Problem 1: Clustering

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, do the necessary initial steps, and exploratory data analysis (Univariate, Bi-variate, and multivariate analysis).

Table 1: Head of the dataset showing the first 5 records

	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 dataset was loaded and the head of the dataset was checked. Table 1 shows the first 5 records of the dataset. From this table, we can see the different variables or columns of the dataset.

Table 2: Information of the dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210 entries, 0 to 209
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   spending                      210 non-null    float64
 1   advance_payments              210 non-null    float64
 2   probability_of_full_payment   210 non-null    float64
 3   current_balance               210 non-null    float64
 4   credit_limit                  210 non-null    float64
 5   min_payment_amt               210 non-null    float64
 6   max_spent_in_single_shopping  210 non-null    float64
dtypes: float64(7)
memory usage: 11.6 KB
```

The dataset has 7 columns and 210 records which is seen in Table 2. There are 210 non-null records in all the 7 columns meaning there are no missing records based on this initial analysis that was done.

Table 3: Data type of the columns in the dataset

```
spending                      float64
advance_payments              float64
probability_of_full_payment   float64
current_balance               float64
credit_limit                  float64
min_payment_amt               float64
max_spent_in_single_shopping  float64
dtype: object
```

The data type is ‘float64’ indicating that the variables are numeric in type which is seen in Table 1 and Table 2.

The shape of the data is (210, 7) meaning the dataset has 210 rows and 7 columns.

Table 4: Missing value of the columns in the dataset

```
spending                      0
advance_payments              0
probability_of_full_payment   0
current_balance               0
credit_limit                  0
min_payment_amt               0
max_spent_in_single_shopping  0
dtype: int64
```

The dataset was further checked for missing values and it is seen from Table 4 that there are no missing values in the dataset.

Table 5: Description of the dataset

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
count	210.000000	210.000000	210.000000	210.000000	210.000000	210.000000	210.000000
mean	14.847524	14.559286	0.870999	5.628533	3.258605	3.700201	5.408071
std	2.909699	1.305959	0.023629	0.443063	0.377714	1.503557	0.491480
min	10.590000	12.410000	0.808100	4.899000	2.630000	0.765100	4.519000
25%	12.270000	13.450000	0.856900	5.262250	2.944000	2.561500	5.045000
50%	14.355000	14.320000	0.873450	5.523500	3.237000	3.599000	5.223000
75%	17.305000	15.715000	0.887775	5.979750	3.561750	4.768750	5.877000
max	21.180000	17.250000	0.918300	6.675000	4.033000	8.456000	6.550000

Table 5 shows the description or the summary of the dataset. It can be seen that there are 7 different columns in this dataframe and all of them have 210 values. By looking at Table 5, we are able to deduce that ‘spending’ has the highest mean value while ‘probability_of_full_payment’ has the lowest mean value. ‘spending’ has the highest standard deviation value while ‘probability_of_full_payment’ has the lowest standard deviation value. This is probably because some customers spend less and some customers spend more. It is up to the individual about how much to spend. Hence the standard deviation is high. The mean and median values are approximately equal in all variables.

Univariate Analysis:

Table 6: Skewness of the variables of the dataset

```
max_spent_in_single_shopping    0.561897
current_balance                 0.525482
min_payment_amt                 0.401667
spending                        0.399889
advance_payments                0.386573
credit_limit                    0.134378
probability_of_full_payment    -0.537954
dtype: float64
```

1. **Spending:**

Table 7: Description of 'spending'

```
Description of spending
-----------------------------------------------------------
count    210.000000
mean      14.847524
std        2.909699
min       10.590000
25%       12.270000
50%       14.355000
75%       17.305000
max       21.180000
Name: spending, dtype: float64 Distribution of spending
```

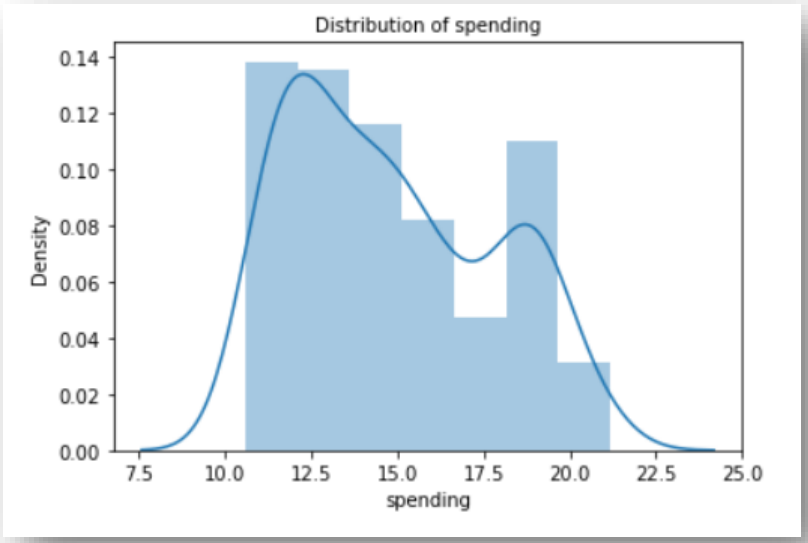

Figure 1: Univariate distribution of 'spending'

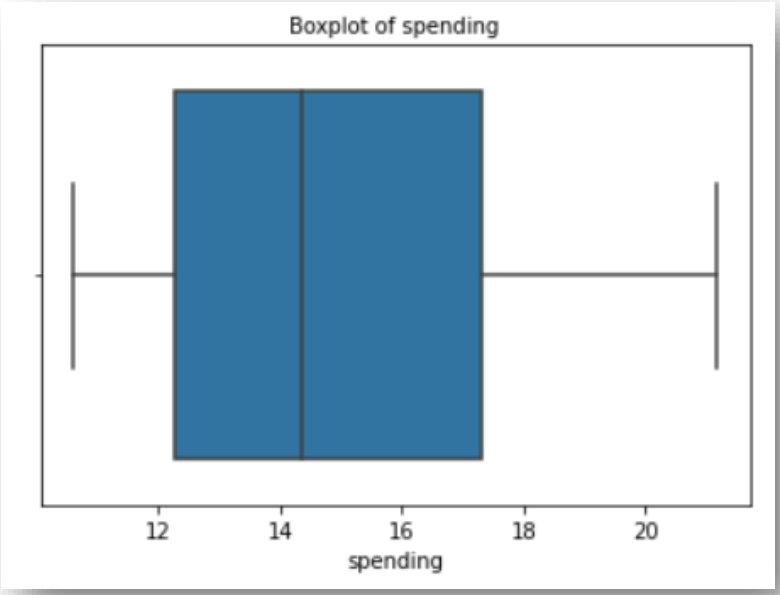

Figure 2: Boxplot showing the distribution of 'spending'

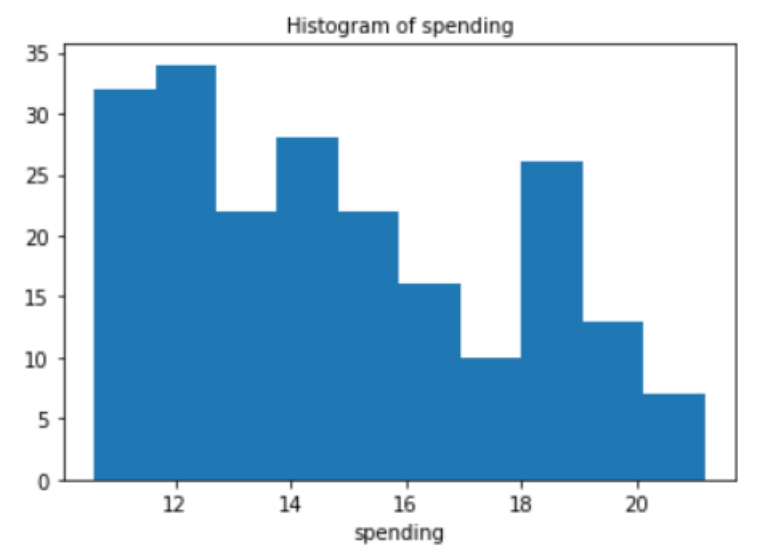

Figure 3: Histogram of 'spending'

Univariate analysis of ‘spending’ is done to understand the patterns and distribution of the data. From Figure 2, we can see that the Box plot of ‘spending’ variable has no outliers. However the distribution of the data is moderately right skewed which is seen in Figure 1. This is also seen in Table 6 where the skewness values are given. The skewness value of ‘spending’ variable is 0.399899. From Table 7, it is seen that the mean of the data is 14.85 meaning the amount spent by the customer per month on average is 14850 (data is in 1000s). The maximum amount spent is 21.18 meaning the maximum amount spent by the customer per month is 21180 (data is in 1000s). The distribution plot shows the dist of data from 10 to 22.

2. **advance_payments:**

Table 8: Description of 'advance_payments'

```
Description of advance_payments
-------------------------------------------------------------------------------
count    210.000000
mean      14.559286
std        1.305959
min       12.410000
25%       13.450000
50%       14.320000
75%       15.715000
max       17.250000
Name: advance_payments, dtype: float64 Distribution of advance_payments
```

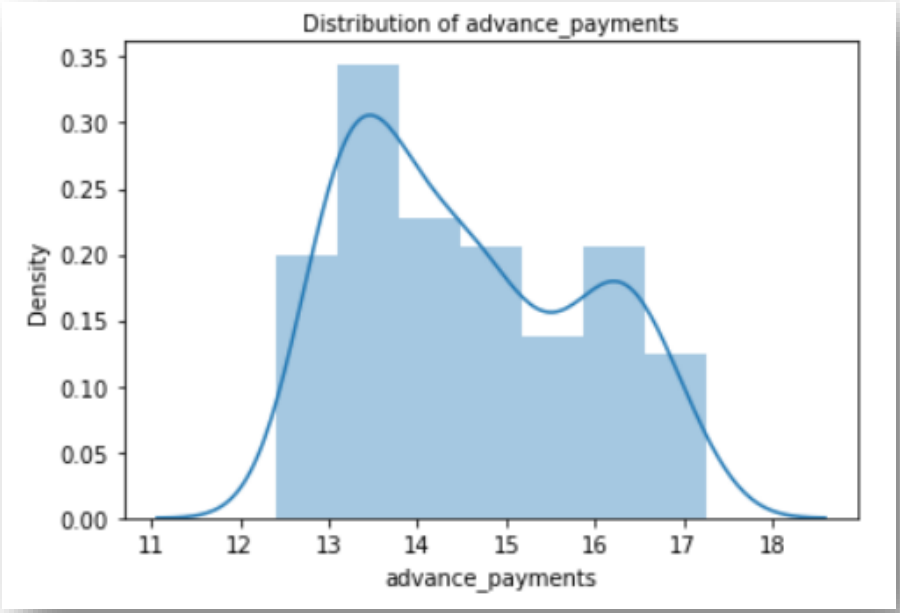

Figure 4: Univariate distribution of 'advance_payments'

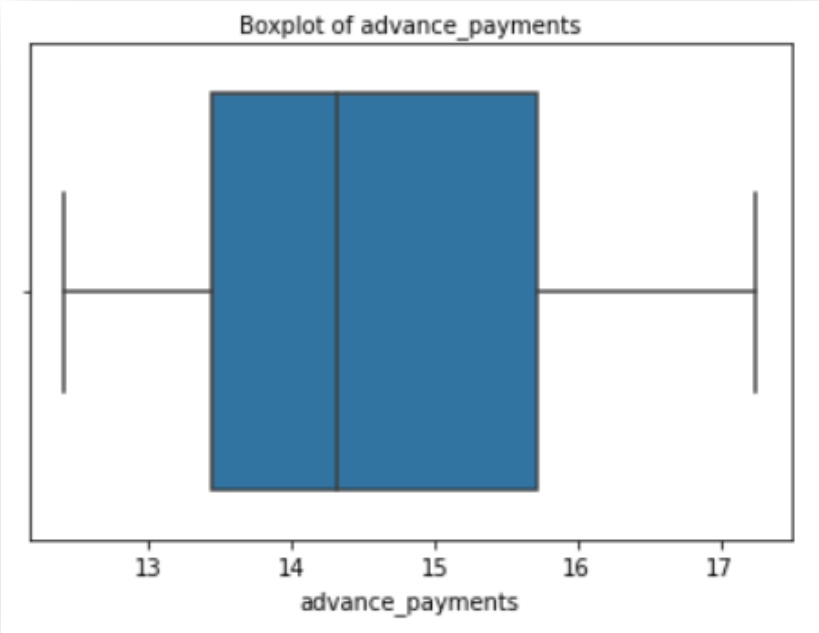

Figure 5: Boxplot showing the distribution of 'advance_payments'

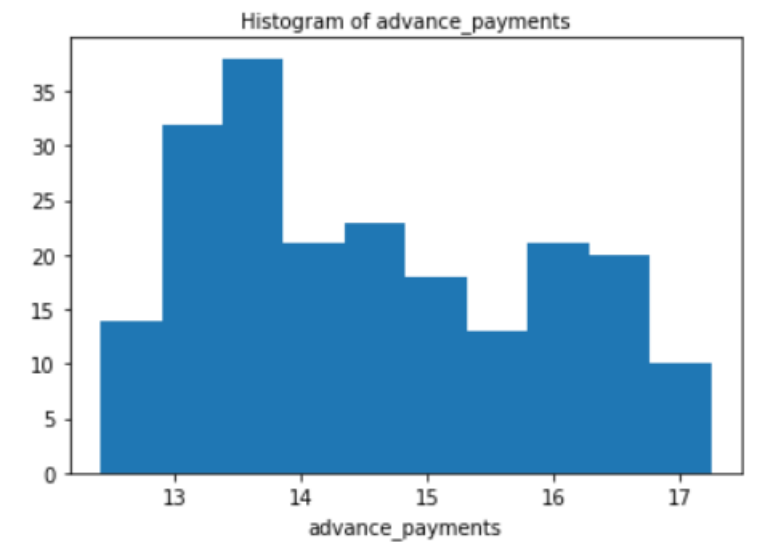

Figure 6: Histogram of 'advance_payments'

Univariate analysis of 'advance_payments' is done to understand the patterns and distribution of the data. From Figure 5, we can see that the Box plot of 'advance_payments' variable has no outliers. However the distribution of the data is moderately right skewed which is seen in Figure 4. This is also seen in Table 6 where the skewness values are given. The skewness value of 'advance_payments' variable is 0.386573. From Table 8, it is seen that the mean of the data is 14.55 meaning the amount paid by the customer in advance by cash is 1455 on average (data is in 100s). The maximum amount paid by the customer in advance by cash 17.25 (1725 as data is in 100s). The distribution plot shows the dist of data from 12 to 17.

3. probability_of_full_payment:

Table 9: Description of 'probability_of_full_payment'

```
Description of probability_of_full_payment
--------------------------------------------------------------------------------
count    210.000000
mean       0.870999
std        0.023629
min        0.808100
25%        0.856900
50%        0.873450
75%        0.887775
max        0.918300
Name: probability_of_full_payment, dtype: float64 Distribution of probability_of_full_payment
```

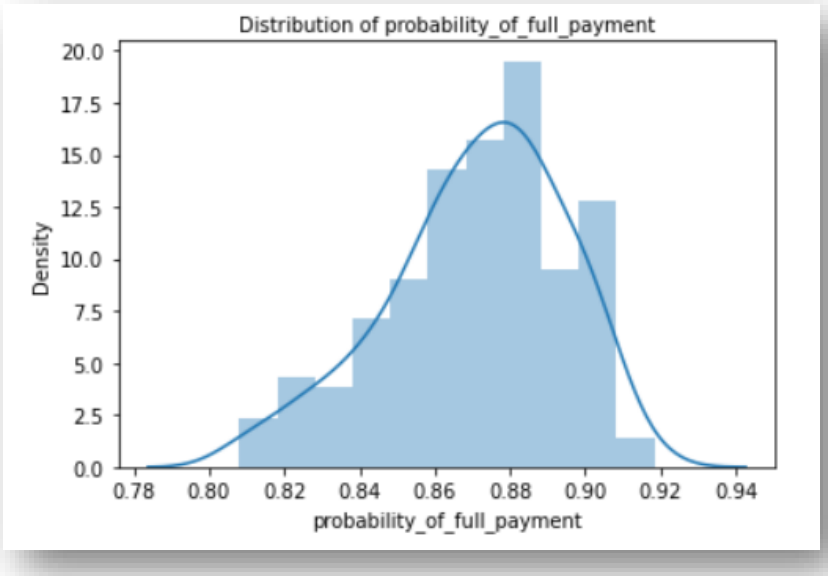

Figure 7: Univariate distribution of 'probability_of_full_payment'

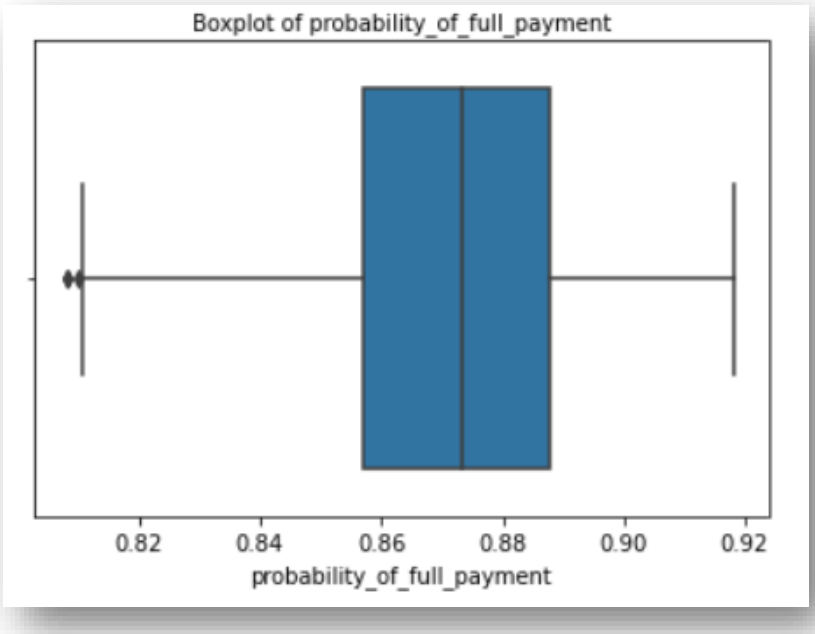

Figure 8: Boxplot showing the distribution of 'probability_of_full_payment'

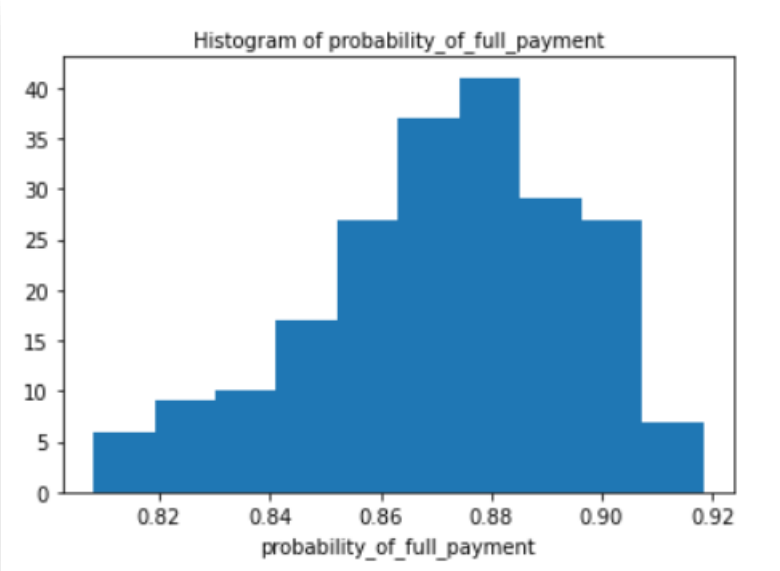

Figure 9: Histogram of 'probability_of_full_payment'

Univariate analysis of 'probability_of_full_payment' is done to understand the patterns and distribution of the data. From Figure 8, we can see that the Box plot of 'probability_of_full_payment' variable has outliers. The distribution of the data is moderately left skewed which is seen in Figure 7. This is also seen in Table 6 where the skewness values are given. The skewness value of 'probability_of_full_payment' variable is -0.537954. From Table 9, it is seen that the mean of the data is 0.87 meaning the probability of payment done in full by the customer to the bank is 0.87 on average. The distribution plot shows the dist of data from 0.80 to 0.92.

4. **current_balance:**

Table 10: Description of 'current_balance'

```
Description of current_balance
-------------------------------------------------------------------------------
count    210.000000
mean       5.628533
std        0.443063
min        4.899000
25%        5.262250
50%        5.523500
75%        5.979750
max        6.675000
Name: current_balance, dtype: float64 Distribution of current_balance
```

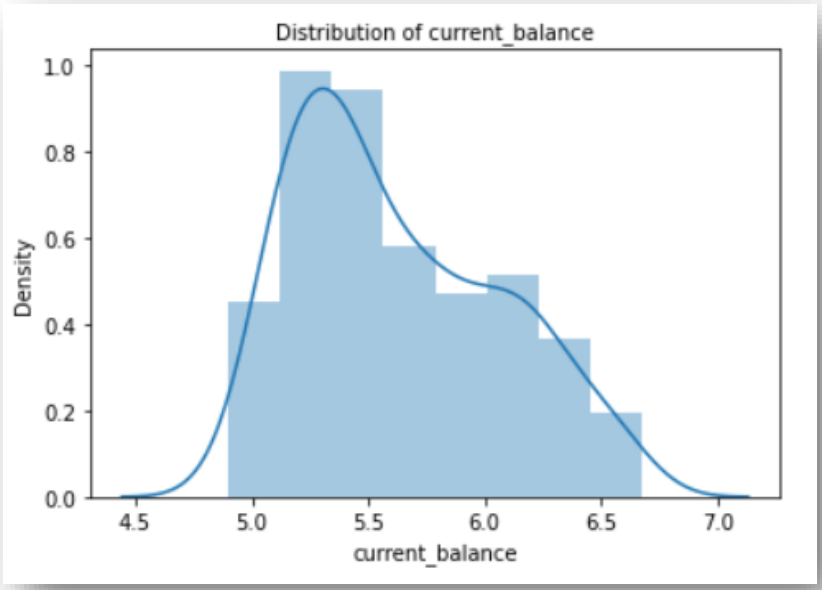

Figure 10: Univariate distribution of 'current_balance'

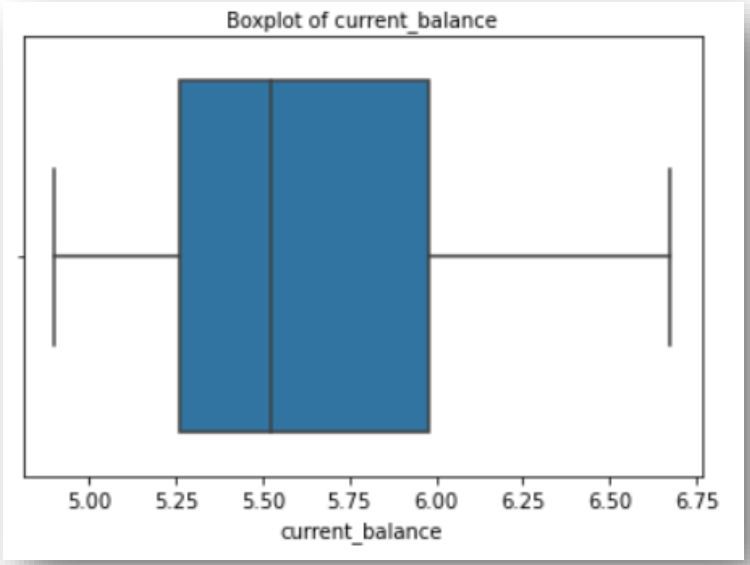

Figure 11: Boxplot showing the distribution of 'current_balance'

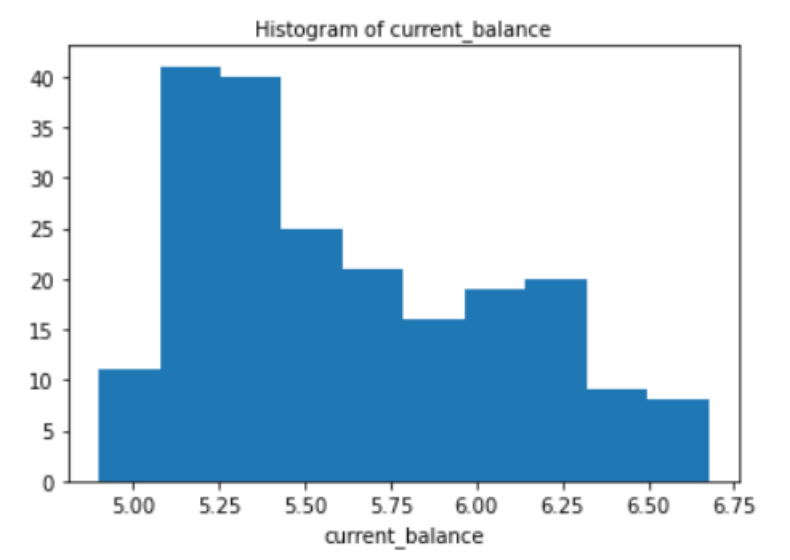

Figure 12: Histogram of 'current_balance'

Univariate analysis of ‘current_balance’ is done to understand the patterns and distribution of the data. From Figure 11, we can see that the Box plot of ‘current_balance’ variable has no outliers. However the distribution of the data is moderately right skewed which is seen in Figure 10. This is also seen in Table 6 where the skewness values are given. The skewness value of ‘current_balance’ variable is 0.525482. From Table 10, it is seen that the mean of the data is 5.62 meaning the balance amount left in the account to make purchases is 5620 on average (data is in 1000s). The dist plot shows the distribution of data from 5.0 to 6.5.

5. **credit_limit:**

Table 11: Description of 'credit_limit'

```
Description of credit_limit
----------------------------------------------------------------
count    210.000000
mean       3.258605
std        0.377714
min        2.630000
25%        2.944000
50%        3.237000
75%        3.561750
max        4.033000
Name: credit_limit, dtype: float64 Distribution of credit_limit
```

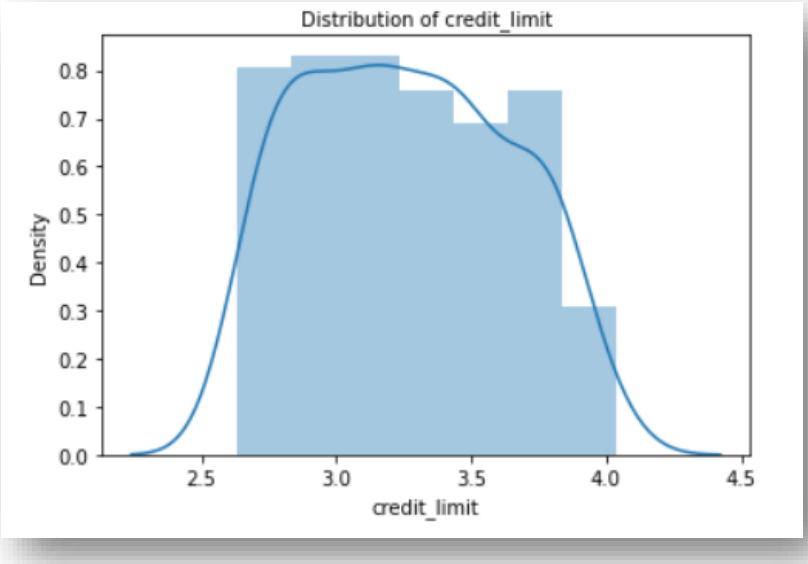

Figure 13: Univariate distribution of 'credit_limit'

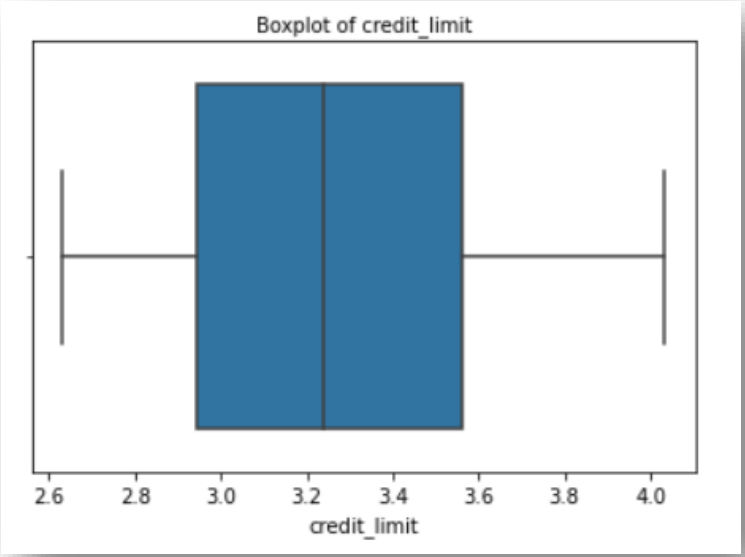

Figure 14: Boxplot showing the distribution of 'credit_limit'

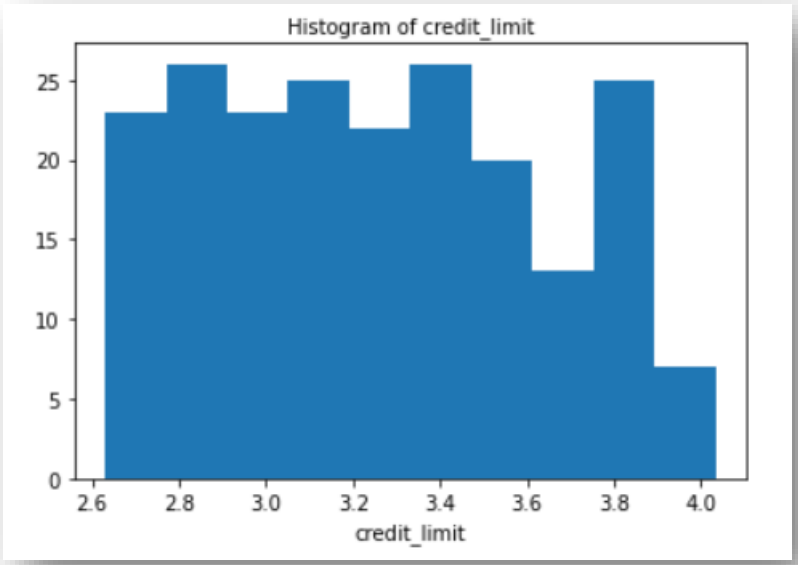

Figure 15: Histogram of 'credit_limit'

Univariate analysis of 'credit_limit' is done to understand the patterns and distribution of the data. From Figure 14, we can see that the Box plot of 'credit_limit' variable has no outliers. However the distribution of the data is moderately right skewed which is seen in Figure 13. This is also seen in Table 6 where the skewness values are given. The skewness value of 'credit_limit' variable is 0.134378. From Table 11, it is seen that the mean of the data is 3.26 meaning the limit of the amount in credit card is 32600 on average (data is in 10000s). The maximum credit limit is 4.03 meaning the limit of the amount in credit card is 40300 (data is in 10000s). The dist plot shows the distribution of data from 2.5 to 4.0.

6. **min_payment_amt:**

Table 12: Description of 'min_payment_amt'

```
Description of min_payment_amt
-------------------------------------------------------------------------
count    210.000000
mean       3.700201
std        1.503557
min        0.765100
25%        2.561500
50%        3.599000
75%        4.768750
max        8.456000
Name: min_payment_amt, dtype: float64 Distribution of min_payment_amt
```

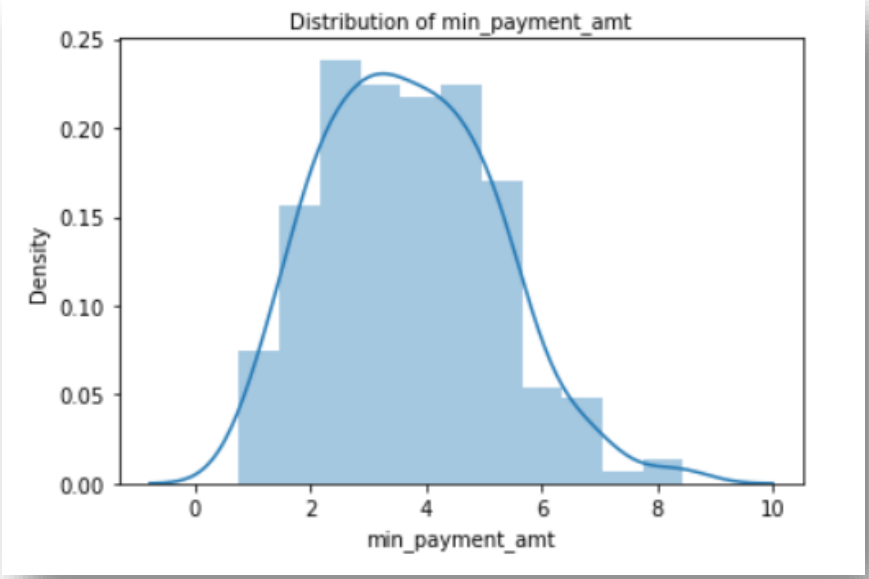

Figure 16: Univariate distribution of 'min_payment_amt'

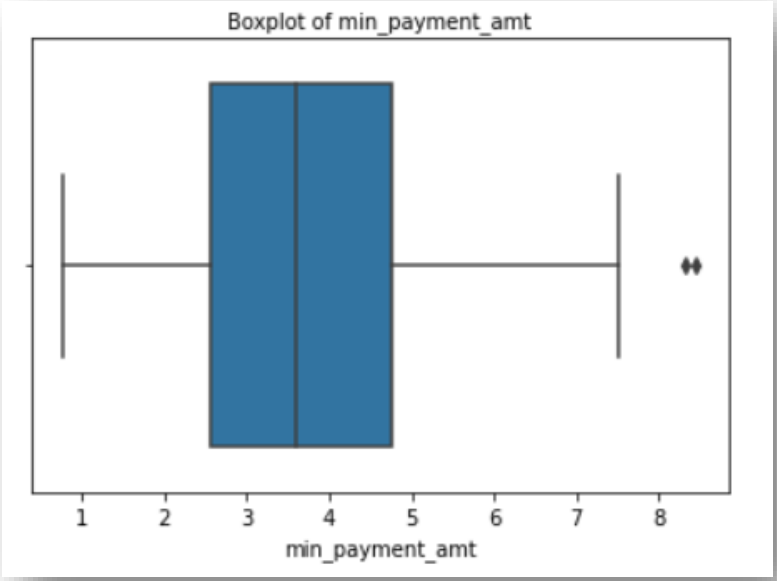

Figure 17: Boxplot showing the distribution of 'min_payment_amt'

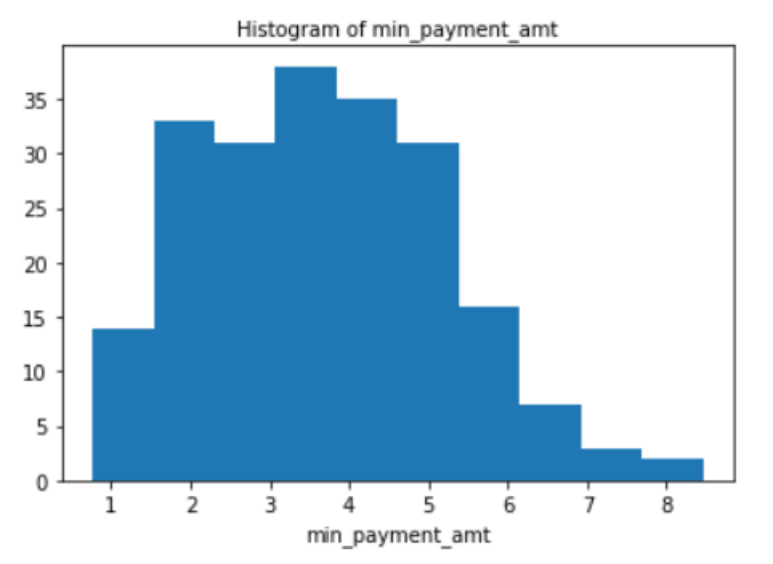

Figure 18: Histogram of 'min_payment_amt'

Univariate analysis of ‘min_payment_amt’ is done to understand the patterns and distribution of the data. From Figure 17, we can see that the Box plot of ‘min_payment_amt’ variable has outliers. The distribution of the data is moderately right skewed which is seen in Figure 16. This is also seen in Table 6 where the skewness values are given. The skewness value of ‘min_payment_amt’ variable is 0.401667. From Table 12, it is seen that the mean of the data is 3.7 meaning the minimum paid by the customer while making payments for purchases made monthly is 370 on average (data is in 100s). The dist plot shows the distribution of data from 2 to 8.

7. 'max_spent_in_single_shopping':

Table 13: Description of 'max_spent_in_single_shopping'

```
Description of max_spent_in_single_shopping
-------------------------------------------------------------------------------
count    210.000000
mean       5.408071
std        0.491480
min        4.519000
25%        5.045000
50%        5.223000
75%        5.877000
max        6.550000
Name: max_spent_in_single_shopping, dtype: float64 Distribution of max_spent_in_single_shopping
```

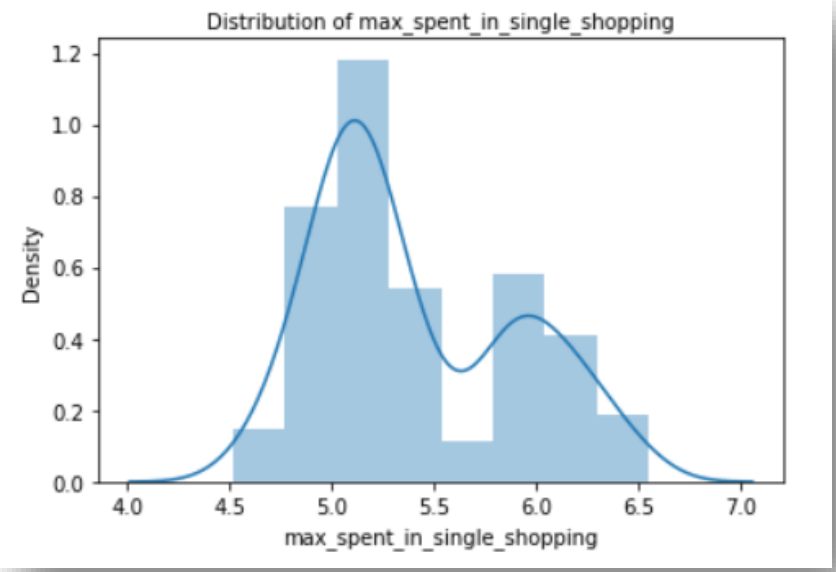

Figure 19: Univariate distribution of 'max_spent_in_single_shopping'

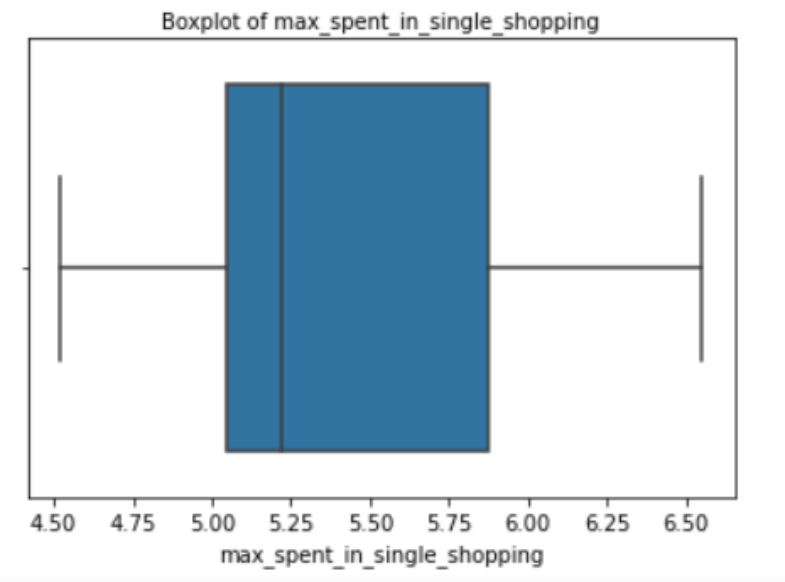

Figure 20: Boxplot showing the distribution of 'max_spent_in_single_shopping'

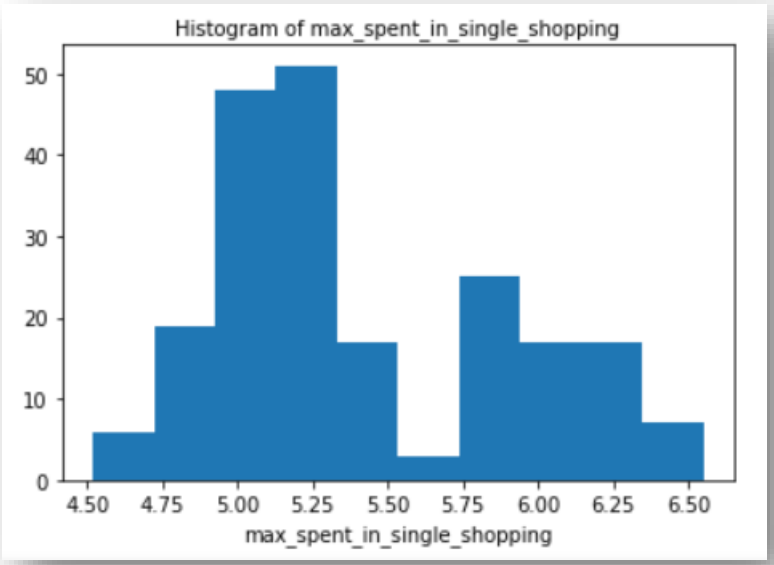

Figure 21: Histogram of 'max_spent_in_single_shopping'

Univariate analysis of ‘max_spent_in_single_shopping’ is done to understand the patterns and distribution of the data. From Figure 20, we can see that the Box plot of ‘max_spent_in_single_shopping’ variable has no outliers. The distribution of the data is moderately right skewed which is seen in Figure 19. This is also seen in Table 6 where the skewness values are given. The skewness value of ‘max_spent_in_single_shopping’ variable is 0.561897. From Table 13, it is seen that the mean of the data is 5.41 meaning the maximum amount spent in one purchase is 5410 on average (data is in 1000s). The dist plot shows the distribution of data from 4.5 to 6.5.

Bivariate Analysis:

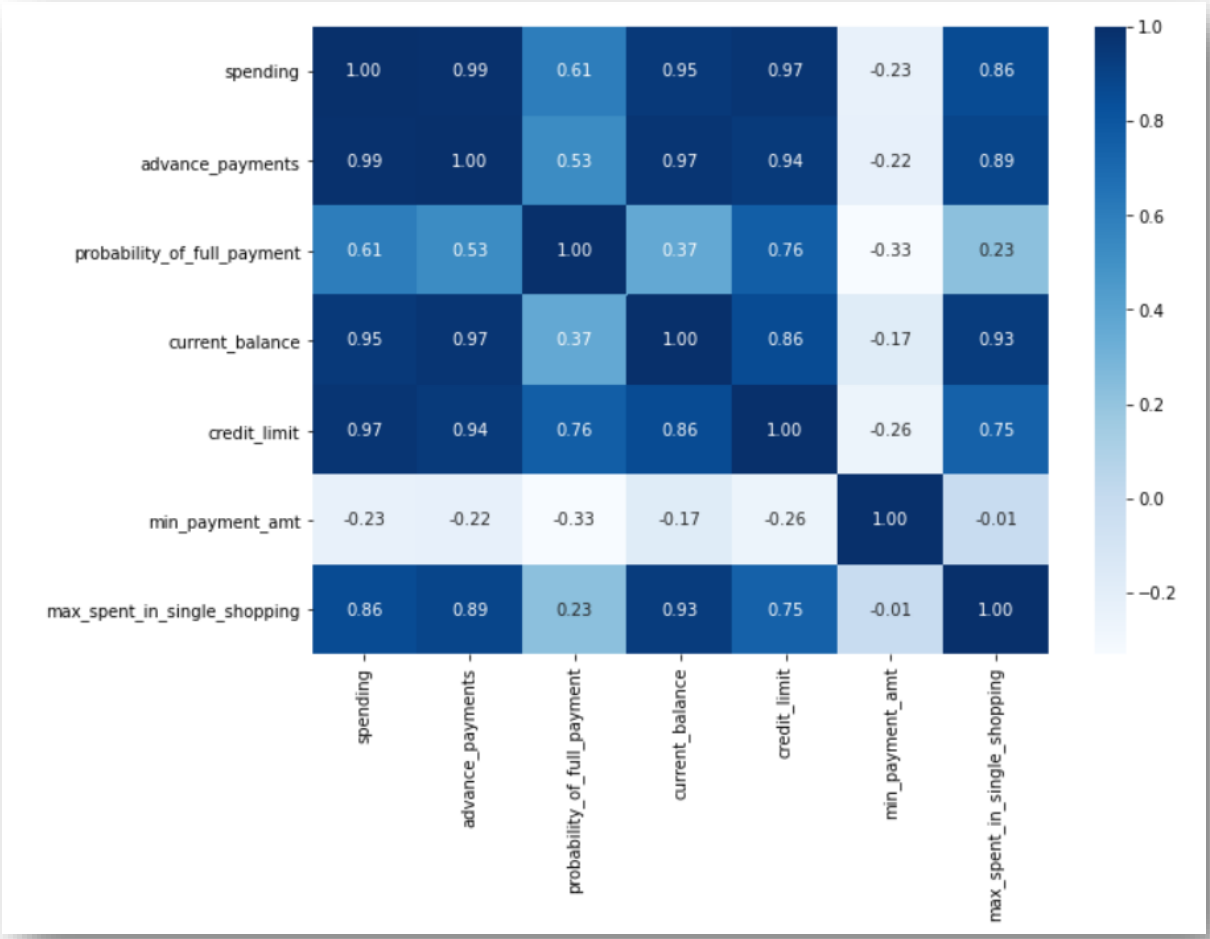

Figure 22: Heat map showing the bivariate analysis of the dataset

Bivariate analysis is done using the help of a heat map. A heat map is used to understand the correlation between two numerical values in a dataset. Figure 22 shows the heat map of the dataset.

Multivariate Analysis:

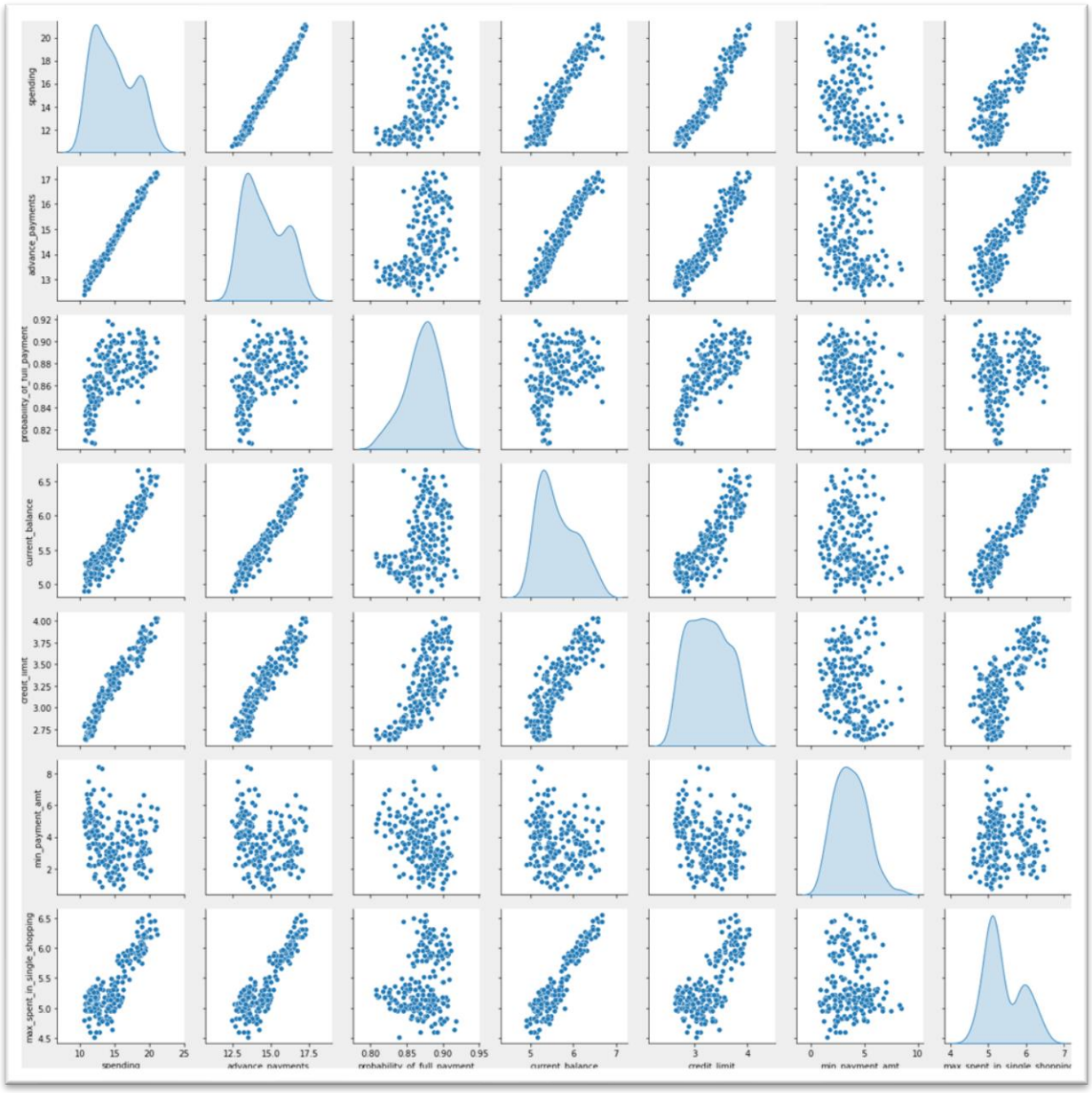

Figure 23: Pair plot showing the multivariate analysis of the dataset

Multivariate analysis is done using the help of a pair plot to understand the relationship between all the numerical values in the dataset. Pair plot can be used to compare all the variables with each other to understand the patterns or trends in the dataset. Figure 23 shows the pair plot of the dataset.

Observations:

From Figure 23 we can observe that there:

Strong positive correlation between:

- spending & advance_payments

- spending & current balance
- advance_payments & current_balance
- current balance and max spent
- credit_limit & spending
- spending & current_balance
- credit_limit & advance_payments
- max_spent_in_single_shopping current_balance

All variables have negative correlation with min-payment amount

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

Scaling is a way of representing a dataset. Scaling needs to be done as a dataset has features or variables with different 'weights' for each feature. In such cases, it is suggested to transform the features so that all the features are in the same 'scale'. This is called scaling. Scaling also makes it easier to compare the features now that the weightage on each feature is the same.

Scaling can be done by Z-score method and Min-max method. Z-score method is used when you want to centralize the data (weight-based). Example: principal component analysis (PCA), neural network etc., Min-max method is used when the data is distance based. Example: Clustering, KNN etc., Models have to be implemented only after scaling is done.

In the given dataset, there are 7 numerical variables. Scaling is done on all the 7 numerical variables. Among the 7 numerical variables, spending, advance_payments, current_balance, credit_limit, min_payment_amt and max_spent_in_single_shopping are the variables in which the values are associated with money and probability_of_full_payment is the variable in which the values are given as probabilities. Spending, advance payments are way higher than the other values of the dataset. Therefore, the dataset has to be scaled. Only then all the variables can be compared and weighed equally.

Min-max method is used for scaling this dataset as this problem is based on clustering. The scaled data will range between 0 and 1.

$$\text{Min-max scaling} = \frac{(X - Xmin)}{(Xmax - Xmin)}$$

Standard scaler method is also used to scale the data.

Table 14: Description of the dataset before scaling

	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

Table 15: Description of the dataset after scaling using Min-max method

	count	mean	std	min	25%	50%	75%	max
spending	210.0	0.402032	0.274759	0.0	0.158640	0.355524	0.634089	1.0
advance_payments	210.0	0.444067	0.269826	0.0	0.214876	0.394628	0.682851	1.0
probability_of_full_payment	210.0	0.570767	0.214423	0.0	0.442831	0.593013	0.723004	1.0
current_balance	210.0	0.410773	0.249473	0.0	0.204533	0.351633	0.608530	1.0
credit_limit	210.0	0.448043	0.269219	0.0	0.223806	0.432644	0.664113	1.0
min_payment_amt	210.0	0.381633	0.195498	0.0	0.233575	0.368474	0.520570	1.0
max_spent_in_single_shopping	210.0	0.437751	0.241989	0.0	0.258986	0.346627	0.668636	1.0

Table 16: Description of the dataset after scaling using Standard scaler method

	count	mean	std	min	25%	50%	75%	max
spending	210.0	9.148766e-16	1.002389	-1.466714	-0.887955	-0.169674	0.846599	2.181534
advance_payments	210.0	1.097006e-16	1.002389	-1.649686	-0.851433	-0.183664	0.887069	2.065260
probability_of_full_payment	210.0	1.243978e-15	1.002389	-2.668236	-0.598079	0.103993	0.711677	2.006586
current_balance	210.0	-1.089076e-16	1.002389	-1.650501	-0.828682	-0.237628	0.794595	2.367533
credit_limit	210.0	-2.994298e-16	1.002389	-1.668209	-0.834907	-0.057335	0.804496	2.055112
min_payment_amt	210.0	5.302637e-16	1.002389	-1.956769	-0.759148	-0.067469	0.712379	3.170590
max_spent_in_single_shopping	210.0	-1.935489e-15	1.002389	-1.813288	-0.740495	-0.377459	0.956394	2.328998

The data scaled by standard scaler method is used for further analysis.

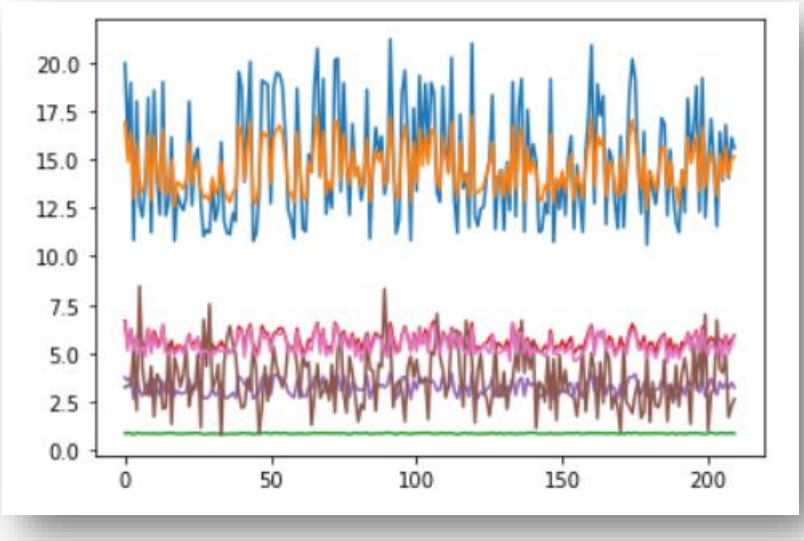

Figure 24: Plot of the dataset before scaling

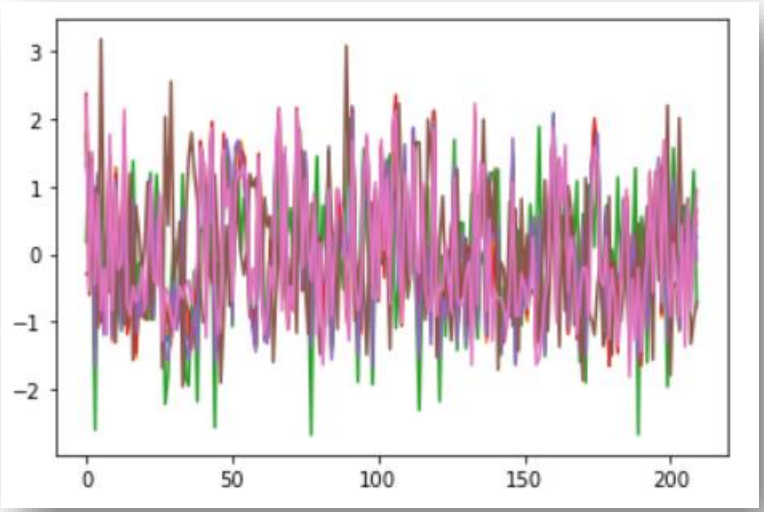

Figure 25: Plot of the dataset after scaling using Standard scaler method

From Table 14, Table 15 and Table 16, we are able to see how the values have changed in scale. This is also seen in Figure 24 and Figure 25. The values may seem to be different but however they are only scaled. The dataset is brought into one unit of comparison. To prove this, a histogram is plotted.

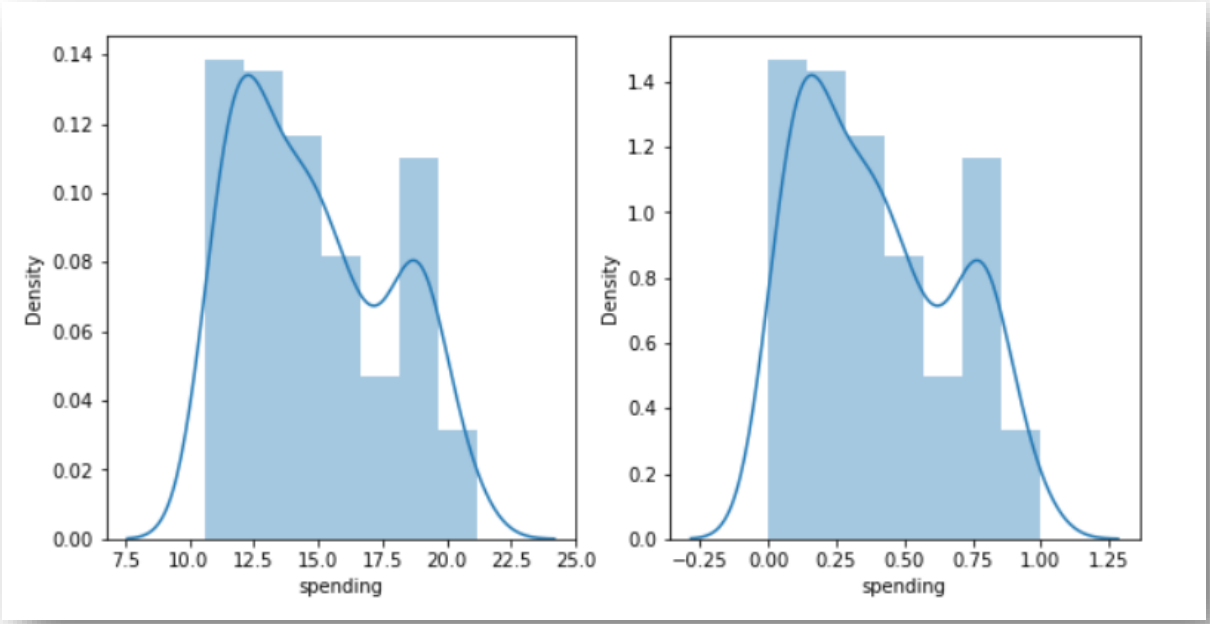

Figure 26: Graph showing the histogram of 'spending' variable before scaling (left) and the histogram of 'spending' variable after scaling by Standard scaler method(right)

From Figure 26, we can see that the plot of the 'spending' variable before and after scaling is the same but the scale has changed. This is the case for all the other numerical variables that has been scaled using the min-max method.

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

Hierarchical clustering

There are 2 methods to perform hierarchical clustering.

1. **Average method:**

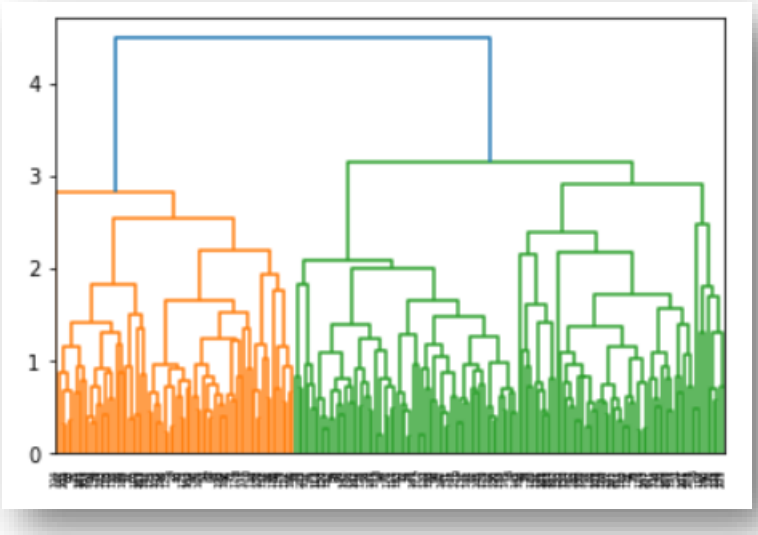

Figure 27: Dendrogram

Figure 27 shows the dendrogram that is got as a result of hierarchical clustering. However there are too many entries and hence it is not clearly visible. Therefore truncation is done using truncate_mode to make it readable.

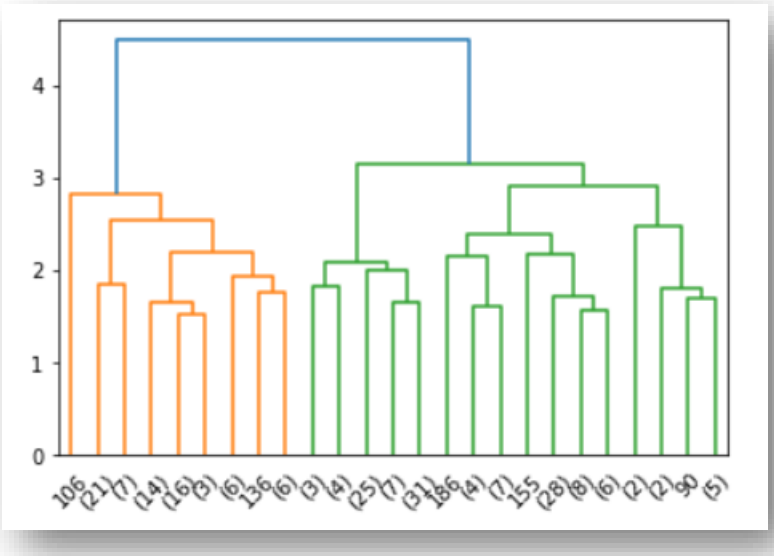

Figure 28: Dendrogram after truncating

From the dendrogram shown in Figure 28, it is seen that the data has been clustered into 3 different groups (the different colours). There clusters are now mapped to the dataset. To do this, fclusters and the criterion maxclust is used.

Table 17: Dataset with the average method clusters

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	clusters_avg
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

Table 18: Dataset grouped average clustering method

clusters	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

2. Agglomerative method:

In this method, the linkage method used is average and affinity used is Euclidean.

Table 19: Dataset grouped by agglomerative clustering method

clusters_agglo	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Freq
0	14.217077	14.195846	0.884869	5.442000	3.253508	2.768418	5.055569	65
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

It is seen from Table 18 and Table 19 that in both average method and agglomerative method of hierarchical clustering the data was split into 3 clusters:

1. High Spending group (cluster 1) with 75 entries
2. Medium Spending group (cluster 0) with 65 entries
3. Low Spending group with (cluster 2) 70 entries

The variable min_payment_amt signifies the minimum paid by the customer while making payments for purchases made monthly. The high and medium spending group clear the credits on time. Hence they will not have to make monthly minimum payments. They will also have high advance payments thereby reducing their minimum monthly payment. When the advance is low, automatically the monthly minimum value will rise. Therefore,

- High spending group – This group consists of the people who are having the highest values in

all the variables except the “min_payment_amt”.

- Medium spending group – This group falls in between the other two clusters, except that it has the least value in “min_payment_amt”.
- Low spending group – This group consists of the people who have the least values in all variables, and highest value in “min_payment_amt”.

1.4 Apply K-Means clustering on scaled data and determine optimum clusters. Apply elbow curve and silhouette score. Explain the results properly. Interpret and write inferences on the finalized clusters.

Kmeans inertia was calculated by using ‘for loop’ given in Figure 29. Kmeans clustering for 10 clusters was done and the corresponding inertia value was found out for all the 10 clusters. The values are shown in Table 20. This is later appended into an array, wss.

```
for i in range(1,11):
    KM = KMeans(n_clusters=i, random_state=1)
    KM.fit(df1_scaled)
    wss.append(KM.inertia_)
```

Figure 29: 'for loop' used to calculate Kmeans inertia

Table 20: Kmeans inertia values

```
[1469.9999999999995,
 659.1717544870411,
 430.65897315130064,
 371.38509060801107,
 327.2127816566134,
 289.315995389595,
 262.98186570162267,
 241.8189465608603,
 223.91254221002728,
 206.3961218478669]
```

The Kmeans inertia values in Table 20 was used to plot the graph between WSS and the number of clusters to find the optimum number of clusters.

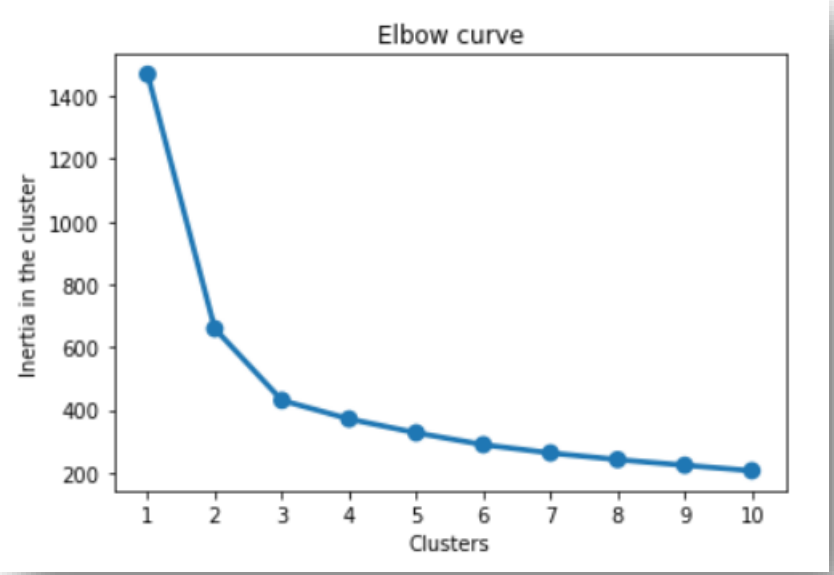

Figure 30: Elbow curve

The elbow curve shown in Figure 30 is a useful visual tool to select the number of clusters. On the X-axis are shown the number of clusters and on the Y-axis are shown the inertia in the cluster. If there is a distinct break point in the line joining the inertia beyond which the line becomes approximately horizontal, then that point may be taken as the value of number of clusters.

It is found from the plot shown in Figure 30 that at number of clusters=3, there is maximum change or a break point in the slope of the line. Hence 3 can be taken as the optimum number of clusters.

This is further cross checked with the silhouette score.

```
k_means = KMeans(n_clusters = 3,random_state=1)
k_means.fit(df1_scaled)
labels3 = k_means.labels_
```

```
silhouette_score(df1_scaled, labels3, random_state=1)
```

```
0.40072705527512986
```

Figure 31: Silhouette score for 3 clusters

```
k_means = KMeans(n_clusters = 4,random_state=1)
k_means.fit(df1_scaled)
labels4 = k_means.labels_
```

```
silhouette_score(df1_scaled, labels4, random_state=1)
```

```
0.3276547677266192
```

Figure 32: Silhouette score for 4 clusters

From Figure 31 and Figure 32, we can see that the silhouette score is highest for 3 clusters and hence shows that the data is split well into 3 clusters.

Table 21: Dataset with the kmeans method clusters

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

Table 22: Dataset grouped by kmeans clustering method

clusters_kmeans	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	14.437887	14.337746	0.881597	5.514577	3.259225	2.707341	5.120803
1	11.856944	13.247778	0.848253	5.231750	2.849542	4.742389	5.101722
2	18.495373	16.203433	0.884210	6.175687	3.697537	3.632373	6.041701

The silhouette width is calculated and the minimum was is found out.

```
silhouette_samples(df1_scaled, labels3).min()
```

```
0.002713089347678376
```

Figure 33: Silhouette width minimum score

The minimum score is shown in Figure 33. It is found to be positive. Therefore mapping is done correctly.

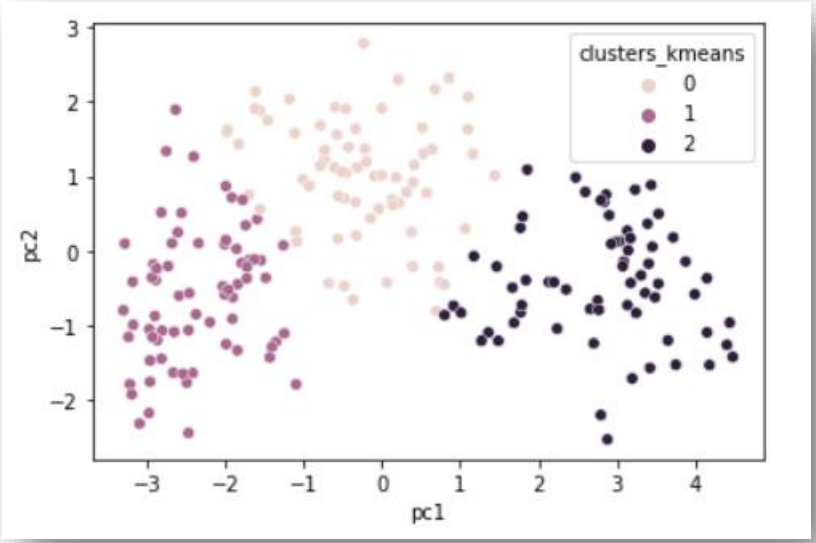

Figure 34: Kmeans clustering data points

Observations:

From Table 22, it was found that the data was split into three clusters using kmeans clustering. To visualize these clusters, this is further plotted and shown in Figure 34. The three clusters are:

1. High spending group (cluster 2)
2. Medium spending group (cluster 0)
3. Low spending group (cluster 1)

The variable min_payment_amt signifies the minimum paid by the customer while making payments for purchases made monthly. The high and medium spending group clear the credits on time. Hence they will not have to make monthly minimum payments. They will also have high advance payments thereby reducing their minimum monthly payment. When the advance is low, automatically the monthly minimum value will rise. Therefore,

- High spending group – This group consists of the people who are having the highest values in all the variables except the “min_payment_amt”.
- Medium spending group – This group falls in between the other two clusters, except that it has the least value in “min_payment_amt”.
- Low spending group – This group consists of the people who have the least values in all variables, and highest value in “min_payment_amt”.

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

Table 23: Kmeans clustering data summary

clusters_kmeans	0	1	2
spending	14.437887	11.856944	18.495373
advance_payments	14.337746	13.247778	16.203433
probability_of_full_payment	0.881597	0.848253	0.884210
current_balance	5.514577	5.231750	6.175687
credit_limit	3.259225	2.849542	3.697537
min_payment_amt	2.707341	4.742389	3.632373
max_spent_in_single_shopping	5.120803	5.101722	6.041701

Group 1: High Spending Group (cluster number 2):

High spending group people are the people who spend more. Hence if they are given reward points or offers they are most likely to shop even more thereby spending more. The credit limit of the group can be increased thereby encouraging them for more purchases. Since the maximum_spent_in_single_shopping is highest in this group, they must have a good amount of balance in the bank. Their credit cards can accumulate points when they make purchases and will be able to provide offers when they have a certain amount of credit points. This way, they are encouraged to shop more, use their credit cards on high valued objects to get more points easily compared to other banks.

Group 2: Medium Spending Group (cluster number 0):

Medium spending group people are customers who pay bills and do purchases. They maintain comparatively a good credit score. Therefore they are the potential target customers. They will be able to increase credit limit or lower down interest rate. They can be made to purchase more by offering them premium cards or loyalty cards to increase or promote their transactions. Another way to increase spending of the medium spending group is by tying up with premium ecommerce sites, travel portal, travel airlines/hotel, as this will encourage them to spend more.

Group 3: Low Spending Group (cluster number 1):

Low spending group people should be given reminders for payments so that they pay on time. Offers can be provided on early payments to improve their payment rate. The spending of this group can be increased by tying up with grocery stores, utilities (electricity, phone, gas, others). These are the essentials that all the people will definitely spend on. Therefore tying up with these commodities and giving them offers will make them slowly increase their spending.