DS7333 Quantifying the World: Case Study 4­

Rick Fontenot, Feby Thomas Cheruvathoor

1. **Introduction**

­

The goal for this case study is to use historical financial data to predict which companies are at risk of going bankrupt. These predictions will be used by the finance division of investment firms to divest of companies in advance that would cause significant losses in the future. For this study random forest and XGBoost models will be used to make predictions on bankruptcy classification.

1. **Methods**

**Data description**

The raw data set contains 43,405 observations of financial data separated into 5 files broken into the following timeframes:

**1stYear** contains financial rates from 1st year of the forecasting period and corresponding class label that indicates bankruptcy status after 5 years. The data contains 7027 instances (financial statements), 271 represents bankrupted companies, 6756 firms that did not bankrupt in the forecasting period.

**2ndYear** contains financial rates from 2nd year of the forecasting period and corresponding class label that indicates bankruptcy status after 4 years. The data contains 10173 instances (financial statements), 400 represents bankrupted companies, 9773 firms that did not bankrupt in the forecasting period.

**3rdYear** contains financial rates from 3rd year of the forecasting period and corresponding class label that indicates bankruptcy status after 3 years. The data contains 10503 instances (financial statements), 495 represents bankrupted companies, 10008 firms that did not bankrupt in the forecasting period.

**4thYear** contains financial rates from 4th year of the forecasting period and corresponding class label that indicates bankruptcy status after 2 years. The data contains 9792 instances (financial statements), 515 represents bankrupted companies, 9277 firms that did not bankrupt in the forecasting period.

**5thYear** contains financial rates from 5th year of the forecasting period and corresponding class label that indicates bankruptcy status after 1 year. The data contains 5910 instances (financial statements), 410 represents bankrupted companies, 5500 firms that did not bankrupt in the forecasting period.

The bankrupt companies were analyzed in the period 2000-2012, while the still operating companies were evaluated from 2007 to 2013.

Each year contains 64 identical features of financial metrics including working capital and ratios such as total assets / total liabilities and net profit / sales. For a description of all 64 features see appendix section II.

**Processing data and Feature Creation**

Python tool **scipy.io.arff.loadarff** was used to read in the arff files provided.

All features were properly read in numeric types. The target “bankrupt” initially read in as an object with binary vales encoded as [b'0', b'1']. A dictionary of d={b'0': 0, b'1': 1} was created to convert the target into 0 and 1 values for modeling and classification.

Studying the breakdown of the percentage of companies that went bankrupt by the yearly time frame shows that there are higher rates of bankruptcy as the number of years increase.

Percentage of Bankrupt companies by Year



Figure 1: Average Bankruptcy status by year

A feature “year” indicating the number of years of prior to bankruptcy was created to use along with the financial features in modeling.

**Cross Validation**

In order to compare the models for overfitting, we first set aside 10% of the data into a validation set (stratified shuffled data to get random observations with similar proportion of records in the target classes).

For the training data used to build models, we used 10-Fold cross validation and the mean accuracy across the folds for model assessment. In addition to accuracy, we will also summarize the precision and recall for each class, for comparison.

We performed the validation split prior to further data processing so that the process is valid to make predictions on unseen data.

**Class Imbalance**

With all 5 yearly files of data combined 4.8% of the companies went bankrupt (95.2% not bankrupt). With this class imbalance, simply predicting all companies “not bankrupt” would result in 95.2% accuracy. Models need to account for this imbalance and the benchmark to beat for accuracy is high.

Each of the modeling tools have parameters to handle class imbalance with Random Forest having the class\_weight parameter where “balanced” mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data as n\_samples / (n\_classes \* np.bincount(y))

The XGBoost Classifier has the scale\_pos\_weight parameter which balances of positive and negative weights. Bankrupt is the positive class and with 95.2% not bankrupt / 4.8% bankrupt the nominal weight is approximately 19.8%

In addition to these internal model parameters, a secondary evaluation using SMOTE to resample the training set to be balanced was performed for comparison.

**Random Forest modeling**

A random forest model using default parameters for the sklearn RandomForestClassifier was created as an initial benchmark for further model comparisons.

Beyond the base random forest model, a randomized CV grid search to maximize accuracy was performed with the following parameter ranges:

'max\_depth': [5, 15, 25, 50],

'n\_estimators': [50, 100, 300, 500],

'min\_samples\_split': [2, 5, 10],

'min\_samples\_leaf': [5, 10, 15],

'class\_weight': ['None', 'balanced']

The best mix of parameters for the tuned model:

'n\_estimators': 300

'min\_samples\_split': 5

'min\_samples\_leaf': 5

'max\_depth': 25

'class\_weight': 'balanced'

Of particular importance, setting the parameter class\_weight to ‘balanced’ to handle the class imbalance improved the model.

A secondary random forest model with SMOTE resampled training data was performed with the class\_weight parameter set to ‘None’

**XGBoost modeling**

A base model for XGBoost was created with the following parameters:

n\_estimators: 500

max\_depth:10

objective: ’multi:softmax'

num\_class: 2

eta: 0.01

early\_stopping\_rounds: 2

Beyond the base XGBoost model, a randomized CV grid search to maximize accuracy was performed with the following parameter ranges:

'n\_estimators':[500,1000],

'eta':[0.001, 0.01,0.1],

'max\_depth':[5,15,20],

'max\_leaves': [30,50,70],

'min\_child\_weight': [5, 10, 15],

'gamma': [1, 1.5, 2, 2.5],

'subsample': [0.6, 0.8, 1.0],

'scale\_pos\_weight': [1,10,20,30]

'early\_stopping\_rounds': 5

'eval\_metric': 'logloss'

The best mix of parameters for the tuned model:

eta: 0.1

gamma: 1

max\_depth: 20

max\_leaves: 50

min\_child\_weight: 10

n\_estimators: 1000

scale\_pos\_weight: 30

subsample: 0.8

Of particular importance, setting the parameter scale\_pos\_weight to 30 to handle the class imbalance improved the model. This is higher than the nominal value of 19.8 we calculated as potential starting point in sections above

A secondary XGBoost model with SMOTE resampled training data was performed with the scale\_pos\_weight parameter set to 1.

1. **Results**

The base model using random forest obtained 96.4% accuracy but suffers poor recall on the minority class bankrupt. XGBoost and tuning hyperparameters improved performance metrics, see a summary of all models below:

Classification Performance Summary



Figure 2: Classification Report for each model considered

The best performing models were the XGBoost models with tuned hyperparameters. Using the scale\_pos\_weight or the SMOTE resampling to handle class imbalance provided similar classification performance on XGBoost. The resampling method is more involved and the training time and grid searches take considerably longer, so using the original imbalanced data with parameter weighting is the preferred model.

Note that the grid search to tune random forest hyperparameters increased training accuracy but suffered from overfitting as the accuracy on the validation set was lower than the default random forest model

Note that SMOTE resampling looked promising with higher training set accuracies but also suffered from overfitting and the accuracy on validation sets were comparable to using the class weight parameters on the original imbalanced data within the models.

The following features were most important in the XGBoost decision making:

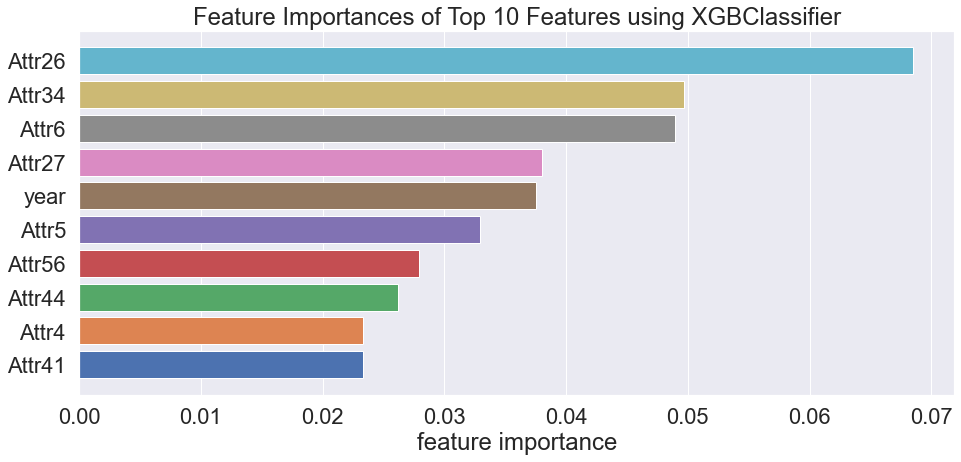


Figure 3: Feature importance obtained from XGBClassifier() model

Attribute 26 was most important feature and represents the companies (net profit + depreciation) / total liabilities. The rate of bankruptcy shows an exponential relationship where the lower the ratio the higher rate of bankruptcy. This makes sense intuitively as companies income level needs to cover liabilities or risk going bankrupt.

Attribute 34 was the second most important feature and represents the companies operating expenses / total liabilities. The relationship to bankruptcy rate is not linear or as intuitive to understand as attribute 26. Further discussion with domain experts may help explain this relationship, but it does show that the XGBoost model is good at capturing non-linear relationships that tools like logistic regression may not handle as well.

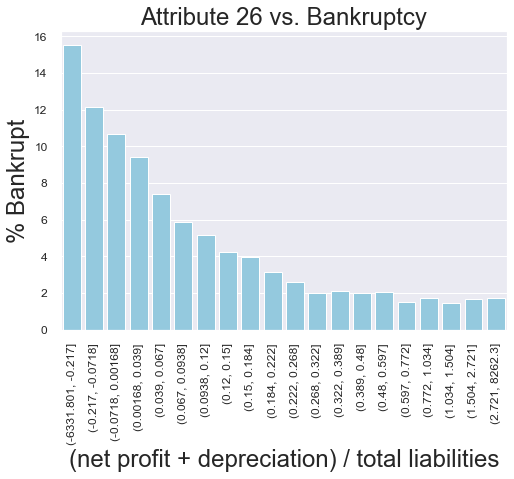
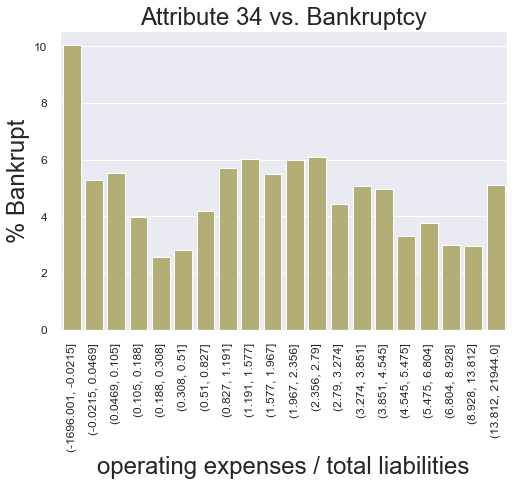
 

Figure 4: Attr26 Bins vs. Bankruptcy Rate Figure 5: Attr34 Bins vs. Bankruptcy Rate

Also note that the ‘year’ feature that was added ranks 5th in importance and did have a positive effect on classification performance.

A confusion matrix using the standard 0.5 probability cutoff shows that 60% of the true bankrupt companies were correctly predicted. Only 0.03% of the non-bankrupt companies were misclassified as bankrupt predictions.

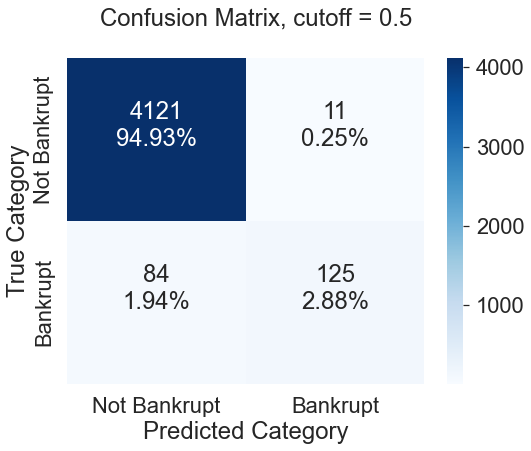


Figure 6: XGB-tuned Confusion Matrix w/ 0.5 prob. cutoff

Precision and Recall trade-offs by shifting the discrimination threshold on predicted probabilities of bankruptcy were studied in the plot below:

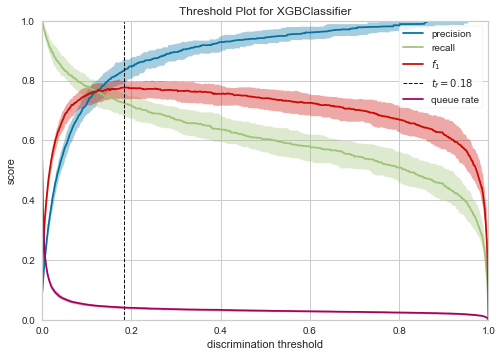


Figure 7: XGB-tuned Precision & Recall vs. probability cutoff

A probability threshold of 0.18 maximizes the f1 score to 0.75 (vs. 0.72 using the 0.5 probability cutoff). While shifting this threshold increased from 60% to 71% of the true bankrupt companies being correctly predicted, it also increases misclassifications on non-bankrupt companies.

Experimenting with cutoff points shows that a 0.83 probability threshold maximizes the true bankrupt companies predicted as such (47%) while not misclassifying any of the non-bankrupt companies. See matrices below:

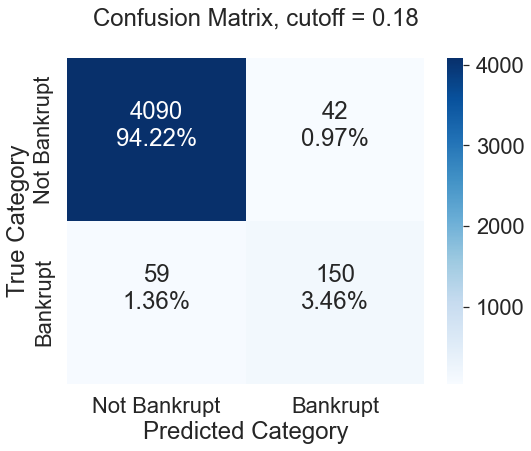
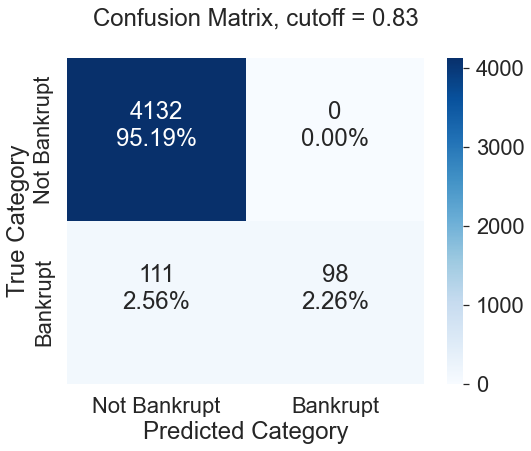
 

Figure 8: Confusion Matrix w/ 0.18 prob. Cutoff Figure 9: Confusion Matrix w/ 0.83prob. cutoff

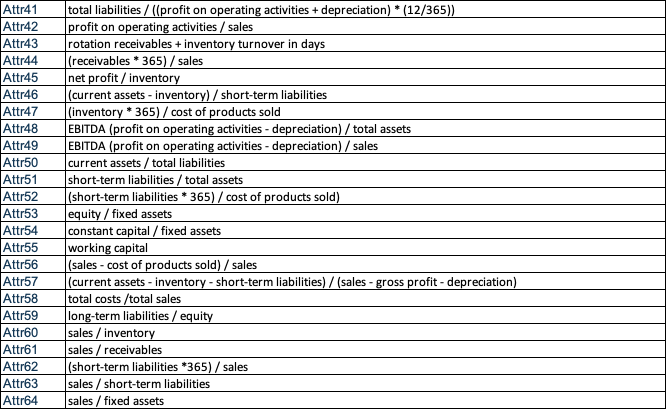
1. **Conclusions**

A probability threshold of 0.18 maximizes the f1 score to 0.75 (vs. 0.72 using the 0.5 probability cutoff). While shifting this threshold increased from 60% to 71% of the true bankrupt companies being correctly predicted, it also increases misclassifications on non-bankrupt companies.

**Appendix**

1. **Feature Descriptions**





1. **Code**

A rendered notebook containing code for this analysis can be accessed at: