

CCA-6 : Utilizing Clustering Algorithms to Segment Customers

Overview

Learning Objectives

Steps:

Step 1: Importing necessary Libraries

Step 2 : Set up paths and ensure utility modules are accessible

Step 3: Loading Configuration and Set up paths

Step 4: loading the pre-processed datasets

Step 5 : Apply K-Means Algorithms and visualize the clusters(assumed 3 clusters)

Step 6: Applying K-Means Algorithmns

Results

Conclusion:

Next Steps:

Overview

In this notebook, we will load the pre-processed dataset obtained by data processing member. After loading the pre-processed which includes both min-max scaled and standard scaled dataset, we will be using K-Means algorithms to shows the process of k-means algorithms. For this process we will assumed the number of cluster to be 3 on both min-max scaled and standard scaled dataset. We will be using features monthly charges and tenure for this process to make our work more realistic.

Learning Objectives

The main learning objectives of this task is that we will learn how to apply K-Means algorithms using the specific features or characteristics to segment the customers. Besides that, we will visualize the outcomes of the K-Means algorithms using number of optimal cluster 3 on both scaled dataset.

Note: We are just assuming the number of cluster to be 3. We will find the optimal number of cluster in next step.

Steps:

Step 1: Importing necessary Libraries

In this step we will be Importing the necessary libraries that will be used in our K-Means algorithms and to visualize the cluster characteristics.

```
1 import os
2 import sys
3 import json
4 import pandas as pd
5 from sklearn.cluster import KMeans
6 import matplotlib.pyplot as plt
7 import seaborn as sns from sklearn.metrics
8 import silhouette_score
9
```

Step 2 : Set up paths and ensure utility modules are accessible

So, basically in this process we just set up the paths and make sure that the required utility modules are easily accessible.

```
1 # Ensure the utils module can be found
2 Notebook_dir = os.path.dirname(os.path.abspath(''))
3 Project_root = os.path.abspath(os.path.join(notebook_dir, '..'))
4 Utils_path = os.path.join(project_root, 'utils')
5
6 If utils_path not in sys.path:
7     sys.path.append(utils_path)
8
9 from data_loader import load_data
10 from data_cleaner import clean_data
11
12
```

Step 3: Loading Configuration and Set up paths

In this process we will configure the configuration from config.json and we will convert the relative path into absolute path to make sure we will be using the correct dataset path for our project.

```
1 # Load configuration
2 config_path = os.path.join(os.path.dirname(os.path.abspath('')), '..', 'config.json')
3 with open(config_path, 'r') as f:
4     config = json.load(f)
5
6 # Set up paths
7 project_root = os.path.dirname(os.path.dirname(os.path.abspath('')))
8 raw_data_path = os.path.join(project_root, config['raw_data_path'])
9
10
```

Step 4: loading the pre-processed datasets

In this process we load the required pre-processed datasets which includes both min-max and scaled dataset for clustering analysis.

```
1 # Load datasets
2 df_min_max_scaled = pd.read_csv(min_max_scaled_data_path)
3 df_standard_scaled = pd.read_csv(standard_scaled_data_path)
4
```

Step 5 : Apply K-Means Algorithms and visualize the clusters(assumed 3 clusters)

In this process we will apply k-means algorithm's by assuming number of clusters which is 3 and visualize the clusters.

```
1 def apply_kmeans_and_visualize(df, scaling_label, n_clusters):
2     features = df[['tenure', 'MonthlyCharges']]
3     kmeans = KMeans(n_clusters=n_clusters, random_state=42)
4     kmeans.fit(features)
5     df['Cluster'] = kmeans.labels_
6
```

```
7 plt.figure(figsize=(10, 6))
8 sns.scatterplot(data=df, x='tenure', y='MonthlyCharges', hue='Cluster', palette='viridis')
9 plt.title(f'Customer Segments ({scaling_label})')
10 plt.show()
11
```

Step 6: Applying K-Means Algorithms

```
1 n_clusters = 3
2 apply_kmeans_and_visualize(df_min_max_scaled, 'Min-peMax Scaled', n_clusters)
3 apply_kmeans_and_visualize(df_standard_scaled, 'Standard Scaled', n_clusters)
```

Results

From the K-Means algorithms we got different results. For both scaled datasets we got separate visualizations of customer cluster and each clusters represent their own basic features and characteristics based on their tenure and monthly charges.

Conclusion:

In conclusion we can say that in this process we selected the features of the datasets as tenure and monthly charges and using that feature we applied the K-Means algorithms to segments the customers. As a result, we visualize the clusters characteristics based on tenure and monthly charges. So the main aim of this process is to show the method of applying K-means and visualize the clusters. These outcomes can be used by the business organization to identified the characterises of the customers and make plan and strategy accordingly.

Next Steps:

So, the next steps includes:

- Determining the optimal number of clusters
- And train the K-Means Algorithms using the determined optimal number of clusters