Project Summary

Predictive Modeling for Loan Investment Strategies

1st May, 2024

Contributors: Kristen Dmello, Aditya Kolpe, Parth Lokhande
Advisor: Leman Akoglu

Data Provided By: LendingClub

Table of Contents

1. Executive Summary

This project leveraged LendingClub's historical loan data to develop predictive models aimed at optimizing investment strategies for loan defaults. Spanning three key phases—data preparation, model development, and strategy formulation—the initiative provided a comprehensive approach to understanding and capitalizing on peer-to-peer lending opportunities.

The project employed a robust data analysis regimen, including:

Phase I: Focused on initial data ingestion and exploratory analysis, setting the stage for deeper insights.

Phase II: Concentrated on data cleaning and preparation, ensuring the quality and reliability of the data for modeling.

Phase III: Developed and evaluated multiple predictive models, including Decision Trees, Random Forest, and Neural Networks, to forecast loan performance and guide investment decisions.

2. Problem Statement

With the rise of peer-to-peer lending platforms like LendingClub, individual investors have the opportunity to invest in personal loans in a way similar to purchasing stocks. LendingClub categorizes loans with a grading system reflecting the risk and return profile based on the borrower’s credit history. The platform offers loans ranging from $1,000 to $40,000 with terms of either 36 or 60 months and allows investments in small increments, called "notes."

As of 2023, LendingClub interest rates vary from 6.7% to 22.8%, influenced by loan duration and borrower ratings. Default rates span from 1.3% to 10.6%, indicating significant variability in risk. Such data highlights the critical need for robust decision-making tools to guide investment choices. Investors can choose from multiple loans, each divided into $25 notes. This structure enables diversification even with smaller investment amounts. The availability of detailed loan data online provides transparency, allowing investors to analyze potential investments deeply.

3. Business Impact

Data-driven investment strategies leverage historical data to predict future loan performance, optimizing return on investments through informed decision-making. By analyzing patterns in interest rates, default rates, and borrower creditworthiness, investors can identify loans with the best risk-return balance.

Our analysis aims to help with three critical aspects of the investment landscape:

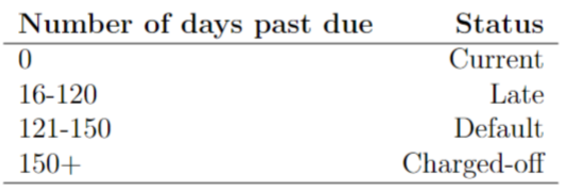
1. Risk Mitigation: Advanced analytics help in understanding potential default risks and economic trends, allowing investors to avoid high-risk loans.
2. Optimized Returns: Data-driven strategies aim to maximize returns by selecting loans that balance risk and reward effectively.
3. Strategic Diversification: Investors can use data to diversify their portfolios across various grades and terms, reducing potential losses.

4. Introduction to the Dataset

In this Study, we leveraged the historical data from loans that were issued on LendingClub between April 2008 and September 2019.

The dataset contains more than hundred features, including the following, for each loan:
1. Loan amount
2. Interest rate
3. Monthly installment amount
4. Several additional attributes about the borrower (type of house ownership, annual income, monthly FICO score, debt-to-income ratio, number of open credit lines, etc.)
5. Loan status (e.g., fully paid, default, charged-off)

Loan status is defined as summarized in Table 1.1. Current refers to a loan that is still being reimbursed in a timely manner. Late corresponds to a loan on which a payment is between 16 and 120 days overdue. If the payment is delayed by more than 121 days, the loan is considered to be in Default. If LendingClub has decided that the loan will not be paid off, then it is given the status of Charged-Off.



Number of days past due	Status
0	Current
16-120	Late
121-150	Default
150+	Charged-off

Table 1.1: Loan statuses in LendingClub

In short, each loan expires 5 months after the term of the loan has ended, and ends in one of two LendingClub states—fully paid or charged-off. In the former, the investor is reimbursed fully for the balance plus interest, and has earned a positive return. The latter, on the other hand, causes loss depending on how much of the loan was paid of before being charged-off.

5. Data Ingestion and Preliminary Analysis

1. We first removed all instances (in our case, rows in the data table) representing loans that are still current (i.e., that are not in status Fully Paid, Charged-Off, or Default), and all loans that were issued before January 1, 2009 since they would not be required for our analysis.
2. Treatment of outliers: We visualized each of the features to check if there are any outliers and removed those instances.
3. For the sake of this Study, we restricted to the following features: *id, loan amnt, funded amnt, term, int rate, grade, emp length, home ownership, annual inc, verification status, issue d, loan status, purpose, dti, delinq 2yrs, earliest cr line, open acc, pub rec, fico range high, fico range low, revol bal, revol util, total pymnt, recoveries*.
4. We saved the resulting data set in a Python “pickle”.

Calculating the Return Variable:
A key measure we needed in working out an investment strategy is the return on each loan, defaulted or otherwise. Calculating the return is complicated by two factors:

(1) the return should take into account defaulted loans, which usually are partially paid off, and
(2) the return should also take into account loans that have been paid early.

Ideally, we would want to take into account potential future reinvestments once a loan is repaid early however this complicates things quite a bit. Instead we introduced three different ways of calculating the return, as described below.

- Method 1 (denoted M1–Pessimistic) supposes that, once the loan is paid back, the investor is forced to sit with the money without reinvesting it anywhere else. This is the worst-case scenario. Under this assumption, the annualized return is calculated as

$$\frac{p-f}{f} \times \frac{12}{t}$$

where

• f is the total amount invested in the loan,
• p is the total amount repaid, and
• t is the term length of the loan in months.

The downside of this method is that the assumption is hardly realistic. On the other hand, it handles defaults gracefully, by spreading the resulting loss over the term of the loan, which is reasonable since the investor was initially intending for their investment to remain “locked up” for that duration.

- Method 2 (denoted M2–Optimistic) supposes that, once the loan is paid back, the investor’s money is returned and the investor can immediately invest in another loan with exactly the same return. In this case, the annualized return is calculated as

  $$\frac{p-f}{f} \times \frac{12}{m}$$

  where
  m is the actual length of the loan in months; i.e., the number of months from the date the loan was issued to the date the last payment was made.

  The assumption that the cash can be reinvested at the same rate may not be realistic.

  However, the main issue with M2 is that if a loan defaults early, annualizing the loss can result in a huge over-estimate of the negative return. For instance, if a loan defaults In the 1st month, the investor loses 100% of the investment. This is the maximum loss, but annualizing it would lead to a 1200% loss (!) In other words, we would be assuming the investor reinvests in an equally risky loan for the 11 remaining months of the year, each of which defaults in the 1st month! Hardly realistic. Instead, you will use the following two-piece formula:

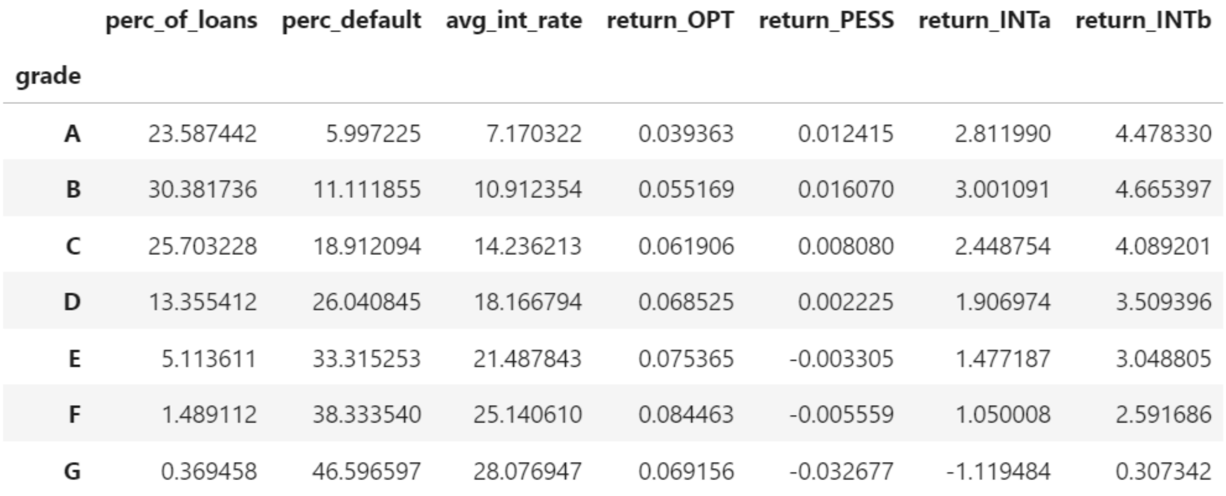
  $$\begin{cases} \frac{p-f}{f} \times \frac{12}{m} & \text{if} \quad p-f>0 \\ \frac{p-f}{f} \times \frac{12}{t} & \text{if} \quad p-f \leq 0 \end{cases}$$

- Method 3 (denoted M3) considers a fixed time horizon and assumes that any revenues paid out from a loan are immediately reinvested at a yearly rate of i%, compounded monthly, until the T-month horizon is over (in this Study, we will consider a 5-year horizon, i.e., T = 60). This method is closest to what would realistically happen.
  Assuming each monthly payment was of size p/m which then are immediately reinvested, we can use the sum of a geometric series to find the total return from the f initially invested:

$$\left\{\left[\frac{p}{m}\times\left(\frac{1-(1+i)^{m}}{1-(1+i)}\right)\right]\times(1+i)^{T-m}-f\right\}\times\frac{1}{f}\times\frac{12}{T}$$

We then populated the average percentage (annual) return per grade in the table below:

- return_OPT: This column represents the average percentage (annual) return assuming the optimal scenario.
- return_PESS: This column represents the average percentage (annual) return assuming the pessimistic scenario.
- return_INTa: This column represents the average percentage (annual) return using the first interest rate assumption (i = 1.4).
- return_INTb: This column represents the average percentage (annual) return using the second interest rate assumption (i = 2).



	perc_of_loans	perc_default	avg_int_rate	return_OPT	return_PESS	return_INTa	return_INTb
grade							
A	23.587442	5.997225	7.170322	0.039363	0.012415	2.811990	4.478330
B	30.381736	11.111855	10.912354	0.055169	0.016070	3.001091	4.665397
C	25.703228	18.912094	14.236213	0.061906	0.008080	2.448754	4.089201
D	13.355412	26.040845	18.166794	0.068525	0.002225	1.906974	3.509396
E	5.113611	33.315253	21.487843	0.075365	-0.003305	1.477187	3.048805
F	1.489112	38.333540	25.140610	0.084463	-0.005559	1.050008	2.591686
G	0.369458	46.596597	28.076947	0.069156	-0.032677	-1.119484	0.307342

Decision Analysis:

- For Conservative Investors: Grade A is ideal due to its low default rate and consistent positive returns across different scenarios. It offers lower returns but significantly less risk.

- For Moderate Investors: Grade B and C might be appealing as they offer higher returns than Grade A (around 10.91% and 14.24% interest rates, respectively) with manageable default rates (11.12% and 18.92%).

