NUS School of Computing
BT2103 Optimization Methods for Business Analytics
Group Project

Group 18	
Student Name	**Student Number**
Jake Khoo	A0217099W
Lee Lin Yee	A0238736U
Megan Poh Gin Yee	A0239807U
Tan Sin Ler	A0240651N

Table of Contents

1. Introduction **3**
2. Exploratory Data Analysis **4**
2.1. Dependent Variable 4
2.2. Independent Variables 4
3. Data pre-processing **10**
4. Model 1: Logistic Regression **11**
4.1. Final Model 11
5. Model 2: Support Vector Machines **12**
5.1. Hyperparameter tuning 12
5.2. Final Model 13
6. Model 3: K-Nearest Neighbours **13**
6.1. Hyperparameter tuning 13
6.2. Final Model 13
7. Model 4: Decision Tree **14**
7.1. Final Model 14
8. Model 5: Random Forests **15**
8.1. Feature Selection 15
8.2. Hyperparameter tuning 16
8.3. Final Model 16
9. Model 6: Neural Network **17**
9.1. Hidden Layer Selection 17
9.2. Hyperparameter tuning 17
10. Model Evaluation and Conclusion **18**
11. References **19**

1. Introduction

In Taiwan, there are many customers who overspend on their credit cards, thus increasing their debt and resulting in late payments to the bank. Hence, to prevent the accumulation of debt, it is important for the bank to accurately estimate customers' credit risk and prevent customers from spending outside their means. This is done by analyzing the information of 30000 bank customers in Taiwan and predicting whether they would default. The following are the 23 attributes used for prediction:

```
LIMIT_BAL (X1) - amount of credit that the customer and the customer's family has (New Taiwan Dollar)
SEX (X2) - gender <!-- 1 = male; 2 = female -->
EDUCATION (X3) - education <!-- 1 = graduate school; 2 = university; 3 = high school; 4 = others -->
MARRIAGE (X4) - marital status <!-- 1 = married; 2 = single; 3 = others -->
AGE (X5) - age (year)

PAY_0 (X6) - how late was the payment for September 2005
PAY_2 (X7) - how late was the payment for August 2005
PAY_3 (X8) - how late was the payment for July 2005
PAY_4 (X9) - how late was the payment for June 2005
PAY_5 (X10) - how late was the payment for May 2005
PAY_6 (X11) - how late was the payment for April 2005
<!-- for PAY_0 to PAY_6, -1 = paid on time; 1 = paid late by one month;  -->
<!-- 2 = paid late by two months; . . .; 8 = paid late by eight months;  -->
<!-- 9 = paid late by nine months and above -->

BILL_AMT1 (X12) - amount of bill statement in September 2005 (NT Dollar)
BILL_AMT2 (X13) - amount of bill statement in August 2005 (NT Dollar)
BILL_AMT3 (X14) - amount of bill statement in July 2005 (NT Dollar)
BILL_AMT4 (X15) - amount of bill statement in June 2005 (NT Dollar)
BILL_AMT5 (X16) - amount of bill statement in May 2005 (NT Dollar)
BILL_AMT6 (X17) - amount of bill statement in April 2005 (NT Dollar)

PAY_AMT1 (X18) - amount of previous payment in September 2005 (NT Dollar)
PAY_AMT2 (X19) - amount of previous payment in August 2005 (NT Dollar)
PAY_AMT3 (X20) - amount of previous payment in July 2005 (NT Dollar)
PAY_AMT4 (X21) - amount of previous payment in June 2005 (NT Dollar)
PAY_AMT5 (X22) - amount of previous payment in May 2005 (NT Dollar)
PAY_AMT6 (X23) - amount of previous payment in April 2005 (NT Dollar)
```

The target feature column Y is whether the bank customer does default payment next month and is predicted to be binary valued 0 (= not default) or 1 (= default).

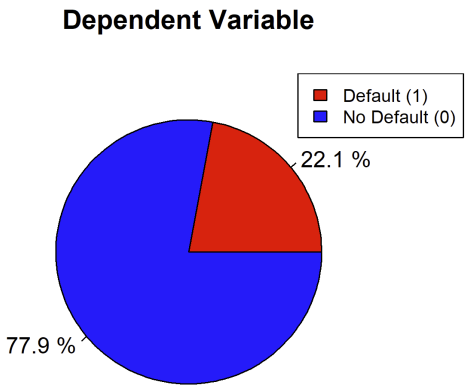
(*Default of Credit Card Clients Data Set*, 2016)

Additionally, in the description of the dataset, the PAY variables was described to represent the repayment status in different months, ranging from April to September 2005. "The measurement scale for the repayment scale is : -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. However, our group observed other values such as 0 and -2 within the dataset.

2. Exploratory Data Analysis

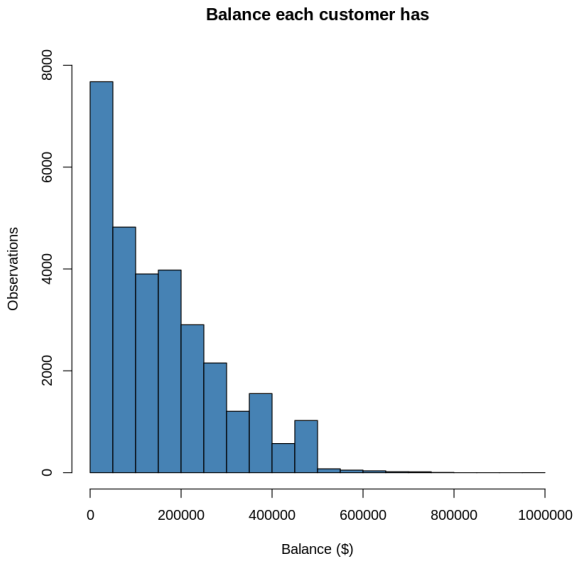
There are 30,000 observations in the dataset.

