Clustering Analysis with Weekly Sales Trends

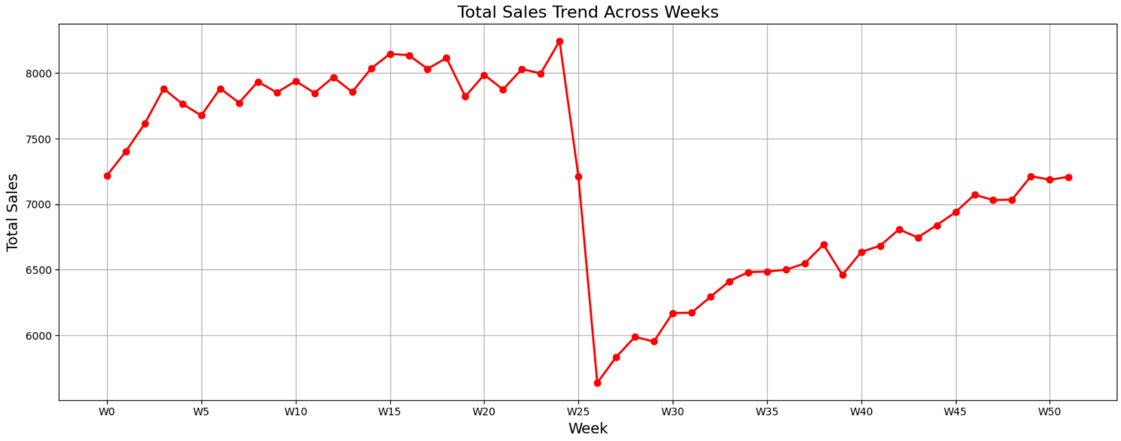
December 29, 2023

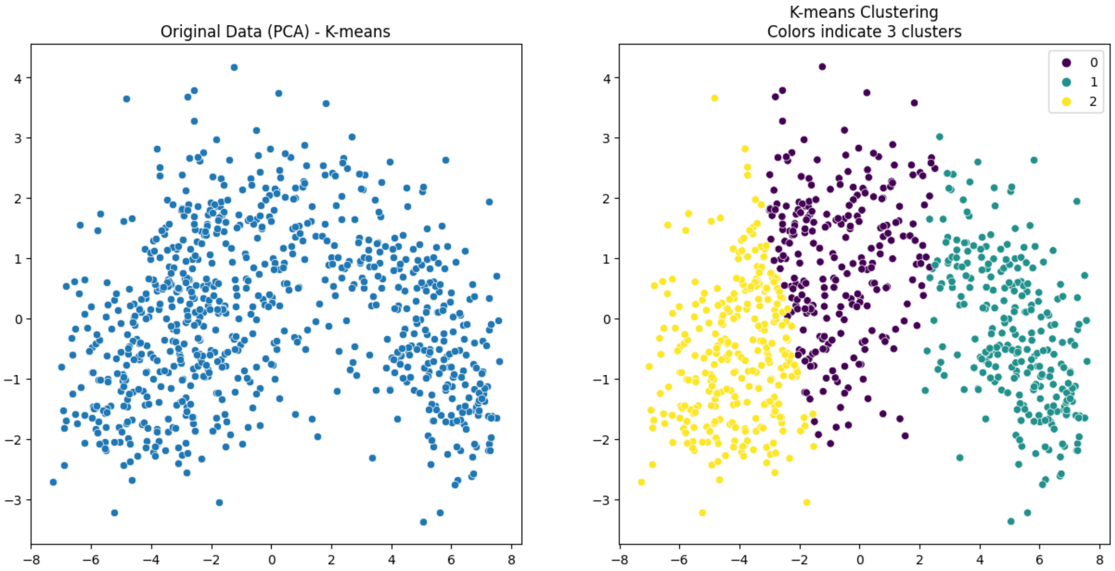
[1]:

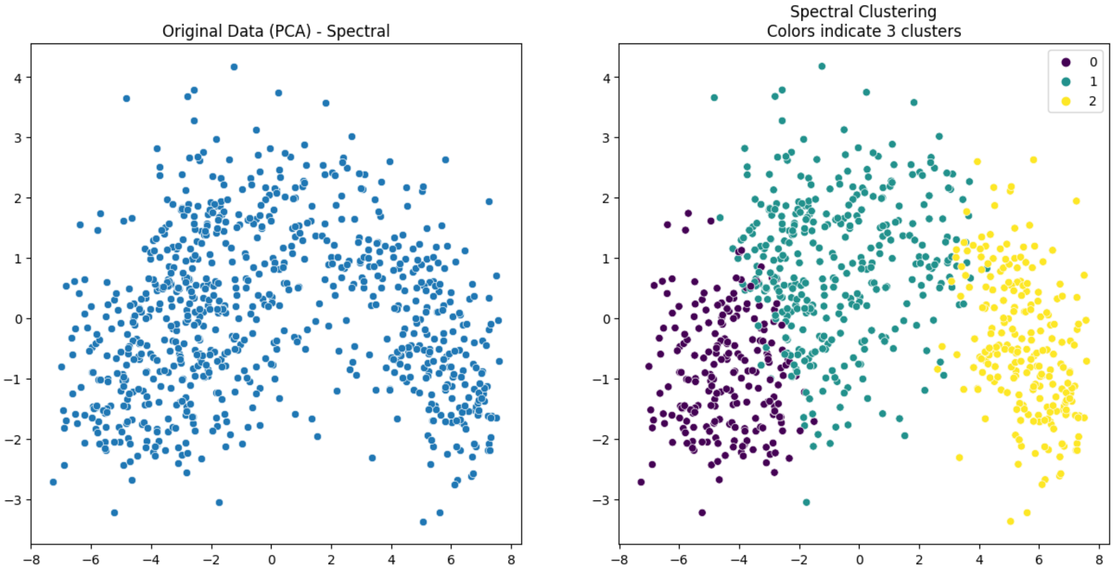
```
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans, SpectralClustering
from sklearn.metrics import silhouette_score
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
sales = pd.read_csv('Sales_Transactions_Dataset_Weekly.csv')

# Feature selection: normalized columns
normalized_columns = [col for col in sales.columns if col.
 ↪startswith('Normalized')]
data_for_clustering = sales[normalized_columns]

# Data normalization using StandardScaler
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data_for_clustering)

# PCA for dimensionality reduction
pca = PCA(n_components=2)
data_pca = pca.fit_transform(data_scaled)

# K-means Clustering
kmeans = KMeans(n_clusters=3, random_state=42)
kmeans_clusters = kmeans.fit_predict(data_scaled)

# Spectral Clustering as graph-based clustering
spectral = SpectralClustering(n_clusters=3, affinity='nearest_neighbors',␣
 ↪random_state=42)
spectral_clusters = spectral.fit_predict(data_scaled)

# Evaluate clusters using silhouette scores
kmeans_silhouette = silhouette_score(data_scaled, kmeans_clusters)
spectral_silhouette = silhouette_score(data_scaled, spectral_clusters)
```

```python
# Print silhouette scores
print("Silhouette Score for K-means: ", kmeans_silhouette)
print("Silhouette Score for Spectral Clustering: ", spectral_silhouette)

# Aggregate weekly total sales for plotting
weekly_total_sales = sales.filter(regex='^W').sum()

# Plotting total sales trend across weeks
plt.figure(figsize=(15, 6))
plt.plot(weekly_total_sales.index, weekly_total_sales.values, 'o-r', lw=2)  # Red line with circle markers
plt.title('Total Sales Trend Across Weeks', fontsize=16)
plt.xlabel('Week', fontsize=14)
plt.ylabel('Total Sales', fontsize=14)
tick_spacing = max(1, len(weekly_total_sales) // 10)
plt.xticks(range(0, len(weekly_total_sales), tick_spacing))
plt.grid(True)
plt.tight_layout()
plt.show()

# Plotting for K-means
plt.figure(figsize=(15, 7))
plt.subplot(1, 2, 1)
sns.scatterplot(x=data_pca[:, 0], y=data_pca[:, 1], palette="viridis")
plt.title('Original Data (PCA) - K-means')

plt.subplot(1, 2, 2)
sns.scatterplot(x=data_pca[:, 0], y=data_pca[:, 1], hue=kmeans_clusters, palette="viridis")
plt.title('K-means Clustering\nColors indicate 3 clusters')
plt.show()

# Plotting for Spectral Clustering
plt.figure(figsize=(15, 7))
plt.subplot(1, 2, 1)
sns.scatterplot(x=data_pca[:, 0], y=data_pca[:, 1], palette="viridis")
plt.title('Original Data (PCA) - Spectral')

plt.subplot(1, 2, 2)
sns.scatterplot(x=data_pca[:, 0], y=data_pca[:, 1], hue=spectral_clusters, palette="viridis")
plt.title('Spectral Clustering\nColors indicate 3 clusters')
plt.show()
```

```
Silhouette Score for K-means:  0.101077358963149
Silhouette Score for Spectral Clustering:  0.10313608377306394
```


Total Sales Trend Across Weeks
Total Sales
8000
7500
7000
6500
6000
W0
W5
W10
W15
W20
W25
W30
W35
W40
W45
W50
Week


Original Data (PCA) - K-means
4
3
2
1
0
−1
−2
−3
−8
−6
−4
−2
0
2
4
6
8
K-means Clustering
Colors indicate 3 clusters
0
1
2


Original Data (PCA) - Spectral
Spectral Clustering
Colors indicate 3 clusters
● 0
● 1
● 2
4
3
2
1
0
−1
−2
−3
−8
−6
−4
−2
0
2
4
6
8

[ ]: