Credit Default Prediction with Machine Learning

Quantitative Analysis Report

CID: 02136857

MSc Risk Management and Financial Engineering

2021-2022 Applied Project

Client Specification

Hello Ted,

Our client is a US lending company. One of its main daily tasks is to assess the risk of a loan applicant defaulting so that it can make a decision on loan approval accordingly. Traditionally, the evaluation of every single loan is done by hand by professional credit analysts using the loan applicant's profile case by case. Now our client seeks to develop models using newer machine learning techniques for improved efficiency and hopefully improved prediction accuracy.

Could you please research on the feasibility of using machine learning models for such predicting loan default risk?

Kind regards,
Adam

Hello Adam,

I will develop several machine learning models for predicting loan default, evaluate their performance and report the results.

Best,
Ted

Table of Contents

Word Count (main body): 2979

1. Introduction

Loan is a vital financial instrument that enables money on the market to flow to where it can be used to generate the most value. There are countless borrowing and lending activities every day. Borrowers rely on loans to pay for their immediate needs and lenders lend out their spare money to gain an interest profit on the money. However, there are usually credit risks associated with lending activities. There are periods in history when the number of loan defaults has skyrocketed. A loan default is when a borrower fails to repay a loan. Credit default risk is the possibility of a lender suffering a loss due to a loan default. For most lending companies, the loss suffered from granting loans to risky borrowers (called credit loss) accounts for the major proportion of their financial loss. Traditionally, to assess credit default risk, each borrower's profile needs to be thoroughly analyzed by human credit analysts, which is a slow and effort consuming task. Fortunately, we now have a powerful tool at our disposal – machine learning, which has the potential to find and interpret patterns in complex data at speed.

In this paper, the performance of various machine learning models at predicting loan default will be evaluated based on following performance metrics.

2. Performance Metrics

The metrics we will be using to measure the performance of the models are Accuracy, Confusion Matrix, F1 Score and ROC-AUC score.

2.1 Accuracy

Accuracy of a model is simply the proportion of predictions it makes being correct. Note however, if used solely, this metric can be misleading when the dataset is imbalanced. For example, on a dataset containing 99% non-defaults and only 1% defaults, an extremely high accuracy could be achieved simply by predicting all loans as 'non-default'.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Number\ of\ total\ predictions}$$

2.2 Confusion Matrix

We identify 'default' as positive and 'non-default' as negative. A confusion matrix is made up of the follow 4 numbers:
True Positive: the number of positive (default) predictions being actually positive.
True Negative: the number of negative (non-default) predictions being actually negative.
False Positive: the number of positive (default) predictions being actually negative (non-default).

False Negative: the number of negative (non-default) predictions being actually positive (default).

	Predicted: **Positive**	Predicted: **Negative**
Actual: **Positive**	True Positive	False Negative
Actual: **Negative**	False Positive	True Negative

Table 1: Confusion Matrix

2.3 F1 Score

F1 Score is the harmonic mean of Recall (or True Positive Rate) and Precision, and it is a number between 0 and 1. Higher F1 Score indicates better model performance at distinguishing between the two classes.

$$F_1\ Score = \frac{2(Precision \cdot Recall)}{Precision + Recall}$$

$$Precision = \frac{True\ Positve}{True\ Positve + False\ Positive}$$

$$Recall = \frac{True\ Positve}{True\ Positve + False\ Negative}$$

2.4 ROC-AUC Score

ROC (Receiver Operating Characteristic) curve is a plot of the True Positive Rate (or Recall) against the False Positive Rate at different levels of classification threshold.

$$False\ Positve\ Rate = \frac{Fasle\ Positve}{Fasle\ Positve + True\ Negative}$$

AUC (Area Under Curve) is the area under the ROC curve and is an indicator of the performance of the model. The larger the area (ROC-AUC score), the better the model is at distinguishing between different classes. At model that makes random predictions has an AUC score of 0.5. A near perfect model has an AUC score of close to 1.

3. Data

The data we are going to work with in this report is a portion of the datasets that record the details of all the accepted loans between 2007 and 2020 released by Lending Club, a US peer-to-peer lending company. In this section we will explore and

visualize the data to get a better understanding of it. We will also explain the data preprocessing and feature engineering steps.

3.1 Dependent Variables

The dependent variable that we aim to predict is the loan_status. It can be either 'Fully Paid' or 'Charged Off'. We discover that the data, when classified by 'Fully Paid'(non-default) and 'Charged Off'(default), appears highly unbalanced as there are much more non-defaults than defaults, as shown in Figure 1.

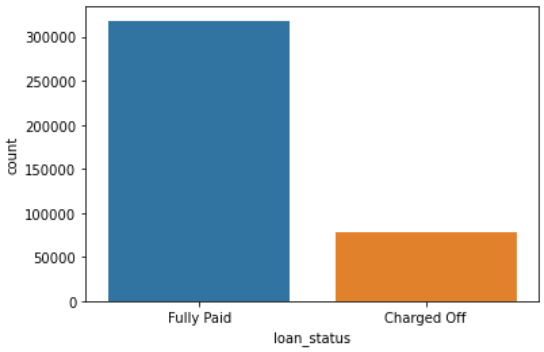

Figure 1: Data before Balancing

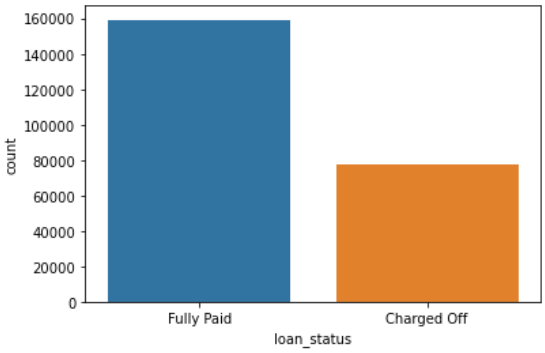

Figure 2: Data after Balancing

As mentioned previously, this can potentially make the model accuracy result misleading since the null accuracy is too high (null accuracy is the accuracy of simply predicting all outcomes as 'non-default', the most frequent class in the data). Also, when the data is skewed, the machine learning algorithms might not have enough 'default' samples to study in order to learn about the characteristics of defaulted loans. So, we randomly remove fully repaid loan observations to reduce the size of non-default samples. After that, as shown in Figure 2, the data becomes more balanced.

3.2 Independent Variables (Quantitative)

Among the 26 independent variables in the data, there are 12 quantitative variables:

1. 'loan_amnt': loan amount (in dollars)
2. 'int_rate': interest rate of the loan
3. 'installment': monthly payment amount of the loan
4. 'annual_inc': annual income of the borrower
5. 'dti': borrower's debt to income ratio
6. 'open_acc': number of open credit accounts the borrower owns
7. 'total_acc': number of total credit accounts the borrower owns
8. 'mort_acc': number of mortgage accounts the borrower owns
9. 'pub_rec': number of public derogatory records the borrower has
10. 'revol_bal': total revolving balance of the borrower
11. "revol_util: borrower's credit utilization rate
12. 'pub_rec_bankruptcies': number of public bankruptcy records the borrower has

Figure 3 shows the correlations between all the quantitative variables. It appears that 'installment' (monthly payment) correlates almost perfectly with 'loan amount', which is reasonable. So, we drop variable 'installment'.

