Final Report

Team #: 37

Team Members:
Bo Bi (bbi6)
Xinying Lucy Lu (Xlu340)
Wen Gu (wguu)
Zhen (Jenny) Wang (zwang3384)

Table of Content

Project Overview

Background

Today predictive modeling is one of the primary applications in activating automation and precise decision making across industries and sectors. It is time to rethink how the banking industry can utilize data and patterns to decrease cost, increase accuracy and provide tools that help make the right decision precisely and quickly. Loans are one of the core businesses of banks. The main profit comes directly from the loan's interest. Banks grant a loan after an intensive process of verification and validation. For a long period of time, the process of loan evaluation and approval has been a manual task conducted by a loan officer or a representative from the bank. To approve a loan application, the approver is responsible for determining if an application is at high risk (high probability of default) or low risk based on limited information at hand. This manual decision-making process not only is cost-ineffective, but also leaves room for error, bias and subjectivity. Most importantly, the manual process does not provide assurance if the applicant is able to repay the loan leaving unpredictable financial risks to the banks. The percentage of outstanding loans left as unpaid after a prolonged period of missed payment is known as the default rate. To maximize profits for a bank, we want to reduce as much default rate as possible while maintaining a low operational cost. In this project, we aim to examine the factors that correlated with this default probability for small business loans granted by banks and guaranteed by the Small Business Association (SBA) in the United States. We then explore various predictive models to find out the best model that can indicate if a loan should be approved or not with a high accuracy rate that in turn helps reduce default rate for the bank.

Problem Statement

Manual loan approval process is prone to error, bias and subjectivity leaving unpredictable financial risks to the banks. We need a prediction model that can help banks make faster and more accurate loan approval decisions to automate loan approving processes and in turn maximizing profits through reducing loan default and operational cost.

Hypothesis

There are 3 primary areas that we hypothesize to be correlated with the small business loan default rate, macroeconomic factors, loan application attributes and information about the small business. We reviewed existing academic research to help us identify potential factors correlated with small business loan approval to establish the initial hypothesis. Through the report written by Roijmans[4], where he suggested that machine learning algorithms are capable of leveraging macroeconomic features to improve overall classification performance, we considered including macroeconomic data for our model.

Overview of Data

Based on our research above, we selected four data sets to assess the small business loan default rate and its correlation with macroeconomic factors, loan application attributes and information about the small business. There are a total 34 columns across four datasets. Out of these 34 columns, we selected 16 meaningful, relevant variables for initial data exploration. For example, we removed the variable “Bank Name” due to its irrelevance to the research and we also removed the “Zipcode” column due to its redundancy as the variable “State” provides similar information. Through the initial review of the dataset, we conducted feature engineering as well as identified variables that require further transformation. Before feature selection, we also conducted exploratory data analysis to supplement final features for modeling.

Datasets

Source	**Definition**
SBA Loan Historical Data	This file contains the Small Business Administration’s loan record from 1961-2014 (Total 899,164 observations). Each data point represents a loan that a bank granted to a small business, and SBA guaranteed a portion of the loan.
Monthly US unemployment rate	US unemployment rate from 1964 to 2014
Brave-Butters-Kelley Indexes (BBKI)	The BBKI index is used to estimate the monthly GDP growth rate of the US.***[6]***
Federal Fund Effective Rate	FFER can be used to estimate the cost of borrowing money

Data Cleaning and Feature Engineering

SBA loan data[5]: There are 27 columns in the file. Some of the transformations we did are 1) converted categorical data into integers and engineered several new features 2) created a loan period based on the loan disbursement date and term. Then we combined the macroeconomic data with the SBA loan data. (See **Appendix B** for full transformation details**)**

Brave-Butters-Kelley Indexes (BBKI): A feature called “Recession” is engineered based on BBKI index and loan period. In the time period between loan disbursement date -30 days to end date+30 days, if BBKI<0 for more than 6 months, Recession=1. Otherwise Recession=0. The reason we used 6-month is according to an investopedia article: “A popular rule of thumb is that two consecutive quarters of decline in gross domestic product (GDP) constitute a recession.”***[2]***

Monthly US unemployment rate since 1948[8]: We extracted the US unemployment rate from 1948 to 2022. A feature called “HighUnemployment " is engineered based on unemployment rate data and loan period. In the time period between loan disbursement date -30 days to end date+30 days, if unemployment>6% for more than 6 months period, HighUnemployment=1. Otherwise

HighUnemployment=0. The reason why we picked 6% as the threshold is according to moneychimp website: “An unemployment rate of about 4% - 6% is considered"healthy`."[1] We used a 6 month threshold similar to the recession definition above.

<u>Federal Fund Effective Rate(FFER) [7]:</u> We extracted the data from 1954 to 2022. A feature called “HighInterestRate " was engineered based on FFER data and loan period. In the time period between loan disbursement date -30 days to end date+30 days, if FFER>8% for more than 6 months, HighInterestRate=1. Otherwise HighInterestRate=0. The reason we picked 8% as the value threshold is because the interest rate above it will be on the top 17% of all the monthly interest rates in the data. The high interest rate could cause the companies unable to afford the loan or get additional loans to survive.

Exploratory Data Analysis

<u>Identify Multicollinearity:</u> Before feature selection, we explored the variables to better understand the independent variables to see if multicollinearity exists. Multicollinearity occurs when independent variables in a regression model are correlated. Features with high correlation are more linearly dependent and hence have almost the same effect on the dependent variable. If the degree of correlation between independent variables is high, this correlation could become a problem when we fit the models and interpret the results.When two features have high correlation, we will drop one of the two features in the feature selection process.We first used heatmap to visualize high correlation between variables, then we used VIF to measure the strength of the correlations. VIF estimates how much the variance of a regression coefficient is inflated due to multicollinearity in the model.The combination of heathmap and VIF score helped us to identify potential features to drop for the feature selection. In the head map below (**Chart A**), we saw two sets of strong correlations, “Create Job” vs. “Retained Job” (100%) and “SBA Portion” vs. “RevLineCr” (-0.71). When we calculated the VIF score for each variable, we observed very high VIF for “Create Job” and “Retained Job”. Contrastingly, the VIF for both “SBA Portion” and “RevLineCr” were within a reasonable range (**Table 1**). After dropping “Create Job” as a variable, we saw the VIF score of “Retained Job” reduced from 112.86 to 1.00 (**Table 1**).

Chart A: Correlation Coeff.

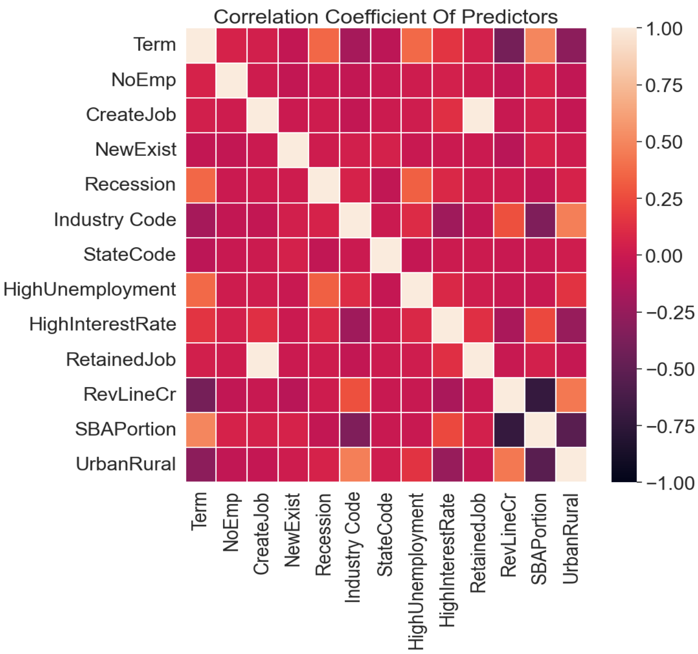


Table 1: Top VIF Variables

Variable	VIF	VIF(Removed)
CreateJob	122.93	N/A
RetainedJob	122.86	1.00
SBAPortion	2.58	2.58
RevLineCr	2.06	1.34
UrbanRural	1.41	2.06
Term	1.34	1.41

