hierarchical-clustering

December 14, 2023

[29]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly as py
from scipy.spatial import distance
from scipy.cluster.hierarchy import linkage, dendrogram, fcluster
from sklearn.cluster import AgglomerativeClustering
```

[30]:
```
df = pd.read_csv('/kaggle/input/ccdata/CC GENERAL.csv')
df.head()
```

[30]:

	CUST_ID	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES
0	C10001	40.900749	0.818182	95.40	0.00
1	C10002	3202.467416	0.909091	0.00	0.00
2	C10003	2495.148862	1.000000	773.17	773.17
3	C10004	1666.670542	0.636364	1499.00	1499.00
4	C10005	817.714335	1.000000	16.00	16.00

	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHASES_FREQUENCY
0	95.4	0.000000	0.166667
1	0.0	6442.945483	0.000000
2	0.0	0.000000	1.000000
3	0.0	205.788017	0.083333
4	0.0	0.000000	0.083333

	ONEOFF_PURCHASES_FREQUENCY	PURCHASES_INSTALLMENTS_FREQUENCY
0	0.000000	0.083333
1	0.000000	0.000000
2	1.000000	0.000000
3	0.083333	0.000000
4	0.083333	0.000000

	CASH_ADVANCE_FREQUENCY	CASH_ADVANCE_TRX	PURCHASES_TRX	CREDIT_LIMIT
0	0.000000	0	2	1000.0
1	0.250000	4	0	7000.0
2	0.000000	0	12	7500.0

```
3                0.083333                    1              1        7500.0
4                0.000000                    0              1        1200.0

      PAYMENTS  MINIMUM_PAYMENTS  PRC_FULL_PAYMENT  TENURE
0   201.802084        139.509787          0.000000      12
1  4103.032597       1072.340217          0.222222      12
2   622.066742        627.284787          0.000000      12
3     0.000000               NaN          0.000000      12
4   678.334763        244.791237          0.000000      12
```

[31]:
```
df.shape
```

[31]:
```
(8950, 18)
```

[32]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8950 entries, 0 to 8949
Data columns (total 18 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   CUST_ID                           8950 non-null   object
 1   BALANCE                           8950 non-null   float64
 2   BALANCE_FREQUENCY                 8950 non-null   float64
 3   PURCHASES                         8950 non-null   float64
 4   ONEOFF_PURCHASES                  8950 non-null   float64
 5   INSTALLMENTS_PURCHASES            8950 non-null   float64
 6   CASH_ADVANCE                      8950 non-null   float64
 7   PURCHASES_FREQUENCY               8950 non-null   float64
 8   ONEOFF_PURCHASES_FREQUENCY        8950 non-null   float64
 9   PURCHASES_INSTALLMENTS_FREQUENCY  8950 non-null   float64
 10  CASH_ADVANCE_FREQUENCY            8950 non-null   float64
 11  CASH_ADVANCE_TRX                  8950 non-null   int64
 12  PURCHASES_TRX                     8950 non-null   int64
 13  CREDIT_LIMIT                      8949 non-null   float64
 14  PAYMENTS                          8950 non-null   float64
 15  MINIMUM_PAYMENTS                  8637 non-null   float64
 16  PRC_FULL_PAYMENT                  8950 non-null   float64
 17  TENURE                            8950 non-null   int64
dtypes: float64(14), int64(3), object(1)
memory usage: 1.2+ MB
```

[33]:
```
df.describe()
```

[33]:
```
             BALANCE  BALANCE_FREQUENCY     PURCHASES  ONEOFF_PURCHASES  \
count    8950.000000        8950.000000   8950.000000       8950.000000
mean     1564.474828           0.877271   1003.204834        592.437371
```

std	2081.531879	0.236904	2136.634782	1659.887917
min	0.000000	0.000000	0.000000	0.000000
25%	128.281915	0.888889	39.635000	0.000000
50%	873.385231	1.000000	361.280000	38.000000
75%	2054.140036	1.000000	1110.130000	577.405000
max	19043.138560	1.000000	49039.570000	40761.250000

