# **INTRODUCTION**

This report investigates the influencing factors of the likelihood of a borrower not paying debts and build a default model to predict whether an entity which have a loan would default in the end. Before a financial institution, such as a bank, approve a loan or credit application, it is pretty crucial to evaluate risk of default related to this client. In the financial market, there are some official credit agencies such as Moody’s, Standard & Poor and Fitch engaging on developing default modal or credit rating model for bonds, companies or even countries. Their responsibilities are providing guidance on how safe the bond/company is. In other words, how likely is the bond issuer pay back investor’s money. Specifically, the following table describes the basic level of credit rating system.

Table 1: Crediting rating systems of credit agencies

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **S&P's** | **Moody's** | **Fitch** |
| Highest quality | Investment grade | AAA | Aaa | AAA |
| High quality | AA+ | Aa1 | AA+ |
| AA | Aa2 | AA |
| AA- | Aa3 | AA- |
| Strong payment capacity | A+ | A1 | A+ |
| A | A2 | A |
| A- | A3 | A- |
| Adequate payment capacity | BBB+ | Baa1 | BBB+ |
| BBB | Baa2 | BBB |
| BBB- | Baa3 | BBB- |
| Likeyly to fulfil obligations, ongoing uncertainty | Speculative grade | BB+ | Ba1 | BB+ |
| BB | Ba2 | BB |
| BB- | Ba3 | BB- |
| High credit risk | B+ | B1 | B+ |
| B | B2 | B |
| B- | B3 | B- |
| Very high credit risk | CCC+ | Caa1 | CCC+ |
| CCC | Caa2 | CCC |
| CCC- | Caa3 | CCC- |
| Near default with possibility of recovery | CC | Ca | CC |
|  |  | C |
| Default | SD | C | DDD |
| D |  | DD |
|  |  | D |

Therefore, we can use this information to evaluate the risk of an investment and build a portfolio with durable risk.

However, these credit agencies can only evaluate bonds which are public traded or large business. For small size and medium size companies, banks have a hard time finding public information evaluating their individual default risks. Each bank has to do the background investigation to create their own credit rating system for their clients. Banks can obtain a lot of useful information from this system, but the problem is that the system can only reflect the clients’ history instead of future behavior. Banks still cannot predict whether a particular client would pay back the loan or default. To solve this problem, this report explores several models to predict the result of loans based on some features of a client such as industry, year\_of\_operation, principal etc. Then, we evaluate the performance of each model using predication accuracy and confusion matrix and provide a model with the highest accuracy, sensitivity and AUC score.

# **EXPLORATION**

We collect the data from a private database of a commercial bank in the United States. This data set has 14 variables describing detailed information about each loan record. Specifically,

* Credit rating states the likelihood of making payments on time, similar to the simple version of credit rating system used by credit agencies mentioned above. It has 5 levels, including A+, A, B, C and D.
* Industry describes the industry that the borrower is in, such as Educational Services, Manufacturing and Retail Trade.
* State shows the geographic information of the borrower.
* Year of operation shows the age of each business who borrow the money.
* Fico score is a type of credit scores calculated based on information in credit report provided by the Experian, Equifax and TransUnion.
* Status describes the status of each loan. Current means the loan is still going on. Defaulted means the borrower failed to pay back the money. Paid in full means the borrower paid back the debt on time.
* Other variables include principal, interest rate, term, number of guarantors, business employee count, high risk industry, broker, homeowner.

## Data cleaning

To provide more convincing result, this report cleaned the data. For example, in terms of variable industry, this report replaces the category “Other Services (except Public Administration)” into “Other Services” to combine two other terms. Then, we correct some typo, such as “accomodation” to “accommodation”. Besides, for the missing value in the variable industry, we classify them into “Other Services”. Then we drop the record with a fico score of 0 because minimum value of FICO score is 300. Besides, there are dollar sign $ and comma in the value of principal, so we remove these non-digital characters and convert principal into integer type. The status is the dependent variable in our model. In this report, we only analyze sample with an ultimate result, whether the borrower paid back the loan or defaulted, so we drop all record with the status of current. Finally, we drop all records with any missing value on any variables.

This report transformation all categorical variables into dummy variables. For example, we created five dummy variables for variable of credit rating, including credit\_rating\_A+, credit\_rating\_A, credit\_rating\_B, credit\_rating\_C and credit\_rating\_D. Similarly, we did the same conversion for industry. For dependent variable status, we converted category default into 1 and other categories into 0.

## Descriptive Statistics

Table 2: descriptive statistics for all variables

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **count** | **mean** | **std** | **min** | **25%** | **50%** | **75%** | **max** |
| **years\_of\_operation** | 4709 | 10.64953 | 7.991417 | 0 | 5 | 8.6 | 13.8 | 67 |
| **principal** | 4709 | 138240.4 | 104804 | 25000 | 60000 | 100000 | 200000 | 500000 |
| **interest\_rate** | 4709 | 0.141907 | 0.04472 | 0.0499 | 0.1119 | 0.1399 | 0.1629 | 0.2879 |
| **term** | 4709 | 40.88511 | 14.90832 | 6 | 36 | 36 | 60 | 60 |
| **status** | 4709 | 0.086855 | 0.281652 | 0 | 0 | 0 | 0 | 1 |
| **fico\_score** | 4709 | 706.1172 | 44.57986 | 604 | 672 | 703 | 738 | 843 |
| **number\_of\_guarantors** | 4709 | 1.323636 | 0.536434 | 1 | 1 | 1 | 2 | 6 |
| **business\_employee\_count** | 4709 | 14.52219 | 53.18933 | 0 | 4 | 7 | 14 | 2000 |
| **high\_risk\_industry** | 4709 | 0.137184 | 0.344078 | 0 | 0 | 0 | 0 | 1 |
| **Broker** | 4709 | 0.305585 | 0.460704 | 0 | 0 | 0 | 1 | 1 |
| **Homeowner** | 4709 | 0.816946 | 0.386752 | 0 | 1 | 1 | 1 | 1 |

The table shows the overview of our detailed data. The minimum value of years of operation is 0, which means within one year. The maximum of years of operation is 67. The mean of principal and interest rate are 138240.4 and 0.1419 respectively. The mean value of status is 0.0869, which means that 8.69% of firms in our sample defaulted. The mean of high risk industry, broker and homeowner are 0.1372, 0.3056 and 0.8169 respectively meaning there are 13.72% of firms are in high risk industry, 30.56% are brokers and 81.69% are homeowner.

# **METHODOLOGY**

The final sample contains 4709 observations. This report splitted the full sample into training dataset (70%) and testing dataset (30%). We used training dataset to build the model to predict and used the testing dataset to evaluate the performance of different models. This report applied several algorithms to predict whether a borrower would default, such as logistic regression, random forest, boosting and support vector machine (SVM).

**Logistic Regression**

Given the research variable that predict is “status” which is a binary variable, this report applied logistic regression. The model is shown below.



**Code**

**from** sklearn**.**linear\_model **import** LogisticRegression

lm **=** LogisticRegression**()**

lm**.**fit**(**train\_x**,**train\_y**)**

predict\_lm **=** lm**.**predict**(**test\_x**)**

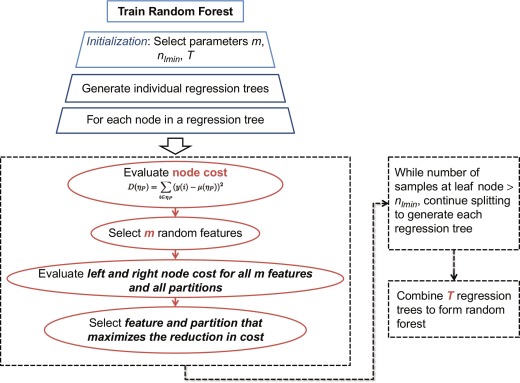
accuracy\_lm **=** accuracy\_score**(**test\_y**,** predict\_lm**)**

confusion\_matrix**(**test\_y**,** predict\_lm**)**

**Random Forest**

Random forest is similar to boostrapping algorithm with decision tree model. This algorithm evolves from decision trees and conducts a bunch of decision trees to get the most voted prediction as the result. Basically, random forest makes a vast number of classifiers form a strong classifier.

Figure 1: Random forest algorithm explanation



**Code**

rf **=** RandomForestClassifier**(**n\_estimators**=**1000**)**

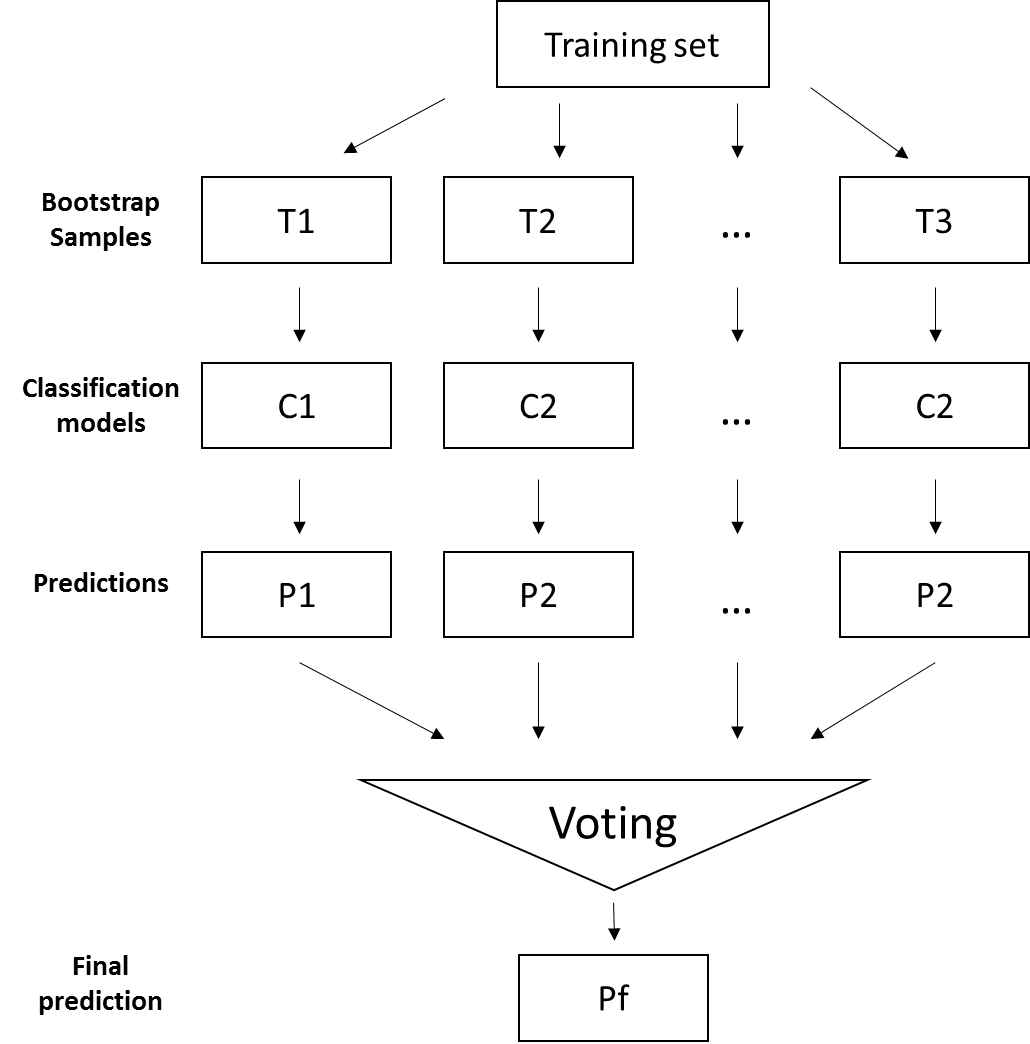
rf**.**fit**(**train\_x**,**train\_y**)**

predict\_rf **=** rf**.**predict**(**test\_x**)**

**Bagging**

Bagging, also known as Bootstrap aggregating, it is a type of ensemble machine learning algorithm. It aggregates a lot of weak models to make the final prediction.

Figure 2: Bagging algorithm explanation



**Code**

tree **=** DecisionTreeClassifier**(**criterion**=**'entropy'**,** max\_depth**=None)**

bag **=** BaggingClassifier**(**base\_estimator**=**tree**,** n\_estimators**=**1000**,** max\_samples**=**1.0**,** max\_features**=**1.0**,** bootstrap**=True,** bootstrap\_features**=False,** n\_jobs**=**1**,** random\_state**=**1**)**

bag**.**fit**(**train\_x**,**train\_y**)**

predict\_bag **=** bag**.**predict**(**test\_x**)**

**Support Vector Machine (SVM)**

Support vector machine is a supervised learning algorithm that classify data into one of two categories. It finds the hyperplane between to categories and maximized the margin of distance.

**Code**

sv **=** svm**.**SVC**()**

sv**.**fit**(**train\_x**,**train\_y**)**

predict\_svm **=** sv**.**predict**(**test\_x**)**

# EVALUATION

This report applied three metrics to evaluate the performance of the three models above, including accuracy, sensitivity, confusion matrix and ROC. Notably, the reason we pay attention to sensitivity is that in this research problem, we care more about true positive. We want to target default records as many as possible. If we wrongly reject a non-default client loan application, we only loss interest gained from the loan. However, if we wrongly classify clients that would default into non-default group and lend money to them. We would loss the principal and interest. Therefore, sensitivity is more important than specificity in this case. The following shows the detailed performance information of each model.

**Logistic Regression**

**Code**

accuracy\_lm **=** accuracy\_score**(**test\_y**,** predict\_lm**)**

tn\_lm**,** fp\_lm**,** fn\_lm**,** tp\_lm **=** confusion\_matrix**(**test\_y**,** predict\_lm**).**ravel**()**

sensitivity\_lm **=** tp\_lm **/** **(**tp\_lm**+**fn\_lm**)**

**print(**"Accuracy is %s" **%**accuracy\_lm**)**

**print(**"Sensitivity is %s" **%**sensitivity\_lm**)**

**print(**confusion\_matrix**(**test\_y**,** predict\_lm**))**

#Logistic regression ROC

fpr\_lm**,**tpr\_lm**,**thresholds **=** roc\_curve**(**test\_y**,** predict\_lm**,**pos\_label**=**1**)**

auc\_lm **=** round**(**auc**(**fpr\_lm**,** tpr\_lm**),**4**)**

plt**.**plot**(**fpr\_lm**,**tpr\_lm**,**linewidth**=**2**,**label**=**"ROC"**)**

plt**.**xlabel**(**"false presitive rate"**)**

plt**.**ylabel**(**"true presitive rate"**)**

plt**.**ylim**(**0**,**1**)**

plt**.**xlim**(**0**,**1**)**

plt**.**plot**([**0**,** 1**],** **[**0**,** 1**],** '--'**,** color**=(**0 **,** 0**,** 1**))**

plt**.**legend**(**loc**=**4**)**

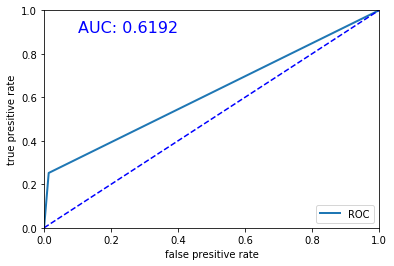
plt**.**text**(**0.1**,**0.9**,**'AUC: %s' **%**auc\_lm**,**fontdict**={**'size'**:**'16'**,**'color'**:**'b'**})**

plt**.**show**()**

Table 3: Accuracy, sensitivity and confusion matrix for logistic regression

|  |  |  |  |
| --- | --- | --- | --- |
| Accuracy: 0.9285  Sensitivity: 0.2523 | | Reference | |
| Non-default (0) | Default (1) |
| Prediction | Non-default (0) | 1284 | 18 |
| Default (1) | 83 | 28 |

Figure 3: ROC for logistic regression



**Random Forest**

**Code**

accuracy\_rf **=** accuracy\_score**(**test\_y**,** predict\_rf**)**

tn\_rf**,** fp\_rf**,** fn\_rf**,** tp\_rf **=** confusion\_matrix**(**test\_y**,** predict\_rf**).**ravel**()**

sensitivity\_rf **=** tp\_rf **/** **(**tp\_rf**+**fn\_rf**)**

**print(**"Accuracy is %s" **%**accuracy\_rf**)**

**print(**"Sensitivity is %s" **%**sensitivity\_rf**)**

**print(**confusion\_matrix**(**test\_y**,** predict\_rf**))**

#random forest ROC

fpr\_rf**,**tpr\_rf**,**thresholds **=** roc\_curve**(**test\_y**,** predict\_rf**,**pos\_label**=**1**)**

auc\_rf **=** round**(**auc**(**fpr\_rf**,** tpr\_rf**),**4**)**

plt**.**plot**(**fpr\_rf**,**tpr\_rf**,**linewidth**=**2**,**label**=**"ROC"**)**

plt**.**xlabel**(**"false presitive rate"**)**

plt**.**ylabel**(**"true presitive rate"**)**

plt**.**ylim**(**0**,**1**)**

plt**.**xlim**(**0**,**1**)**

plt**.**plot**([**0**,** 1**],** **[**0**,** 1**],** '--'**,** color**=(**0 **,** 0**,** 1**))**

plt**.**legend**(**loc**=**4**)**

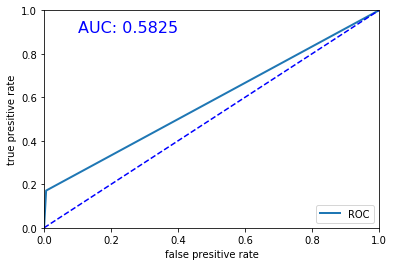
plt**.**text**(**0.1**,**0.9**,**'AUC: %s' **%**auc\_rf**,**fontdict**={**'size'**:**'16'**,**'color'**:**'b'**})**

plt**.**show**()**

Table 4: Accuracy, sensitivity and confusion matrix for random forest

|  |  |  |  |
| --- | --- | --- | --- |
| Accuracy: 0.9292  Sensitivity: 0.1712 | | Reference | |
| Non-default (0) | Default (1) |
| Prediction | Non-default (0) | 1294 | 8 |
| Default (1) | 92 | 19 |

Figure 4: ROC for random forest



**Bagging**

**Code**

accuracy\_bag **=** accuracy\_score**(**test\_y**,** predict\_bag**)**

tn\_bag**,** fp\_bag**,** fn\_bag**,** tp\_bag **=** confusion\_matrix**(**test\_y**,** predict\_bag**).**ravel**()**

sensitivity\_bag **=** tp\_bag **/** **(**tp\_bag**+**fn\_bag**)**

**print(**"Accuracy is %s" **%**accuracy\_bag**)**

**print(**"Sensitivity is %s" **%**sensitivity\_bag**)**

**print(**confusion\_matrix**(**test\_y**,** predict\_bag**))**

#bagging ROC

fpr\_bag**,**tpr\_bag**,**thresholds **=** roc\_curve**(**test\_y**,** predict\_bag**,**pos\_label**=**1**)**

auc\_bag **=** round**(**auc**(**fpr\_bag**,** tpr\_bag**),**4**)**

plt**.**plot**(**fpr\_bag**,**tpr\_bag**,**linewidth**=**2**,**label**=**"ROC"**)**

plt**.**xlabel**(**"false presitive rate"**)**

plt**.**ylabel**(**"true presitive rate"**)**

plt**.**ylim**(**0**,**1**)**

plt**.**xlim**(**0**,**1**)**

plt**.**plot**([**0**,** 1**],** **[**0**,** 1**],** '--'**,** color**=(**0 **,** 0**,** 1**))**

plt**.**legend**(**loc**=**4**)**

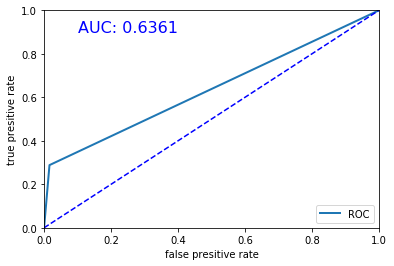
plt**.**text**(**0.1**,**0.9**,**'AUC: %s' **%**auc\_bag**,**fontdict**={**'size'**:**'16'**,**'color'**:**'b'**})**

plt**.**show**()**

Table 5: Accuracy, sensitivity and confusion matrix for bagging

|  |  |  |  |
| --- | --- | --- | --- |
| Accuracy: 0.9292  Sensitivity: 0.2883 | | Reference | |
| Non-default (0) | Default (1) |
| Prediction | Non-default (0) | 1281 | 21 |
| Default (1) | 79 | 32 |

Figure 5: ROC for bagging



**Support Vector Machine (SVM)**

**Code**

accuracy\_svm **=** accuracy\_score**(**test\_y**,** predict\_svm**)**

tn\_svm**,** fp\_svm**,** fn\_svm**,** tp\_svm **=** confusion\_matrix**(**test\_y**,** predict\_svm**).**ravel**()**

sensitivity\_svm **=** tp\_svm **/** **(**tp\_svm**+**fn\_svm**)**

**print(**"Accuracy is %s" **%**accuracy\_svm**)**

**print(**"Sensitivity is %s" **%**sensitivity\_svm**)**

**print(**confusion\_matrix**(**test\_y**,** predict\_svm**))**

#svm ROC

fpr\_svm**,**tpr\_svm**,**thresholds **=** roc\_curve**(**test\_y**,** predict\_svm**,**pos\_label**=**1**)**

auc\_svm **=** round**(**auc**(**fpr\_svm**,** tpr\_svm**),**4**)**

plt**.**plot**(**fpr\_svm**,**tpr\_svm**,**linewidth**=**2**,**label**=**"ROC"**)**

plt**.**xlabel**(**"false presitive rate"**)**

plt**.**ylabel**(**"true presitive rate"**)**

plt**.**ylim**(**0**,**1**)**

plt**.**xlim**(**0**,**1**)**

plt**.**plot**([**0**,** 1**],** **[**0**,** 1**],** '--'**,** color**=(**0 **,** 0**,** 1**))**

plt**.**legend**(**loc**=**4**)**

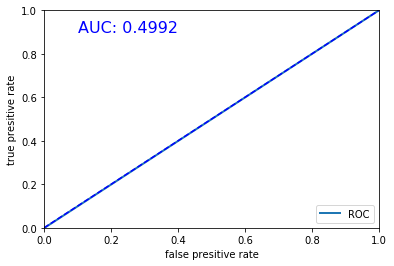
plt**.**text**(**0.1**,**0.9**,**'AUC: %s' **%**auc\_svm**,**fontdict**={**'size'**:**'16'**,**'color'**:**'b'**})**

plt**.**show**()**

Table 6: Accuracy, sensitivity and confusion matrix for SVM

|  |  |  |  |
| --- | --- | --- | --- |
| Accuracy: 0.9200  Sensitivity: 0.0000 | | Reference | |
| Non-default (0) | Default (1) |
| Prediction | Non-default (0) | 1300 | 2 |
| Default (1) | 111 | 0 |

Figure 6: ROC for SVM



**Comparison**

Figure 7: Accuracy comparison among four models

The column chart shows that bagging model and random forest got the highest accuracy rate (0.9292) followed by logistic regression (0.9285). Besides, SVM got the lowest accuracy rate (0.92) among our models.

Figure 8: Sensitivity comparison among four models

The chart above states the similar result that bagging model had the highest sensitivity rate (0.2883). Following is the logistic regression (0.2523).

Figure 9: AUC comparison among four models

The chart shows that bagging model had the highest AUC score (0.6361). Following is the logistic regression (0.6192).

# **CONCLUSION**

Based on a loan for small business dataset from a commercial bank, this report come up with four models, namely logistic regression, random forest, bagging and support vector machine, trying to predict whether a client would default. We splitted the sample into training dataset and testing dataset to train the models and evaluate the performance of each one of them. This report applied three metrics that are accuracy, sensitivity and confusion matrix, to compare the performance. The result showed that bagging model got the highest accuracy rate and sensitivity rate. So, we would use bagging algorithm to build our default prediction model.

The bagging model is the best one among four models we proposed, but there are still some aspects that we can improve our prediction model.

**1. Expand the dataset**

In this report, the final dataset only contains 4709 observations and 70% of them (3296) were used as training set to build the model. With the time passing by, the bank can collect more loan’s detailed information which can improve the predictive accuracy dramatically.

**2. Collect more features**

In this dataset, we only apply some basic features of each borrower, such as geographic information, year of operation, employee amount and so on. If the bank can get more detailed information such as their default history, current debt balance, and financial status information, we can improve our model to a higher level.

# ETHICAL ISSUE - THE UTILITARIAN APPROACH

This report applied utilitarian approach to analysis the default predictive model we build.

**Benefits**

1. Help banks and other financial lenders better manage risk of loans and debts.

2. Save loss cost. Banks can avoid more loan default to save loss cost.

3. Make non-default clients easier to apply for loan.

**Harms**

1. The rich get richer and the poor get poorer. The clients which are more likely to default are those that are more desperate for money.

2. The model would provide wrong result for some outliners.

Based on the benefits and harms analysis, we can know that the predictive model would help banks conduct risk management and allocate their resources more efficiently but it would make some client who are desperate for loan loss opportunities and also let banks loss some potential clients. In my opinion, we can mitigate the harms using some measures such as setting aside some allowance for those clients with terrible default history but promising future. Therefore, we think our model has more benefits than harms and is ethical to our society.