CLUSTERING ANALYSIS

PREPARED BY
MURALIDHARAN N

CLUSTERING ANALYSIS

A leading bank wants to develop a customer segmentation to give promotional offers to its customers. They collected a sample that summarizes the activities of users during the past few months. You are given the task to identify the segments based on credit card usage.

Data Dictionary for Market Segmentation:

1. spending: Amount spent by the customer per month (in 1000s)
2. advance payments: Amount paid by the customer in advance by cash (in 100s)
3. probability_of_full_payment: Probability of payment done in full by the customer to the bank
4. current balance: Balance amount left in the account to make purchases (in 1000s)
5. credit limit: Limit of the amount in credit card (10000s)
6. min_payment_amt : minimum paid by the customer while making payments for purchases made monthly (in 100s)
7. max_spent_in_single_shopping: Maximum amount spent in one purchase (in 1000s)

1.1 Read the data and do exploratory data analysis. Describe the data briefly.

So, we will import all the necessary libraries for cluster analysis,

Import numpy as np

Import pandas as pd

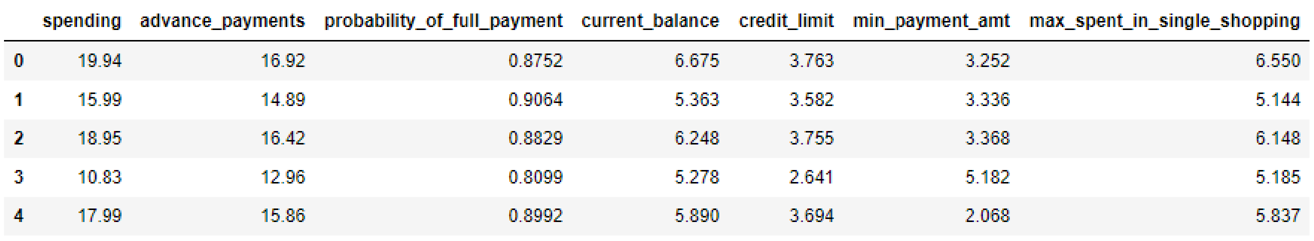
Import matplotlib.pyplot as plt

Import seaborn as sns

From sklearn.cluster import KMeans

From sklearn.metrics import silhouette samples, silhouette score

Reading the data,



	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	19.94	16.92	0.8752	6.675	3.763	3.252	6.550
1	15.99	14.89	0.9064	5.363	3.582	3.336	5.144
2	18.95	16.42	0.8829	6.248	3.755	3.368	6.148
3	10.83	12.96	0.8099	5.278	2.641	5.182	5.185
4	17.99	15.86	0.8992	5.890	3.694	2.068	5.837

- The data seems to be perfect
- The shape of the data is (210, 7)
- The info of the data indicates that all values are float
- No Null values in the data
- No missing values in the data

Description of the Data

	count	mean	std	min	25%	50%	75%	max
spending	210.0	14.847524	2.909699	10.5900	12.27000	14.35500	17.305000	21.1800
advance_payments	210.0	14.559286	1.305959	12.4100	13.45000	14.32000	15.715000	17.2500
probability_of_full_payment	210.0	0.870999	0.023629	0.8081	0.85690	0.87345	0.887775	0.9183
current_balance	210.0	5.628533	0.443063	4.8990	5.26225	5.52350	5.979750	6.6750
credit_limit	210.0	3.258605	0.377714	2.6300	2.94400	3.23700	3.561750	4.0330
min_payment_amt	210.0	3.700201	1.503557	0.7651	2.56150	3.59900	4.768750	8.4560
max_spent_in_single_shopping	210.0	5.408071	0.491480	4.5190	5.04500	5.22300	5.877000	6.5500

We have 7 variables,

- No null values present in any variables.
- The mean and median values seems to be almost equal.
- The standard deviation for spending is high when compared to other variables.
- No duplicates in the dataset

Checking for Duplicates

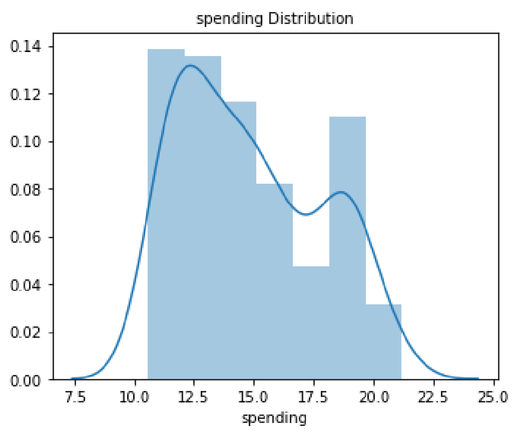
```
dups = df_clust.duplicated()
print('Number of duplicate rows = %d' % (dups.sum()))
```

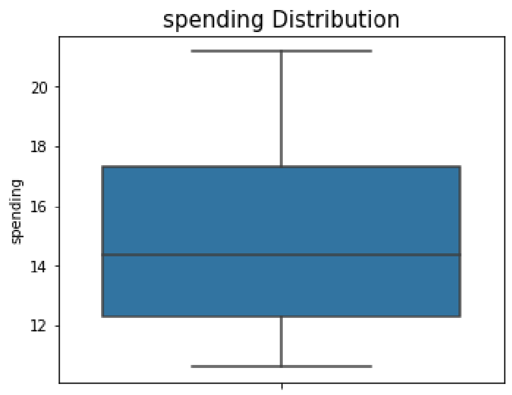
```
Number of duplicate rows = 0
```

Exploratory Data Analysis

Univariate / Bivariate analysis

Helps us to understand the distribution of data in the dataset. With univariate analysis we can find patterns and we can summarize the data and have understanding about the data to solve our business problem.



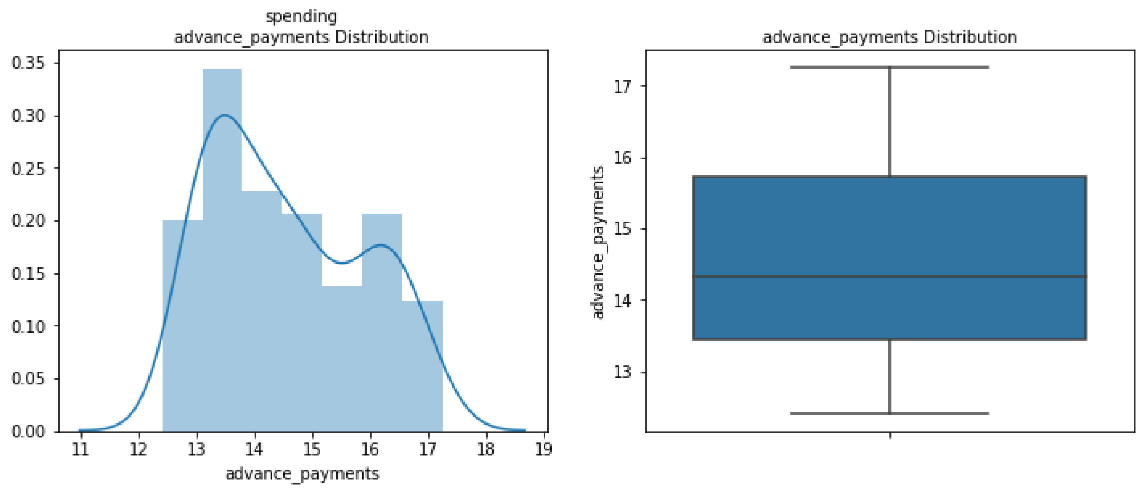


The box plot of the spending variable shows no outliers.

