Alonso Salcido, Chris Haub

Lab 4, XGBoost

**Introduction**

This project was designed to predict bankruptcy using Random Forest and Extreme Gradient. The objective of predicting financial distress is to create a predictive model that combines various econometric measures to anticipate the financial state of a company. This project presents an approach for bankruptcy prediction that employs Extreme Gradient Boosting to learn an ensemble of decision trees. Additionally, to incorporate higher-order statistics into the data and impose prior knowledge on data representation, a new concept called synthetic features was introduced.

A synthetic feature is created by combining various econometric measures using arithmetic operations such as addition, subtraction, multiplication, and division. Each synthetic feature can be considered as a single regression model that evolves over time. To assess the effectiveness of our approach, we analyzed data gathered from Polish companies in five different tasks related to predicting bankruptcy in the first, second, third, fourth, and fifth year.

**Methods**

5 Augmented Archive Files (AARF) were provided and utilized to evaluate the effectiveness of predicting bankruptcy. The AARF files contained data on Polish companies, including their financial statements, and were collected over a period of five years. The data included various financial ratios, such as liquidity ratios, profitability ratios, and solvency ratios, as well as other measures such as size, age, and sector. The dataset is highly imbalanced with most companies not experiencing bankruptcy. After initially inspecting the files, they were imported and concatenated into data frame.

Chart, bar chart

Description automatically generated

The dataset only contains one prescribed target outcome and that is column 65, bankruptcy. Missing values were visualized. Only one column, (current assets - inventories) / long-term liabilities was missing over 20% of the data. One column, sales (n) / sales (n-1), was missing over 10% of the data. Rest of the columns were all missing less than 10% of data.

Chart, histogram

Description automatically generated

Next, the training data was prepared to create subset of the data with non-null values for the target variable and selecting only the relevant features for training and excluding the target variable.

To deal with the large number of missing values in the column (current assets - inventories) / long-term liabilities the data was imputed using the values of the k-nearest neighbors. In this case, the KNNImputer class from the scikit-learn library was used, with a parameter of the four nearest neighbors used for imputation. The imputed values were then merged with the original data frame using concat from pandas. Finally, the column of the original 'data' data frame was updated with the imputed values.

Random Forest was performed using k-folds cross validation to evaluate the model’s performance. Grid search was completed using balanced data to adjust weights based on number of samples in each class.

Finally, we used XGBoost to obtain the highest scores. We used matplotlib to plot the data of the training and test set results

Chart, line chart

Description automatically generated

**Feature Importance**

XGBoost was ranked by feature importance. With (current assets - inventory) / short-term liabilities as most important followed by profit on operating activities / financial expenses coming in second.

Chart

Description automatically generated

**Results**

The Random Forest k-folds cross-validation scores for each fold were 90.3%, 89.2%, 88.2%, 89.8%, 90.0%, with a mean of 89.50% and a standard deviation of 0.74. The best result from grid search was 91.7%. The best parameters were 200 estimators, min sample split 8, min sample leaf 2, using square root with max depth of 30 on the entropy criterion. XGBoost helped us get a higher accuracy. After running a 5 fold cross validation on XGBoost, the model has a mean ACU score of 95.03% and a standard deviation of 0.57%.

Diagram, schematic

Description automatically generated

**Appendix A**Text

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generated