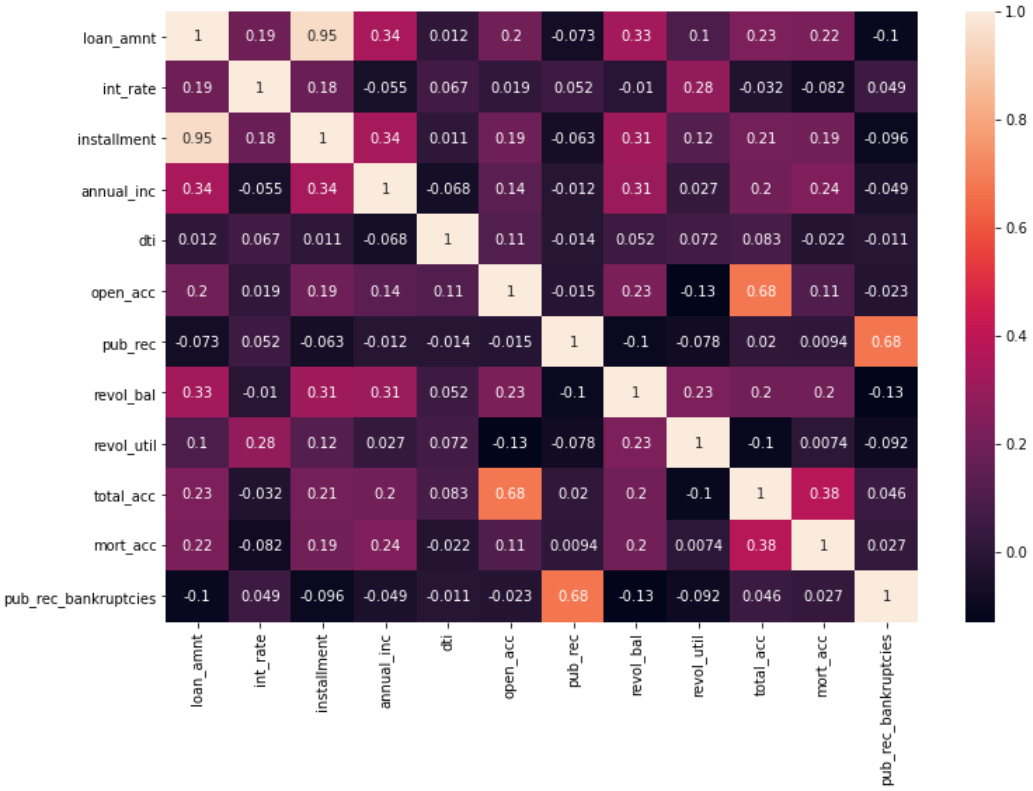

Figure 3: Correlation Heatmap

Figure 4 shows the correlations between loan status and the quantitative variables (we define 'Charged Off' as 1 and 'Fully Paid' as 0).

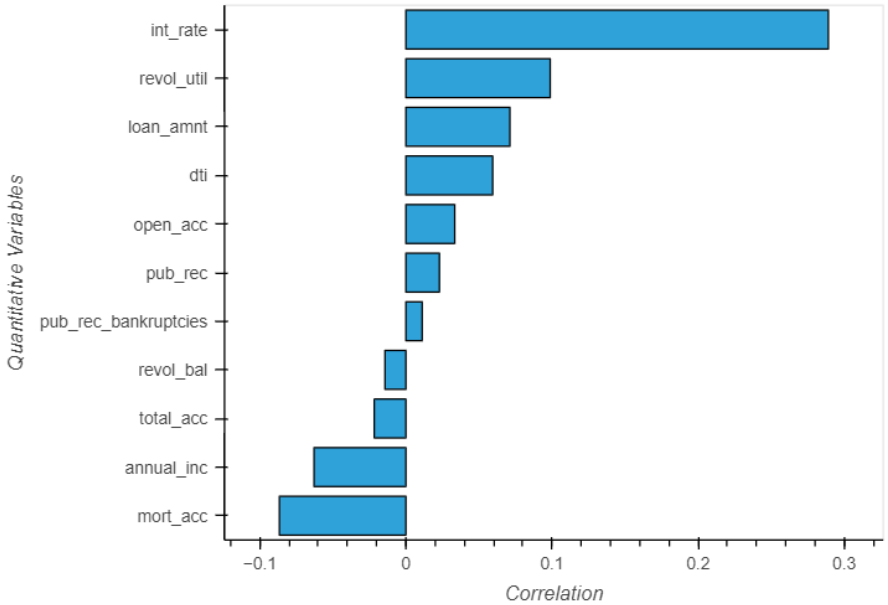

Figure 4: Correlation between Loan status and Quantitative Variables

It appears that, from the loans' side, greater loan amount and higher interest rate can cause loans to be defaulted more often, which is as expected as it is more difficult to repay those kinds of loans. From the borrowers' side, borrowers with more credit accounts (especially mortgage accounts), higher income and higher revolving balance will be more likely to fully repay their loans. Borrowers with more publicly recorded bankruptcies, more public derogatory records, more open credit accounts, higher debt to income ratio and higher credit utilization rate are more likely to default.

3.3 Independent Variables (Qualitative)

There are 14 qualitative (categorical) variables in the data:

1. 'term': duration of the loan, which can be either 36 months or 60 months
2. 'grade': grade assigned by Lending Club
3. 'sub_grade': sub grade assigned by Lending Club
4. 'home_ownership': status of the borrower's home, which can be 'OWN', 'RENT', 'MORTGAGE', or 'OTHER'
5. 'verification_status': shows whether the borrower's income is verified by Lending Club. Possible values are 'Verified', 'Source Verified', and 'Not Verified'.
6. 'purpose': intended purpose of the loan
7. 'initial_list_status': Initial listing status of the loan, which can be 'w' (whole) or 'f' (fractional)
8. 'application_type': shows whether the loan is requested by a single borrower or by two co-borrowers. Possible values are 'INDIVIDUAL' and 'JOINT'
9. 'address': borrower's address
10. 'earliest_cr_line': opening date of the earliest credit account of the borrower is opened. Values are in the form of 'month-year' such as 'Aug-2007'.
11. 'emp_title': borrower's job title
12. 'emp_length': borrower's employment length, which can take 11 different values ('< 1 year', '1 year', '2 years', …, '10 years', '10+ years')
13. 'title': title of the loan
14. 'issue_d': issue date of the loan

Among those, the following 5 variables are dropped as they are considered not useful for making predictions:

- 'issue_d': since when making prediction for a new loan, its issue date cannot be known in advance
- 'grade': since variable 'sub_grade' is more detailed and contains strictly more information than it
- 'emp_title': since there are too many unique job titles in the data which make it difficult to convert to dummy variable
- 'emp_length': since the default rates are very close (at around 33%) across all different employment lengths, which implies that this variable has little influence over default likelihood
- 'title': since it contains roughly the same information as the variable 'purpose'

Figure 5 show the number of defaults and non-defaults in each subgrade category. With no surprise, higher grade loans have low default rates while lower grade loans are defaulted much more often.

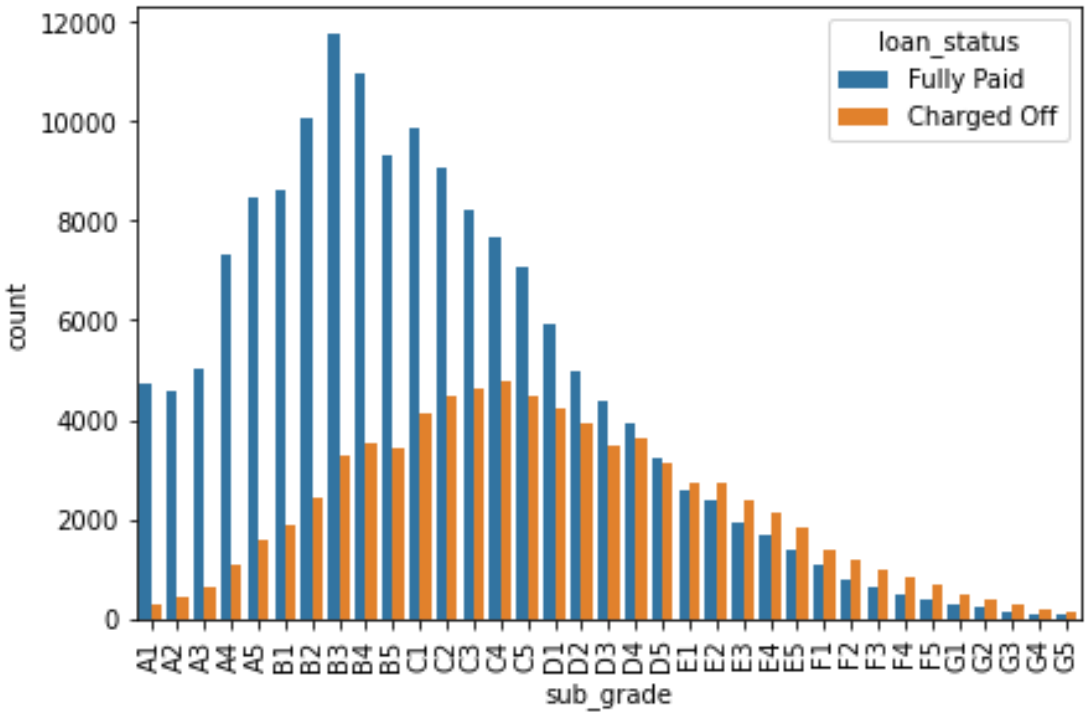

Figure 5: Loan Status by Subgrades

We can see from figure 6 that loans with longer term length are more likely to be defaulted. We see from figure 7 that borrowers whose income is verified or source-verified are less likely to default.

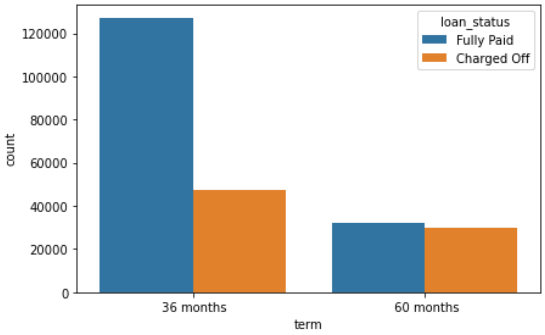

Figure 6: Loan Status by Term

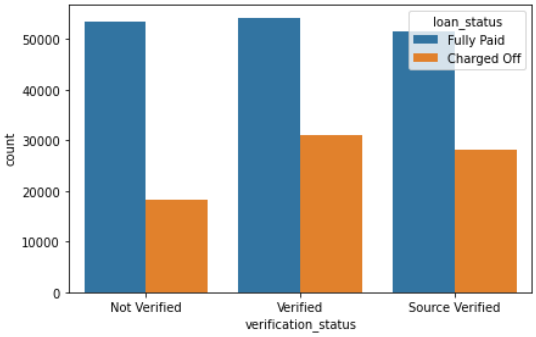

Figure 7: Loan Status by Verification Status

Figure 8 shows that Borrowers who own their home are more likely to default than those who have mortgages on their home and is less likely to default than those who rent their home. Figure 9 shows that default ratio on loans with initial list status of 'whole' is roughly similar to that on loans with initial list status of 'fractional'

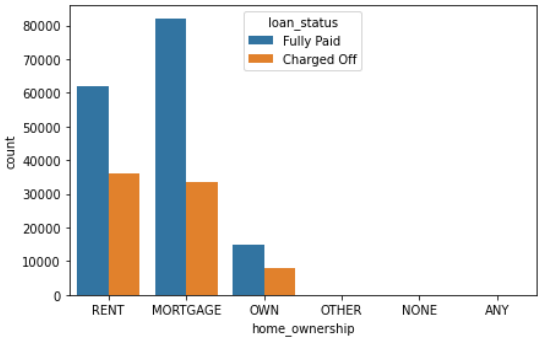

Figure 8: Loan Status by Home Ownership

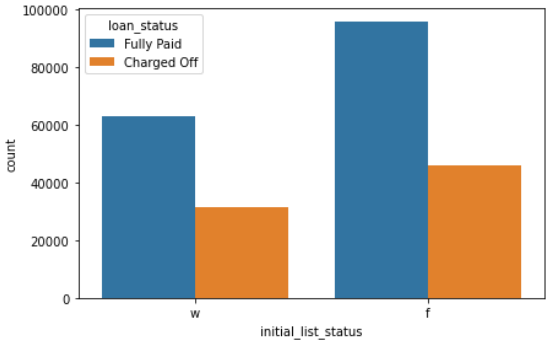

Figure 9: Loan Status by Initial Listing Status

A new variable 'zip_code' is created by extracting the zip code from variable 'address'. Then variable 'address' is dropped. Variable 'term' is transferred from string to integer (i.e. '36 months' becomes '36' and '60 months' becomes '60'). Dummy variables are created for categorical variables 'sub_grade, 'home_ownership', 'verification_status', 'purpose', 'initial_list_status', and 'application_type'.

3.4 Other Data Preprocessing Steps

Outliers: outliers are identified and removed.
Missing values: missing values are delt with by either removing the rows (observations) they below to or filled it in based on the average values of the columns (variables) they belong to.
Duplicates: duplicated rows are checked for and removed.
Rescaling: the entire dataset is normalized as the final step.

4. Methodology

In this section, the three machine learning methods evaluated in this report will be briefly explained. They are Neural Networks, Random Forest and XGBoost. We use 70% of data for training and 30% for testing.

4.1 Neural Network

Neural Networks, or more formally Artificial Neural Networks, are algorithms inspired by the architecture that supports information flow in a biological brain. A Neural Network is made up of a collection of nodes/units called neurons that are partially or fully connected to each other and form a graph that is directed, weighted, and consists of three types of sequential layers: an input layer, an output layer, and zero or more hidden layers in between. Every node is connected to zero or more nodes in adjacent layers. As the first layer in the network, the input layer contains the inputs to the model. Every node at one of the hidden layers has a weight assigned for each connected node at the previous layer. After the node receives the information inputs from its connected

nodes in the previous layer, it sums up the weighted inputs, adds a bias, applies an activation function (such as Sigmoid function or Rectified Linear Unit) and then pass the result to its connected nodes at the next layer. As the final layer in the network, the output layer contains the outputs (predictions) make by the model. (Acharya, 2022)

There are 2 phases when training a Neural Network: forward propagation and backward propagation. In forward propagation, the model goes through the above process using the input to reach a predicted output and calculates the error of the prediction. Then in backward propagation, the partial derivatives of the error with respect to the all the weights and biases in the network is calculated for performing gradient descend to minimize the model error through changing those weights and biases. (Gad, 2022)

Our Neural Network model has 3 hidden layers with 200 hidden units in each. It uses 'Adam' Optimizer, learning rate of 0.001, and dropout rate of (0.1, 0.2, 0.1, 0).

4.2 XGBoost

Different from traditional machine learning methods that aims to pick the best models to train, XGBoost (Extreme Gradient Boosting) is a type of ensemble algorithm that combines multiple base learners, usually weak learners (models that perform only slightly better than random guessing), to reach a final prediction and achieve high overall performance. It uses both bagging and boosting processes.

Bagging (Bootstrap Aggregation): Averaging the predictions given by the base learners trained on various bootstrap samples of original data. It can decrease overall variance. Usually, averaging is used for regression problem and majority voting is used for classification problem.

Boosting: Training base learners sequentially on various subsets of original data that are weighted based on the previous classification correctness. If a datapoint has been wrongly classified by the previous base learner, a higher weight will be assigned to it. In another word, the data used to train each base learner depends on the performance of the previous base leaner. Normally, the model is initialized by letting the first base learner just predicting a constant value and thus will have high bias. Then, as more base learners are added to it one by one, the bias of the model will reduce progressively. (Hachcham, 2022)

XGBoost can handle both classification and regression problem and is able to study from both linear and non-linear patterns in data. It is optimized from the Gradient Boosting algorithm in the way that its boosting process is fully parallelized, so the training time is hugely reduced as all available CPU cores can be utilized during tree construction process. To avoid overfitting, the model supports L1 (Lasso) and L2 (Ridge) regularization. However, it is worth to take caution that XGBoost is sensitive to outliers. (Hachcham, 2022)

In this report, we use XGBoost model with 200 estimators (200 trees in the ensemble), learning rate of 0.2, gamma of 4 (regularization strength), max depth of 5 (model

complexity), and set the rest of hyperparameters at default.

4.3 Random Forest

Random Forest is also a type of ensemble algorithm and is easy to interpret. It consists of multiple decision trees and combines their outputs to come to a final prediction. It uses Bagging process to let each decision tree trains on a different bootstrap subset of original data and considers only a randomly selected subset of all variables for making decisions (splitting at each node). Although an individual decision tree along is a weak learner that is very sensitive to the data it is trained on and is prone to error, when multiple lowly correlated trees combine together, they will likely offset each other's individual error and thus increase overall accuracy and reduce risk of overfitting, similar to how individual stock risk can be hedged in a large well diversified stock portfolio. (Yiu, 2019)

Random forest can be used for both classification and regression problem. In our case, since it is a classification problem (default or non-default), a majority voting mechanism will be used to collect outputs from all the decision trees to produce a single result. That is, each tree makes its own class prediction (votes for a class) and the class with the most votes will be the overall model's prediction. In this report, we use Random Forest model with 200 decision trees and with the rest of hyperparameters set at default.

5. Results

After training and testing the models, the results about their performance are as following:

Neural Network

	Predicted: **default**	Predicted: **non-default**
Actual: **default**	100157	5106
Actual: **non-default**	21454	30119

Table 2: NN Training Confusion Matrix

	Predicted: **default**	Predicted: **non-default**
Actual**:** **default**	49883	2574
Actual: **non-default**	10577	14978

Table 3: NN Test Confusion Matrix

On training data, Neural Network gets an accuracy score of 83.07%, a confusion matrix as shown in table 2, and an F1 score of 0.69.
On test data, Neural Network gets an accuracy score of 83.14%, a confusion matrix as shown in table 3, an F1 score of 0.69 and a ROC-AUC score of 0.906.
The Neural Network model takes significantly longer time to train than the other two models.

XGBoost

On training data, XGBoost gets an accuracy score of 84.50%, a confusion matrix as shown in Figure 10, and an F1 score of 0.73.
On test data, XGBoost achieves an accuracy score of 83.41%, a confusion matrix as shown in Figure 11, an F1 score of 0.71 and a ROC-AUC score of 0.909.
The XGBoost Model is the fastest among all three models and achieve the highest accuracy, F1 score and ROC-AUC score on test data.

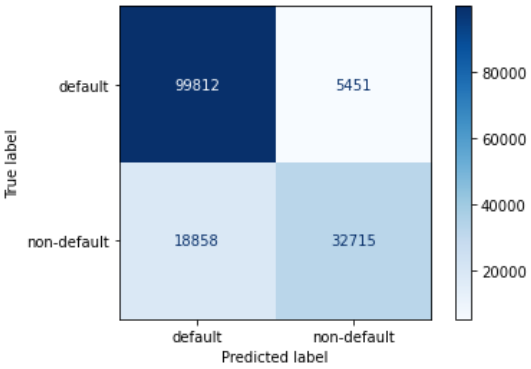

Figure 10: XGB Training Confusion Matrix

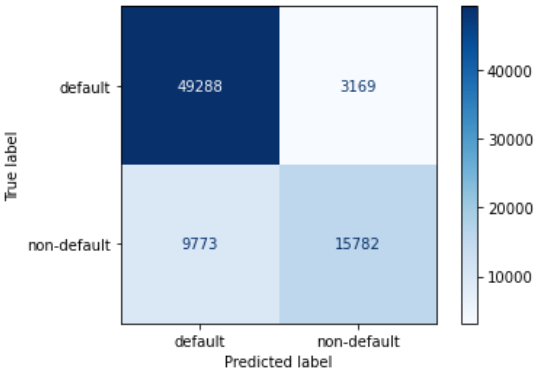

Figure 11: XGB Test Confusion Matrix

Random Forest

On training data, Random Forest gets an accuracy score of 100%, a confusion matrix as shown in Figure 12, and an F1 score of 1.00.
On test data, Random Forest gets an accuracy score of 82.89%, a confusion matrix as shown in Figure 13, an F1 score of 0.68, and a ROC-AUC score of 0.894.
It appears that Random Forest model has heavily overfitted the training data.

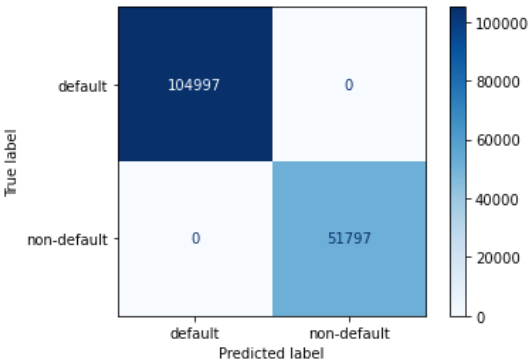

Figure 12: RF Training Confusion Matrix

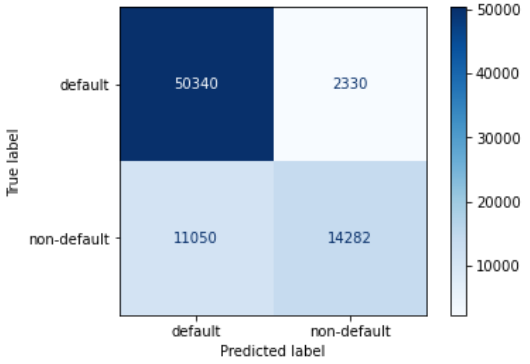

Figure 13: RF Test Confusion Matrix

6. Conclusion

It turns out machine learning models have the ability to perform reasonably well and fast for the default prediction problem, and we do recommend training an XGBoost model for helping to decide on loan approval in the future, as it appears to slightly

outperform the rest of models and is the fastest to train.

However, there does exist some limitations during this project. Firstly, there is a constraint to the size of the data used for training the models due to the limited available computing power. Secondly, also because of the shortage in computing power, we were not able to optimize all the hyperparameters used for the models nor be able to add complexity to the models (e.x. XGBoost models with more estimators and greater max-depth). Furthermore, because of the scale of this project, only three models are evaluated and compared. Some other great machine learning methods, such as Support Vector Machine, Light Gradient Boosting Machine, or even the simple Logistic Regression are also worth to be experimented with in the future.

Bibliography

Beshr, S., 2020. *A Machine Learning Approach To Credit Risk Assessment*. [online] Towards Data Science. Available at: <https://towardsdatascience.com/a-machine-learning-approach-to-credit-risk-assessment-ba8eda1cd11f> [Accessed 2 July 2022].

Sayjadah, Y., Hashem, I., Alotaibi, F. and Kasmiran, K., 2018. Credit Card Default Prediction using Machine Learning Techniques. *2018 Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA)*,.

Majumder, P., 2022. *Predicting Possible Loan Default Using Machine Learning*. [online] Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2022/04/predicting-possible-loan-default-using-machine-learning/> [Accessed 2 July 2022].

Zhang, M., 2021. *Credit Default Prediction based on Machine Learning Models*. [online] Analytics Vidhya. Available at: <https://medium.com/analytics-vidhya/credit-default-prediction-based-on-machine-learning-models-1717601600c9> [Accessed 2 July 2022].

Mahoney, A., 2020. *Credit Risk Modeling with Machine Learning*. [online] Towards Data Science. Available at: <https://towardsdatascience.com/credit-risk-modeling-with-machine-learning-8c8a2657b4c4> [Accessed 2 July 2022].

Vickery, R., 2021. *8 Metrics to Measure Classification Performance*. [online] Towards Data Science. Available at: <https://towardsdatascience.com/8-metrics-to-measure-classification-performance-984d9d7fd7aa> [Accessed 10 July 2022].

Sayah, F., 2021. *Lending Club Loan Defaulters Prediction*. [online] Kaggle. Available at: <https://www.kaggle.com/code/faressayah/lending-club-loan-defaulters-prediction/notebook> [Accessed 17 July 2022].

Saha, S., 2022. *XGBoost vs LightGBM: How Are They Different*. [online] neptuneblog.

Available at: <https://neptune.ai/blog/xgboost-vs-lightgbm> [Accessed 18 July 2022].

Hachcham, A., 2022. *XGBoost: Everything You Need to Know*. [online] neptuneblog. Available at: <https://neptune.ai/blog/xgboost-everything-you-need-to-know> [Accessed 18 July 2022].

Gad, A., 2022. *A Comprehensive Guide to the Backpropagation Algorithm in Neural Networks*. [online] neptuneblog. Available at: <https://neptune.ai/blog/backpropagation-algorithm-in-neural-networks-guide> [Accessed 20 July 2022].

Acharya, A., 2022. *Guide to Building Your Own Neural Network [With Breast Cancer Classification Example]*. [online] neptuneblog. Available at: <https://neptune.ai/blog/neural-network-guide> [Accessed 20 July 2022].

Yiu, T., 2019. *Understanding Random Forest*. [online] Towards Data Science. Available at: <https://towardsdatascience.com/understanding-random-forest-58381e0602d2> [Accessed 26 July 2022].