Spending is positively skewed - 0.399889.

We could also understand there could be chance of multi modes in the dataset.

The dist plot shows the distribution of data from 10 to 22

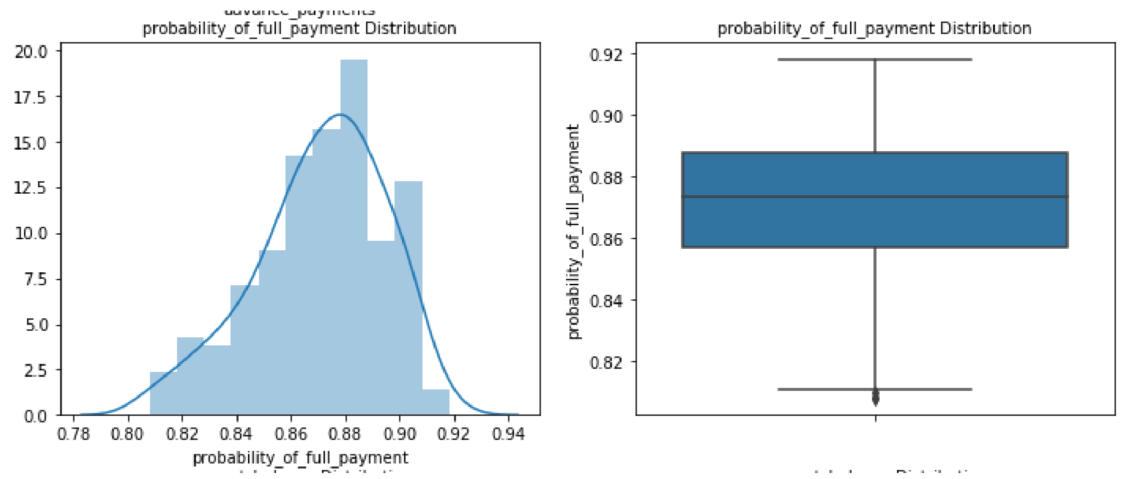


The box plot of the advance payments variable shows no outliers.

advance payments is positively skewed - 0.386573.

We could also understand there could be chance of multi modes in the dataset.

The dist plot shows the distribution of data from 12 to 17

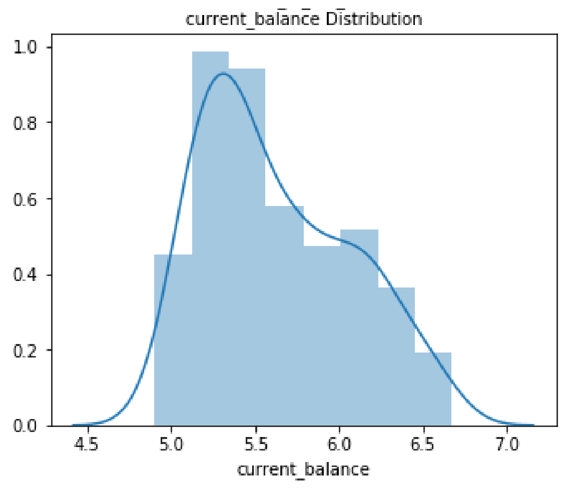


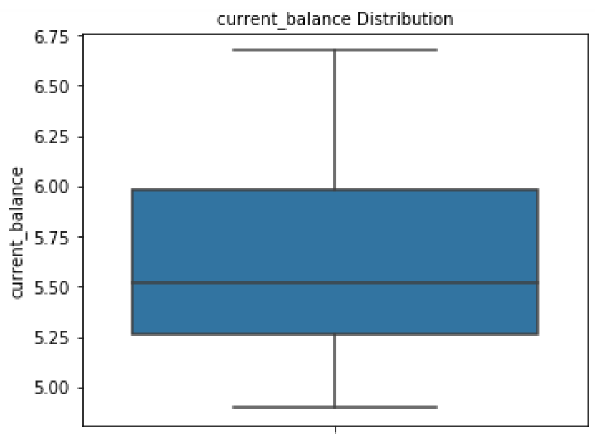
The box plot of the probability of full payment variable shows few outliers.

Probability of full payment is negatively skewed - -0.537954

The dist plot shows the distribution of data from 0.80 to 0.92.

The Probability values is good above 80%

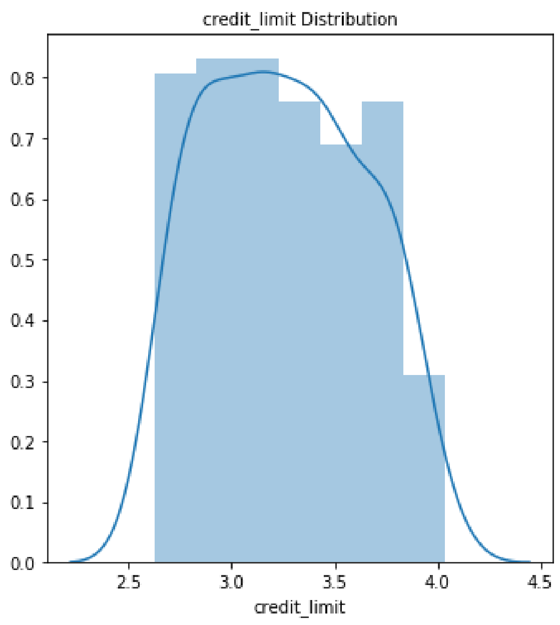


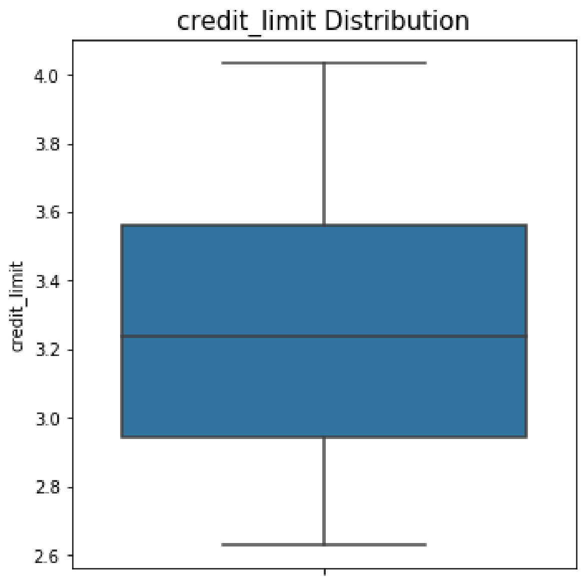


The box plot of the current balance variable shows no outliers.

Current balance is positively skewed - 0.525482

The dist plot shows the distribution of data from 5.0 to 6.5.

