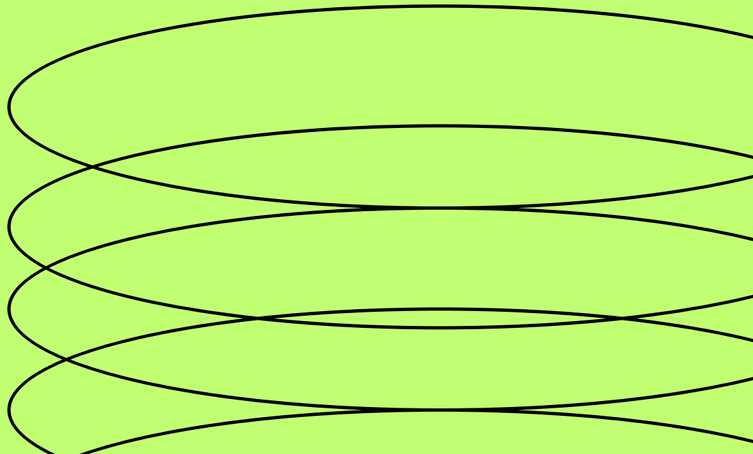
Nurkahfi Amran Rahmada

Loan Credit Assessment

Project Based Internship
Rakamin x ID/X Partners



Nurkahfi Amran Rahmada

Project Introduction

This project serves as final task for data scientist project based internship program hosted by Rakamin x ID/X Partners.

This project aims to create a machine learning model from loan data provided by companies consisting of accepted and rejected loan data which is expected to be able to predict credit risk. The dataset is collected by ID/X Partners from a company.

You can download the dataset in here
You can also see the loan credit data dictionary in here

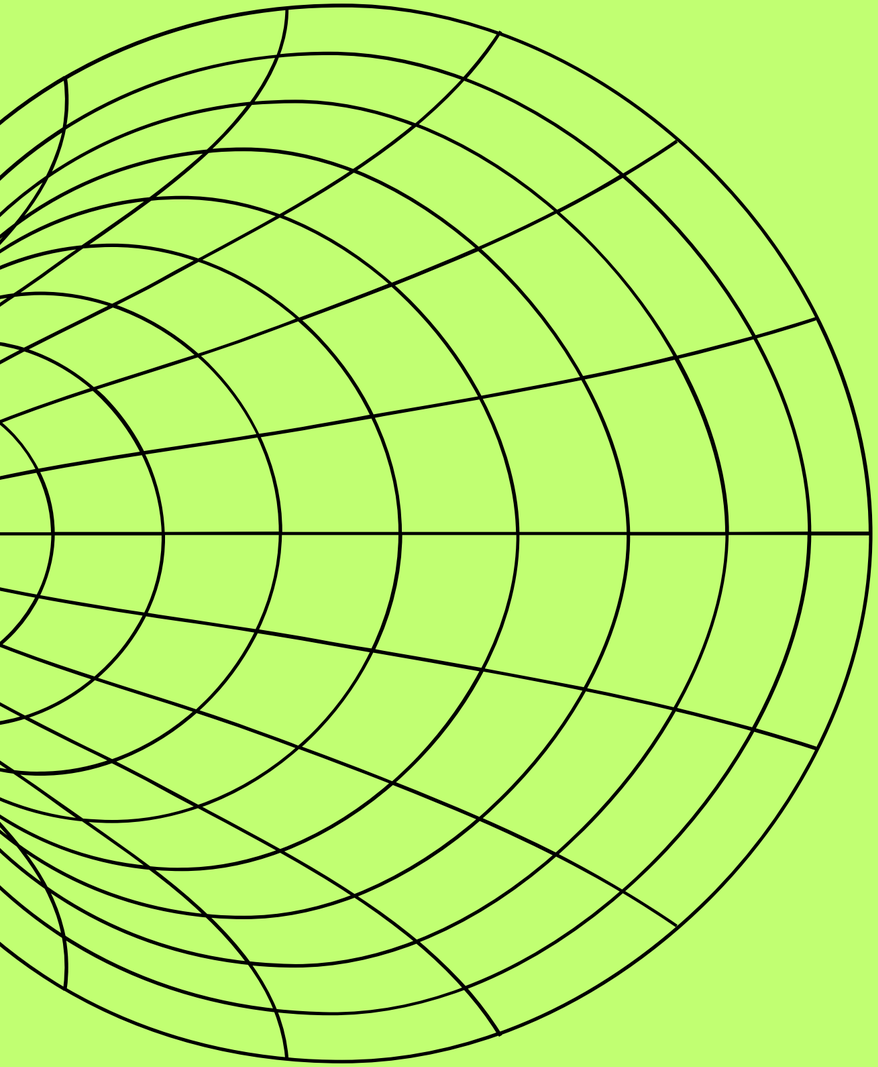
Nurkahfi Amran Rahmada

Business Understanding

Credit risk is the probability of a financial loss resulting from a borrower's failure to repay a loan. Essentially, credit risk refers to the risk that a lender may not receive the owed principal and interest, which results in an interruption of cash flows and increased costs for collection.

Banks can manage credit risk with several strategies. They can set specific standards for lending, including requiring a certain credit score from borrowers. Then, they can regularly monitor their loan portfolios, assess any changes in borrowers' creditworthiness, and make any adjustments.

Nurkahfi Amran Rahmada



Data Preparation

Data Understanding

What should we check :

- The number of rows and columns
  - The dataset has 466,285 rows and 74 columns
- The data type for each columns
  - The dataset has 22 categorical columns and 52 numerical columns
- Sample data
- Number of duplicated data
  - The dataset don’t have duplicated rows

This step aims to make us understand more about the dataset that we will process.

Feature Selection

At this stage, unimportant columns will be cleaned in the hope that it will reduce modeling time later. The following are the criteria for selecting columns :

- Active columns (not 100% missing values)
- Not a categorical data with high unique values (e.g emp_title, title, zip_code)
- Not a highly unbalanced class (e.g pymnt_plan with a ratio of 99.9 : 0.1)
- Not a column containing text (e.g url, desc)
- Not a column with high missing value (e.g mths_since_last_delinq), we limit to columns with < 50% missing values.
- Not a numerical data with high correlation (above 0.7)

Right now, we should have cleaned 42 columns and leaving 32 columns to worked on.

Data Transformation

Mostly, our data transformation steps will be applied to categorical data. There are some data that can be transformed in the same way.

term, grade, sub_grade, emp_length would be transformed into numerical data by classifying some values into number or by removing the string.

earliest_cr_line, issue_d, last_pymnt_d, next_pymnt_d, last_credit_pull_d would be transformed into regular date format. Then we will add 2 new features, pymnt_time and credit_duration.

- *pymnt_time* : this is the distance in months from the *last_pymnt_d* to the *next_pymnt_d*.
- *credit_duration* : this is the number of year between *last_credit_pull_d* and *earliest_cr_line*.

Data Transformation

home_ownership values will be merged ("ANY" and "NONE" to "OTHER") to reduce the result of one hot encoding later.

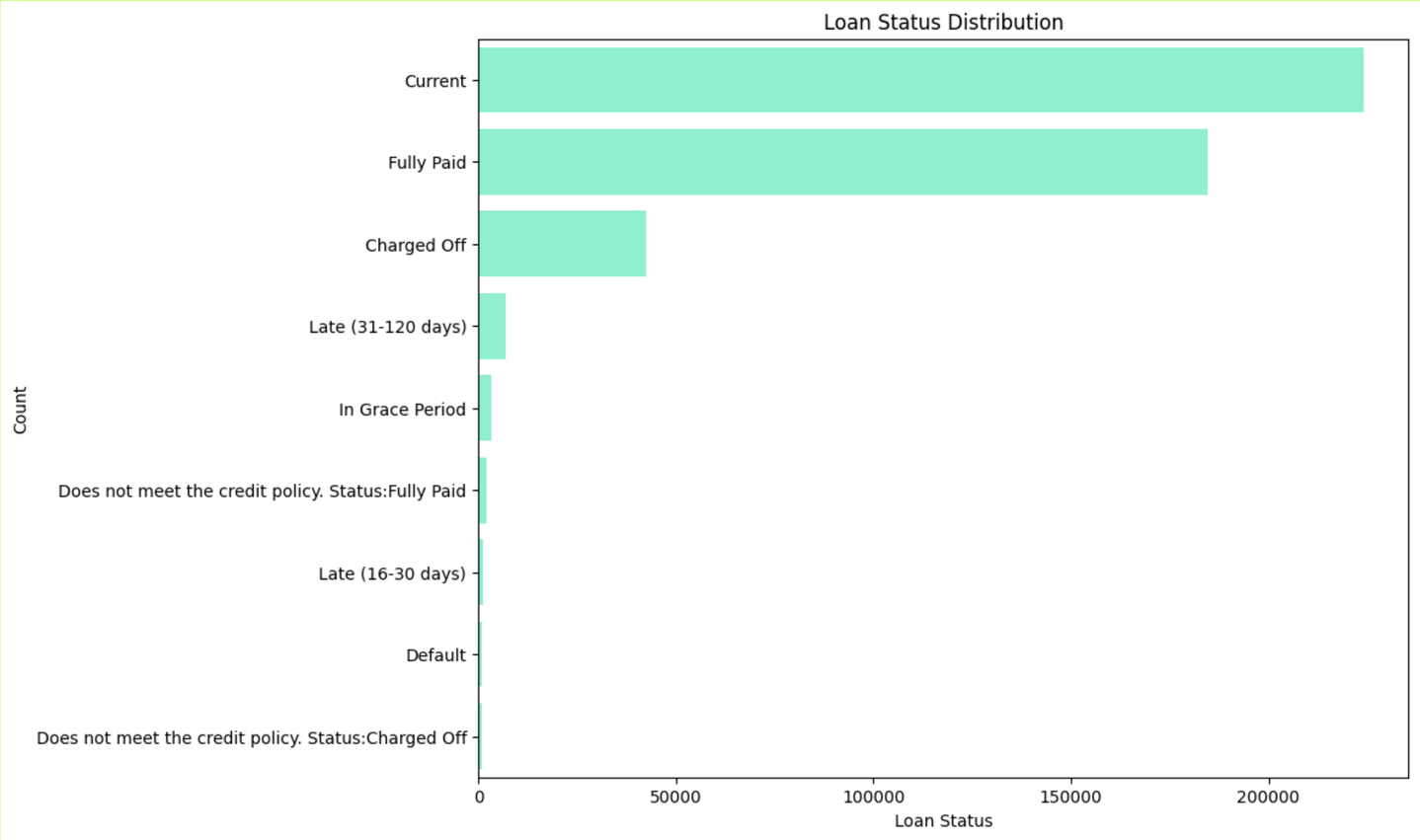
For loan_status, it will be left for now because it will be defined as our target variable later.

With this, our remaining categorical data are :

- home_ownership
- verification_status
- purpose
- initial_list_status

Defining Target Variable

We choose loan_status for our target variable. loan_status contains information about client's current status of the loan. Here are the distribution of unique values :



Defining Target Variable

We need to process loan_status to binary values for our model training later. So let's first classify the values into two groups (good loan and bad loan) based on wheter it is risky or safe.

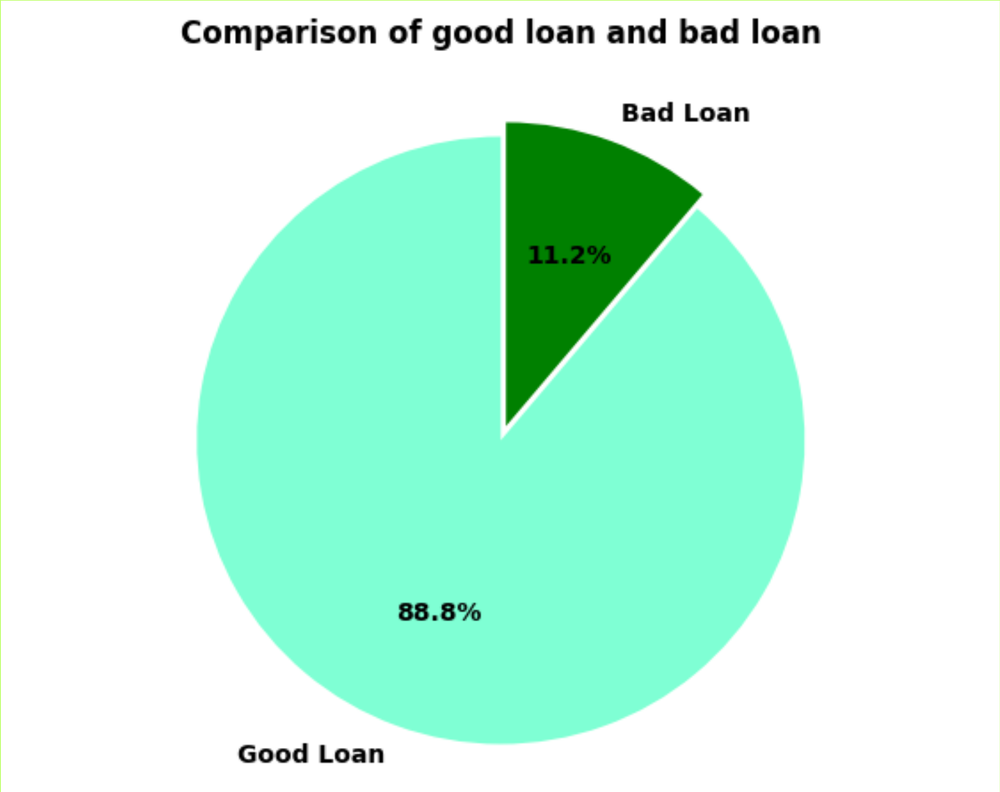
- Current : Safe
- Fully Paid : Safe
- Charged Off : Risk
- Late (31-120 days) : Risk
- In Grace Period : Safe
- Does not meet the credit policy. Status:Fully Paid : Safe
- Late (16-30 days) : Risk
- Default : Risk
- Does not meet the credit policy. Status:Charged Off : Risk

Then we labeled good loan as 1 and bad loan as 0.

Nurkahfi Amran Rahmada

Defining Target Variable

This is the comparison between good loan and bad loan.



It seems imbalanced, so we need to deal with this later on model training.

Handling Missing Values

Since some of the categorical data have transformed into numerical data and have the missing values filled, we will now just focusing on numerical data.

There are 27 columns with numerical data type that still have missing values. We will fill the missing values with mean value from each columns. With the help of SimpleImputer from library sklearn we will have the missing values filled in no time.

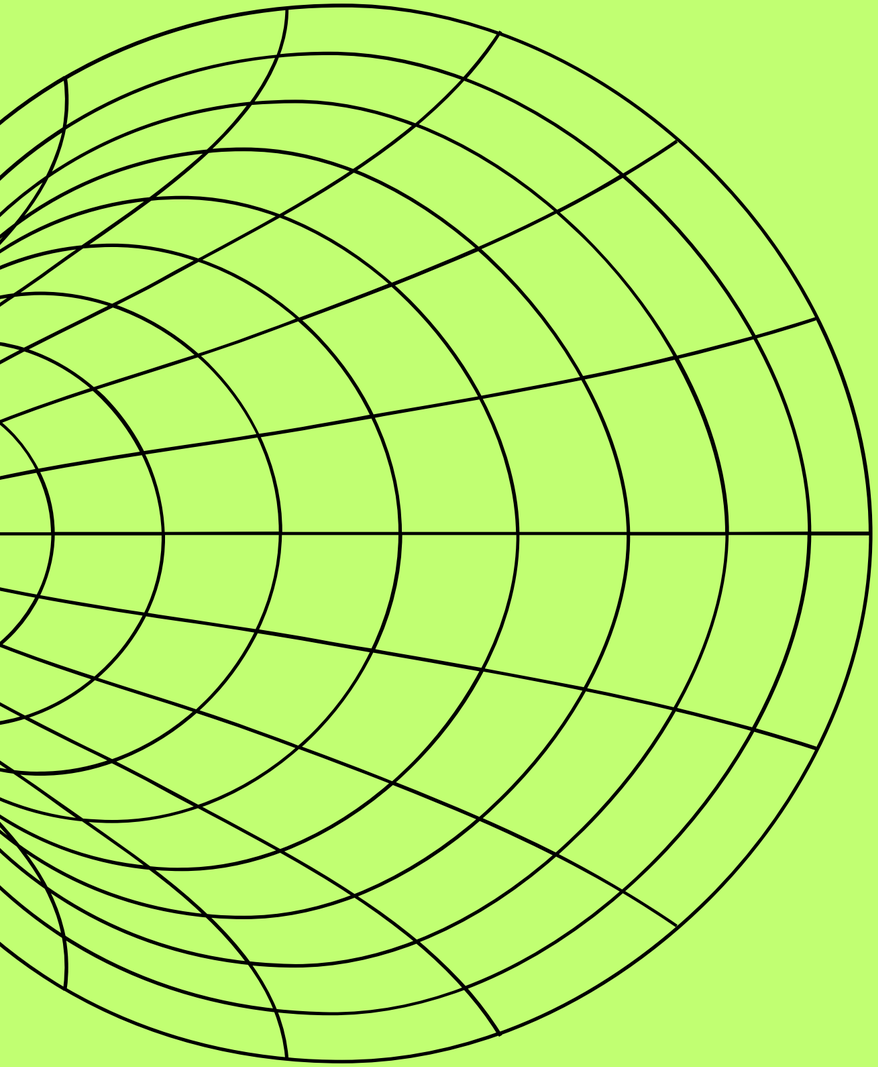
Data Encoding

All the encoding applied to categorical columns using one hot encoding which are applied to:

- home_ownership
- verification_status
- purpose
- initial_list_status

This is the end for data preparation steps, and right now we have 50 columns for training models.

Nurkahfi Amran Rahmada



Model Training

Model Selection

To start the model training, we need to determine what classification algorithm we will use to make the model later. To do this, we will need to test several classification algorithm and see which one get the accuracy better.

We choose 6 different algorithm to be tested for our model. All classification model are trained using preprocessed data and default setting for hyperparameter.

We examined which algorithm has the best performance using our preprocessed data (without balancing classes). We choose accuracy, number of mislabeled, Receiver Operating Characteristics (ROC), and Kolmogorov-Smirnov (KS) as a variable for our consideration.

Model Selection

Model	Accuracy	Mislabled	ROC	KS
Naive Bayes	92%	10979	0.86	0.5751
Logistic Regression	94%	8020	0.87	0.5922
K Nearest Neighbors	92%	10543	0.80	0.5073
Decision Tree	92%	11854	0.80	0.6062
Gradient Boosting	95%	6372	0.93	0.6953
XGBoost	95%	6480	0.93	0.6986

Model Selection

Our consideration lies between Gradient Boosting and XGBoost, because both this algorithm have high accuracy and low mislabeled classification. Then because the result almost tie we choose runtime as our last variable. The difference in runtime for both algorithm is very far.

Gradient Boosting	XGBoost
4 m 21.3 s	5.5 s

With this as a consideration, we choose XGBoost for our classification model.

Next, we will apply hyperparameter tuning to our model for better results.

Model Improvement

We will do hyperparameter tuning to our classification model. We choose max_depth, learning_rate, and n_estimators as our parameter grid.

We search the best parameters with help of GridSearchCV library with our XGBooster model as estimator and ROC AUC for scoring. We use our train data as a comparison.

Then we get our best hyperparameters :

- 'learning_rate': 0.1
- 'max_depth': 5
- 'n_estimators': 300

Model Improvement

By performing hyperparameter tuning, our model should have better performance. Here is the comparasion.

Before hyperparameter tuning:

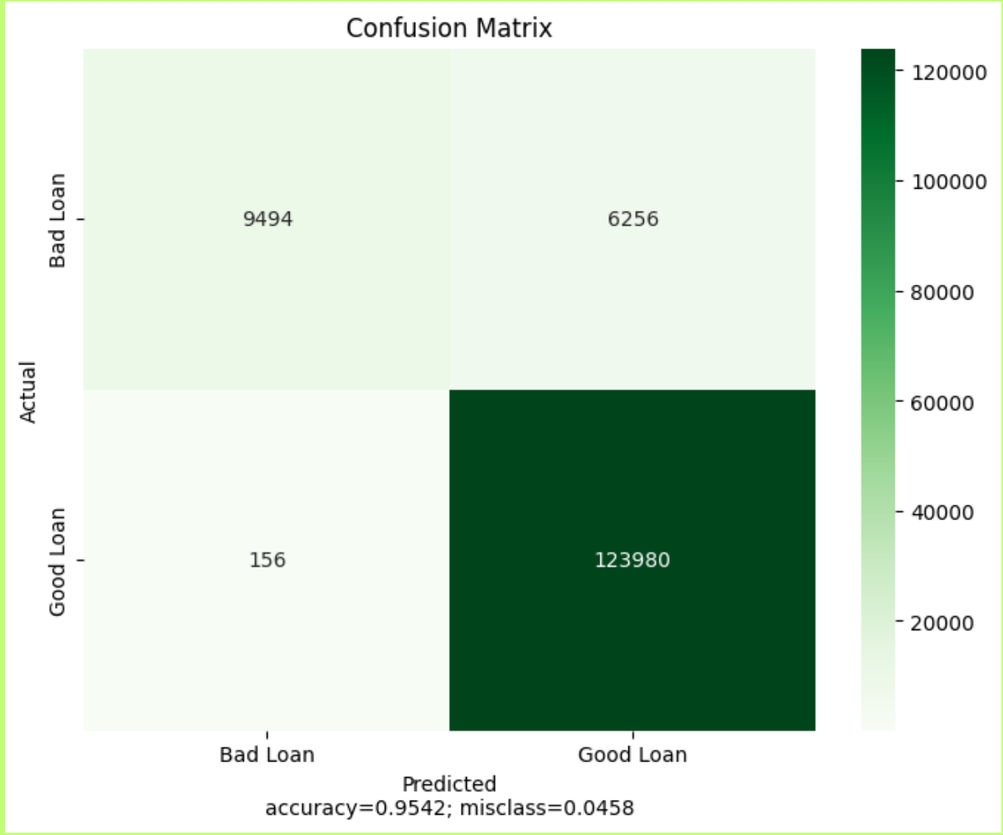
Accuracy	Mislabeled	ROC	KS
95%	6480	0.93	0.6986

After hyperparameter tuning:

Accuracy	Mislabeled	ROC	KS
95%	6412	0.94	0.7014

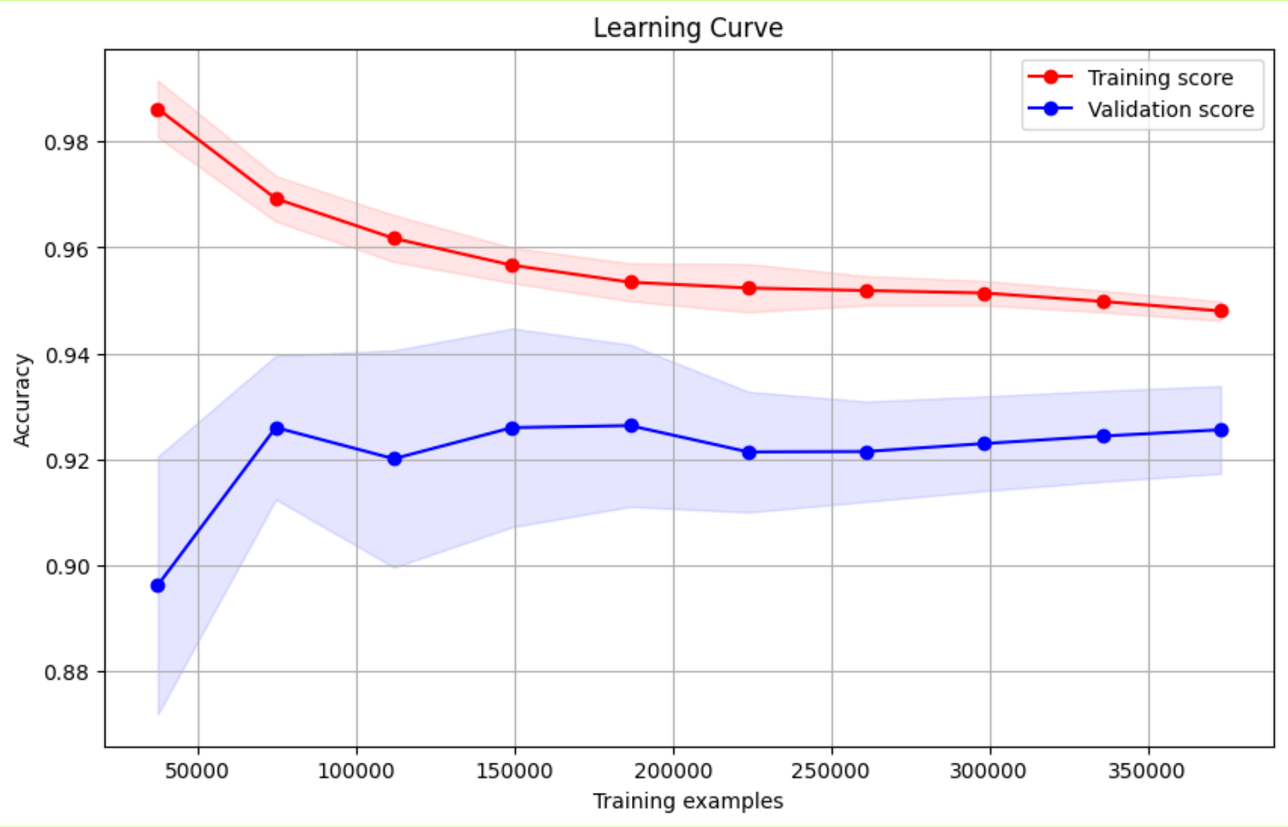
Model Improvement

As you can see, our model perform slightly better. But this did not meet up our expectation. So next we will examine the confusion matrix to better understand our model.



Model Evaluation

It seems there is a chance that our model is overfit or underfit. Let's check that with learning curve.

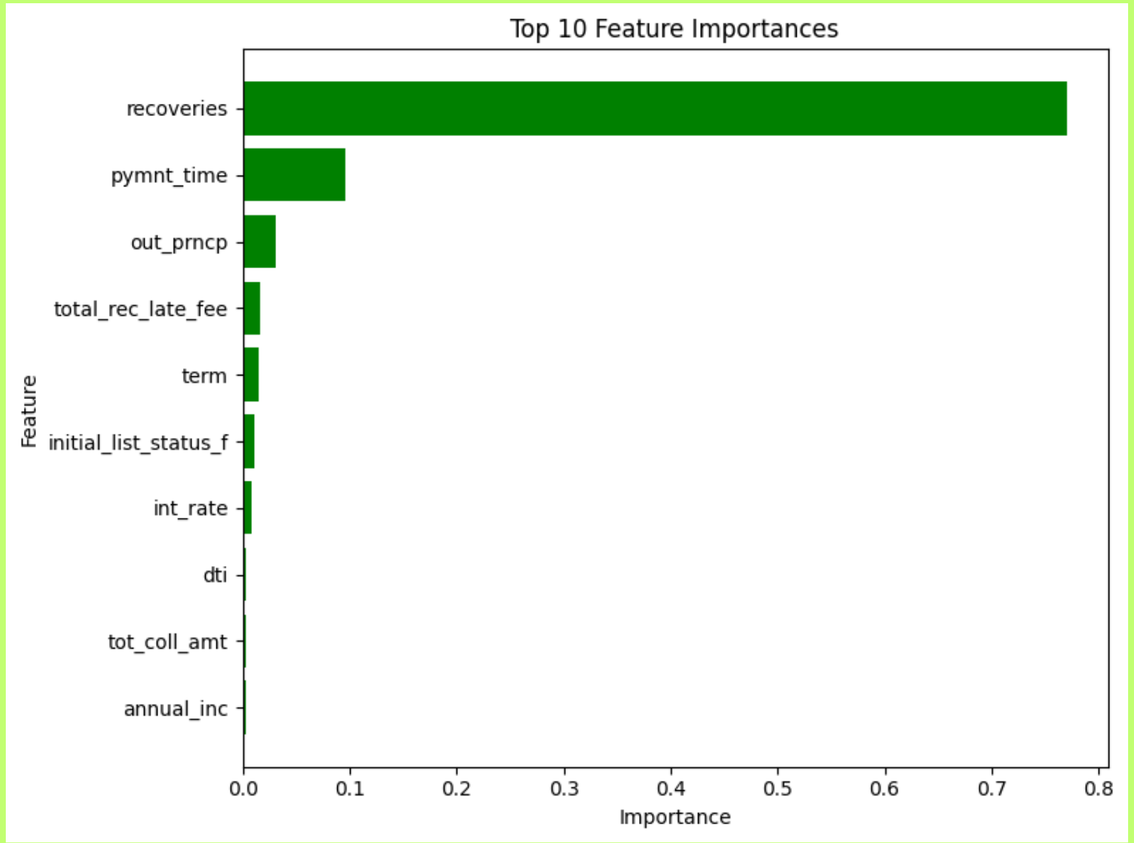


It turns out there is nothing wrong with our model. Next we will see the important features.

Model Evaluation

Our top 10 important features are mostly filled by numerical data. Our top important features is recoveries with 77% score. Then we will select our important features with more than 0.5% score, there are :

- recoveries (77% score)
- pymnt_time (9.63% score)
- out_prncp (3.09% score)
- total_rec_late_fee (1.66% score)
- term (1.45% score)
- initial_list_status_f (1.09% score)
- int_rate (0.8% score)



Model Retrain

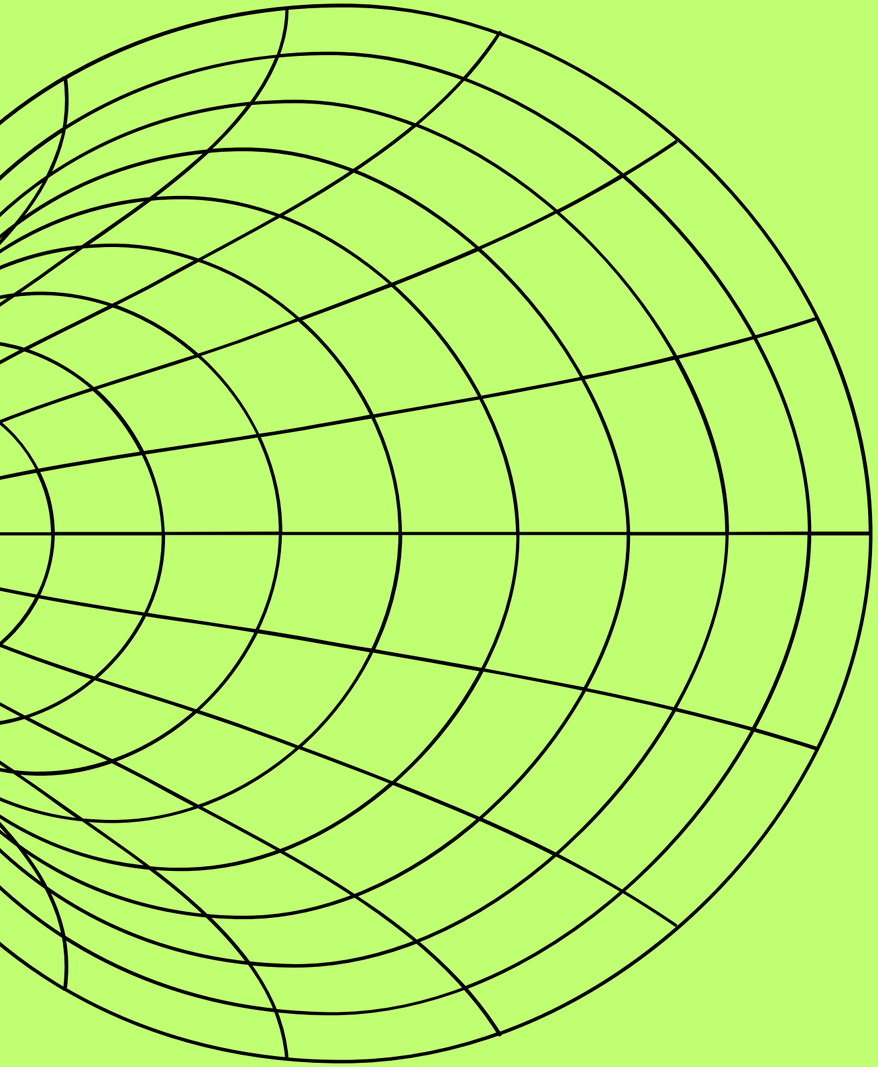
Then we try to retrain our model using only our selected important features. The results are:

Accuracy	Mislabeled	ROC	KS
95%	6423	0.93	0.6725

It turns out that the resulting difference is quite small, so we'll go with that.

So now our model will need less input which will lead to faster computing. With this, our model is finished, we just need to deploy our model to our web application.

Nurkahfi Amran Rahmada



Model Deployment

Preparation

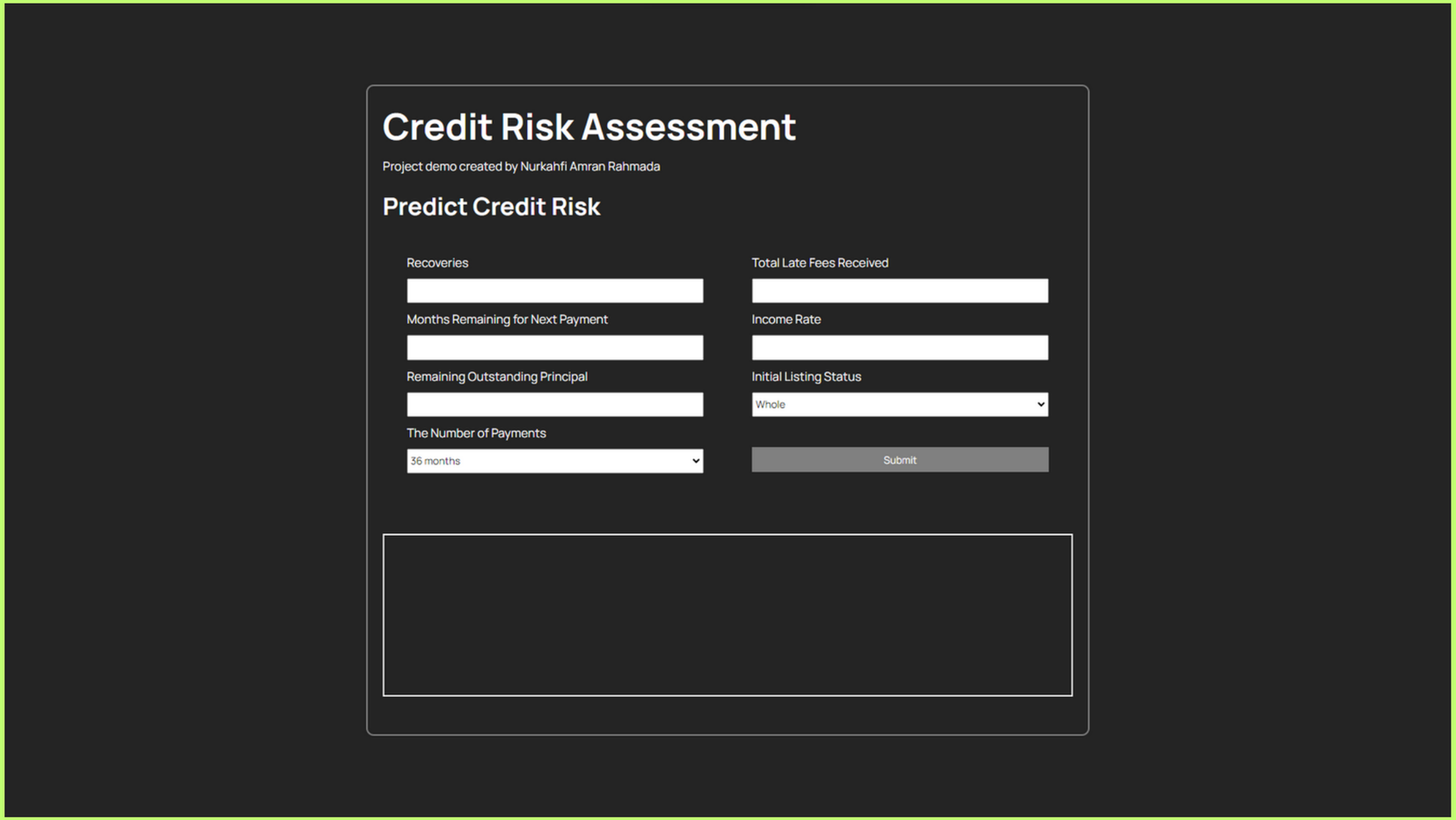
After our model is created, we need to deploy our model so that it can be used properly. We will deploy our model into web based application, so we need to make the web design first.

Then, we will use flask library to help connect python based file to html file. We also need flask to help jsonify our prediction output later.

With that, our project demo is finished, and our model has successfully deployed into our web application.

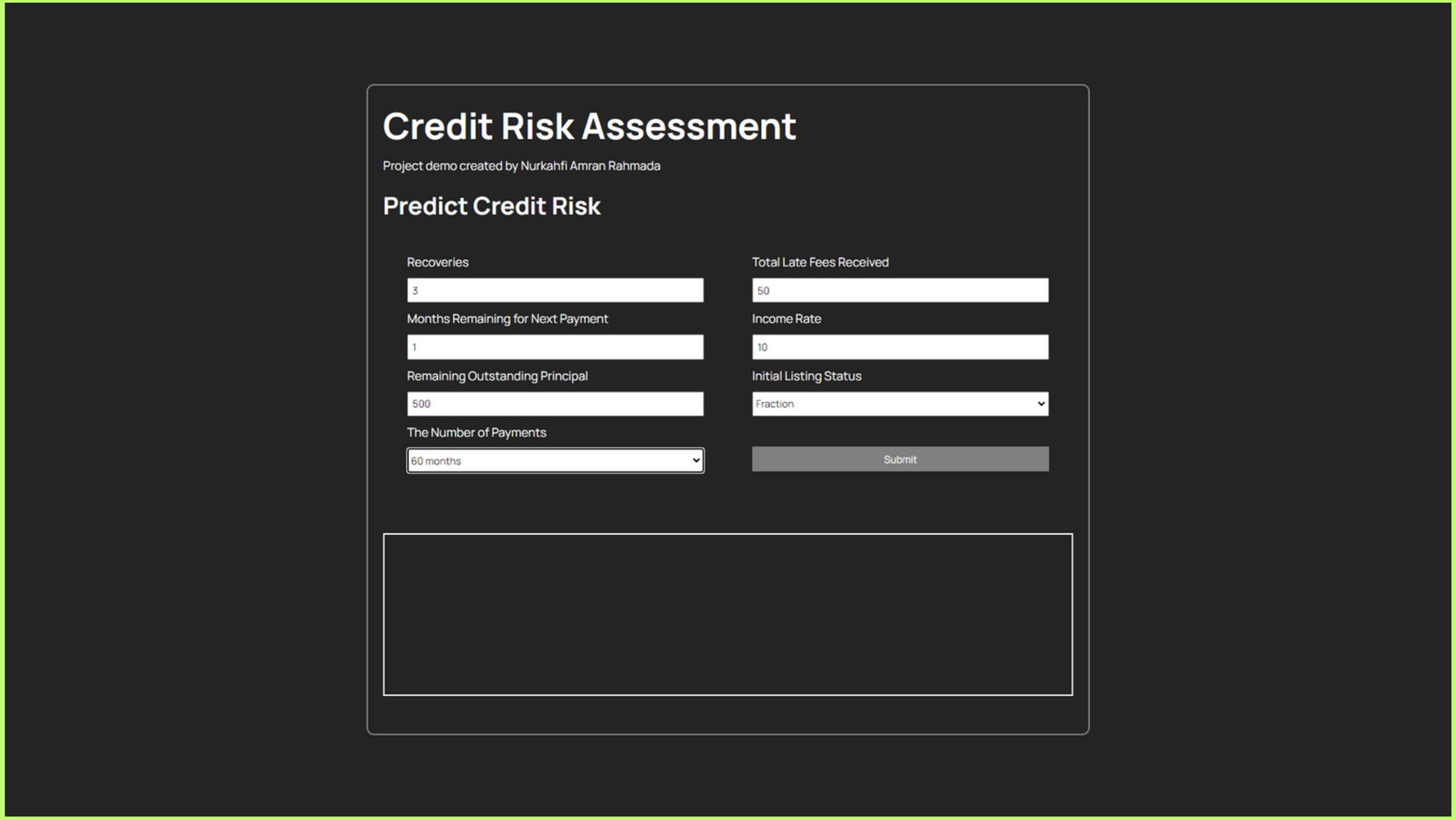
Project Demo

Web Interface :



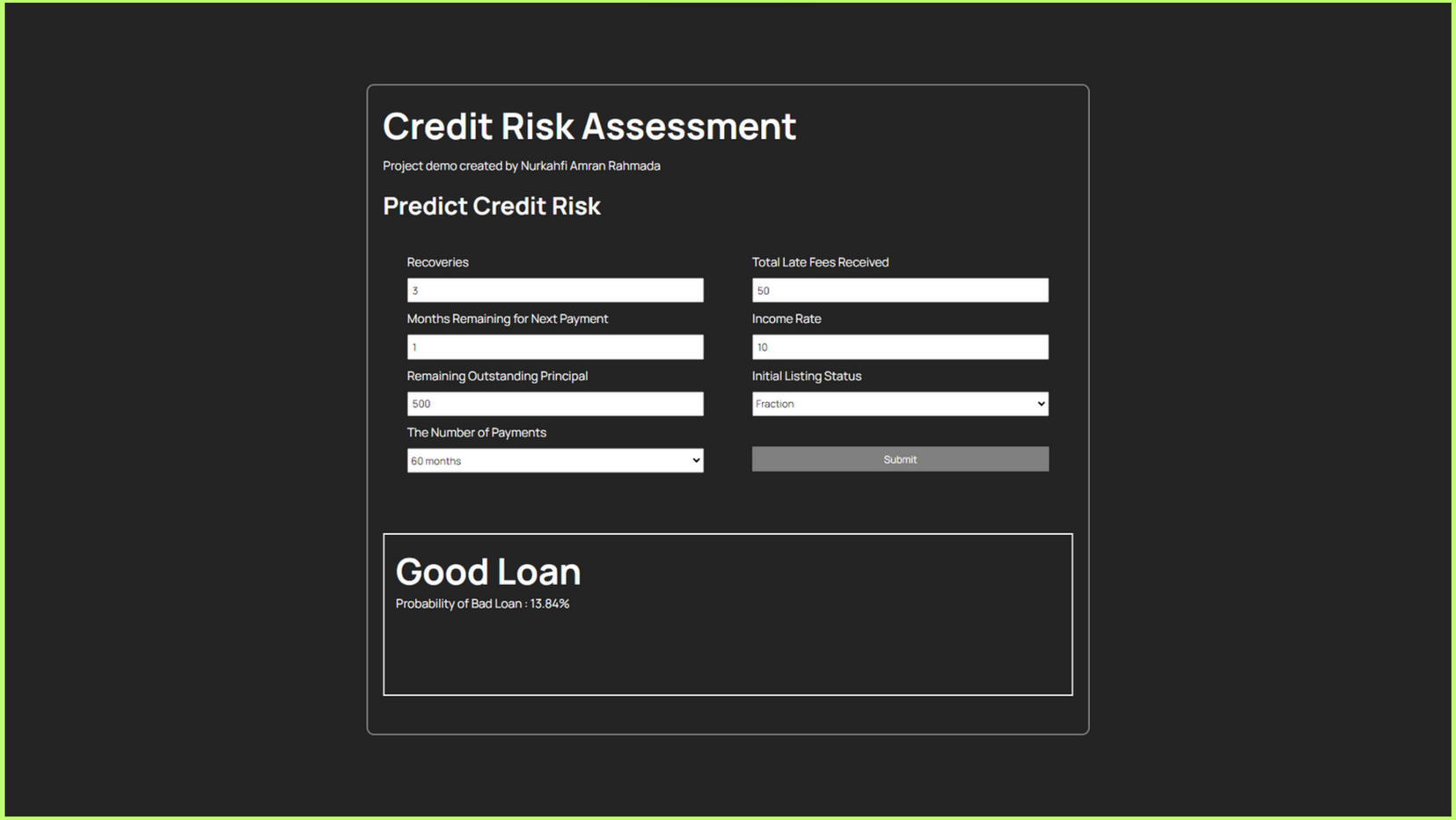
Project Demo

Web Interface (form filled) :

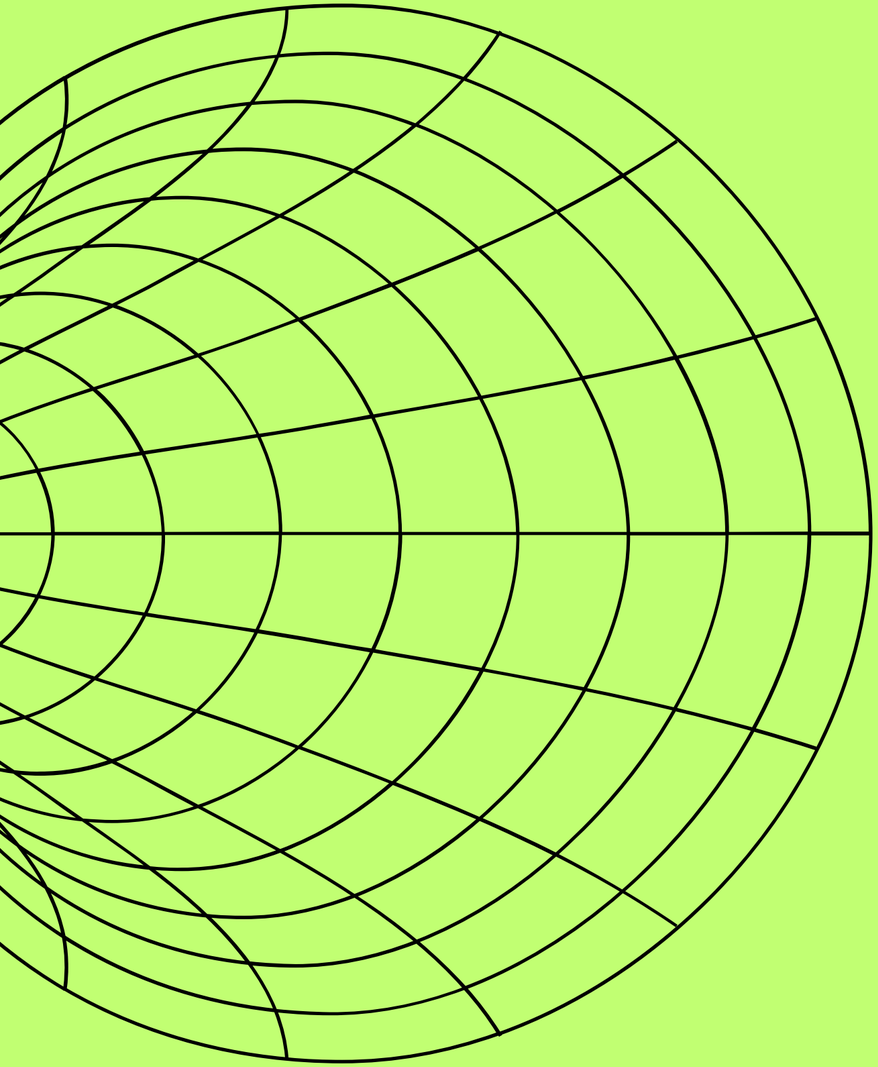


Project Demo

Web Interface (prediction result) :



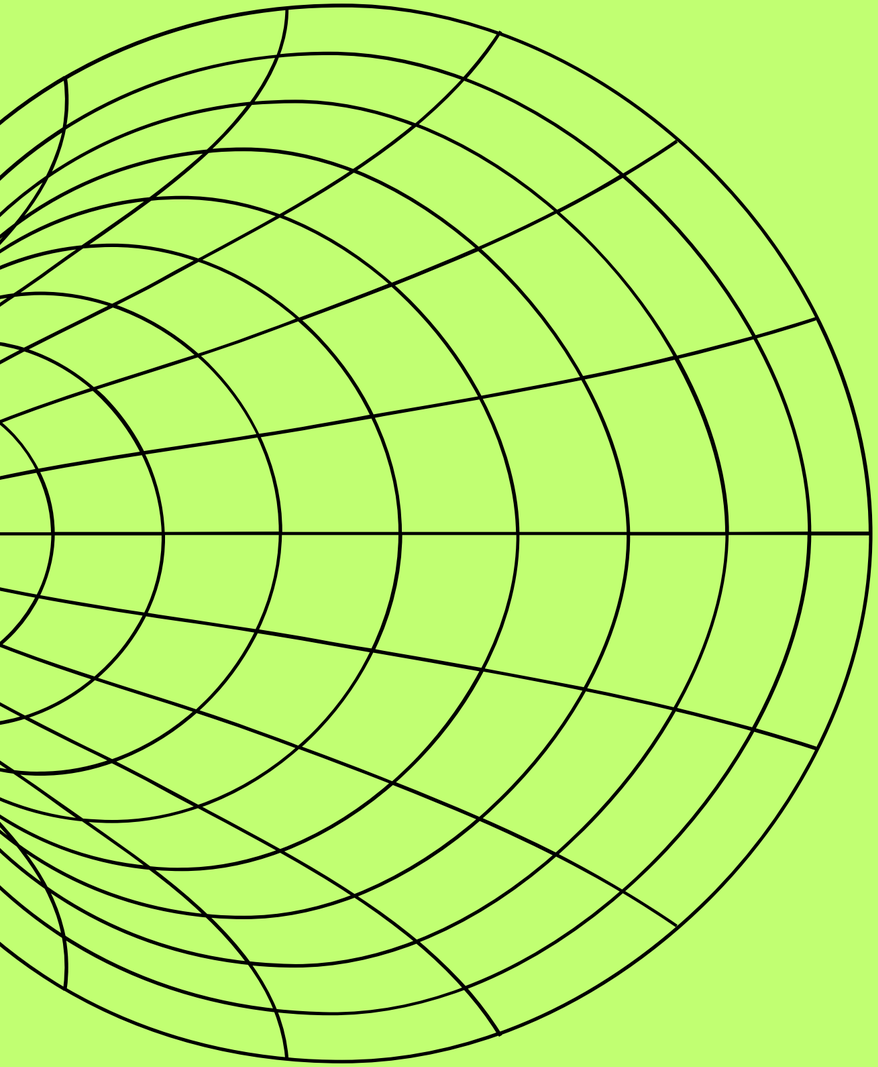
Nurkahfi Amran Rahmada



Business Recommendation

- Recoveries
  - Make sure the client payment plan has been put in place for the loan
- Payment time
  - Avoid giving clients a long period of time to pay their loans
- Remaining Outstanding Principal
  - Make sure that no client has a remaining outstanding principal that is greater than it should be
- Total Late Fees Received
  - Provide loans according to client capabilities to minimize clients receiving late fees
- Term
  - Provide number of payments according to the client's financial capabilities, provide payment amounts for 60 months to clients who really need it
- Initial List Status
  - Prioritize clients who choose fractional for their initial list status
- Income Rate
  - Make sure the client's income rate is verified and they are able to pay the loan

Nurkahfi Amran Rahmada



Evaluation

Suggestion

- More exploration and more understanding can be provided if there is more time to work this project on
- The model still have room for improvement by using ensemble method as comparison.
- The hyperparameter tuning should be done by detailed observations and lots of experiments
- Detailed business understanding should help understanding the data acquisition

Reference

Investopedia. "Credit Risk Definition." Investopedia, accessed January 29, 2024. https://www.investopedia.com/terms/c/creditrisk.asp#:~:text=Error%20Code%3A%20100013)-,What%20Is%20Credit%20Risk%3F,and%20increased%20costs%20for%20collection.

Rakamin Academy

id/x partners

Thank You!