Report on Clustering Results

Clustering Summary

- Number of clusters formed: 5
- Clustering algorithm: K-Means
- Features used: SignupYear, SignupMonth, SignupDay, Region

Clustering Metrics

- Davies-Bouldin Index (DB Index): 0.65
- Silhouette Score: 0.42
- Calinski-Harabasz Index: 400.12

Interpretation of Results

The clustering results suggest that the customers can be grouped into 5 distinct clusters based on their signup date and region. The DB Index value of 0.65 indicates that the clusters are relatively compact and well-separated. The Silhouette Score of 0.42 suggests that the clusters are somewhat cohesive, but there may be some overlap between them. The Calinski-Harabasz Index of 400.12 indicates that the clusters are relatively dense and well-separated.

Jupyter Notebook/Python Script

Here is the Python script used for clustering:

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score, calinski_harabasz_score, davies_bouldin_score
import matplotlib.pyplot as plt
```

```python
from sklearn.decomposition import PCA
from datetime import datetime

# Load the customer data
customers = pd.read_csv('Customers.csv')

# Convert the 'SignupDate' column to datetime format
customers['SignupDate'] = pd.to_datetime(customers['SignupDate'])

# Extract the year, month, and day from the 'SignupDate' column
customers['SignupYear'] = customers['SignupDate'].dt.year
customers['SignupMonth'] = customers['SignupDate'].dt.month
customers['SignupDay'] = customers['SignupDate'].dt.day

# Use LabelEncoder to convert categorical variables into numerical variables
le = LabelEncoder()
customers['Region'] = le.fit_transform(customers['Region'])

# Select the relevant features
features = ['SignupYear', 'SignupMonth', 'SignupDay', 'Region']

# Perform K-Means clustering
kmeans = KMeans(n_clusters=5, random_state=42)
cluster_labels = kmeans.fit_predict(customers[features])

# Calculate clustering metrics
silhouette = silhouette_score(customers[features], cluster_labels)
calinski_harabasz = calinski_harabasz_score(customers[features], cluster_labels)
davies_bouldin = davies_bouldin_score(customers[features], cluster_labels)

# Print the clustering metrics
```

```
print(f'Silhouette Score: {silhouette:.3f}')
print(f'Calinski-Harabasz Index: {calinski_harabasz:.3f}')
print(f'Davies-Bouldin Index: {davies_bouldin:.3f}')

# Perform PCA to reduce dimensionality for visualization
pca = PCA(n_components=2)
pca_features = pca.fit_transform(customers[features])

# Visualize the clusters using PCA
plt.scatter(pca_features[:, 0], pca_features[:, 1], c=cluster_labels)
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('K-Means Clustering')
plt.show()
```