K-means Clustring

```python
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import seaborn as sns

file_path = '/content/Iris.csv'
df = pd.read_csv(file_path)

print(df.head())
```

```
   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
```

```python
data = df.drop(columns=['Id', 'Species'])

scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)

kmeans = KMeans(n_clusters=3, random_state=42)
kmeans_labels = kmeans.fit_predict(data_scaled)

df['KMeans_Labels'] = kmeans_labels

plt.figure(figsize=(8, 6))
sns.scatterplot(x='SepalLengthCm', y='SepalWidthCm',
hue='KMeans_Labels', palette='viridis', data=df)
plt.title('K-means Clustering')
plt.show()
```
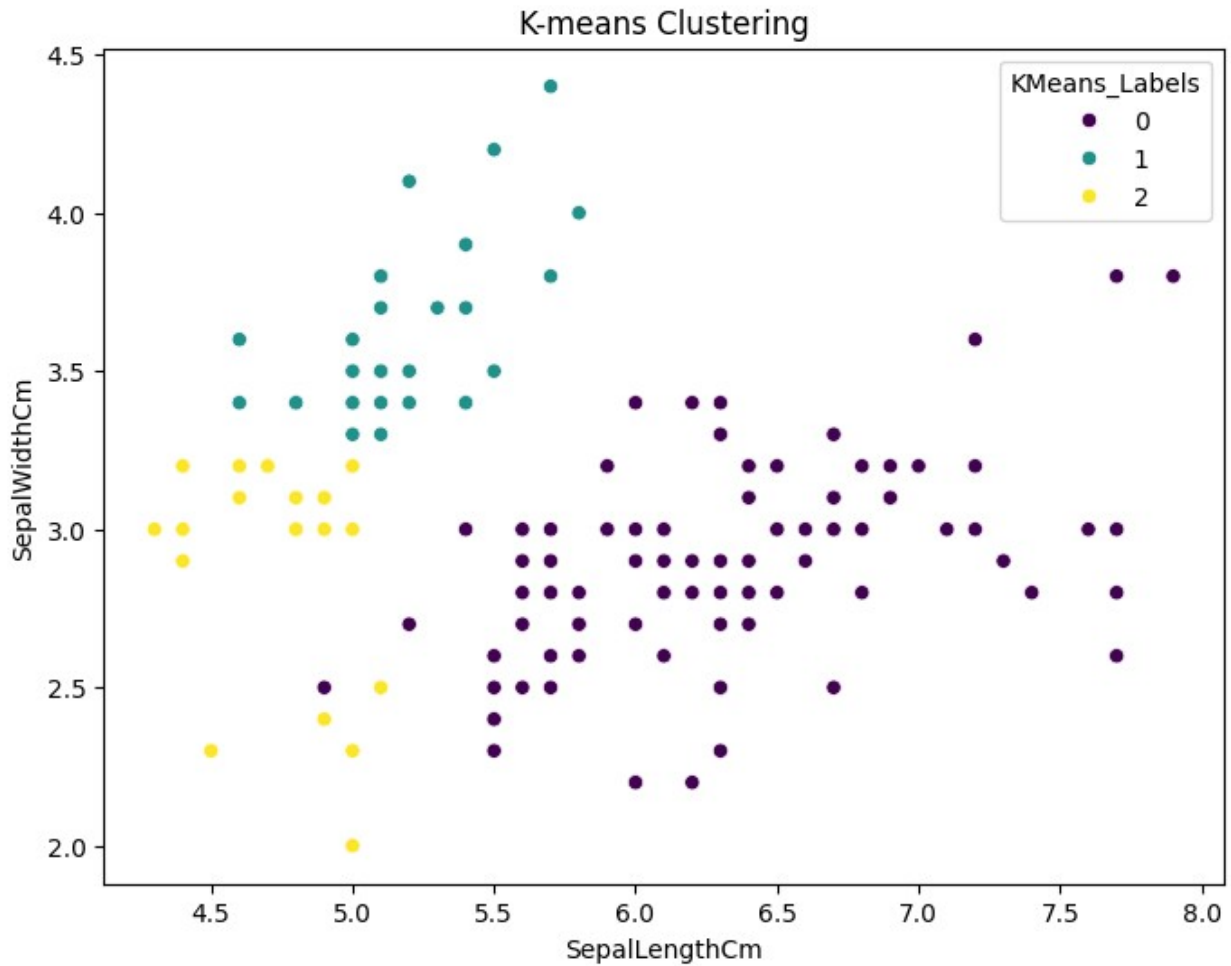
K-means Clustering

DBSCAN Clustering

```python
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import DBSCAN
from sklearn.preprocessing import StandardScaler
import seaborn as sns

file_path = '/content/Iris.csv'
df = pd.read_csv(file_path)

print(df.head())

   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

data = df.drop(columns=['Id', 'Species'])

scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)

dbscan = DBSCAN(eps=0.5, min_samples=5)
dbscan_labels = dbscan.fit_predict(data_scaled)

df['DBSCAN_Labels'] = dbscan_labels

plt.figure(figsize=(8, 6))
sns.scatterplot(x='SepalLengthCm', y='SepalWidthCm',
hue='DBSCAN_Labels', palette='viridis', data=df)
plt.title('DBSCAN Clustering')
plt.show()
```