Challenges with Imbalance Data

We discovered that our dataset is imbalanced at a ratio of approximate 5:1 (non-default loan samples vs. defaulted loan samples) (**Chart B**). The problem with training machine learning models with an imbalanced dataset is that the model will be biased towards the majority class. As observed, when we tried to run different classification models with the imbalanced datasets, we saw incredibly false positive rates ranging from 83.41% to 9.39%, in addition to the high accuracy scores (range from 82.79% to 95.04%). For example, with imbalanced data, the KNN model would have 86.9% accuracy but a false positive rate of 83.41% as KNN uses a majority voting scheme Our Decision Tree would have given us a false prediction of 92.3% accuracy with a false positive rate of 17.76%. See full results of modeling with imbalance data in **Appendix A**.Therefore, we decided to correct the imbalance data through random oversampling. We downsampled the non-defaulted loans. We randomly selected 120,000 data points from the original 500,000 non-defaulted loan samples to match the size of defaulted loans. After downsampling, our data set is balanced to a ratio of approximate 1:1(non-default vs. default loan samples) **(Table 2).**

Chart B

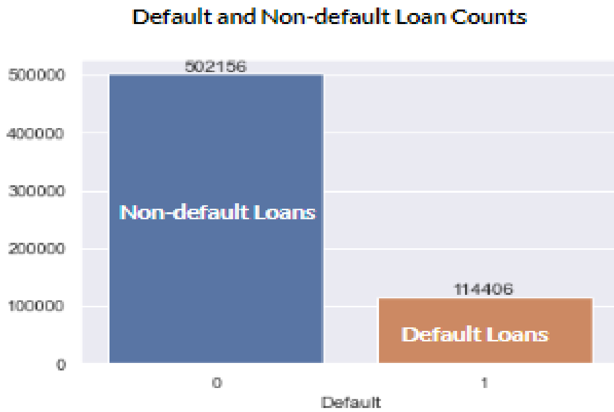


Table 2

Datapoint Count Before/After Balancing Data

	Before	After
Non-Defaulted Loan Count	502,156	120,000
Defaulted Loans Count	114,406	114,406
Ratio	4.4	1.0

Feature Selection

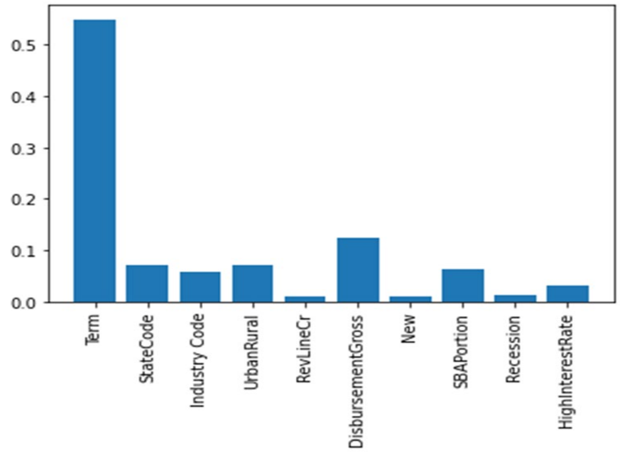
Chart C

Boruta Feature Selection Result

Passes the test: Term - Ranking: 1 ✓
Doesn't pass the test: NoEmp - Ranking: 4 ✗
Doesn't pass the test: Franchise - Ranking: 5 ✗
Passes the test: StateCode - Ranking: 1 ✓
Passes the test: Industry Code - Ranking: 1 ✓
Doesn't pass the test: CreateJob - Ranking: 3 ✗
Doesn't pass the test: RetainedJob - Ranking: 7 ✗
Passes the test: UrbanRural - Ranking: 1 ✓
Passes the test: RevLineCr - Ranking: 1 ✓
Doesn't pass the test: LowDoc - Ranking: 7 ✗
Passes the test: DisbursementGross - Ranking: 1 ✓
Passes the test: New - Ranking: 1 ✓
Doesn't pass the test: RealEstateBacked - Ranking: 7 ✗
Passes the test: SBAPortion - Ranking: 1 ✓
Passes the test: Recession - Ranking: 1 ✓
Doesn't pass the test: HighUnemployment - Ranking: 2 ✗
Passes the test: HighInterestRate - Ranking: 1 ✓

Boruta feature selection method ***[3]*** is used to determine the importance of features. Boruta uses feature importance scores from random forest models to rank the features. It introduces shadow features, which are copies of original features but with randomly mixed values, so that their distribution remains the same but they are not significant to any model. Boruta selection is a multi iteration process. In each iteration, first shadow features are generated, and fed to the random forest model with the original features. Original features' importance is then compared with the highest importance of a shadow feature. Features which significantly outperform best shadow features are selected. Features which significantly underperform best shadow feature are rejected and removed from the set for all subsequent iterations. After conducting Boruta for balanced data, Term, StateCode, Industry Code, UrbanRural, RevLineCr, DisbursementGross, New, SBAPortion, Recession, and HighInterestRate variables passed the test. NoEmp, Franchise, CreateJob, LowDoc, HighUnemployment, RetainedJob and RealEstateBacked failed the test **(Chart C).** We noted that Term and disbursement amount are the most important two features **(Chart D).**

Chart D: Feature Importance



Overview of Modeling

Our initial objective is to train six models (Logistic Regression Model, Support Vector Machine (SVM), K-Nearest-Neighbors(KNN), Decision Tree, XGBoost, and Random Forest). Based on the cross-validation accuracy, we then pick the model with the highest accuracy for further testing.

Methodology

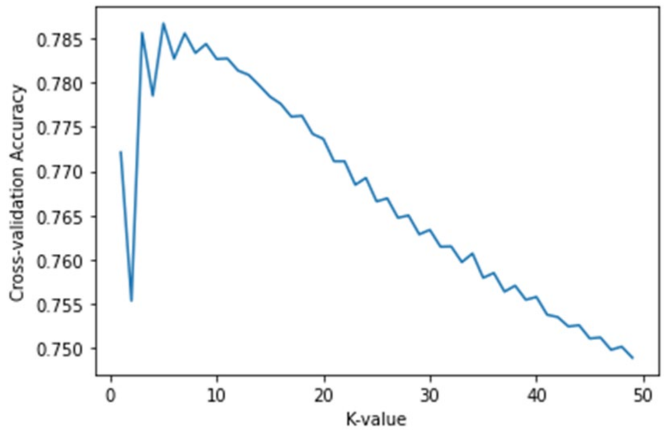
We first split the data and use 75% of the data to train the following 6 models.:

Logistic Regression Model: We tried this traditional classification model and used 10-fold cross validation and calculated the accuracy for each fold. Then we calculated the mean and standard deviation of the 10 accuracies.

Support Vector Machine (SVM): We selected to use this model since it can efficiently perform a non-linear classification. We tried to use the GridSearchCV() function with 3-fold validation to find the best combination of C gamma and kernel. We tried standardizing our data, narrowing the tuning range and decreasing the folds for cross validation. But we still failed to get the result due to the large dataset and low laptop CPU capacity.

K-Nearest-Neighbors(KNN): We chose this model because it is easy to interpret, understand, and implement. We used a loop with 10-fold cross validation to find the optimal K with the highest model accuracy in the range from 1 to 50. We also plotted the K-Value vs. Cross Validation Accuracy chart (**Chart E**). Based on the results, when K=5, the model reaches the highest accuracy.

Chart E: KNN CV values



Decision Tree: We applied the 10-folds Grid Search Cross Validation method to fit the decision tree with the training data. We recognized that the decision tree model has its limitations: there exist possibilities of overfitting due to single tree fitting. We want to continue exploring with other advanced tree models: Random Forest and gradient-boosted decision tree (GBDT).

Random Forest Algorithm: The Random Forest Algorithm combines the output of multiple random Decision Trees to generate the final output. One of the advantages of using random forest trees is to avoid overfitting and produce a more accurate result since it's fitting trees on random selection of data. Similar to decision tree model fitting, we applied the randomized search cross validation.

Extreme Gradient Boosting (XGBoost): Similar to random forests, but XGBoost uses additive methods to build trees one at a time with gradient boosting to learn the optimal discriminative model for prediction. We used 10-folds cross validation to get the average cross validation accuracy.

Model Training Result

Based on the model training results below (**Table 3**), we noted that the high false positive ratio issue due to imbalanced data was solved so that we were confident to select the model based on the average accuracy. Overall, XGBoost model reached the highest average cross validation accuracy value of **92.80%** while keeping other statistics good as well. See **Appendix C** for detailed confusion matrix, classification report and ROC curves.

Table 3 Model Training Result

	SVM	Logistic Regression	KNN	Decision Tree	Random Forest	XGBoost Forest
Optimal Hyperparameter **Optimization method:** - gridsearchCV - randomizedCV - Loop	Fail to get the result due to low laptop CPU capacity	N/A	K=5	Max_depth: 12 Min_samples_split: 4 Min_samples_leaf: 6 Criterion: entropy	N_estimators: 100 Min_samples_split: 2 Min_samples_ leaf: 2 Max_depth: 20 Criterion: gini	n_estimators: 100 min_samples_split: 3 min_samples_leaf: 5 max_depth: 10 criterion: entropy
Average CV Accuracy	N/A	71.73%	78.67%	91.38%	92.57%	92.80%
Std. of Accuracy	N/A	0.28%	0.36%	0.11%	0.14%	0.13%
FNR	N/A	28.56%	13.28%	7.03%	4.62%	5.44%
FPR	N/A	27.94%	15.72%	8.01%	4.18%	5.13%
AUC Score	N/A	0.78	0.94	0.98	0.99	0.99

Prediction without Macroeconomic Data

We also developed a model without Macroeconomic data to study how good we can predict loan default when we don't have a good Macroeconomic forecast. We removed "High Interest Rate" and "Recession" macroeconomic features. When loan approval decisions are made, the Macroeconomic data during the loan term is future information. For short term loans, we usually have a Macroeconomic forecast. However, for mid term and long term loans, the Macroeconomic forecast may not be very accurate. Our training result shows that the model without Macroeconomic data can still make great predictions. XGBoost CV result using training data w/o Macroeconomic Data (**Table 4**) showed an average of 91.78% accuracy and 0.98 AUC score. The False Positive Rate (FPR) and False Negative Rate (FNR) got slightly worse but all below 9%. Therefore, we can still leverage the model when Macroeconomic data is not available.

Table 4. XGBoost CV result with and w/ Macroeconomic

	With Macroeconomic Data	No Macroeconomic Data
Accuracy	92.80%	**91.78%**
FNR	5.44%	6.72%
FPR	5.13%	8.62%
AUC Score	0.99	0.98

Testing on Selected Model - XGBoost Forecast

We tested our final XGBoost model accuracy using the 25% reserved test data both with and without macroeconomic data. As the below figure shows **(Table 5)**, both models consistently deliver a strong result with high accuracy and AUC score, and low FNR and FPR scores. Overall, the model with macroeconomic data is slightly better in performance compared to the model without the macroeconomic data, but overall, both models perform strongly in predicting unseen test data . See **Appendix D** for detailed confusion matrix, classification report and ROC curves.

Table 5. XGBoost CV testing result with and w/o Macroeconomic Data Comparison

	With Macroeconomic Data	No Macroeconomic Data
Accuracy	91.72%	91.11%
FNR	6.96%	7.75%
FPR	7.28%	10.06%
AUC Score	0.98	0.97

Key Takeaway

We obtained several important key takeaways or observations from the project. Throughout our project, we have learned and practiced using multiple performance indicators such as accuracy, FPR, FNR and AUC scores to detect issues in sample data and model. We also found that for small business loan default prediction, variables such as term and disbursement amount have the highest importance feature. Those factors can be important for small business banks to consider when building loan approval related models.

Based on our result, we have observed that models with macroeconomic data perform slightly better than models without macroeconomic data. We suspect the reason is because the macroeconomic data makes the prediction more accurate but is subject to the loan period. For example, when loan terms are long, the macroeconomic conditions can be unpredictable.

Conclusion

Business Impact for Banks

We want to use the same SBA Loan Historical data set to quantify the benefit of implementing the XGboost model we developed. Based on the raw dataset, we calculated that the banks grant a total of 899,164 loans and 157,558 of the total are defaulted which is approx. 17.5% of total. The gross amount of loans approved by the banks is $173,257,192,433 and the total default amount is $12,141,676,859. The default rate is close to 7% (12,141,676,859/173,257,192,433).

Based on our model testing result, our model could reach a very low false positive rate of approximately 7.28%. Therefore if the banks used our model to make the approval decision for these loans, the number of default loans could decrease to 65,663 (157,558- 899,164*(17.5%-7.28%)). With that said the bank might decrease the amount of default loans by 58%((157,558-65,663)/157,558). The bank could avoid a total loss of $7,042,172,578 (12,141,676,859*58%) due to loan defaults. The default rate would reduce from 7% to 3% ((12,141,676,859-7,042,172,578)/173,257,192,433)).

In addition, We assume a bank's loan representative approves 800 loans per year and a loan representative's salary is $50,000/year. If the bank uses the model to replace loan representatives, the bank can save 56,197,750 (899,164/800*50,000) of total labor costs.Therefore, the bank is able to save 7,042,172,578+56,197,750 = 7,098,370,328 for all the loans approved by banks in the data set. That is approximately 4% of the gross loan amount approved. If the interest rate for the loan is 4%/year, the bank is able to save one year's revenue by implementing the model.

Next Steps

Develop the production pipeline for new loan application predictions using the XGBoost model we developed. The model should also be refitted every 6 months to include the new loan application data to improve model accuracy.

Future Research Recommendation

The main goal of this study is to predict SBA default loans in relevance to the macroeconomic variables. However, the performance of macroeconomic variables in default classification machine learning algorithms has not been tested. In reality the future macroeconomic performance is unpredictable, we need to better understand the relationship between the prediction result and the timeframe of the macroeconomic data. Therefore, we recommend future research to continue to focus on best understanding macroeconomic variances.

Citations

1. *Unemployment Rate - Definition*, http://www.moneychimp.com/glossary/unemployment_rate.htm.

2. Team, The Investopedia. “Recession: Meaning in Economics with Causes.” *Investopedia*, Investopedia, 21 Oct. 2022, https://www.investopedia.com/terms/r/recession.asp#:~:text=A%20recession%20is%20a%20significant,%2C%20consumer%20demand%2C%20and%20employment.

3. “Wrapper Algorithm for All Relevant Feature Selection [R Package Boruta Version 7.0.0].” *The Comprehensive R Archive Network*, Comprehensive R Archive Network (CRAN), 21 May 2020, https://cran.r-project.org/web/packages/Boruta/.

4. Roijmans, Sjef. “Macroeconomic Factors in Loan Default Prediction: A Machine Learning Based Approach.” Netspar, 3 Feb. 2021, https://www.netspar.nl/en/publication/macroeconomic-factors-in-loan-default-prediction-a-machine-learning-based-approach/.

5. Lin, Mei. “Should This Loan Be Approved or Denied?”: A Large Dataset with Class Assignment Guidelines. 5 Apr. 2018, https://www.tandfonline.com/doi/full/10.1080/10691898.2018.1434342?scroll=top&needAccess=true.“Bureau of Labor Statistics Data.” U.S. Bureau of Labor Statistics, U.S. Bureau of Labor Statistics, https://data.bls.gov/timeseries/LNS14000000.

6. “Brave-Butters-Kelley Real Gross Domestic Product.” FRED, 3 Oct. 2022, https://fred.stlouisfed.org/series/BBKMGDP.

7. “Federal Funds Effective Rate.” FRED, 9 Oct. 2022, https://fred.stlouisfed.org/series/FEDFUNDS

8. “Bureau of Labor Statistics Data.” U.S. Bureau of Labor Statistics, U.S. Bureau of Labor Statistics, https://data.bls.gov/timeseries/LNS14000000.

Appendix

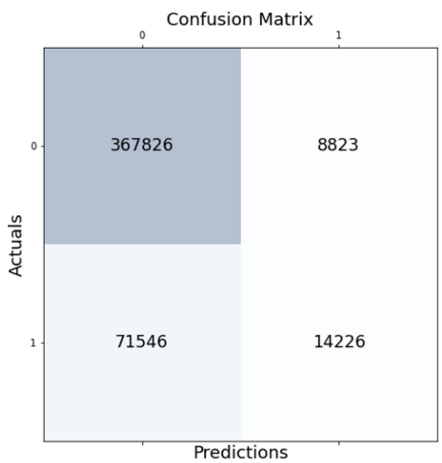
Appendix A: Model Training Results with Imbalanced Data

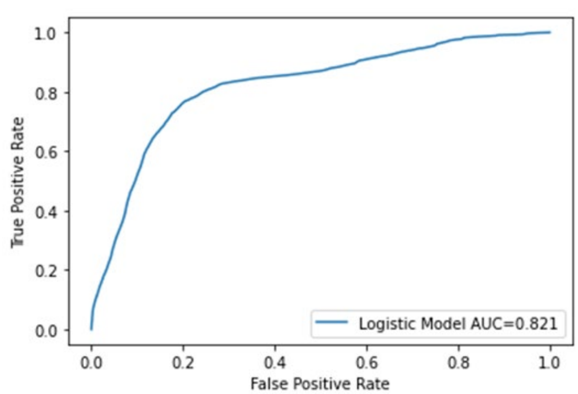
Table Summary:

	SVM	Logistic Regression	KNN	Decision Tree	Random Forest	XGBoost Forest
Optimal Hyperparameter **Optimization method:** - gridsearchCV - randomizedCV - Loop	Fail to get the result due to low laptop CPU capacity	N/A	K=9	Max_depth: 12 Min_samples_spli: 5 Min_samples_lea: 2	N_estimators: 500 Min_samples_split: 2 Min_samples_ leaf: 2 Max_depth: 30 Criterion: entropy	n_estimators: 100 min_samples_split: 3 min_samples_leaf: 5 max_depth: 10 criterion: entropy
Average CV Accuracy	**N/A**	**82.79%**	**86.86%**	**92.34%**	**94.45%**	**95.04%**
Std. of Accuracy	N/A	0.37%	0.08%	0.17%	0.13%	0.20%
FNR	N/A	2.34%	3.34%	2.52%	0.60%	1.65%
FPR	**N/A**	**83.41%**	**44.78%**	**17.76%**	**9.39%**	**13.07%**

Confusion Matrix, Classification Report and ROC Curves for Each Model:

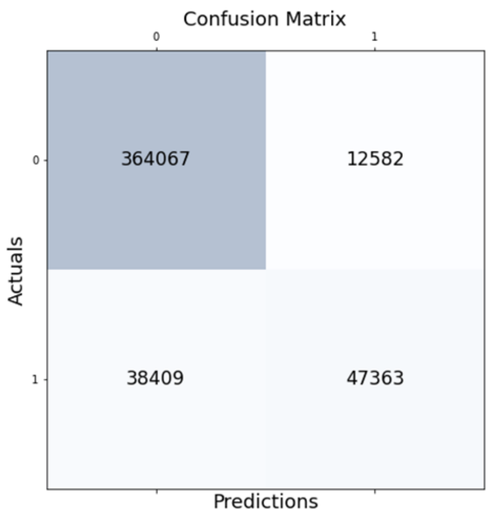
<u>Logistic regression Model</u>

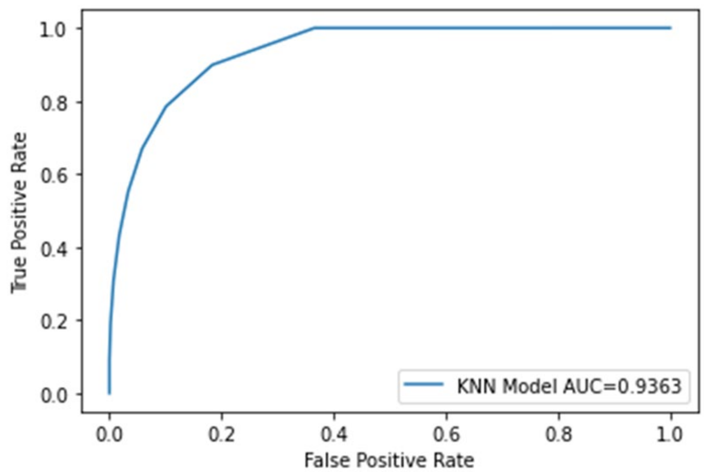




```
Classification Report for Logistic Regression Model:
              precision    recall  f1-score   support

           0     0.8372    0.9766    0.9015    376649
           1     0.6172    0.1659    0.2615     85772

    accuracy                         0.8262    462421
   macro avg     0.7272    0.5712    0.5815    462421
weighted avg     0.7964    0.8262    0.7828    462421

false negative rate is: 0.023424992499648214
false positve rate is: 0.8341416779368559
```

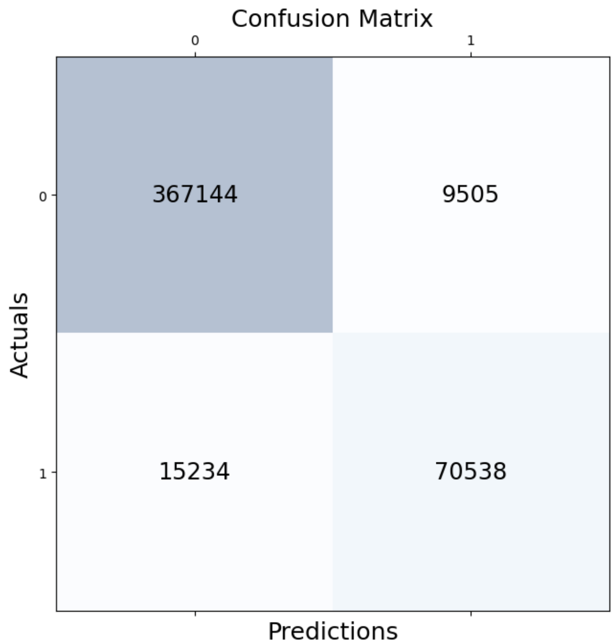
KNN Model

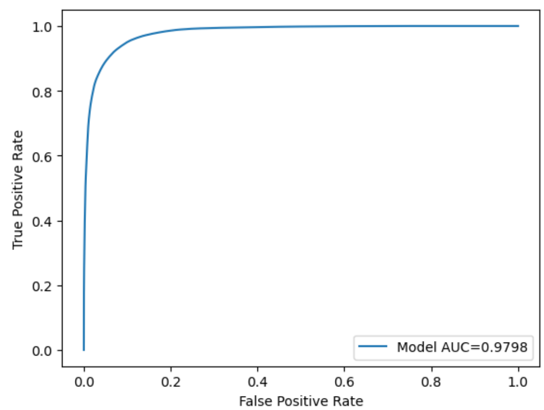





```
Classification Report for KNN Model:
              precision    recall  f1-score   support

           0     0.9046    0.9666    0.9346    376649
           1     0.7901    0.5522    0.6501     85772

    accuracy                         0.8897    462421
   macro avg     0.8473    0.7594    0.7923    462421
weighted avg     0.8833    0.8897    0.8818    462421

false negative rate is: 0.033405106611195036
false positve rate is: 0.44780347899081285
```

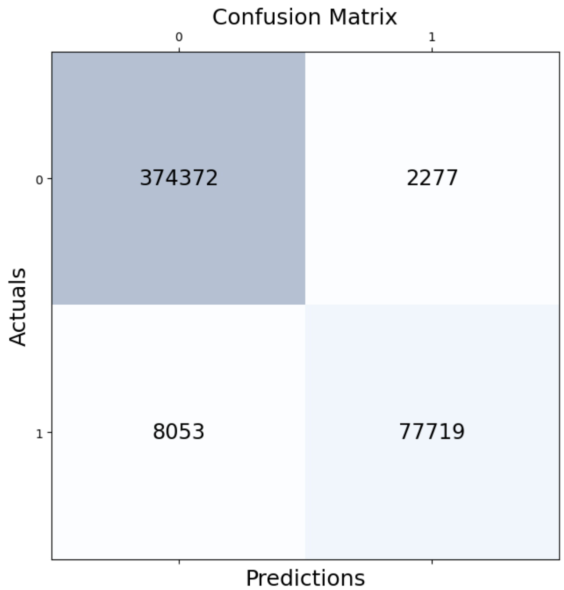
Decision Tree Model

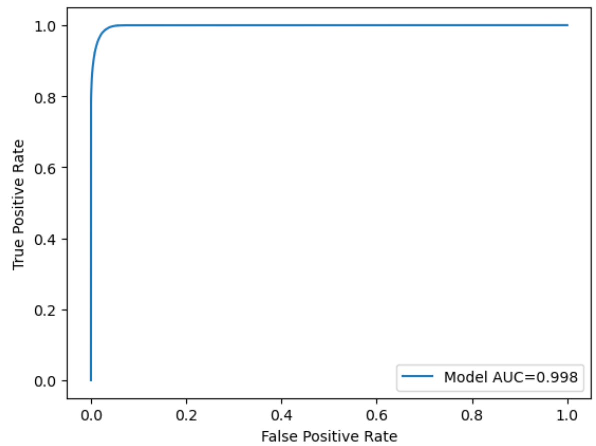





```
Classification Report for Model:
              precision    recall  f1-score   support

           0     0.9602    0.9748    0.9674    376649
           1     0.8813    0.8224    0.8508     85772

    accuracy                         0.9465    462421
   macro avg     0.9207    0.8986    0.9091    462421
weighted avg     0.9455    0.9465    0.9458    462421

false negative rate is: 0.02523569689551811
false positve rate is: 0.1776104089912792
```

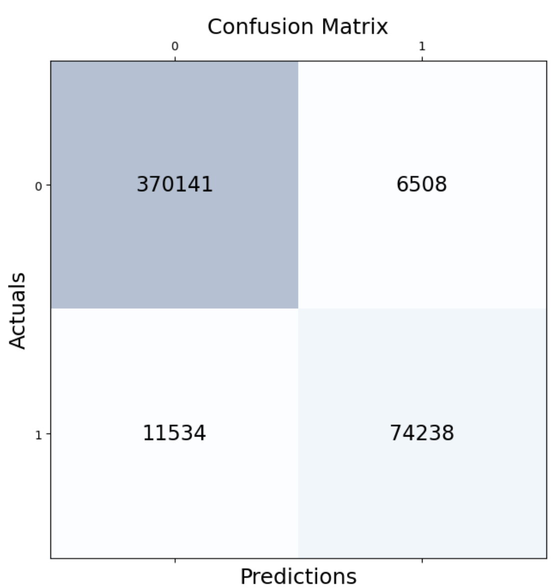
Random Forest Model

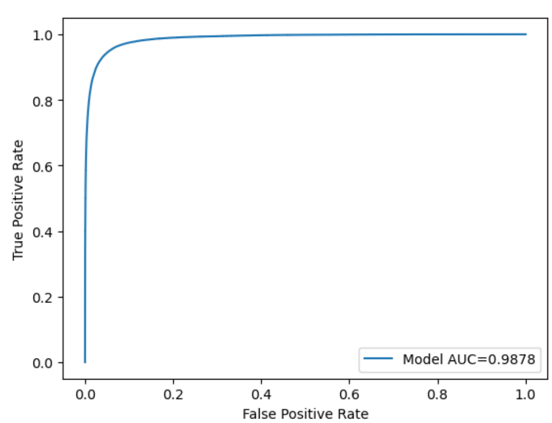





```
Classification Report for Model:
              precision    recall  f1-score   support

           0     0.9789    0.9940    0.9864    376649
           1     0.9715    0.9061    0.9377     85772

    accuracy                         0.9777    462421
   macro avg     0.9752    0.9500    0.9620    462421
weighted avg     0.9776    0.9777    0.9774    462421

false negative rate is: 0.006045416289436585
false positve rate is: 0.09388844844471389
```

XGBoost Model





```
Classification Report for Model:
              precision    recall  f1-score   support

           0     0.9698    0.9827    0.9762    376649
           1     0.9194    0.8655    0.8917     85772

    accuracy                         0.9610    462421
   macro avg     0.9446    0.9241    0.9339    462421
weighted avg     0.9604    0.9610    0.9605    462421

false negative rate is: 0.01727868652246521
false positve rate is: 0.13447278832252949
```

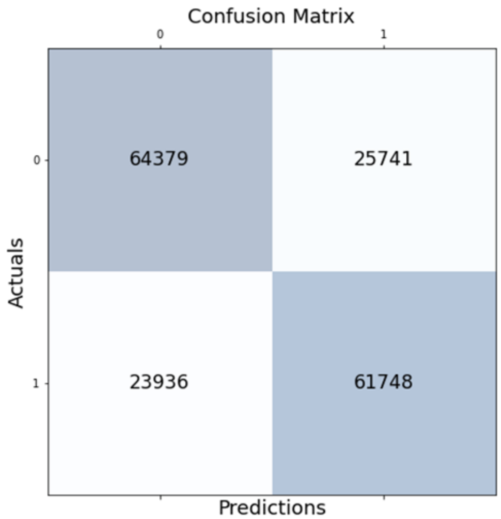
Appendix B: Data Manipulation and Transformation

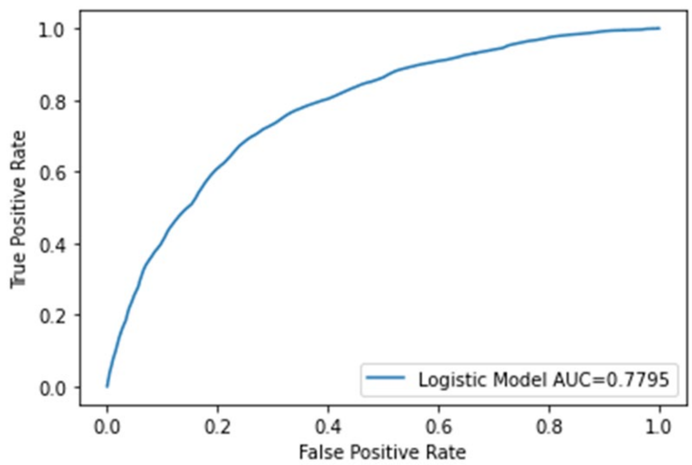
- LoanNr_ChkDgt: loan number and unique identifier. We didn't include it in the model as we don't think it would impact the probability of default.
- Name: name of the borrower. We didn't include it in the model as we don't think it would impact the probability of default.
- City, State, Zip: these represent the geo location of the borrower. We kept state as a feature of interest and converted it to categorical integers. We didn't use city or zip since they are too granular.
- Bank, BankState: the lender name and state. We didn't include them in the model as we don't think they would impact the probability of default.
- NAICS: North American industry classification system code. This represents which industry the borrower is in. The first two digits of the NAICS classification represent the economic sector. We extracted the first two digits and included them in the study.
- ApprovalDate and ApprovalFY: date and year of the loan approval date. They are not used as features in the model since we believe DisbursementDate is more representative of when the borrower receives the money.
- Term: term of the loan in the number of months. It is included as a feature without transformation.
- NoEmp: number of employees of the borrower. It is included as a feature without transformation.
- NewExist: whether the borrower is an existing business (in existence for more than 2 years or more) or a new business (in existence for less than or equal to 2 years). A feature called "New" is deducted from this column. For new business, New=1 and for old business, New=0.
- CreateJob: number of jobs created. It is included as a feature without transformation.
- RetainedJob: number of jobs retained. It is included as a feature without transformation for unbalanced data model but dropped for balanced data model.
- FranchiseCode: A column named "Franchise" is deducted from this column. If FranchiseCode <=1, Franchise=0 means no franchise. Otherwise Franchise=1 means there are franchises.
- UrbanRural: 1 means Urban, 2 means rural, 0 means undefined. It is included as a feature without transformation.
- RevLineCr: revolving line of credit: 1 means Yes, 0 means No. Missing data is dropped. It is included as a feature.

- LowDoc: In order to process more loans efficiently, a “LowDoc Loan” program was implemented where loans under $150,000 can be processed using a one-page application. 1 means Yes, 0 means No. Missing data is dropped. It is included as a feature.
- ChgOffDate: The date when a loan is declared to be in default. We didn’t use this column in the feature as we use MIS_Status to determine if the loan defaulted.
- ChgOffPrinGr: Charged-off amount and MIS_Status: charge off status. These two columns are used together to determine if a loan defaulted. If MIS_Status='CHGOFF' then y=1. Or if ChgOffPrinGr>100, response y is set to 1. For the rest of the cases, y is set to 0 as not default.
- DisbursementDate: loan disbursement date. It is used with loan terms to deduct the loan period and merge with other datasets.
- DisbursementGross: Amount disbursed to the borrower. It is included as a feature without transformation.
- BalanceGross: Gross amount outstanding. Not used as a feature.
- GrAppv: not used since it is the same as DisbursementGross.
- SBA_Appv: SBA’s guaranteed amount of approved loan. A column called “SBAPortion” is deducted from this column. SBAPortion=SBA_Appv/GrAppv. It means what percentage of the loan is backed by SBA.
- RealEstateBacked: it is an engineered feature. If Term > 240 then RealEstateBacked=1. It means the loan is backed by real estate. If Term<=240 then RealEstateBacked=0.
- EndDate: end date of the loan. It is an engineered feature which is the disbursement date offset by loan terms (number of months).

Appendix C:
Model Training Results with Balanced Data (Confusion Matrix, Classification Report and ROC Curves for Each Model)

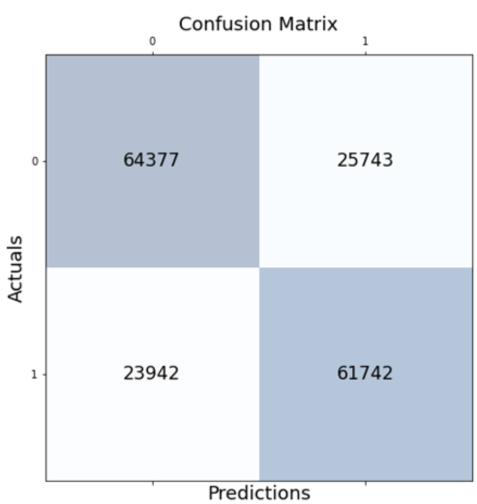
<u>**Logistic regression Model with Macroeconomic Data**</u>

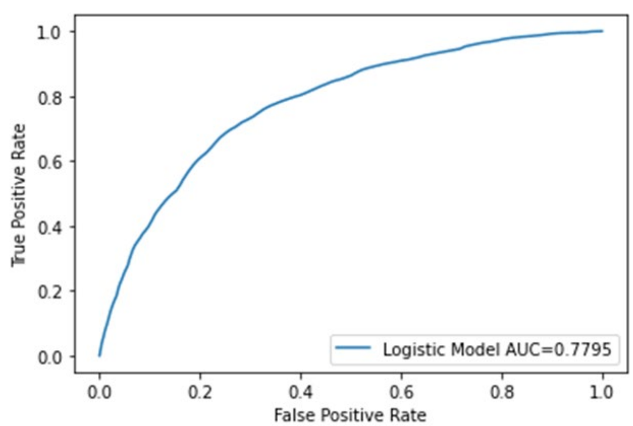





```
Classification Report for Logistic Regression Model:
              precision    recall  f1-score   support

           0     0.7290    0.7144    0.7216     90120
           1     0.7058    0.7206    0.7131     85684

    accuracy                         0.7174    175804
   macro avg     0.7174    0.7175    0.7174    175804
weighted avg     0.7177    0.7174    0.7175    175804

false negative rate is: 0.28563027075011094
false positve rate is: 0.2793520377199944
```

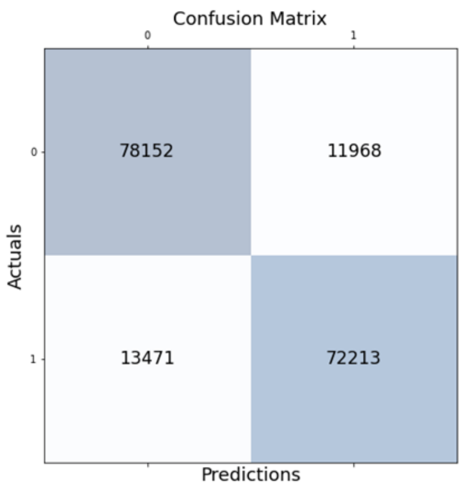
Logistic regression Model w/o Macroeconomic Data

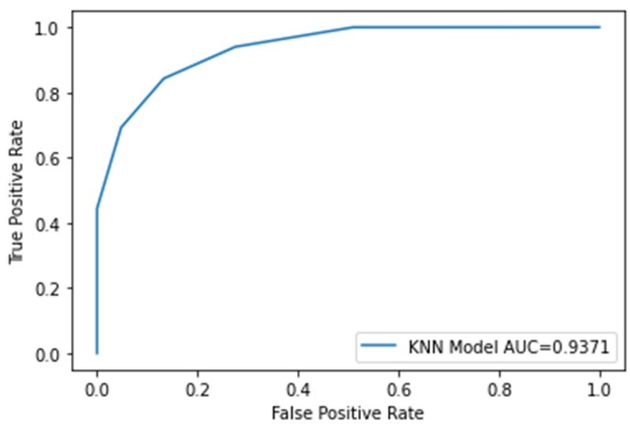




```
Classification Report for Logistic Regression Model:
              precision    recall  f1-score   support

           0     0.7289    0.7143    0.7216     90120
           1     0.7057    0.7206    0.7131     85684

    accuracy                         0.7174    175804
   macro avg     0.7173    0.7175    0.7173    175804
weighted avg     0.7176    0.7174    0.7174    175804

false negative rate is: 0.285652463382157l
false positve rate is: 0.2794220624620699
```

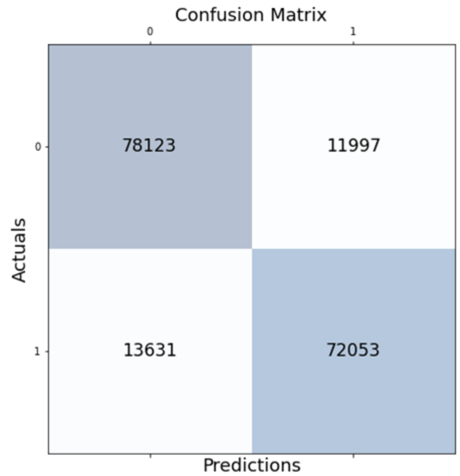
KNN Model with Macroeconomic Data

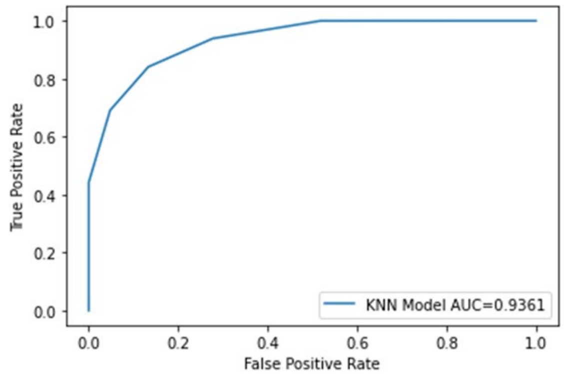





```
Classification Report for KNN Model:
              precision    recall  f1-score   support

           0     0.8530    0.8672    0.8600     90120
           1     0.8578    0.8428    0.8502     85684

    accuracy                         0.8553    175804
   macro avg     0.8554    0.8550    0.8551    175804
weighted avg     0.8553    0.8553    0.8553    175804

false negative rate is: 0.13280071016422548
false positve rate is: 0.15721721674991831
```

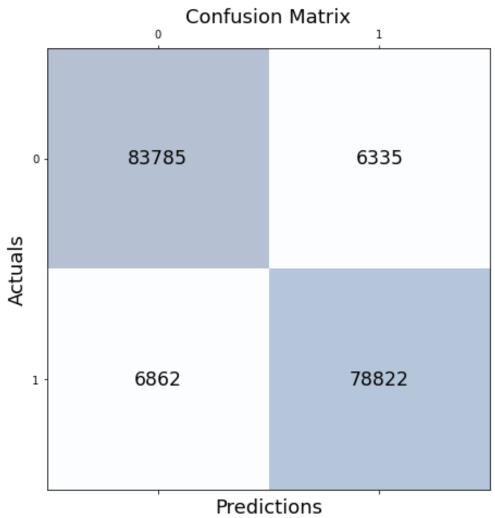
KNN Model w/o Macroeconomic Data

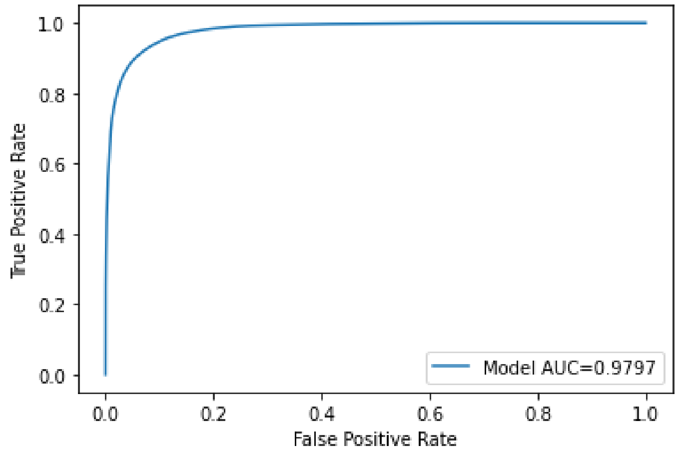





```
Classification Report for KNN Model:
              precision    recall  f1-score   support

           0     0.8514    0.8669    0.8591     90120
           1     0.8573    0.8409    0.8490     85684

    accuracy                         0.8542    175804
   macro avg     0.8544    0.8539    0.8541    175804
weighted avg     0.8543    0.8542    0.8542    175804

false negative rate is: 0.1331225033288948
false positve rate is: 0.15908454320526585
```

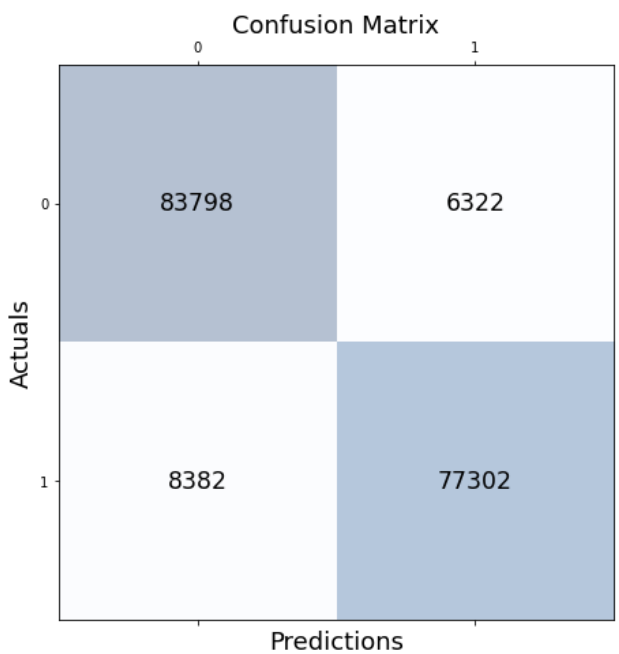
Decision Tree w/ Econ

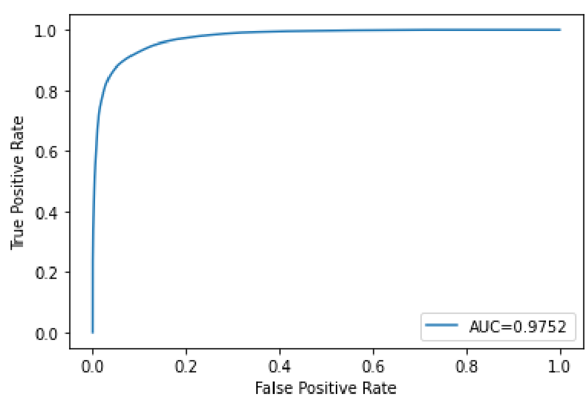





```
Classification Report for Model:
              precision    recall  f1-score   support

           0     0.9243    0.9297    0.9270     90120
           1     0.9256    0.9199    0.9228     85684

    accuracy                         0.9249    175804
   macro avg     0.9250    0.9248    0.9249    175804
weighted avg     0.9249    0.9249    0.9249    175804

false negative rate is: 0.07029516200621394
false positve rate is: 0.08008496335371831
```

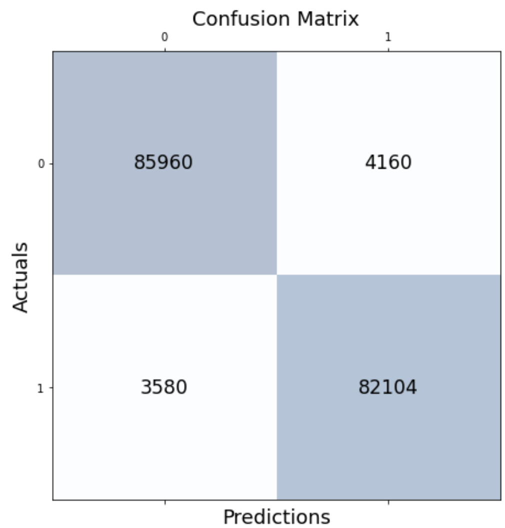
Decision Tree w/o Econ

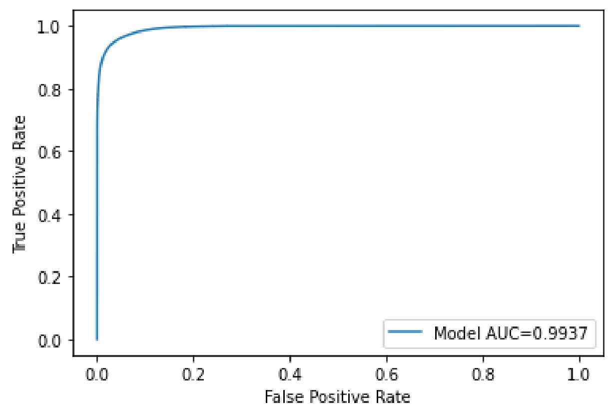




```
Classification Report for XGboost Model:
              precision    recall  f1-score   support

           0     0.9051    0.9224    0.9137     29880
           1     0.9177    0.8994    0.9084     28722

    accuracy                         0.9111     58602
   macro avg     0.9114    0.9109    0.9111     58602
weighted avg     0.9113    0.9111    0.9111     58602

false negative rate is: 0.07757697456492638
false positve rate is: 0.10061973400181046
```

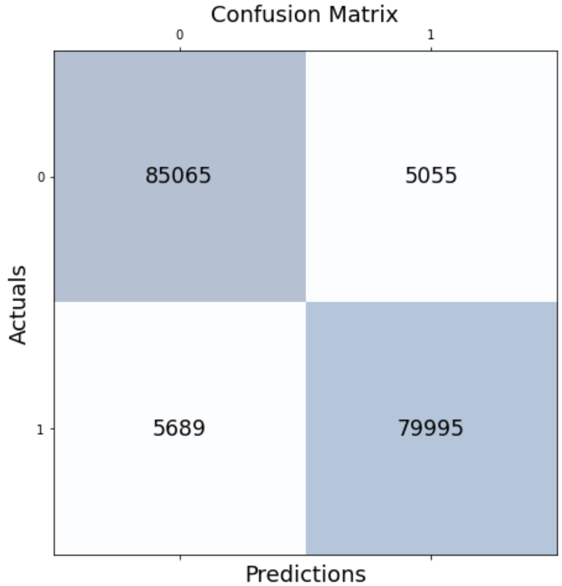
Random Forest w/ Econ

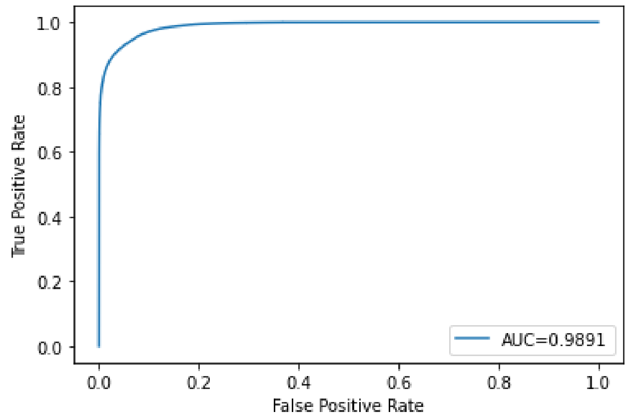





```
Classification Report for Model:
              precision    recall  f1-score   support

           0     0.9600    0.9538    0.9569     90120
           1     0.9518    0.9582    0.9550     85684

    accuracy                         0.9560    175804
   macro avg     0.9559    0.9560    0.9560    175804
weighted avg     0.9560    0.9560    0.9560    175804

false negative rate is: 0.0461606746560142
false positve rate is: 0.04178142943840157
```

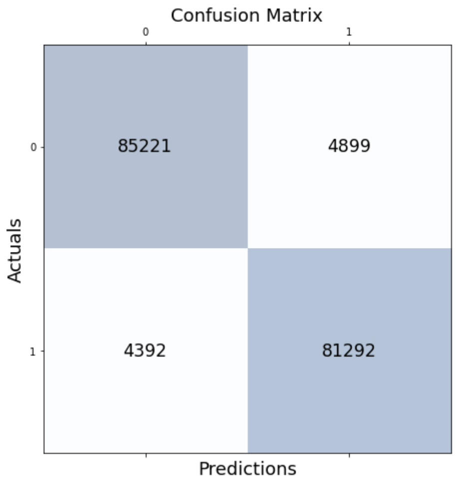
Random Forest w/o Econ

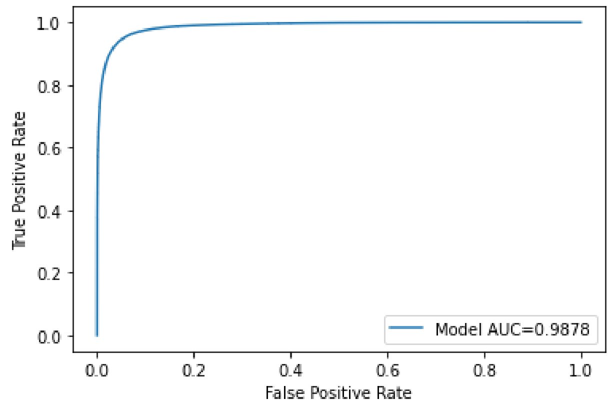





```
Classification Report for Model:
              precision    recall  f1-score   support

           0     0.9091    0.9298    0.9193     90120
           1     0.9244    0.9022    0.9132     85684

    accuracy                         0.9164    175804
   macro avg     0.9167    0.9160    0.9162    175804
weighted avg     0.9165    0.9164    0.9163    175804

false negative rate is: 0.0701509098979139
false positve rate is: 0.0978245646795201
```

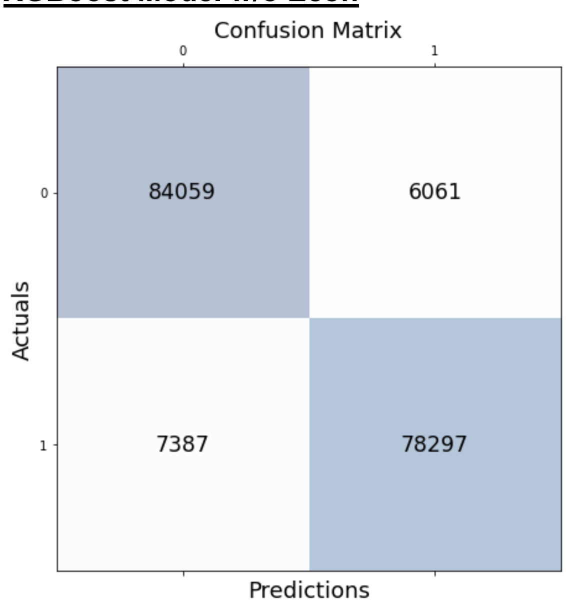
XGBoost w/ Econ

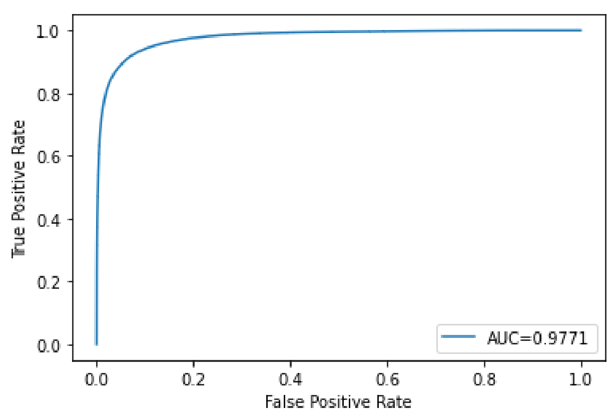





```
Classification Report for Model:
              precision    recall  f1-score   support

           0     0.9510    0.9456    0.9483     90120
           1     0.9432    0.9487    0.9459     85684

    accuracy                         0.9472    175804
   macro avg     0.9471    0.9472    0.9471    175804
weighted avg     0.9472    0.9472    0.9472    175804

false negative rate is: 0.05436085219707057
false positve rate is: 0.05125811119929041
```

XGBoost Model w/o Econ

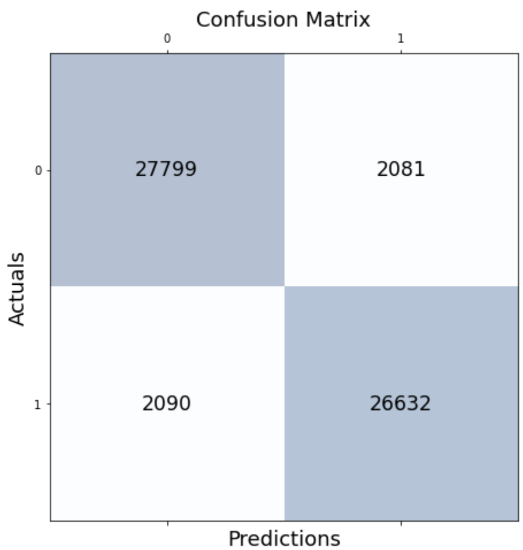


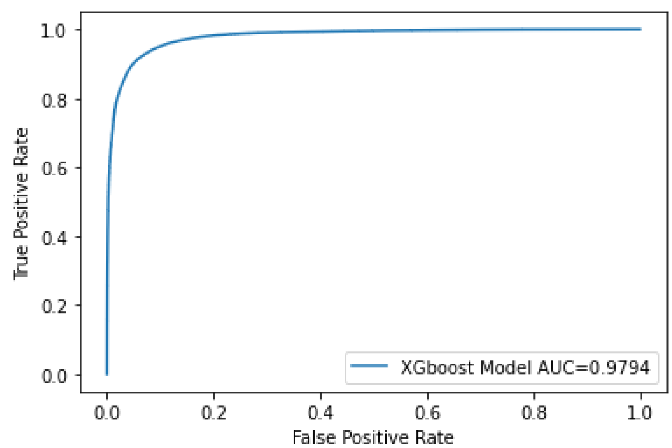


```
Classification Report for Model:
              precision    recall  f1-score   support

           0     0.9192    0.9327    0.9259     90120
           1     0.9282    0.9138    0.9209     85684

    accuracy                         0.9235    175804
   macro avg     0.9237    0.9233    0.9234    175804
weighted avg     0.9236    0.9235    0.9235    175804

false negative rate is: 0.06725477141588992
false positve rate is: 0.08621212828532748
```

Appendix D:
Model Testing Results with Balanced Data (Confusion Matrix, Classification Report and ROC Curves for Each Model)

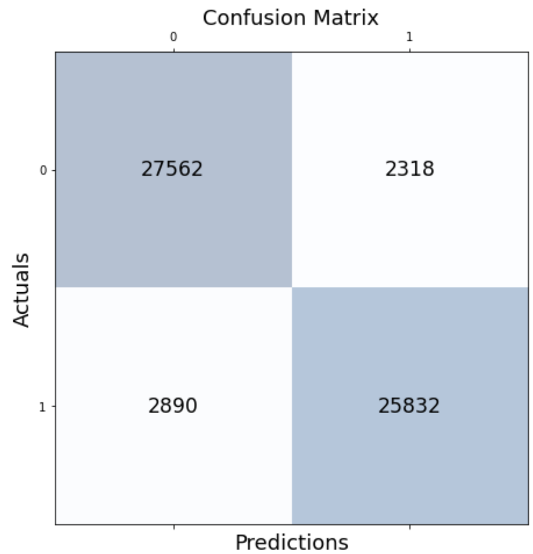
XGBoost Model w/ Econ

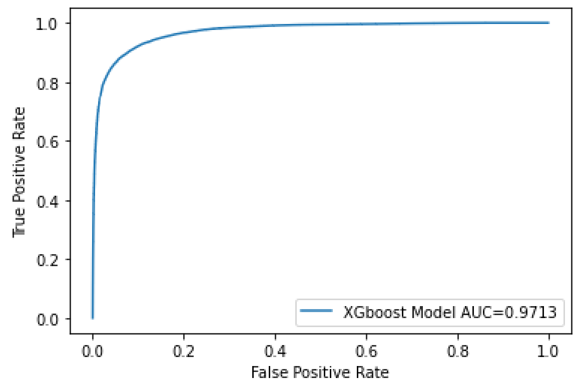





```
Classification Report for XGboost Model:
              precision    recall  f1-score   support

           0     0.9301    0.9304    0.9302     29880
           1     0.9275    0.9272    0.9274     28722

    accuracy                         0.9288     58602
   macro avg     0.9288    0.9288    0.9288     58602
weighted avg     0.9288    0.9288    0.9288     58602

false negative rate is: 0.06964524765729585
false positve rate is: 0.07276652043729545
```

XGBoost Model w/o Econ






```
Classification Report for XGboost Model:
              precision    recall  f1-score   support

           0     0.9051    0.9224    0.9137     29880
           1     0.9177    0.8994    0.9084     28722

    accuracy                         0.9111     58602
   macro avg     0.9114    0.9109    0.9111     58602
weighted avg     0.9113    0.9111    0.9111     58602

false negative rate is: 0.07757697456492638
false positve rate is: 0.10061973400181046
```

Appendix E: Full Project Timeline

Stage	**Project Milestone**	**Details**
Stage 1: Data Preparation	Data Exploration	Completed
	Data CleanUp/Handling	
	Data Transforming	
	Data Sampling	
Stage 2: Modeling	Feature Selection	Completed
	Model Training and Selection	Completed

Stage 3: Testing	Performance Analysis	- Hyperparameter Tuning for XGBoost Forest model - Develop second model without macroeconomic data due to the unpredictability of future economy - Evaluate risk of overfitting
	Test Setup	Testing for model accuracy using 25% reserved test data
	Validation Results	
Stage 4: Conclusion	Provide recommendations for future researches	Final Presentation Video (10-12 minutes)
	Discussion limitations	Final Report (8-10 pages)
	Final thoughts	