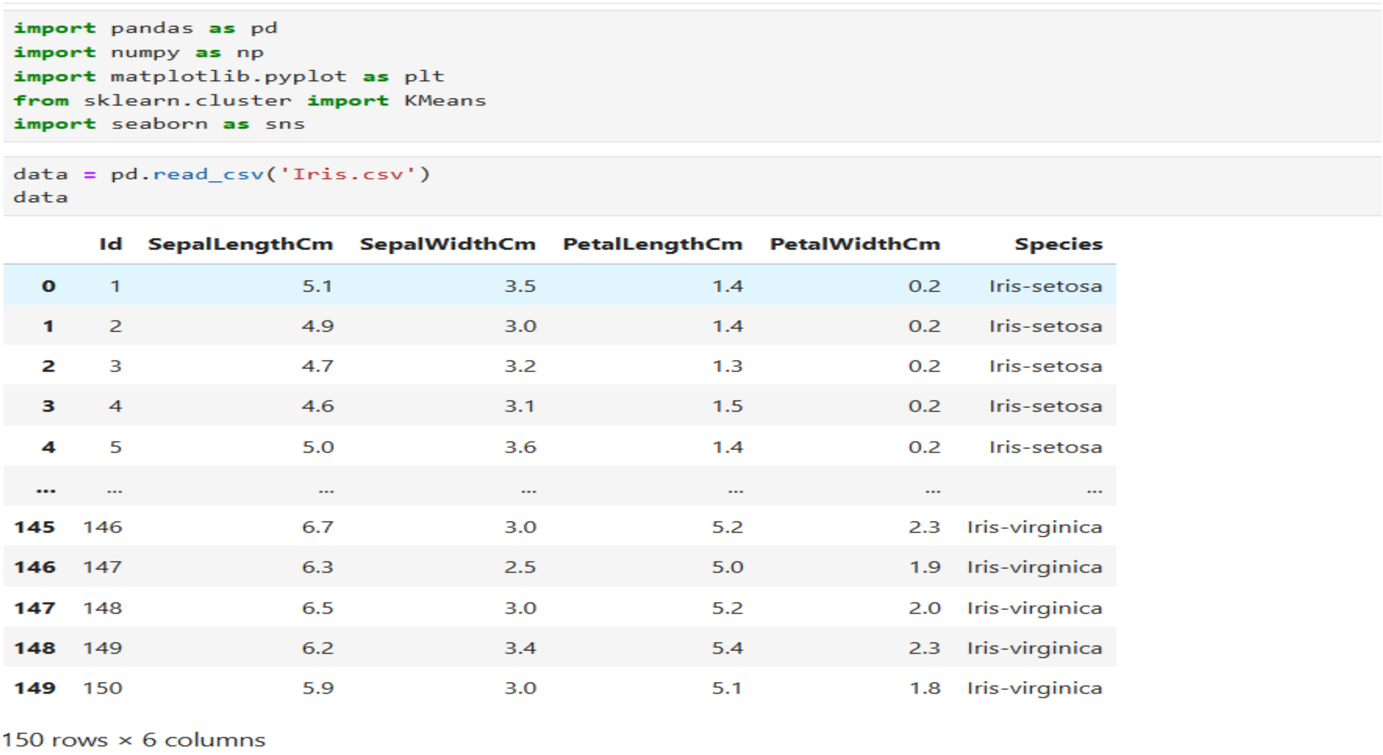
Assignment No.4

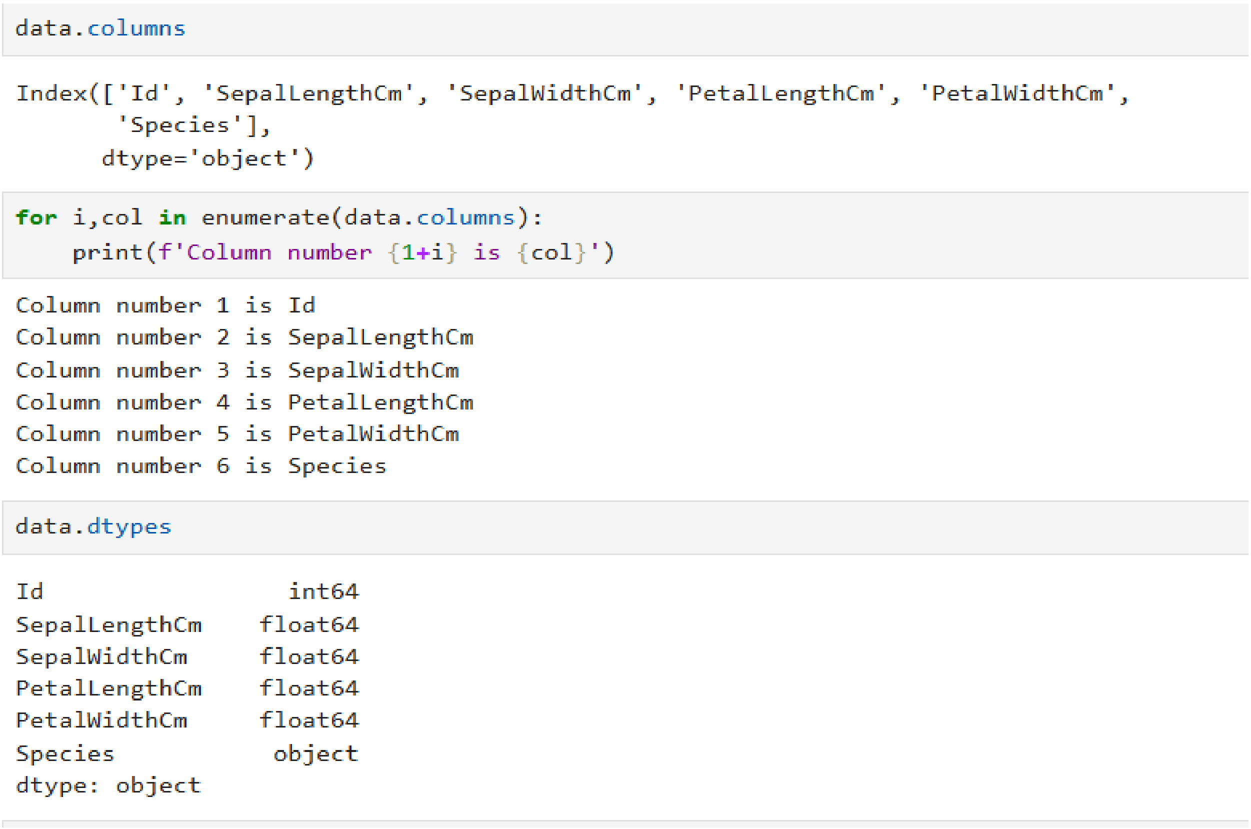


```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
import seaborn as sns
```

```
data = pd.read_csv('Iris.csv')
data
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns



```
data.columns
```

```
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
       'Species'],
      dtype='object')
```

```
for i,col in enumerate(data.columns):
    print(f'Column number {1+i} is {col}')
```

```
Column number 1 is Id
Column number 2 is SepalLengthCm
Column number 3 is SepalWidthCm
Column number 4 is PetalLengthCm
Column number 5 is PetalWidthCm
Column number 6 is Species
```

```
data.dtypes
```

```
Id                 int64
SepalLengthCm    float64
SepalWidthCm     float64
PetalLengthCm    float64
PetalWidthCm     float64
Species           object
dtype: object
```

```
data.drop('Id', axis=1, inplace=True)
data.head()
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

```
data.isna().sum()
```

```
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

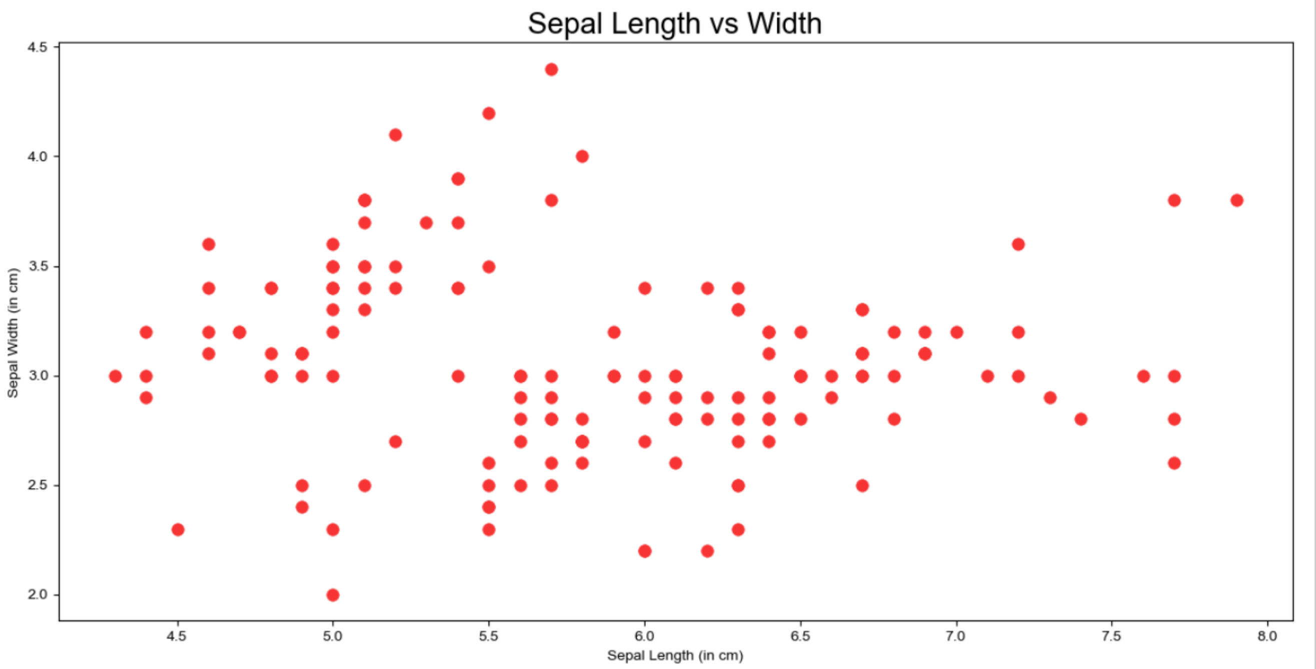
```
target_data = data.iloc[:,4]
target_data.unique()
```

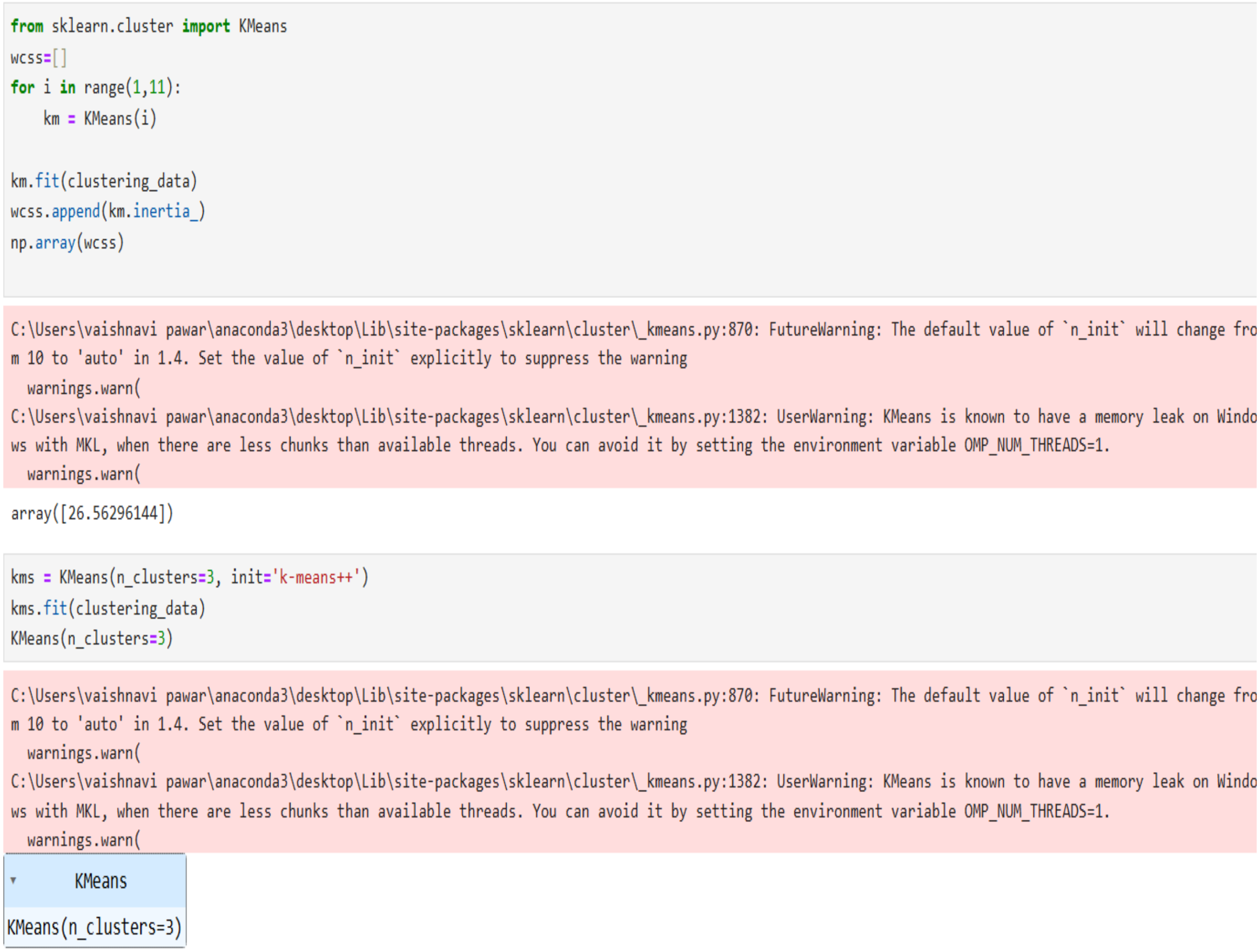
```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

```
clustering_data = data.iloc[:,[0,1,2,3]]
clustering_data.head()
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
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

```
fig, ax = plt.subplots(figsize=(15,7))
sns.set(font_scale=1.5)
ax = sns.scatterplot(x=data['SepalLengthCm'],y=data['SepalWidthCm'], s=70, color='#f73434',
edgecolor='#f73434', linewidth=0.3)
ax.set_ylabel('Sepal Width (in cm)')
ax.set_xlabel('Sepal Length (in cm)')
plt.title('Sepal Length vs Width', fontsize = 20)
plt.show()
```





```
from sklearn.cluster import KMeans
wcss=[]
for i in range(1,11):
    km = KMeans(i)

km.fit(clustering_data)
wcss.append(km.inertia_)
np.array(wcss)
```

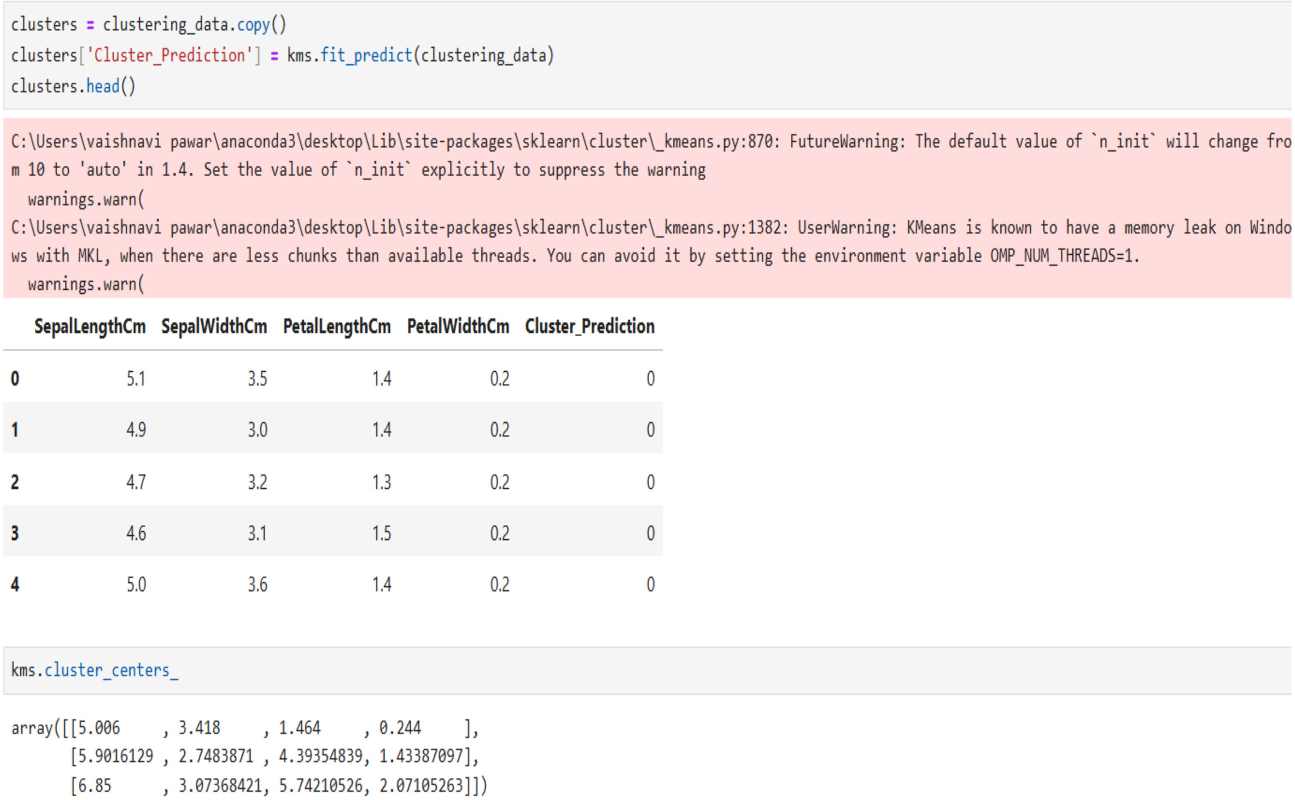
```
C:\Users\vaishnavi pawar\anaconda3\desktop\Lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will change fro
m 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\vaishnavi pawar\anaconda3\desktop\Lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWarning: KMeans is known to have a memory leak on Windo
ws with MKL, when there are less chunks than available threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

```
array([26.56296144])
```

```
kms = KMeans(n_clusters=3, init='k-means++')
kms.fit(clustering_data)
KMeans(n_clusters=3)
```

```
C:\Users\vaishnavi pawar\anaconda3\desktop\Lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will change fro
m 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\vaishnavi pawar\anaconda3\desktop\Lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWarning: KMeans is known to have a memory leak on Windo
ws with MKL, when there are less chunks than available threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

```
KMeans(n_clusters=3)
```



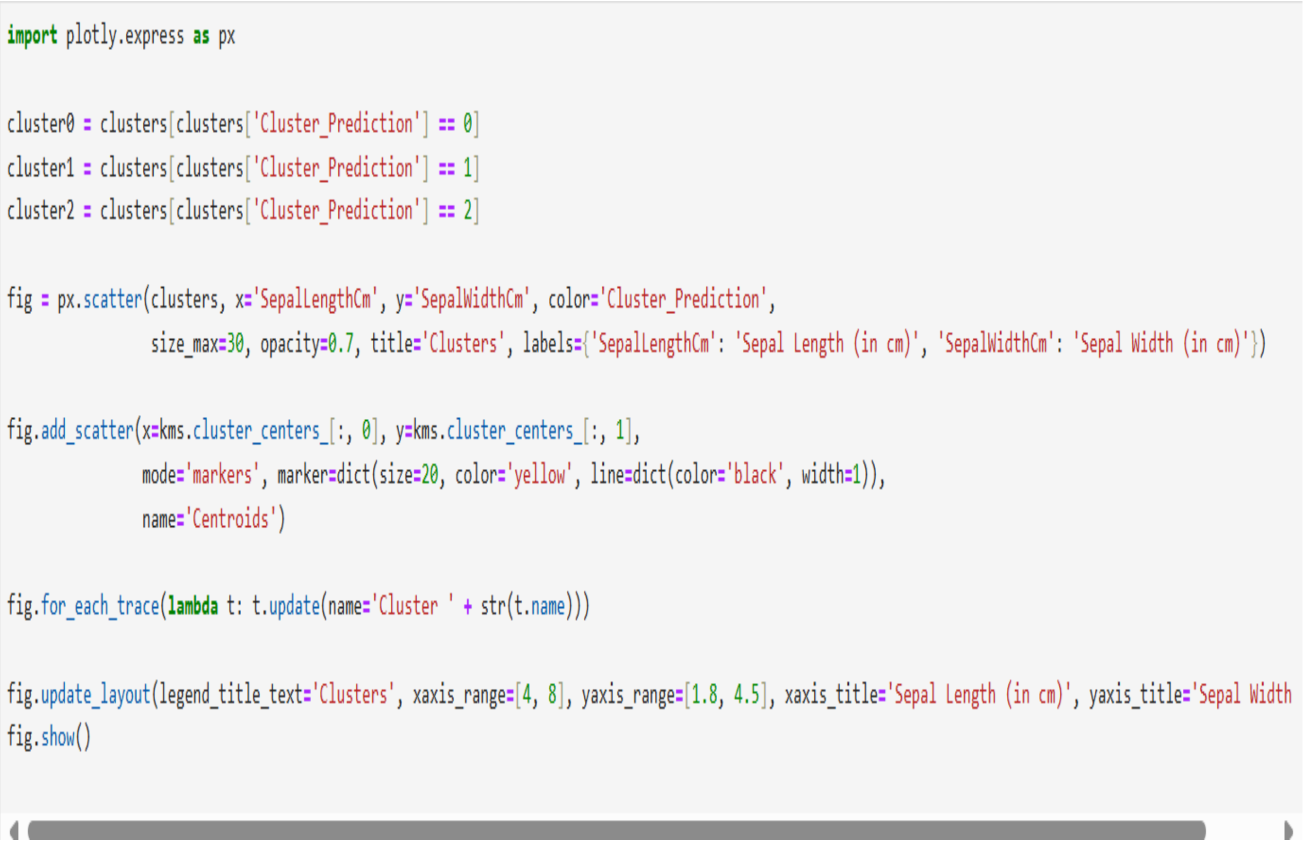
```
clusters = clustering_data.copy()
clusters['Cluster_Prediction'] = kms.fit_predict(clustering_data)
clusters.head()
```

```
C:\Users\vaishnavi pawar\anaconda3\desktop\Lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will change fro
m 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\vaishnavi pawar\anaconda3\desktop\Lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWarning: KMeans is known to have a memory leak on Windo
ws with MKL, when there are less chunks than available threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

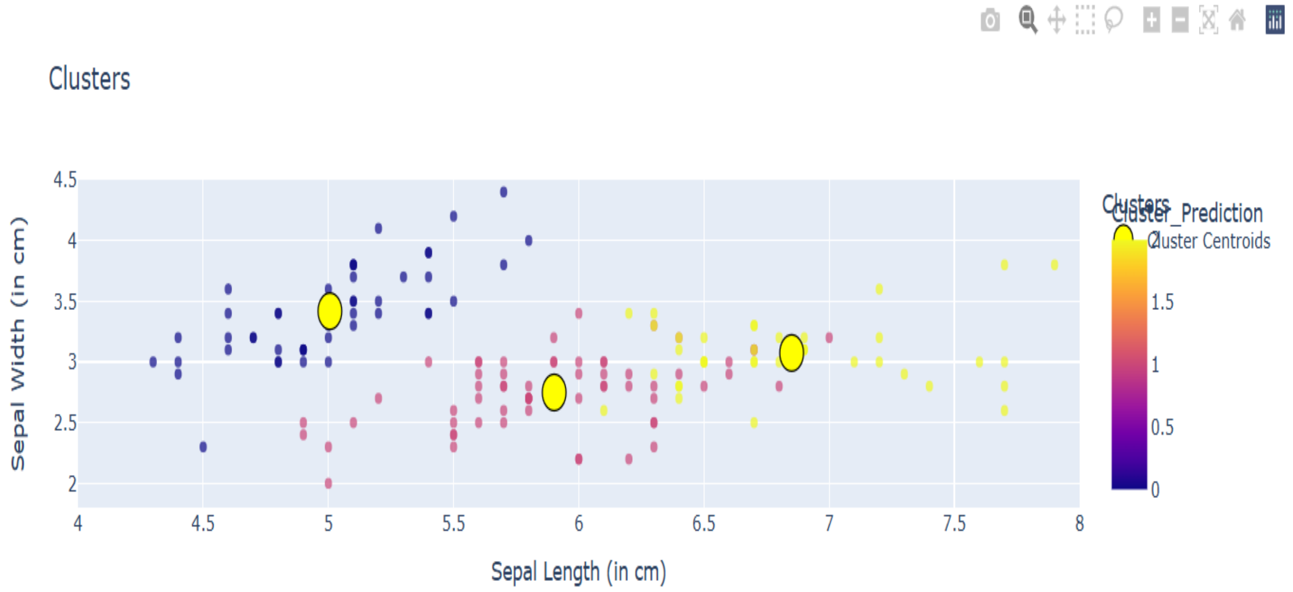
	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Cluster_Prediction
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

```
kms.cluster_centers_
```

```
array([[5.006     , 3.418     , 1.464     , 0.244     ],
       [5.9016129 , 2.7483871 , 4.39354839, 1.43387097],
       [6.85      , 3.07368421, 5.74210526, 2.07105263]])
```



```
import plotly.express as px

cluster0 = clusters[clusters['Cluster_Prediction'] == 0]
cluster1 = clusters[clusters['Cluster_Prediction'] == 1]
cluster2 = clusters[clusters['Cluster_Prediction'] == 2]

fig = px.scatter(clusters, x='SepalLengthCm', y='SepalWidthCm', color='Cluster_Prediction',
                 size_max=30, opacity=0.7, title='Clusters', labels={'SepalLengthCm': 'Sepal Length (in cm)', 'SepalWidthCm': 'Sepal Width (in cm)'})

fig.add_scatter(x=kms.cluster_centers_[:, 0], y=kms.cluster_centers_[:, 1],
                mode='markers', marker=dict(size=20, color='yellow', line=dict(color='black', width=1)),
                name='Centroids')

fig.for_each_trace(lambda t: t.update(name='Cluster ' + str(t.name)))

fig.update_layout(legend_title_text='Clusters', xaxis_range=[4, 8], yaxis_range=[1.8, 4.5], xaxis_title='Sepal Length (in cm)', yaxis_title='Sepal Width
fig.show()
```


Clusters
Clusters
Cluster_Prediction
Cluster Centroids
2
1.5
1
0.5
0
Sepal Width (in cm)
4.5
4
3.5
3
2.5
2
4
4.5
5
5.5
6
6.5
7
7.5
8
Sepal Length (in cm)