In [1]:

```
#importing libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler, normalize
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
#importing dataset
creditcard_df = pd.read_csv('D:\\Data Science for Business Package\\2. Marketing Dep
creditcard_df
```

Out[1]:

	CUST_ID	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	I
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	
...	...	...	...	...	...	
8945	C19186	28.493517	1.000000	291.12	0.00	
8946	C19187	19.183215	1.000000	300.00	0.00	
8947	C19188	23.398673	0.833333	144.40	0.00	
8948	C19189	13.457564	0.833333	0.00	0.00	
8949	C19190	372.708075	0.666667	1093.25	1093.25	

8950 rows × 18 columns

In [2]:

```
#features
# CUSTID: Identification of Credit Card holder
# BALANCE: Balance amount left in customer's account to make purchases
# BALANCE_FREQUENCY: How frequently the Balance is updated, score between 0 and 1 (1
# PURCHASES: Amount of purchases made from account
# ONEOFFPURCHASES: Maximum purchase amount done in one-go
# INSTALLMENTS_PURCHASES: Amount of purchase done in installment
# CASH_ADVANCE: Cash in advance given by the user
# PURCHASES_FREQUENCY: How frequently the Purchases are being made, score between 0
# ONEOFF_PURCHASES_FREQUENCY: How frequently Purchases are happening in one-go (1 =
# PURCHASES_INSTALLMENTS_FREQUENCY: How frequently purchases in installments are bei
# CASH_ADVANCE_FREQUENCY: How frequently the cash in advance being paid
# CASH_ADVANCE_TRX: Number of Transactions made with "Cash in Advance"
# PURCHASES_TRX: Number of purchase transactions made
# CREDIT_LIMIT: Limit of Credit Card for user
# PAYMENTS: Amount of Payment done by user
# MINIMUM_PAYMENTS: Minimum amount of payments made by user
# PRC_FULL_PAYMENT: Percent of full payment paid by user
# TENURE: Tenure of credit card service for user
```

In [3]:

```
creditcard_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
```

In [4]:

```
1 creditcard_df.describe()
```

Out[4]:

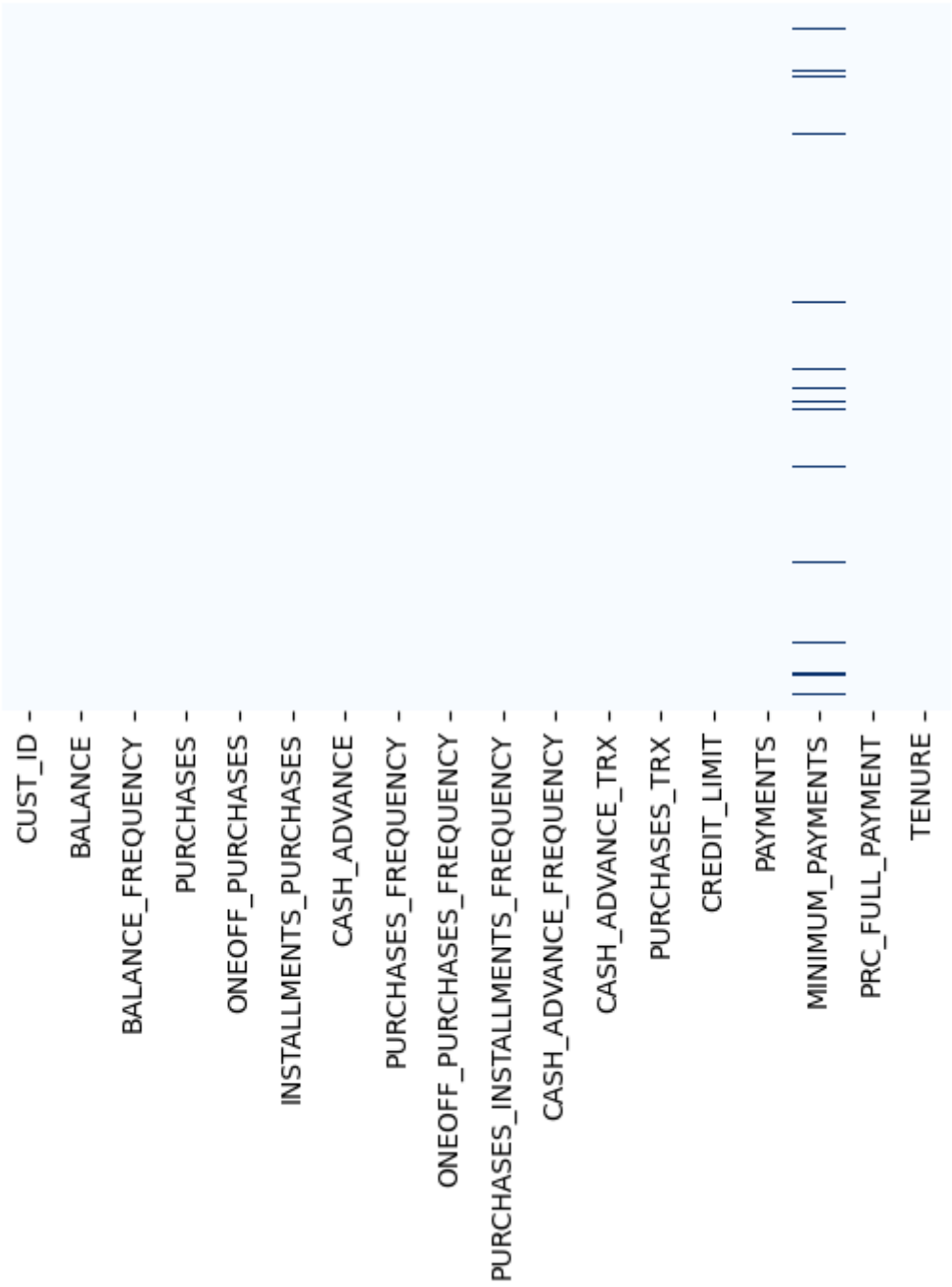
	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLI
count	8950.000000	8950.000000	8950.000000	8950.000000	
mean	1564.474828	0.877271	1003.204834	592.437371	
std	2081.531879	0.236904	2136.634782	1659.887917	
min	0.000000	0.000000	0.000000	0.000000	
25%	128.281915	0.888889	39.635000	0.000000	
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	

In [5]:

```
#visualizing null values
sns.heatmap(creditcard_df.isnull(), yticklabels = False, cbar = False, cmap="Blues")

```

Out[5]:

<AxesSubplot: >



In [6]:

```
creditcard_df.isnull().sum()
```

Out[6]:

```
CUST_ID                             0
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                        1
PAYMENTS                            0
MINIMUM_PAYMENTS                  313
PRC_FULL_PAYMENT                    0
TENURE                              0
dtype: int64
```

In [7]:

```
# Fill up the missing elements
creditcard_df.loc[(creditcard_df['MINIMUM_PAYMENTS'].isnull() == True), 'MINIMUM_PAY

```

In [8]:

```
# Fill up the missing elements
creditcard_df.loc[(creditcard_df['CREDIT_LIMIT'].isnull() == True), 'CREDIT_LIMIT']
```

In [9]:

```
# checking duplicate values
creditcard_df.duplicated().sum()
```

Out[9]:

```
0
```

In [10]:

```
1 creditcard_df.head()
```

Out[10]:

	CUST_ID	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INS
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	

In [11]:

```
1 #droping feature
2 creditcard_df.drop("CUST_ID", axis = 1, inplace= True)
3 creditcard_df
```

Out[11]:

	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLME
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	

8950 rows × 17 columns

In [12]:

```
plt.figure(figsize=(10,50))
for i in range(len(creditcard_df.columns)):
  plt.subplot((len(creditcard_df.columns)), 1, i+1)
  sns.distplot(creditcard_df[creditcard_df.columns[i]], kde_kws={"color": "b", "lw":
  plt.title(creditcard_df.columns[i])

plt.tight_layout()

```

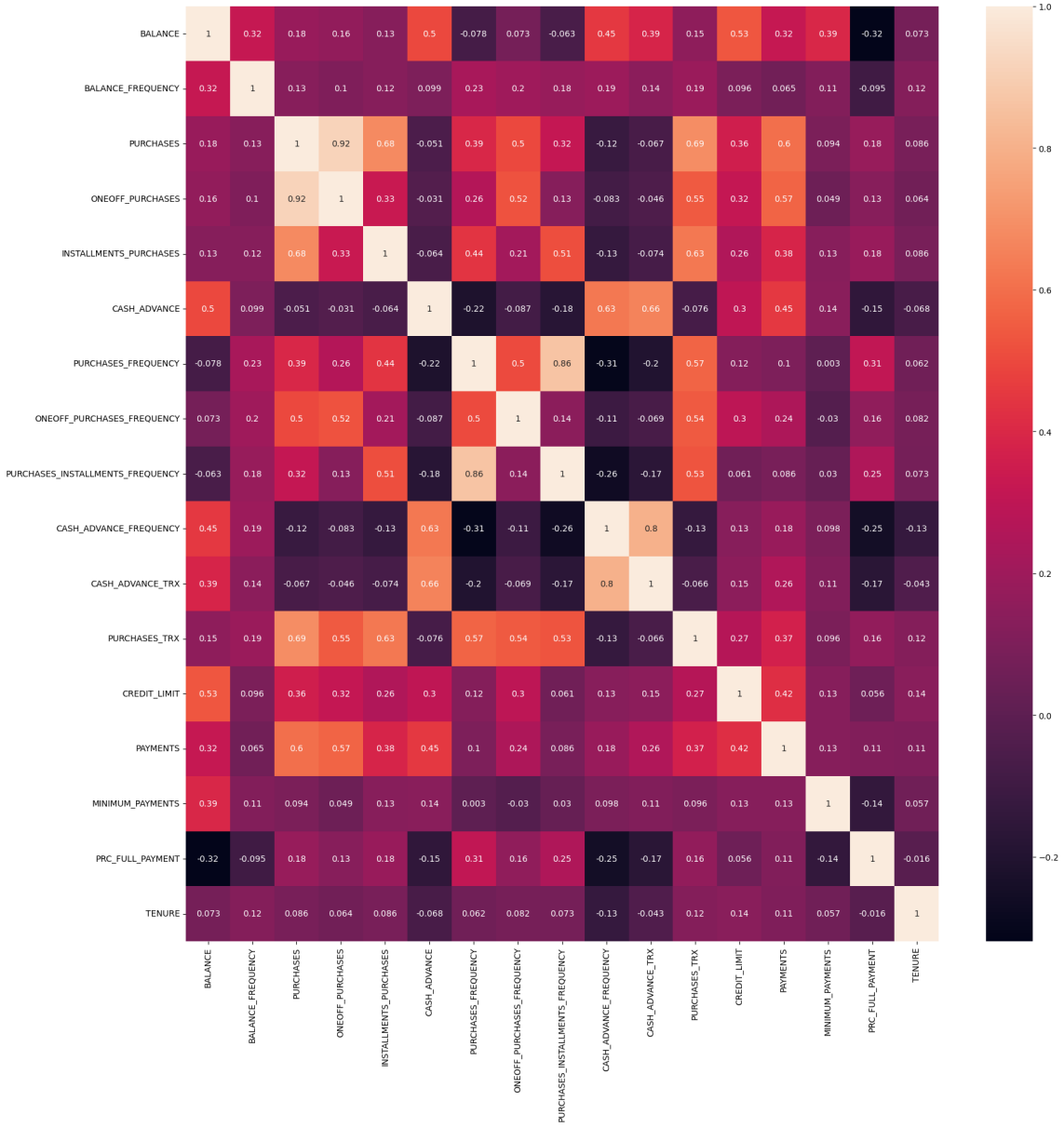
```
C:\Users\Pradeep\AppData\Local\Temp\ipykernel_25616\747322071.py:4: Use
rWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.1
4.0.

Please adapt your code to use either `displot` (a figure-level function
with
similar flexibility) or `histplot` (an axes-level function for histogra
ms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (http
s://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(creditcard_df[creditcard_df.columns[i]], kde_kws={"colo
r": "b", "lw": 3, "label": "KDE"}, hist_kws={"color": "g"})
C:\Users\Pradeep\AppData\Local\Temp\ipykernel_25616\747322071.py:4: Use
rWarning:
```

In [13]:

```
#correlation plot
f, ax = plt.subplots(figsize = (20, 20))
sns.heatmap(creditcard_df.corr(), annot = True)

```

Out[13]:

<AxesSubplot: >



In [14]:

```
#scaling the data
scaler = StandardScaler()
creditcard_df_scaled = scaler.fit_transform(creditcard_df)
```

In [15]:

```
#elbow plot for no of clusters
scores_1 = []

range_values = range(1, 20)

for i in range_values:
  kmeans = KMeans(n_clusters = i)
  kmeans.fit(creditcard_df_scaled)
  scores_1.append(kmeans.inertia_)

plt.plot(scores_1, 'x-')
plt.title('Finding the right number of clusters')
plt.xlabel('Clusters')
plt.ylabel('Scores')
plt.show()

```

```
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
```

```
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
```

In [16]:

```
#apply KMEANS
kmeans = KMeans(8)
kmeans.fit(creditcard_df_scaled)
labels = kmeans.labels_
```

```
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
```

In [17]:

```
print(kmeans.cluster_centers_.shape)
print("centroid point\n",kmeans.cluster_centers_)
```

```
(8, 17)
centroid point
 [[-1.20442106e-01  4.02699267e-01  5.79851494e-01  7.03609956e-01
   7.81975316e-02 -3.33108401e-01  9.98102693e-01  1.91406317e+00
   2.13137767e-01 -4.22686809e-01 -3.36697878e-01  6.64851547e-01
   4.68463996e-01  1.72617202e-01 -1.56173965e-01  4.63366546e-01
   2.74995888e-01]
 [-7.00762713e-01 -2.14030902e+00 -3.09740586e-01 -2.34356396e-01
  -3.01708455e-01 -3.20376411e-01 -5.53141715e-01 -4.41318119e-01
  -4.38249132e-01 -5.21411574e-01 -3.76356028e-01 -4.18151679e-01
  -1.73838959e-01 -1.91518868e-01 -2.56660487e-01  2.88039803e-01
   2.02767206e-01]
 [ 5.45478694e-03  4.03039463e-01 -3.54119574e-01 -2.38698657e-01
  -3.98669620e-01 -1.04268851e-01 -8.40918849e-01 -3.82633698e-01
  -7.51078473e-01  8.72301669e-02 -3.96464753e-02 -4.77542444e-01
  -3.07989926e-01 -2.53183607e-01 -1.31176639e-02 -4.55115693e-01
   2.73532981e-01]
 [ 1.10639616e+00  4.75193093e-01  2.59960254e+00  1.86486936e+00
   2.72075736e+00 -1.22047896e-01  1.16917640e+00  1.57834607e+00
   1.29097192e+00 -2.61630966e-01 -1.21752321e-01  3.25571270e+00
   1.34920399e+00  1.53000655e+00  6.17127927e-01  2.47010309e-01
   3.35231702e-01]
 [ 1.66021988e+00  3.96119231e-01 -2.22143812e-01 -1.57786787e-01
  -2.35440730e-01  1.98178259e+00 -4.83145870e-01 -2.11910088e-01
  -4.24003122e-01  1.91150544e+00  1.90809584e+00 -2.71298441e-01
   1.00452777e+00  7.89642588e-01  5.41194989e-01 -3.95893013e-01
   7.50189565e-02]
 [-3.58012219e-01  3.32525127e-01 -2.95407248e-02 -2.40512089e-01
   3.71816764e-01 -3.61514344e-01  9.96780085e-01 -3.80547652e-01
   1.20778087e+00 -4.71384129e-01 -3.59029975e-01  1.95018510e-01
  -2.59745161e-01 -2.12570099e-01 -2.98669130e-02  3.13075087e-01
   2.55465614e-01]
 [-3.34901439e-01 -3.49885235e-01 -2.87298367e-01 -2.12974259e-01
  -2.87679154e-01  6.80198997e-02 -2.03948289e-01 -2.84236496e-01
  -2.27466186e-01  3.07808330e-01 -1.79210756e-04 -3.87811550e-01
  -5.62208511e-01 -3.91666335e-01 -2.08921003e-01  1.27483622e-02
  -3.19766866e+00]
 [ 1.86590651e+00  3.32878117e-01  1.25968437e+01  1.31111821e+01
   5.69638874e+00 -4.17760945e-03  1.03332159e+00  2.17063444e+00
   8.66220718e-01 -4.76183389e-01 -2.14853617e-01  4.56655817e+00
   3.17419740e+00  8.99113558e+00  1.06481726e+00  1.23631700e+00
   2.95702050e-01]]
```

In [18]:

```
cluster_centers = pd.DataFrame(data = kmeans.cluster_centers_, columns = [creditcard
cluster_centers
```

Out[18]:

	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS_
0	-0.120442	0.402699	0.579851	0.703610	
1	-0.700763	-2.140309	-0.309741	-0.234356	
2	0.005455	0.403039	-0.354120	-0.238699	
3	1.106396	0.475193	2.599603	1.864869	
4	1.660220	0.396119	-0.222144	-0.157787	
5	-0.358012	0.332525	-0.029541	-0.240512	
6	-0.334901	-0.349885	-0.287298	-0.212974	
7	1.865907	0.332878	12.596844	13.111182	

In [19]:

```
# In order to understand what these numbers mean, let's perform inverse transformati
cluster_centers = scaler.inverse_transform(cluster_centers)
cluster_centers = pd.DataFrame(data = cluster_centers, columns = [creditcard_df.colu
cluster_centers
```

Out[19]:

	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENT
0	1313.784751	0.972666	2242.066489	1760.285787	
1	105.896392	0.370251	341.439298	203.453753	
2	1575.828506	0.972747	246.622905	196.246490	
3	3867.345034	0.989840	6557.295728	3687.738545	
4	5020.082361	0.971108	528.591155	330.543621	
5	819.302614	0.956043	940.090620	193.236564	
6	867.405752	0.794386	389.387444	238.943722	
7	5448.201718	0.956126	27916.555652	22354.314348	

In [20]:

```
# concatenate the clusters labels to our original dataframe
creditcard_df_cluster = pd.concat([creditcard_df, pd.DataFrame({'cluster':labels})],
creditcard_df_cluster.head()
```

Out[20]:

	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS
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	

In [21]:

```
# Obtain the principal components
pca = PCA(n_components=2)
principal_comp = pca.fit_transform(creditcard_df_scaled)
principal_comp
```

Out[21]:

```
array([[-1.68222111, -1.07644631],
       [-1.13829816,  2.50649826],
       [ 0.96968089, -0.38350334],
       ...,
       [-0.92620277, -1.8107928 ],
       [-2.33654826, -0.65798708],
       [-0.55641888, -0.40048456]])
```

In [22]:

```
# Create a dataframe with the two components
pca_df = pd.DataFrame(data = principal_comp, columns =['pca1','pca2'])
pca_df.head()
```

Out[22]:

	pca1	pca2
0	-1.682221	-1.076446
1	-1.138298	2.506498
2	0.969681	-0.383503
3	-0.873629	0.043177
4	-1.599434	-0.688579

In [23]:

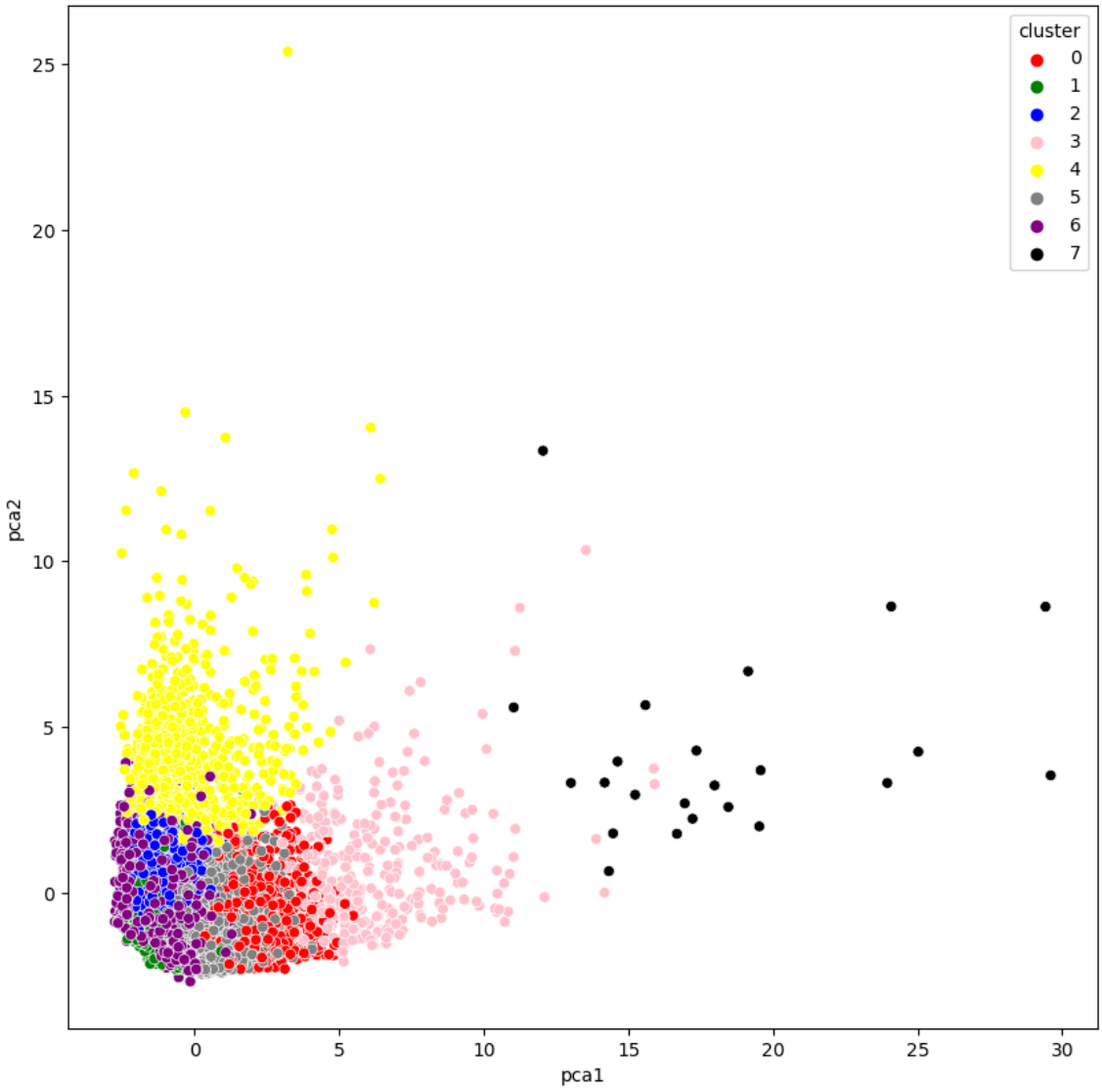
```
# Concatenate the clusters labels to the dataframe
pca_df = pd.concat([pca_df,pd.DataFrame({'cluster':labels})], axis = 1)
pca_df.head()
```

Out[23]:

	pca1	pca2	cluster
0	-1.682221	-1.076446	2
1	-1.138298	2.506498	4
2	0.969681	-0.383503	0
3	-0.873629	0.043177	2
4	-1.599434	-0.688579	2

In [24]:

```
plt.figure(figsize=(10,10))
ax = sns.scatterplot(x="pca1", y="pca2", hue = "cluster", data = pca_df, palette =['
plt.show()
```



In [25]:

```
from tensorflow.keras.layers import Input, Add, Dense, Activation, ZeroPadding2D, Ba
from tensorflow.keras.models import Model, load_model
from tensorflow.keras.initializers import glorot_uniform
from keras.optimizers import SGD

encoding_dim = 7

input_df = Input(shape=(17,))


# Glorot normal initializer (Xavier normal initializer) draws samples from a truncat

x = Dense(encoding_dim, activation='relu')(input_df)
x = Dense(500, activation='relu', kernel_initializer = 'glorot_uniform')(x)
x = Dense(500, activation='relu', kernel_initializer = 'glorot_uniform')(x)
x = Dense(2000, activation='relu', kernel_initializer = 'glorot_uniform')(x)

encoded = Dense(10, activation='relu', kernel_initializer = 'glorot_uniform')(x)

x = Dense(2000, activation='relu', kernel_initializer = 'glorot_uniform')(encoded)
x = Dense(500, activation='relu', kernel_initializer = 'glorot_uniform')(x)

decoded = Dense(17, kernel_initializer = 'glorot_uniform')(x)

# autoencoder
autoencoder = Model(input_df, decoded)

#encoder - used for our dimention reduction
encoder = Model(input_df, encoded)

autoencoder.compile(optimizer= 'adam', loss='mean_squared_error')

```

In [26]:

```
1 autoencoder.fit(creditcard_df_scaled, creditcard_df_scaled, batch_size = 128, epochs
```

```
Epoch 1/25
70/70 [==============================] - 5s 38ms/step - loss: 0.5431
Epoch 2/25
70/70 [==============================] - 3s 43ms/step - loss: 0.2705
Epoch 3/25
70/70 [==============================] - 3s 39ms/step - loss: 0.2065
Epoch 4/25
70/70 [==============================] - 3s 44ms/step - loss: 0.1836
Epoch 5/25
70/70 [==============================] - 3s 45ms/step - loss: 0.1544
Epoch 6/25
70/70 [==============================] - 3s 45ms/step - loss: 0.1404
Epoch 7/25
70/70 [==============================] - 3s 46ms/step - loss: 0.1321
Epoch 8/25
70/70 [==============================] - 4s 58ms/step - loss: 0.1238
Epoch 9/25
70/70 [==============================] - 4s 54ms/step - loss: 0.1278
Epoch 10/25
70/70 [==============================] - 3s 43ms/step - loss: 0.1199
Epoch 11/25
70/70 [==============================] - 3s 47ms/step - loss: 0.1081
Epoch 12/25
70/70 [==============================] - 3s 46ms/step - loss: 0.0996
Epoch 13/25
70/70 [==============================] - 3s 43ms/step - loss: 0.0973
Epoch 14/25
70/70 [==============================] - 3s 40ms/step - loss: 0.0900
Epoch 15/25
70/70 [==============================] - 2s 36ms/step - loss: 0.0834
Epoch 16/25
70/70 [==============================] - 3s 39ms/step - loss: 0.0862
Epoch 17/25
70/70 [==============================] - 3s 42ms/step - loss: 0.0789
Epoch 18/25
70/70 [==============================] - 3s 41ms/step - loss: 0.0712
Epoch 19/25
70/70 [==============================] - 3s 42ms/step - loss: 0.0705
Epoch 20/25
70/70 [==============================] - 3s 40ms/step - loss: 0.0664
Epoch 21/25
70/70 [==============================] - 3s 47ms/step - loss: 0.0646
Epoch 22/25
70/70 [==============================] - 3s 44ms/step - loss: 0.0690
Epoch 23/25
70/70 [==============================] - 3s 43ms/step - loss: 0.0606
Epoch 24/25
70/70 [==============================] - 4s 51ms/step - loss: 0.0616
Epoch 25/25
70/70 [==============================] - 3s 45ms/step - loss: 0.0547
```

Out[26]:

```
<keras.callbacks.History at 0x218ceb769a0>
```

In [27]:

```
pred = encoder.predict(creditcard_df_scaled)
print(len(pred))
```

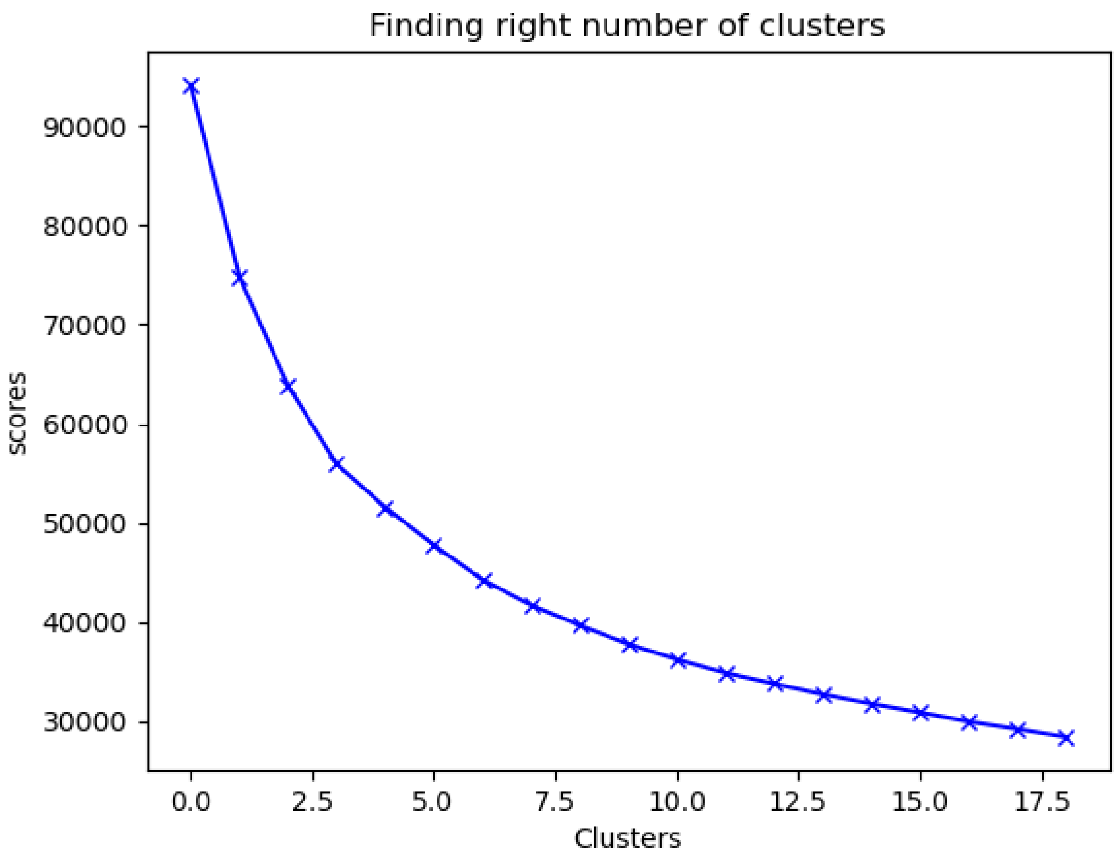
```
280/280 [==============================] - 2s 7ms/step
8950
```

In [28]:

```
scores_2 = []

range_values = range(1, 20)

for i in range_values:
  kmeans = KMeans(n_clusters= i)
  kmeans.fit(pred)
  scores_2.append(kmeans.inertia_)

plt.plot(scores_2, 'bx-')
plt.title('Finding right number of clusters')
plt.xlabel('Clusters')
plt.ylabel('scores')
plt.show()
```

```
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
```

```
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
```



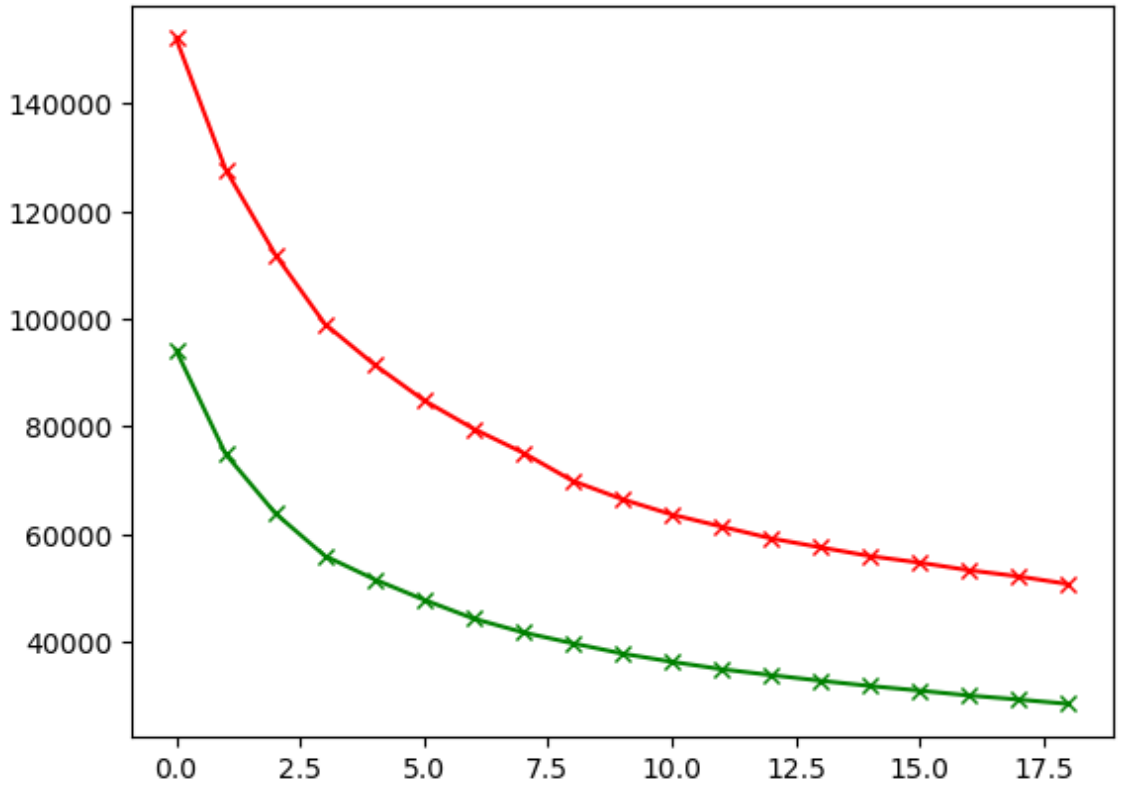
In [29]:

```
plt.plot(scores_1, 'bx-', color = 'r')
plt.plot(scores_2, 'bx-', color = 'g')
```

```
C:\Users\Pradeep\AppData\Local\Temp\ipykernel_25616\3067751309.py:1: UserW
arning: color is redundantly defined by the 'color' keyword argument and t
he fmt string "bx-" (-> color='b'). The keyword argument will take precede
nce.
  plt.plot(scores_1, 'bx-', color = 'r')
C:\Users\Pradeep\AppData\Local\Temp\ipykernel_25616\3067751309.py:2: UserW
arning: color is redundantly defined by the 'color' keyword argument and t
he fmt string "bx-" (-> color='b'). The keyword argument will take precede
nce.
  plt.plot(scores_2, 'bx-', color = 'g')
```

Out[29]:

```
[<matplotlib.lines.Line2D at 0x218d08d96d0>]
```



In [30]:

```
kmeans = KMeans(4)
kmeans.fit(pred)
labels = kmeans.labels_
y_kmeans = kmeans.fit_predict(creditcard_df_scaled)
```

```
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Pradeep\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
```

