The Business Context

A major bank wants to better predict the likelihood of default for its customers, as well as identify the key drivers that determine this likelihood. They hope that this would inform the bank's decisions on who to give a credit to and what credit limit to provide, as well as also help the bank have a better understanding of their current and potential customers, which would inform their future strategy, including their planning of offering targeted credit products to their customers.

The Data

The bank collected data on 25 000 of their existing clients. Of those, 1 000 were randomly selected to participate in a pilot described below. Data about the remaining 24 000 is in the file "MMA867 A3 – credit data.xls". The dataset contains various information, including demographic factors, credit data, history of payment, and bill statements of credit card customers from April to September, as well as information on the outcome: did the customer default or not in October.

Pilot Project

Your department wants to pilot a new product, a short-term credit line with the limit of 25,000, and for the purposes of this assignment assume that the line is for 1 month at 2% per month. More so, assume that the client who was issued credit and repaid it will more likely use your bank for similar short-term financing needs in the future, which has an additional lifetime value (CLV) of 1,000. However, if the client will default, then you will be able to recover only 20,000 out of 25,000 credit granted.

The ultimate question: which of the 1 000 "new applicants" in the pilot should be issued credit?
In your analyses, please make the following simplifying assumptions:

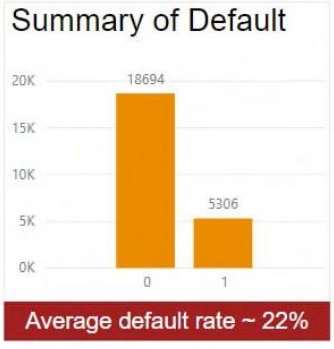
- Defaults on the previously issued credit is not your problem
- All the clients who will be offered the credit line will use it in full
- Your cost of capital = 0

In other words, for each client in the pilot, if the credit is issued and repaid, then the bank earns a profit of 25,000*2% + 1,000 = 1,500; if the credit is granted but the client defaults, then the bank loses 25,000 - 20,000 = 5,000? And if the credit is not issued, then the profit=loss=0.

1. Exploratory Data Analysis

1.1. Balance of Target Variable

We have analyzed the distribution of the target variable and find out that around 22% of the total customers have defaulted. The balance is not ideal for many of the classification algorithms however it's also not materially unbalanced. We have evaluated the benefit of using over sampling techniques (SMOTE in particular) and the resulting gain in AUC < 1%. The trade-off in complexity vs. model performance is too high to use SMOTE in this case.

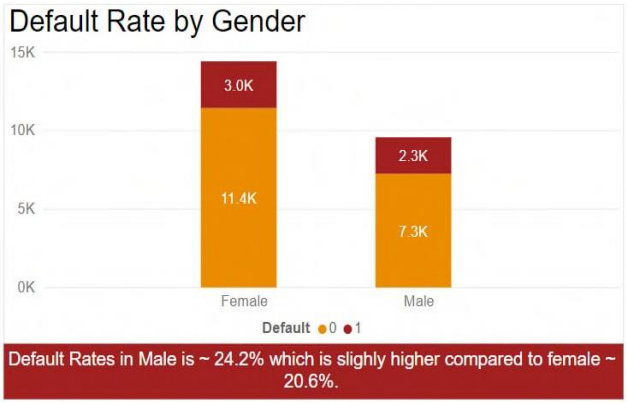


1.2. Missing Values

Dealing with missing values is the first step in almost any machine learning problem. Fortunately, there were not many missing values in this particular dataset. Only missing values as characterized by '0' were found in Marriage and Education. Since both of these are categorical variable, they were treated as it's own category and a surrogate feature was created.

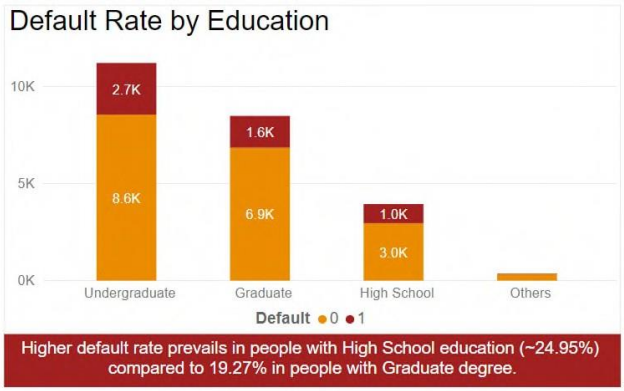
1.3. Default Rate by Gender

In order to create the right features, it is important we analyze the data in different dimensions. We began our analysis by splitting the data into genders. Based on our analysis we concluded that the data is almost evenly split between Male and Female with 40% of the data points pertaining to Male and 60% to Female. The probability of a default being Male is slightly higher at 24% compared to 21% in Female.



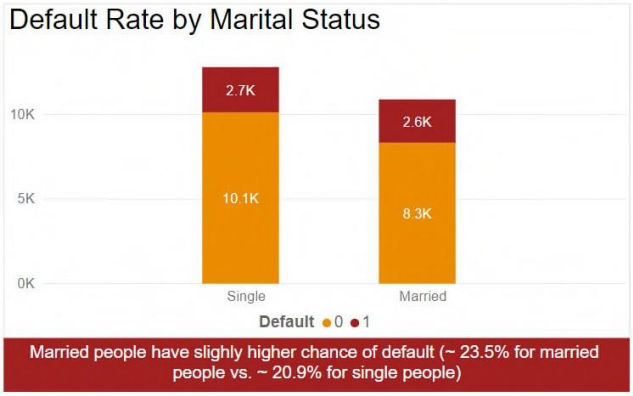
1.4. Default Rate by Education

The majority of the customers (slightly shy of 50%) are people with an under-graduate degree. Our analysis shows the highest default rate occurs in customers with High School education (~25%). Our hypothesis based on this analysis is that, the higher the education level, the lower the chance of default. The default rate in customers with graduate degree is only 19%.



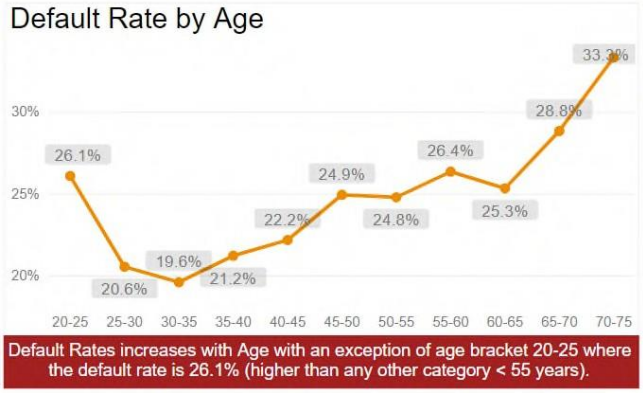
1.5. Default Rate by Marital Status

The dataset is evenly split between Married and Single customers (single customers are 53% of total). Our hypothesis based on the analysis shows that Married people have a slightly higher chance of default ~24% compared to 21% in customers who are Single. There are few data points for which marital status is missing, however they are not material in this particular case.



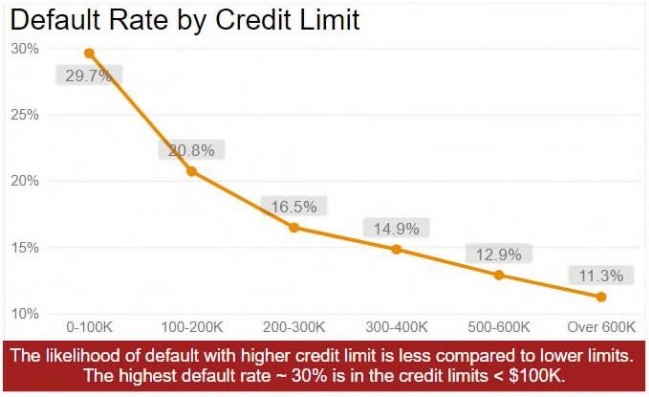
1.6. Default Rate by Age

In order to analyze default rate by Age, we have created equal size bin to divide the customers into age bucket. Our analysis shows the majority of the customers are less than 35 years of age. Our hypothesis based on this analysis shows that default rates increases with age with an exception of age bracket 20-25 where the default rate is 26.1% (higher than any other category < 55 years).



1.7. Default Rate by Limit Balance

In order to analyze the default rate by limit balance, we have created six bins to divide the customers into limit buckets. Our analysis shows that 62% of total customers have credit limit less than $200,000. Our hypothesis derived from the analysis shows that the likelihood of default with higher credit limit is less compared to lower limits. The highest default rate ~ 30% is in the credit limits < $100K.

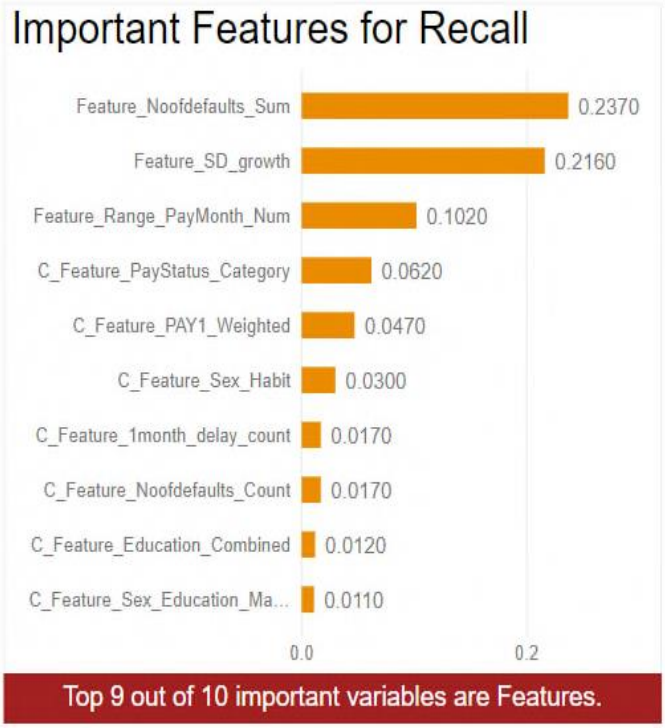


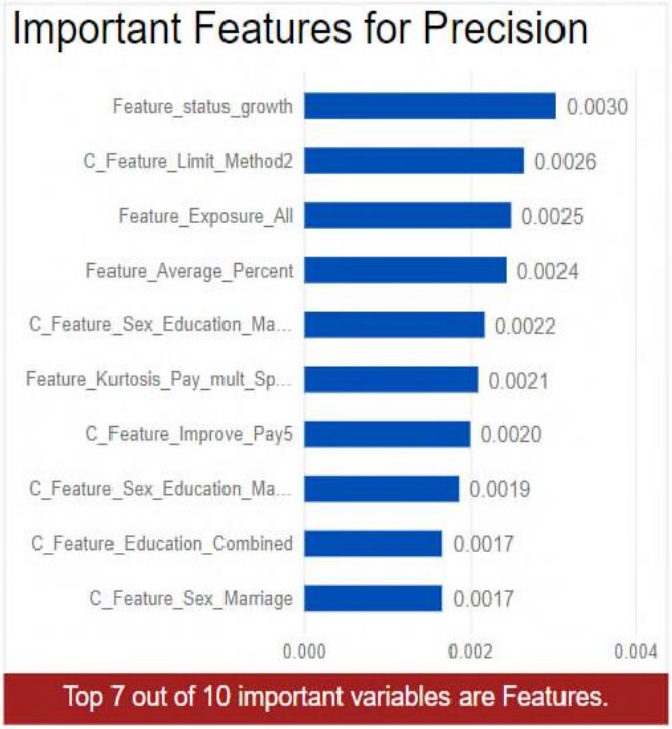
2. Feature Engineering based on EDA

After performing preliminary data analysis we have built our base model without any feature engineering. However, for the purpose of this report, this section is shown before describing any models. Continuing on our EDA process, we have derived some meaningful insight from the data and based on our hypothesis, we have started creating features. We ended up creating over 160 features and then tuned the model parameters to avoid over fitting. Our best model (Two Class Boosted Decision Tree) is using ***total of 86 variable of which 68 (~80%) are the features we have created***. ***In fact top 9 out of 10 important variables in our model are features***. The complete list of the features is given in annexure. Top 10 features are described below with their contribution in Recall metric.

Table 1 – Top 10 Features in our best Model

Feature	Score	Example	Definition
Feature_Noofdefaults_Sum	0.2448	4	Sum of all pay status where pay status > 0
Feature_sumofall_paystatus	0.0305	-2	Sum of all pay status
C_Feature_2month_delay_count	0.0132	2	Count of times payments were delayed by 2 months
C_Feature_Noofdefaults_Count	0.0115	2	Count of all pay status where pay status > 0
C_Feature_PayStatus_PayScore_Month_1	0.0106	Under AND 2	Interaction of Pay Status in M1 and Payment in M1 / Bill in M2
Feature_SD_bill	0.0085	1608	Standard Deviation of bill amount over last 6 Months
Feature_Payment_Expense	0.0081	0.1248	Average of monthly payment divided by monthly expense (feature)
Feature_TotalPayment_6Months	0.0079	689	Total Payment in last 6 months
C_Feature_Pay_last2_status	0.0064	2 AND 2	Pay Status of M1 and M2
Feature_Expense_1	0.0064	1500	Assuming bill amount is c/f balance for all the spending minus all the payments. This feature intends to calculate incremental expenditure for the month of bill (Mx)





3. Model Building

R and Azure Machine Learning Lab were used for building models. The 5 initial base models were built in R and used the original 25 feature dataset. Various feature engineering techniques were then used to expand the dataset to 56 variables and the models were re-run in R which generated slightly improved results. A second round of feature engineering led to 135 new variables which brought the dataset total to 160. With this amount of data the models had difficulty running in R due to processing power limitations so Azure was used instead.

Some of our models in Azure Machine Learning Lab have required well over 60 hours to process. In total we have performed over 100 hours of experiment in Azure.

3.1. Choice of Evaluation Metrics

Even before building the models, choosing the correct evaluation metrics was an important decision. We know the data is unbalanced (~78% people do not default) which means even if we do not build a model and classify everybody as 'no default' we could achieve 78% accuracy. However, this kind of model is obviously not good as we have other considerations such as type 1 and type 2 error cost and the right balance between recall and precision (F1 score could be a good indication as well). Hence we have chosen to optimize all our models for AUC and compared them based on AUC and Recall.

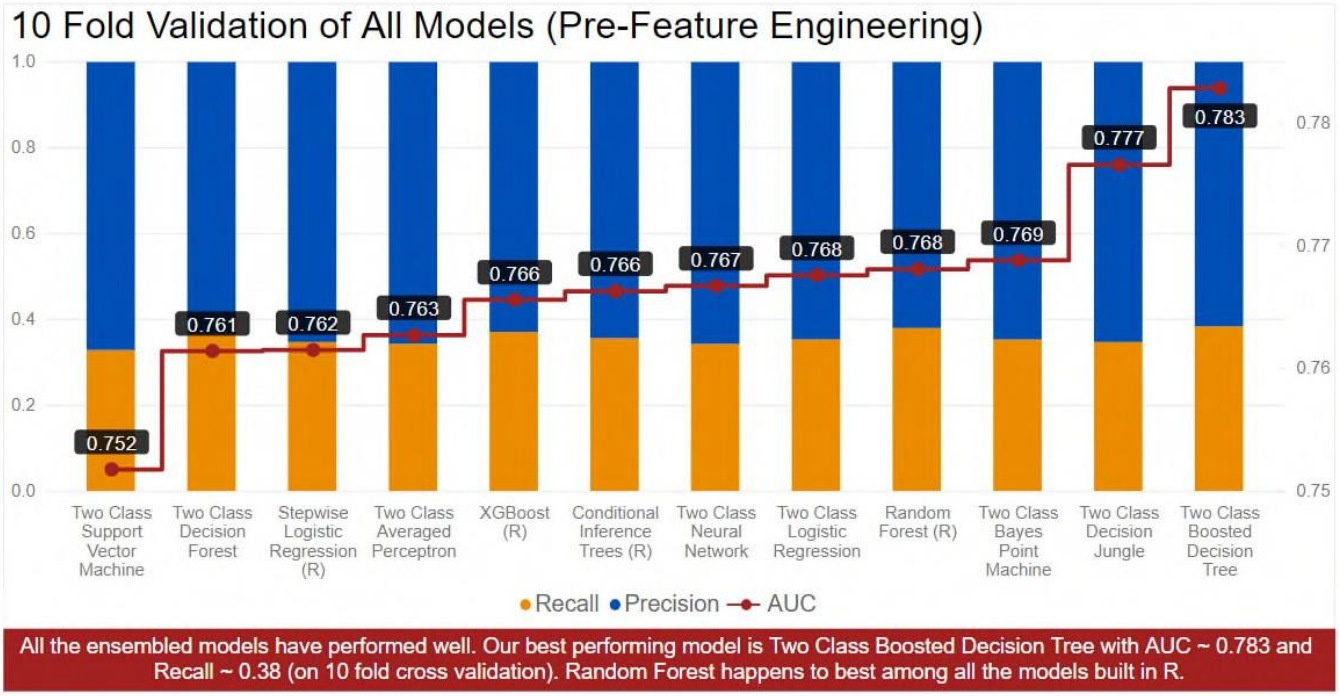
3.2. Cross Validation Technique

Instead of using a random split, we have used 10 fold validation technique in building all our models which means the dataset is divided in 10 equal folds (2,4000 / 10 = 2,400) and model is trained on

9 folds and prediction is made on 10th fold. Validation fold keeps rotating until all the folds have been predicted. Values for AUC and Recall shown below is the mean result from all 10 folds.

3.3. Base Models

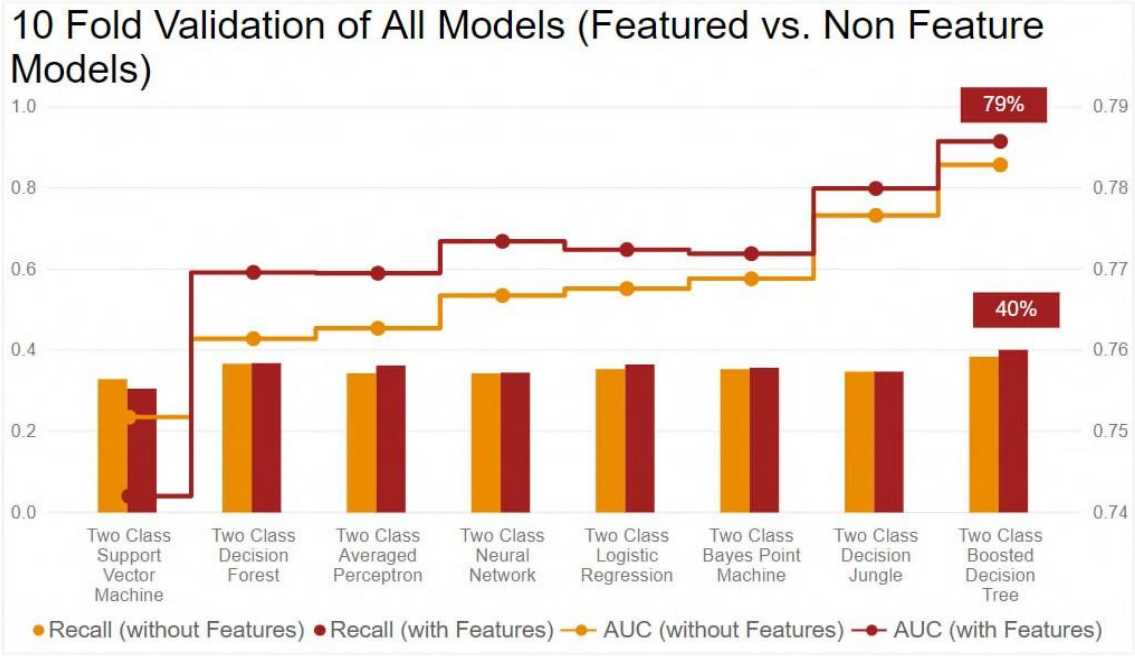
We have built over 12 different models (4 in R and 8 in Azure). The results from all the base models (excluding feature engineering) shows that our best performing model is Two Class Boosted Decision Tree (AUC ~ 0.78 and Recall ~ 0.38). Our worst performing base model is Two Class SVM (AUC ~ .75 and Recall ~ 0.33).



Threshold for Recall and Precision = 0.5

3.4. Models with Feature Engineering

After building the minimal viable product and observing the key evaluation metrics in base models, we have started including features in the model. Several attempts and combination of feature groups have been included and excluded to observe the behavior of models. After several attempts, we concluded that it is best to use tuning algorithms using permutation feature importance function to evaluate the combination of features. Due to size of our data model after including all the features, performance and hyper tuning the parameters of our ensemble models became the challenge. The results below shows the improvement in the model after including features. Almost all the models have shown improvement in the performance except for Two Class SVM. Two Class Boosted Decision Tree is still the base model and we were able to improve AUC by ~0.7 and recall by 2% after including features.



Two Class Boosted Decision Tree is an ensemble learning method in which the second tree corrects for the errors of the first tree, the third tree corrects for the errors of the first and second trees, and so forth. Predictions are based on the entire ensemble of trees together that makes the prediction[1]

Once this fact has been established that two class boosted decision tree is our best model, we have then deployed the model to answer all the questions and analysis required in Q1 and Q3 of the assignment.

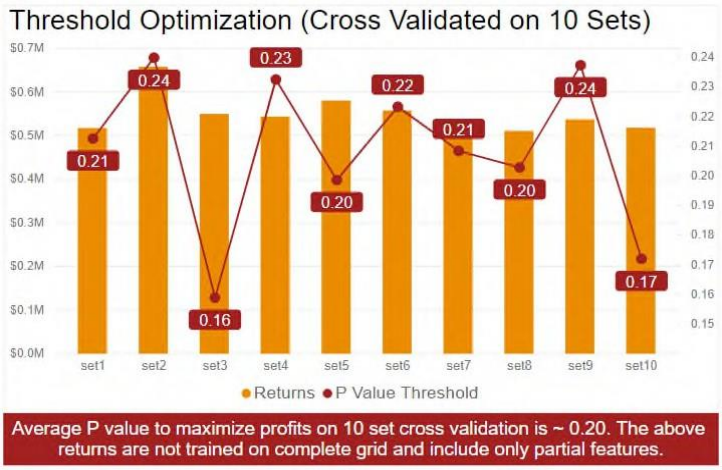
4. Choosing the Right Threshold for Classifying Default

After building multiple models and choosing the best performing model (Two Class Boosted Decision Tree), it was time to come up with something important i.e. right threshold percentage. Since we already know the financial impact for type 1 and type 2 errors, it is now a matter of building an optimization model to choose the threshold that will result in the highest profit, but how? Every cross validation will result in a different threshold percentage that will maximize the profit for that particular case. The most important question here is how the result from one validation set can be generalized for the prediction of unknown 1000 customers. After all, we don't need a threshold percentage that performs well on one particular validation set.

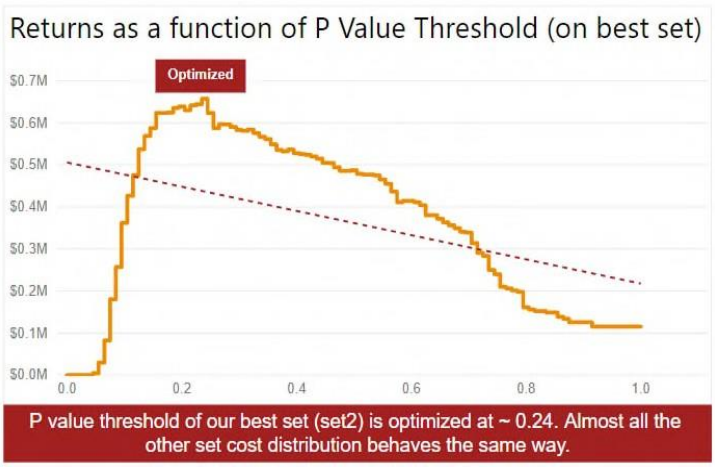
In order to come up with a right threshold criteria we have decided to split the data over 20 times using different seed each time. Each time our train set will contain 95% of the data i.e. 23,000 records and validation set would contain 1,000 records. We have then used Evolutionary Algorithm[2] in Solver to come up with the optimum threshold for each validation set which was then averaged for our final prediction. For the purpose of clarity only ten results are shown below.

[1] https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/two-class-boosted-decision-tree#bkmk_research

[2] https://www.solver.com/excel-solver-change-options-evolutionary-solving-method



Average P value to maximize profits on 10 set cross validation is ~ 0.20. The above returns are not trained on complete grid and include only partial features.



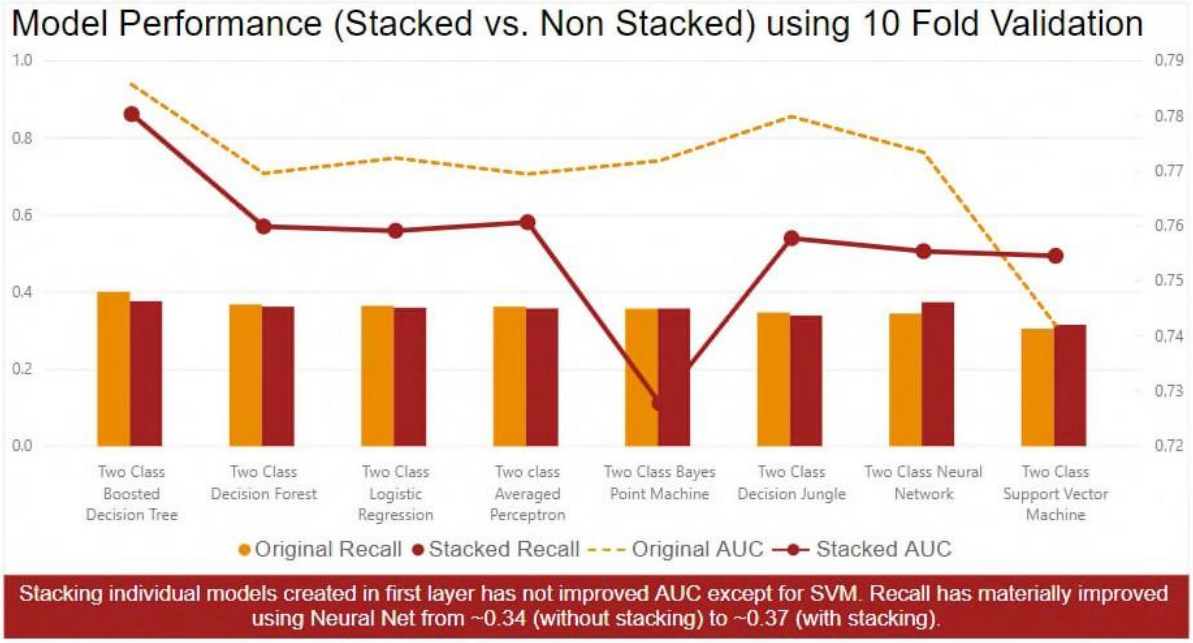
P value threshold of our best set (set2) is optimized at ~ 0.24. Almost all the other set cost distribution behaves the same way.

5. An Attempt at Stacking Models

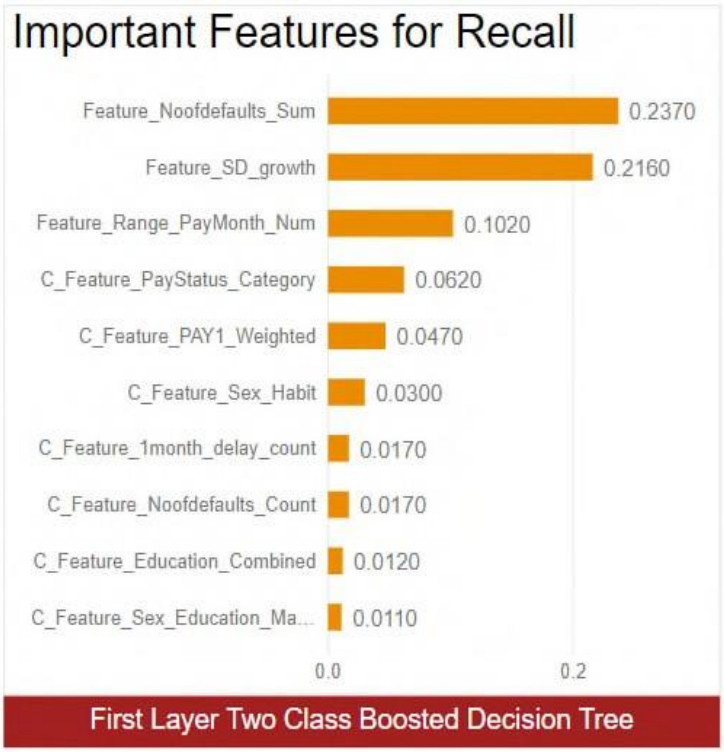
We have also explored an approach of stacking all the models we have built to further improve the performance of our models. In this approach we have used 17 different models with each of them optimizing hyper parameters. Our objective was to build diverse enough base of models. We call it our model library. First, we have split the original dataset into 60% and 40% and have used the first set to train and predict P values on the second set which we call our 'Scored Dataset'. This is the first layer of our model. Scored dataset from this layer became our train dataset for second layer where we have used different models to see how they perform while stacking. Example of differences is given below:

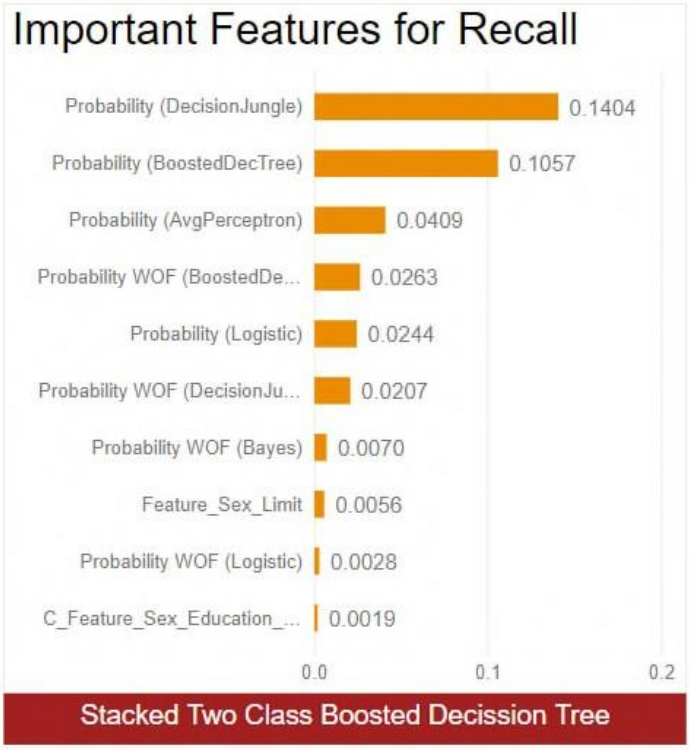
ID	Logistic	Decision Jungle	Average Percept.	Boosted Tree	Boosted Tree WOF	Average Percept. WOF	Decision Jungle WOF	Bayes WOF	Logistic WOF
1	0.0880	0.1258	0.1076	0.1293	0.1455	0.1311	0.1696	0.1393	0.1348
2	0.0744	0.1050	0.0939	0.0837	0.0770	0.1278	0.0818	0.1084	0.1073
3	0.0814	0.0636	0.0756	0.0745	0.0846	0.1022	0.0591	0.0772	0.0792
4	0.4901	0.6525	0.5386	0.6092	0.4286	0.5948	0.7070	0.4987	0.5157

All the P values obtained by predicting our score set in first layer were then used to train our model in second layer in an attempt to perform final prediction on 1,000 customers. However, we have done cross validation in second layer to see how models are performing and have then compared it with individual models built in first layer. The process itself was too complicated and resource intensive and may not be practically deployable in real life scenario. ***However, we only did this for learning purpose.***



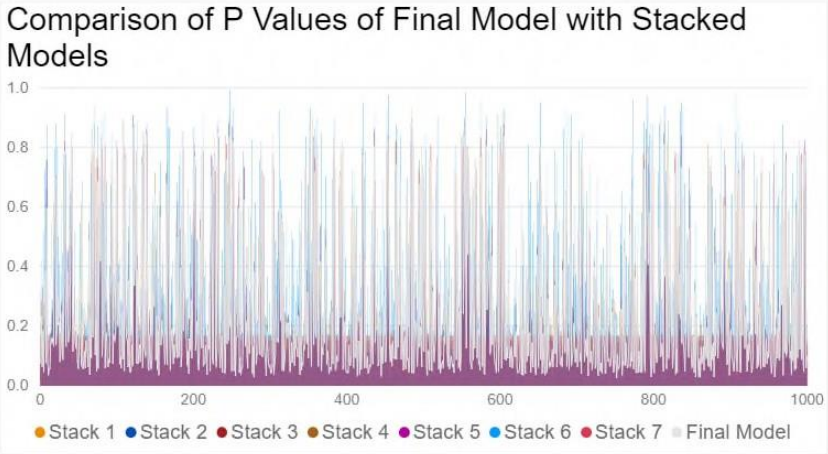
We have tried stacking our models using 8 different algorithms and none of them have seem to improve our AUC on cross validation set. We have then compared the AUC in second layer with the AUC obtained in first layer to see how stacking has affected the performance of our models. The model performance has not improved in any algorithm except for SVM. Our guess for declining performance is because the model in second layer is not getting enough data to train. At this point, we were curious to know how the importance of variables have changed after stacking so we decided to compare feature importance after stacking with the first layer best model.



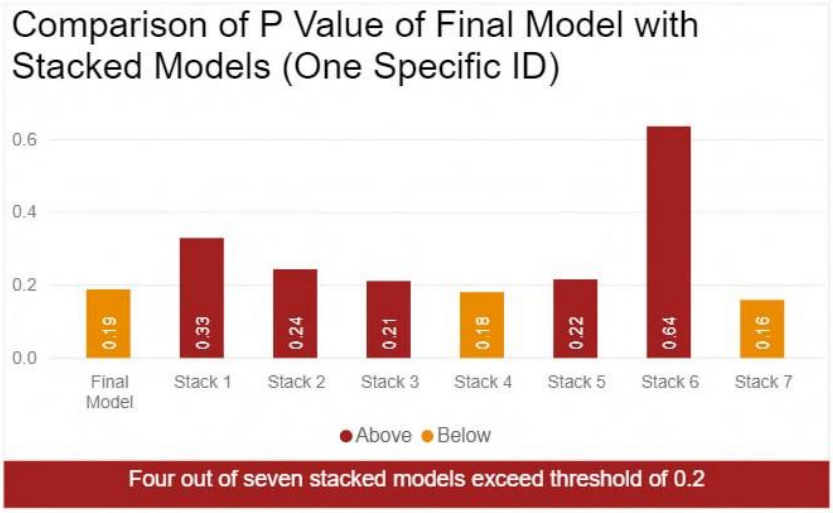


6. Adjustment of Border Line Threshold to control Type 1 Errors

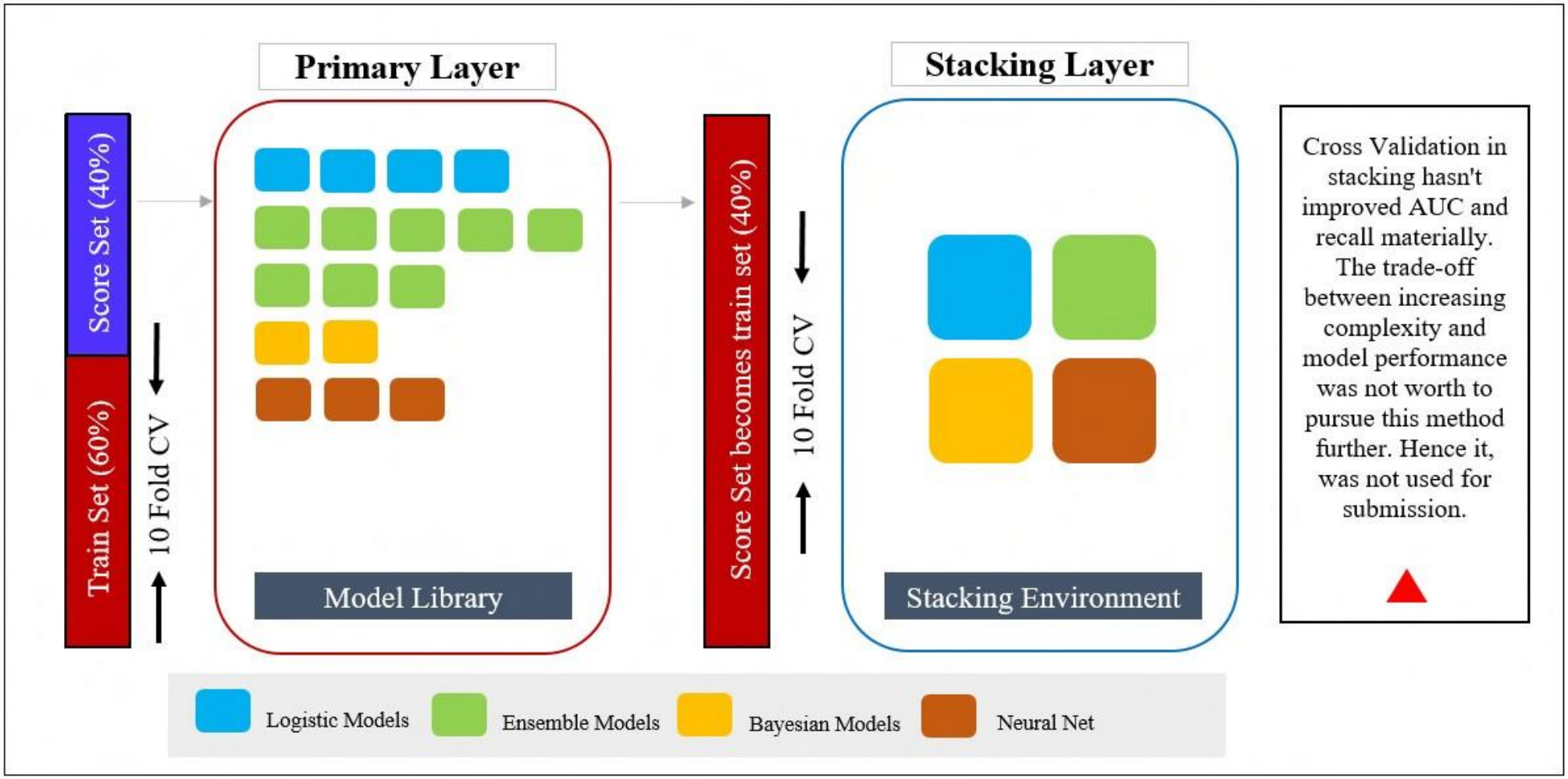
After stacking the results from stage 1 using different algorithms, we have used this stacked model with different algorithms to predict the P values for final score set (1,000 customers) and compared it with our final model from Stage 1 (Two Class Boosted Decision Tree). We have observed some significant differences in P values resulting from different stack vs. non stack models.



During cross validation we have observed some sharp deviations in customers with P value > 0.15 based on our final model from first layer in comparison with stacked models. It was not possible to do the adjustments for the entire set as it is a manual process so we have chosen to investigate all the P values 1 standard deviation (~0.025) away from average threshold of 0.2. We have then analyzed the results of those records from our best model as well as stacked model. If the number of models predicting P value higher than threshold exceed 50% of total models we have added a constant value in the predicted P value result from first layer to push the specific customer on the other side of fence i.e. predicting as 'default' even though our best model probability <0.2 but within 1 standard deviation. In this way, we didn't had to reduce our threshold in general but we were able to target just type 1 risks. One example of such case is given below.



VISUAL REPRESENTATION OF STACKING PROCESS



7. Notion of "Responsible AI"

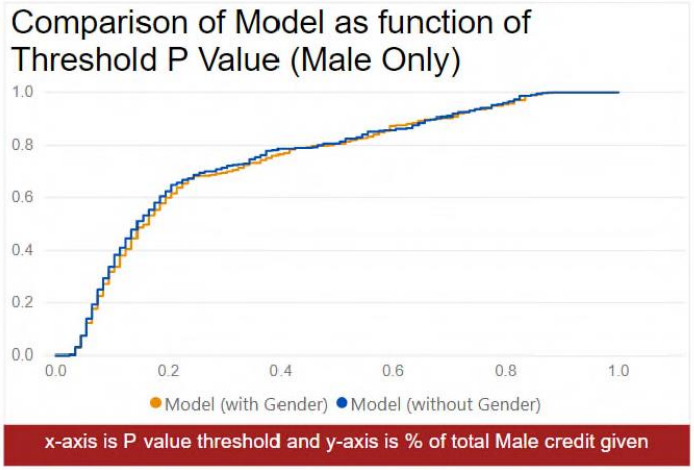
This section of the report answers question 3 from the assignment. First two parts of the question are answered together followed by our debate on equality and ethics in algorithmic decision making. Going back to the discussion in section 1 (EDA), our hypothesis shows that generally the probability of default being Male is slightly higher ~ 24% compared to Female where probability is ~ 21%. Based on this, even before removing gender and all associated features from the model, we knew that removing gender and associated variables will improve the situation for Male customers in our scoring dataset (1,000 customers). In order to quantify the impact we have excluded Gender variable and all the associated features and the comparisons are shown below.

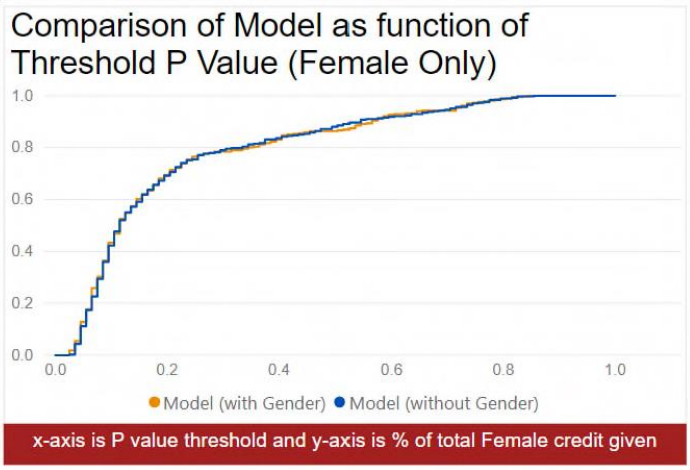
Table 2 – Comparison of Model with Gender vs. without Gender

Prediction	Model with Gender			Model without Gender		
	Male	**Female**	**Total**	**Male**	**Female**	**Total**
Default	147	190	337	138	200	338
No Default	220	443	663	229	433	662
Total	367	633	1,000	367	633	1,000
% of Default	**40%**	**30%**	**34%**	**38%**	**32%**	**34%**

Threshold = 0.2

Upon comparing these two models we have now quantified the impact of gender and associated variable in our base model. Even though based on our feature selection, gender itself is not an important variable, however there are multiple features interacting with gender causing biased against Male. There is a slight improvement of default % in Male from 40% (based on model with Gender) to 38% in model without considering gender. In order to answer second part of the question we have analyzed the change % as a function of probability threshold for Male and Female separately.





Debate on Responsible AI

The discussion about responsible AI and anti-discrimination legislation is one that upon initial analysis seems simple to fix. For example the EU legislation in the Treaty of the Functioning of European Union (TFEU) and the "Gender Directive" mentioned in our case are a clear attempt by the EU to eliminate the use of gender in predictive model building. This method used by the EU of eliminating the use of gender in models is a popular method in many academic resources. One such resource is the book titled "Techniques for Discrimination-Free Predictive Models" by Faisal Kamiran et.al, depicts clearly 3 methods for producing a discrimination-free predictive model. The first method mentioned was precisely as we have done in our analysis, eliminate the gender and associated features directly from the dataset upfront so that the model doesn't learn from the gender data. Legislation in the US is very different from that in Europe. The major difference is that the collection of gender data in first place is prohibited. While in Europe you can collect gender data but not use it in model building, the US law prohibits even the collection. This causes a very interesting situation for model builders. In Europe for example you have access to gender data so you can quantify the outcome from using and not using gender as a variable. In the US you wouldn't have the gender data to make that inference.

Book titled "Techniques for Discrimination-Free Predictive Models" by Faisal Kamiran et.al, depicts clearly 3 methods for producing a discrimination-free predictive model.

Not having all the information about your dataset, in this case missing the gender information, presents a significant risk to your predictive model's accuracy. For example, let's assume we were in the US and our test dataset was all male, but our prediction was to cover potential applicants of all genders, then our predictive model would be strongly biased to predict only how male applicants would perform. This inherent bias would be undetectable without the gender data and would ultimately produce a less accurate predictive model.

It is important to remember that in our analysis the "Gender" variable alone did not form part of our optimal model but features interacting with genders were included. In our analysis we have proven that by eliminating the gender features a different number of approved loans was obtained. As you can see from the two comparison charts above, in both cases male and female, there are deviations in the models as depicted by the lines not being exactly on top of one another. What this shows is that for given thresholds, there are differences to the number of approved loans for both males and females. Our findings were that the model without gender showed a slight increase to the male share of approved loans and a slight decrease to the female share. Specifically of the 1000 applicants, 9 more males would receive loans and 10 fewer females would be accepted if the gender variables were to be eliminated. When considering these numbers as a share of each gender, the male share of accepted loans increased 2.5% when gender variables were eliminated and the female share decreased 1.6%. So depending on the number of applications, these deviations could potentially have a great effect on the profitability of the lending program.

When considering the two approaches by the EU and US, our analysis shows that there could be a difference to the profitability of a lending program depending on if gender variables are included in a model. Also there could be biases in the data that are explained by the gender data, which can be corrected for if the gender data is at least known.

Annexures

Annexure I – Comparison of Base Model with Final Model

Annexure II – R scripts for Base Model

Annexure III – List of Features