Enterprise Endgame Evaluator

Nick Pysklywec
Faculty of Engineering
Western University
London, Ontario
npysklyw@uwo.ca

Hisham Azzi
Faculty of Engineering
Western University
London, Ontario
hazzi@uwo.ca

Varun Noah Chauhan
Faculty of Engineering
Western University
London, Ontario
vchauha4@uwo.ca

Stella Lee
Faculty of Engineering
Western University
London, Ontario
jlee3465@uwo.ca

***Abstract*—The financial world is turbulent, many companies grow, while others lose all value, and fall to bankruptcy. The recent explosion of financial data being available, has led to these trends to be capable of being predicted. The financial reports can be used with machine learning to determine the fortunes of companies. Tools for machine learning will be used to create various models and make comparisons. The following models are to be created and further evaluated in performance (more relevant models may also be used if more efficient): Random forest Models, kNN, Logistic Regression, XG boost Decision Tree, and Support Vector Machines. In addition, we will also be conducting feature analysis to determine the best indicators of bankruptcy from the 96 features in the dataset.**

I. Introduction

Bankruptcy prediction is the problem of identifying financial struggles in businesses for the purposes of effective risk management and decision-making, allowing investors and stakeholders to take actions early to minimize economic losses. Bankruptcy prediction has been studied since the 1930s, meaning it has been an essential aspect for many companies throughout the years [1].

Some traditional methods of predicting bankruptcy include financial ratio analysis, where the ratio between debt and equity is used to assess a company's financial leverage. It is calculated by dividing a company's total liabilities by its shareholder equity, and a result of two or higher is considered to be an indicator that a company might be leading to bankruptcy [2, 3]. However, this method provides results with limited accuracy and reliability, with only 57.14% overall accuracy for three-years prior bankruptcy [4].

Consequently, in recent years, machine learning models have become a promising tool for predicting bankruptcy by applying them to several financial data. In this paper, we experimented with five different machine learning models. Our goal is to analyze their accuracy and efficiency to identify the most effective predictor of bankruptcy, in order to help investors and businesses to make better decisions in managing financial risk. We also aim to determine which variables are the best to use when predicting bankruptcy with machine learning models.

II. Related Work

A literature review was conducted to gain intuition into past work done into bankruptcy and financial forecasting. It was important to see the results produced to know which models to consider using for this report, alongside the methods that may be used in data preprocessing.

A comprehensive and well regarded experiment into firm bankruptcy was done recently with a basis on the COVID-19 pandemic [12]. The goal was to produce a total of nine models(random forest, SVM, XGBoost) that would detect bankruptcy in the next periods of 30, 90 and 180 days. These models were trained on an artificially balanced dataset of a 20000 sample 57 feature(financial ratios) dataset. Random Forest and XGboost models proved most effective in this task with test set accuracy in the six models from 98-99%, whereas the SVM models proved less effective relative, but still good with test set accuracy ranging from 82-90

Another paper from 2018 focusing on the Random Forest model for our application was analyzed [13]. Bank financial statements from 65 banks were used, each with 20 financial ratios. Ultimately, the paper demonstrated the success of Random Forest in this application, with training and testing accuracy 100% and 94%. Finally a paper that compared KNN, Neural Network, and Random Forest models was analyzed [14]. The data used was a balanced UCI machine learning dataset "Bankruptcy Rates", that consisted of 250 samples. Random Forest produced an accuracy of 0.99%, KNN an accuracy of 0.985%(5 neighbors), and Neural Network accuracy of (0.995% with 0.3 dropout)/ (0.984% with no dropout). Looking through these three papers provided the most background information to allow the group to make more educated decisions going forward in our research process.

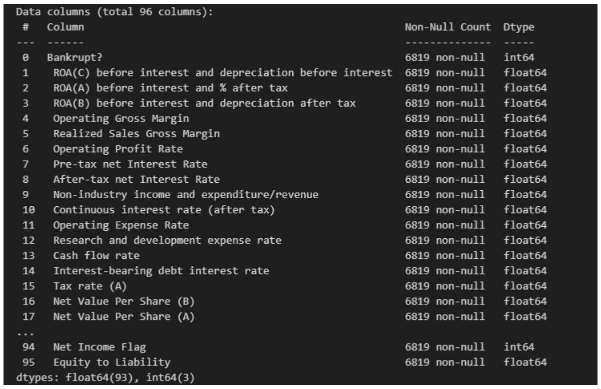
III. Procedure

The purpose of this experiment is to compare the accuracy of different machine learning models in predicting bankruptcy when given various financial variables. The approach taken to this experiment is the following:

A) Dataset Analysis
B) Dataset Preprocessing
C) Determine Feature Importance
D) Model Training
E) Model Testing
F) Result Analysis

A. Dataset Analysis

For this experiment, we wanted to use a dataset with a broad range of financial metrics, as one of our goals is to determine which variables are the most predictive of bankruptcy when using machine learning models. The dataset chosen for this experiment is from the Taiwan Economic Journal, with 95 financial features for 6819 companies where bankruptcy was determined based on the business regulations of the Taiwan Stock Exchange for the years between 1999 and 2009. The first column of the metadata file, data.csv, is called “Bankrupt?” and its value is 0 if the company is considered bankrupt and 1 if not. The feature dictionary is presented in Figure 1.



```
Data columns (total 96 columns):
 #   Column                                                   Non-Null Count  Dtype
---  ------                                                   --------------  -----
 0   Bankrupt?                                                6819 non-null   int64
 1    ROA(C) before interest and depreciation before interest 6819 non-null   float64
 2    ROA(A) before interest and % after tax                  6819 non-null   float64
 3    ROA(B) before interest and depreciation after tax       6819 non-null   float64
 4    Operating Gross Margin                                  6819 non-null   float64
 5    Realized Sales Gross Margin                             6819 non-null   float64
 6    Operating Profit Rate                                   6819 non-null   float64
 7    Pre-tax net Interest Rate                               6819 non-null   float64
 8    After-tax net Interest Rate                             6819 non-null   float64
 9    Non-industry income and expenditure/revenue             6819 non-null   float64
 10   Continuous interest rate (after tax)                    6819 non-null   float64
 11   Operating Expense Rate                                  6819 non-null   float64
 12   Research and development expense rate                   6819 non-null   float64
 13   Cash flow rate                                          6819 non-null   float64
 14   Interest-bearing debt interest rate                     6819 non-null   float64
 15   Tax rate (A)                                            6819 non-null   float64
 16   Net Value Per Share (B)                                 6819 non-null   float64
 17   Net Value Per Share (A)                                 6819 non-null   float64
...
 94   Net Income Flag                                         6819 non-null   int64
 95   Equity to Liability                                     6819 non-null   float64
dtypes: float64(93), int64(3)
```

Fig. 1. Feature Dictionary Sample

B. Dataset Preprocessing

We first identified which columns were not scaled, and found that 24 features were not. So our first step was using a StandardScaler to normalize our data, where it transforms the data so that the input features have zero mean and variance so that they all have the same range of values which increases our models’ performances. Therefore, by using StandardScaler we ensure that all the features are equally considered. The data contained no empty cells or null values and no duplicate data. In analyzing the data we identified that the target value, bankruptcy indicator, had a significant imbalance in the difference between bankrupt and non-bankrupt companies. Infact, we observed that 96.8% of companies are non-bankrupt and 3.2% are bankrupt. We concluded that we must balance the data in order to build an ideal model capable of learning between the two types of companies.

To tackle this issue we adopted SMOTE, Synthetic Minority Over-sampling Technique, which is a technique that helps balance unbalanced datasets by creating synthetic examples of the smaller class. It works by selecting a random sample from the minority class and finding its k-nearest neighbors. The synthetic samples are then generated by combining the selected sample and its k-nearest neighbors to ensure it fits the data [6]. We chose this method instead of reducing the dataset or introducing weights as the dataset would suffer from undersampling and thus affect the model’s accuracy. The sample collection shape prior to SMOTE was (6819, 95) .Below we detail the effects of using SMOTE on the dataset. As seen here, there is substantial difference in the number of bankruptcy encounters compared to non-bankrupt ones.

Fig. 2. ”Distribution Prior to SMOTE”

Class	Distribution
0	0.968
1	0.032

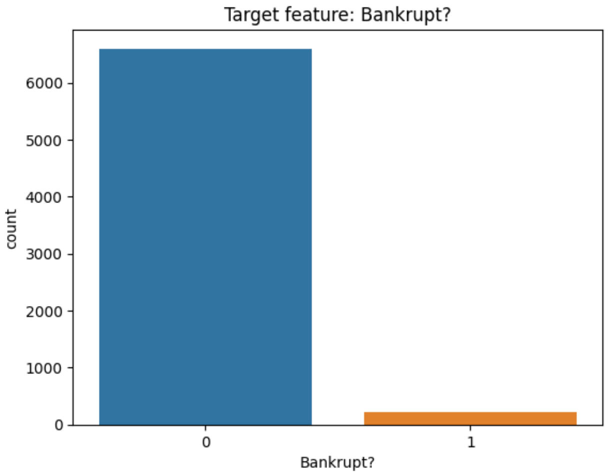


Fig. 3. Bar chart to visualize distribution

After applying SMOTE the issue is resolved with 50% occurrences of each and thus balancing the dataset:

Fig. 4. ”Distribution After SMOTE”

Class	Distribution
0	0.50
1	0.50

The sample collection shape is changed to (13198, 95) after appyling SMOTE.

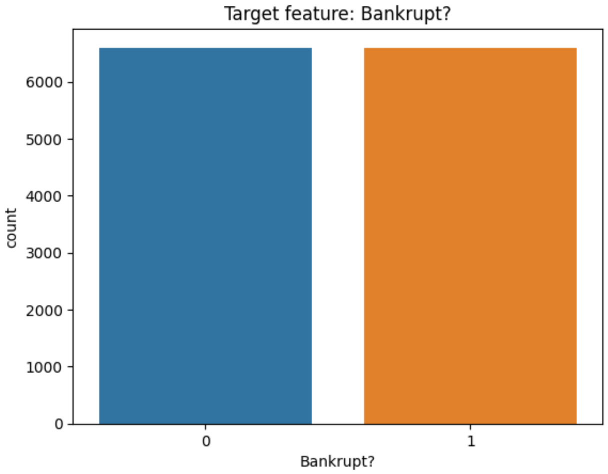


Fig. 5. Bar chart to visualize distribution after SMOTE

C. Feature Importance

Considering that the data has 96 features, which is a large number of features, we decided to observe which features had the most impact on the result in an effort to reduce the dimensionality of the data by eliminating the redundant features and resulting in a more accurate model [7]. We first decided to observe the relationship between features and how it affected whether a company is going bankrupt or not. As seen in Figure 6, we plotted the relationship between bankrupt events and Net income to total assets, where we observe a substantial difference in results.

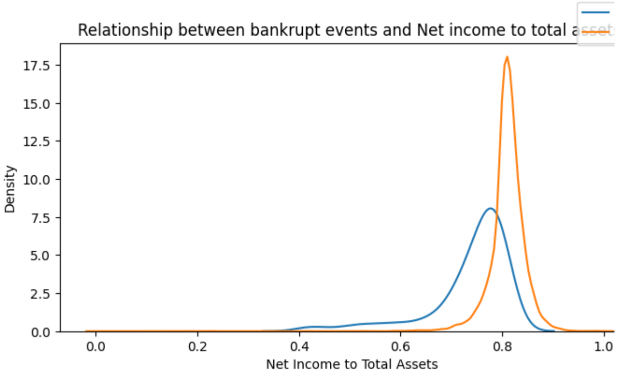


Fig. 6. Bankrupt Events vs Net Income to Total Assets

In contrast, Figure 7, shows the relationship b bankrupt events and Total Asset Growth Rate is less dep as they both have their peaks at similar values.

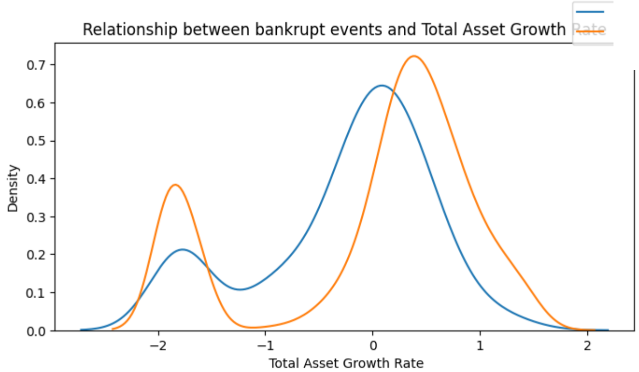


Fig. 7. Bankrupt Events vs Total Asset Growth

This gave us confidence that some features had more effect than others in determining the outcome.

To select all the features that will be used in our model, we decided to implement ExtraTreesClassifier to identify the most relevant features. It's a machine learning algorithm that essentially works by constructing a large number of decision trees from a random subset of features and then combining them to make a prediction. It repeats this process over and over again, each time randomizing the subset of features and observing the change in accuracy to identify which features have the most impact [8].

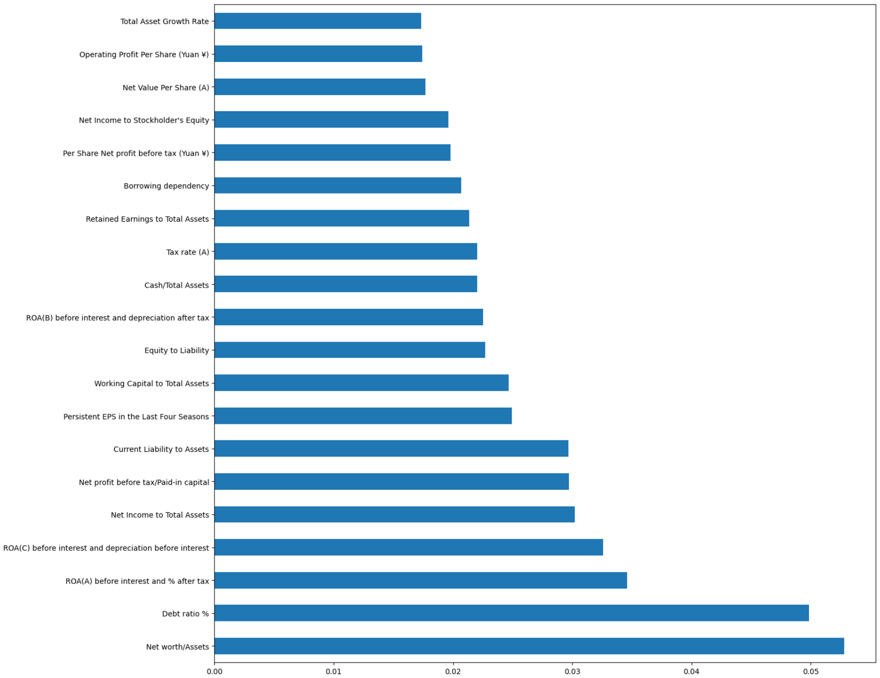


Fig. 8. Top 20 Features by Importance Score

We used the feature_importances_ attribute of the ExtraTreesClassifier, to sort each feature by importance score as seen in Figure 8 and based on trial and error decided that the top 20 features gave us the best results when training the model. Figure 9 shows a plotted heatmap of the selected features that highlights that features used in the model are highly correlated to each to the target variable.

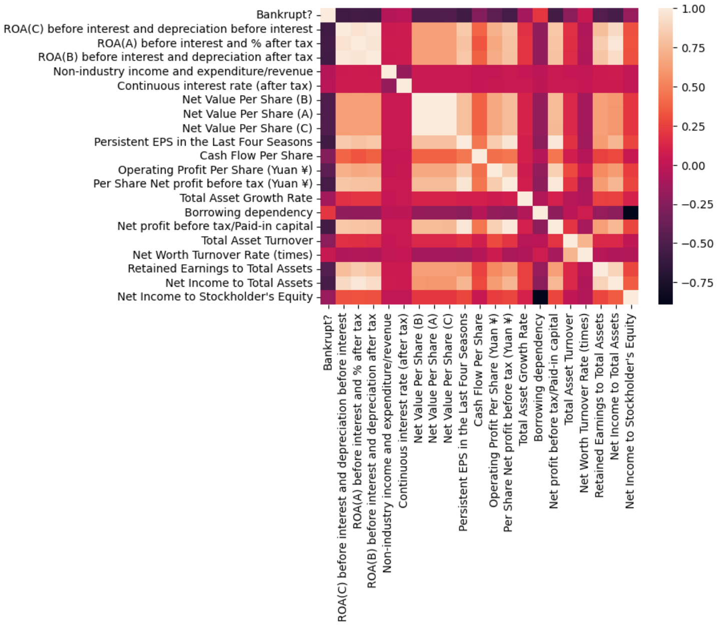


Fig. 9. Heatmap of Top 20 Features by Correlation

In conclusion, by using the ExtraTreesClassifier for feature selection we improved our model's performance by decreasing complexity and reducing overfitting.

D. Models

1) Support Vector Machines: Support Vector Machines (SVMs) are used for classification, regression, and outlier detection via supervised learning methods [5]. The most optimal decision boundary, also called the hyperplane, that puts the variables into different classes can be found through SVMs.

The SVM model for this experiment is set up using the ”SVC” function from the scikit-learn library. The ”gamma” parameter is configured to ”auto” and ”probability” is set to ”True.” The hyperparameters for the model include “C,” which determines the size of the hyperplane's margin. This is set to [1, 5, 10, 20]. The “kernel” hyperparameter allows us to choose the kernel functions to be used for the model. This is set to [’rbf’, ’linear’, ’sigmoid’]. These hyperparameters use grid search meaning the SVM model is trained on all combinations of hyperparameters and finds the best setting.

2) Random Forest: Random forest is a widely used machine learning method for classification and regression use cases [15]. We can define Random Forest as an ensemble method that aggregates results from multiple models. A random forest could be called a forest of decision trees. A decision tree provide a means of classification and regression. Each tree, dependent on regression or classification, will output a final score or class. These will either be averaged or voted amongst the trees to determine a random forest classification.

We use Sklearn's implementation of the Random Forest algorithm. The Sklearn implementation is simple to use, allowing for customizing of the model. In our instance we changed the random state of the model to 777, and the n_estimators hyperparameter. n_estimators are used to specify the number of trees in the Random Forest. The number of trees numerically determined to be the best with grid search was 100.

3) AdaBoost: We also utilize another ensemble method of Machine Learning AdaBoost(Adaptive Boosting [16]. In this ensemble method weak classifiers named stumps are used. The aggregated class votes are used similar to a Random Forest, except in this instance each stump has a certain importance to it(its opinion is more valuable). A stump consists of a small one level decision tree.

We use the Sklearn Adaboost implementation [17]. The hyperparameters used are n_estimators to specify decision stumps and the learning rate. Grid search gives 50 trees, and a learning rate of 1 as the best values for these hyperparameters.

4) Logistic Regression: Logistic regression is a popular classification method [18]. A line is fit to the data by minimization of a loss function and application of gradient descent. A prediction value is obtained from this line. A sigmoid function is then applied to the prediction value, and this sigmoid will return a value between 0 and 1. Finally, based on this sigmoid output, we have some thresholds that will be used to classify our example.

Sklearn provides an implementation of logistic regression that proved well for our task [19]. We used hyperparameters C, solver, and multi_class. C is a parameter used to specify regularization strength. Solver allows us to customize the algorithm to use in the optimization. The C value used for the model was 5, the solver method used was ‘liblinear’, and the multi_class hyperparameter was set to ’auto’.

5) KNN Classifier: KNN (k nearest neighbors) is a classification method [20]. For classification, an unlabeled sample first collects the closest k (hyperparameter) samples. Per each of these k samples, a label vote amongst these is done. The class that is most prevalent amongst these neighbors will be given to the unlabeled sample. Scikit-Learn provides a simple implementation of the KNN algorithm [21]. The best k hyperparameter was chosen to be 4.

E. Model Testing

In addition to models to use for machine learning applications, Sklearn provides functions to calculate metrics such as accuracy, precision and recall [21]. To facilitate model testing, each model was fitted to the train data, then these models predicted test results based on some test data. These test predictions were used with Sklearn functions to calculate precision, recall, and an f1-score. We see below the results from each of the five models being tested (fig 3.).

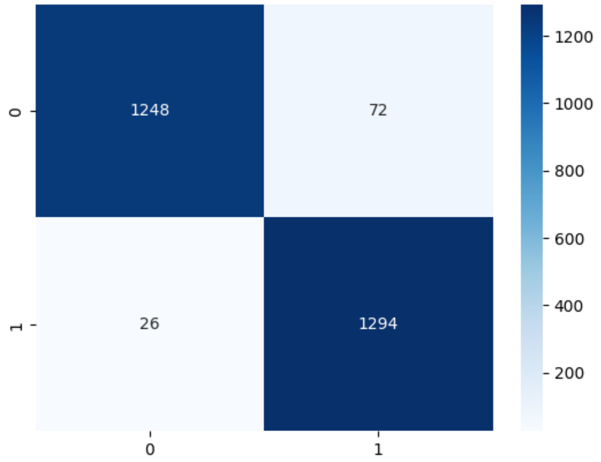
The RF(Random Forest) model produces the highest scores by far, with Adaboost being second, KNN third, SVM, and finally Logistic Regression. Using these scores, we generated

Fig. 10. ”Test Results from Experiment”

Model	Accuracy	Precision	Recall	f1-score
KNN	90.91	0.91	0.91	0.91
RF	96.29	0.96	0.96	0.96
LR	87.50	0.88	0.88	0.87
AdaBoost	93.75	0.94	0.94	0.94
SVM	88.45	0.89	0.88	0.88

a confusion matrix for the Random Forest model below. The confusion matrix begins demonstrates the following sequence in clockwise starting at the left corner [True Negative, False Positive, True Positive, False Negative].

Fig. 11. ”Confusion Matrix for RF Test Results”



We see a visualization of the model success, a high rate of True Positives and True negatives, with low rates of False Positives and False Negatives.

F. Experiment Results

Looking at the features we can see that the debt ratio, and net worth have the largest feature importance. This is directly in line with conventional financial standards where the debt ratio is the first metric looked at when gauging the financial health of a company. [11]

Comparing the 5 models we can see that the random forest classifier has both the highest accuracy, F1, recall, and precision. Random forest models are known to perform extremely well outside the box and require little in data preparation tasks. Random forest is also known to have overfitting issues, but looking at the test accuracy and precision compared to the other models showed us that the model was not overfitting.[9] These results made us confident that the random forest classifier is the best out of the 5 tested at predicting financial viability of companies. These results are also corroborated in a paper investigating the effectiveness of random forest at predicting financial failures from data collected by Turkish banks during the period of 1994 - 2004. They achieved an accuracy of 94% in the test split, compared to an accuracy of 90.90% for an SVM model, and 81.81% for a k neighbors model.[10] These results are very comparable to the results we found considering their similarity in data source, as well as the problem they were trying to classify. This further gave us confidence in our findings that predicting financial outcomes with the Random Forest regressor looks to be the most effective at getting results.

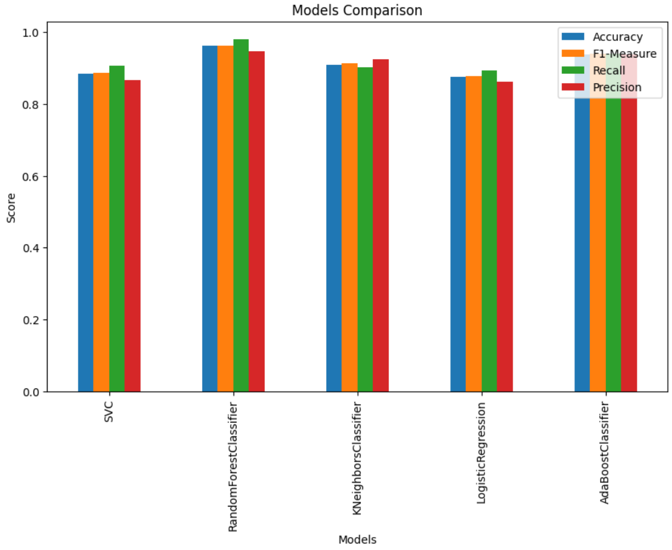


Fig. 12. Visualization of Test Results

IV. Conclusion

In conclusion, our findings closely match the findings of others. From the feature importance analysis we concluded that the assets, and debt ratio have the biggest impact on whether a company will go bankrupt or not. This is directly in line with conventional business practices. In addition, we found that among the 5 non linear regressors explored that the random forest classifier outperformed all the others in all 4 metrics we looked at (accuracy, precision, F1, and recall). These results compounded with findings of Rustam and Saragih in their paper titled: “Predicting Bank Financial Failures using Random Forest”[10] made us confident in our findings.

References

[1] A. Narvekar and D. Guha, “Bankruptcy prediction using machine learning and an application to the case of the covid-19 recession,” Data Science in Finance and Economics, vol. 1, no. 2, pp. 180–195, 2021.

[2] J. Fernando, “Debt-to-equity (D/E) ratio formula and how to interpret it,” Investopedia, 05-Apr-2023. [Online]. Available: https://www.investopedia.com/terms/d/debtequityratio.asp. [Accessed: 10-Apr-2023].

[3] A. Narvekar and D. Guha, “Bankruptcy prediction using machine learning and an application to the case of the covid-19 recession,” Data Science in Finance and Economics, 2021. [Online]. Available: https://www.investopedia.com/articles/active-trading/081315/financial-ratios-spot-companies-headed-bankruptcy.asp [Accessed: 10-Apr-2023].

[4] G. Giannopoulos and S. Sigbjørnsen, “Prediction of bankruptcy using financial ratios in the Greek market,” Theoretical Economics Letters, 15-Mar-2019. [Online]. Available: https://www.scirp.org/journal/paperinformation.aspx?paperid=92181 [Accessed: 10-Apr-2023].

[5] “1.4. Support Vector Machines,” Scikitlearn. [Online]. Available: https://scikit-learn.org/stable/modules/svm.html. [Accessed: 10-Apr-2023].

[6] D. S. Goswami, “Applying smote for class imbalance with just a few lines of Code Python,” Medium, 18-Feb-2021. [Online]. Available: https://towardsdatascience.com/applying-smote-for-class-imbalance-with-just-a-few-lines-of-code-python-cdf603e58688. [Accessed: 10-Apr-2023].

[7] J. Brownlee, “How to calculate feature importance with python,” MachineLearningMastery.com, 20-Aug-2020. [Online]. Available: https://machinelearningmastery.com/calculate-feature-importance-with-python/. [Accessed: 10-Apr-2023].

[8] M. Grogan, “Feature selection techniques in Python: Predicting hotel cancellations,” Medium, 09-Sep-2020. [Online]. Available: https://towardsdatascience.com/feature-selection-techniques-in-python-predicting-hotel-cancellations-48a77521ee4f. [Accessed: 10-Apr-2023].

[9] J. Kho, “Why random forest is My Favorite Machine Learning Model,” Medium, 12-Mar-2019. [Online]. Available: https://towardsdatascience.com/why-random-forest-is-my-favorite-machine-learning-model-b97651fa3706. [Accessed: 10-Apr-2023].

[10] “Predicting bank financial failures using Random Forest — IEEE ...,” May-2018. [Online]. Available: https://ieeexplore.ieee.org/document/8471718. [Accessed: 11-Apr-2023].

[11] T. I. Team, “Financial ratios to spot companies headed for bankruptcy,” Investopedia, 15-Mar-2023. [Online]. Available: https://www.investopedia.com/articles/active-trading/081315/financial-ratios-spot-companies-headed-bankruptcy.asp. [Accessed: 10-Apr-2023].

[12] A. Narvekar and D. Guha, “Bankruptcy prediction using machine learning and an application to the case of the covid-19 recession,” Data Science in Finance and Economics, 11-Oct-2021. [Online]. Available: https://www.aimspress.com/article/doi/10.3934/DSFE.2021010?viewType=HTML. [Accessed: 10-Apr-2023].

[13] Z. Rustam and G. S. Saragih, “Predicting bank financial failures using random forest,” IEEE Explore, Sep-2018. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8471718/. [Accessed: 11-Apr-2023].

[14] W. Zhang, “Machine learning approaches to predicting company bankruptcy,” Journal of Financial Risk Management, 13-Dec-2017. [Online]. Available: https://www.scirp.org/html/4-2410244_81016.htm#t1. [Accessed: 10-Apr-2023].

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, “Scikit-Learn: Machine learning in Python,” Journal of Machine Learning Research, 01-Jan-1970. [Online]. Available: https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html. [Accessed: 10-Apr-2023].

[16] A. Desarda, “Understanding the ADABOOST algorithm,” Built In, Feb-2023. [Online]. Available: https://builtin.com/machine-learning/adaboost. [Accessed: 10-Apr-2023].

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, “Scikit-Learn: Machine learning in Python,” Journal of Machine Learning Research, 01-Jan-1970. [Online]. Available: https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html. [Accessed: 10-Apr-2023].

[18] S. Mondal, “Regression analysis: Beginners comprehensive guide (updated 2023),” Analytics Vidhya, 14-Feb-2023. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/12/beginners-take-how-logistic-regression-is-related-to-linear-regression/. [Accessed: 10-Apr-2023].

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, “Scikit-Learn: Machine learning in Python,” Journal of Machine Learning Research, 01-Jan-1970. [Online]. Available: https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html. [Accessed: 10-Apr-2023].

[20] T. Srivastava, “A complete guide to K-Nearest Neighbors (updated 2023),” Analytics Vidhya, 15-Feb-2023. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/. [Accessed: 10-Apr-2023].

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, “Scikit-Learn: Machine learning in Python,” Journal of Machine Learning Research, 01-Jan-1970. [Online]. Available: https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html. [Accessed: 10-Apr-2023]