In [31]:

```
df_cluster_dr = pd.concat([creditcard_df, pd.DataFrame({'cluster':labels})], axis =
df_cluster_dr.head()
```

Out[31]:

	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS
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	

In [32]:

```
pca = PCA(n_components=2)
prin_comp = pca.fit_transform(pred)
pca_df = pd.DataFrame(data = prin_comp, columns =['pca1','pca2'])
pca_df.head()
```

Out[32]:

	pca1	pca2
0	-1.792409	-0.329980
1	0.594816	1.449199
2	-0.427213	-0.965590
3	-0.833271	0.035266
4	-1.744368	-0.176150

In [33]:

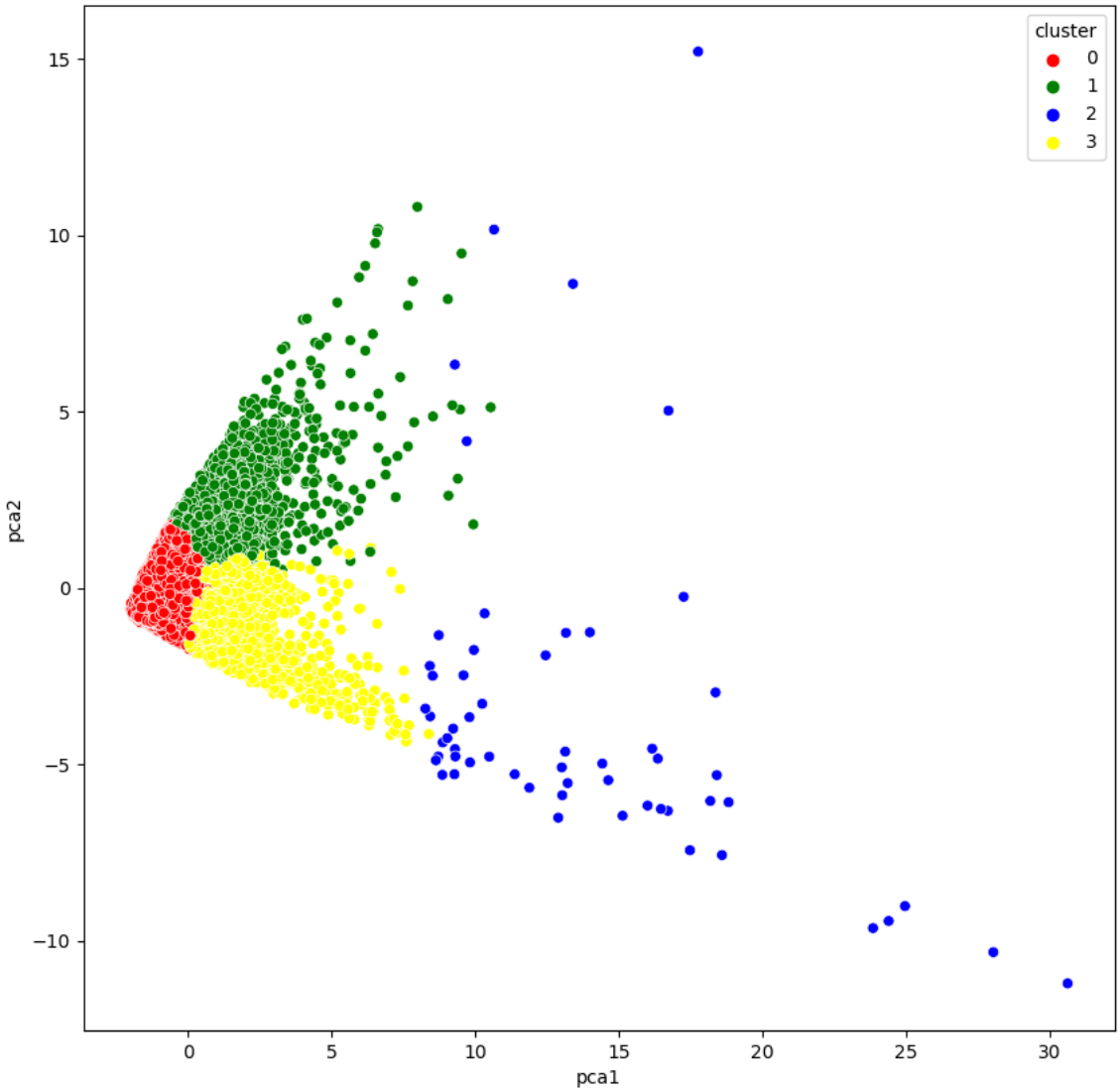
```
pca_df = pd.concat([pca_df,pd.DataFrame({'cluster':labels})], axis = 1)
pca_df.head()
```

Out[33]:

	pca1	pca2	cluster
0	-1.792409	-0.329980	0
1	0.594816	1.449199	1
2	-0.427213	-0.965590	0
3	-0.833271	0.035266	0
4	-1.744368	-0.176150	0

In [34]:

```
plt.figure(figsize=(10,10))
ax = sns.scatterplot(x="pca1", y="pca2", hue = "cluster", data = pca_df, palette =['
plt.show()
```



In [ ]:

```
1
```