	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHASES_FREQUENCY	\
count	8950.000000	8950.000000	8950.000000	
mean	411.067645	978.871112	0.490351	
std	904.338115	2097.163877	0.401371	
min	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.083333	
50%	89.000000	0.000000	0.500000	
75%	468.637500	1113.821139	0.916667	
max	22500.000000	47137.211760	1.000000	

	ONEOFF_PURCHASES_FREQUENCY	PURCHASES_INSTALLMENTS_FREQUENCY	\
count	8950.000000	8950.000000	
mean	0.202458	0.364437	
std	0.298336	0.397448	
min	0.000000	0.000000	
25%	0.000000	0.000000	
50%	0.083333	0.166667	
75%	0.300000	0.750000	
max	1.000000	1.000000	

	CASH_ADVANCE_FREQUENCY	CASH_ADVANCE_TRX	PURCHASES_TRX	CREDIT_LIMIT	\
count	8950.000000	8950.000000	8950.000000	8949.000000	
mean	0.135144	3.248827	14.709832	4494.449450	
std	0.200121	6.824647	24.857649	3638.815725	
min	0.000000	0.000000	0.000000	50.000000	
25%	0.000000	0.000000	1.000000	1600.000000	
50%	0.000000	0.000000	7.000000	3000.000000	
75%	0.222222	4.000000	17.000000	6500.000000	
max	1.500000	123.000000	358.000000	30000.000000	

	PAYMENTS	MINIMUM_PAYMENTS	PRC_FULL_PAYMENT	TENURE
count	8950.000000	8637.000000	8950.000000	8950.000000
mean	1733.143852	864.206542	0.153715	11.517318
std	2895.063757	2372.446607	0.292499	1.338331
min	0.000000	0.019163	0.000000	6.000000
25%	383.276166	169.123707	0.000000	12.000000
50%	856.901546	312.343947	0.000000	12.000000
75%	1901.134317	825.485459	0.142857	12.000000
max	50721.483360	76406.207520	1.000000	12.000000

[34]:
```
df.isnull().sum().sort_values(ascending=False)
```

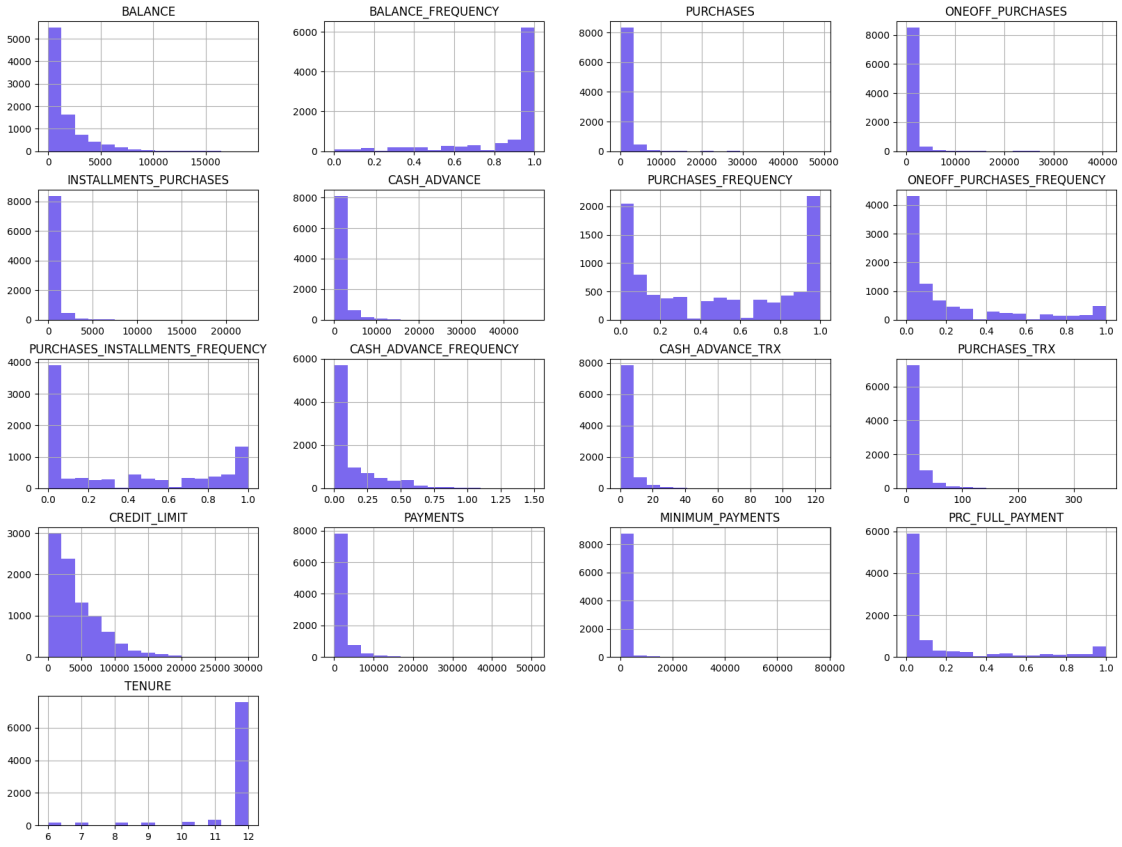
[34]:
```
MINIMUM_PAYMENTS                    313
CREDIT_LIMIT                          1
CUST_ID                               0
BALANCE                               0
PRC_FULL_PAYMENT                      0
PAYMENTS                              0
PURCHASES_TRX                         0
CASH_ADVANCE_TRX                      0
CASH_ADVANCE_FREQUENCY                0
PURCHASES_INSTALLMENTS_FREQUENCY      0
ONEOFF_PURCHASES_FREQUENCY            0
PURCHASES_FREQUENCY                   0
CASH_ADVANCE                          0
INSTALLMENTS_PURCHASES                0
ONEOFF_PURCHASES                      0
PURCHASES                             0
BALANCE_FREQUENCY                     0
TENURE                                0
dtype: int64
```

[35]:
```
df.loc[(df['MINIMUM_PAYMENTS'].isnull()==True),'MINIMUM_PAYMENTS'] =␣
↪df['MINIMUM_PAYMENTS'].mean()
df.loc[(df['CREDIT_LIMIT'].isnull()==True),'CREDIT_LIMIT'] = df['CREDIT_LIMIT'].
↪mean()
```

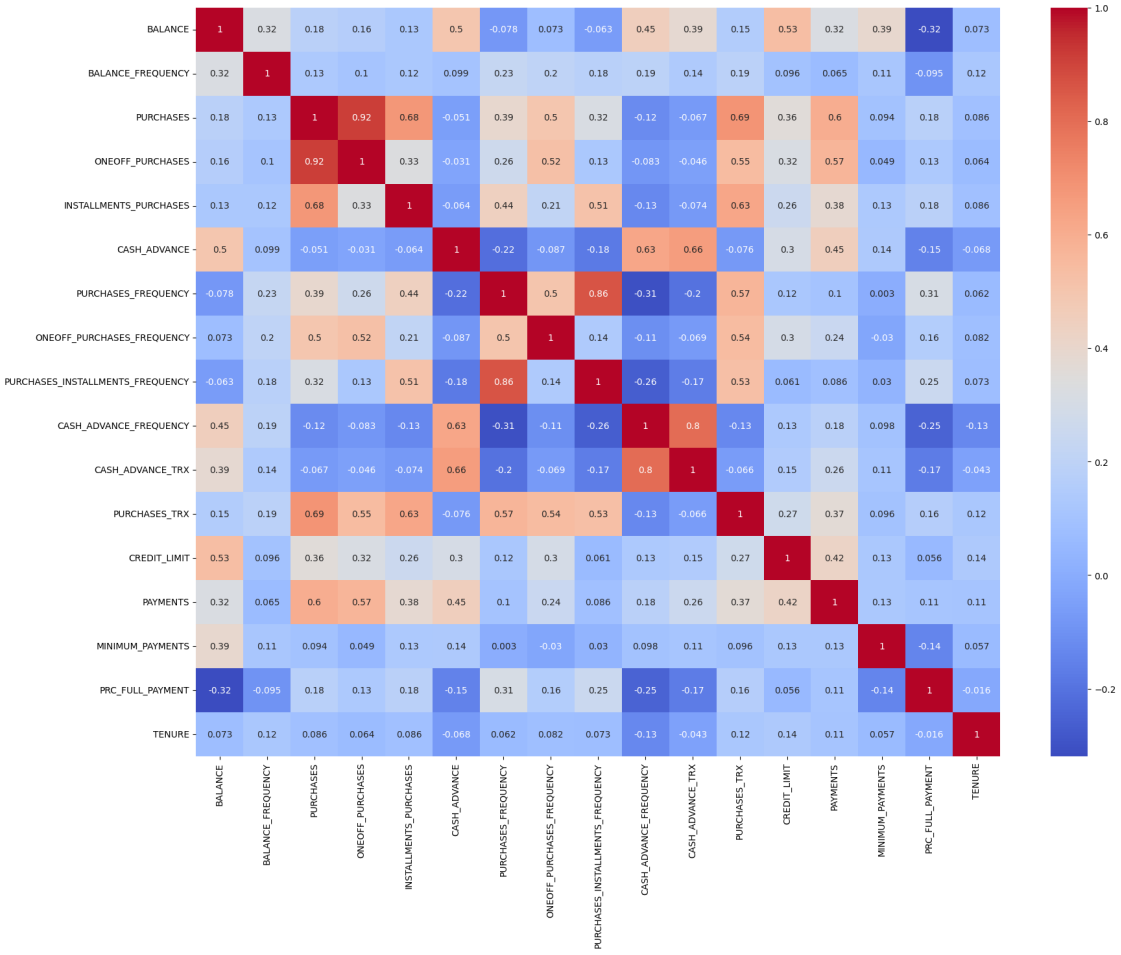
[36]:
```
df.isnull().sum().sort_values(ascending=False).head()
```

[36]:
```
CUST_ID             0
BALANCE             0
PRC_FULL_PAYMENT    0
MINIMUM_PAYMENTS    0
PAYMENTS            0
dtype: int64
```

[37]:
```
num_col = df.select_dtypes(exclude=['object']).columns
df[num_col].hist(bins=15, figsize=(20, 15), layout=(5,␣
↪4),color="mediumslateblue");
```



[38]:
```
df[num_col].corr()
plt.subplots(figsize=(20,15))
sns.heatmap(df[num_col].corr(),annot = True, cmap = "coolwarm");
```



Filling in missing data with KnnImputer

[39]:
```
from sklearn.impute import KNNImputer
imputer = KNNImputer()
imp_data = pd.DataFrame(imputer.fit_transform(df[num_col]),columns=df[num_col].
  ↪columns)
imp_data.isna().sum()
```

[39]:
```
BALANCE                             0
BALANCE_FREQUENCY                   0
PURCHASES                           0
ONEOFF_PURCHASES                    0
INSTALLMENTS_PURCHASES              0
CASH_ADVANCE                        0
PURCHASES_FREQUENCY                 0
ONEOFF_PURCHASES_FREQUENCY          0
PURCHASES_INSTALLMENTS_FREQUENCY    0
CASH_ADVANCE_FREQUENCY              0
```

```
CASH_ADVANCE_TRX                    0
PURCHASES_TRX                       0
CREDIT_LIMIT                        0
PAYMENTS                            0
MINIMUM_PAYMENTS                    0
PRC_FULL_PAYMENT                    0
TENURE                              0
dtype: int64
```

[40]:
```
imp_data
```

[40]:
```
          BALANCE  BALANCE_FREQUENCY  PURCHASES  ONEOFF_PURCHASES  \
0       40.900749           0.818182      95.40              0.00
1     3202.467416           0.909091       0.00              0.00
2     2495.148862           1.000000     773.17            773.17
3     1666.670542           0.636364    1499.00           1499.00
4      817.714335           1.000000      16.00             16.00
…             …                  …         …                 …
8945    28.493517           1.000000     291.12              0.00
8946    19.183215           1.000000     300.00              0.00
8947    23.398673           0.833333     144.40              0.00
8948    13.457564           0.833333       0.00              0.00
8949   372.708075           0.666667    1093.25           1093.25

      INSTALLMENTS_PURCHASES  CASH_ADVANCE  PURCHASES_FREQUENCY  \
0                      95.40      0.000000             0.166667
1                       0.00   6442.945483             0.000000
2                       0.00      0.000000             1.000000
3                       0.00    205.788017             0.083333
4                       0.00      0.000000             0.083333
…                        …            …                    …
8945                  291.12      0.000000             1.000000
8946                  300.00      0.000000             1.000000
8947                  144.40      0.000000             0.833333
8948                    0.00     36.558778             0.000000
8949                    0.00    127.040008             0.666667

      ONEOFF_PURCHASES_FREQUENCY  PURCHASES_INSTALLMENTS_FREQUENCY  \
0                       0.000000                          0.083333
1                       0.000000                          0.000000
2                       1.000000                          0.000000
3                       0.083333                          0.000000
4                       0.083333                          0.000000
…                          …                                 …
8945                    0.000000                          0.833333
8946                    0.000000                          0.833333
8947                    0.000000                          0.666667
```

```
8948                    0.000000                          0.000000
8949                    0.666667                          0.000000

      CASH_ADVANCE_FREQUENCY  CASH_ADVANCE_TRX  PURCHASES_TRX  CREDIT_LIMIT  \
0                   0.000000               0.0            2.0        1000.0
1                   0.250000               4.0            0.0        7000.0
2                   0.000000               0.0           12.0        7500.0
3                   0.083333               1.0            1.0        7500.0
4                   0.000000               0.0            1.0        1200.0
…                        …                 …              …           …
8945                0.000000               0.0            6.0        1000.0
8946                0.000000               0.0            6.0        1000.0
8947                0.000000               0.0            5.0        1000.0
8948                0.166667               2.0            0.0         500.0
8949                0.333333               2.0           23.0        1200.0

         PAYMENTS  MINIMUM_PAYMENTS  PRC_FULL_PAYMENT  TENURE
0      201.802084        139.509787          0.000000    12.0
1     4103.032597       1072.340217          0.222222    12.0
2      622.066742        627.284787          0.000000    12.0
3        0.000000        864.206542          0.000000    12.0
4      678.334763        244.791237          0.000000    12.0
…           …                …                 …      …
8945   325.594462         48.886365          0.500000     6.0
8946   275.861322        864.206542          0.000000     6.0
8947    81.270775         82.418369          0.250000     6.0
8948    52.549959         55.755628          0.250000     6.0
8949    63.165404         88.288956          0.000000     6.0

[8950 rows x 17 columns]
```

Hierarchical Clustering

[42]:
```
from pycaret.classification import create_model, setup
clu = setup(imp_data,
            normalize=True,
            pca=True,
            session_id=123)
```

```
<pandas.io.formats.style.Styler at 0x7a51c39494b0>
```

[44]:
```
from pycaret.clustering import *
models()
```

[44]:
```
                                    Name  \
ID
kmeans                  K-Means Clustering
```

```
ap                      Affinity Propagation
meanshift             Mean Shift Clustering
sc                      Spectral Clustering
hclust              Agglomerative Clustering
dbscan     Density-Based Spatial Clustering
optics                    OPTICS Clustering
birch                      Birch Clustering
kmodes                   K-Modes Clustering

                                                    Reference
ID
kmeans                          sklearn.cluster._kmeans.KMeans
ap          sklearn.cluster._affinity_propagation.Affinity…
meanshift               sklearn.cluster._mean_shift.MeanShift
sc               sklearn.cluster._spectral.SpectralClustering
hclust      sklearn.cluster._agglomerative.AgglomerativeCl…
dbscan                          sklearn.cluster._dbscan.DBSCAN
optics                          sklearn.cluster._optics.OPTICS
birch                             sklearn.cluster._birch.Birch
kmodes                                   kmodes.kmodes.KModes
```

