PROJECT - 4

PYTHON: CREDIT CARD CASE STUDY - SEGMENTATION

ABSTRACT:

The case requires developing a customer segmentation to define marketing strategy.
The sample dataset summarizes the usage behaviour of 9000 active credit card holder during the last 6 months.

SUMMARIES OF PROBLEM, DATA, METHODS, AND TECHNOLOGIES:

❖ PROBLEM SUMMARY

The project can be internally divided into four tasks namely –

A> Preparing data built with intelligent KPIs.
B> Provide the detailed insights/observations based on the analysis.
C> Cluster Analysis.
D> Providing strategic insights.

❖ DATA SUMMARY

The input data provided is in CSV data format. The data need to be imported using 'read_csv' function of 'pandas' library.

The data required lots of data manipulation in terms of data conversion and cleaning.

There are 18 columns namely –

DATA DICTIONARY:

CUST_ID: Credit card holder ID
BALANCE: Monthly average balance (based on daily balance averages)
BALANCE_FREQUENCY: Ratio of last 12 months with balance
PURCHASES: Total purchase amount spent during last 12 months
ONEOFF_PURCHASES: Total amount of one-off purchases
INSTALLMENTS_PURCHASES: Total amount of installment purchases
CASH_ADVANCE: Total cash-advance amount
PURCHASES_ FREQUENCY: Frequency of purchases (Percent of months with at least one purchase)
ONEOFF_PURCHASES_FREQUENCY: Frequency of one-off-purchases
PURCHASES_INSTALLMENTS_FREQUENCY: Frequency of installment purchases
CASH_ADVANCE_ FREQUENCY: Cash-Advance frequency
AVERAGE_PURCHASE_TRX: Average amount per purchase transaction
CASH_ADVANCE_TRX: Average amount per cash-advance transaction
PURCHASES_TRX: Average amount per purchase transaction
CREDIT_LIMIT: Credit limit
PAYMENTS: Total payments (due amount paid by the customer to decrease their statement balance) in the period
MINIMUM_PAYMENTS: Total minimum payments due in the period.
PRC_FULL_PAYMEN: Percentage of months with full payment of the due statement balance
TENURE: Number of months as a customer

Email: banerjee.kaustav@outlook.com
Github: https://github.com/KBanerjee90

❖ METHODS SUMMARY

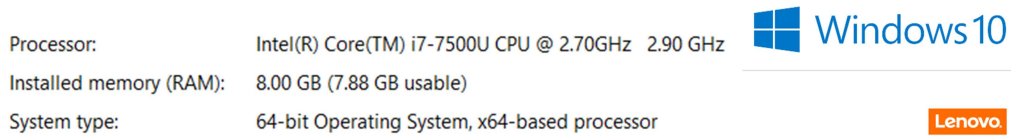
Table shows the wide variety of data pre-processing, analysis, and visualization techniques that I applied to complete the tasks as part of the project –

Task ID	Task Details	Analytical Techniques	Visualization Techniques
Data Manipulation and preparation	1> Preparing data built with intelligent KPIs. 2> Provide the detailed insights/observations based on the analysis.	Descriptive statistics Straightforward data manipulation	pandas_profiling.ProfileReport seaborn (regplot and heatmap) matplotlib.pyplot (bar, scatter plot)
Modelling and performance	1> Cluster Analysis. 2> Providing strategic insights.	K-MEANS cluster Analysis	

❖ TECHNOLOGIES SUMMARY

The following list summarizes the technology that I used:

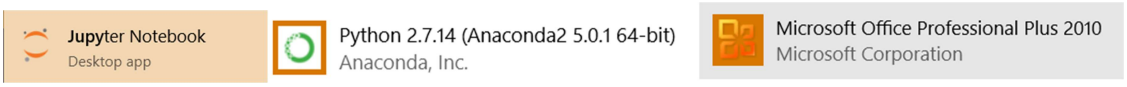
Computing platforms:



Microsoft Excel 2010 on the single-machine platform

Jupyter Notebook on the single-machine platform

Python 2.7.14 (Anaconda 2 5.0.1 64bit) on the single-machine platform



Email: banerjee.kaustav@outlook.com
Github: https://github.com/KBanerjee90

SUMMARY OF DATA PROCESSING AND MODELLING CLUSTERS

❖ DATA PROCESSING:

I downloaded the input csv data in into python data frame –

```
credit_data = pd.read_csv('CREDIT CARD USERS DATA.csv')
```

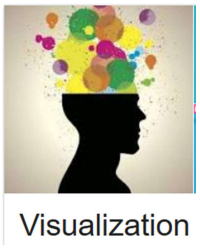
a. Deriving **New KPIs** by using straight forward data manipulation and descriptive statistics

```
# 1. Monthly Average Purchases -
credit['MNTHLY_AVG_PURCHASE'] = credit['PURCHASES']/credit['TENURE']

# 2. Monthly Cash Advance -
credit['MONTHLY_AVG_CASH_ADVANCE'] = credit['CASH_ADVANCE']/credit['TENURE']

# 3. Purchase by Type –
 if ((credit.ONEOFF_PURCHASES == 0) & (credit.INSTALLMENTS_PURCHASES == 0)):
     return 'NONE'
 if ((credit.ONEOFF_PURCHASES > 0) & (credit.INSTALLMENTS_PURCHASES == 0)):
     return 'ONE_OFF'
 if ((credit.ONEOFF_PURCHASES > 0) & (credit.INSTALLMENTS_PURCHASES > 0)):
      return 'BOTH_ONEOFF_INSTALLMENT'
 if ((credit.ONEOFF_PURCHASES == 0) & (credit.INSTALLMENTS_PURCHASES > 0)):
      return 'INSTALLMENTS'

#4. Avg Amount per cash advance transaction is equivalent to CASH_ADVANCE_TRX

#5. Avg Amount per purchase is equivalent to AVERAGE_PURCHASE_TRX

#6. LIMIT USAGE (Credit Score - Lower implies customers are maintaining their balance properly)
credit['LIMIT_USAGE'] = credit.apply(lambda x: x['BALANCE']/x['CREDIT_LIMIT'],axis =1)

#7. PAYMENT_MINPAYMENT
#The where clause is being used to avoid div by zero error
credit['PAYMENT_MINPAYMENT']= np.where(credit['MINIMUM_PAYMENTS']== 0,
credit['PAYMENTS'], credit['PAYMENTS']/credit['MINIMUM_PAYMENTS'])
```

b. **Insights** from new KPIs –

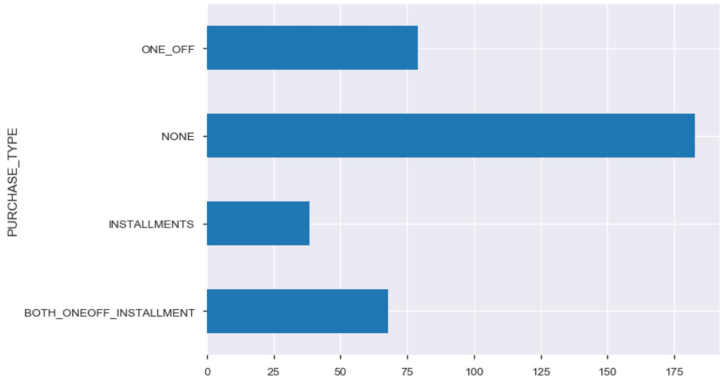
- Customers with instalment payments are paying dues
- Customers who do not do ONOFF or INSTALLMENTS take more cash advance
- Customers with instalment purchases have good credit score

Email: banerjee.kaustav@outlook.com
Github: https://github.com/KBanerjee90

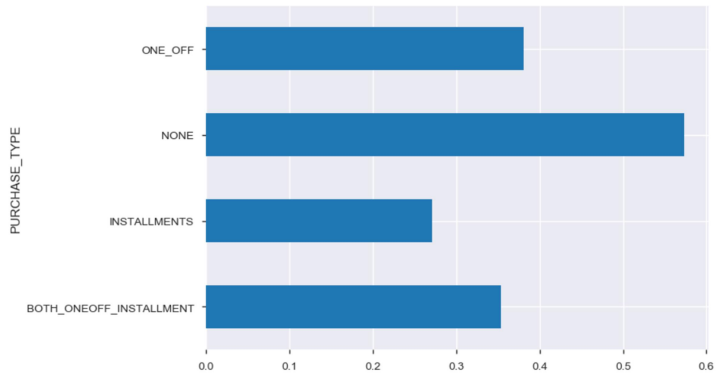


Visualization the insights from plotting the KPIs against PURCHASE_TYPE

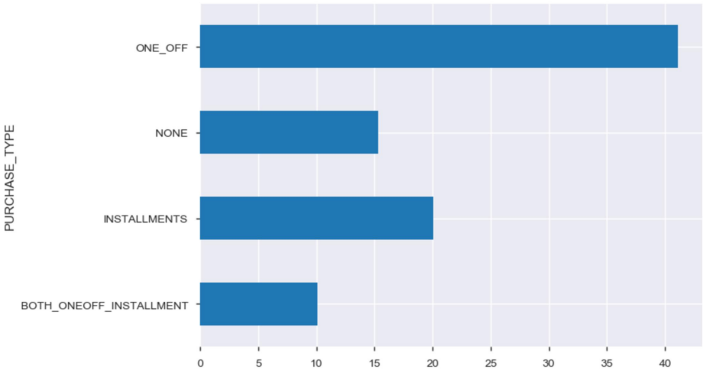
#PLOT 1: PURCHASE_TYPE VS MONTHLY_AVG_CASH_ADVANCE



#PLOT 2: PURCHASE_TYPE VS LIMIT_USAGE

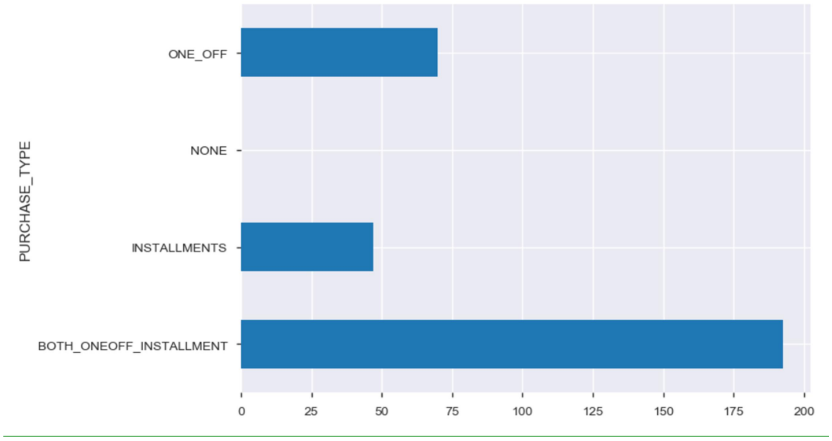


#PLOT 3: PURCHASE_TYPE VS PAYMENT_MINPAYMENT



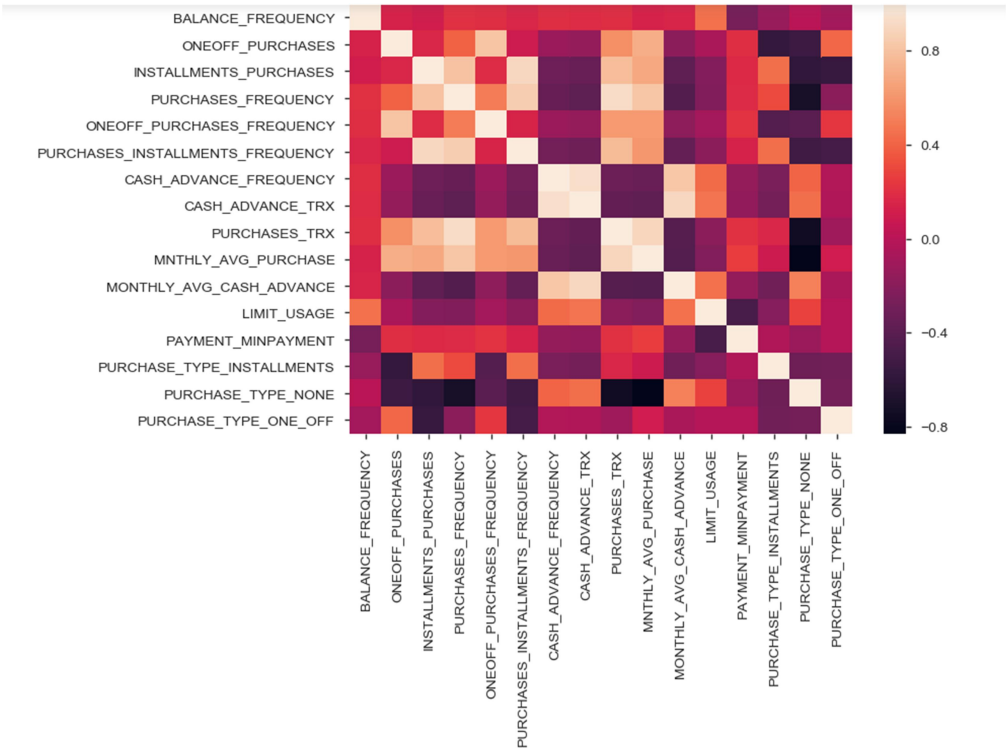
Email: banerjee.kaustav@outlook.com
Github: https://github.com/KBanerjee90

#PLOT 4: PURCHASE_TYPE VS MNTHLY_AVG_PURCHASE



c. **Dividing data** between numeric and categorical variables-
   a. Log transformation of numeric variables
   b. Getting the dummies of categorical variables

d. Checking for **multi collinearity** using heatmap



Correlation data available at – output\ credit_data_correlation_matrix.csv

Email: banerjee.kaustav@outlook.com
Github: https://github.com/KBanerjee90

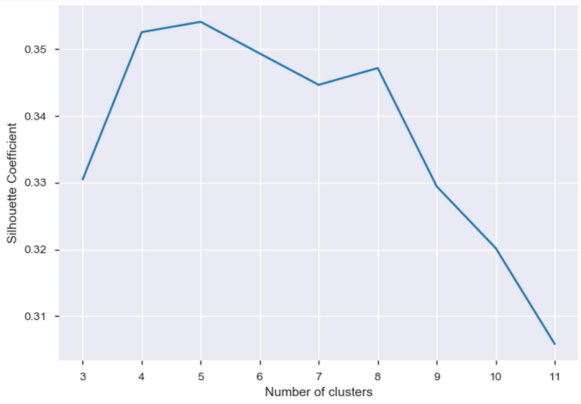
e. Standardizing the data and applying **PCA** to get the optimal component

*Since 5 components are explaining about **88**% of the variance we select 5 components*

```
var_ratio
```

```
{2: 0.58417805680543466,
 3: 0.73832371403905706,
 4: 0.8231815341875961,
 5: 0.88287829066608126,
 6: 0.91702854611044182,
 7: 0.94250389088991038,
 8: 0.96107618837464415,
 9: 0.9737386611377391,
 10: 0.9840729849308939,
 11: 0.98936725369750611,
 12: 0.99257748700336268,
 13: 0.99530819888459787,
 14: 0.99791598529606274}
```

f. Using **Silhouette Coefficient** to get optimal cluster - The solution can be **4** or **5** or **8**



Note

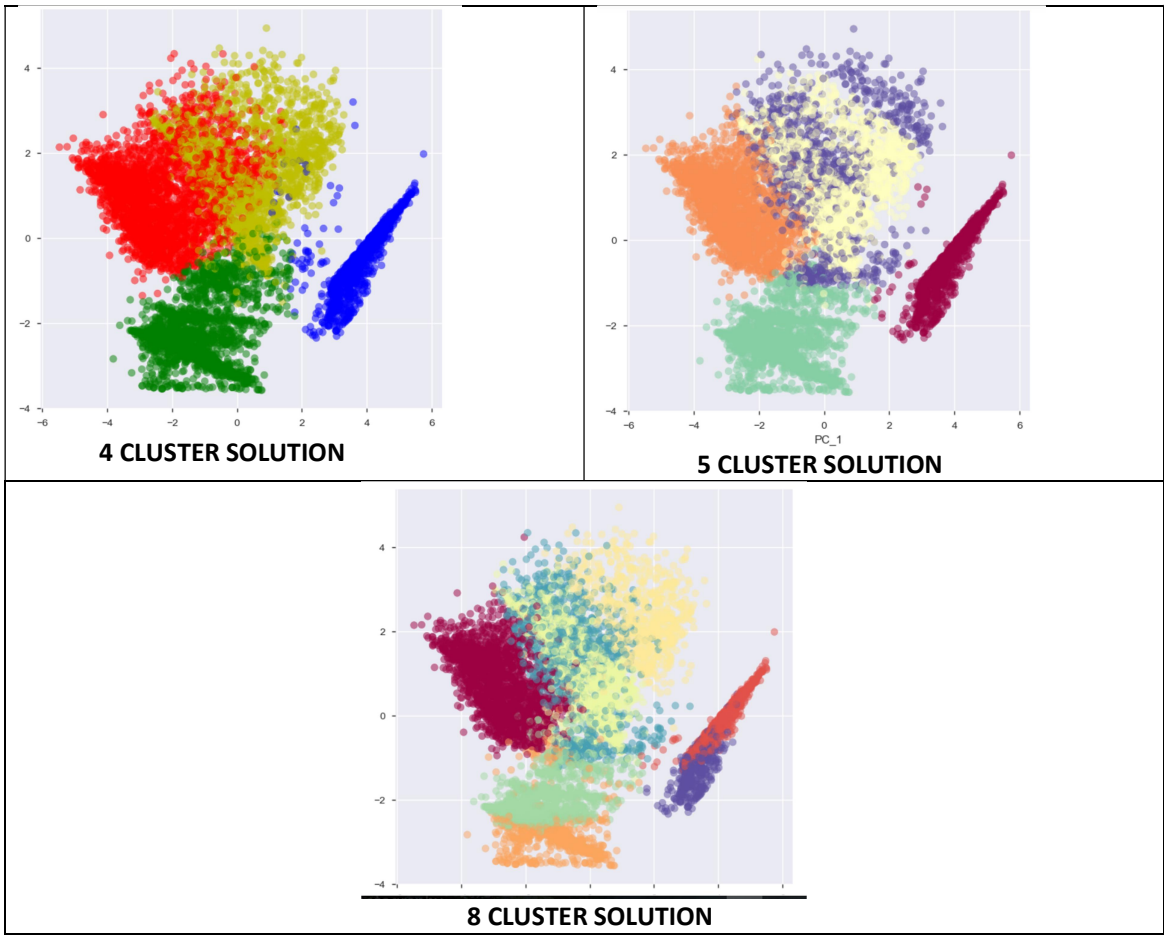
The solution can be 4 or 5 or 8

Refer to output files, profile reports at: output/

Email: banerjee.kaustav@outlook.com
Github: https://github.com/KBanerjee90

❖ CLUSTER SOLUTIONS

I performed cluster analysis using **4**, **5** and **8** clusters to find the optimal cluster having best distinguishing characteristics



4 CLUSTER SOLUTION

5 CLUSTER SOLUTION

8 CLUSTER SOLUTION

RESULTS AND OUTCOME:

Overall Profiling report for K MEANS -

20% or more above Overall			4 Cluster K-means				5 Cluster K-means					8 Cluster K-means							
20% or less below Overall		Overall	1	2	3	4	1	2	3	4	5	1	2	3	4	5	6	7	8
Segment size			29%	23%	26%	22%	23%	22%	20%	23%	13%	21%	18%	8%	8%	14%	16%	11%	5%
PURCHASES_TRX	Avg	14.71	34.82	0.07	11.86	6.68	0.01	36.78	6.49	11.63	21.63	37.56	0.01	6.73	5.93	7.62	14.03	24.37	0.01
MNTHLY_AVG_PURCHASE	Avg	86.18	204.18	0.23	46.87	66.10	0.01	223.08	66.86	46.23	108.90	230.44	0.02	30.99	53.69	75.97	53.89	118.97	0.01
MONTHLY_AVG_CASH_ADVANCE	Avg	88.98	68.85	186.51	32.77	77.49	183.43	7.54	44.93	9.08	273.66	6.26	190.39	3.80	204.14	3.43	12.25	245.71	156.32
LIMIT_USAGE	Avg	0.39	0.36	0.58	0.26	0.38	0.57	0.27	0.35	0.21	0.63	0.27	0.69	0.02	0.61	0.25	0.32	0.61	0.14
CASH_ADVANCE_TRX	Avg	3.25	2.87	6.56	1.00	2.85	6.31	0.25	1.49	0.20	11.22	0.20	7.13	0.07	7.60	0.12	0.29	9.97	3.38
PAYMENT_MINPAYMENT	Avg	20.29	8.32	15.01	22.24	40.00	15.24	12.75	43.73	21.80	3.41	8.75	1.98	72.85	3.25	61.27	2.85	3.55	62.96
PURCHASE_TYPE_INSTALLMENTS	Yes%	25%	0.00	0.02	0.97	0.00	0.01	0.00	0.00	0.97	0.21	0.00	0.01	0.88	0.00	0.00	0.97	0.22	0.00
PURCHASE_TYPE_NONE	Yes%	23%	0.00	0.97	0.00	0.00	0.99	0.00	0.00	0.00	0.00	0.00	0.99	0.00	0.00	0.00	0.00	0.00	1.00
PURCHASE_TYPE_ONE_OFF	Yes%	21%	0.00	0.00	0.00	0.97	0.00	0.00	0.97	0.00	0.15	0.00	0.00	0.00	0.99	0.96	0.00	0.00	0.00
CREDIT_LIMIT	Avg	4494.28	5823.00	4069.47	3360.41	4504.12	4032.78	5841.19	4332.91	3174.45	5670.83	5937.00	4130.65	3557.84	4562.40	4471.57	3032.11	5589.92	3665.81

*We can conclude with confidence that **4** CLUSTER should be best solution since it is giving us perfectly distinguishable features; while **5** cluster/**8** cluster solutions have multiple overlapping.*

Email: banerjee.kaustav@outlook.com
Github: https://github.com/KBanerjee90

4 Clusters solution are clearly distinguishing with following references –

Cluster 1 - This group is about **29%** of the total customer base. This group of customers have highest monthly Average Purchases and have good Credit Score. This group is doing both instalments as well as on off purchases and have comparatively low minimum payment.

Cluster 2 - This group is about **23%** of the total customer base. This group is taking maximum cash advance and has poor credit score. This group is more inclined to NO purchase transaction.

Cluster 3 - This group is about **26%** of the total customer base. This group is paying dues and are doing maximum instalments purchase.

Cluster 4 - This group is about **22%** of the total customer base. This group is more inclined to ON_OFF purchases and have high minimum payment.

For **5** cluster solution - do not have distinguishable features -

Cluster 1 and **Cluster 5** overlap each other on cash advance and credits score.

Cluster 2 and **Cluster 5** overlap each other on monthly purchases.

Profiling report is available at – output\Segmentation_cluster analysis_profiling - CreditCard.xlsx

Solution is available at – solution\ Credit_Card_Segmentation_Solution.ipynb

RECOMMENDATIONS

Cluster 1 customers can be given loyalty point and offers; since they have good credit score they can even be offered **add on credit cards** to boost their purchases. They are our ideal customers.

Cluster 2 customers have poor credit score; these customers should be given reminders for payments. Offers can be provided on early payment/full payments to improve their payment rate.

Cluster 3 customers are instalment customers; they can be offered discounts/offers on **ON-OFF** purchases. However make sure to send reminders before instalment due dates. They can be offered loyalty points upon completed transactions.

Cluster 4 customers can be given offers on various instalments/offers on EMI. This can boost them to do instalments purchase along with ON_OFF purchases. Customers of this group can be offered discount/offer on next transactions upon Full Payment since they are prone to minimum payment.

REFERENCES

https://github.com/

https://www.kaggle.com/

Analytixlabs Class 16-17 example Telco Segmentation

Email: banerjee.kaustav@outlook.com
Github: https://github.com/KBanerjee90
Linkedin: https://www.linkedin.com/in/kaustav-banerjee-584525149