City University of New York (CUNY)
CUNY Academic Works

Dissertations, Theses, and Capstone Projects CUNY Graduate Center

6-2021

An Analysis of Machine Learning Techniques for Economic Recession Prediction

Sheridan Kamal
The Graduate Center, City University of New York

How does access to this work benefit you? Let us know!

More information about this work at: https://academicworks.cuny.edu/gc_etds/4378
Discover additional works at: https://academicworks.cuny.edu

This work is made publicly available by the City University of New York (CUNY).
Contact: AcademicWorks@cuny.edu

AN ANALYSIS OF MACHINE LEARNING TECHNIQUES FOR ECONOMIC RECESSION PREDICTION

by

SHERIDAN KAMAL

A master's capstone project submitted to the Graduate Faculty in Data Analysis and Visualization in partial fulfillment of the requirements for the degree of Master of Science, The City University of New York

2021

© 2021

SHERIDAN KAMAL

All Rights Reserved

An Analysis of Machine Learning Techniques for Economic Recession Prediction

by

Sheridan Kamal

This manuscript has been read and accepted for the Graduate Faculty in Data Analysis and Visualization in satisfaction of the capstone project requirement for the degree of Master of Science.

______________________	____
Date	Howard Everson
	Thesis Advisor
______________________	____
Date	Matthew Gold
	Executive Officer

THE CITY UNIVERSITY OF NEW YORK

Abstract

An Analysis of Machine Learning Techniques for Economic Recession Prediction

by

Sheridan Kamal

Advisor: Howard Everson

In this project I used the supervised machine learning methods logistic regression, decision tree classifier, k nearest neighbor classifier, and support vector classifier, to determine the best method to predict economic recessions. To do this, I used the function train_test_split to create training and testing sets and the function TimeSeriesSplit to create walk-forward cross validation sets to use when tuning the model parameters. Each machine learning method was trained on both scaled and unscaled data and was performed using default parameters and using the tuned parameters so that there were four models of each method. It was determined that the tuned Support Vector Classifier model trained on scaled data with a precision-recall area under the curve (PR AUC) score of 0.83 was the optimal model to predict economic recessions.

Table of Contents

Digital Manifest

I. PDF of Capstone Project White Paper

II. Zip file containing the contents of the GitHub repository which includes the final capstone project dataset and the capstone project Jupyter notebook (https://github.com/sheri-kamal/DAV-Capstone-SP2021)

Note on Technical Specifications

This project runs on Python 3 using the Jupyter notebook architecture therefore it is necessary to have Python 3 and Jupyter notebook installed. Instead of downloading them separately, Anaconda (Anaconda3) can be downloaded and also allows for easy software and package management. As an alternative to Jupyter notebook called JupyterLab may also be used.

To collect the data and run the models the following Python 3 packages must be installed:

- warnings
- numpy
- pandas
- quandl
- matplotlib
- seaborn
- sklearn

It is important to note that a large amount of data is imported using the Quandl library which limits anonymous API calls to 50 calls per day, therefore it is not advisable to run the cell importing data from the Quandl API (cell 1) more than 2 times.

List of Figures

List of Tables

Introduction

In the fields of Economics and Finance, a central idea is "the best indicator of future performance is past performance", which is why many economic forecasting models are heavily dependent on historical data when making predictions.

For my capstone project, I used machine learning methods to predict the onset of economic recessions over the past 45 years (1976 to 2021). By using relatively recent developments in machine learning methods I expect to be able to get a sense of where the U.S. economy is heading even if I cannot predict specific periods of economic recession. These forecasting methods, no doubt, will be useful for the Federal Reserve and other policy makers because if the economy is heading towards a recession they may be able to take appropriate and timely steps to mitigate the effects of an economic downturn. This analysis, I suspect, will also be useful for stock market analysts and stock traders because if it looks like a recession is looming they may want to shift funds from equities into other financial instruments.

Due to the pandemic caused by the novel Coronavirus (COVID-19), I noticed that the stock markets have been turbulent as a result of fears that the pandemic will have on different industries and trade. In the early days of the pandemic these fears have caused financial markets to tumble at an alarming rate and at times triggering the "circuit breakers" to halt trading. It is safe to say if the pandemic is prolonged then the negative impacts on the financial markets and the economy as a whole could push us further into a recession, which we have now been in since 3/1/2020 and continue to be in as of the date of submission of this paper. It is precisely these thoughts which drove me to research predicting recessions using machine learning methods.

My interest in this topic directly stems from my undergraduate background in Quantitative Economics and was reinforced by the machine learning course that I have taken as a graduate student in the Data Analysis and Visualization program. The machine learning course taught me how to use machine learning methods for classification and prediction with multiple features whereas in the past I have used deep learning algorithms for prediction, but they are more univariate in nature and this was a multivariate problem. The machine learning course gave me the foundations needed in order to conduct this project.

Related Work

In their paper Azhar Iqbal and Kyle Bowman (2018) used various machine learning models and statistical data mining methods to determine if this would improve recession prediction accuracy. The models selected for their analysis were gradient boosting, random forest, data mining (logit/probit), and benchmark-probit models. From their research it was determined that the machine learning models (gradient boosting and random forest models) provided more accurate results than the statistical data mining models.

Alireza Yazdani (2020) used a probit benchmark model, a generalized linear model with regularization, a support vector machine model with radial basis function, a random forest model, a random forest model and a XGBoost ensemble with regularization, and a single hidden layer feedforward neural network model to predict economic recessions. From his research it was determined that the random forest model outperformed the other machine learning models, which is consistent with Iqbal and Bowman (2018), with the highest accuracy, highest receiver operating characteristic (ROC) AUC, highest F-score, highest H-measure, highest Kolmogorov–Smirnov (KS) value, highest precision, and highest sensitivity.

With the work of Iqbal and Bowman (2018) and Yazdani (2020) as background, my capstone project has expanded on the use of machine learning models for economic recession prediction by using decision tree classifier, k nearest neighbor classifier, and support vector classifier machine learning classifier models rather than the non-classifier machine learning models gradient boosting, random forest, generalized linear, and feedforward neural network models used in the other research papers as a way to expand on their research. I also used the accuracy and precision-recall (PR) area under the curve (AUC) scores, which is the area under the generated precision-recall curve, as well as the ROC AUC scores, which is the area under the generated receiver operating characteristic curve, for my model evaluation rather than relying completely on the ROC curve and AUC scores as Iqbal and Bowman (2018) did in their research and used the PR AUC score as an alternate metric that Yazdani (2020) did not include in his model evaluations as a way to expand on his research.

Methodology

The data used in this project was a combination of economic and financial data. Since the features needed for the dataset were are not conveniently included in a downloadable dataset, each feature was downloaded separately and combined them together into one data frame in the Jupyter notebook used for the project running Python 3. The best place to pull each economic feature and the economic recession dates from is the FRED (Federal Reserve Economic Data from the Federal Reserve Bank of St. Louis) because I can be sure the data is reliable and timely. To retrieve the economic data, a Python package named Quandl was used, which also has the added bonus of automatically calculating selected transformations or different data frequencies. The financial feature, the S&P 500 index data, was downloaded from Yahoo! Finance so that the data could be transformed in Excel and then I imported the modified dataset as a CSV file (Microsoft Excel Comma Separated Values File) in the Jupyter notebook used for this project.

After creating the data frame of the features (including the transformed features) and the recession labels, a correlation heat map (Figure 1) was generated to determine which features will make up my final dataset. The goal was to choose features that were not highly correlated with each other, but were still correlated with the target feature.

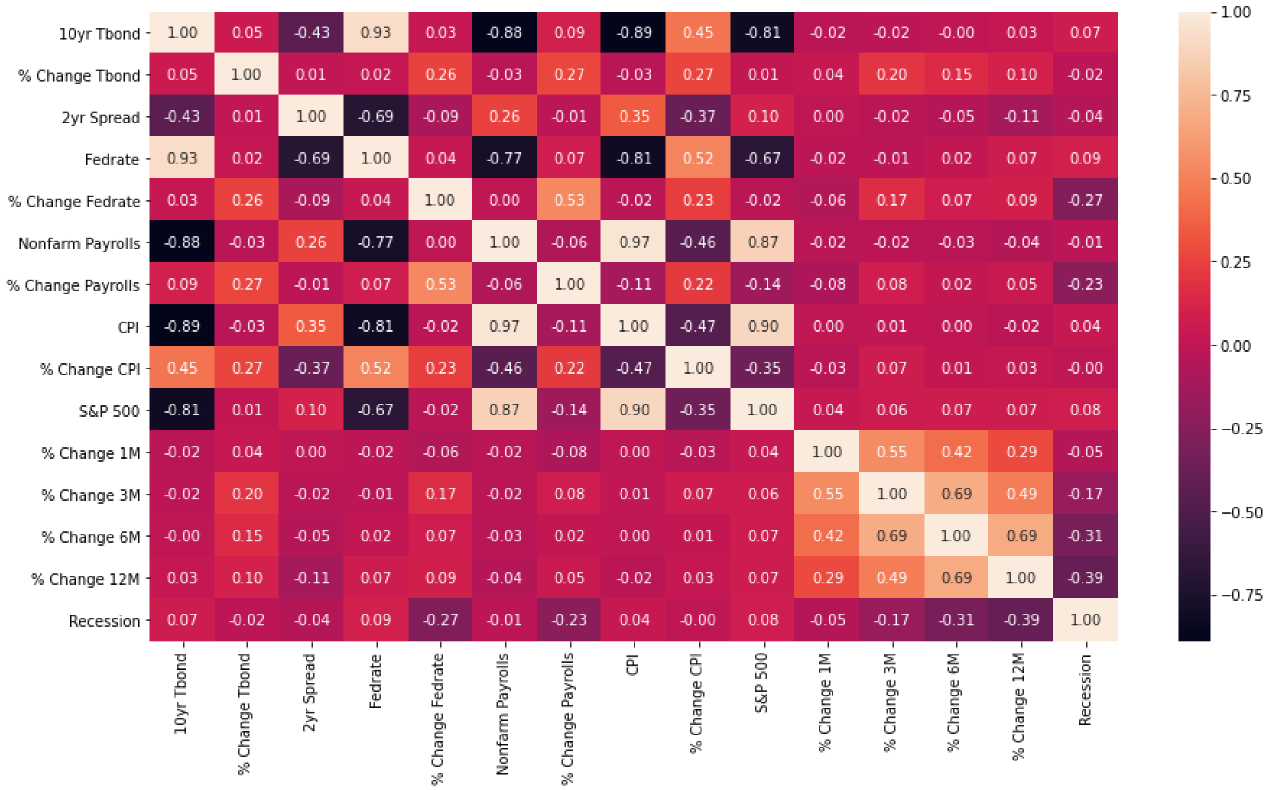

Figure 1: Correlation heat map for feature selection

For the final dataset the following 6 features were chosen; 10yr Tbond (10-Year Treasury Constant Maturity Rate), 2yr Spread (10-Year Treasury Constant Maturity Minus 2-Year Treasury Constant Maturity), % Change Fedrate (1-Month Percentage Change of the Effective Federal Funds Rate), % Change Payrolls (1-Month Percentage Change of All Employees, Total Nonfarm), CPI (Consumer Price Index for All Urban Consumers: All Items in U.S. City Average), % Change 12M (1-Year Percentage Change of the S&P 500 Index Price), which span from 6/1/1976 to 3/1/2021 for a total of 539 data points.

Before moving on some data exploration was needed to determine the extent of the class imbalance, which occurs when one label has more occurrences than another, as it is obvious that there were fewer recession labels than there were non-recession labels. The recession labels are only present in 12.987% of the data while 87.013% of the data is non-recession data. This is a clear case of class imbalance and must be kept in mind when interpreting the machine learning models.

Next, using the function train_test_split, which creates training and testing sets, the project data was split into training and testing datasets where 70% of the data will be used for training and 30% of the data will be used for testing. After this split, the training data contained data from 6/1/1976 to 10/1/2007 and the testing data contained data from 11/1/2007 to 3/1/2021. Since I will also be using cross validation with the function GridSearchCV, which iterates through all possible combinations for parameter values for an estimator and selects the best parameters, when tuning the models 5 walk-forward cross validation splits were generated using TimeSeriesSplit and have been visualized in Figure 2.

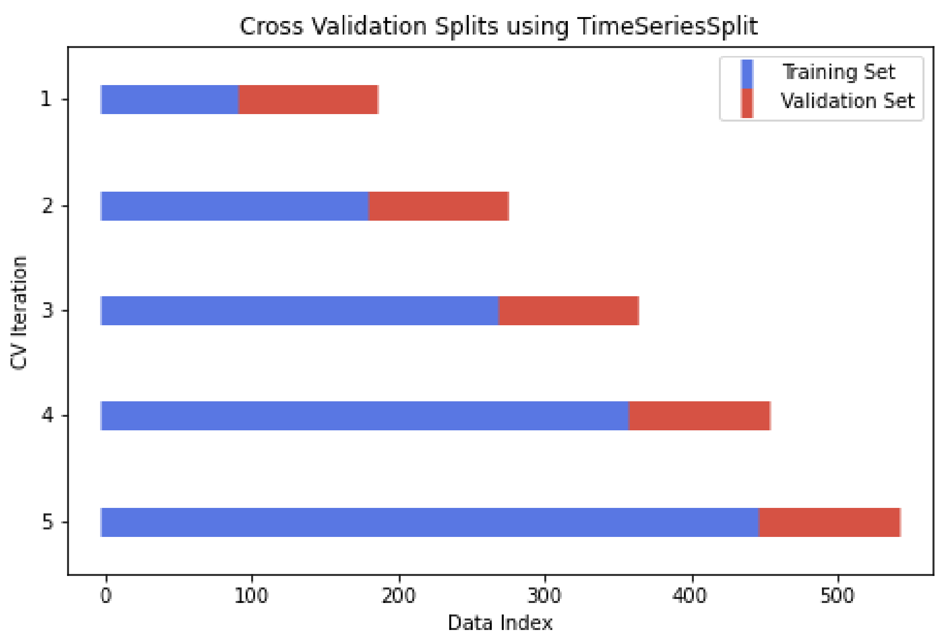

Figure 2: Walk-forward cross validation splits with TimeSeriesSplit

Now that we have our training and testing data we can move on to modeling. In this project I used logistic regression, decision tree classifier, k nearest neighbor classifier, and support vector classifier models to predict economic recessions. To do this I will be creating 4 versions of each model to select the optimal model. There will be 2 models trained on unscaled data and 2 models trained on scaled data which was scaled using StandardScaler. I will be training 2 models with default parameters (all support vector classifier models have the altered parameters probability=True and random_seed=1 to obtain the probabilities for use in the PR and ROC curves and to provide reproducible results) using the unscaled and scaled training data and 2 models using the unscaled and scaled training data with parameters that I will be tuning (Table 1) which will be selected by GridSearchCV and use TimeSeriesSplit as the cross validator. The possible parameter values are from options that the model parameters can accept and are not user defined values or functions.

Table 1: Models and parameters that will be tuned using GridSearchCV and TimeSeriesSplit

Model	Parameters to be Tuned with Possible Values
Logistic Regression	'C': np.logspace(-5, 2, 50) 'penalty': ['l1', 'l2']
Decision Tree Classifier	"max_features": range(1, 10) "min_samples_leaf": range(1, 10) "criterion": ["gini", "entropy"]
K Nearest Neighbor Classifier	'n_neighbors': range(1, 21)
Support Vector Classifier	'C': np.logspace(-5, 2, 50) 'kernel': ['linear', 'rbf', 'sigmoid'] 'gamma': ['scale', 'auto']

After training the default models and tuned models, of which the chosen parameters are shown in Table 2, I calculated each model's accuracy and created a classification report as well as a confusion matrix. I then calculated the PR AUC score and generated the PR curve, which is a plot of the recall (true positive rate) against the precision for different probability thresholds, as well as calculated the ROC AUC score and generated the ROC curve, which is a plot of the false positive rate against the true positive rate (recall), to get an idea of the model's performance.

Table 2: Chosen parameters for tuned models

Data Type	Model	Chosen Parameters
Unscaled	Logistic Regression	'C' = 1e-05

		'penalty' = 'l2'
	Decision Tree Classifier	"max_features" = 2 "min_samples_leaf" = 1 "criterion" = "gini”
	K Nearest Neighbor Classifier	'n_neighbors' = 10
	Support Vector Classifier	'C' = 0.5179474679231213 'kernel' = 'rbf' 'gamma' = 'auto'
Scaled	Logistic Regression	'C' = 1.0 'penalty' = 'l2'
	Decision Tree Classifier	"max_features" = 3 "min_samples_leaf" = 4 "criterion" = "gini”
	K Nearest Neighbor Classifier	'n_neighbors' = 3
	Support Vector Classifier	'C' = 10 'kernel' = 'rbf' 'gamma' = 'scale'

Results

Table 3: Metrics for models using default parameters and unscaled data

Model	Accuracy	ROC AUC	PR AUC
Logistic Regression	80.25%	0.69	0.64
Decision Tree Classifier	84.57%	0.61	0.37
K Nearest Neighbor Classifier	80.25%	0.50	0.20
Support Vector Classifier	80.25%	0.56	0.53

Table 4: Metrics for models using default parameters and scaled data

Model	Accuracy	ROC AUC	PR AUC
Logistic Regression	87.65%	0.71	0.71
Decision Tree Classifier	87.04%	0.67	0.47
K Nearest Neighbor Classifier	87.65%	0.74	0.58
Support Vector Classifier	81.48%	0.93	0.79

Table 5: Metrics for models tuned with GridSearchCV and using unscaled data

Model	Accuracy	ROC AUC	PR AUC
Logistic Regression	91.29%	0.55	0.45
Decision Tree Classifier	95.16%	0.77	0.53
K Nearest Neighbor Classifier	91.29%	0.50	0.20
Support Vector Classifier	92.90%	0.37	0.23

Table 6: Metrics for models tuned with GridSearchCV and using scaled data

Model	Accuracy	ROC AUC	PR AUC
Logistic Regression	87.65%	0.71	0.71
Decision Tree Classifier	88.27%	0.77	0.53
K Nearest Neighbor Classifier	87.65%	0.72	0.54
Support Vector Classifier	83.95%	0.94	0.83

Tables 3 to 6 summarize the metrics of the 16 models. One thing about these results that was not expected was the model parameters that were tuned and trained using the unscaled data had the highest accuracy when it was expected that the model parameters that were tuned and trained using the scaled data would have the highest accuracy whereas these models had the second highest accuracy.

Basing model performance on accuracy, the optimal model is the tuned Decision Tree Classifier model trained on unscaled data with an accuracy of 95.16%. Basing model performance on the ROC AUC score, the optimal model is the tuned Support Vector Classifier model trained on scaled data with a score of 0.94. Basing model performance on the PR AUC score, the optimal model is the tuned Support Vector Classifier model trained on scaled data with a score of 0.83.

Due to the issue of the class imbalance and the fact that I am considering each class to be equally important the best metric to base model performance on is the PR AUC score, which means the optimal model to predict economic recessions is the tuned Support Vector Classifier model trained on scaled data with a PR AUC score of 0.83 (PR curve shown in Figure 3). The second best model is the default Support Vector Classifier model trained on scaled data with a PR AUC score of 0.79 (PR curve shown in Figure 4) and the third best model to round out the top three is the default Logistic Regression model trained on scaled data, as the tuned model had the same parameters, with a PR AUC score of 0.71 (PR curve shown in Figure 5).

Given the fact that the PR and ROC AUC scores increase when training on scaled data it is safe to say that using scaled training data improves the performance of classifier models even when there is a clear class imbalance. It is also safe to say that the optimal model to predict economic recessions is the Support Vector Classifier model.

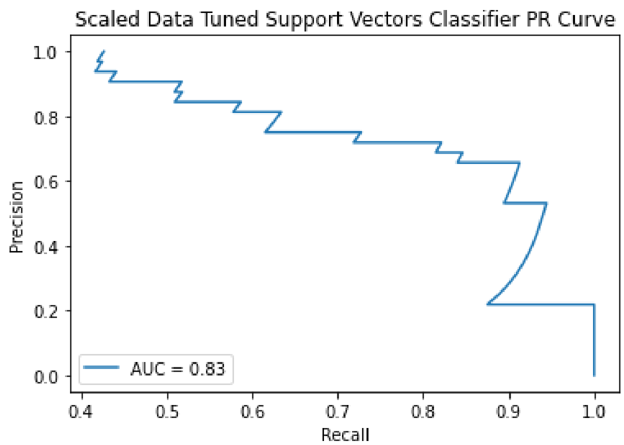

Figure 3: PR Curve for the tuned Support Vector Classifier model trained on scaled data

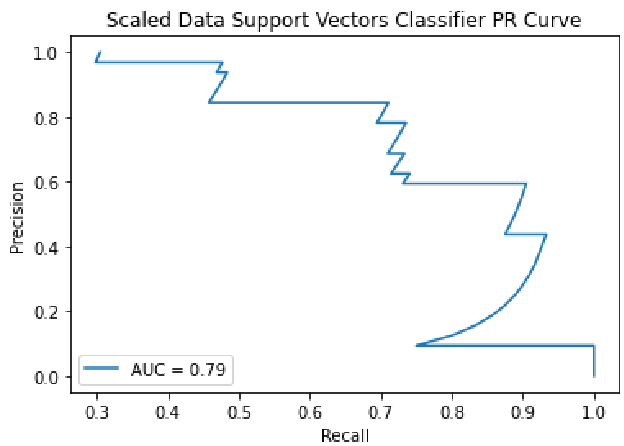

Figure 4: PR Curve for the default Support Vector Classifier model trained on scaled data

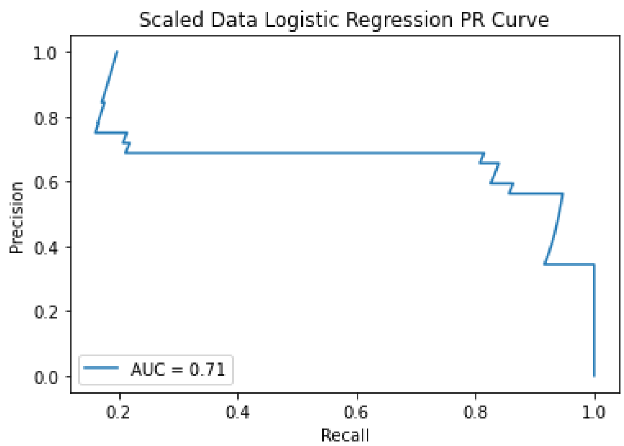

Figure 5: PR Curve for the default Logistic Regression model trained on scaled data

Conclusion

The purpose of this project was to use machine learning techniques to predict economic recessions. To do this I generated training and testing data from a compiled set of economic and financial features, fit the training data using logistic regression, decision tree classifier, k nearest neighbor classifier, and support vector classifier models, and used the testing data to make our predictions. After generating the model predictions I evaluated the model's performance by also generating the accuracy, PR AUC score, PR curve, ROC AUC score, and ROC curve. Due to the class imbalance in the target feature, the PR AUC score was the chosen metric to choose the best performing model, which turns out to be the tuned Support Vector Classifier model trained on scaled data with a PR AUC score of 0.83 (PR curve shown in Figure 3).

As the random forest model was the optimal model chosen by Iqbal and Bowman (2018) as well as Yazdani (2020) I would have expected the optimal model chosen by this project to be the decision tree classifier model, even if it would not have performed as well as those in the mentioned research papers, because random forest models combines the output of multiple randomly created decision trees to generate the final output, but this was not the case. In Yazdani's (2020) research the support vector model was the second best model whereas in this project the support vector classifier model was the optimal model, which could suggest that classifier models have a different effect than their non-classifier counterparts on predicting economic recessions.

It is important to note that it is impossible to predict economic recessions with 100% accuracy. There is always a chance that external factors could cause economic and financial confidence to fall, which can in turn cause the risk of an economic recession to increase. One such case is the COVID-19 pandemic that we have found ourselves in since February 2020 and remains as of the date of this paper. When the pandemic hit and cities started to shutdown mass panic and economic slowdown caused economic and financial confidence to fall and pushed us into a recession in March 2020 and the economy has yet to rebound and pull us out of the recession. The economic slowdown was not something that could be predicted although the effects could be felt.

If I were to expand on this project in the future I would choose a different set of economic and financial features as some of the economic features I included in this project only go back to 1976, which is why my data starts from this year. I believe if economic features that go back further were included then more recession and non-recession dates would be able to be included as well and with more data the models could learn and predict better improving the PR AUC score and the ROC AUC score.

Appendix

Precision-Recall Curves for All Other Models

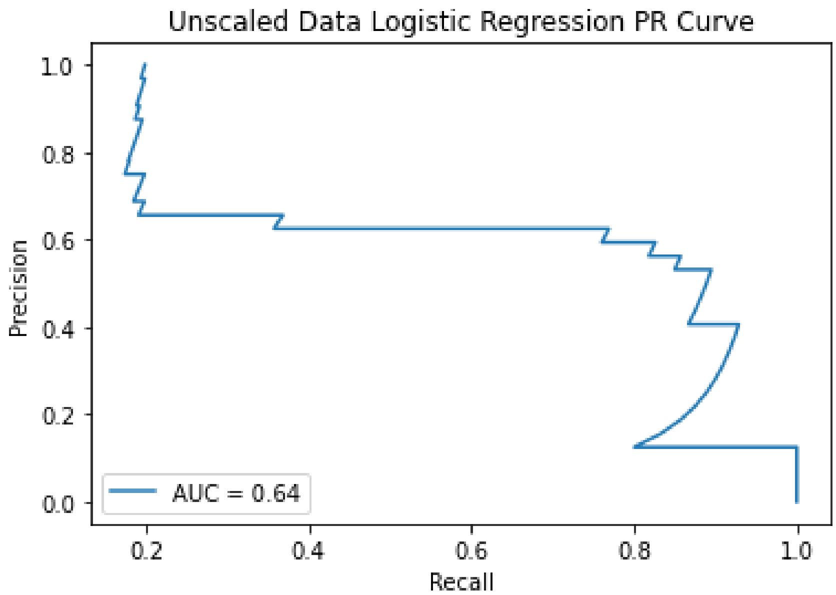

Figure 6: PR Curve for the default Logistic Regression model trained on unscaled data

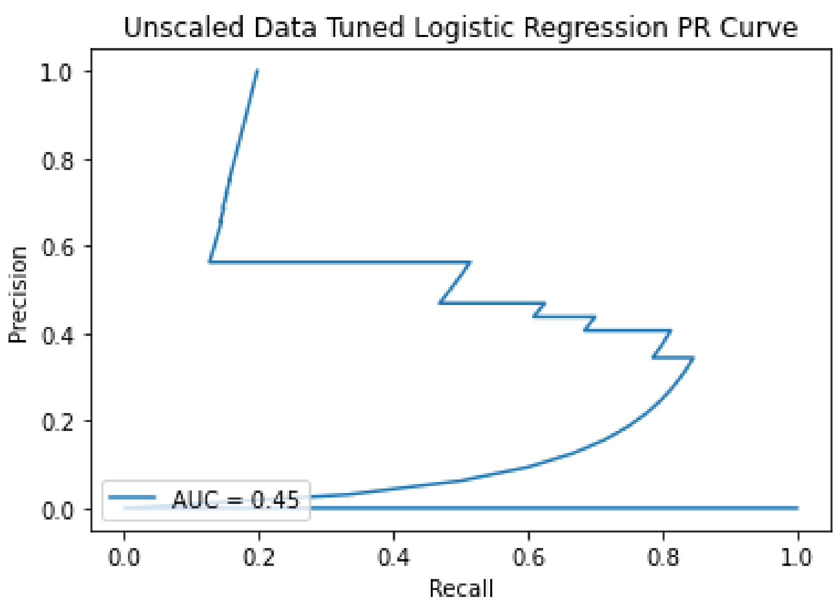

Figure 7: PR Curve for the tuned Logistic Regression model trained on unscaled data

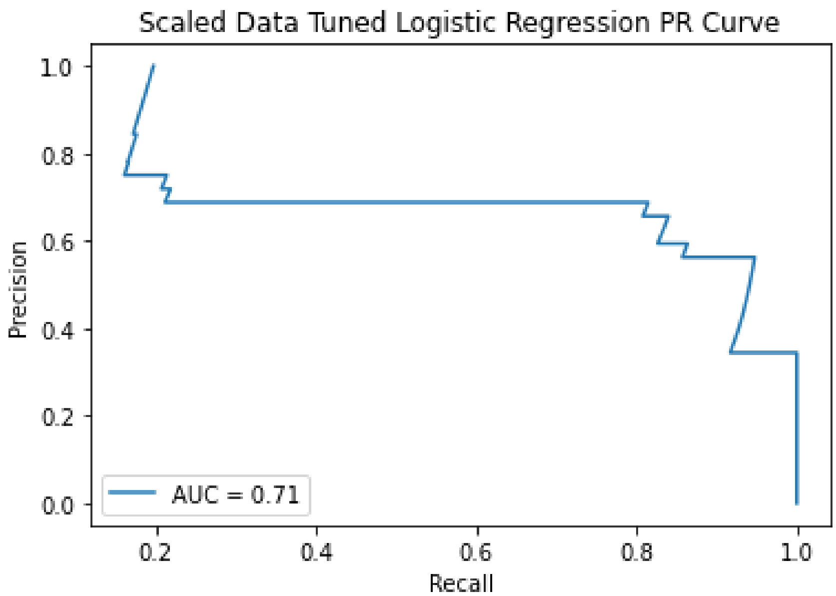

Figure 8: PR Curve for the tuned Logistic Regression model trained on scaled data

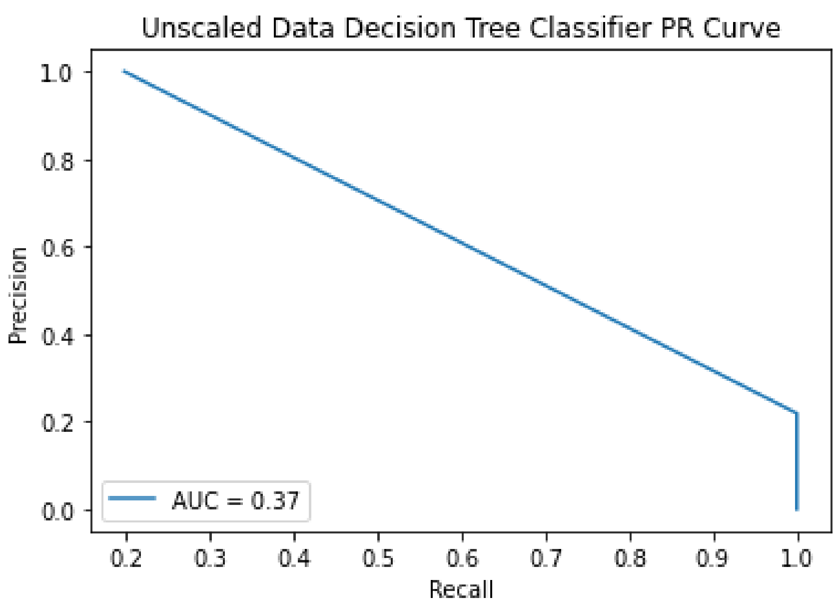

Figure 9: PR Curve for the default Decision Tree Classifier model trained on unscaled data

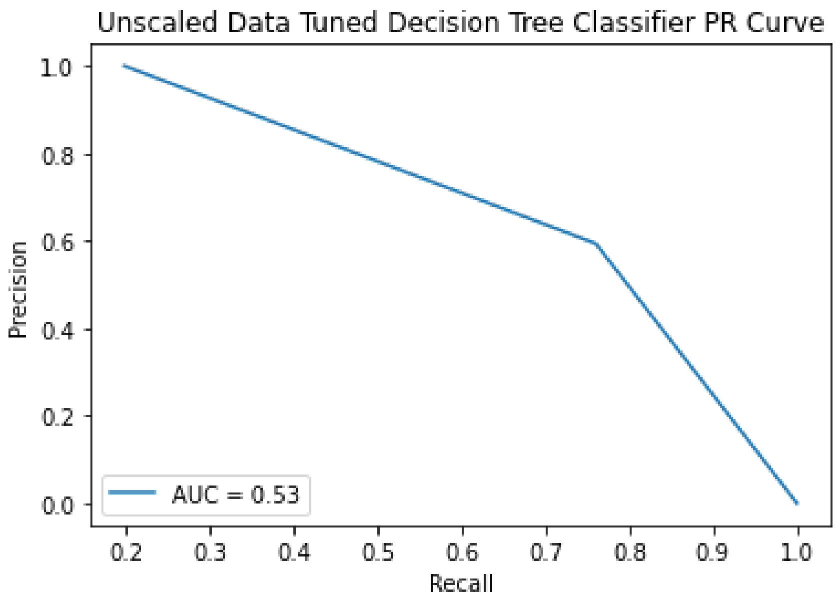

Figure 10: PR Curve for the tuned Decision Tree Classifier model trained on unscaled data

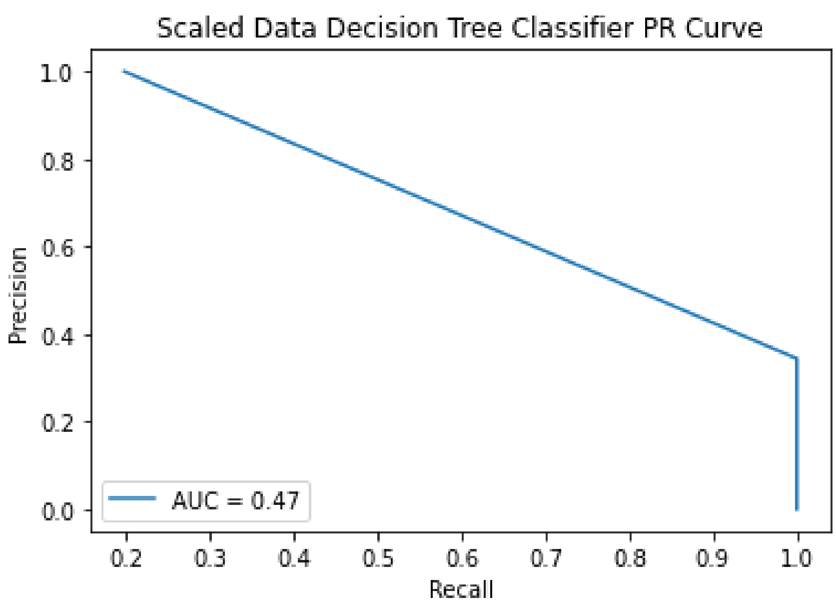

Figure 11: PR Curve for the default Decision Tree Classifier model trained on scaled data

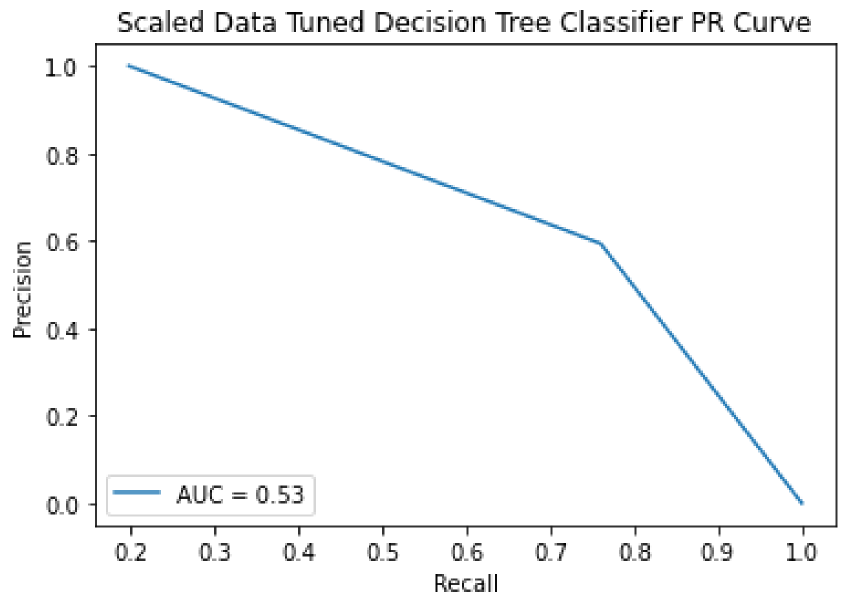

Figure 12: PR Curve for the tuned Decision Tree Classifier model trained on scaled data

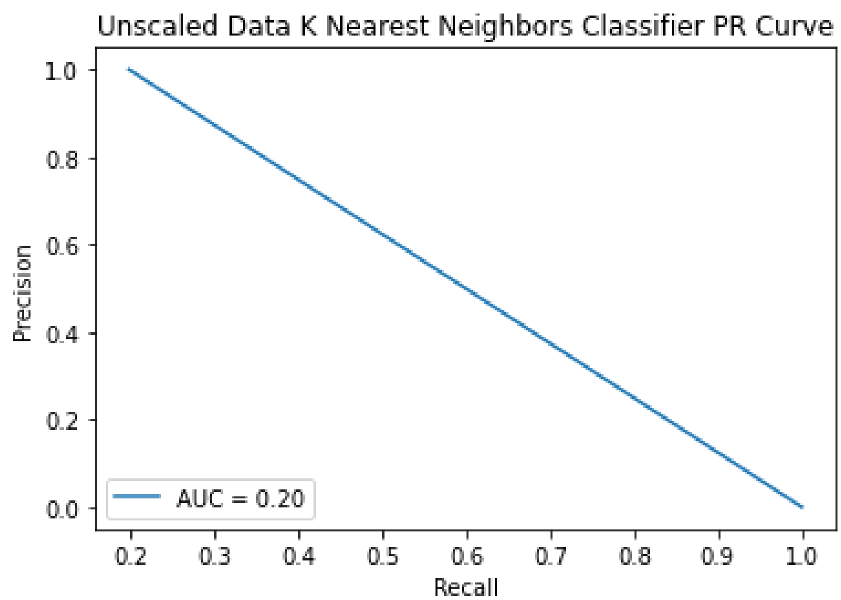

Figure 13: PR Curve for the default K Nearest Neighbors Classifier model trained on unscaled data

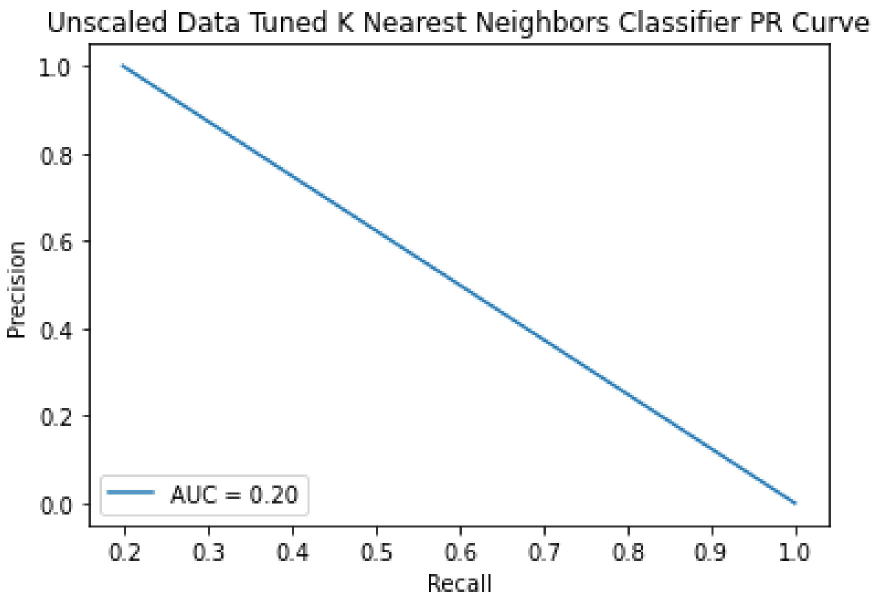

Figure 14: PR Curve for the tuned K Nearest Neighbors Classifier model trained on unscaled data

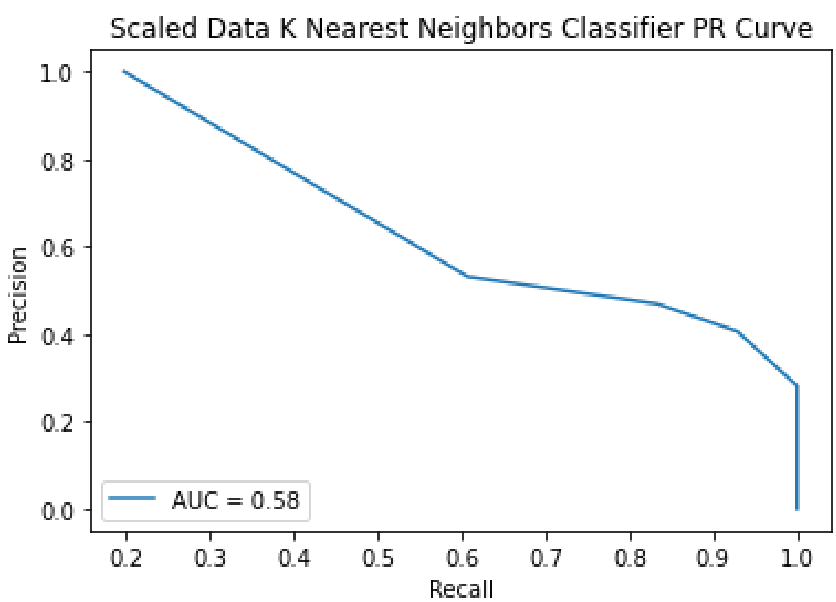

Figure 15: PR Curve for the default K Nearest Neighbors Classifier model trained on scaled data

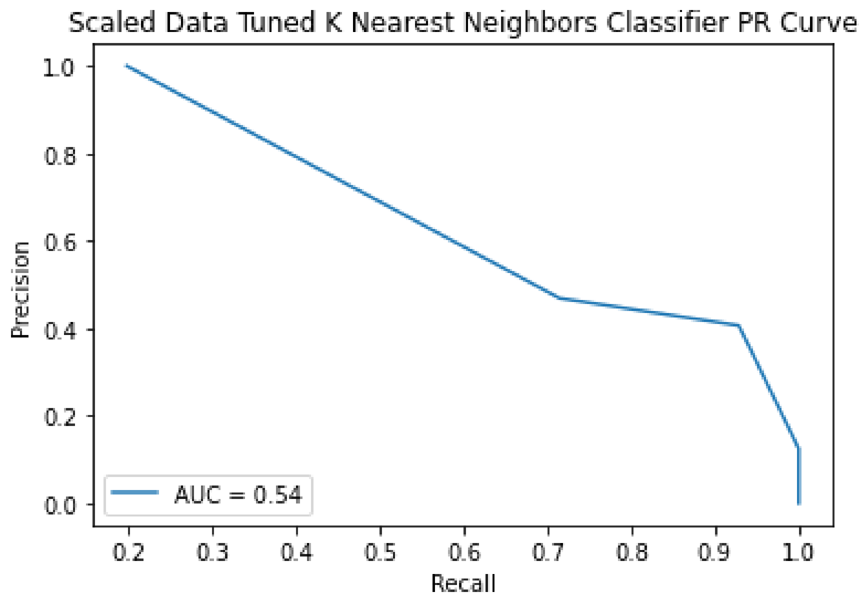

Figure 16: PR Curve for the tuned K Nearest Neighbors Classifier model trained on scaled data

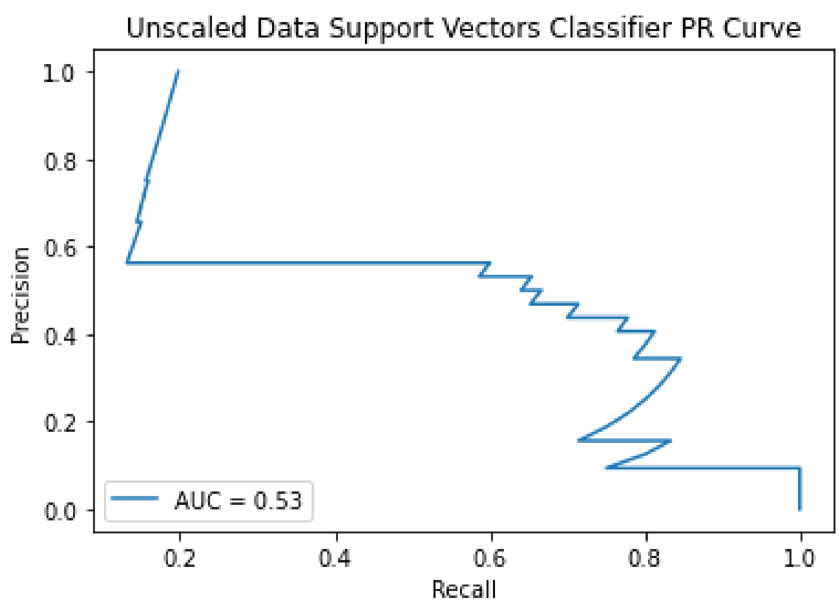

Figure 17: PR Curve for the default Support Vector Classifier model trained on unscaled data

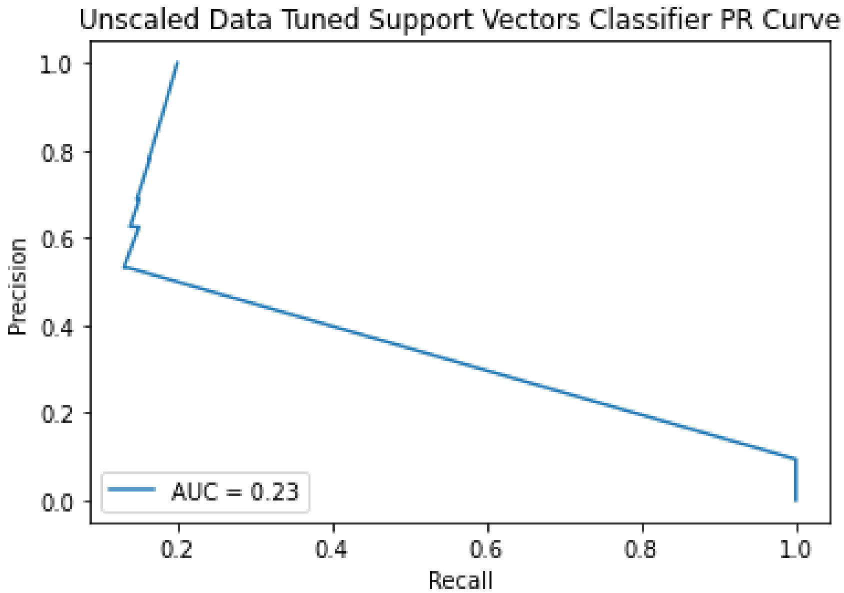

Figure 18: PR Curve for the tuned Support Vector Classifier model trained on unscaled data

List of Variables

Table 7: List of variables used in the capstone project code

Variable Name	Description
tbond_10yr	10-Year Treasury Constant Maturity Rate
tbond_10yr_pchange	1-Month Percentage Change of the 10-Year Treasury Constant Maturity Rate
spread_2yr	10-Year Treasury Constant Maturity Minus 2-Year Treasury Constant Maturity
fedrate	Effective Federal Funds Rate
fedrate_pchange	1-Month Percentage Change of the Effective Federal Funds Rate
nonfarm_payrolls	All Employees, Total Nonfarm
nonfarm_payrolls_pchange	1-Month Percentage Change of All Employees, Total Nonfarm
cpi	Consumer Price Index for All Urban Consumers: All Items in U.S. City Average
cpi_pchange	1-Month Percentage Change of the Consumer Price Index for All Urban Consumers: All Items in U.S. City Average
recessions	NBER economic recession dates
snp_500	S&P 500 Index Price
df	Pandas Dataframe containing all features
data	Pandas Dataframe containing only selected features
X	Features to model
y	Target variable
X_train	Training features
y_train	Training target variable
X_test	Testing features
y_test	Testing target variable
tscv	TimeSeriesSplit cross validator
logreg	Instance of the Logistic Regression model
logpredict	Prediction values of default Logistic Regression model trained with unscaled data
confusion_matrix	Creates confusion matrix from y_test and model prediction values
logreg_pr_auc	PR AUC of the Logistic Regression model using unscaled data
fpr	False positive rate for the PR and ROC curves
tpr	True positive rate for the PR and ROC curves
threshold	Threshold values for the PR and ROC curves
logreg_roc_auc	ROC AUC of the Logistic Regression model using unscaled data

c_space	Values for 'C' parameter selection
param_grid	Dictionary of model parameters and potential values
logreg_cv	Model of best parameters for Logistic Regression model using unscaled data
y_pred_prob	Model prediction probabilities using unscaled data
steps	List of tuples ('name', function) for use in pipeline
pipeline	Pipeline for steps
logpredict_scaled	Predictions for Logistic Regression using pipeline to scale data
logreg_scaled_pr_auc	PR AUC of the Logistic Regression model using scaled data
y_pred_prob_scaled	Model prediction probabilities using scaled data
logreg_scaled_roc_auc	ROC AUC of the Logistic Regression model using scaled data
parameters	Dictionary of model parameters and potential values
logreg_scaled_cv	Model of best parameters for Logistic Regression model using scaled data
tree	Instance of the Decision Tree Classifier model
treepredict	Prediction values of default Decision Tree Classifier model trained with unscaled data
tree_pr_auc	PR AUC of the Decision Tree Classifier model using unscaled data
tree_roc_auc	ROC AUC of the Decision Tree Classifier model using unscaled data
param_dist	Dictionary of model parameters and potential values
tree_cv	Model of best parameters for Decision Tree Classifier model using unscaled data
treepredict_scaled	Predictions for Decision Tree Classifier using pipeline to scale data
tree_scaled_pr_auc	PR AUC of the Decision Tree Classifier model using scaled data
tree_scaled_roc_auc	ROC AUC of the Decision Tree Classifier model using scaled data
parameter	Dictionary of model parameters and potential values
tree_scaled_cv	Model of best parameters for Decision Tree Classifier model using scaled data
knc	Instance of the K Neighbors Classifier model
kncpredict	Prediction values of default K Neighbors Classifier model trained with unscaled data

knc_pr_auc	PR AUC of the K Neighbors Classifier model using unscaled data
knc_roc_auc	ROC AUC of the K Neighbors Classifier model using unscaled data
k	Range of values for k from 1 to 20
knc_cv	Model of best parameters for K Neighbors Classifier model using unscaled data
kncpredict_scaled	Predictions for K Neighbors Classifier using pipeline to scale data
knc_scaled_pr_auc	PR AUC of the K Neighbors Classifier model using scaled data
knc_scaled_roc_auc	ROC AUC of the K Neighbors Classifier model using scaled data
knc_scaled_cv	Model of best parameters for K Neighbors Classifier model using scaled data
svc	Instance of the Support Vector Classifier model
svcpredict	Prediction values of default Support Vector Classifier model trained with unscaled data
svc_pr_auc	PR AUC of the Support Vector Classifier model using unscaled data
svc_roc_auc	ROC AUC of the Support Vector Classifier model using unscaled data
svc_cv	Model of best parameters for Support Vector Classifier model using unscaled data
svcpredict_scaled	Predictions for Support Vector Classifier using pipeline to scale data
svc_scaled_pr_auc	PR AUC of the Support Vector Classifier model using scaled data
svc_scaled_roc_auc	ROC AUC of the Support Vector Classifier model using scaled data
svc_scaled_cv	Model of best parameters for Support Vector Classifier model using scaled data

Bibliography

Berge, Travis. “Predicting Recessions with Leading Indicators: Model Averaging and Selection Over the Business Cycle.” The Federal Reserve Bank of Kansas City, Jan. 2014.

Brownlee, Jason. “ROC Curves and Precision-Recall Curves for Imbalanced Classification.” *Machine Learning Mastery*, Machine Learning Mastery, 15 Sept. 2020, machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-imbalanced-classification/.

Caruana, Rich, and Alexandru Niculescu-Mizil. "An Empirical Comparison of Supervised Learning Algorithms." *Proceedings of the 23rd International Conference on Machine Learning - ICML '06* (2006). Print.

Chen, J. et al. “Off to the Races: A Comparison of Machine Learning and Alternative Data for Predicting Economic Indicators.” (2019).

Coulombe, P. et al. “How is Machine Learning Useful for Macroeconomic Forecasting.” (2019).

Döring, Matthias. “Interpreting ROC Curves, Precision-Recall Curves, and AUCs.” *Data Science Blog: Understand. Implement. Succeed.*, Data Science Blog: Understand. Implement. Succeed., 16 June 2020, www.datascienceblog.net/post/machine-learning/interpreting-roc-curves-auc/.

Hyndman, Rob J., and George Athanasopoulos. *Forecasting: Principles and Practice;*. Lexington, KY: Otexts, 2018. Print.

Iqbal, Azhar, and Kyle Bowman. “Can Machine Learning Improve Recession Prediction Accuracy?” *Journal of Applied Economics and Business*, vol. 6, no. 4, Dec. 2018, pp. 16–34.

Malik, Lukas. “Replicating ‘Machine Learning Prediction of Recessions’ (Yazdani, 2020).”

Medium, Towards Data Science, 9 Sept. 2020, towardsdatascience.com/replicating-machine-learning-prediction-of-recessions-yazdani-2020-9b9500131c71.

Masini, Ricardo P., et al. “Machine Learning Advances for Time Series Forecasting.” *ArXiv.org*, ArXiv, 18 Jan. 2021, arxiv.org/abs/2012.12802.

Mullainathan, Sendhil, and Jann Spiess. “Machine Learning: An Applied Econometric Approach.” *Journal of Economic Perspectives*, vol. 31, no. 2, 2017, pp. 87–106., doi:10.1257/jep.31.2.87.

Nyman, Rickard, and Paul Ormerod. “Predicting Economic Recessions Using Machine Learning Algorithms.” *ArXiv.org*, ArXiv: General Finance, 3 Jan. 2017, arxiv.org/abs/1701.01428.

Swanson, Norman R., and Halbert White. “A Model Selection Approach to Real-Time Macroeconomic Forecasting Using Linear Models and Artificial Neural Networks.” *Review of Economics and Statistics*, vol. 79, no. 4, 1997, pp. 540–550., doi:10.1162/003465397557123.

Wei, Xiaohui, et al. “Exploring the Financial Indicators to Improve the Pattern Recognition of Economic Data Based on Machine Learning.” *Neural Computing and Applications*, vol. 33, no. 2, 2020, pp. 723–737., doi:10.1007/s00521-020-05094-0.

Yazdani, Alireza. “Machine Learning Prediction of Recessions: An Imbalanced Classification Approach.” *The Journal of Financial Data Science*, vol. 2, no. 4, 2020, pp. 21–32., doi:10.3905/jfds.2020.1.040.

Zhang, Terrence. “Recession Prediction Using Machine Learning.” *Medium*, Towards Data Science, 11 Jan. 2021, towardsdatascience.com/recession-prediction-using-machine-learning-de6eee16ca94.