2.1. Dependent Variable

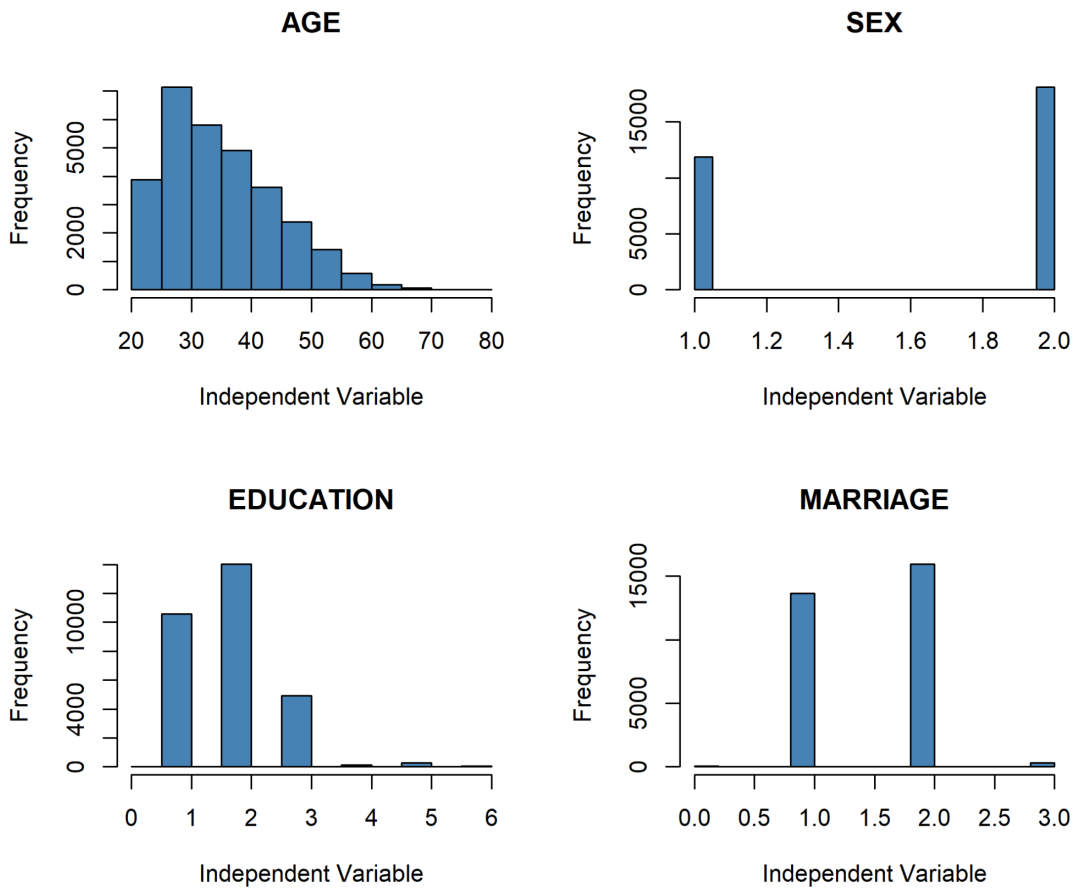


The Pie Chart on the left shows the distribution of whether a bank customer defaults. Based on the chart we can see that it is skewed to non-defaulting which may have implications in our model evaluation. We take positive as 1 and negative as 0 in our confusion matrix.

2.2. Independent Variables



The histogram for the attribute LIMIT_BALANCE as shown on the left reflects that as the amount of balance increases, the fewer the number of customers who have that amount of balance, with the majority having less than 200,000 NT dollars.

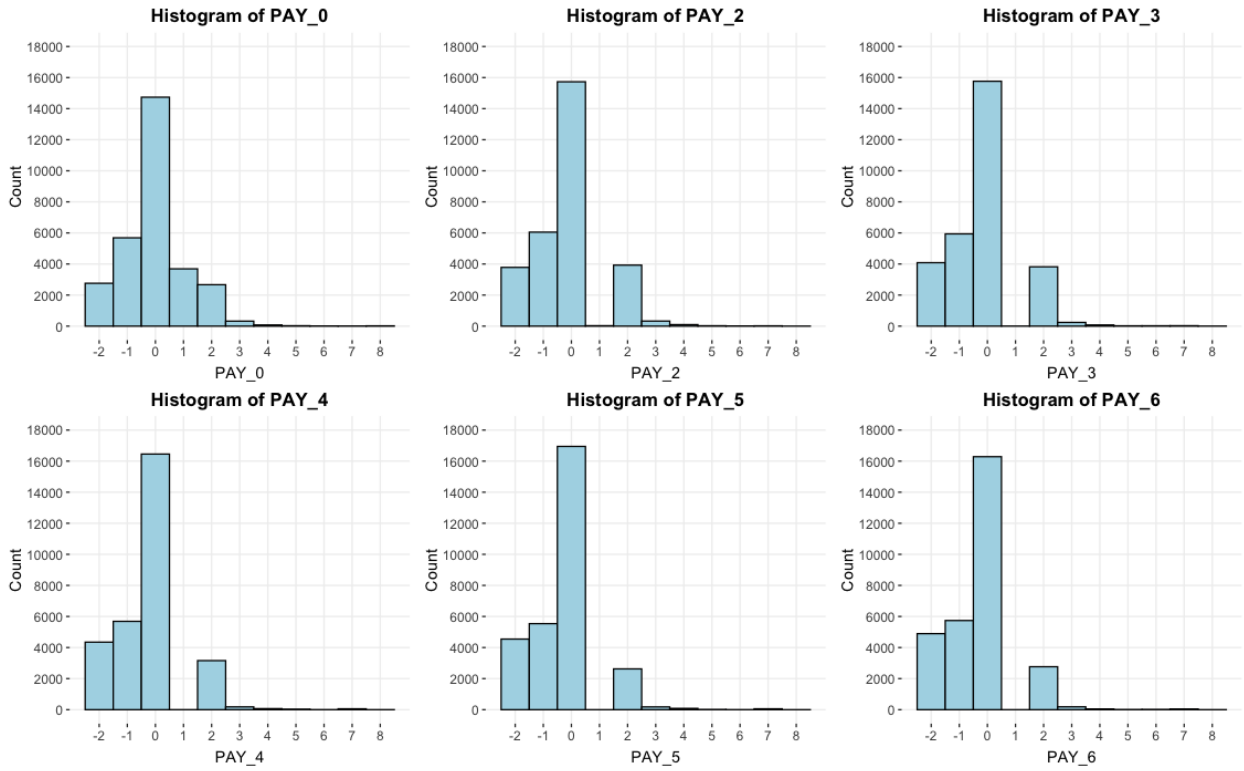


For AGE frequency plot, there is a skew towards the younger age groups observations.

For SEX frequency plot, there are more females observed (2 = female) than males (1 = male).

For EDUCATION frequency plot, the highest frequency of observations comes from the university group (2 = university) compared to the other education levels.

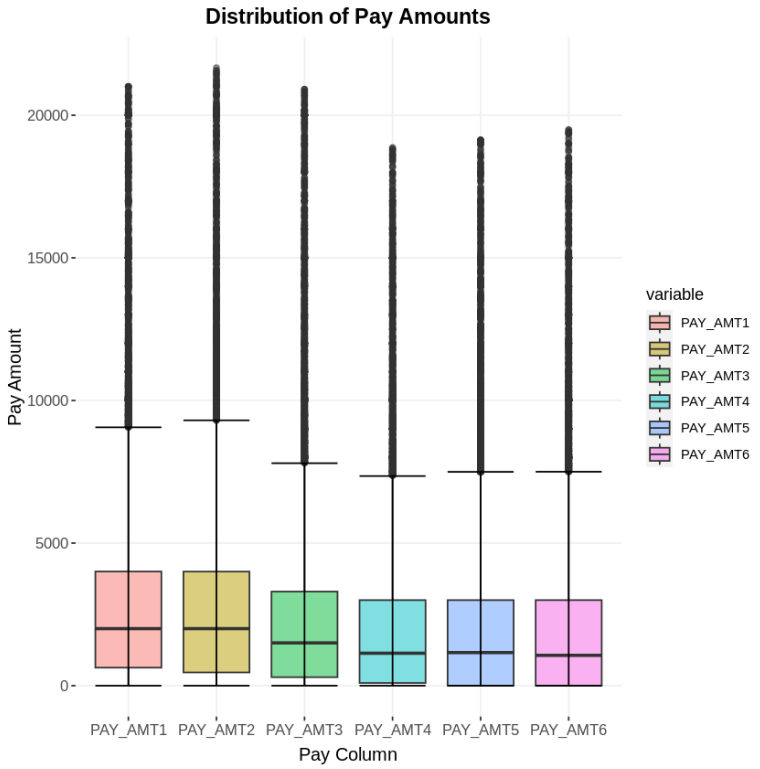
For MARRIAGE frequency plot, majority of the datapoints belong to either the single group (2 = single) or the married group (1 = married). Between them, there is only a slightly higher frequency of observations coming from the single group as opposed to the married group.



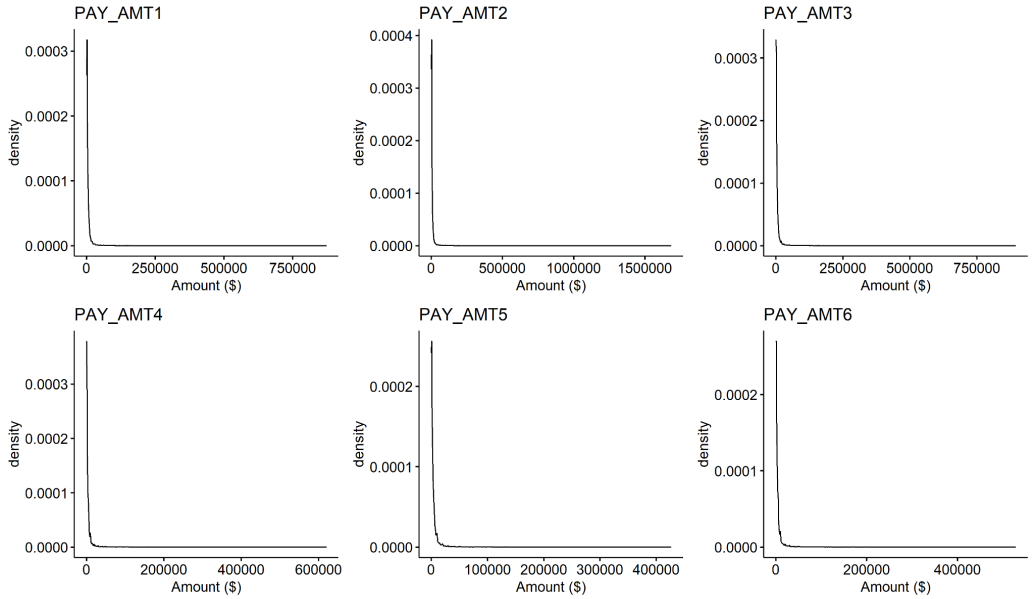
The figure above show the frequency plots for each of the PAY attributes, namely how late the payment was from April to September 2005.

