Corporate credit rating prediction with decision tree models

Yu Shing Cheng[1] **Yan Kit Wong**[1]

Abstract

The corporate credit rating project comes at a critical time as the mid-year of 2023 witnessed a wave of bankruptcy and liquidity crises among banks and corporations. These challenging circumstances arose against the backdrop of a high-interest rate macro environment, which further intensified the need for effective credit risk assessment and management. This project aims to develop an effective and automatic credit rating system for assessing the creditworthiness and risk profile of corporate entities with decision tree models and evaluate the performances. The code is available at https://github.com/LilAiluropoda/corporate-credit-rating.

1. Introduction

Credit rating plays a crucial role in the financial industry as it provides valuable insights to investors, lenders, and other stakeholders regarding the financial health and stability of corporations. The current mechanism of credit rating involves a thorough analysis of financial and non-financial factors, applying rating methodologies, and assigning credit ratings to entities based on their creditworthiness and risk profile, which are performed manually by credit rating agencies. This mechanism can potentially take up a lot of manpower and time. Therefore, by constructing an effective credit rating model, this project aims to enhance the speed, accuracy, consistency, and transparency of the credit rating process. This will assist financial institutions, investors, and regulators in making more informed decisions, managing risks effectively, and promoting stability in the corporate credit market.

In this project, we are particularly interested in the application of the decision tree classifiers, since it has been proven to be constructed relatively fast compared to other methods of classification while obtaining similar and sometimes better accuracy.

[1] UNC-CH, Chapel Hill, NC, United States . Correspondence to: Yu Shing Cheng <cyshi@unc.edu>, Yan Kit Wong <sonicw@unc.edu>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

2. Related Works

2.1. Decision Tree

A decision tree is a graphical representation resembling a flowchart-like tree, in which each internal node signifies a test conducted on a specific attribute. The branches of the tree correspond to the possible outcomes of the test, while each leaf node represents the assigned class label. When a tuple X is presented, the attribute values of the tuple are evaluated against the decision tree. By following a path from the root to a leaf node, we can determine the predicted class for the tuple(Sharma Kumar, 2016).

2.2. Random Forest

The Random Forest algorithm is designed to fit multiple classification trees to a dataset and consolidate their predictions. It operates by first selecting a significant number of bootstrap samples from the dataset. Any observations not included in a bootstrap sample are classified as out-of-bag (OOB) observations. A classification tree is then fitted to each bootstrap sample, with only a limited number of randomly selected variables being available for binary partitioning at each node. The trees are fully grown, and their predictions for the OOB observations are then computed. To determine the predicted class of a specific observation, the algorithm uses a majority vote based on the OOB predictions for that instance, with ties being broken at random (Cutler et al., 2007).

2.3. Gradient Boosting Decision Trees

Gradient boosting trees are ensemble techniques that sequentially construct decision tree learners by fitting the gradients of the residuals from previously built tree learners. The tree-building process begins with a single node and iteratively introduces branches until a specific criterion is fulfilled. When adding branches to each leaf node, the objective is to maximize the reduction in loss following the split (Anghel, A., 2018).

Some common gradient boosting tree algorithms include GBM, XGBoost, LightGBM, and CatBoost. These algorithms are widely used for their performance and efficiency in handling gradient boosting tasks.

2.4. Principal Component Analysis

Principal component analysis (PCA) is a statistical method used to analyze multivariate data tables that consist of interrelated quantitative dependent variables. The objective of PCA is to extract key information from the table and represent it through a set of orthogonal variables known as principal components. By visualizing the observations and variables as points on maps, PCA enables the identification of patterns and similarities within the data (Abdi Williams, 2010).

3. Dataset

A dataset containing 7805 corporate credit ratings by 7 credit rating agencies like Standard Poor's, Egan-Jones, and Mitch from 2010 to 2016 is used for the project, which covers 678 companies from 12 sectors. Note that all the ratings have been remapped to the SP rating scale (AAA to D, 22 grades in total). This dataset is available at https://www.kaggle.com/datasets/kirtandelwadia/corporate-credit-rating-with-financial-ratios. Figure 1 illustrated the distribution of sectors.

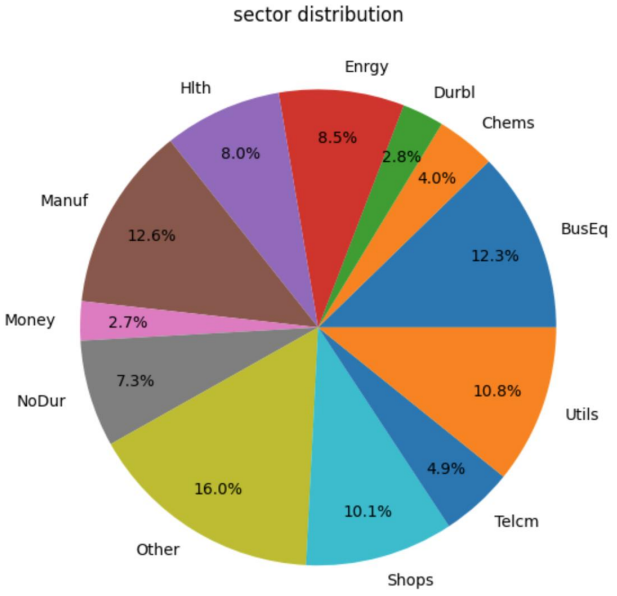

Figure 1. The distribution of sectors.

The columns of the dataset are illustrated in Table 1, where the Rating column will be used as the label, while the rest will be used as features. Irrelevant columns like Rating Agency, Corporation, CIK(Central Index Key), SIC Code, Binary Rating, and Ticker are excluded since they are not helpful in modeling the relationship.

Table 1. Column descriptions of the data set.

ITEM	USED?
RATING AGENCY	
CORPORATION	
RATING	YES
RATING DATE	YES
CIK	
BINARY RATING	
SIC CODE	
SECTOR	YES
TICKER	
CURRENT RATIO	YES
LONG-TERM DEBT/CAPITAL	YES
DEBT/EQUITY RATIO	YES
GROSS MARGIN	YES
OPERATING MARGIN	YES
EBIT MARGIN	YES
EBITDA MARGIN	YES
PRE-TAX PROFIT MARGIN	YES
NET PROFIT MARGIN	YES
ASSET TURNOVER	YES
ROE-RETURN ON EQUITY	YES
RETURN ON TANGIBLE EQUITY	YES
ROA-RETURN ON ASSETS	YES
ROI-RETURN ON INVESTMENT	YES
OPERATING CASH FLOW PER SHARE	YES
FREE CASH FLOW PER SHARE	YES

4. Methodology

4.1. Data Preprocessing

Prior to model construction, a thorough examination of the dataset is performed to identify any missing or inconsistent columns. Subsequently, irrelevant columns are eliminated from the dataset, and a test is conducted to identify and address duplicate rows.

Due to the dataset's imbalanced nature, meaning the sample sizes for each label vary significantly, a remapping procedure is employed to address this issue. Specifically, the labels are reassigned to four distinct grades, denoted as A to D, in order to alleviate the imbalance. The resulting distribution of labels can be visualized in Figure 1.

After that, the sector and rating columns will be represented in one-hot vector form since text datatype cannot be fed into the model directly. In addition, to enhance the performance of the model, each numerical column will be scaled between zero to one with a MinMax scalar.

A correlation test will also be conducted on each column.

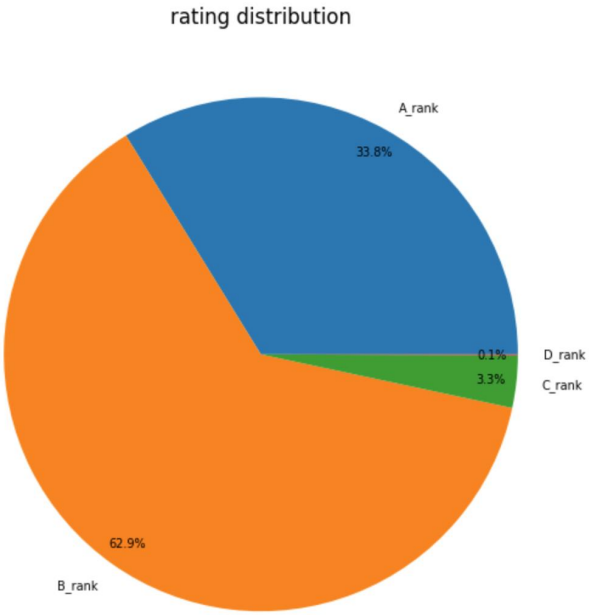

Figure 2. The distribution of labels after remapping.

Table 2. configurations of Random Forest

HYPERPARAMETER	VALUE
bootstrap	TRUE
max depth	40
max features	SQRT
number of estimators	600
random state	1234

Figure 3 illustrated the correlation between each column. A Principal Principle Analysis Decomposition (PCA) will then be performed to reduce correlation and collinearity.

4.2. Model Selection

For the project, decision tree models have been chosen, specifically Random Forest and Gradient Boosting Decision Trees. These selected models will be employed to analyze and process the data.

4.3. Model Construction

XGboost and scikit-learn packages are imported for the construction of the models. The scaled features will be fed into the models and predict the credit rating for the instance.

The optimal hyperparameters for each model are selected by a grid search algorithm. Table 2 showed the configurations of Random Forest while Table 3 showed configurations of Gradient Boosting Decision Trees.

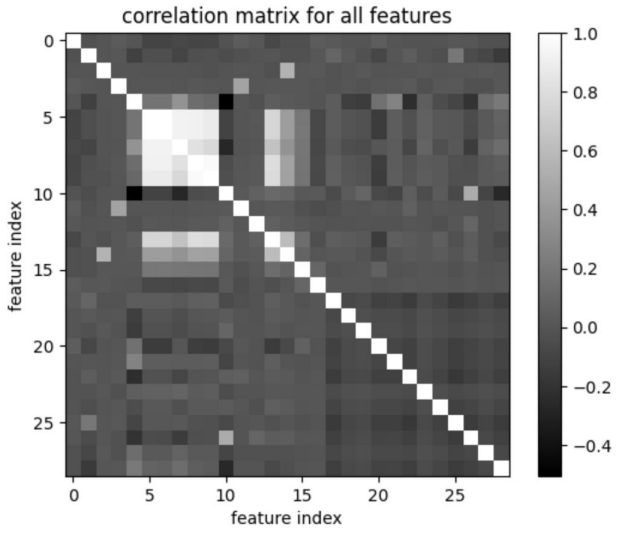

Figure 3. Correlation matrix for all columns.

Table 3. configurations of GBDT

HYPERPARAMETER	VALUE
max depth	6
number of estimators	180
learning rate	0.1

4.4. Model Training

To accommodate the constrained size of the dataset, a 10-fold cross validation approach will be implemented to ensure effective training and validation of the model. This technique involves dividing the dataset into ten subsets, allowing for multiple iterations of training and evaluation to enhance the reliability of the results.

5. Experimental Results

5.1. Evalutation Metrics

Logistic loss, accuracy score, and F1 scores are used to evaluate the performance of the model. The logistic loss will determine the loss of the model and is defined as,

$$L_{log}(y, p) = -(ylog(p) + (1 - y)log(1 - p))$$

The accuracy score will determine the prediction accuracy of the model. Note that since the dataset is imbalanced, the accuracy score will be balanced by calculating the average recall obtained in each class.

The F1 score is a harmonic mean of precision and recall, where an F1 score reaches its best value at 1 and worst score

at 0. F1 score is defined as,

$$\frac{2 * precision * recall}{precision + recall}$$

5.2. Model Performance

Table 4 and Table 5 have demonstrated the performance of Random Forest and GBDT respectively. As we can see, the two models obtained similar performance in terms of accuracy (See Figure 4). However, GBDT showed lower loss and less stable F1 score as shown in Figure 5 and Figure 6.

Table 4. Performance of Random Forest

Metric	*Average*	*Max*	*Min*
MeanLogisticLoss	0.449	0.490	0.374
Accuracy	0.839	0.854	0.815
BalancedAccuracy	0.628	0.753	0.500
F1Score	0.857	0.871	0.837

Table 5. Performance of GBDT

Metric	*Average*	*Max*	*Min*
MeanLogisticLoss	0.397	0.438	0.345
Accuracy	0.842	0.860	0.806
BalancedAccuracy	0.615	0.729	0.483
F1Score	0.860	0.879	0.827

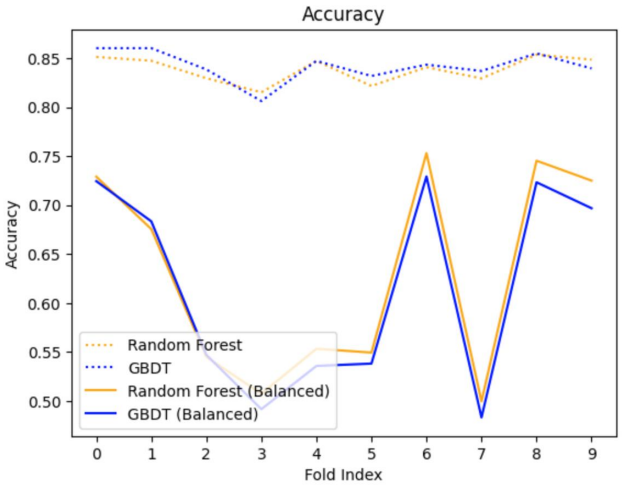

Figure 4. Accuracy comparison of the two models.

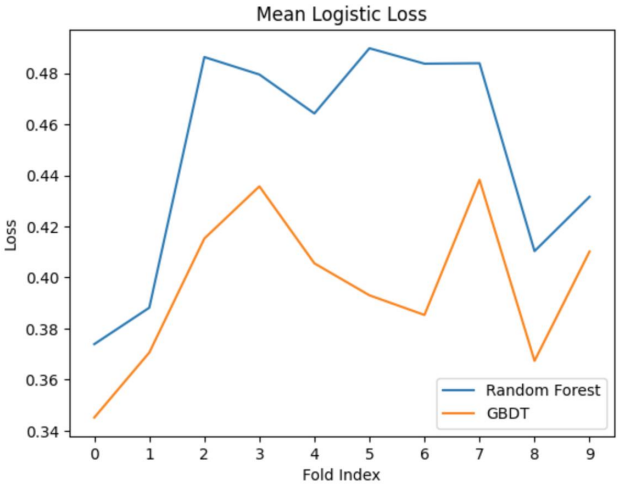

Figure 5. Loss comparison of the two models.

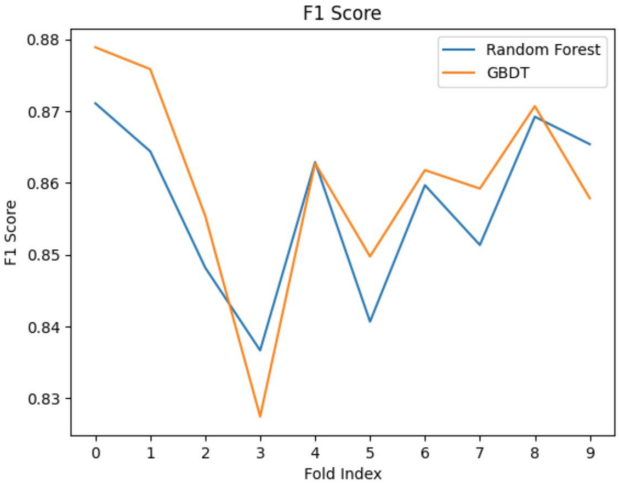

Figure 6. F1 score comparison of the two models.

6. Conclusion

Despite the nonideal accuracy of the current model, we believe it can be further improved by including more features or expanding the size of dataset. We hope to see more work in the future that can achieve our ultimate goal of making manual rating faster, cheaper, and more efficient.

References

Sharma, H., amp; Kumar, S. (2016). A survey on decision tree algorithms of classification in Data Mining. International Journal of Science and Research (IJSR), 5(4), 2094–2097. https://doi.org/10.21275/v5i4.nov162954

Cutler, D. R., Edwards, T. C., Beard, K. H., Cutler, A., Hess, K. T., Gibson, J., amp; Lawler, J. J. (2007). Random forests for classification in ecology. Ecology, 88(11), 2783–2792. https://doi.org/10.1890/07-0539.1

Anghel, A., Papandreou, N., Parnell, T., Palma, A.D., Pozidis, H. (2018). Benchmarking and Optimization of Gradient Boosted Decision Tree Algorithms. ArXiv, abs/1809.04559.

Abdi, H., amp; Williams, L. J. (2010). Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics, 2(4), 433–459. https://doi.org/10.1002/wics.101