- For Aggressive Investors: Grade E and F offer high returns (21.49% and 25.14% interest rates) but also carry high risks (default rates of 33.31% and 38.33%). These grades could yield higher profits but also come with a substantial risk of loss.

Grade A seemed the most prudent choice for most investors. It provides a reasonable return with the lowest risk of default, making it a safer bet in a portfolio, especially for those who prefer a conservative investment strategy.

6. Modeling Approaches

6.1. Predictive Models of Default:

Before training the model we split the dataset into train and test, performed feature engineering and used prepare_data to perform feature selection based on specified feature subsets.

6.1.1 Model Training and Evaluation:

- **Model Setup**: 7 models including Decision Tree (DT), Random Forest (RF), (L1 and L2
- regularized) Logistic Regression (LogR), Naıve Bayes (NB), and (multi-layer) Neural Network (NN) are initialized.
- **Cross-Validation Setup:** For model tuning, GridSearchCV is used to find the optimal model parameters based on specified cross-validation parameters.
- **Model Fitting:** Models are fitted on the training data.
- **Model Evaluation:** Models are evaluated using the performance metrics below:
  - Accuracy, Precision, Recall, F1 Score
  - ROC-AUC
  - Brier Score
  - Calibration Curves
  - Similarity to known rankings (e.g., comparing to a loan grade)

  These are visualized as ROC curves, calibration plots, and sensitivity/specificity curves to visually assess model performance.

6.1.2 Hyper-parameter Tuning:

Hyperparameters provided at the time of model tuning:

● Naive Bayes: None

● L1 regularized logistic regression: solver = ‘liblinear’, cv_parameters = {'C': [0.001, 0.01, 0.1, 1, 10, 100]}

● L2 regularized logistic regression: solver= 'lbfgs', cv_parameters = {'C': [0.001, 0.01, 0.1, 1, 10, 100]}

● Decision tree: 'max_depth': [3, 5, 10, 15], 'min_samples_split': [2, 5, 10]

● Random forest: 'n_estimators': [100, 200], 'max_depth': [5, 10, 15]

● Multi-layer perceptron: 'hidden_layer_sizes': [(100,), (50,), (50, 25)], 'activation': ['logistic', 'relu'], 'solver': ['adam'], 'alpha': [0.0001, 0.01]

● Logistic Regression: 'C': [0.001, 0.01, 0.1, 1, 10, 100], solver='liblinear', max_iter=1000

Optimal Parameters provided at the time of tuning:

● Naive Bayes: None {}

● L1 regularized logistic regression: {'C': 0.001}

● L2 regularized logistic regression: {'C': 0.001}

● Decision tree: {'max_depth': 3, 'min_samples_split': 2}

● Random forest: {'max_depth': 15, 'n_estimators': 100}

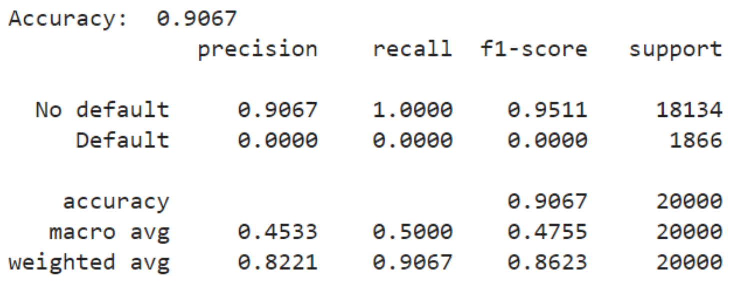
● Multi-layer perceptron: 'hidden_layer_sizes': {'activation': 'logistic', 'alpha': 0.01, 'hidden_layer_sizes': (50,), 'solver': 'adam'}

● Logistic Regression: {'C': 0.001}

6.1.3 Performance Measures used across all models:

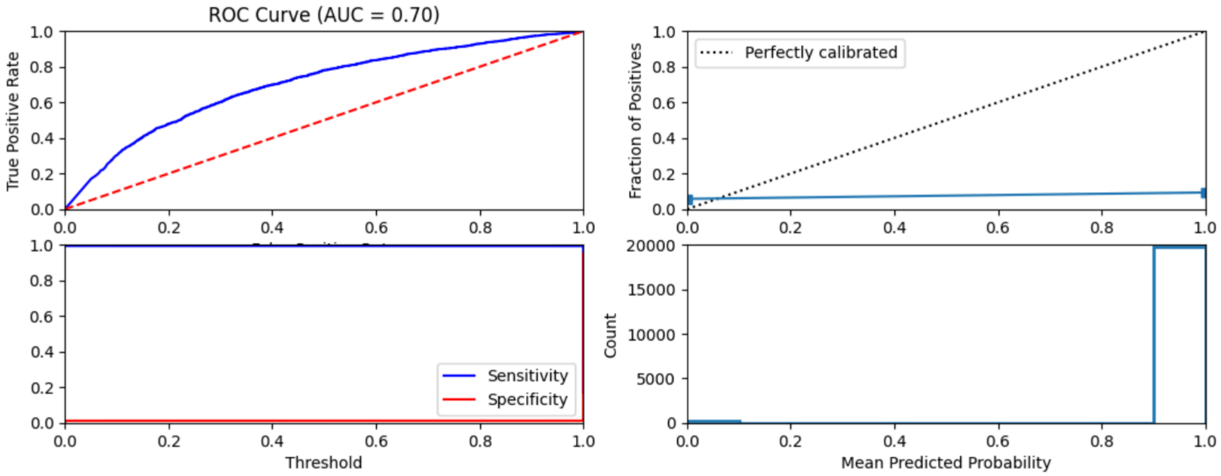
- Accuracy Diagnostics (Accuracy, Precision, Recall, F1 Score)
- ROC Curve
- Threshold(Sensitivity/specificity curve)
- Calibration Curve
- Mean Predicted Probability

Model Name: Naive Bayes

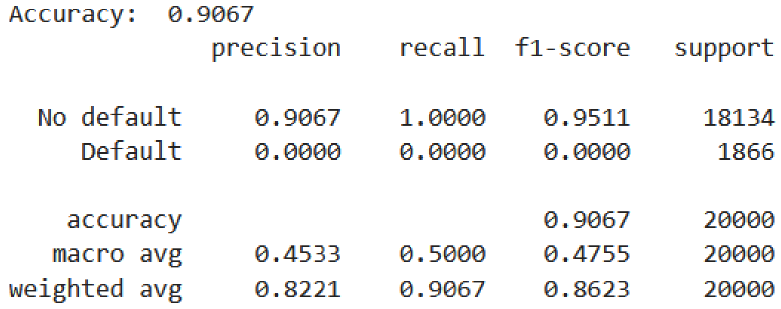


```
Accuracy:  0.9067
              precision    recall  f1-score   support

  No default     0.9067    1.0000    0.9511     18134
     Default     0.0000    0.0000    0.0000      1866

    accuracy                         0.9067     20000
   macro avg     0.4533    0.5000    0.4755     20000
weighted avg     0.8221    0.9067    0.8623     20000
```

Evaluation Result:

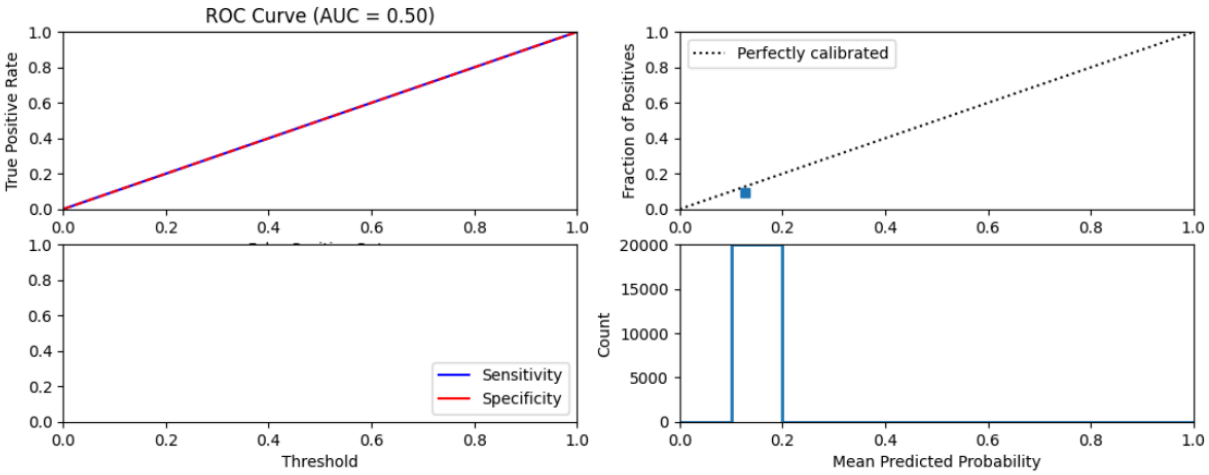


Model Name: L1 regularized logistic regression

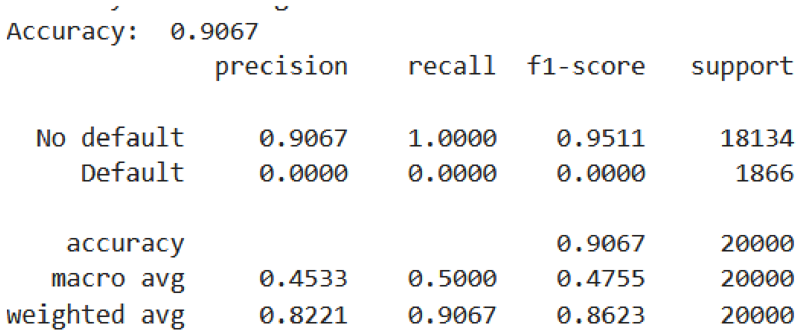


```
Accuracy:  0.9067
              precision    recall  f1-score   support

  No default     0.9067    1.0000    0.9511     18134
     Default     0.0000    0.0000    0.0000      1866

    accuracy                         0.9067     20000
   macro avg     0.4533    0.5000    0.4755     20000
weighted avg     0.8221    0.9067    0.8623     20000
```

Evaluation Result:

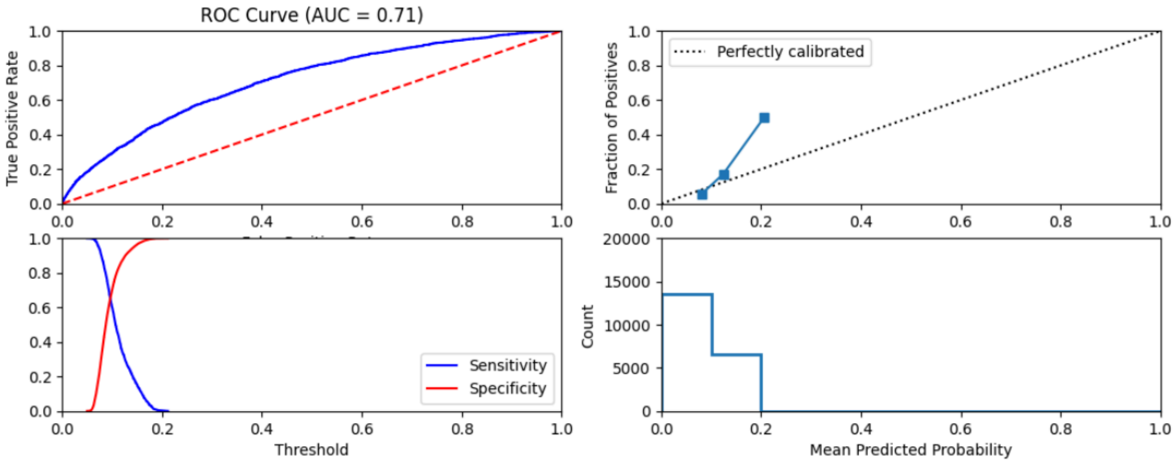


Model Name: L2 regularized logistic regression

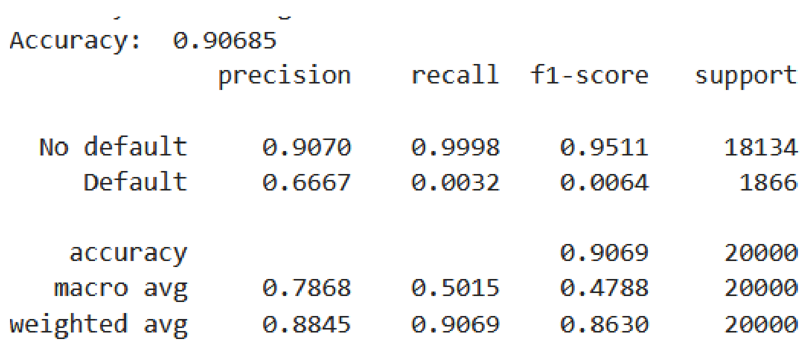


```
Accuracy:  0.9067
              precision    recall  f1-score   support

  No default     0.9067    1.0000    0.9511     18134
     Default     0.0000    0.0000    0.0000      1866

    accuracy                         0.9067     20000
   macro avg     0.4533    0.5000    0.4755     20000
weighted avg     0.8221    0.9067    0.8623     20000
```

Evaluation Result:

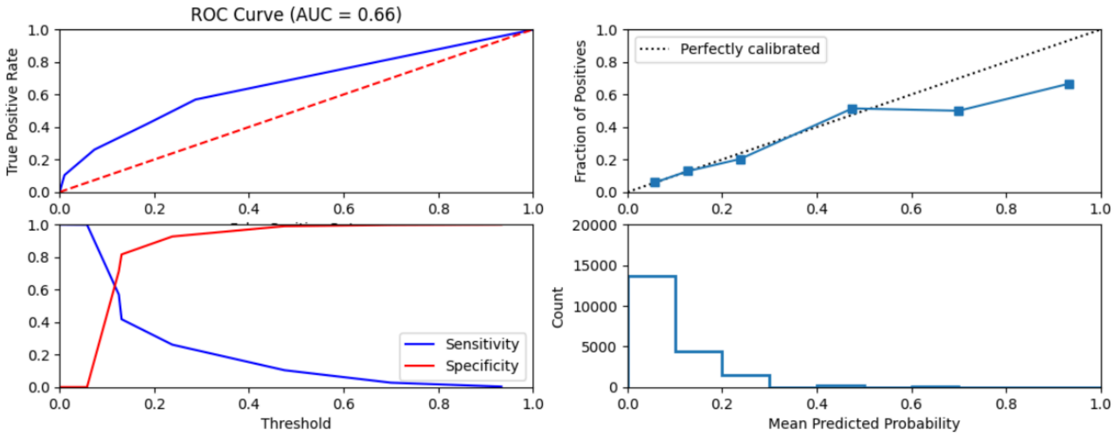


Model Name: Decision tree

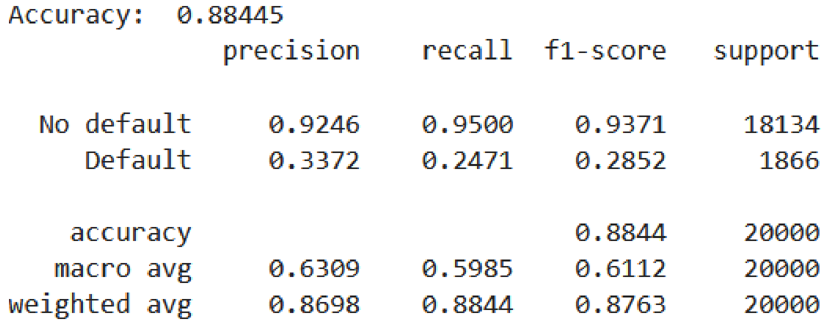


```
Accuracy:  0.90685
               precision    recall  f1-score   support

  No default      0.9070    0.9998    0.9511     18134
     Default      0.6667    0.0032    0.0064      1866

    accuracy                          0.9069     20000
   macro avg      0.7868    0.5015    0.4788     20000
weighted avg      0.8845    0.9069    0.8630     20000
```

Evaluation Result:

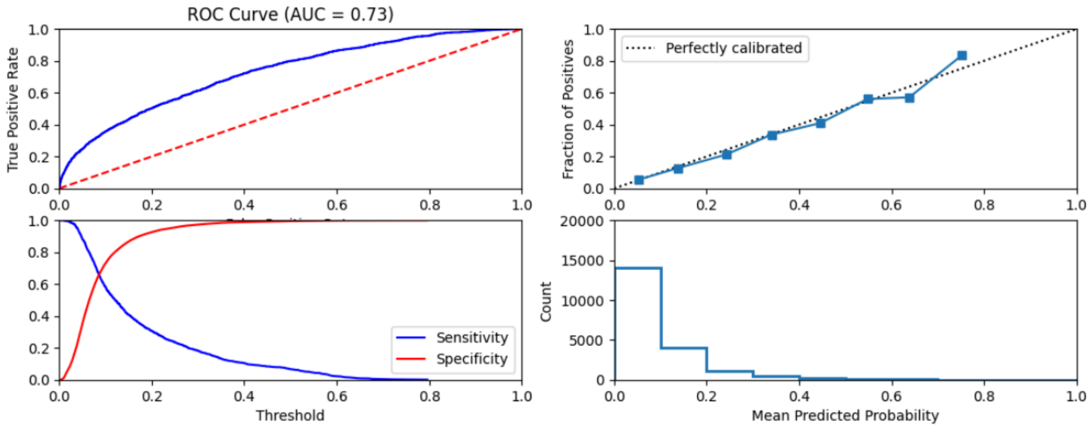


Model Name: Random forest



```
Accuracy:  0.88445
               precision    recall  f1-score   support

  No default      0.9246    0.9500    0.9371     18134
     Default      0.3372    0.2471    0.2852      1866

    accuracy                          0.8844     20000
   macro avg      0.6309    0.5985    0.6112     20000
weighted avg      0.8698    0.8844    0.8763     20000
```

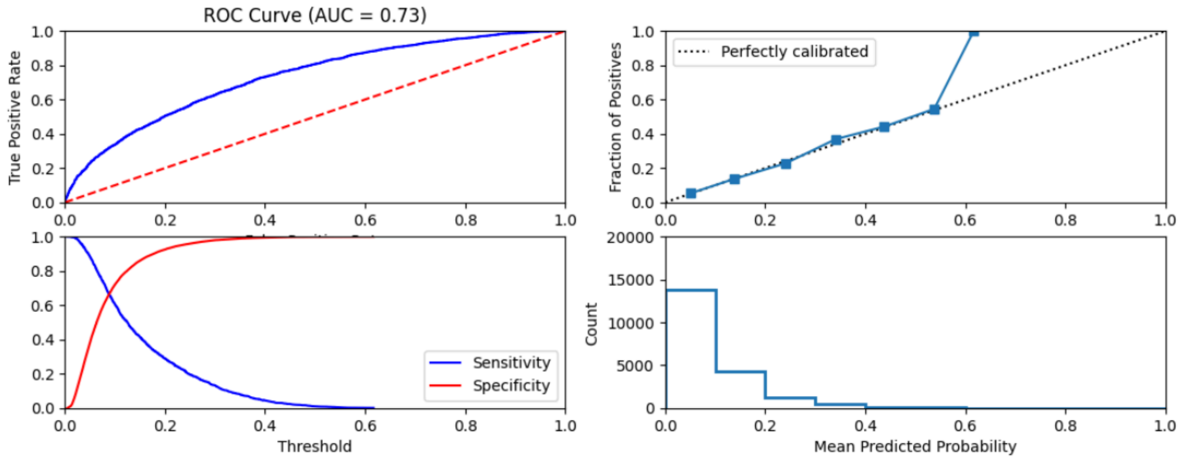
Evaluation Result:



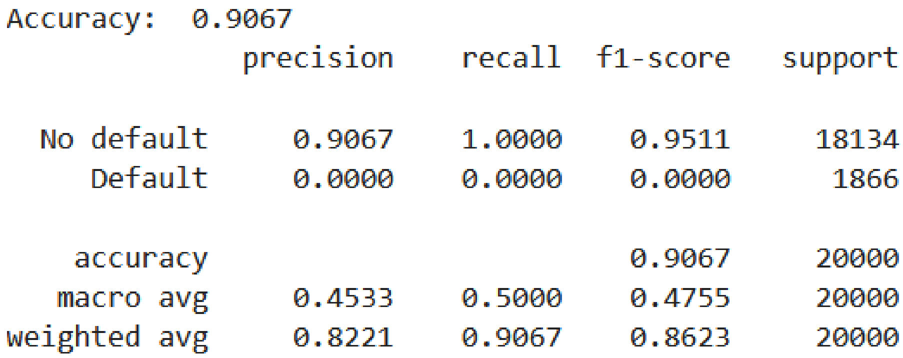
Model Name: Multi-layer perceptron

```
Accuracy:  0.906
              precision    recall  f1-score   support

  No default     0.9094    0.9955    0.9505     18134
     Default     0.4527    0.0359    0.0665      1866

    accuracy                         0.9060     20000
   macro avg     0.6810    0.5157    0.5085     20000
weighted avg     0.8668    0.9060    0.8680     20000
```

Evaluation Result:

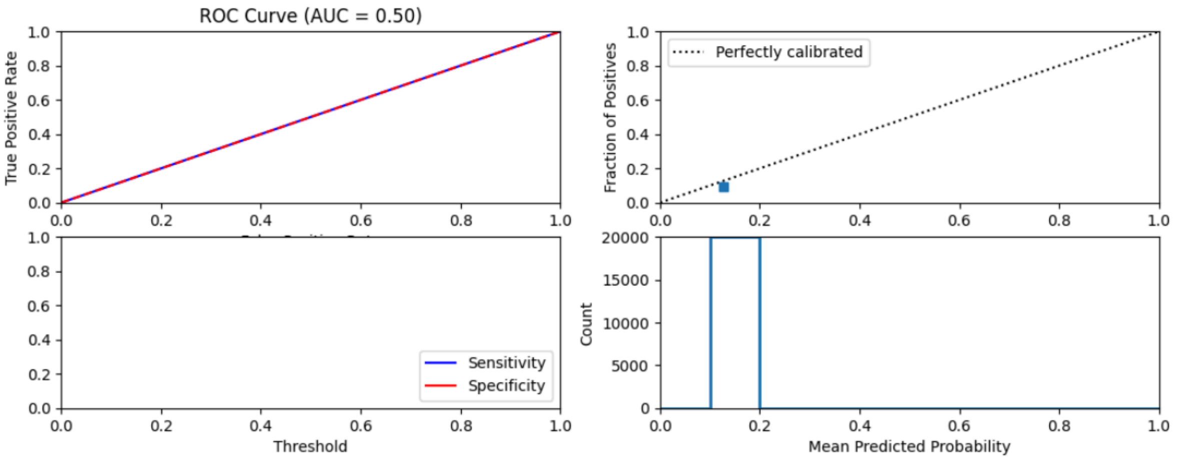


Model Name: Logistic Regression



```
Accuracy:  0.9067
              precision    recall  f1-score   support

  No default     0.9067    1.0000    0.9511     18134
     Default     0.0000    0.0000    0.0000      1866

    accuracy                         0.9067     20000
   macro avg     0.4533    0.5000    0.4755     20000
weighted avg     0.8221    0.9067    0.8623     20000
```

Evaluation Result:



Based on the recall values for both default and the non default class and the AUC of the ROC curve, we picked the **Multi Layer Perceptron** as our best model.

We then remove certain features:

- Grade
- sub grade
- home ownership
- verification status
- loan status

- Dti
- fico range high
- fico range low
- last fico range high
- last fico range low

For training the model 100 times we considered 4 features. ‘Loan_amt’, ‘Funded_amt’, ‘int_rate’, ‘annual_inc’

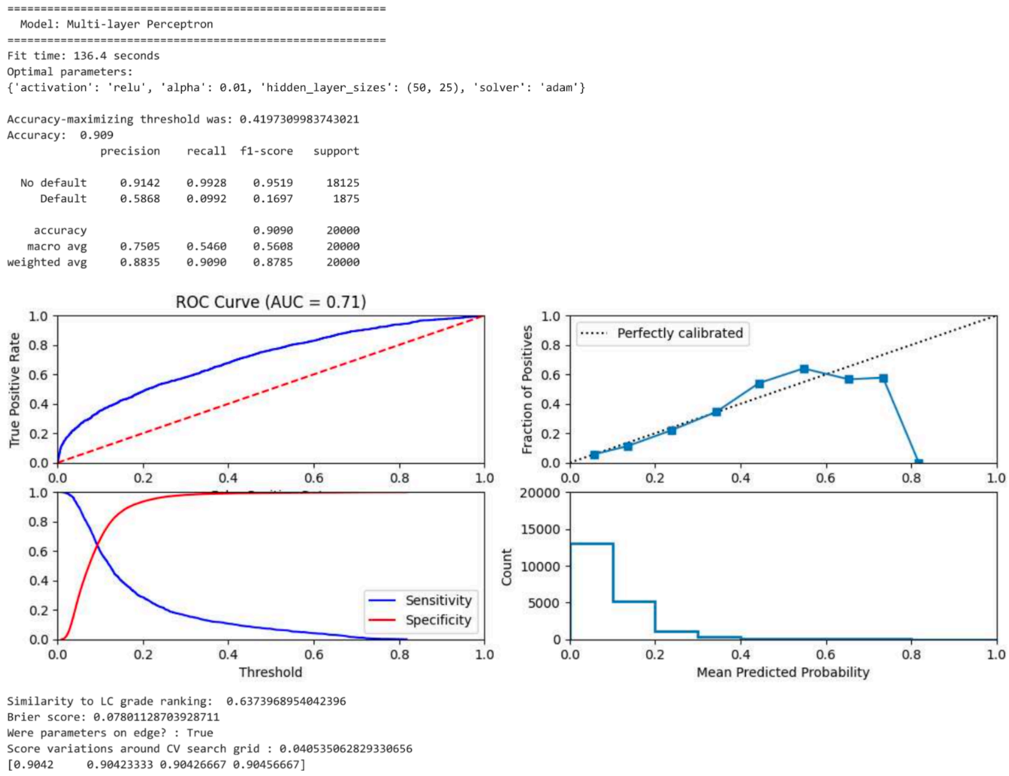
We also trained the model based on all features ignoring the above mentioned features.

We would then use ‘Your Model’ the ‘MLP’ to invoke the function ‘predict_proba’ on our model to get the likelihood scores for loans

We then compare the similarity of the grades from LC and the ‘scores’ from our model

If we ideally had the rubric through which LC gives these grades, we could assign grades to loans based on the scores from our model and then compare them

Since the above method is not possible, we use a correlation statistic like kendall’s TAU. It measures the ordinal association between two measured quantities



```
==========================================================
  Model: Multi-layer Perceptron
==========================================================
Fit time: 136.4 seconds
Optimal parameters:
{'activation': 'relu', 'alpha': 0.01, 'hidden_layer_sizes': (50, 25), 'solver': 'adam'}

Accuracy-maximizing threshold was: 0.4197309983743021
Accuracy:  0.909
              precision    recall  f1-score   support

  No default     0.9142    0.9928    0.9519     18125
     Default     0.5868    0.0992    0.1697      1875

    accuracy                         0.9090     20000
   macro avg     0.7505    0.5460    0.5608     20000
weighted avg     0.8835    0.9090    0.8785     20000
```

```
Similarity to LC grade ranking:  0.6373968954042396
Brier score: 0.07801128703928711
Were parameters on edge? : True
Score variations around CV search grid : 0.040535062829330656
[0.9042     0.90423333 0.90426667 0.90456667]
```

Looking at the similarity to LC grade : there is a 0.63 which indicates a good positive correlation between our model and the LC grades. The correlation is not very high indicating that the LC grades might only be directional.

6.2. Investment Strategies:

6.2.1 Reporting the Results

First, we built the three regression models described above: (1) regressing against all returns, (2) regressing against returns for defaulted loans, and (3) regressing against returns for nondefaulted loans.

In each case, we used each one of the four return variables calculated in Phase II as our target variable (recall M1, M2, M3(1.4%), and M3(2%)) and tried (L1 and L2 Regularized) linear regression, random forest regression, and multi-layer NN regression.

Reporting the performance results in corresponding entries:

	Performance for each return calculation			
Model	M1	M2	M3 (1.4%)	M3 (2%)
L1 regressor	0.0398	0.0370	0.0417	0.0444
L2 regressor	0.0710	0.0568	0.0790	0.0782
Neural Network regressor	0.0932	0.0854	0.1377	0.1370
Random Forest regressor	0.1378	0.1158	0.1599	0.1586

We can tell that Random Forest has performed significantly better than the other models for all the different return calculations.

6.2.2 Reporting the Results

Additionally, suppose we were to invest in 1000 loans using each of the four strategies, we wanted to understand what the returns would be? To do this we averaged our results over 100 independent train/test splits.

	Return calculation			
Strategy	M1	M2	M3 (1.4%)	M3 (2%)
Rand	-0.0591	0.0088	-2.0277	-0.6352
Def	0.0156	0.0562	2.5364	4.1706

Ret	0.0457	0.0542	2.5640	4.2159
DefRet	0.0394	0.0563	2.6022	4.2427
BEST	**0.0457**	**0.0563**	**2.6022**	**4.2427**

Here the best possible solution (denoted Best) corresponds to the top 1000 performing loans in hindsight, that is, the best 1000 loans we could have picked.

Interpretation:

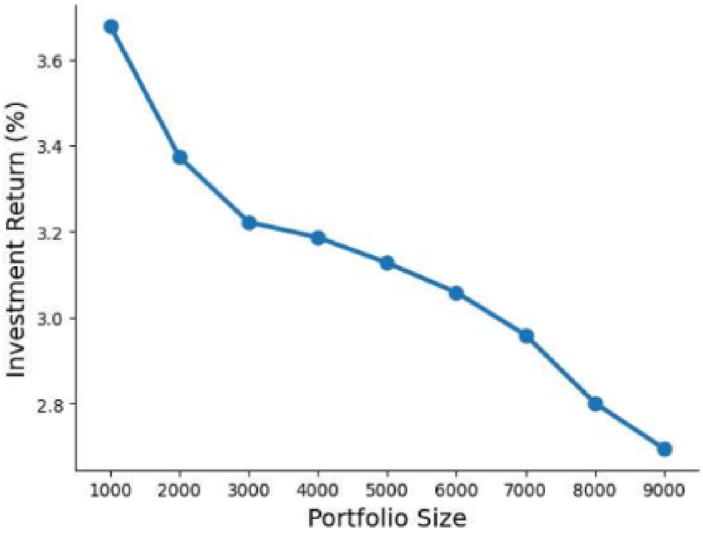
The DefRet model performs best, aligning with expectations. It integrates risk management (through default predictions) and profit maximization (through return predictions), offering a balanced and informed investment strategy.

Using the random strategy results in negative returns or losses in 3 out of 4 cases. This is because random has a high risk of negative returns as this method does not discriminate between high and low-risk loans. It is hence expected to perform worse than strategies that use historical data to make informed decisions.

The other two strategies (Def and Ret) outperform the Random strategy consistently, as they make informed decisions based on historical data and predictive modeling. This approach reduces the risk and increases the potential for higher returns by avoiding loans that are likely to default and selecting those with higher returns. DefRet should theoretically provide higher returns because it not only avoids high-risk loans (like Def) but also selects among the low-risk loans with the highest potential returns. It is also likely to be safer and more stable. While Ret might occasionally achieve higher peak returns by taking on high-risk, high-return loans, DefRet mitigates the risk by assessing which high-return loans have an acceptably low risk of default.

6.2.3 Testing Hypothesis: Increasing the number of loans invested in, would eventually result in lack of good loans to invest in

On plotting the return (using the M 1 return calculation, averaged over 100 runs) versus our portfolio size (i.e., number of loans invested in) we get the below graph:



The plot shows a downward trend in investment returns as the portfolio size increases, which is consistent with the hypothesis that as you expand the number of loans in your portfolio, the average loan quality decreases. This trend reflects a diminishing returns effect. Initially, when selecting the top-ranked loans according to the DefRet strategy, it's picking the loans with the highest expected returns given their risk profile. As you increase the number of loans in your portfolio to invest in, you include loans with progressively lower expected returns and potentially higher risk which decreases the return. The initial steep decline suggests that there may be a saturation point beyond which the average quality of available loans drops more significantly. This could represent a threshold beyond which it becomes increasingly difficult to find loans that meet the stringent criteria for a good investment.

7. Conclusion and Future Work

Conclusions:

- The use of advanced machine learning models provided a substantial edge in predicting loan defaults and optimizing investment strategies.
- Removing LendingClub's derived features in some tests helped in understanding the intrinsic predictive power of raw data features, enhancing the robustness of the predictive models.

- Data-driven investment strategies, particularly those integrating risk and return predictions, significantly enhanced potential returns compared to traditional random or uninformed strategies.

Future Recommendations:

- Continue refining models with newer data and external economic indicators to improve predictive accuracy.
- Explore additional machine learning techniques and feature engineering to further enhance model performance and investment outcomes.