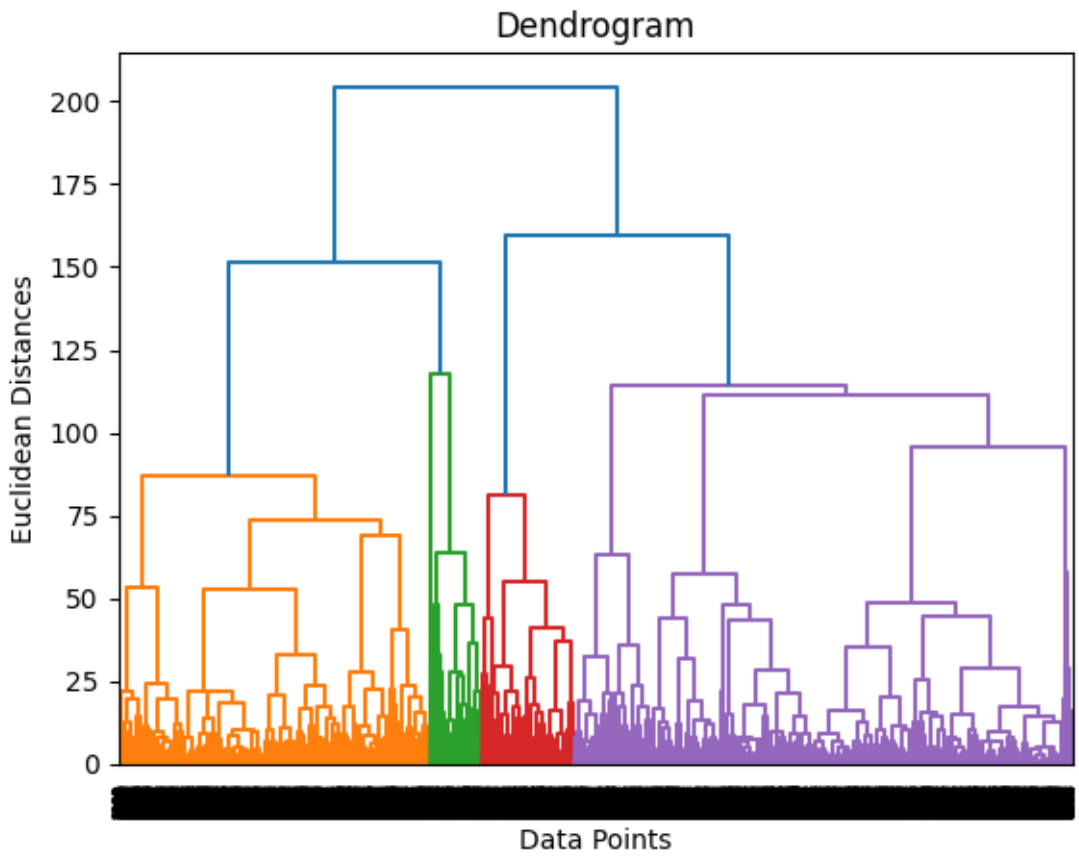
[45]:
```
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import AgglomerativeClustering
import scipy.cluster.hierarchy as sch
import matplotlib.pyplot as plt
```

[46]:
```
# Create a hierarchical clustering model
hierarchical_clust = AgglomerativeClustering(n_clusters=3)  # Adjust the number␣
 ↪of clusters as needed

# Fit the model and get cluster labels
labels = hierarchical_clust.fit_predict(imp_data)

# Add the cluster labels to the original DataFrame
df['Cluster'] = labels

# Display the DataFrame with cluster labels
print(df.head())
```

```
  CUST_ID      BALANCE  BALANCE_FREQUENCY  PURCHASES  ONEOFF_PURCHASES  \
0  C10001    40.900749           0.818182      95.40              0.00
1  C10002  3202.467416           0.909091       0.00              0.00
2  C10003  2495.148862           1.000000     773.17            773.17
3  C10004  1666.670542           0.636364    1499.00           1499.00
4  C10005   817.714335           1.000000      16.00             16.00

   INSTALLMENTS_PURCHASES  CASH_ADVANCE  PURCHASES_FREQUENCY  \
```

```
0                    95.4      0.000000             0.166667
1                     0.0   6442.945483             0.000000
2                     0.0      0.000000             1.000000
3                     0.0    205.788017             0.083333
4                     0.0      0.000000             0.083333

   ONEOFF_PURCHASES_FREQUENCY  PURCHASES_INSTALLMENTS_FREQUENCY  \
0                    0.000000                          0.083333
1                    0.000000                          0.000000
2                    1.000000                          0.000000
3                    0.083333                          0.000000
4                    0.083333                          0.000000

   CASH_ADVANCE_FREQUENCY  CASH_ADVANCE_TRX  PURCHASES_TRX  CREDIT_LIMIT  \
0                0.000000                 0              2        1000.0
1                0.250000                 4              0        7000.0
2                0.000000                 0             12        7500.0
3                0.083333                 1              1        7500.0
4                0.000000                 0              1        1200.0

      PAYMENTS  MINIMUM_PAYMENTS  PRC_FULL_PAYMENT  TENURE  Cluster
0   201.802084        139.509787          0.000000      12        1
1  4103.032597       1072.340217          0.222222      12        0
2   622.066742        627.284787          0.000000      12        0
3     0.000000        864.206542          0.000000      12        0
4   678.334763        244.791237          0.000000      12        1
```

[51]:
```python
# Standardize the data
scaler = StandardScaler()
scaled_data = scaler.fit_transform(imp_data)

# Create a dendrogram to determine the number of clusters
dendrogram = sch.dendrogram(sch.linkage(scaled_data, method='ward'))
plt.title('Dendrogram')
plt.xlabel('Data Points')
plt.ylabel('Euclidean Distances')
plt.show()
```



[52]:
```
from pycaret.clustering import setup, create_model, assign_model
h_clust = create_model('hclust')
```

```
<IPython.core.display.HTML object>

<pandas.io.formats.style.Styler at 0x7a51be2c0ac0>

Processing:   0%|          | 0/3 [00:00<?, ?it/s]
```

[53]:
```
h_results = assign_model(h_clust)
h_results.head()
```

[53]:

	CUST_ID	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	\
0	C10001	40.900749	0.818182	95.400002	0.000000	
1	C10002	3202.467529	0.909091	0.000000	0.000000	
2	C10003	2495.148926	1.000000	773.169983	773.169983	
3	C10004	1666.670532	0.636364	1499.000000	1499.000000	
4	C10005	817.714355	1.000000	16.000000	16.000000	

	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHASES_FREQUENCY	\
0	95.400002	0.000000	0.166667	

```
1                0.000000   6442.945312             0.000000
2                0.000000      0.000000             1.000000
3                0.000000    205.788010             0.083333
4                0.000000      0.000000             0.083333

   ONEOFF_PURCHASES_FREQUENCY  PURCHASES_INSTALLMENTS_FREQUENCY  \
0                    0.000000                          0.083333
1                    0.000000                          0.000000
2                    1.000000                          0.000000
3                    0.083333                          0.000000
4                    0.083333                          0.000000

   CASH_ADVANCE_FREQUENCY  CASH_ADVANCE_TRX  PURCHASES_TRX  CREDIT_LIMIT  \
0                0.000000                 0              2        1000.0
1                0.250000                 4              0        7000.0
2                0.000000                 0             12        7500.0
3                0.083333                 1              1        7500.0
4                0.000000                 0              1        1200.0

      PAYMENTS  MINIMUM_PAYMENTS  PRC_FULL_PAYMENT  TENURE    Cluster
0   201.802078        139.509781          0.000000      12  Cluster 3
1  4103.032715       1072.340210          0.222222      12  Cluster 0
2   622.066772        627.284790          0.000000      12  Cluster 0
3     0.000000        864.206543          0.000000      12  Cluster 0
4   678.334778        244.791245          0.000000      12  Cluster 3
```