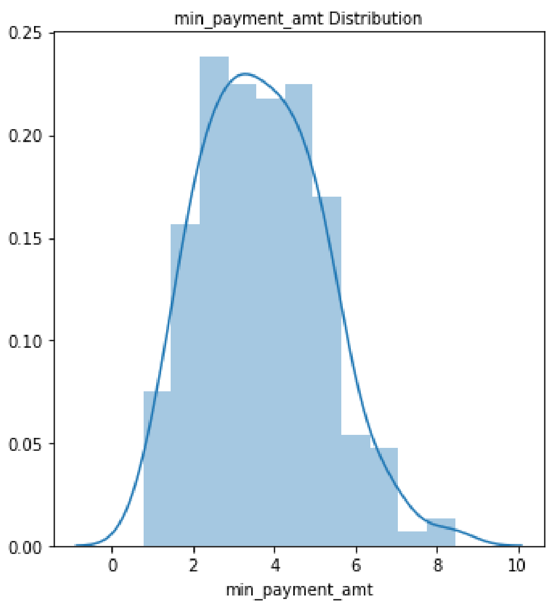


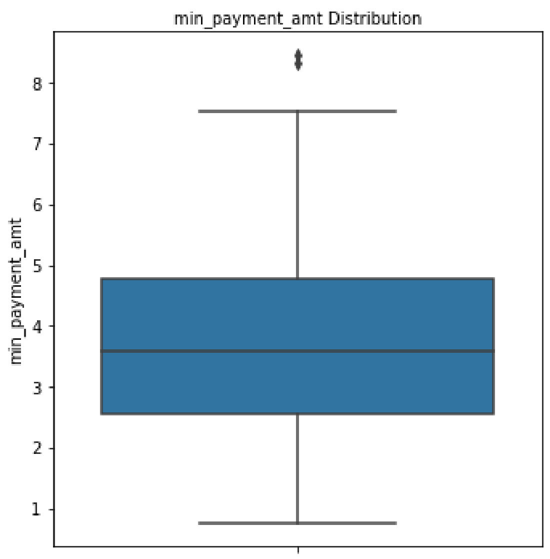


The box plot of the credit limit variable shows no outliers.

Credit limit is positively skewed - 0.134378

The dist plot shows the distribution of data from 2.5 to 4.0

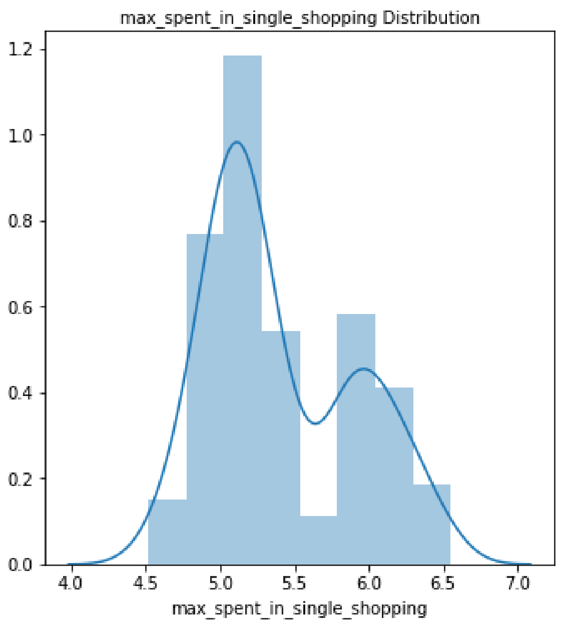


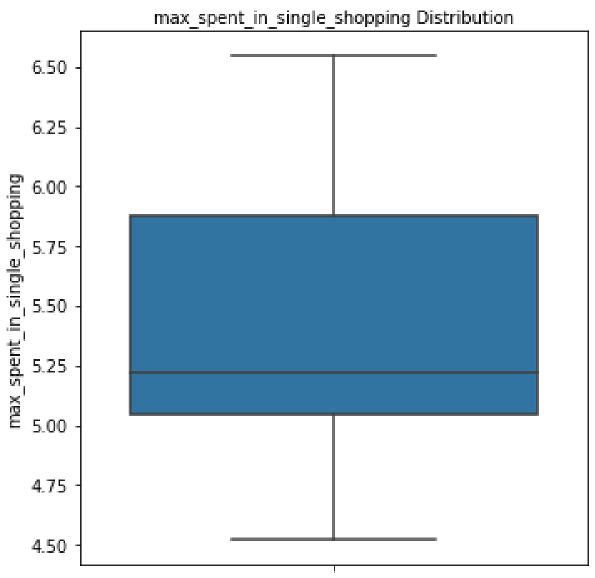


The box plot of the min payment amount variable shows few outliers.

Min payment amount is positively skewed - 0.401667

The dist plot shows the distribution of data from 2 to 8





The box plot of the max spent in single shopping variable shows no outliers.

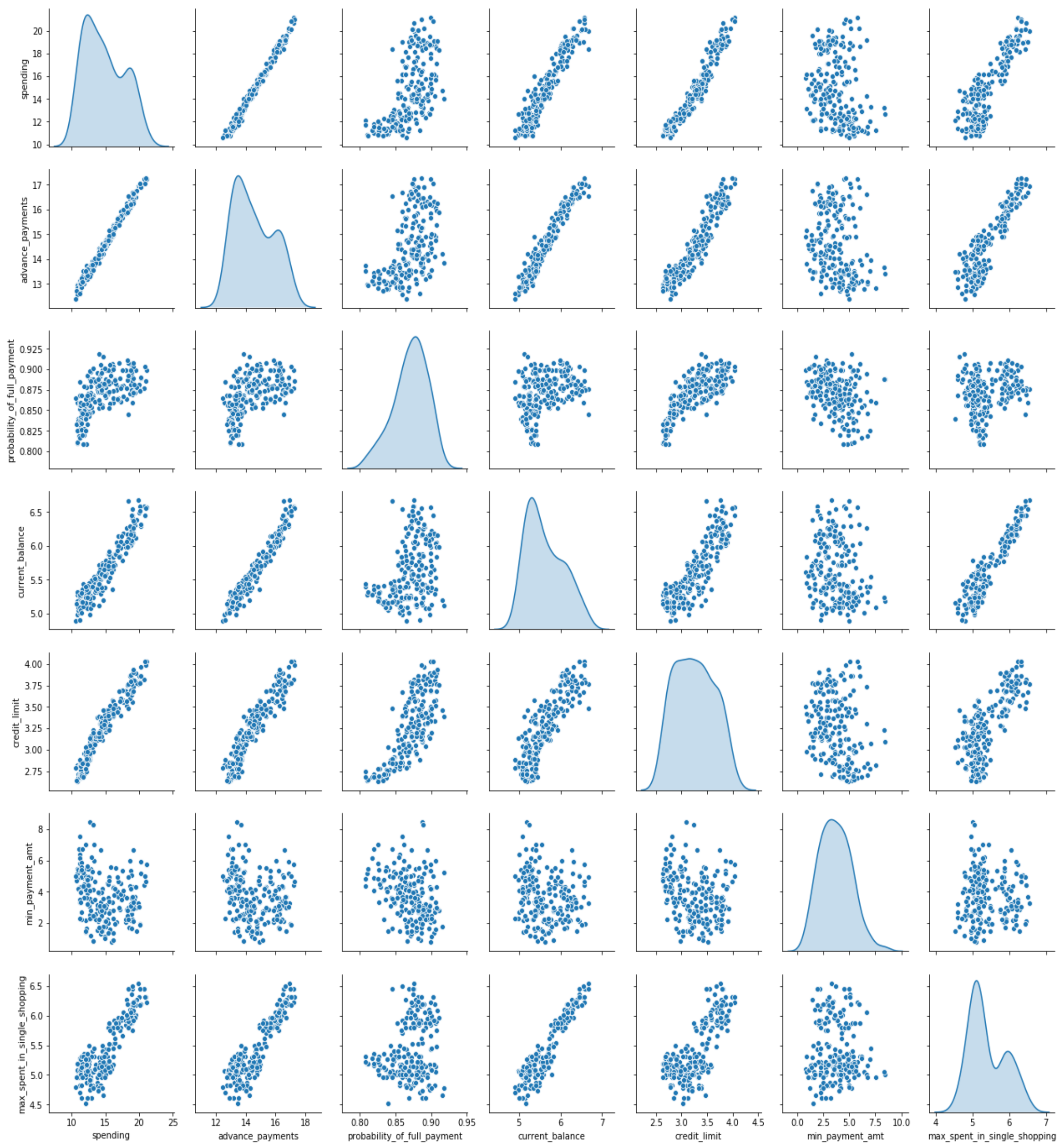
Max spent in single shopping is positively skewed - 0.561897

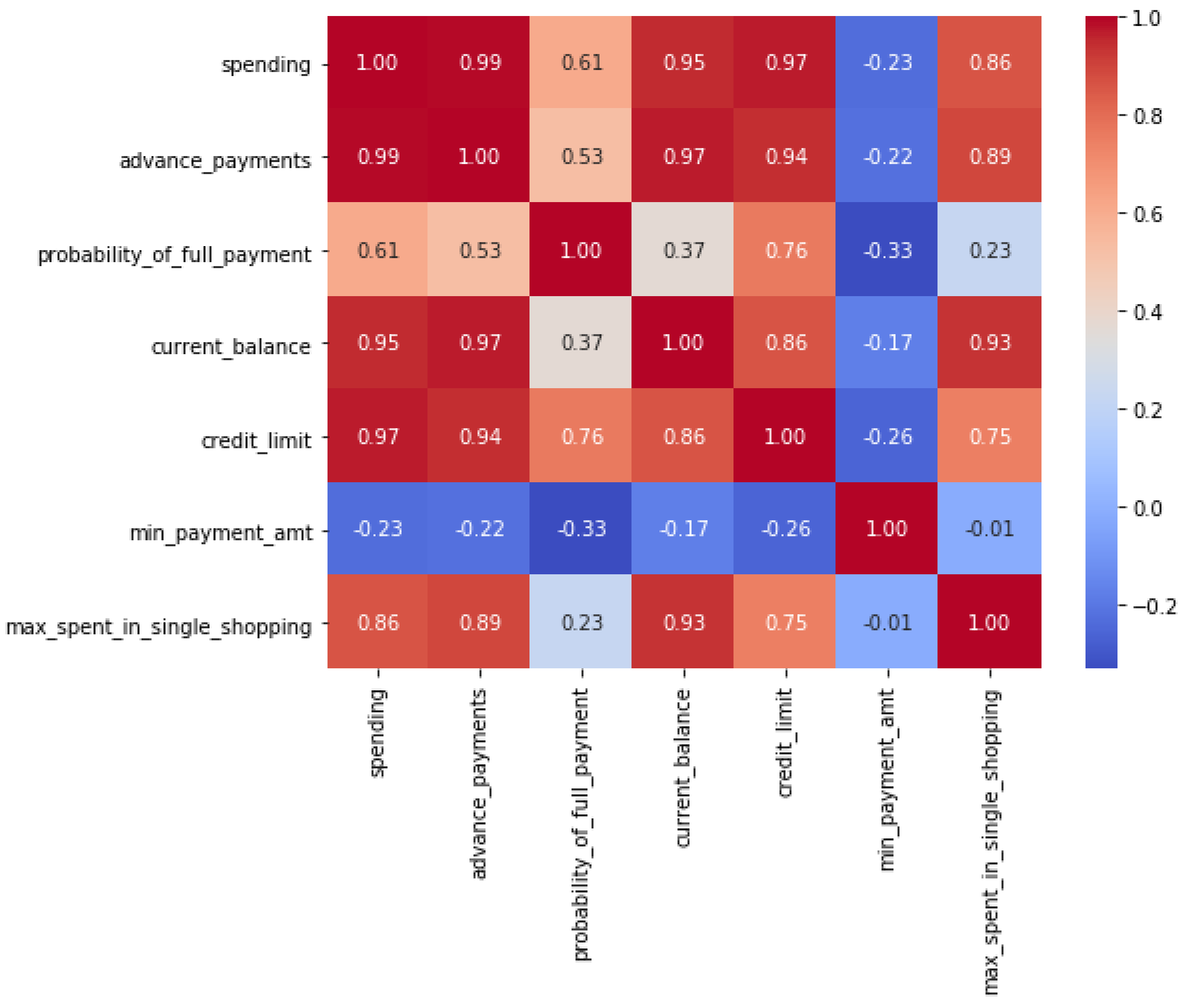
The dist plot shows the distribution of data from 4.5 to 6.5

No outlier treatment – only 3 to 4 values re observed has outlier we are treating them

Multivariate analysis

Check for multicollinearity





Heatmap for Better Visualization

Observations

Strong positive correlation

Between - spending & advance payments,

-advance payments & current balance,

- Credit limit & spending

- Spending & current balance

- credit limit & advance payments

- Max_spent_in_single_shopping current balance

1.2 Do you think scaling is necessary for clustering in this case? Justify

Yes, scaling is very important as the model works based on the distance based computations scaling is necessary for unscaled data.

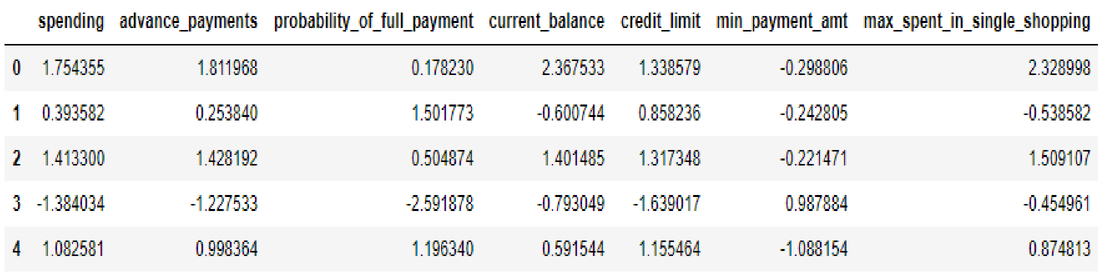
Scaling needs to be done as the values of the variables are in different scales.

Spending, advance payments are in different values and this may get more weightage.

Scaling will have all the values in the relative same range.

I have used standard scalar for scaling

Below is the snapshot of scaled data**.**



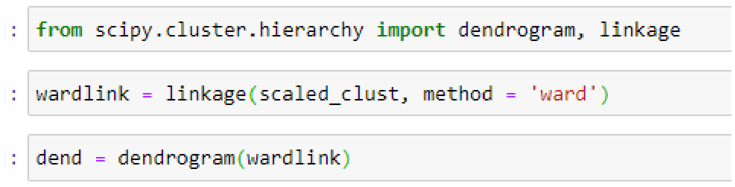
	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	1.754355	1.811968	0.178230	2.367533	1.338579	-0.298806	2.328998
1	0.393582	0.253840	1.501773	-0.600744	0.858236	-0.242805	-0.538582
2	1.413300	1.428192	0.504874	1.401485	1.317348	-0.221471	1.509107
3	-1.384034	-1.227533	-2.591878	-0.793049	-1.639017	0.987884	-0.454961
4	1.082581	0.998364	1.196340	0.591544	1.155464	-1.088154	0.874813

1.3 Apply hierarchical clustering to scaled data. Identify the number of optimum clusters using Dendrogram and briefly describe them

Hierarchical clustering – ward's method & average method

By choosing ward's method to the scaled data,

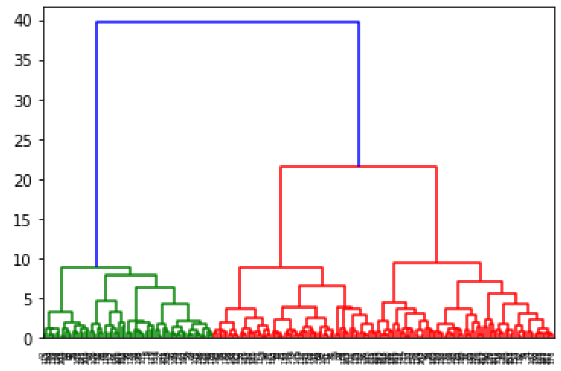
For visualization purposes I have used to Dendrogram



```
: from scipy.cluster.hierarchy import dendrogram, linkage
```

```
: wardlink = linkage(scaled_clust, method = 'ward')
```

```
: dend = dendrogram(wardlink)
```

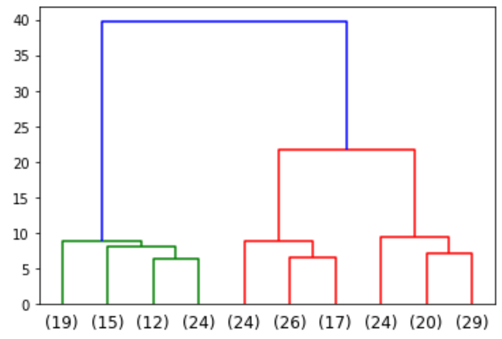


The above dendrogram indicates all the data points have clustered to different clusters by wards method.

To find the optimal number cluster through which we can solve our business objective we use truncate mode = lastp.

Wherein we can give last p = 10 according to industry set base value.

```
dend = dendrogram(wardlink,
                  truncate_mode='lastp',
                  p = 10,
                  )
```



Now, we can understand all the data points have clustered into 3 clusters.

Next to map these clusters to our dataset we can use fclusters

Criterion we can give "maxclust"

```
]: clusters_ward = fcluster(wardlink, 3, criterion='maxclust')
```

```
]: clusters_ward
```

```
]: array([1, 3, 1, 2, 1, 2, 2, 3, 1, 2, 1, 3, 2, 1, 3, 2, 3, 2, 3, 2, 2, 2,
          1, 2, 3, 1, 3, 2, 2, 2, 3, 2, 2, 3, 2, 2, 2, 2, 2, 1, 1, 3, 1, 1,
          2, 2, 3, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 2, 2, 1, 3, 2, 2, 3, 3, 1,
          1, 3, 1, 2, 3, 2, 1, 1, 2, 1, 3, 2, 1, 3, 3, 3, 3, 1, 2, 3, 3, 1,
          1, 2, 3, 1, 3, 2, 2, 1, 1, 1, 2, 1, 2, 1, 3, 1, 3, 1, 1, 2, 2, 1,
          3, 3, 1, 2, 2, 1, 3, 3, 2, 1, 3, 2, 2, 2, 3, 3, 1, 2, 3, 3, 2, 3,
          3, 1, 2, 1, 1, 2, 1, 3, 3, 3, 2, 2, 3, 2, 1, 2, 3, 2, 3, 2, 3, 3,
          3, 3, 3, 2, 3, 1, 1, 2, 1, 1, 1, 2, 1, 3, 3, 3, 3, 2, 3, 1, 1, 1,
          3, 3, 1, 2, 3, 3, 3, 3, 1, 1, 3, 3, 3, 2, 3, 3, 2, 1, 3, 1, 1, 2,
          1, 2, 3, 1, 3, 2, 1, 3, 1, 3, 1, 3], dtype=int32)
```

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	clusters_ward
0	19.94	16.92	0.8752	6.675	3.763	3.252	6.550	1
1	15.99	14.89	0.9064	5.363	3.582	3.336	5.144	3
2	18.95	16.42	0.8829	6.248	3.755	3.368	6.148	1
3	10.83	12.96	0.8099	5.278	2.641	5.182	5.185	2
4	17.99	15.86	0.8992	5.890	3.694	2.068	5.837	1

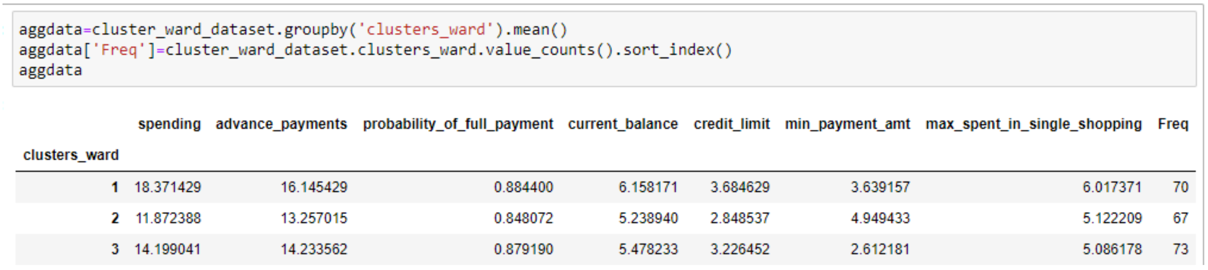
Now, we can look at the cluster frequency in our dataset,

Cluster Frequency

```
]: cluster_ward_dataset.clusters_ward.value_counts().sort_index()
```

```
]: 1    70
   2    67
   3    73
   Name: clusters_ward, dtype: int64
```

Cluster profiling to understand the business problem.



```
aggdata=cluster_ward_dataset.groupby('clusters_ward').mean()
aggdata['Freq']=cluster_ward_dataset.clusters_ward.value_counts().sort_index()
aggdata
```

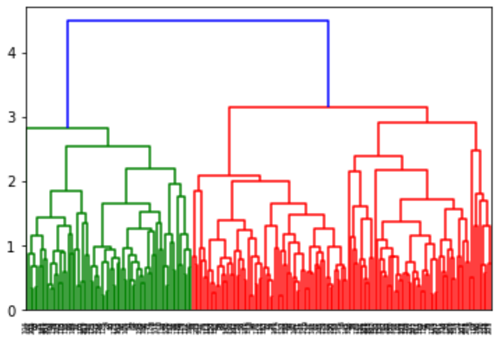
clusters_ward	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Freq
1	18.371429	16.145429	0.884400	6.158171	3.684629	3.639157	6.017371	70
2	11.872388	13.257015	0.848072	5.238940	2.848537	4.949433	5.122209	67
3	14.199041	14.233562	0.879190	5.478233	3.226452	2.612181	5.086178	73

By choosing average method to the scaled data,

Choosing ward linkage method

```
[39]: wardlink_1 = linkage(scaled_clust, method = 'average')
```

```
[40]: dend_1 = dendrogram(wardlink_1)
```

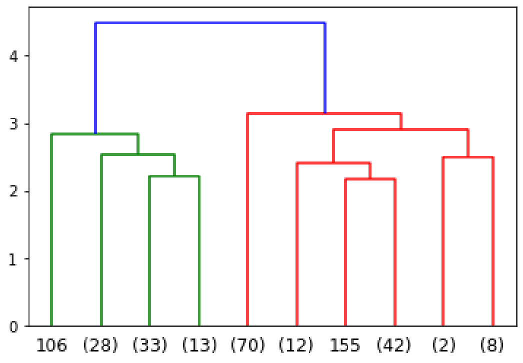


The above dendrogram indicates all the data points have clustered to different clusters by average method.

To find the optimal number cluster through which we can solve our business objective we use truncate mode = lastp.

Wherein we can give last p = 10 according to industry set base value.

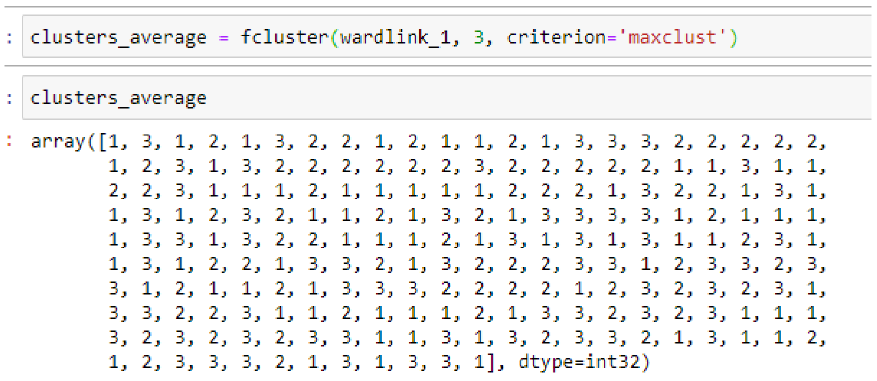
```
]: dend_1= dendrogram(wardlink_1,
                      truncate_mode='lastp',
                      p = 10,
                      )
```



Now, we can understand all the data points have clustered into 3 clusters.

Next to map these clusters to our dataset we can use fclusters

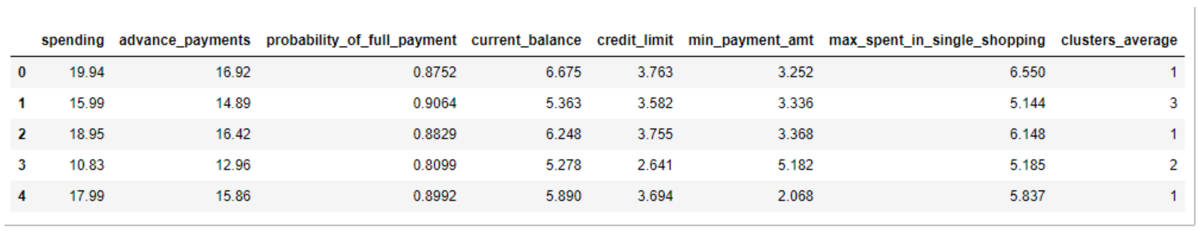
Criterion we can give “maxclust”



```
: clusters_average = fcluster(wardlink_1, 3, criterion='maxclust')
```

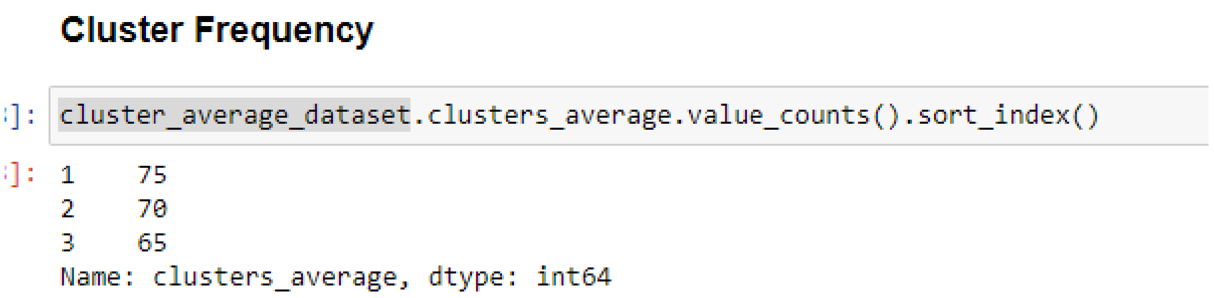
```
: clusters_average
```

```
: array([1, 3, 1, 2, 1, 3, 2, 2, 1, 2, 1, 1, 2, 1, 3, 3, 3, 2, 2, 2, 2, 2,
         1, 2, 3, 1, 3, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 1, 1, 3, 1, 1,
         2, 2, 3, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 2, 2, 1, 3, 2, 2, 1, 3, 1,
         1, 3, 1, 2, 3, 2, 1, 1, 2, 1, 3, 2, 1, 3, 3, 3, 3, 1, 2, 1, 1, 1,
         1, 3, 3, 1, 3, 2, 2, 1, 1, 1, 2, 1, 3, 1, 3, 1, 3, 1, 1, 2, 3, 1,
         1, 3, 1, 2, 2, 1, 3, 3, 2, 1, 3, 2, 2, 2, 3, 3, 1, 2, 3, 3, 2, 3,
         3, 1, 2, 1, 1, 2, 1, 3, 3, 3, 2, 2, 2, 2, 1, 2, 3, 2, 3, 2, 3, 1,
         3, 3, 2, 2, 3, 1, 1, 2, 1, 1, 1, 2, 1, 3, 3, 2, 3, 2, 3, 1, 1, 1,
         3, 2, 3, 2, 3, 2, 3, 3, 1, 1, 3, 1, 3, 2, 3, 3, 2, 1, 3, 1, 1, 2,
         1, 2, 3, 3, 3, 2, 1, 3, 1, 3, 3, 1], dtype=int32)
```



	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	clusters_average
0	19.94	16.92	0.8752	6.675	3.763	3.252	6.550	1
1	15.99	14.89	0.9064	5.363	3.582	3.336	5.144	3
2	18.95	16.42	0.8829	6.248	3.755	3.368	6.148	1
3	10.83	12.96	0.8099	5.278	2.641	5.182	5.185	2
4	17.99	15.86	0.8992	5.890	3.694	2.068	5.837	1

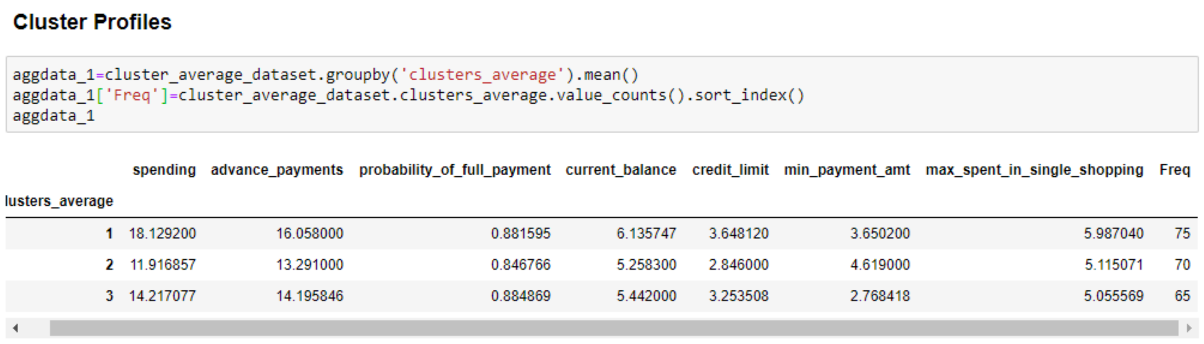
Now, we can look at the cluster frequency in our dataset,



Cluster Frequency

```
]: cluster_average_dataset.clusters_average.value_counts().sort_index()
```

```
]: 1    75
   2    70
   3    65
   Name: clusters_average, dtype: int64
```



Cluster Profiles

```
aggdata_1=cluster_average_dataset.groupby('clusters_average').mean()
aggdata_1['Freq']=cluster_average_dataset.clusters_average.value_counts().sort_index()
aggdata_1
```

lusters_average	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Freq
1	18.129200	16.058000	0.881595	6.135747	3.648120	3.650200	5.987040	75
2	11.916857	13.291000	0.846766	5.258300	2.846000	4.619000	5.115071	70
3	14.217077	14.195846	0.884869	5.442000	3.253508	2.768418	5.055569	65

Observation

Both the method are almost similar means, minor variation, which we know it occurs.
There was not too much variations from both methods

Cluster grouping based on the dendrogram, 3 or 4 looks good. Did the further analysis, and based on the dataset had gone for 3 group cluster

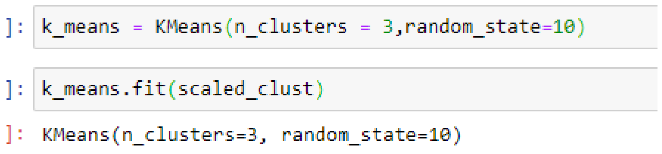
And three group cluster solution gives a pattern based on high/medium/low spending with max_spent_in_single_shopping (high value item) and probability_of_full_payment (payment made).

1.4 Apply K-Means clustering on scaled data and determine optimum clusters. Apply elbow curve and silhouette score.

K-means clustering,

Randomly we decide to give n_clusters = 3 and we look at the distribution of clusters according to the n_clusters.

We apply K-means technique to the scaled data.

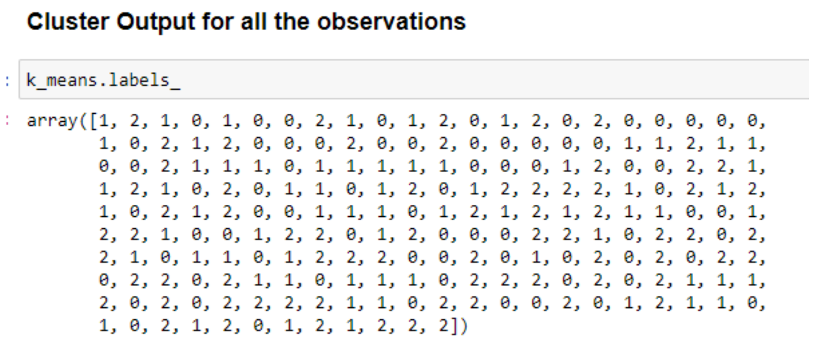


```
]: k_means = KMeans(n_clusters = 3,random_state=10)

]: k_means.fit(scaled_clust)

]: KMeans(n_clusters=3, random_state=10)
```

Cluster output for all the observations in the dataset,



Cluster Output for all the observations

```
: k_means.labels_

: array([1, 2, 1, 0, 1, 0, 0, 2, 1, 0, 1, 2, 0, 1, 2, 0, 2, 0, 0, 0, 0, 0,
         1, 0, 2, 1, 2, 0, 0, 0, 2, 0, 0, 2, 0, 0, 0, 0, 0, 1, 1, 2, 1, 1,
         0, 0, 2, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 2, 0, 0, 2, 2, 1,
         1, 2, 1, 0, 2, 0, 1, 1, 0, 1, 2, 0, 1, 2, 2, 2, 2, 1, 0, 2, 1, 2,
         1, 0, 2, 1, 2, 0, 0, 1, 1, 1, 0, 1, 2, 1, 2, 1, 2, 1, 1, 0, 0, 1,
         2, 2, 1, 0, 0, 1, 2, 2, 0, 1, 2, 0, 0, 0, 2, 2, 1, 0, 2, 2, 0, 2,
         2, 1, 0, 1, 1, 0, 1, 2, 2, 2, 0, 0, 2, 0, 1, 0, 2, 0, 2, 0, 2, 2,
         0, 2, 2, 0, 2, 1, 1, 0, 1, 1, 1, 0, 2, 2, 2, 0, 2, 0, 2, 1, 1, 1,
         2, 0, 2, 0, 2, 2, 2, 2, 1, 1, 0, 2, 2, 0, 0, 2, 0, 1, 2, 1, 1, 0,
         1, 0, 2, 1, 2, 0, 1, 2, 1, 2, 2, 2])
```

We have 3 clusters 0,1,2

To find the optimal number of clusters, we can use k-elbow method

Calculating WSS for other values of K - Elbow Method

```
wss =[]
```

```
for i in range(1,11):
    KM = KMeans(n_clusters=i,random_state=1)
    KM.fit(scaled_clust)
    wss.append(KM.inertia_)
```

```
wss
```

```
[1469.9999999999998,
 659.171754487041,
 430.6589731513006,
 371.38509060801096,
 327.21278165661346,
 289.31599538959495,
 262.98186570162267,
 241.81894656086033,
 223.91254221002725,
 206.39612184786694]
```

To find the inertia value for all the clusters from 1 to 11, I used a for loop to find the optimal number of clusters.

The silhouette score for 3 clusters is good

```
k_means = KMeans(n_clusters = 3,random_state=1)
k_means.fit(scaled_clust)
labels = k_means.labels_
```

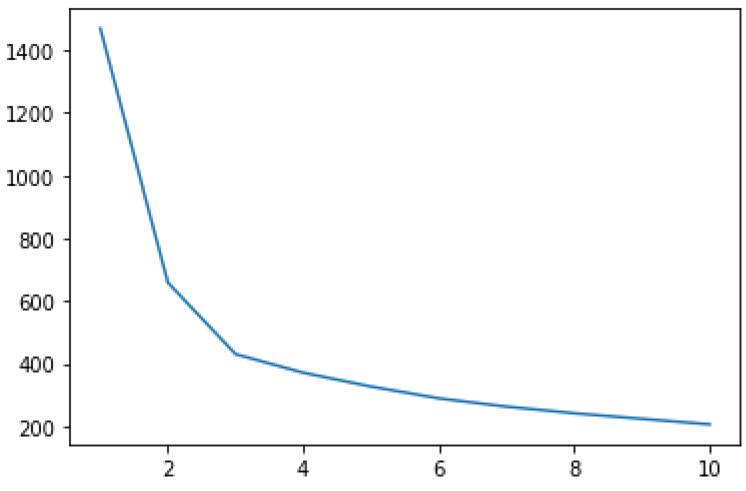
```
silhouette_score(scaled_clust,labels,random_state=1)
```

```
0.4007270552751299
```

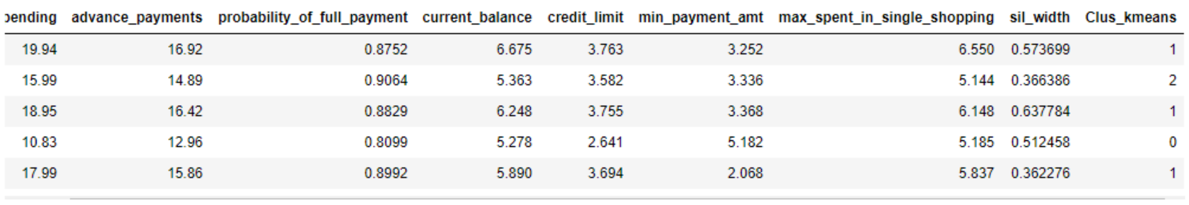
The elbow curve seen here also shows us after 3 clusters there is no huge drop in the values, so we select 3 clusters.

```
plt.plot(range(1,11), wss)
```

```
[<matplotlib.lines.Line2D at 0x1df0fbc2908>]
```



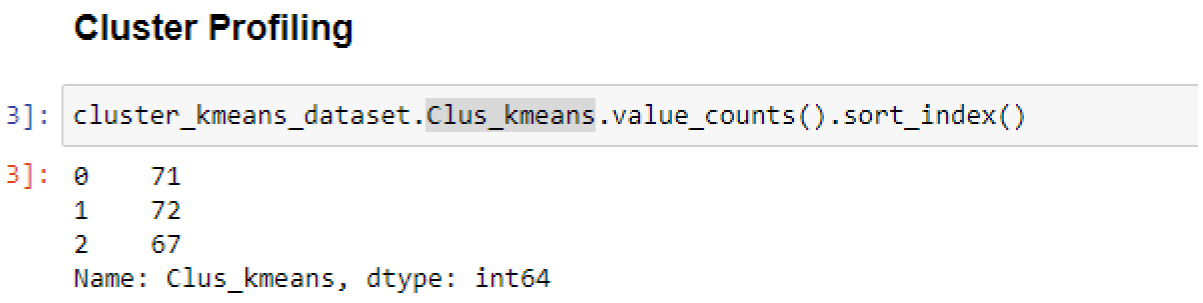
So adding the cluster results to our dataset to solve our business objective.



ɔending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	sil_width	Clus_kmeans
19.94	16.92	0.8752	6.675	3.763	3.252	6.550	0.573699	1
15.99	14.89	0.9064	5.363	3.582	3.336	5.144	0.366386	2
18.95	16.42	0.8829	6.248	3.755	3.368	6.148	0.637784	1
10.83	12.96	0.8099	5.278	2.641	5.182	5.185	0.512458	0
17.99	15.86	0.8992	5.890	3.694	2.068	5.837	0.362276	1

This table shows the clusters to the dataset and also individual sil_width score.

Cluster frequency

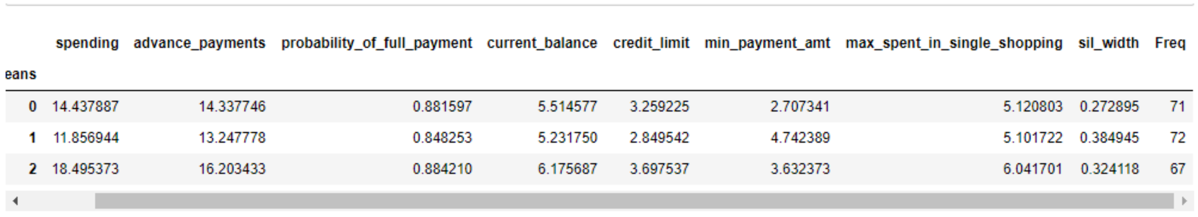


Cluster Profiling

```
3]: cluster_kmeans_dataset.Clus_kmeans.value_counts().sort_index()
```

```
3]: 0    71
    1    72
    2    67
    Name: Clus_kmeans, dtype: int64
```

This frequency shows frequency of clusters to the dataset.



eans	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	sil_width	Freq
0	14.437887	14.337746	0.881597	5.514577	3.259225	2.707341	5.120803	0.272895	71
1	11.856944	13.247778	0.848253	5.231750	2.849542	4.742389	5.101722	0.384945	72
2	18.495373	16.203433	0.884210	6.175687	3.697537	3.632373	6.041701	0.324118	67

3-Group clusters via K- Means has equal split of percentage of results.

Cluster 0 – Medium

Cluster 1 – low

Cluster 2 – High

Observation

By K- Mean's method we can at cluster 3 we find it optimal after there is no huge drop in inertia values. Also the elbow curve seems to show similar results.

The silhouette width score of the K – means also seems to very less value that indicates all the data points are properly clustered to the cluster. There is no mismatch in the data points with regards to clustering

Cluster grouping based on the dendrogram, 3 or 4 looks good. Did the further analysis, and based on the dataset had gone for 3 group cluster

And three group cluster solution gives a pattern based on high/medium/low spending with max_spent_in_single_shopping (high value item) and probability_of_full_payment (payment made).

1.5 Describe cluster profiles for the clusters defined. Recommend different promotional strategies for different clusters.

Group 1: High Spending Group –
Giving any reward points might increase their purchases. –
Maximum max_spent_in_single_shopping is high for this group, so can be offered discount/offer on next transactions upon full payment –
Increase their credit limit and –
Increase spending habits –
Give loan against the credit card, as they are customers with good repayment record. –
Tie up with luxury brands, which will drive more one_time_maximun spending

Group 2: Low Spending Group - customers should be given remainders for payments. Offers can be provided on early payments to improve their payment rate. - Increase their spending habits by tying up with grocery stores, utilities (electricity, phone, gas, others)

Group 3: Medium Spending Group - They are potential target customers who are paying bills and doing purchases and maintaining comparatively good credit score. So we can increase credit limit or can lower down interest rate. - Promote premium cards/loyalty cars to increase transactions. - Increase spending habits by trying with premium ecommerce sites, travel portal, travel airlines/hotel, as this will encourage them to spend more