import mode
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix
import seaborn as sns
```

```
In [2]: # load dataset
data = load_breast_cancer()
list(data.target_names)
```

Out[2]: ['malignant', 'benign']

```
In [3]: # check data size and target size
print("Data size : ",data.data.shape)
print("Target size : ",data.target.shape)

Data size :  (569, 30)
Target size :  (569,)
```

```
In [4]: # define kMeans and fit data into model
kmeans = KMeans(n_clusters=2,random_state=129)
prediction = kmeans.fit_predict(data.data)
```

```
In [5]: # Cluster shape 2 - class 30 -features
kmeans.cluster_centers_.shape
```

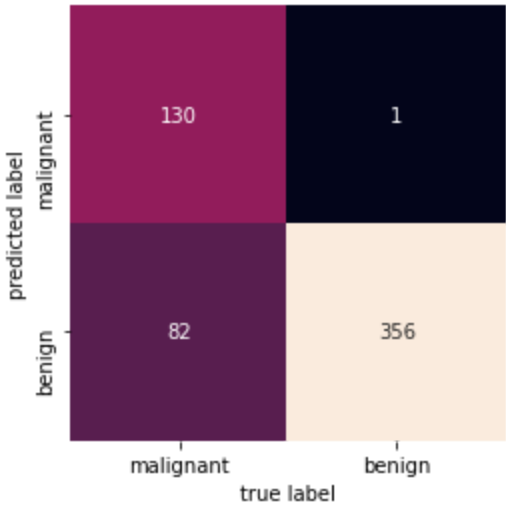
Out[5]: (2, 30)

```
In [6]: # find accuracy score
labels = np.zeros_like(prediction)
for i in range(2):
    mask = (prediction==i)
    labels[mask] = mode(data.target[mask])[0]
accuracy = accuracy_score(data.target,labels)
print("Accuracy is : ",accuracy)

Accuracy is :  0.8541300527240774
```

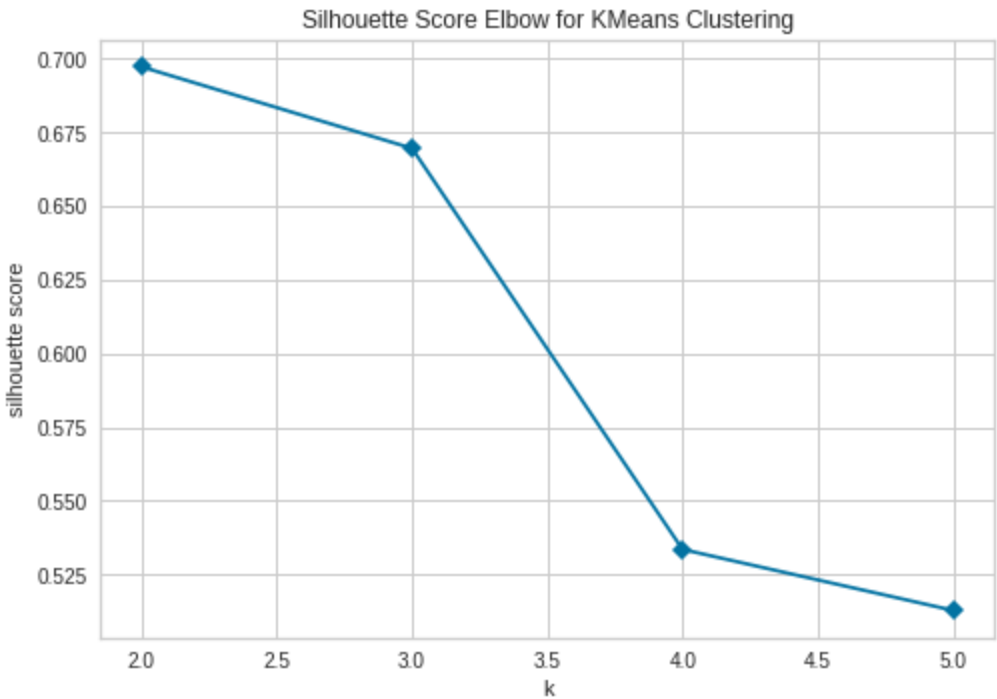
```
In [7]: # create confusion matrix using heatmap
mat = confusion_matrix(data.target, labels)
ax = sns.heatmap(mat.T, square=True, annot=True, fmt='d',cbar=False, xticklabels=data.target_names,
                yticklabels=data.target_names)
ax.set_ylim(2,0,0)
plt.xlabel('true label')
plt.ylabel('predicted label')
```

Out[7]: Text(91.68, 0.5, 'predicted label')



```
In [8]: # Find the best number of cluster for clustering using KElbowVisualizer
from yellowbrick.cluster import KElbowVisualizer
model = KMeans(random_state=0)
visualizer = KElbowVisualizer(model,k=(2,6),metric='silhouette',timings=False)
visualizer.fit(data.data)
visualizer.poof()

/home/nihar/.local/lib/python3.6/site-packages/sklearn/utils/deprecation.py:143: FutureWarning: The sklearn
n.metrics.classification module is deprecated in version 0.22 and will be removed in version 0.24. The cor
responding classes / functions should instead be imported from sklearn.metrics. Anything that cannot be imp
orted from sklearn.metrics is now part of the private API.
  warnings.warn(message, FutureWarning)
/home/nihar/.local/lib/python3.6/site-packages/yellowbrick/utils/kneed.py:182: YellowbrickWarning: No "kne
e" or "elbow point" detected This could be due to bad clustering, no actual clusters being formed etc.
  warnings.warn(warning_message, YellowbrickWarning)
/home/nihar/.local/lib/python3.6/site-packages/yellowbrick/utils/kneed.py:140: YellowbrickWarning: No 'kne
e' or 'elbow point' detected This could be due to bad clustering, no actual clusters being formed etc.
  warnings.warn(warning_message, YellowbrickWarning)
/home/nihar/.local/lib/python3.6/site-packages/yellowbrick/cluster/elbow.py:343: YellowbrickWarning: No 'kn
ee' or 'elbow' point detected, pass `locate_elbow=False` to remove the warning
  warnings.warn(warning_message, YellowbrickWarning)
```



```
Out[8]: <AxesSubplot:title={'center': 'Silhouette Score Elbow for KMeans Clustering'}, xlabel='k', ylabel='silhouett
e score'>
```

```
In [9]: # Check siihouette_score
from sklearn.metrics import silhouette_score
print(silhouette_score(data.data, labels))

0.6972646156059464
```