In [1]:
```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
```

In [2]:
```
url = "https://raw.githubusercontent.com/SharmaNatasha/Machine-Learning-using-Python/master/Datasets/IRIS.csv"
df = pd.read_csv(url)
```

In [ ]:
```
df.head(5)
```

In [4]:
```
x = df.iloc[:,1:4]
y = df.iloc[:,4]
```

In [5]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=0)
```

In [6]:
```
KNN_Classifier = KNeighborsClassifier(n_neighbors = 3, p = 2, metric='minkowski')
```

In [7]:
```
#Train the model

KNN_Classifier.fit(x_train, y_train)

#Let's predict the classes for test data

pred_test = KNN_Classifier.predict(x_test)

#Find the accuracy score

accuracy_score(y_test,pred_test)
```

Out[7]:
```
0.9777777777777777
```

In [8]:
```
KMeans_Cluster = KMeans(n_clusters = 3)
y_class = KMeans_Cluster.fit_predict(x)
```

```
---------------------------------------------------------------------
----
NameError                                 Traceback (most recent call l
ast)
<ipython-input-8-1068023d3ae0> in <module>
----> 1 KMeans_Cluster = KMeans(n_clusters = 3)
      2 y_class = KMeans_Cluster.fit_predict(x)

NameError: name 'KMeans' is not defined
```

In [9]:
```
import numpy as np
import pandas as pd
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

#Load the dataset
url = "https://raw.githubusercontent.com/SharmaNatasha/Machine-Learning
-using-Python/master/Datasets/IRIS.csv"
df = pd.read_csv(url)

#quick look into the data
df.head(5)

#Separate data and label
x = df.iloc[:,1:4].values
```

In [10]:
```
KMeans_Cluster = KMeans(n_clusters = 3)
y_class = KMeans_Cluster.fit_predict(x)
```

In [11]:
```
KMeans_Cluster
```

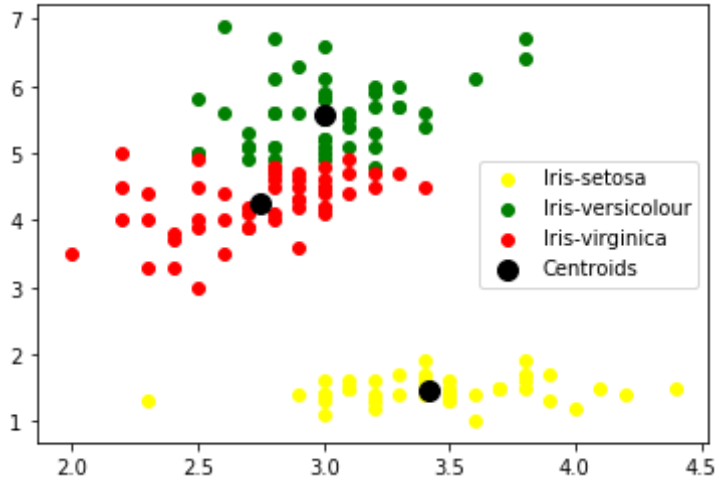
Out[11]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=3, n_init=10, n_jobs=None, precompute_distances='aut
```

```
o',
       random_state=None, tol=0.0001, verbose=0)
```

In [13]:
```
plt.scatter(x[y_class == 0, 0], x[y_class == 0, 1], c = 'yellow', label
 = 'Iris-setosa')
plt.scatter(x[y_class == 1, 0], x[y_class == 1, 1], c = 'green', label
= 'Iris-versicolour')
plt.scatter(x[y_class == 2, 0], x[y_class == 2, 1], c = 'red', label =
'Iris-virginica')
plt.scatter(KMeans_Cluster.cluster_centers_[:, 0], KMeans_Cluster.clust
er_centers_[:,1], s = 100, c = 'black', label = 'Centroids')

plt.legend()
```

Out[13]:
```
<matplotlib.legend.Legend at 0x2017976aa20>
```



In [14]:
```
import math
import numpy as np
import pandas as pd
import matplotlib.pyplot as pyplot
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
vectorizer = TfidfVectorizer()

corpus = [
 'the brown fox jumped over the brown dog',
 'the quick brown fox',
 'the brown brown dog',
 'the fox ate the dog'
]

query = ["brown"]

X = vectorizer.fit_transform(corpus)
Y = vectorizer.transform(query)
```

In [15]:
```
print(vectorizer.get_feature_names())
print(X.toarray())
print(Y)
```

```
['ate', 'brown', 'dog', 'fox', 'jumped', 'over', 'quick', 'the']
[[0.         0.54267123 0.27133562 0.27133562 0.42509967 0.42509967
  0.         0.44366929]
 [0.         0.44181486 0.         0.44181486 0.         0.
  0.69218835 0.3612126 ]
 [0.         0.84003859 0.4200193  0.         0.         0.
  0.         0.3433933 ]
 [0.58680608 0.         0.37455072 0.37455072 0.         0.
  0.         0.61243951]]
  (0, 1)        1.0
```

In [16]:
```
results=[]
results.append(cosine_similarity(Y, X.toarray()))
```

In [17]:
```
results
```

Out[17]:
```
[array([[0.54267123, 0.44181486, 0.84003859, 0.        ]])]
```

In [18]:
```
cosine_similarity(Y, X.toarray())
```

Out[18]: array([[0.54267123, 0.44181486, 0.84003859, 0.   ]])

In [ ]: