Disclaimer: *The work presented is our team's (Team5) work and our team's work alone.*

ABSTRACT

Prachi Gupta

❖ Contents

1. Executive Summary

    - Introduction

2. About data

3. Data Preprocessing

4. Data Visualization

    - Univariate Analysis
    - Multivariate Analysis
    - Correlation Analysis
    - Principal Component Analysis

5. Running Prediction Models

6. Performance Evaluation

7. Conclusion

1. Executive Summary

For any lender, one of the most important tasks is to predict the event or probability of default by a borrower. The credit card industry predicts the default event (yes/no) on a credit card balance by a customer at any given point of time.

Objective of the Project

The project's objective is to develop and choose the best prediction model for default on the credit card bill payment by using various statistical tools and modeling techniques.

Methods and material:

We have used JMP Pro 12 and Microsoft Excel to perform the various analysis. On the sample data, we initially understood the meaning of each variable using the data dictionary and online research.

Conducting univariate analysis, both statistical and graphical, helped process the columns individually. We successfully took care of missing values and outliers for each column by imputing them appropriately. Following the univariate analysis, multivariate methods were conducted to understand the overall data variability. We executed multivariate correlations to understand how the variables related to each other which helped in deciding what variables to include in our analysis and what variables to avoid which helped in reducing data redundancy and complexity.

Finally, we observed the variation of various predictors with the target variable.

Introduction:-

In a market, any purchase activity requires payment in return. All over the world, there has been a rapid enhancement from traditional cash payments to plastic and digital payments. In the developed world and increasingly in the developing world, we see that much of the payments happen through credit cards. Any company issuing a credit card to a customer gives a certain credit limit, which is the maximum balance allowed on the credit card. The company makes this decision of what credit limit to be given on the basis of the spending needs, payment ability and risk assessment of the customer. At the core of these decisions is the prediction by the company of whether a customer will default on the credit balance or not. If the company is able to accurately identify which customers will default, they can limit their risk by reducing the credit limit allowed or not allowing the customer to spend any further or take other suitable decisions. The classification of status i.e. in future customer will be defaulter or non-defaulter is a challenging task for the company. The defaulter prediction is a binary classification problem. Default prediction is at the heart of any credit decision or limit allocation by a credit card company. In this project, six months data on approximately thirty thousand customers is analyzed. We

implemented various classification algorithms namely Neural Network, Logistic Regression, Decision Tree, Boostrap Forest and Boosted Tree to predict the event of customer default in the next month.

2. About data:-

a. Data source:

The data set used for this project is titled 'UCI Credit Card'. We have picked up the data from Kaggle.com. This data is about credit card customers from Taiwan from the year 2005.

The data contains various attributes such as Limit Balance and customer demographic data such as Gender, Education, Marital Status, Age. It also has variables for repayment Status, bill amount and payment amount for previous six months. It is stored in csv format.

b. Data description:

In this data there are about thirty thousand observations. It has 23 dependent variables and one response variable which is binary coded as 0 or 1 for payment default. The list of variable is as follows,

Variable Name	Type
ID	Integer
Limit Bal	Integer
Sex	Categorical Integer
Education	Categorical Integer
Marriage	Categorical Integer
Age	Integer
Pay_0, Pay_2, Pay_3, Pay_4, Pay_5, Pay_6	Categorical Integers
Bill_Amt1-6 (for each of the six months from Apr-Sep 2005)	Integers
Pay_Amt1-6 (for each of the six months from Apr-Sep 2005)	Integers
Default Payment Next Month	Binary

c. Terminology:-

Below is the description of some terms related with this project

i) Credit Card:-

Credit card is a type of financial account. By using credit cards, customers make use of an issuer company's money instead of their own to pay for a product or service today, and over time, they repay the company. They can delay the payment made by paying a small amount called minimum due and can pay the remaining amount subsequently. For this benefit of easy repayment on someone else's money, customers will often need to pay interest, as expected with other types of loans.

ii) Default:-

When payments are not made in time and according to the agreement signed by the card holder, the account is said to be in default.

3. Data Preprocessing

Recoding of Categorical Variables from Numeric to Descriptive Strings

This helps in ease of identification and analysis.

1. **Sex:** Recoding numeric values 1 and 2 as Male and Female respectively.

2. **Education:** Recoding numeric values as follows.

   1 = graduate school; 2 = university; 3 = high school; 4 = others

   The data also had few observations with values 0, 5, 6. We considered them to be missing since no explanation is provided for them.

   We have clubbed all the missing and other values in others and recoded the numeric values into description strings as below.

3. **Marital Status:** Recoding numeric values as follows.

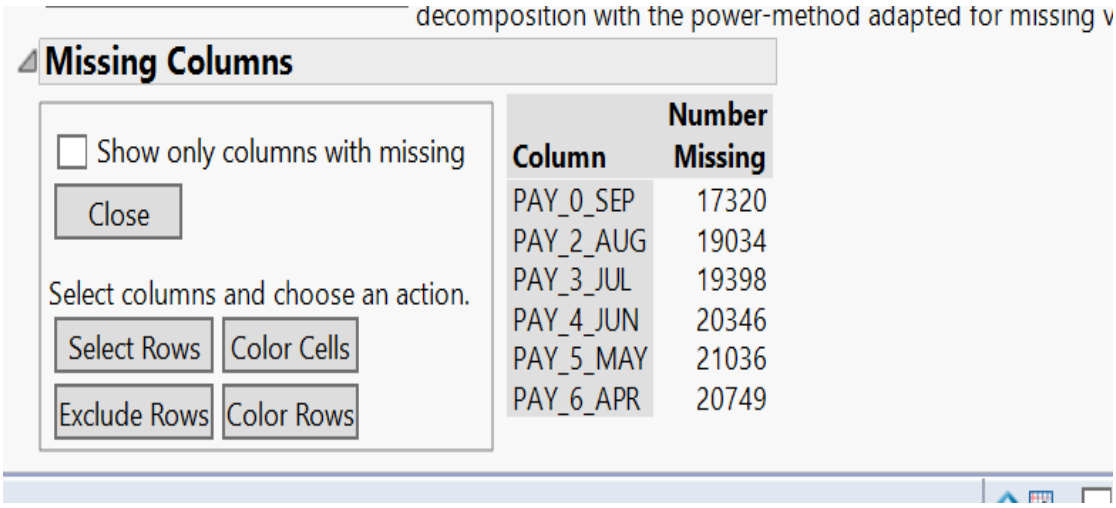
   1 = married; 2 = single; 3 = others

   There are few missing observations with value 0.

   We recoded the missing values into others.

Missing Value in Pay Variable

The pay variables were analyzed and it was observed that the values of 0 and -2 in the pay variables were missing values.



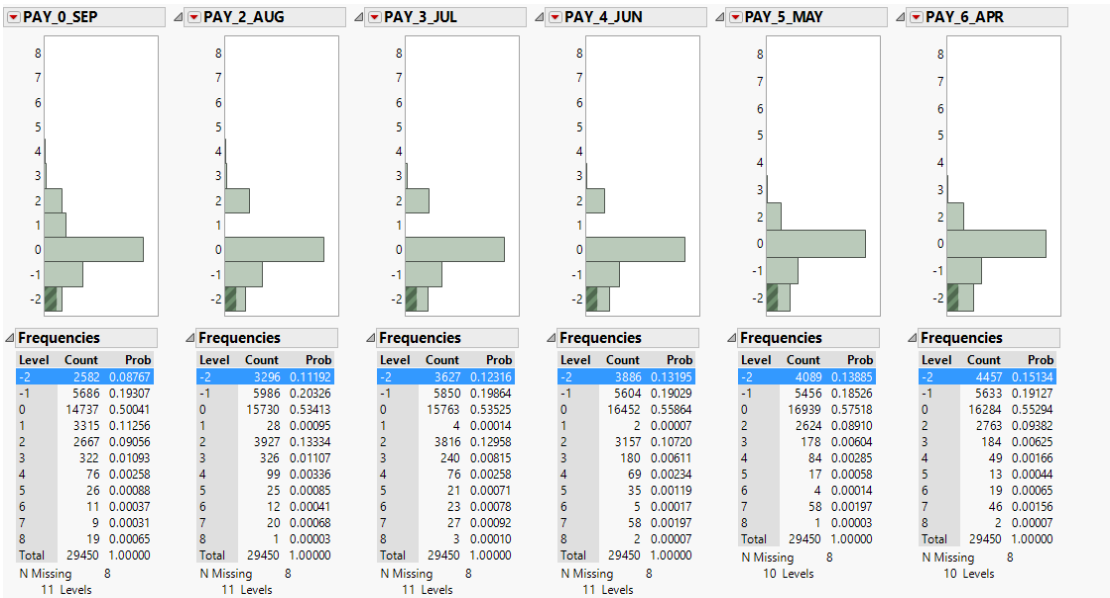
Column	Number Missing
PAY_0_SEP	17320
PAY_2_AUG	19034
PAY_3_JUL	19398
PAY_4_JUN	20346
PAY_5_MAY	21036
PAY_6_APR	20749

Standardization of Missing Values

The missing values of 0 and -2 were to a single value of 0.

Handling the Missing Value

Before processing, the pay variables had the following distribution.



PAY_0_SEP Level	Count	Prob
-2	2582	0.08767
-1	5686	0.19307
0	14737	0.50041
1	3315	0.11256
2	2667	0.09056
3	322	0.01093
4	76	0.00258
5	26	0.00088
6	11	0.00037
7	9	0.00031
8	19	0.00065
Total	29450	1.00000

N Missing 8
11 Levels

PAY_2_AUG Level	Count	Prob
-2	3296	0.11192
-1	5986	0.20326
0	15730	0.53413
1	28	0.00095
2	3927	0.13334
3	326	0.01107
4	99	0.00336
5	25	0.00085
6	12	0.00041
7	20	0.00068
8	1	0.00003
Total	29450	1.00000

N Missing 8
11 Levels

PAY_3_JUL Level	Count	Prob
-2	3627	0.12316
-1	5850	0.19864
0	15763	0.53525
1	4	0.00014
2	3816	0.12958
3	240	0.00815
4	76	0.00258
5	21	0.00071
6	23	0.00078
7	27	0.00092
8	3	0.00010
Total	29450	1.00000

N Missing 8
11 Levels

PAY_4_JUN Level	Count	Prob
-2	3886	0.13195
-1	5604	0.19029
0	16452	0.55864
1	2	0.00007
2	3157	0.10720
3	180	0.00611
4	69	0.00234
5	35	0.00119
6	5	0.00017
7	58	0.00197
8	2	0.00007
Total	29450	1.00000

N Missing 8
11 Levels

PAY_5_MAY Level	Count	Prob
-2	4089	0.13885
-1	5456	0.18526
0	16939	0.57518
2	2624	0.08910
3	178	0.00604
4	84	0.00285
5	17	0.00058
6	4	0.00014
7	58	0.00197
8	1	0.00003
Total	29450	1.00000

N Missing 8
10 Levels

PAY_6_APR Level	Count	Prob
-2	4457	0.15134
-1	5633	0.19127
0	16284	0.55294
2	2763	0.09382
3	184	0.00625
4	49	0.00166
5	13	0.00044
6	19	0.00065
7	46	0.00156
8	2	0.00007
Total	29450	1.00000

N Missing 8
10 Levels

To handle the missing value in Pay variable we have imputed them with the most logical value that is the previous month pay variable value.

So for example the PAY_0_SEP has the missing value it is replaced by the value of PAY_2_AUG. If that value is also missing we have taken previous month missing

value . So if PAY_0_SEP and PAY_2_AUG are missing we will take value of PAY_3_JUL.

After replacing this with the most logical value

	Count	Number of columns missing
1	8709	0
2	1974	1
3	2002	2
4	1670	3
5	1345	4
6	13758	5

Below we show the formulae for this treatment of missing values of each month's pay variable

Formula for creating new Pay_may

```
If( PAY_5_MAY == 0 ⇒ PAY_6_APR
    else           ⇒ PAY_5_MAY )
```

Create new Pay variable for May such that if Pay variable of May month has missing value (0), then take value of previous month (April)

Formula for creating new Pay_Jun

```
If( PAY_4_JUN == 0 ⇒ new pay may
    else           ⇒ PAY_4_JUN )
```

New Pay variable for June will take the value of new Pay variable of May month if the existing Pay variable for June has missing value (0), else it will take the value of existing Pay variable for June.

Formula for creating new Pay_Jul

```
If( PAY_3_JUL == 0 ⇒ new pay jun
    else           ⇒ PAY_3_JUL )
```

New Pay variable for July will take the value of new Pay variable of June month if the existing Pay variable for July has missing value (0), else it will take the value of existing Pay variable for July.

Formula for creating new Pay_Aug

```
If( PAY_2_AUG == 0 ⇒ new pay july
    else           ⇒ PAY_2_AUG )
```

New Pay variable for Aug will take the value of new Pay variable of July month if the existing Pay variable for Aug has missing value (0), else it will take the value of existing Pay variable for Aug.

Formula for creating new Pay_Sep

```
If( PAY_0_SEP == 0 ⇒ new pay aug
    else           ⇒ PAY_0_SEP )
```

New Pay variable for Sep will take the value of new Pay variable of Aug month if the existing Pay variable for Sep has missing value (0), else it will take the value of existing Pay variable for Seps.

At the end, only those values were left missing (0) in pay variables which were missing (0) for all six months.

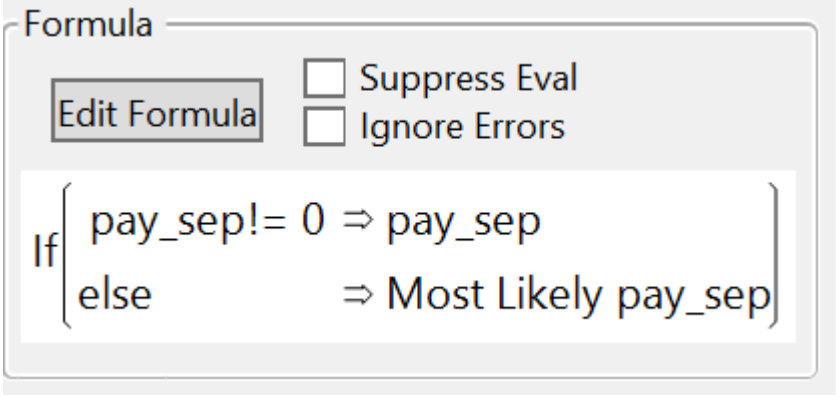
To handle the rows with all the missing value in Pay variable

To get the value of that we ran a bootstrap model and predicted the value of pay variable

Prob ==1

```
< 299 ⇒ If( else ⇒ 0.07135409683659 )
⇒ If( Is Missing( LIMIT_BAL ) | LIMIT_BAL < 230000 ⇒ If( Is Missing( PAY_AMT5_APR ) | PAY_AMT5_APR < 353 ⇒ 0.01612482429003
                                                          else ⇒ If( Is Missing( PAY_AMT4_MAY ) | PAY_
                                                                  else )
      else ⇒ If( ! Is Missing( BILL_AMT5_MAY )
                 & BILL_AMT5_MAY >= 73011 ⇒ 0.08952271048132
                 else ⇒ 0.58437978006897 )
```

After this we replaced the missing value of Pay_Sep variable with the predicted values.



Modifying variable type

We modified all the categorial variables such as Marital Status, Sex, Education, Default Variable from **Continuous to Nominal**.

Handling data inconsistency

There are instances where default indicator = 1 even when all bill amounts are less than or equal to zero. Modifying the default indicator to 0 in such instances to resolve the data inconsistency of default with no bill required.

Division of data into Training, Validation and Test data

Before running the models, we create a validation column. We divide the data using stratified sampling into 50% training, 30% validation and 20% test.

We will train the models on training dataset and then validate the model on validation. We will pick up the best model and lastly test it in test dataset.

4. Data Visualization

Keeping in view the above discussion, the following objectives has been laid down;

1) To study the relationship between available variables and status of default.
2) To model the relationship between available variables and status of default.

We will first perform univariate analysis of all the variables.

Univariate Analysis

In this section, we will give the description of each variable. We will also check the relation of all variables with default status by using frequency table. The frequency distribution of each variable is as follows.

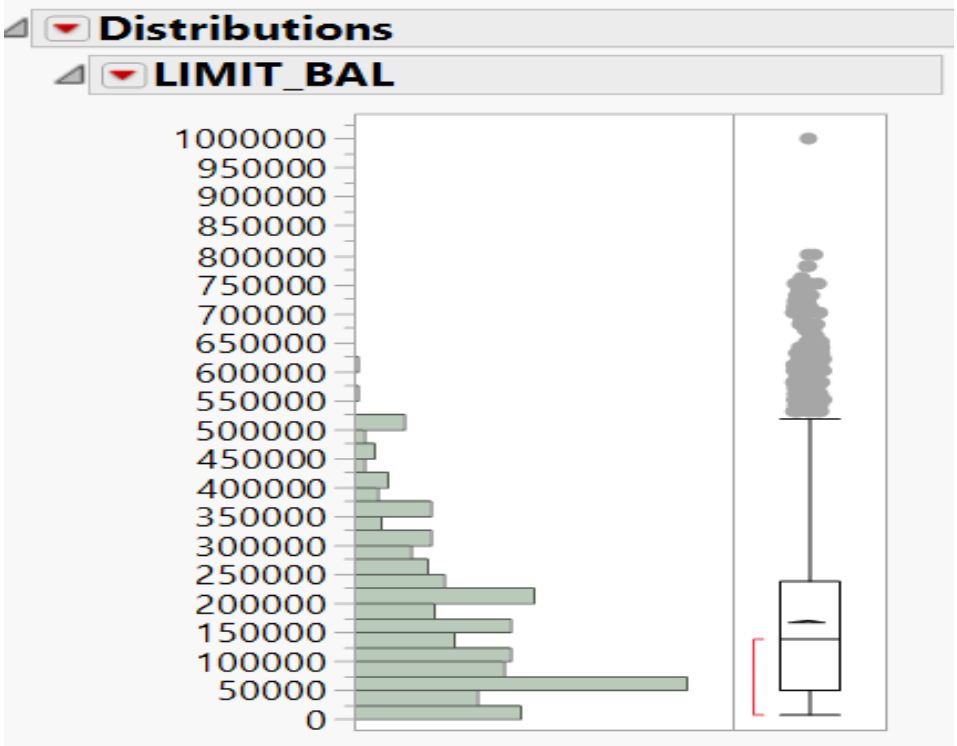
ID:-

Data type | Modeling data type: Numeric | Continuous (Statiscally: Discrete)

Analysis -The column has a unique value for each row without any missing values. It does not help in explaining any pattern in the data but can be useful in identifying the borrower who is likely to default

Limit_Bal:-

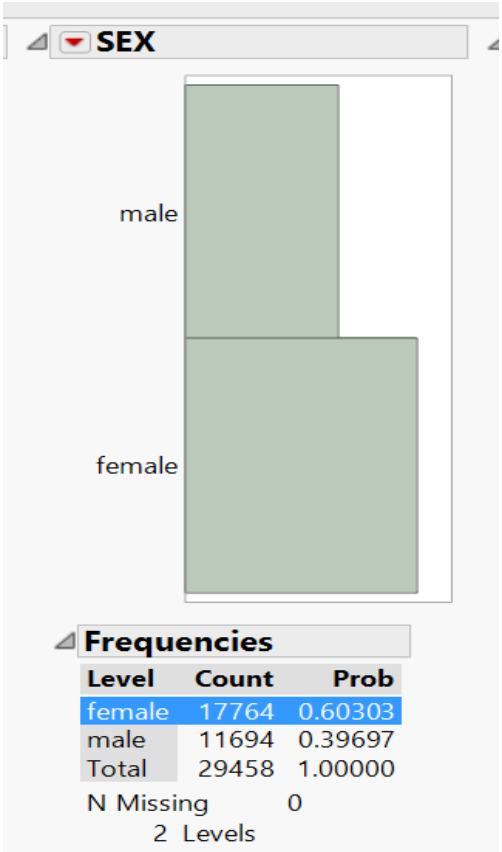
Data type | Modeling data type: Numeric | Continuous (Statiscally: Discrete)



Analysis -Credit Amount allowed to the customer in dollars. It is of integer type. This column didn't contain any missing data.

Sex:-

Data type | Modeling data type: Integer and Binary

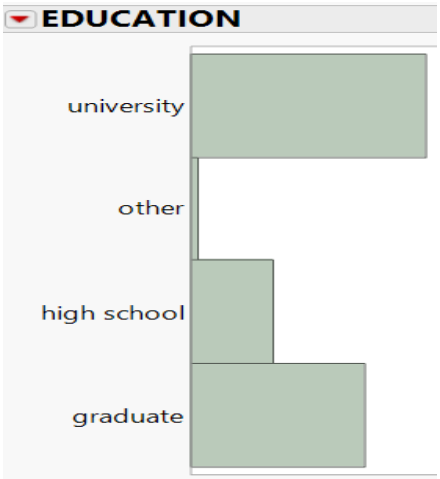


Analysis: This column categorizes the population into male and female. We see that there are more females than males in the data.

Education:-

Data type | Modeling data type: Integer | Categorical variable.

This is a categorical variable. This gives the level of education of the customer.



Frequencies

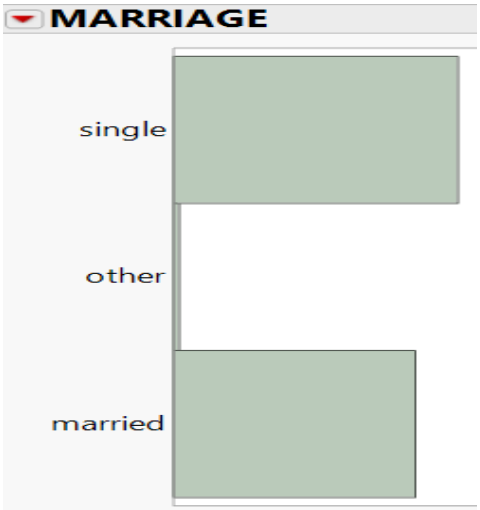
Level	Count	Prob
graduate	10309	0.34996
high school	4834	0.16410
other	467	0.01585
university	13848	0.47009
Total	29458	1.00000

N Missing 0

Analysis: After treating the missing observations and recoding into the categories, we observe that 47% are university educated. 35% are graduate, and 16% have studied till high school. Very few customers fall in Others.

Marital Status:-

Data type | Modeling data type: Integer| Categorical variable.



Frequencies

Level	Count	Prob
married	13382	0.45427
other	374	0.01270
single	15702	0.53303
Total	29458	1.00000

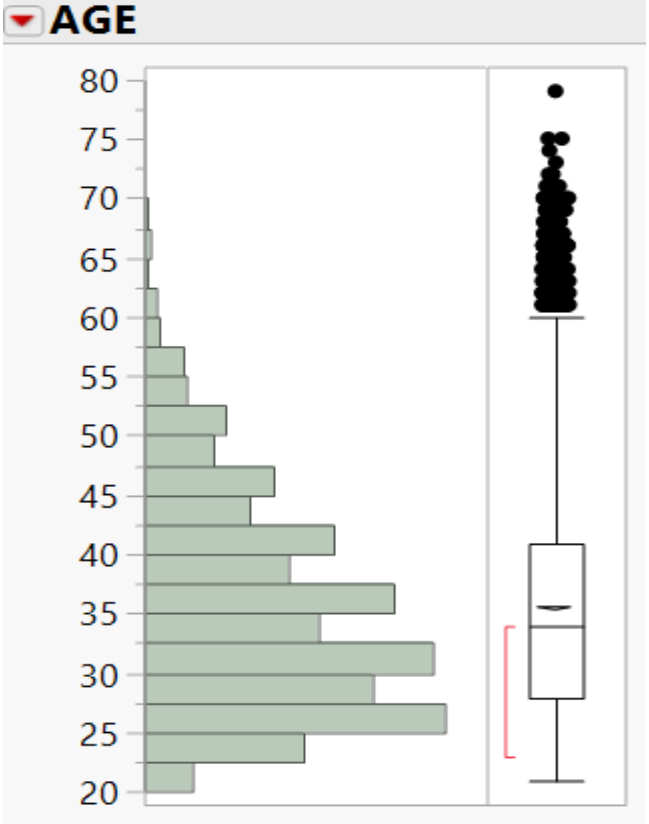
N Missing 0

Analysis: This gives the marital status of the customer – single, married, other.

53% of our customers are single and 45% are married

Age

Data type | Modeling data type: Numeric | Continuous



Quantiles

100.0%	maximum	79
99.5%		63
97.5%		56
90.0%		49
75.0%	quartile	41
50.0%	median	34
25.0%	quartile	28
10.0%		25
2.5%		23
0.5%		22
0.0%	minimum	21

Summary Statistics

Mean	35.429595
Std Dev	9.1872903
Std Err Mean	0.0535286
Upper 95% Mean	35.534513
Lower 95% Mean	35.324676
N	29458

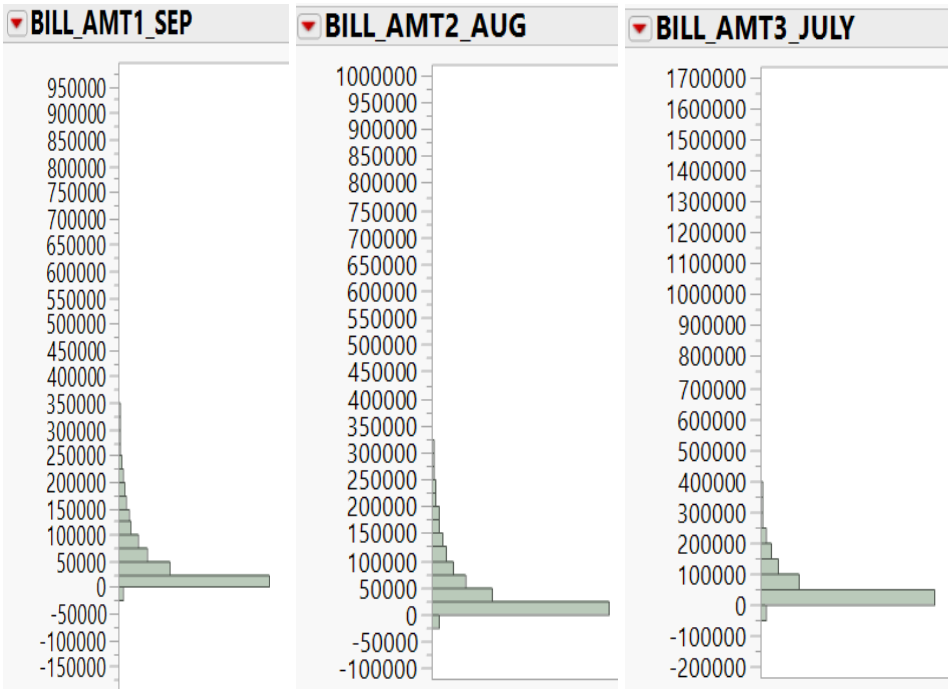
Analysis: This column gives the age of the customer in years. and is of integer type. This column didn't contain any missing data. It is a good practice to create suitable buckets for ease of analysis and model building.

Customers are aged from 20 to 80 with 99% customers under the age of 60. Mean and median ages of the data is at around 35 years

BILL_AMT

Data type | Modeling data type: Integer | Continuous

There are six bill amount variables. We will look at the distribution of few of the variables below.

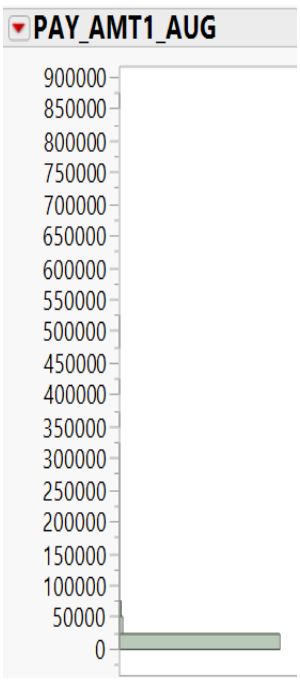


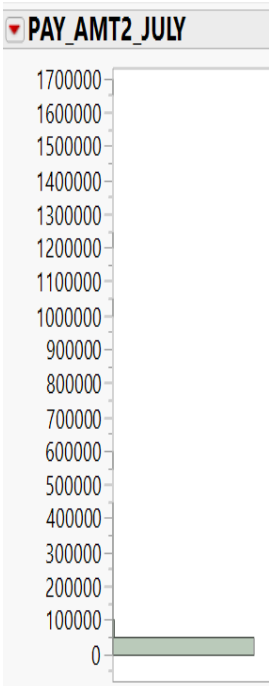
Analysis: There are six bill amount variables. Each variable represents the amount of bill statement of a month each from September, 2005 to April 2005. The unit is Taiwanese dollars.

From above charts, we observe that many customers have zero or negative bill amounts. For most customers, bill amounts do not exceed 300K

PAY_AMT

Data type | Modeling data type: Integer | Continuous





Quantiles		
100.0%	maximum	873552
99.5%		100061.28
97.5%		32459.325
90.0%		10464.4
75.0%	quartile	5045.75
50.0%	median	2197.5
25.0%	quartile	1005
10.0%		0
2.5%		0
0.5%		0
0.0%	minimum	0

Quantiles		
100.0%	maximum	1684259
99.5%		120060.64
97.5%		33504.2
90.0%		10543.4
75.0%	quartile	5000
50.0%	median	2052.5
25.0%	quartile	1000
10.0%		0
2.5%		0
0.5%		0
0.0%	minimum	0

Analysis: There are six pay amount variables. Each variable represents the amount of payment made on the previous months' bill each from September, 2005 to April 2005. The unit is Taiwanese dollars.

From above charts, we see that many customers had zero payment amounts in at least one month. In Aug'05, median pay amount was only 2200.

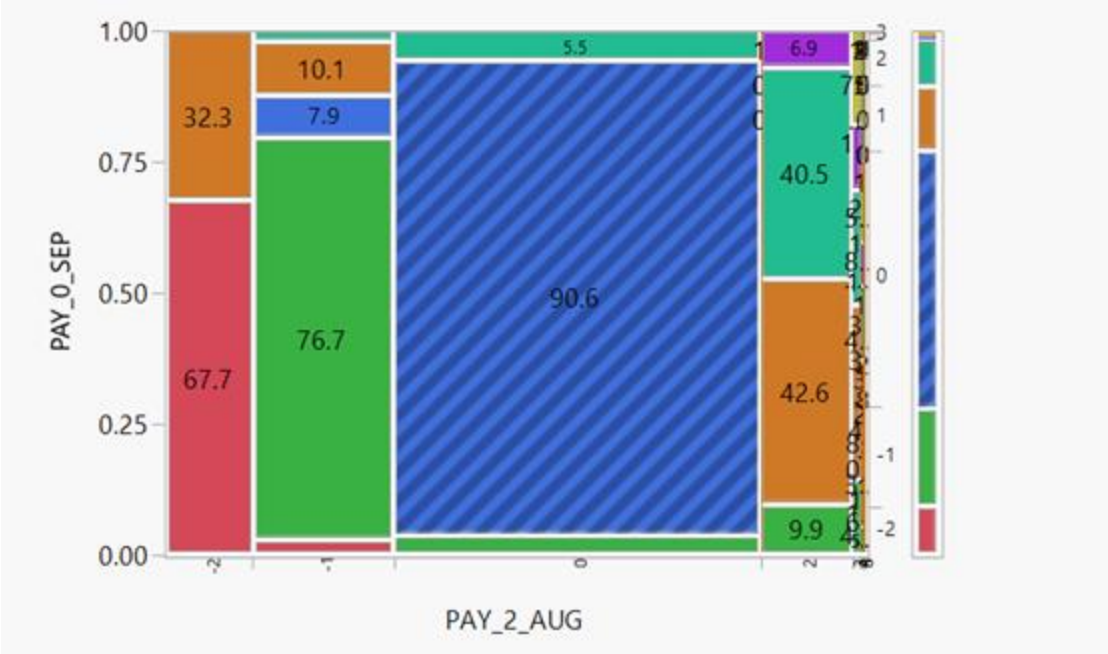
PAY Variable

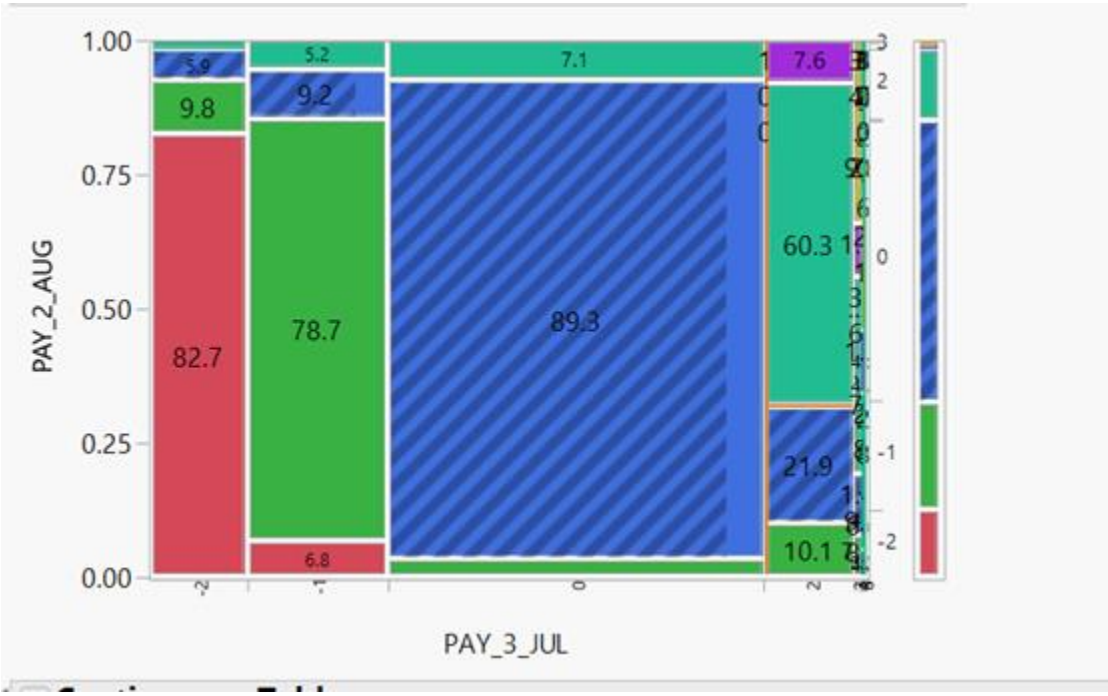
Data type | Modeling data type: Numeric | Ordinal

Analysis: There are six pay variables. Each variable represents repayment status for each month from September, 2005 to April 2006.
The variable values are (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months, ... 8=payment delay for eight months, 9=payment delay for nine months and above).

There are values which says status as 0 and -2 . These are considered as missing values.

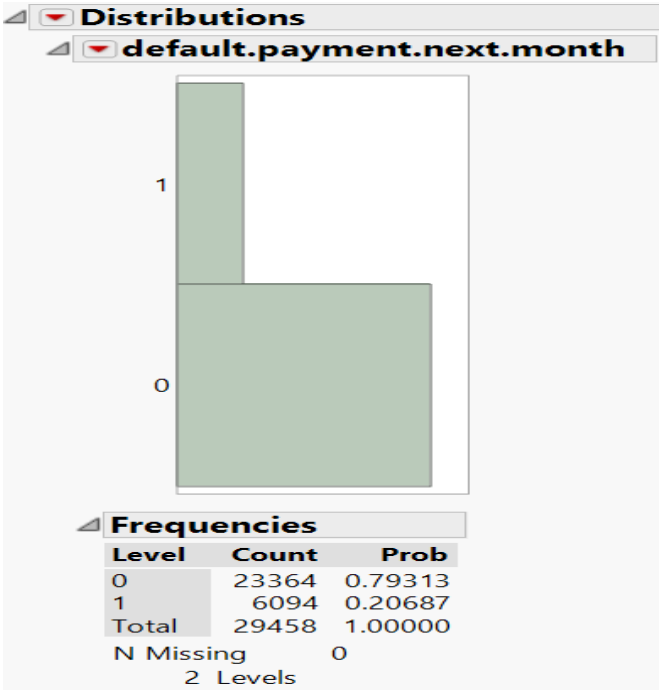
1) Checking the correlation between Pay variables





Default.payment.next.month

Data type | Modeling data type: Numeric | Nominal



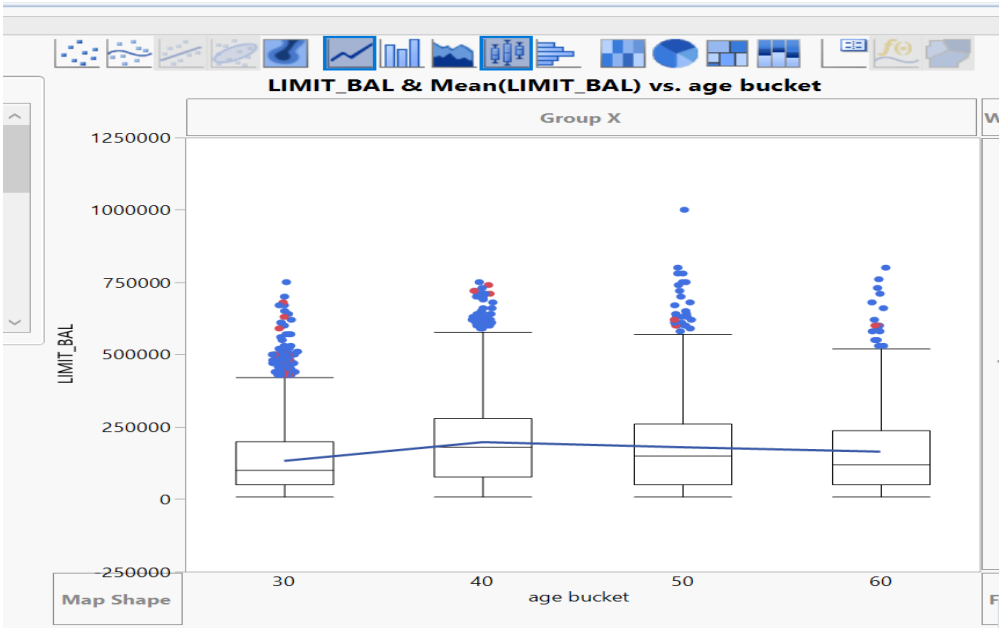
Analysis: Number of customers who defaulted (1's) is lesser than number of who did not default (0's). There are around 20% defaulters in the data.

Multivariate Analysis

Now we will see relation between

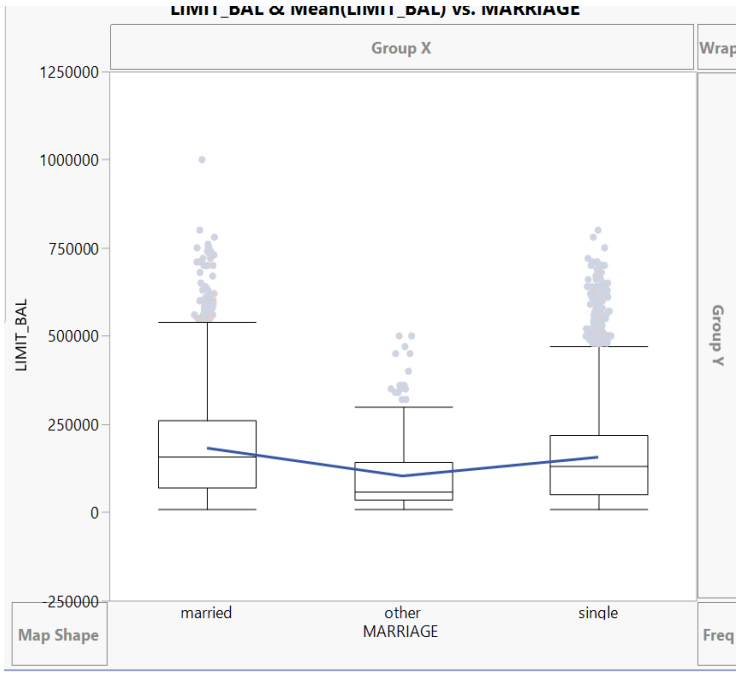
- Two or more predictor variables
- Predictor variable and target response variable

1) Analysis between: Personal Balance Limit and Age



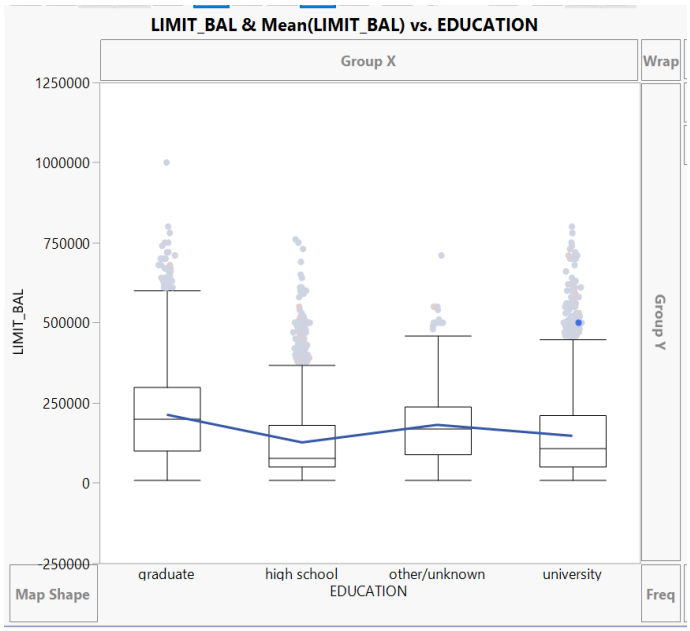
Analysis - People with age between 31 and 40 have higher limit than people in other age bands.

2) Analyzing relation between Limit Balance and Marital Status



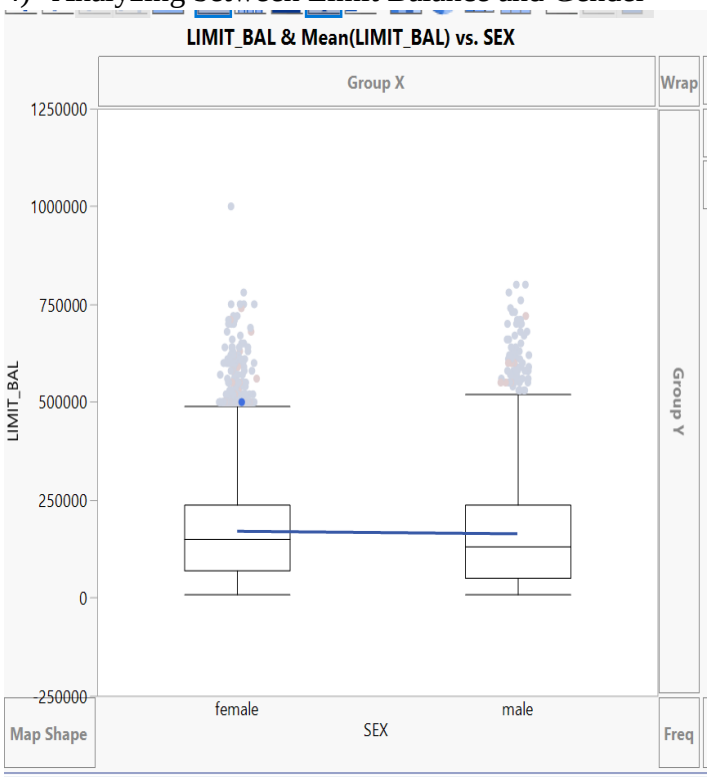
Analysis – Married customers on an average are getting a slightly higher limit balance than single customers in the data.

3) Analyzing relation between Limit Balance and Education



Analysis – On an average, we see that in the data, customers having a graduate education have a higher limit balance than those with university education. Those with only a high school education have the lowest limit balances.
This finding also gels with intuitive thinking that higher educated customers are more likely to be better earning and hence their higher spending potential warrants a higher limit balance.

4) Analyzing between Limit Balance and Gender



Analysis: There is not much difference in the limit allocation to males vs females. Males get only a slightly more limit balance in general than females.

Analysis with target reponse variable

Before proceeding with the models, we will look into the relationship of the predictor variables with the default variable.

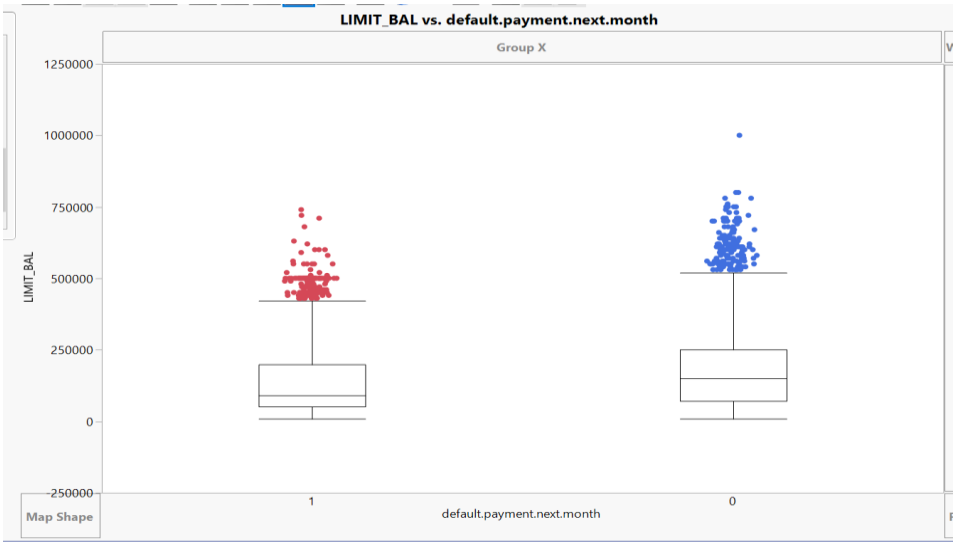
1) Analysis between: **default.payment.next.month** and **limit_bal**

   For purposes of analysis, we created limit balance buckets –
   <=50K, 50K-100K, 100K-150K, 150K-200K, >200K as 50K, 100K, 150K, 200K, 250K.
   Checking the mean default rate for each bucket.

Tabulate

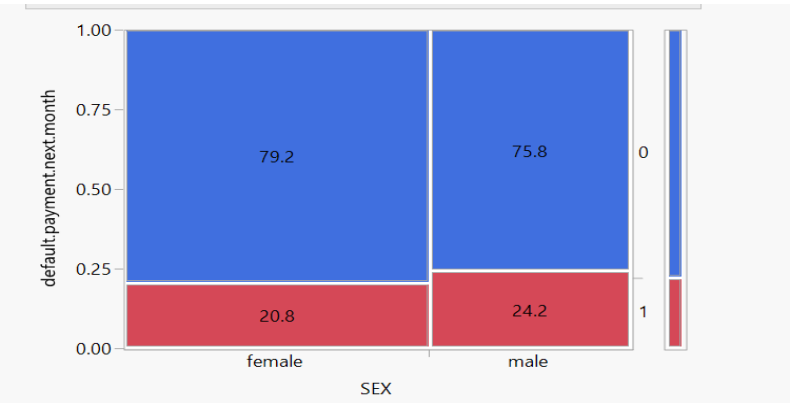
limit bucket	default.payment.next.month Mean
50000	0.312590258632
100000	0.2513078049801
150000	0.1984891898932
200000	0.1581658938691
250000	0.1216143881811

limit bucket	default.payment.next.month Mean
50000	
100000	
150000	
200000	
250000	



Analysis - From above table and graph, we can say that the customers with lower limit balance are more likely to default and there is negative correlation between limit balance and default.

2) Analysis between: **default.payment.next.month** and **Sex**



Frequencies

Level	Count	Prob
female	17756	0.60292
male	11694	0.39708
Total	29450	1.00000

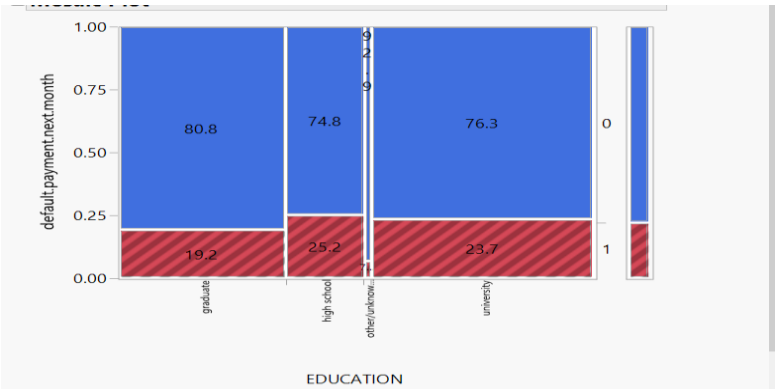
Analysis - Male customers are marginally more likely to default (24%) compared to female customers (21%).
There is no intuitive or straightforward study which may prove that males are in general more likely to default than females. And the marginal difference in the two populations prove that this may just be a coincidence.
In any case, we should be careful in using gender (sex) as a discriminatory variable in our prediction. Many regulatory agencies specifically forbid any discrimination on the basis of gender, age, marital status, education, sexual orientation, religion etc.

For the purposes of this prediction model, since nothing has been specified, we will be making use of the all available variables and data.

3) Analysis between: **default.payment.next.month** and **Education**

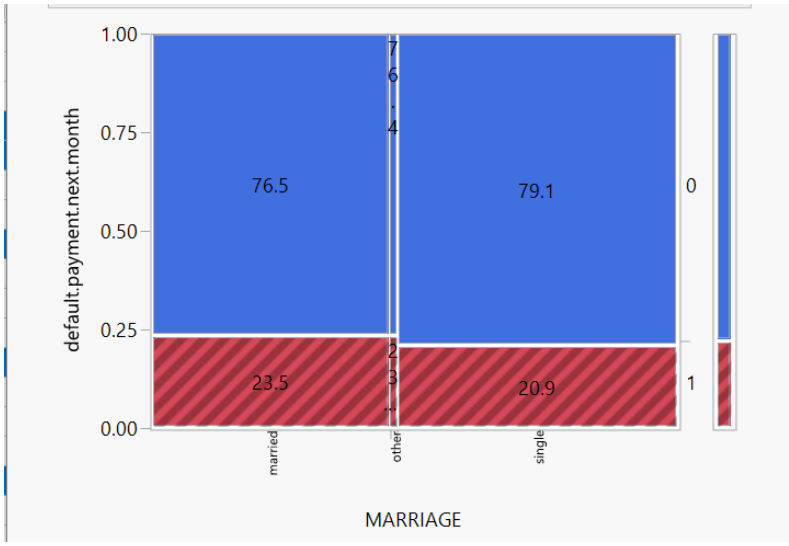
	default.payment.next.month
EDUCATION	**Mean**
graduate	0.1707246095645
high school	0.2387256930079
other	0.06852248394
university	0.2273252455228

	default.payment.next.month
EDUCATION	**Mean**
graduate	
high school	
other	
university	



Analysis – Within the three main categories – graduate school, university and high school – we see that customers with higher education are less likely to default. This can be explained from the fact that customers with more education would likely have more income and thus more financial strength. Thus they would be less likely to default.

4) Analysis between: **default.payment.next.month** and **Marital Status**



Analysis - This shows that in the given data, married customers are slightly more likely to default than single.

5) Analysis between: **default.payment.next.month** and **Age bucket**
To compare with age, we have made age buckets as following: <=30, 30-40, 40-50, >=50

The following relationship is observed

Tabulate

	default.payment.next.month
age bucket	Mean
30	0.2145287356322
40	0.188576868158
50	0.217598097503
60	0.2277498858969

	default.payment.next.month
age bucket	Mean
30	
40	
50	
60	

Analysis: - Except for lowest age group of <=30 age, as the age increase above that, the default rate increases.

The fact that the data shows that customers with age<=30 have a higher chance of default ties in with the fact that Taiwan in pre-2005 period (from which the data is based), faced these circumstances. Credit card companies used to give credit card to customers with little income requirements. College going students were also issued cards which led to them using these but not having the ability to repay.

6) Analysis between: **default.payment.next.month** and **Bill amount bucket**

We create buckets of Bill amounts as below.

```
If BILL_AMT1_SEP <= 0 ⇒ "<0"
   else ⇒ If BILL_AMT1_SEP <= 10000 ⇒ "0-10K"
            else ⇒ If BILL_AMT1_SEP <= 30000 ⇒ "10K-30K"
                     else ⇒ If BILL_AMT1_SEP <= 60000 ⇒ "30K-60K"
                              else ⇒ If BILL_AMT1_SEP <= 100000 ⇒ "60K-100K"
                                       else ⇒ ">100K"
```

Above is an example for Bill amount of Sep 2005.

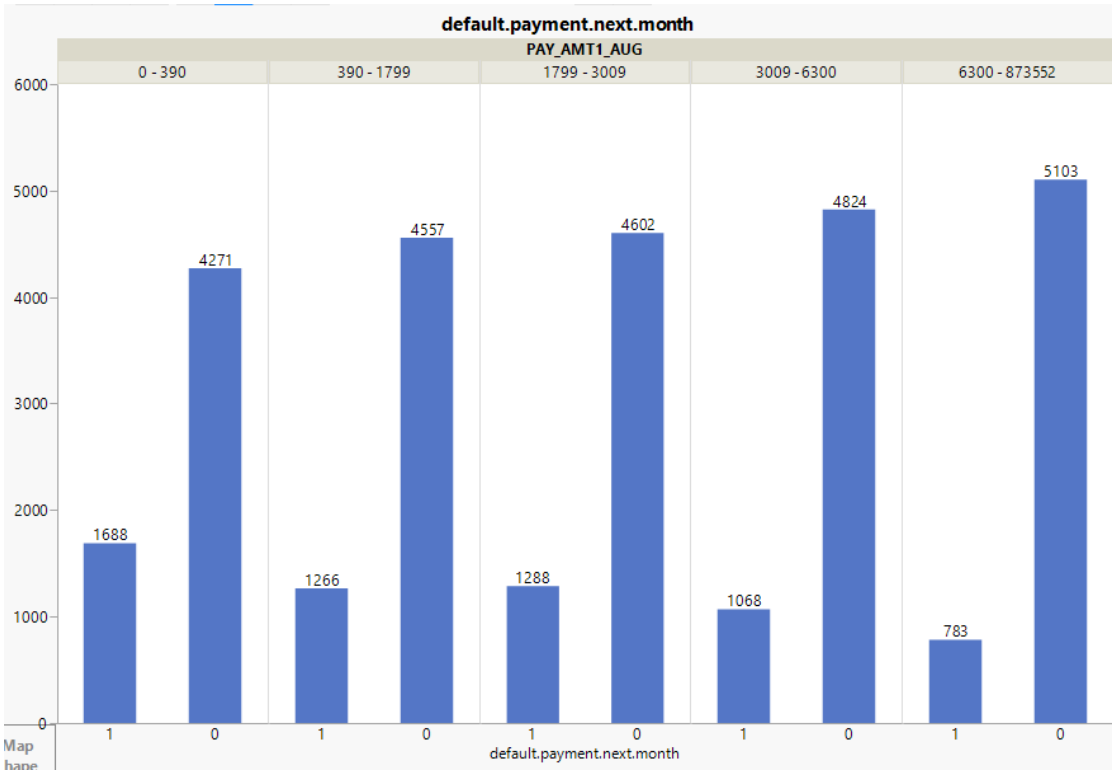
	default.payment.next.month
bill sep bucket	**Mean**
<0	0.0491245136187
>100K	0.195067264574
0-10K	0.216301320092
10K-30K	0.2427812811152
30K-60K	0.2209844034839
60K-100K	0.2124542124542

	default.payment.next.month
bill sep bucket	**Mean**
<0	
>100K	
0-10K	
10K-30K	
30K-60K	
60K-100K	

Analysis: Default rate is highest at medium bill amount range

7) Analysis between: **default.payment.next.month** and **Payment amount**

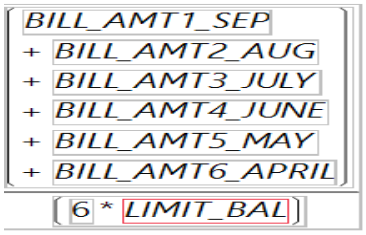
As an example, we see relation of default variable with Payment amount variable of Aug 2005.



Analysis: Payment amount and default rate has a good declining correlation. Default rate keeps increasing with decreasing payment amount.

8) Analysis between: **default.payment.next.month** and **Utilization**

We create utilization variable as ratio of 'Avg Bill amount per month' and 'Limit Balance as below.



Then we create utilization buckets

```
If
  utilization <= 1 ⇒ "<=1"
  else ⇒ If
    utilization <= 2 ⇒ "<=2"
    else ⇒ If
      utilization <= 3 ⇒ "<=3"
      else ⇒ If
        utilization <= 4 ⇒ "<=4"
        else ⇒ ">4"
```

Then we see the relation between utilization buckets and default rates.

util buckets	default.payment.next.month Mean
<=0.2	0.1457480497413
<=0.5	0.1946074834923
<=1	0.2809441565918
>1	0.3422291993721

util buckets	default.payment.next.month Mean
<=0.2	
<=0.5	
<=1	
>1	

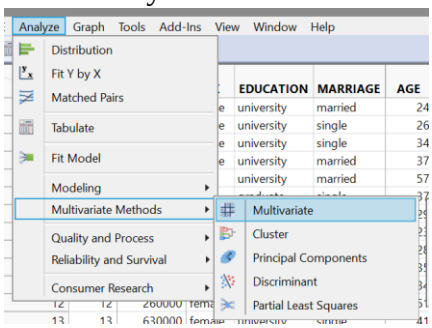
Analysis: Default rate increases with increasing utilization. This is on expected lines as customers who are more credit hungry to use up most of their available limit are more likely to default.

Correlation Analysis

We will conduct correlation analysis of the variables. This is done to identify which of the variables are useful in the model prediction. If the variables are highly correlated to each other, then only one of them gives us sufficient value in the model.

To check for correlation between variables, we follow the below steps.

Go to Analyze -> Multivariate Methods -> Select 'Multivariate'.



Select all the continuous variables you want as follows and click on 'Y, Columns' option in the dialog box and click 'OK'.

We can check all the correlations between all the variables using the correlations matrix and understand how each variable is related with others.

1) Correlation between Bill Amount variables

Correlations

	BILL_AMT1_SEP	BILL_AMT2_AUG	BILL_AMT3_JULY	BILL_AMT4_JUNE	BILL_AMT5_MAY	BILL_AMT6_APRIL
BILL_AMT1_SEP	1.0000	0.9511	0.8914	0.8591	0.8284	0.8011
BILL_AMT2_AUG	0.9511	1.0000	0.9278	0.8916	0.8586	0.8303
BILL_AMT3_JULY	0.8914	0.9278	1.0000	0.9234	0.8830	0.8523
BILL_AMT4_JUNE	0.8591	0.8916	0.9234	1.0000	0.9397	0.9003
BILL_AMT5_MAY	0.8284	0.8586	0.8830	0.9397	1.0000	0.9459
BILL_AMT6_APRIL	0.8011	0.8303	0.8523	0.9003	0.9459	1.0000

Each cell in a correlation matrix has a value between -1 and 1.
-1 means perfect negative correlation. 1 means perfect positive correlation. 0 means no correlation. The further the value is away from 0, the higher the correlation.

Analysis: All the cells have values greater than 0.8, which means that there is a very high correlation among all the bill amount variables.
This means if the bill amount is in a certain range in a month for a customer, then it is highly likely that the other month's bill amounts would also be in similar ranges.
Since the variables are highly dependent on each other, they convey similar value in the model. So we will use only of these variables in building the model.

2) Correlation between Pay Amount variables

Multivariate

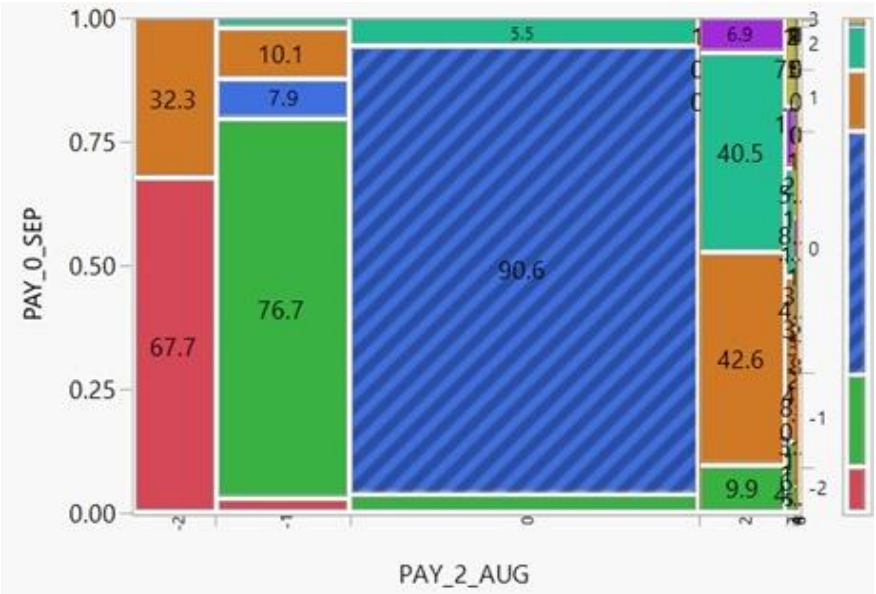
Correlations

	PAY_AMT1_AUG	PAY_AMT2_JULY	PAY_AMT3_JUN	PAY_AMT4_MAY	PAY_AMT5_APR	PAY_AMT6_MAR
PAY_AMT1_AUG	1.0000	0.2856	0.2522	0.1996	0.1485	0.1857
PAY_AMT2_JULY	0.2856	1.0000	0.2448	0.1801	0.1809	0.1576
PAY_AMT3_JUN	0.2522	0.2448	1.0000	0.2163	0.1592	0.1627
PAY_AMT4_MAY	0.1996	0.1801	0.2163	1.0000	0.1518	0.1578
PAY_AMT5_APR	0.1485	0.1809	0.1592	0.1518	1.0000	0.1549
PAY_AMT6_MAR	0.1857	0.1576	0.1627	0.1578	0.1549	1.0000

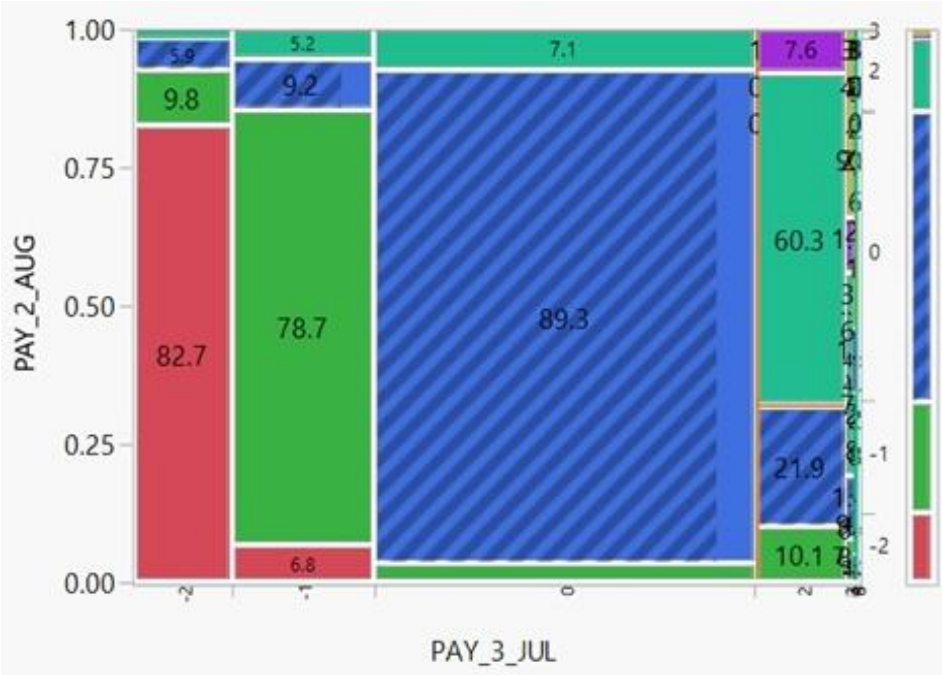
Analysis: Cell values are less than 0.3 – very low correlation.
We cannot predict pay amount value of any month on the basis of pay amount value in one month for a customer.
Based on this knowledge, it is useful to include all the pay amount variables in the model.

3) Correlation between Payment Status variables

Checking correlation between Payment Status variables of Sept and Aug months.



Checking correlation between Payment Status variables of Aug and July months.

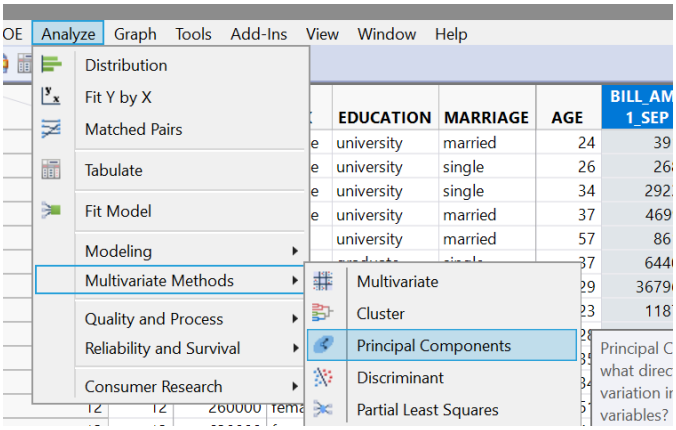


Principal Component Analysis

For variables with high correlation among them, it is useful to conduct Principal Component Analysis (PCA) on these variables.

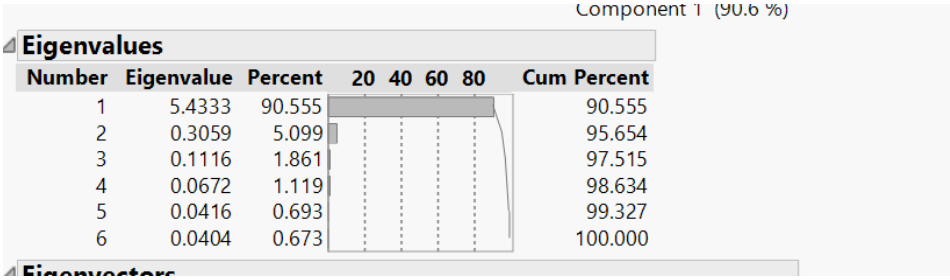
The steps are for this are as follows.

Select Analyze -> Multivariate Methods -> Principal Components



click on the small red triangle and click Eigen Value

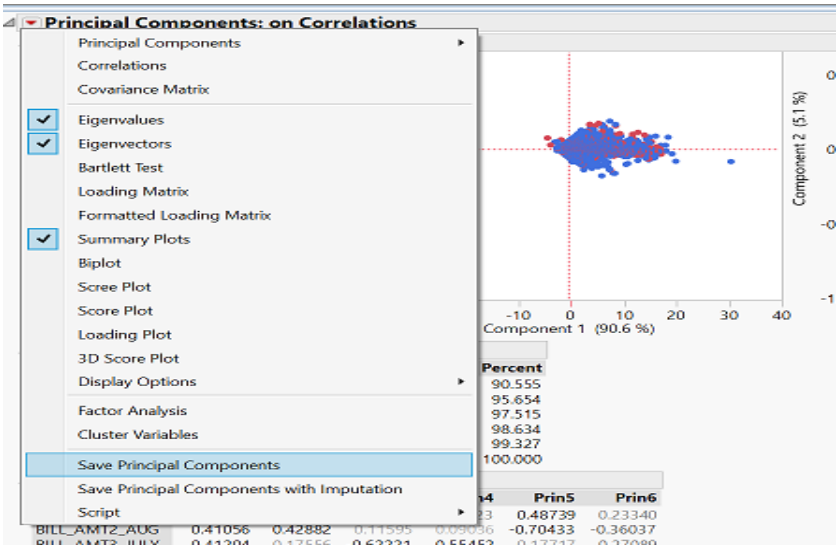
We will do PCA on the Bill amount variables as below.



Eigenvalues

Number	Eigenvalue	Percent	Cum Percent
1	5.4333	90.555	90.555
2	0.3059	5.099	95.654
3	0.1116	1.861	97.515
4	0.0672	1.119	98.634
5	0.0416	0.693	99.327
6	0.0404	0.673	100.000

Based on the output of the principal components analysis, we decided to retain two Principal Components which capture variability of 95.64% and help us reduce the considered 6 variables into 2 variables. To save the principal components click on the small red triangle next to the "Principal Components" and select Save "Principal Components" as follows-



The formulae for the two Principal components are as follows-

1. Principal Component 1

```
0.00000544403518
* BILL_AMT1_SEP
+ 0.00000576835206
  * BILL_AMT2_AUG
+ 0.00000594143756
  * BILL_AMT3_JULY
+ 0.00000644711748
  * BILL_AMT4_JUNE
+ 0.00000674852719
  * BILL_AMT5_MAY
+ 0.0000067292399
  * BILL_AMT6_APRIL
+ -1.6544108597521
```

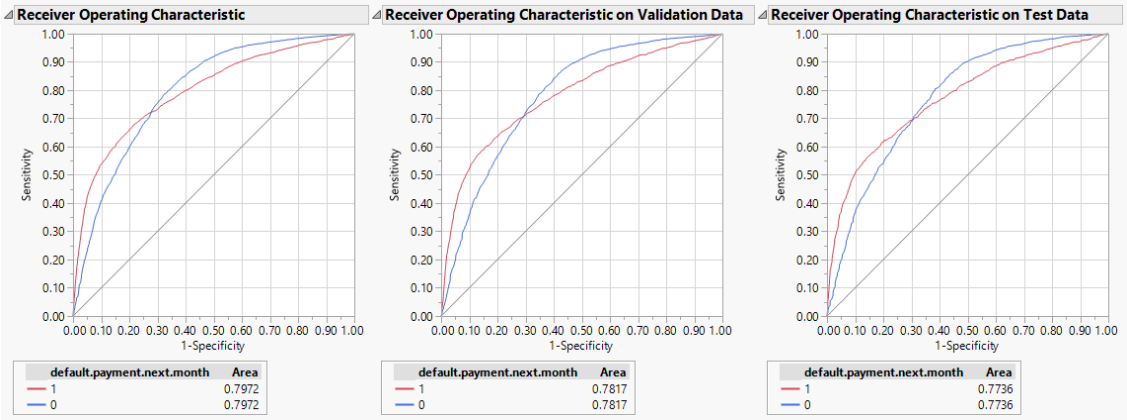
2. Principal Component 2

```
0.00000728533217
* BILL_AMT1_SEP
+ 0.00000602492578
  * BILL_AMT2_AUG
+ 0.0000025314633
  * BILL_AMT3_JULY
+ -0.0000028098685
  * BILL_AMT4_JUNE
+ -0.0000070747054
  * BILL_AMT5_MAY
+ -0.0000088824061
  * BILL_AMT6_APRIL
+ -0.0364620901123
```

5. Running Prediction Models

Here, we will run various models and compare them using metrics such as misclassification, accuracy, specificity, sensitivity, type 2 error and ROC.

(1) Logistic Model:



Confusion Matrix

Training

Actual default.payment.next.month	Predicted 1	Predicted 0
1	1208	1838
0	526	11153

Validation

Actual default.payment.next.month	Predicted 1	Predicted 0
1	708	1120
0	355	6652

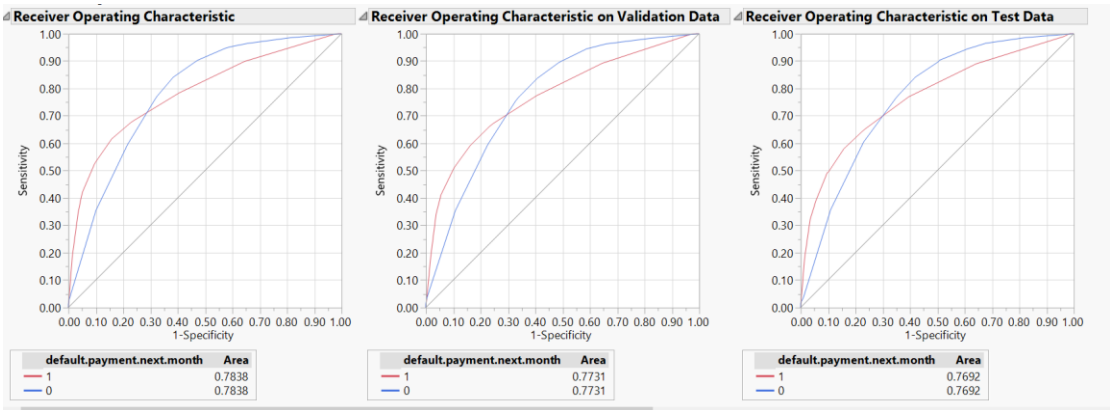
Test

Actual default.payment.next.month	Predicted 1	Predicted 0
1	439	780
0	208	4463

Analysis:

The model performed better in validation than training in Sensitivity metric. However, in other areas, validation was not better than training.

(2) Decision Tree Model



Fit Details

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.2090	0.1901	0.1837	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.3003	0.2757	0.2672	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.4032	0.4129	0.4162	∑ -Log(ρ[j])/n
RMSE	0.3520	0.3571	0.3589	√ ∑(y[j]-ρ[j])^2/n
Mean Abs Dev	0.2478	0.2512	0.2515	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.1607	0.1657	0.1698	∑ (ρ[j]≠ρMax)/n
N	14725	8835	5890	n

Confusion Matrix

Training

Actual default.payment.next.month	Predicted 1	Predicted 0
1	1248	1798
0	561	11118

Validation

Actual default.payment.next.month	Predicted 1	Predicted 0
1	730	1098
0	366	6641

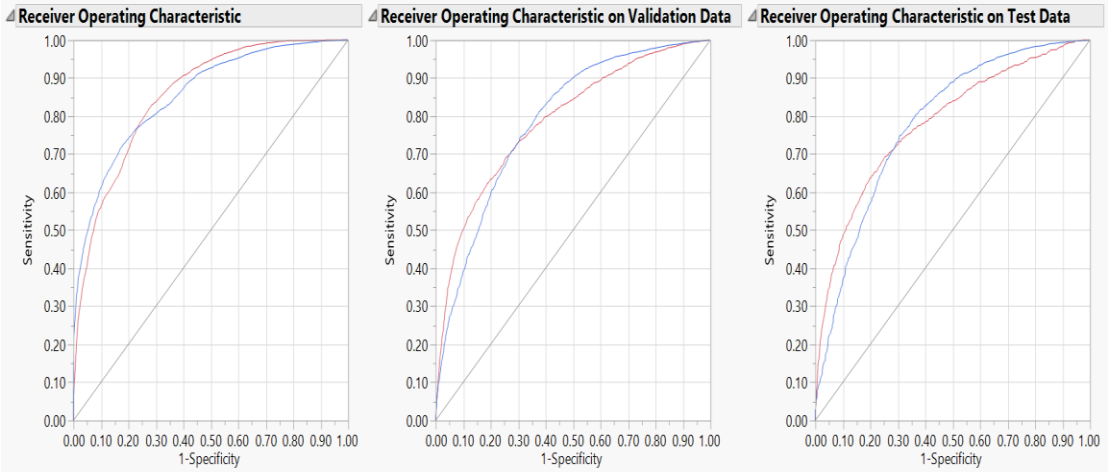
Test

Actual default.payment.next.month	Predicted 1	Predicted 0
1	456	763
0	234	4437

Analysis:
Model did not outperform in validation compared to training.

(3) Bootstrap Forest

ROC Curve



Overall Statistics

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.2801	0.2084	0.1997	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.3886	0.2995	0.2882	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.3670	0.4036	0.4081	∑ -Log(ρ[j])/n
RMSE	0.3380	0.3539	0.3557	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.2391	0.2506	0.2519	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.1574	0.1643	0.1677	∑ (ρ[j]≠ρMax)/n
N	14725	8835	5890	n

Confusion Matrix

Training

Actual default.payment.next.month	Predicted 0	Predicted 1
0	11141	538
1	1780	1266

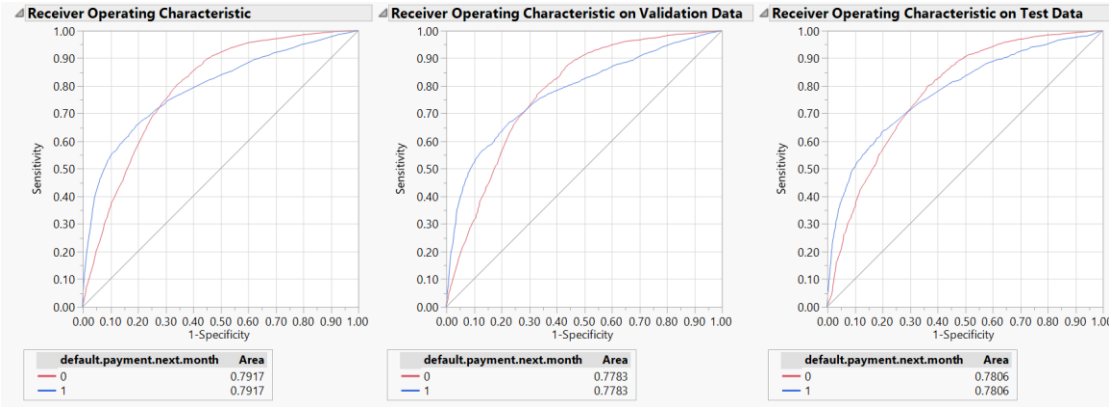
Validation

Actual default.payment.next.month	Predicted 0	Predicted 1
0	6652	355
1	1097	731

Test

Actual default.payment.next.month	Predicted 0	Predicted 1
0	4443	228
1	760	459

(4) Boosted tree



Overall Statistics

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.2094	0.1911	0.1867	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.3007	0.2770	0.2711	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.4030	0.4124	0.4147	∑ -Log(ρ[j])/n
RMSE	0.3517	0.3564	0.3578	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.2536	0.2577	0.2586	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.1597	0.1632	0.1667	∑ (ρ[j]≠ρMax)/n
N	14725	8835	5890	n

Confusion Matrix

Training

Actual	Predicted	
default.payment.next.month	0	1
0	11188	491
1	1861	1185

Validation

Actual	Predicted	
default.payment.next.month	0	1
0	6690	317
1	1125	703

Test

Actual	Predicted	
default.payment.next.month	0	1
0	4478	193
1	789	430

(5) Neural Model

Neural

Validation Column: Validation 2

Model Launch

Model NTanH(3)

Training

default.payment.next.month

Measures	Value
Generalized RSquare	0.2672837
Entropy RSquare	0.1837738
RMSE	0.3623493
Mean Abs Dev	0.2631588
Misclassification Rate	0.1874363
-LogLikelihood	6126.8348
Sum Freq	14725

Confusion Matrix

Actual	Predicted	
default.payment.next.month	1	0
1	912	2134
0	626	11053

Confusion Rates

Actual	Predicted Rate	
default.payment.next.month	1	0
1	0.299	0.701
0	0.054	0.946

Validation

default.payment.next.month

Measures	Value
Generalized RSquare	0.2306742
Entropy RSquare	0.1564665
RMSE	0.3684571
Mean Abs Dev	0.267733
Misclassification Rate	0.1926429
-LogLikelihood	3799.5402
Sum Freq	8835

Confusion Matrix

Actual	Predicted	
default.payment.next.month	1	0
1	542	1286
0	416	6591

Confusion Rates

Actual	Predicted Rate	
default.payment.next.month	1	0
1	0.296	0.704
0	0.059	0.941

Test

default.payment.next.month

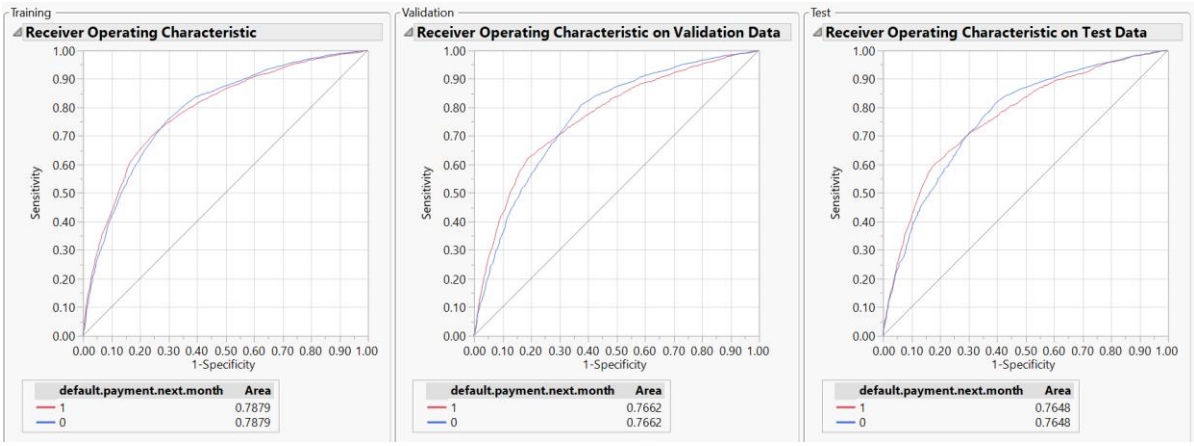
Measures	Value
Generalized RSquare	0.2274803
Entropy RSquare	0.1541118
RMSE	0.369429
Mean Abs Dev	0.2684826
Misclassification Rate	0.195416
-LogLikelihood	2540.4763
Sum Freq	5890

Confusion Matrix

Actual	Predicted	
default.payment.next.month	1	0
1	324	895
0	256	4415

Confusion Rates

Actual	Predicted Rate	
default.payment.next.month	1	0
1	0.266	0.734
0	0.055	0.945



6. Performance Evaluation

In this study, we have used the following performance measures:
Misclassification, accuracy, sensitivity, specificity.

Misclassification = It is a rate of false classified instances.

(FP+FN)/(TP+FP+TN+FN)

The lower the value, the better is the model.

Accuracy = It is a rate of true classified instances.

(TP+TN)/(TP+FP+TN+FN)
The higher the value, the better is the model.

Sensitivity= It is a rate of true positive classified instances.

TP /(TP +FN)
The higher the value, the better is the model.

Specificity=It is a ratio of true negative instances and total observed negative instances.

TN /(TN +FP)
The higher the value, the better is the model.

Additionally, we also look at Area under ROC curve and Type 2 error.
The higher the area under ROC, the better is the model.
Also for a better model, Type 2 error should be low.

Training

Model	Mean of Misclassification Rate	Mean of Accuracy	Mean of Sensitivity	Mean of Specificity	Mean of Area Under ROC	Mean of Type 2 error
Logistic regression	19.6%	80.4%	51.5%	92.2%	81.74%	1838
Decision Tree	16.1%	83.9%	40.97%	95.19%	78.38%	1798
Bootstrap Forest	15.7%	84.3%	41.6%	95.4%	85.55%	1780
Boosted Tree	16.0%	84.0%	38.9%	95.8%	79.17%	1760
Neural	18.7%	81.3%	29.9%	94.6%	78.79%	2134

Validation

Model	Mean of Misclassification Rate	Mean of Accuracy	Mean of Sensitivity	Mean of Specificity	Mean of Area Under ROC	Mean of Type 2 error
Logistic regression	19.6%	80.4%	54.1%	90.5%	80.10%	1120
Decision Tree	16.6%	83.4%	39.93%	94.77%	77.31%	1098
Bootstrap Forest	16.4%	83.6%	40.0%	94.9%	78.86%	1097
Boosted Tree	16.3%	83.7%	38.5%	95.5%	77.83%	1089
Neural	19.3%	80.7%	29.6%	94.1%	76.62%	1286

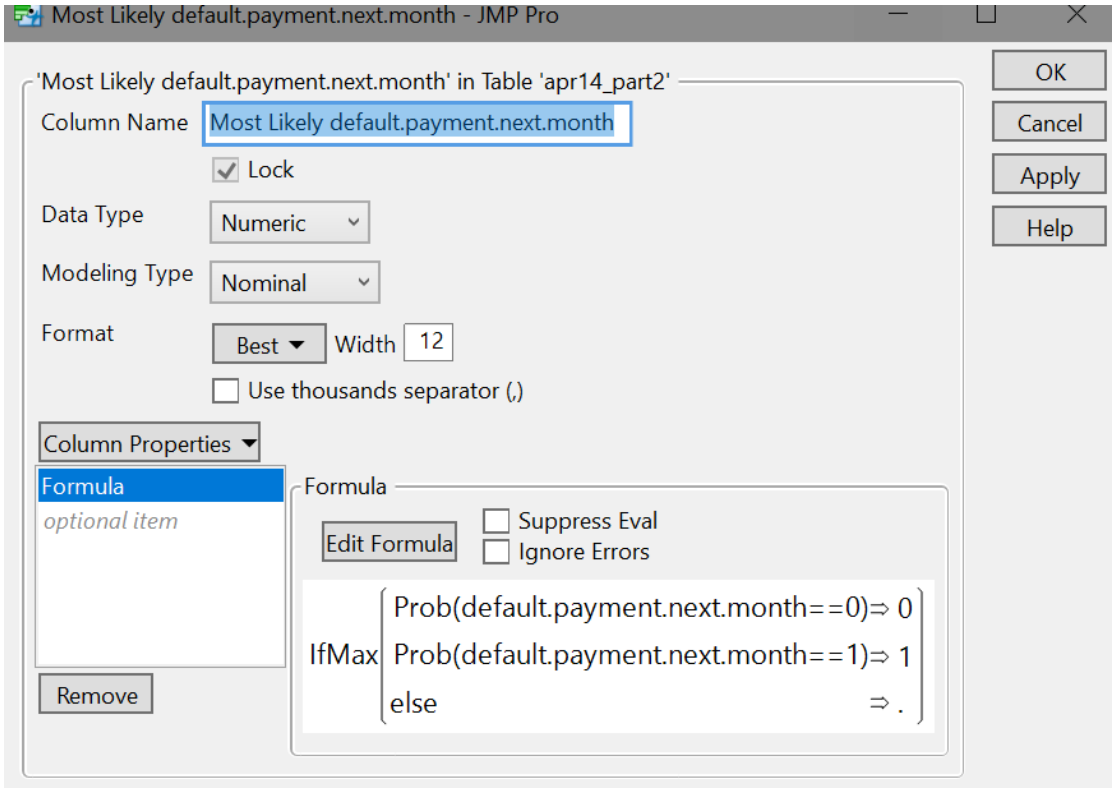
Test

Model	Mean of Misclassification Rate	Mean of Accuracy	Mean of Sensitivity	Mean of Specificity	Mean of Area Under ROC	Mean of Type 2 error
Logistic regression	20.7%	79.3%	49.1%	91.0%	78.74%	780
Decision Tree	17.0%	83.0%	37.40%	94.99%	76.92%	763
Bootstrap Forest	16.8%	83.2%	37.7%	95.1%	79.28%	760
Boosted Tree	16.7%	83.3%	35.3%	95.9%	78.06%	745
Neural	19.5%	80.5%	26.6%	94.5%	76.48%	895

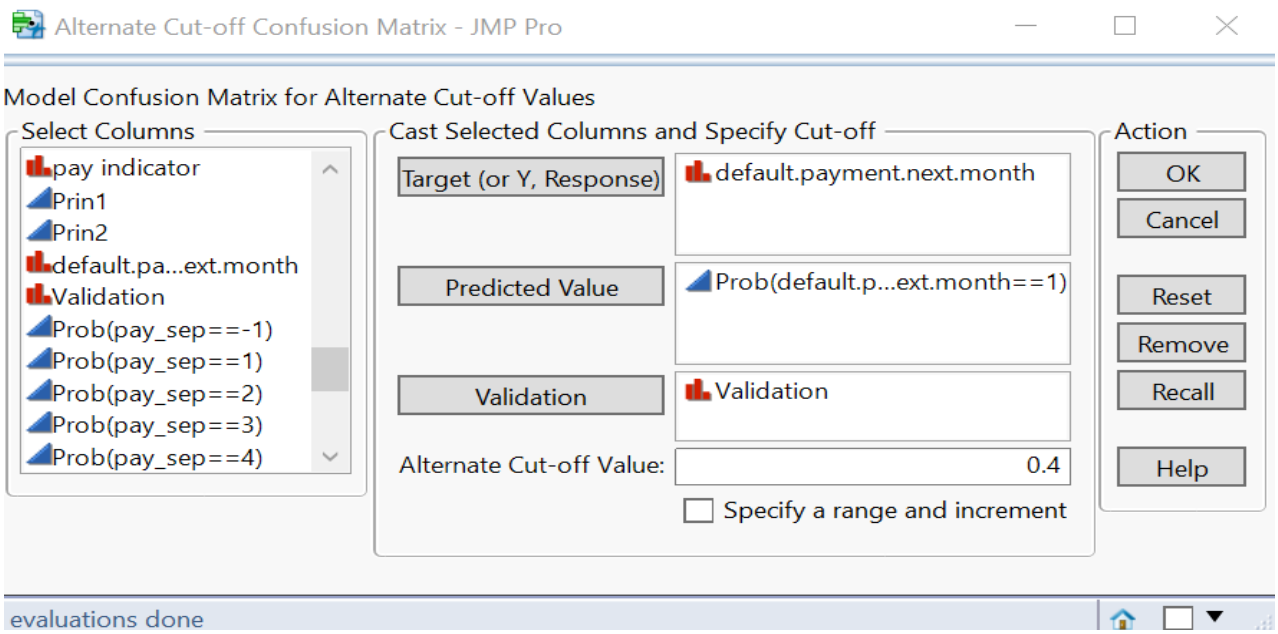
7. Conclusion

Comparing the different model metrics, we see that Bootstrap Forest model has the least Type 2 error and least misclassification error. Hence we choose this as our model for predicting card member bill default.

Model Building



Reducing the cut off



Confusion Matrix for Cut-off = 0.4

Target: default.payment.next.month
Predictor: Bootstrap Forest

	Modeling Data					
	Training		Validation		Test	
	Predicted		Predicted		Predicted	
default.payment.next.month	0	1	0	1	0	1
0	10900	779	6517	490	4353	318
1	1647	1399	982	846	688	531

8 rows have been excluded.

Confusion Rates: Training (0.4)

	Predicted	
	0	1
default.payment.next.month	Row %	Row %
0	93.33%	6.67%
1	54.07%	45.93%

Confusion Rates: Validation (0.4)

	Predicted	
	0	1
default.payment.next.month	Row %	Row %
0	93.01%	6.99%
1	53.72%	46.28%

Confusion Rates: Test (0.4)

	Predicted	
	0	1
default.payment.next.month	Row %	Row %
0	93.19%	6.81%
1	56.44%	43.56%

Analysis -

Confusion Matrix - JMP Pro

Confusion Matrix for Cut-off = 0.3

Target: default.payment.next.month
Predictor: Bootstrap Forest

	Modeling Data					
	Training		Validation		Test	
	Predicted		Predicted		Predicted	
default.payment.next.month	0	1	0	1	0	1
0	10235	1444	6110	897	4079	592
1	1285	1761	774	1054	534	685

8 rows have been excluded.

Confusion Rates: Training (0.3)

	Predicted	
	0	1
default.payment.next.month	Row %	Row %
0	87.64%	12.36%
1	42.19%	57.81%

Confusion Rates: Validation (0.3)

	Predicted	
	0	1
default.payment.next.month	Row %	Row %
0	87.20%	12.80%
1	42.34%	57.66%

Confusion Rates: Test (0.3)

	Predicted	
	0	1
default.payment.next.month	Row %	Row %
0	87.33%	12.67%
1	43.81%	56.19%

evaluations done

Future Scope:

For such models, certain other variables are highly useful in predicting cardmember default.
They are
(1) Income: This gives the debt capacity of an individual which gives an idea whether

a customer will default given the balance the customer owes.

(2) **Occupation:** Certain macro-economic factors may trigger a downturn in a particular sector leading to hardships for people belonging to certain occupations.

(3) **Number of Credit Cards:** This gives an idea about credit hungry nature of the customers.

(4) **Number of external delinquencies:** If the customer has been delinquent or default externally, the chances would be higher of a default internally as well.

(5) **Total size of wallet:** What is the total spend of the customer both internally and externally.

(6) **Share of wallet:** Out of the total spend by the customer, what is the amount spent by the customer with the company's credit card. If this number is low, this gives an indication that the customer is high spender externally. This can also have significant impact on default prediction.