Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. You can add Java/Python ML library classes/API in the program.

```
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.mixture import GaussianMixture
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

# Load dataset
data = pd.read_csv('/content/Mall_Customers.csv')

# Selecting features for clustering
features = ['Annual Income (k$)', 'Spending Score (1-100)']
X = data[features]

# Standardize the features
scaler = StandardScaler()
scaled_data = scaler.fit_transform(X)

# Apply EM algorithm
em_model = GaussianMixture(n_components=5, random_state=42)
em_clusters = em_model.fit_predict(scaled_data)

# Apply k-means algorithm
kmeans_model = KMeans(n_clusters=5, random_state=42)
kmeans_clusters = kmeans_model.fit_predict(scaled_data)

# Plot the results
plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
plt.scatter(X.iloc[:, 0], X.iloc[:, 1], c=em_clusters, cmap='viridis')
plt.title('EM Clustering')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')

plt.subplot(1, 2, 2)
plt.scatter(X.iloc[:, 0], X.iloc[:, 1], c=kmeans_clusters,
cmap='viridis')
plt.title('K-Means Clustering')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
```

```
plt.tight_layout()
plt.show()
```