Frequency plot of PAY_0 depicts the distribution of how late the payment was for September 2005, with there being a highest amount of people paying on time but but the amount paid is not enough to clear the total balance and the customer is meeting the minimum payment (= 0)

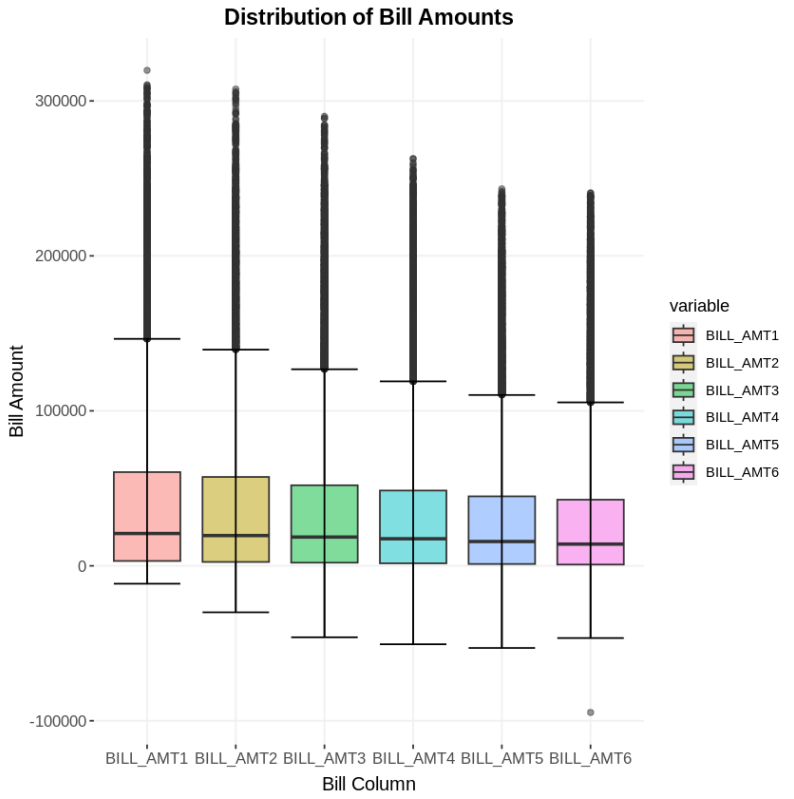
Similar results were found for the frequency plots of PAY_2, PAY_3, PAY_4, PAY_5 and PAY_6 which depicts the distribution of how late the payment was for August, July, June, May and April of 2005 respectively, with there being a highest amount of people paying on time but but the amount paid is not enough to clear the total balance and the customer is meeting the minimum payment (= 0) as well.



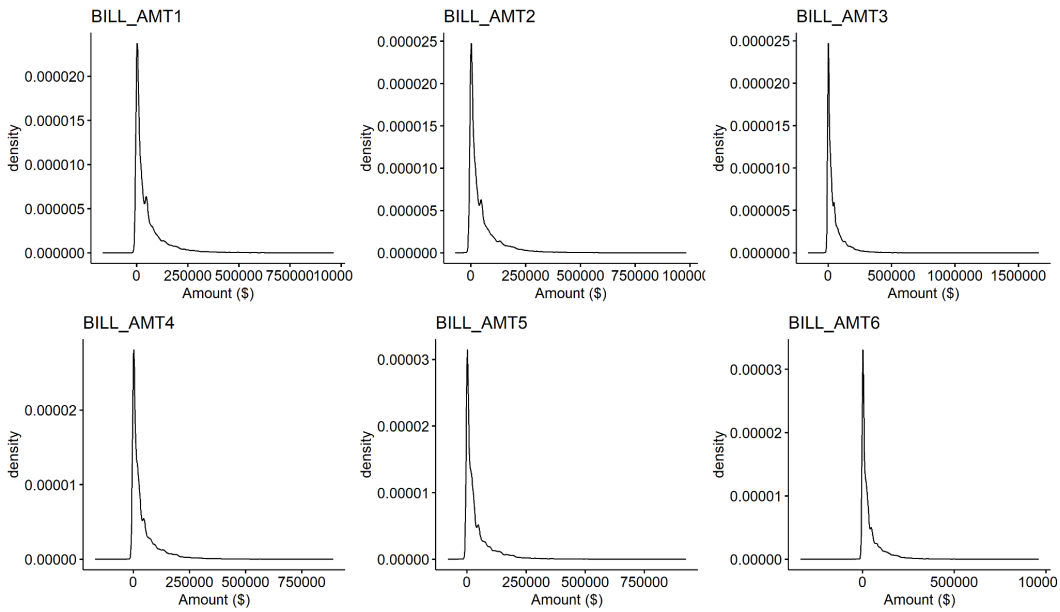
The figure above show box plots of PAY_AMT after removing outliers (0.75 + 4*IQR). We observe that the values have a decreasing trend from 1 to 6. Since PAY_AMT1 is the latest month, this could show that the customers are trying to pay off their bills after realising that they owe more money. There is a large number of outliers.



Since the distributions of the PAY_AMT variables are not normal, our group did not normalise the data. There is a wide distribution in PAY_AMT however, most people have very low PAY_AMT.

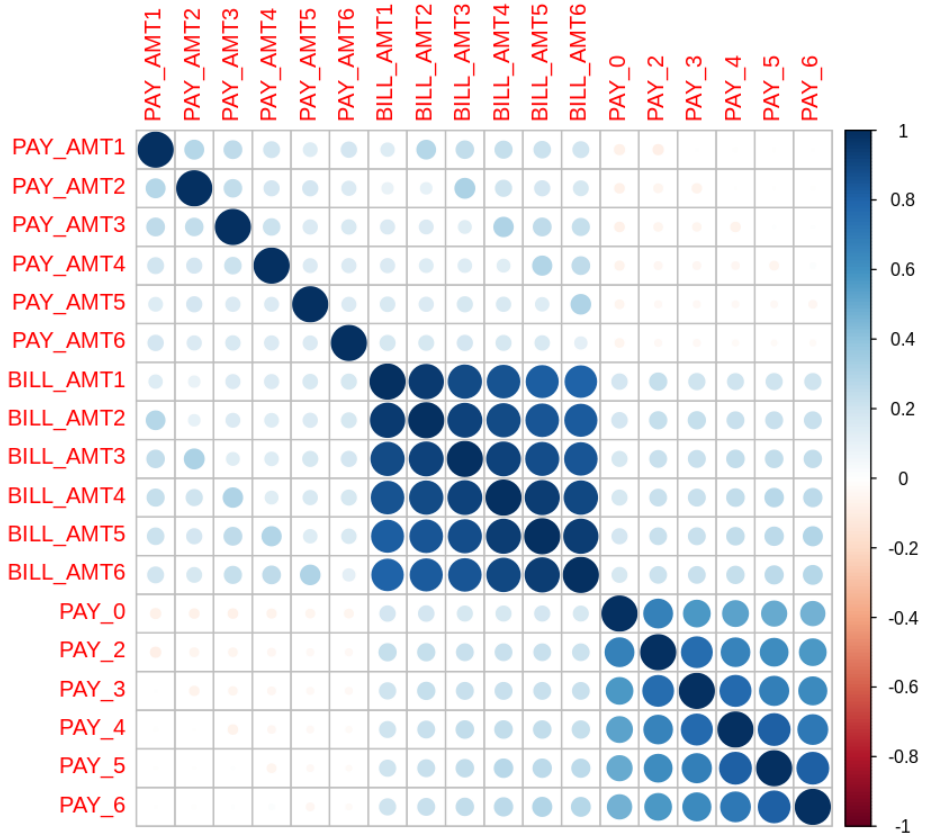


The figure above shows box plots of BILL_AMT after removing outliers (0.75 + 4*IQR). It is weird that there are negative bill values but this may be due to customers overpaying the previous months. The mean for BILL_AMT across the month stays constant across the months but the 75 percentile shows an increasing rate from the first to the last month.



Since the distributions of the BILL_AMT variables are not normal, our group did not normalise the data.

Correlation plot



We decided to plot the PAY_AMTs, BILL_AMTs and PAYs data as we thought that they should be correlated because high bills should lead to high pay amounts for the average user to pay off their bills. High bills should also lead to high pays because users are unable to pay off their bills.

There are several interesting observations from the correlation plots.

Firstly, the bill amounts are strongly correlated between each other with r values at around 0.8. Hence, we decided to remove 5 BILL_AMT variables and keep 1. We decided to retain BILL_AMT3 as it is strongly correlated with the other bill amounts. Highly correlated features are unlikely to present new information to the machine learning models.

There is also a pattern of correlation between PAY_AMT(i) and BILL_AMT(i+1). Although the r value is not very high around 0.4, the pattern stands out and an explanation for this may be that because users pay their bills the next month, a high bill will lead to a high pay for the next month.

Lastly, We thought that PAY will have a strong correlation with each other however, the values are not significant enough to remove as they may provide new information.

3. Data pre-processing

In the description of the dataset, EDUCATION was described to have 4 values - 1 for graduate school, 2 for university, 3 for high school and 4 for others. Our group observed other values such as 0, 5 and 6 within the dataset. We decided to change all these vaguely categorised datapoints to have the EDUCATION value of 4 so that they can be classified as “Others”

In the description of the dataset, MARRIAGE was described to have 3 values - 1 for married, 2 for single, and 3 for others. Our group observed other values such as 0 within the dataset. We decided to change all these vaguely categorised datapoints to have the MARRIAGE value of 3 so that they can be classified as “Others”

When looking at the dataset, our group noticed that one of the variables were named strangely., where the PAY variables skipped a numerical value from PAY_0 to PAY_2. To allow the dataset to look more coherent, our group decided to change the column name from PAY_0 to PAY_1.

In the description of the dataset, the PAY variables was described to represent the repayment status in different months, ranging from April to September 2005. “The measurement scale for the repayment scale is : -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. However, our group observed other values such as 0 and -2 within the dataset. Upon further investigation of the dataset, we realized that generally, datapoints with value 0 corresponds to a customer who has made a payment on time, but the amount paid is not enough to clear the total balance, and the customer is meeting the minimum payment. On the other hand, datapoints with value -2 corresponds to a customer with no credit to pay. Our group decided to group all datapoints with a `PAY` value of 0, -1 and -2 together. To let the values seem more coherent, we decided to change the end value of the variable to 0.

We decided to remove BILL_AMTs 1,2,4,5,6 because of the strong correlation and lack of additional information if we choose to include one additional bill amount.

One of the major issues that novice users fall into when dealing with unbalanced datasets relates to the metrics used to evaluate their model. Using simpler metrics like accuracy_score can be misleading. In a dataset with highly unbalanced classes, if the classifier always "predicts" the most common class without performing any analysis of the features, it will still have an accuracy of 77.9, which is misleadingly high.

One way to deal with this issue was to use over and undersampling methods on the train data. We decided to use both on the train data. Initially the train test split is 75 - 25. With 22500 values for the train data and 7500 for the test data. Since the dataset is unbalanced, our group used ovun sampling from the ROSE database. Through this method, minority examples will be oversampled, while majority examples are undersampled. After the ovum sampling, there would be 7,500 data points left in the train set creating a 50 - 50 split. We decided to reduce the train set as the time complexity for some models were very large. We only sampled the train set because the test set contains the real world scenario and the model should be tested on the real world scenarios. However, there are many schools of thought on sampling before or after data splitting.

4. Model 1: Logistic Regression

4.1. Final Model

We used logistic regression to train a model and used the model to predict whether customers would default in the test dataset. Setting the threshold to 0.5, meaning that we predict a customer would default when the probability is more than 0.5, we obtain the following confusion matrix

Actual / Pred	1	0
1	TP = 952	FN = 713
0	FP = 890	TN = 4945

From the above matrix, we can obtain the following summary statistics:

Sensitivity = TP / (TP+FN) = 0.572
Accuracy = (TP+TN)/(TP+TN+FP+FN) = 0.786
Precision = TP / (TP + FP) = 0.517
False Negative Rate = FN / (FN + TP) = 0.428
Harmonic Mean = 0.650
Average Class Accuracy = 0.695
F1 Score = 0.543

As the cost of identifying a default customer as non-default is higher than identifying a non-default customer as default, we seek to increase the sensitivity value of the model. This can be done by decreasing the threshold value. When the threshold value is decreased to 0.3, we obtain the following confusion matrix.

Actual / Pred	1	0
1	TP = 1571	FN = 94
0	FP = 4692	TN = 1143

Sensitivity = TP / (TP+FN) = 0.944
Accuracy = (TP+TN)/(TP+TN+FP+FN) = 0.362
Precision = TP / (TP + FP) = 0.251
False Negative Rate = FN / (FN + TP) = 0.0565
Harmonic Mean = 0.395
Average Class Accuracy = 0.587
F1 Score = 0.397

We can see that decreasing the threshold from 0.5 to 0.3 increases the sensitivity from 0.58 to 0.94 and decreases the FNR from 0.41 to 0.064. With this change in threshold value, we are now more confident that the model will be able to accurately flag out customers who would default.

5. Model 2: Support Vector Machines

5.1. Hyperparameter tuning

There are many different parameters that we can fine-tune in an SVM model, including the cost, kernel, weights, and whether cross-modelling is utilised or not.

Firstly, our group tried to use different cost values. The cost parameter determines the penalty for misclassifying training examples. When cost is low, the model allows for more misclassifications in the train set. When cost is high, misclassifications will be penalised heavily. Note that the cost parameter cannot be too high to avoid overfitting.

	0.1	1	10
Accuracy	0.7931	0.8055	0.7991
Precision	0.5347	0.5712	0.5506
Sensitivity	0.5231	0.4961	0.5165
False Negative Rate	0.4769	0.5039	0.4835
Harmonic Mean	0.6608	0.6869	0.6727
Avg Class Accuracy	0.6997	0.7163	0.7075

From the statistics computed, our group decided that a **cost of 1** would be optimal for the prediction.

Next, our group also used common kernels ("linear", "radial", "polynomial", "sigmoid") to run the svm model. Linear kernel is used when the data is separable by a linear boundary; Polynomial kernel (radial kernel) transforms the input data into a higher-dimensional (infinite-dimensional) feature space, where the similarity between two data points is measured by the polynomial (Gaussian) function. Lastly, the sigmoid kernel is used when the input data is not separable by a linear or polynomial boundary. A sigmoid function is used to transform the input data.

	linear	radial	polynomial	sigmoid
Accuracy	0.8055	0.7599	0.7909	0.6548
Precision	0.5712	0.4687	0.5303	0.3420
Sensitivity	0.4961	0.6120	0.5093	0.6006
False Negative Rate	0.5039	0.3880	0.4907	0.3994
Harmonic Mean	0.6869	0.6113	0.6565	0.4885
Avg Class Accuracy	0.7163	0.6737	0.6959	0.5983

From the statistics computed, our group decided that a **linear kernel** is best suited for the svm model.

5.2. Final Model

For the final model, after choosing a cost of 1 and a linear kernel, the accuracy that we got for an SVM model is 0.806.

	linear
Accuracy	0.8055
Precision	0.5712
Sensitivity	0.4961
False Negative Rate	0.5039
Harmonic Mean	0.6869
Avg Class Accuracy	0.7163

6. Model 3: K-Nearest Neighbours

6.1. Hyperparameter tuning

In K-Nearest Neighbours (KNN) algorithm, “k” refers to the number of nearest neighbours that are used to predict the label of a new data point. When a new data point is introduced to the KNN algorithm, it will identify the “k” nearest data points in the training set, based on some distance metric, and predicts the label of the new data point based on the labels of those “k” nearest neighbours.

If k is too small, the algorithm will be sensitive to noise and outliers, resulting in overfitting. If k is too large, the algorithm might lose local information and perform poorly on the training set.

To select a good k value, our group ran the knn function several times, with differing values of k.

Summary statistics:

	50	100	150	200
Accuracy	0.5712	0.5745	0.5540	0.5501
Precision	0.2905	0.2939	0.2826	0.2811
Sensitivity	0.6456	0.6535	0.6559	0.6589
False Negative Rate	0.3544	0.3465	0.3441	0.3411
Harmonic Mean	0.4323	0.4365	0.4232	0.4215
Avg Class Accuracy	0.5676	0.5710	0.5625	0.5616

6.2. Final Model

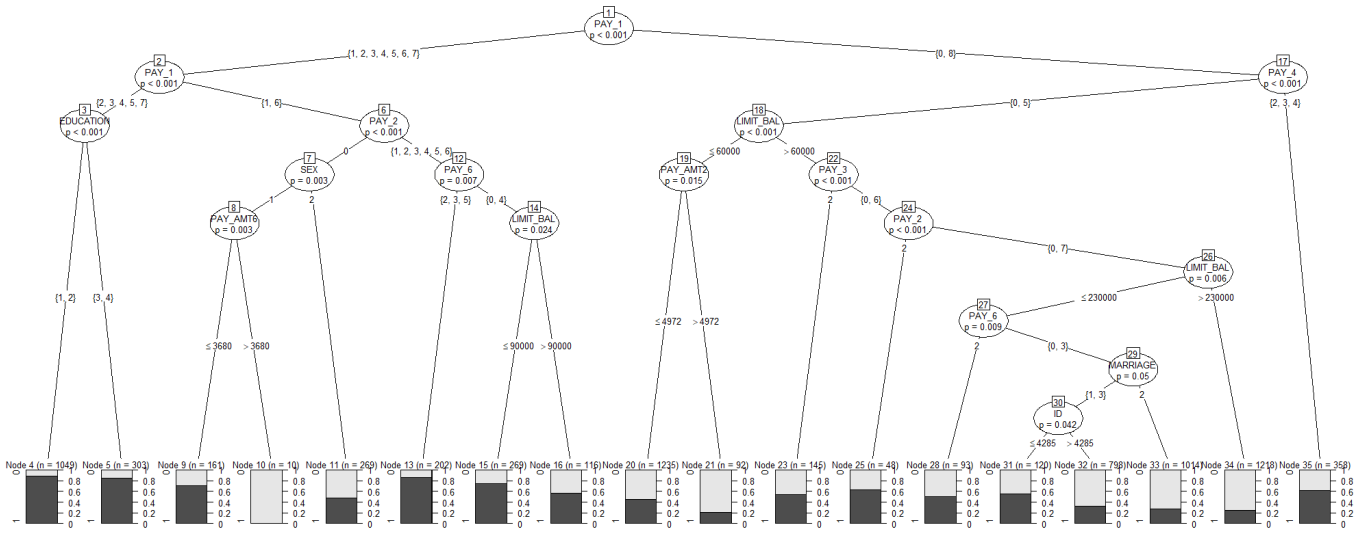
From the models run, our group decided that a k value of 100 would be optimal for the KNN algorithm. With this k value, the accuracy that our group got was 57.5%.

7. Model 4: Decision Tree

7.1. Final Model

	Train	Test
Accuracy	0.7020	0.7528
Precision	0.7615	0.4552
Sensitivity	0.5842	0.5772
False Negative Rate	0.4158	0.4228
Harmonic Mean	0.7101	0.5976
Avg Class Accuracy	0.7134	0.6623

We ran a decision tree with all the variables in the original dataset excluding BILL_AMT 1,2,4,5,6. It created the Decision tree below with 18 terminal nodes with different probabilities of default. The model does not have a very high accuracy but it has a low false negative rate, which we will be discussing more about in the final evaluation.



8. Model 5: Random Forests

8.1. Feature Selection

	Train	Test
Accuracy	0.9997333	0.7693
Precision	0.9997321	0.4842
Sensitivity	0.9997321	0.5970
False Negative Rate	0.0002679	0.4030
Harmonic Mean	0.9997333	0.6239
Avg Class Accuracy	0.9997333	0.6805

Firstly we applied a random forest for all the features of the dataset.

As seen from the table, the accuracy is 99.9% for the train dataset and 76.9% for the test dataset.

PAY_1	343.990914091169
ID	343.953708770924
BILL_AMT3	317.269188703323
PAY_AMT2	271.415173239328
PAY_AMT1	268.637830727583
AGE	267.969193038474
LIMIT_BAL	266.62883371638
PAY_AMT3	258.537817168736
PAY_AMT5	240.886938319173
PAY_AMT6	240.647391706845
PAY_AMT4	236.247371206832
PAY_2	138.75503203993
EDUCATION	101.596803128147
PAY_3	94.0254535605481
PAY_4	86.9215922914582
MARRIAGE	57.3493459729427
SEX	49.2580940757974

Next, we tried to find the top 7 features of the dataset to build a random forest and possibly help improve the accuracy of the model. We chose 7 features out of 19 because N / 3 ≈ 7, and it is recommended to have around N / 3 features selected for a regression so that there is enough strength to provide good test set accuracy.

Excluding the feature ID, from the above output, we can deduce that the top 7 features are PAY_1, BILL_AMT3, PAY_AMT1, PAY_AMT2, LIMIT_BAL, AGE and PAY_AMT3.

	Train	Test
Accuracy	0.98933	0.7388
Precision	0.99034	0.4346
Sensitivity	0.98821	0.5868
False Negative Rate	0.01179	0.4132
Harmonic Mean	0.98934	0.5794
Avg Class Accuracy	0.98934	0.6518

The accuracy we obtained for the test dataset is now 73.9% (3 s.f.). which is actually lower than if we included all 24 features. Hence choosing only the top 7 features did not help in improving the model's accuracy and we found that all 24 features should be included as they all influence the model substantially.

8.2. Hyperparameter tuning

We used 400, 500, 600, 700 and 800 decision trees as our parameters.

	400	500	600	700	800
Accuracy	0.7703	0.7687	0.7669	0.7709	0.7691
Precision	0.4860	0.4829	0.4799	0.4872	0.4838
Sensitivity	0.6030	0.5934	0.5958	0.6066	0.6012
False Negative Rate	0.3970	0.4066	0.4042	0.3934	0.3988
Harmonic Mean	0.6257	0.6226	0.6201	0.6270	0.6238
Avg Class Accuracy	0.6822	0.6794	0.6780	0.6833	0.6808

8.3. Final Model

Our final model uses all the variables without feature selection and we obtain the highest accuracy of 0.771(3 s.f.) at 700 decision trees.

9. Model 6: Neural Network

For neural networks we are required to convert all the data into numeric variables to use the package neuralnet. We decided to use all the variables to predict the default value. Our input layer is all the variables with X number and the output layer is the probability of default.

9.1. Hidden Layer Selection

The function for the neural network allows us to specify our hidden layers. A rule of thumb is that our hidden layer should be between the size of the input and output however since we have many inputs and outputs this was not an issue.

Another guideline is that the number of neurons should be ⅔ the size of the input layer + the output layer. Since the input layer which is the number of variables that was used to predict default is 19 and the output layer is 1. A good number of neurons with good time complexity is around 19 * ⅔ + 1 = 13. However due to the time complexity, we decided to run 2 hidden layers with 7 to 11 neurons in total.

9.2. Hyperparameter tuning

	3,4	3,5	3,6	4,3	4,4	4,5	5,5	5,3	5,6
Accuracy	0.7029	0.5713	0.24693	0.29707	0.5567	0.6811	0.7428	0.4780	0.7336
Precision	0.3139	0.2640	0.22618	0.23262	0.2679	0.3137	0.3221	0.2555	0.3238
Sensitivity	0.2853	0.5207	0.98799	0.94234	0.5754	0.3676	0.1435	0.7063	0.1838
False Negative Rate	0.7147	0.4793	0.01201	0.05766	0.4246	0.6324	0.8565	0.2937	0.8162
Harmonic Mean	0.4511	0.3983	0.36246	0.36734	0.4038	0.4523	0.4575	0.3909	0.4598
Avg Class Accuracy	0.5576	0.5374	0.56904	0.55273	0.5439	0.5620	0.5555	0.5434	0.5582

We decided to run the following combinations with 2 hidden layers to find the best neural network model.

(3,4), (3,5), (3,6), (4,3) , (4,4) , (4,5), (5,3), (5,5), (5,6)

Based on the hidden layers, the best accuracy is (5,5) with an accuracy of 0.743

	linear	radial	polynomial	sigmoid
Accuracy	0.8055	0.7599	0.7909	0.6548
Precision	0.5712	0.4687	0.5303	0.3420
Sensitivity	0.4961	0.6120	0.5093	0.6006
False Negative Rate	0.5039	0.3880	0.4907	0.3994
Harmonic Mean	0.6869	0.6113	0.6565	0.4885
Avg Class Accuracy	0.7163	0.6737	0.6959	0.5983

10. Model Evaluation and Conclusion

Best Models (all numbers are to 3 significant figures)

Model	**Logistic Regression**	**SVM**	**KNN**
Parameters	t = 0.5	Cost = 1, linear kernel	k value = 100
Accuracy	0.786	0.806	0.575
Precision	0.517	0.571	0.294
Sensitivity	0.572	0.496	0.654
False Negative Rate	0.428	0.504	0.347
Harmonic Mean	0.650	0.687	0.437
Average Class Accuracy	0.695	0.716	0.571
F1 Score	0.543	0.531	0.45

Model	**Decision Tree**	**Random Forests**	**Neural Network**
Parameters	N.A	nTrees = 700	Hidden Layer = (5,5)
Accuracy	0.753	0.771	0.743
Precision	0.455	0.487	0.322
Sensitivity	0.577	0.607	0.144
False Negative Rate	0.423	0.393	0.857
Harmonic Mean	0.598	0.627	0.458
Average Class Accuracy	0.662	0.683	0.556
F1 Score	0.509	0.540	0.199

Comparing the accuracy of all 6 models, it can be observed that the model with highest accuracy is the SVM model with an accuracy of 0.806.

Comparing the sensitivity values (True Positive Rate) of all 6 models, we can conclude that the model with the highest sensitivity is the KNN model at 0.648.

We compare the models using sensitivity values as higher sensitivity means the classifier model is more sensitive to detecting positive instances and hence higher sensitivity means the model is better at predicting actual positive values correctly. A substantial number of customers who default can cause an accumulation of debt for the bank. Hence, it is important for banks to accurately predict and thus identify most customers who will default.

We can also look at the F1 score especially since the dataset is unbalanced and has a large proportion of people not defaulting. The F1 score will take into account the false positives and false negatives and gives a better picture of the model especially for unbalanced dataset. The dataset with the best F1 score is the logistic regression model.

A good model should also flag out more customers that should be defaulting as it is better to prepare for those customers. Thus a smaller false negative rate is preferred so that fewer cases of defaulting are missed.

We believe that logistic regression is the best model because it overall has a good accuracy, high F1 score, high sensitivity and low false negative rates as compared to the other models.

11. References

default of credit card clients Data Set. (2016, January 26). UCI Machine Learning Repository. Retrieved April 14, 2023, from https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients