This report covers the data mining results of a historic loan application dataset provided by a private financial institution - Universal Plus. I assumed all records in the given dataset as personal loans, and Universal Plus approved every loan application.

Business Understanding

According to Universal Plus's data, 22.2% of customers would default, and 77.8% paid credit back. On average limit per customer, there is a 26% difference between customers who would default and customers who repaid credit, which means I can assume that Universal Plus assessed the customer's credit risk and set the credit limit based on credit risk. However, the average bill statement and average pay amount have no significant difference between the two groups, which means Universal Plus suffered huge losses from default customers. Therefore, the main business objectives of Universal Plus are 1) Minimise the likelihood of approving a loan to a customer who is likely to default, and 2) Maximise the likelihood of approving a loan to a customer who is likely to pay back.

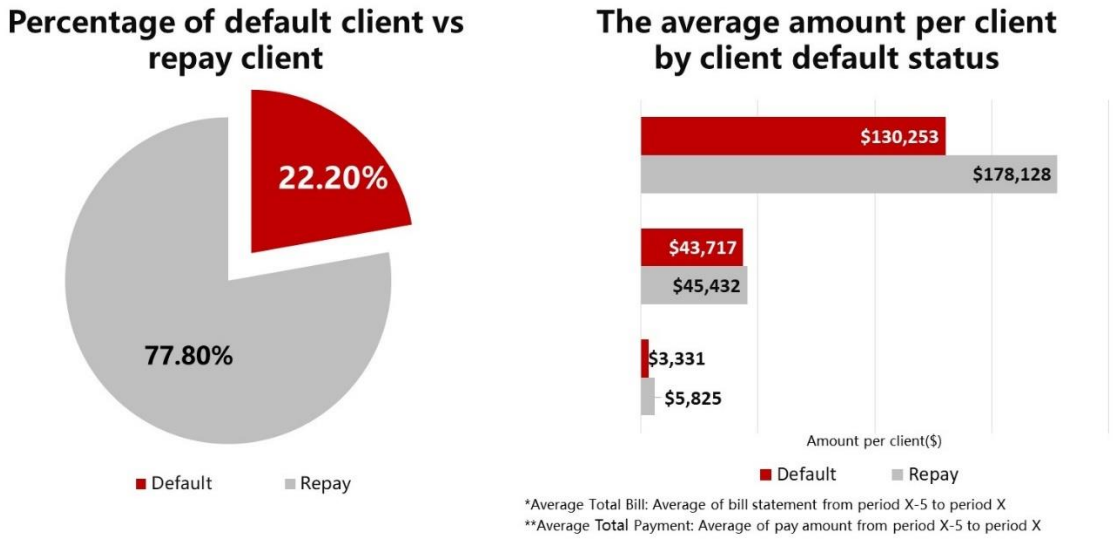

Figure 1As-Is Analysis

Data Understanding and Preparation

The given dataset has 31,375 instances of a historic loan application from period x-5 to period x with 39 variables, consisting of qualitative values such as "MARRIAGE" and quantitative values such as "BILL1". Firstly, I remove 1,339 duplicate observations and 210 NA observations, remaining 29,826 in the end. I also removed "ID", and "CM_HIST" (criminal history) variables since ID is an identifier that has nothing to do with predicting default, and CM_HIST(criminal history) values for every record are 0. "AGE_CTG" is directly derived from "AGE"; therefore, I

calculated information gain and removed "AGE_CTG", which has lower information gain than "AGE".

After data cleansing, I made a new variable assuming that customers' payment statuses would worsen over time if they were likely to default. In the data, "PY1", "PY2", "PY3", "PY4", "PY5", "PY6" represent the repayment status in period X to period X-5, respectively. These variables have values from -2 to 9 according to the period of repayment delay. The new variables are made by comparing the repayment status with the previous period, assigning 1 to the worsened case and 0 to the case that the repayment status remained the same or improved. For example, if "PY1" is 3 and "PY2" is 1, I assign 1 to "PY1D". After creating "PY1D" to "PY5D", I sum up 5 values and save them as a new variable named "sumPYD". If "sumPYD" is 2, it means a customer's repayment status worsened twice during the given period.

PY1	PY2	PY3	PY4	PY5	PY6	New Variable
3	1					PY1D = 1
	1	0				PY2D = 1
		0	0			PY3D = 0
			0	-1		PY4D = 1
				-1	-1	PY5D = 0
						SumPYD = 3

Figure 2 Example of new variables

Before modelling, I split train and test data with a ratio of 8:2. Since our target variable "CLASS" has a skewed proportion, I oversample training data. After oversampling, I have 23,861 observations for our training data which is still too big and time-consuming to train and test several models. 8,000 to 10,000 observations are usually considered big enough for a training model, and I make a subset of training data with 9,544 observations.

Modelling and Evaluation

After data partition, I built 5 machine learning models: Linear regression, support vector machine(SVM), decision tree, random forest, and gradient boosting machine(GBM). For the linear regression model, I predict that if the probability is equal or greater than 0.5, I assign it to 1(default). For the SVM model, I try to tune our model with cost parameters. I used two different libraries, which are tree and partykit, for the decision tree model to find the better model, and the better performance decision model is built from the tree library. I make a hyperparameter matrix for random forest model using mtry_val, nodesize_val, and sampsize_val. Then I find the best combination of hyperparameters for the minimum expected cost. To find the best model using GBM, I tuned three hyperparameters: n.trees(number of trees), interaction.depth, and cv.folds. I tried the " cv " and the " OOB "method to find the number of trees. With trial-and-error, I built models with five different machine learning algorithms.

To evaluate our models, I defined a performance indicator - expected cost. I adopted the most well-known cost matrix for credit risk prediction, the so-called German credit dataset, and it was published as part of the Statlog project.

From the confusion matrix in figure 3, A means that a bank approves a loan, and a customer would pay the credit back. B means a bank denies a loan, but a customer can pay the credit back. C means a bank approves a loan, but a customer cannot pay back. D means a bank denies a loan, and a customer would default. From this matrix, B makes the opportunity cost and C makes cost from default loan as follows: "If a good customer is rejected, the cost is an opportunity cost, the foregone profit of 1. If a bad customer is approved for a loan, the cost is the lost loan principal of 5"[Elkan C., 2001]. To summarise, I select the final model with minimum expected cost with more than 70% accuracy.

$$Expected\ Cost = Probability(B) * 100 + Probability(C) * 500$$

Confusion Matrix		Actual	
		Repay	Default
Predict	Repay (Approve loan)	A	C
	Default (Deny loan)	B	D

Cost Matrix		Actual	
		Repay	Default
Predict	Repay (Approve loan)		(C) 5
	Default (Deny loan)	(B) 1	

Figure 3 Confusion Matrix and Cost Matrix

The expected cost of linear regression, SVM, decision tree, random forest and GBM are 36.79, 39.17, 36.85, 36.37 and 34.45. To sum up, I chose GBM with the minimum expected cost as our final model.

	Accuracy	Precision	Recall	Expected Cost
Linear Regression	72.10%	41.19%	64.57%	36.79
SVM	72.46%	41.07%	59.82%	39.17
Decision Tree	73.30%	43.02%	62.25%	36.85
Random Forest	77.80%	49.36%	58.82%	36.37
GBM	75.12%	45.23%	65.41%	34.45

Figure 4 Performance result

The loss from a default loan is generally calculated by subtracting any earnings, including paid principal, interests, and collateral, from the total loan amount. However, I cannot find any information from the given dataset, and I assumed the sum of the limit ($849,117,680) from default customer as the cost from default loan for Universal Plus– which is the cost of C in the

confusion matrix. By applying the formula of the cost matrix (cost of B: C = 1:5), the cost of B equals one-fifth of the cost of C, which is $169,823,536. The expected cost without model is calculated by proportion customer went to default in total customer * cost of C ((451+853)/5965*$849,117,680 = $185,583,160), while with our model the expected cost could be successfully reduced to $58,504,208(1033/5965* $169,823,536 + 451/5965 * $849,117,680). To summarise, Universal Plus can save $127,078,952 with our model, which means Universal Plus can decrease costs by 68.5%.

Confusion Matrix		Actual	
		Repay	Default
Predict	Repay (Approve loan)	3625	451
	Default (Deny loan)	1033	853

Figure 5 GBM confusion matrix with 5,965 observations in test data

Deployment

I have a one-year deployment plan for this model. For the first three months, I plan to deploy our model in portions with 50%, 70%, and 90% of the whole clients. I will continuously monitor and tune our model with newly collected data with a monthly test. From April 2022, I will be in the full deployment stage. I plan to test our model quarterly from this stage, and if accuracy goes down to 70%, I will tune our model.

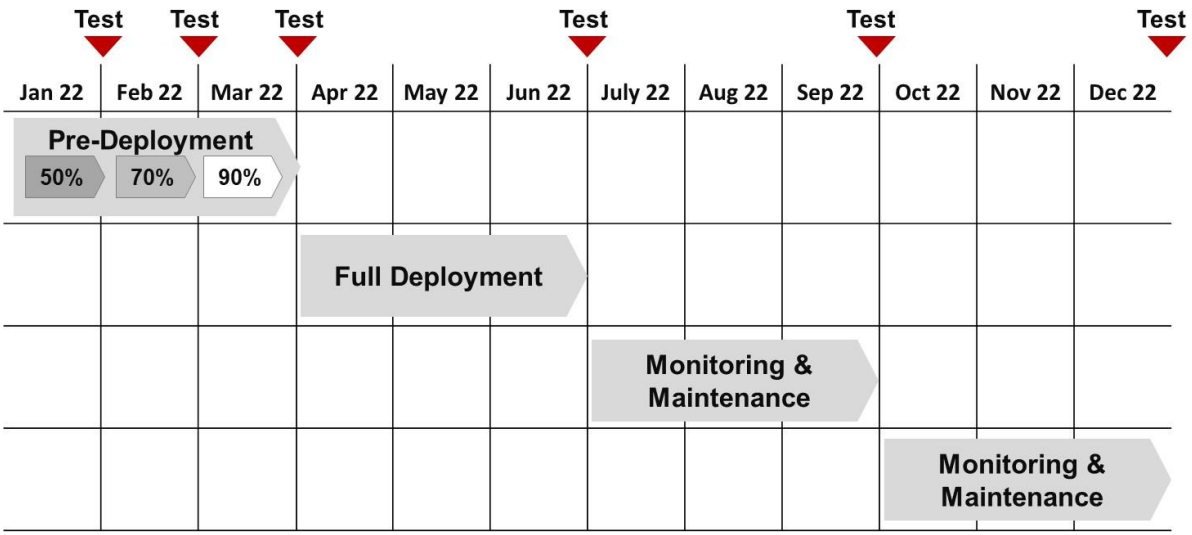

Figure 6 Deployment Plan