Bankruptcy Prediction of Polish Company

Kris Ghimire, Rajesh Satluri, Suchismita Moharana

June 21, 2021

**Abstract**

One of the biggest lose to any company is when company go bankrupt due to various investment strategies. Company would spend any kind of resources to stop the bankruptcy if they can find out before hand when and how could it possibly be bankrupt. In this research, Polish companies’ historical data will be used to predict bankruptcy in future.

**1. Introduction:**

Bankruptcy is defined as a situation in which company is no longer financially stable and goes through legal process to get help in eliminating or repaying its dept under the counsel and protection of federal court. It is an unfortunate fact that many small companies sometime run through financial troublesome. Being able to predict if the company will suffer through bankruptcies in future is not only essential for analyzing the financial and operational condition and make better business decision. Accurate prediction of bankruptcy is of important concern not only to company but also to other stakeholders such as investor, financial institutes, policymakers, employees, and consumers. It helps stakeholders to better prepare for and protect themselves against any major potential financial shocks.

Prediction

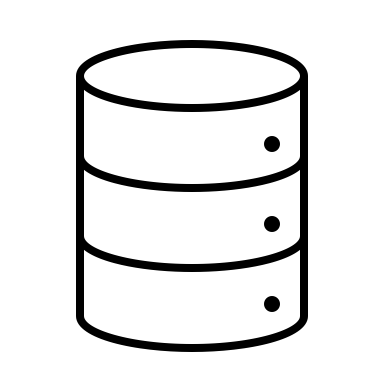
History Data

ML classifier

 No bankrupt



Bankrupt



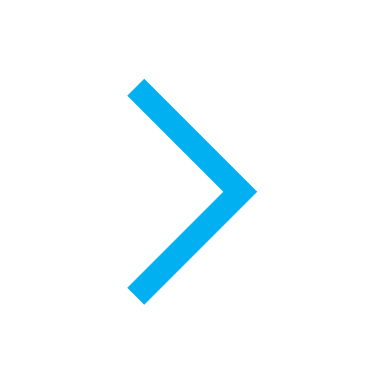
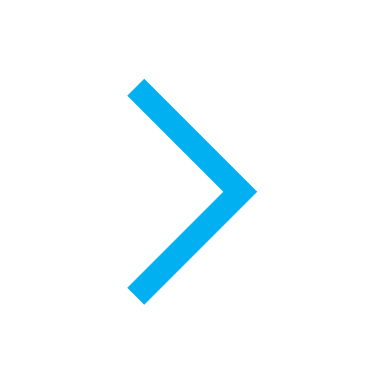
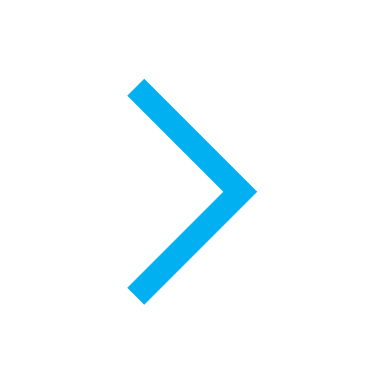
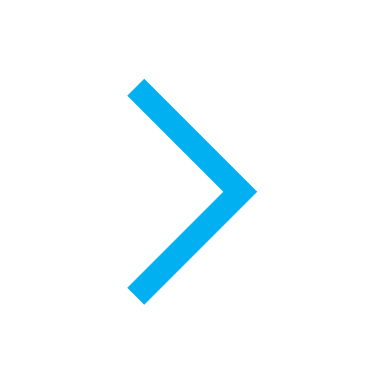
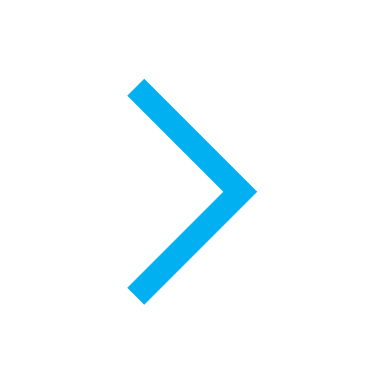
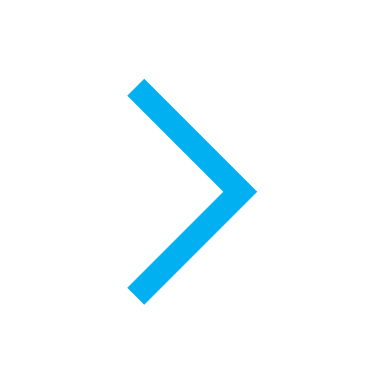
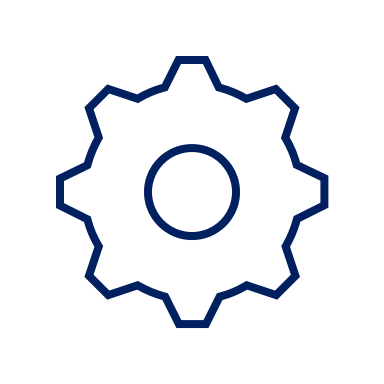
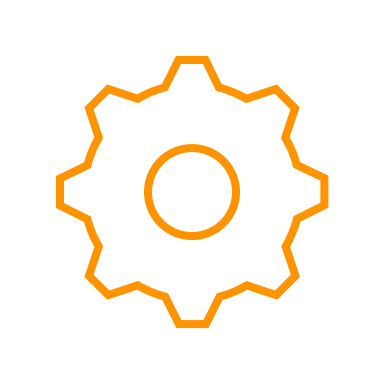
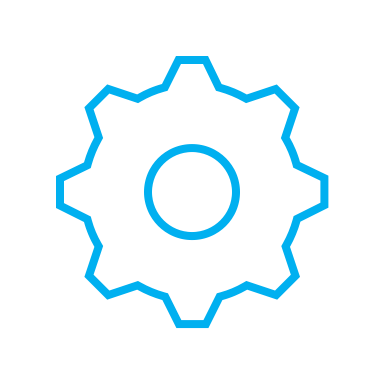


Figure 1:

## 2. Methods

**Dataset:**

Table

Description automatically generated

Table 1

The data set is Polish companies’ bankruptcy data, which contains 43,405 observations and 65 variables. The first 64 variables are the predictors, including all sorts of companies’ current financial status and properties such as net profit, total liabilities and working capital against sales, remaining earnings as well as total assets. The 65th variable is the response, with values b’1’ and b’0’ reflecting whether the company will be bankrupt the next year. Upon visualizing target features, (figure 2) we noticed unequal distribution of classes bankrupt and no bankrupt. Only 2,091out of 41,314 observations are bankrupt, indicating the data is highly imbalanced which is quite common and always a challenge in classification problems. As with most machine learning algorithms, uneven distribution of class ratio in classifiers could leads to an inaccurate estimate of class prior which could then potentially decrease the predictive performance. Moreover, there exist 5835 missing values in the predictors, so an imputation technique should be performed.

Chart, bar chart

Description automatically generated

Figure 2

**Data Exploration:** One of the observations we had is there was significant bankruptcy in 5th year data. We are not sure here if the collected dataset was random or it included almost all companies from a particular category. Since we are not sure about how dataset was collected, we decided not to consider the year factor as one of the reasons of bankruptcy. Ex: in 2020, a lot of small companies filed bankruptcy because they went thru heavy loss because of corona virus. Now, we know that it happened because of a total unrelated external agent. Here we will go ahead without considering the year factor at all. For handling null values, we did impute the missing data with median.

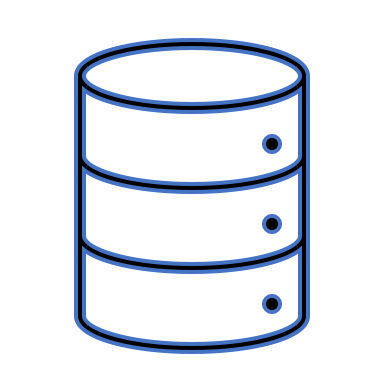
**Model Assumption:** Here in this case study/ research paper we are dealing with mostly ensemble algorithms. Ensemble algorithms are kind of like meta-algorithm that combines many other small ML technique such as decision trees into one big predictive model to decrease variance and bias and improve prediction. Hence, like the linear models, ensemble model does not have any separate assumption of their own. One key focus of these algorithm is to make sure we have tuned parameters to fit the model.

**Train Test Split:** While dataset is split into training and test set (80/20 split) to keep test data separate and to perform modeling only on training set. And later use test set to test the model general performance.

**ML model pipelines:** We will follow ML pipeline to build the classification model. Figure 5.

**Training**

Historical Dataset



Data preparation/ Data Cleaning- Remove duplicates, check outliers, impute NA with median etc.

Standard Scaling

Class Balanced – class-weight ,SMOTE, MSMOTE

Exploratory Analysis: Check normality, Feature dependency, outlier detection

Features

Feature Scaling

Standard Scaling

Machine Learning classifiers (LR,RF,XGboost )

Feature Scaling

5 folds Cross-validation

**Prediction**

Test Set Prediction

Learned Classifier Model Selection, Best Model

Metric Evaluation

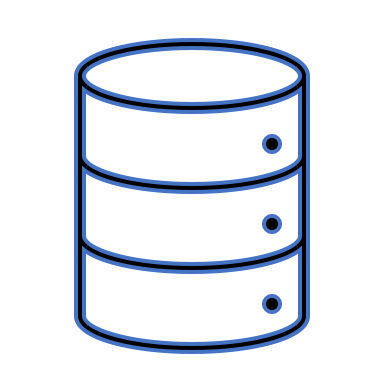
-Accuracy

-Precision

-Recall

- F1 Score

Test Historical data



Data preparation/ Data Cleaning- Remove duplicates, check outliers, impute NA with median etc.

Recommended Label

No Bankrupt Bankrupt

*Figure 5: ML Flowchart*

**Imbalance target features:** Dataset has imbalanced target features. Modeling on imbalanced target features would give evaluation metrics that are not accurate. The key concept to getting accurate predictive model is to balance the data before fitting predictive model. Hence, We will implement two approaches, first we will use algorithm default parameter to balance data called class-weight = ‘balanced’ , next we will use SMOTE technique. We will build both models, with unbalanced data as well as balanced data using SMOTE.

**Cross-Validation**: We will use StratifiedKFold cross-validation with k=5 because with stratifiedKFold, the class distribution in the dataset is preserved in the training and test splits. Shuffle is set to true so that the splitting will be random.

**Decision Tres:**

Decision trees (DT) are a method to classify data into segments using a yes/no logic. They are formed from partition trees. Partition tree even though not used directly, forms the basis of all other algorithms such as Random Forest and XGBoost, that we will implement to solve the bankruptcy problems. Contrary to linear algorithms which uses line to separate targets, partition tress are used to solve the nonlinear problems and can be decrease or increase in any direction. How the decision trees work is, we partition data into bins. For instance, as shown in figure 1 below, if value in column A is less than 5 place it in bin A else place in in bin B. So, what happens is we go through data and make decision based on the data. The decision is made based on side that has the maximum information gain. The Maximum information gain is calculated using two different technique Gini or entropy. Either Gini or entropy can be used however, based on type of data and problem we are trying to solve, we can pick the one that give better performance metric. We repeat the process of splitting until we reach the stopping criteria. The stopping criteria can be as simple or complex as we want. We can have a set number of decisions such as we can continue until five decision or 10 decision which can be set using parameter max depth.

Yes

Bankruptcy

Total assets < 1.5M

No bankruptcy

Total Profit < 2M

No

Total Sales > 5M

No

Yes

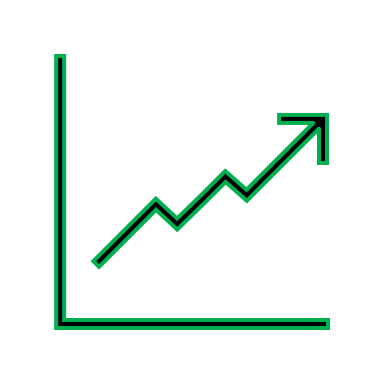
No

Yes

No Bankruptcy

Bankruptcy

Figure 2: Decision Tree

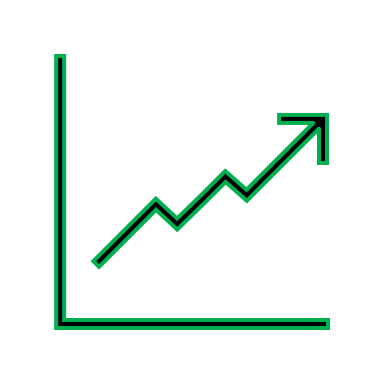
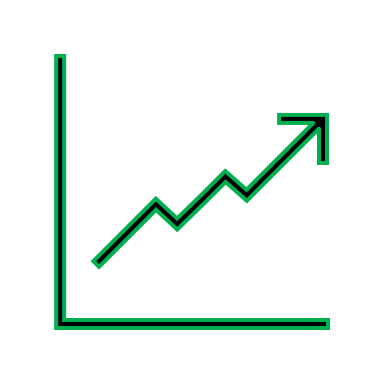
**Random Forest:** One major drawback of DT is that as the number of data and depth of the trees increase, they are more prone to overfit. Which means that model might work really well with training data but will perform poorly on the test data. To overcome this issue, we use Random Forest (RF). Random Forest is an ensemble model that fits a number of DT classifiers on various sub-samples of the dataset and uses an averaging of all DT to improve the model performance and checks the over-fitting issues (Figure 3). Sub-sample of dataset in RF are done with replacement called bootstrapping. Also with RF classifiers, all the trees are trained in parallel (bagging model / bootstrap aggregation). The key concept in RF algorithms is that by randomly picking the data and random selection of features at each split, we will have each tree to be uncorrelated with each other trees. Hence, due to this randomness, the bias of the forest normally slightly increases. However, due to averaging of all DT, the overall variance of the RF model decreases significantly as compared to increase in bias and hence RF give us an overall better model. Random forest, while on surface seeming to be a simple algorithm, is really one of the best initial guesses that we can find in modern data science.



DT1

X1

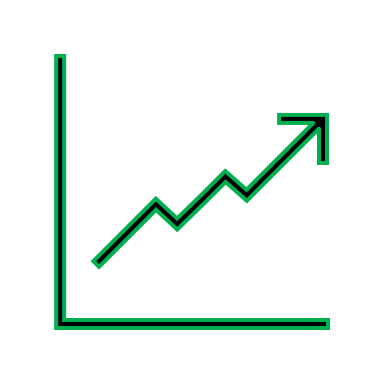
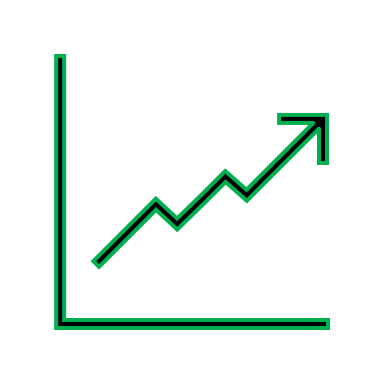
X2



DT2

X1

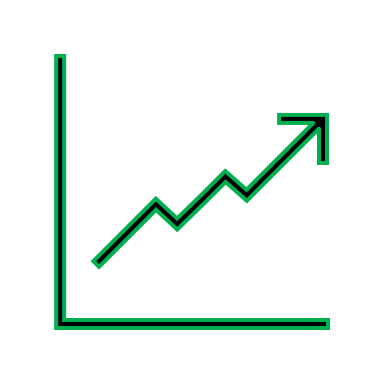
X2



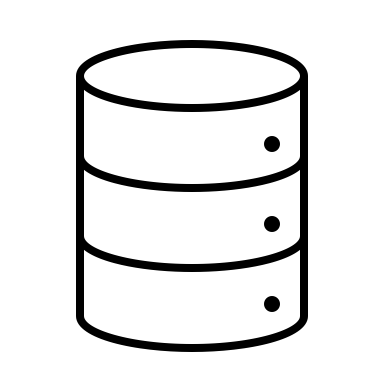
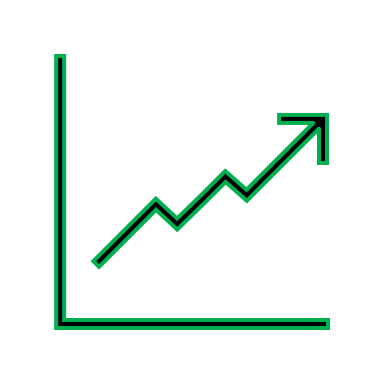
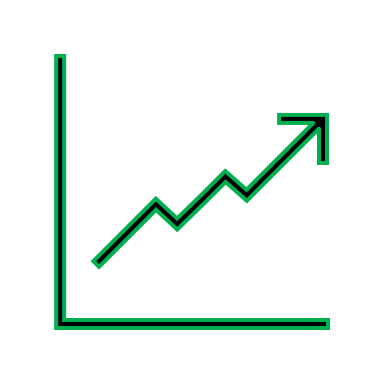
DT3

X1

X3



X3



Data input

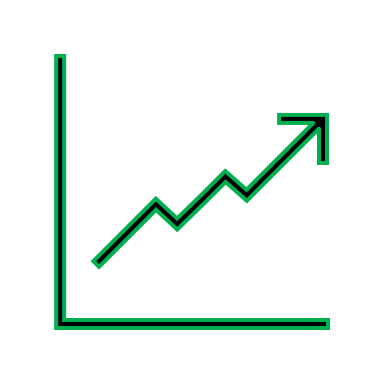
All trees

Prediction:

DT1:

DT2:

DT2:



Avg prediction green

RF Pred = 



Figure 4: Random Forest

**XGBoost Classification:** XGBoost which stands for extreme Gradient Boosting) relies on the concept of boosting. Boosting is a way to make weak learners produce a strong learner by iteratively fitting on the residual (sequential manner) from the previous round instead of target until it reaches to the point where the residual is normally distributed. The key advantage of XGboost is its high-performance speed compared to other model and its regularization parameter (figure 4) that successfully reduce the variance. Besides speeds and regularization XGboost also leverages a learning rate (shrinkage) and subsamples from the feature like RF, which increases its ability to generalize even further. Another concept that makes Xgboost so powerful is, it uses an approximate loss (use any loss function). So instead of actually going through the calculus and finding specific updates rule, xgboost approximate that loss.

Penalties

General loss function

Text

Description automatically generated

Text

Description automatically generated with medium confidence

T= Nodes and leaves and = Slope

Figure 6: xgboost lost function

With xgboost instead of penalizing slopes we penalize trees. The above penalty term makes tree to be smaller as the number of trees increases. Also, we can notice that not only is slope w score involved in the equation, it’s also the second order of the score, hence making it analogous to L2 regularization.

**Feature Importance:** Knowing which features could possibly lead to company bankruptcy might be of useful information for company to make better decision as to where they need to focus more. Hence, using RF classifier in figure below we have soon the top 10 importance features. We can see that feature Attr37 which corresponds to its exact name (current assets – inventories) / long-term liabilities from data dictionary are the most importance feature followed by Attr27 which stands for profit on operating activities / financial expenses.

Chart, bar chart

Description automatically generated

**Figure 10: RF Top Features**

**3. Result:**

We could use any classifiers to build model to classify bankruptcy. However, main objective of this case study is to show that xgboost performs better than other classifiers logistic regression, decision tree and random forest. Therefore, first we begin with building simple logistic regression (LR) model. Figure below shows that LR mode perform poorly for this problem with mean accuracy of 67%. For the next few ensemble models, to getter and more generalize score we will perform parameter tunning using Randomsearch technique instead of GridSearch as RandomSearch tends to run much faster than GridSearch and the difference in metric between two is not much difference. Hence, using the best parameters we built RF model as our second classifier. Figure below, shows that RF model does much better than LR model with mean accuracy of 97.4%.

As our goal is to build model that bit’s RF model, next we built XGboost model. We can see that based on accuracy score XGboost model does the best classification.

Chart, bar chart

Description automatically generated

Dataset we have used to build model is not a balanced dataset. Relying solely on accuracy score can be misleading. To further compare Xgboost with RF model we used other model validation metrics, precision, recall, ROC and F1 score. The AUC-ROC tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting no bankruptcy classes as 0 and bankruptcy classes as 1. Since LR model is our base model, other validation metrics are not shown in this paper. From the confusion matrix and ROC curve below in figure we can see that XGboost gives better score for precision and f1-score compared to RF. While AUC-ROC score for both RF and XGboost are quite similar. In term of f1-sore which is a combination of precision and recall and conveys the balance between those two, shows that XGboost bits RF.

**Model 2:**

Chart, treemap chart

Description automatically generatedChart, line chart

Description automatically generated

*Figure: Random Forest confusion matrix Figure : Random Forest ROC curve*

Table

Description automatically generated

*Figure 9. Random Forest Classification Report*

**Model 3:**

Chart, treemap chart

Description automatically generated Chart, line chart

Description automatically generated

*Figure 11. XGboost ROC Curve*

*Figure 10: XGboost Confusion matrix metrics*

Table

Description automatically generated

Table

Description automatically generated

*Figure 12: XGboost classification Report*

**4. Conclusion:**

We built three models in this research RandomForest, XGBoost and Logistic Regression. We used Logistic regression is for result comparison only. We did extensive comparison between RandomForest and XGBoost for performance. We observed that XGBoost faster than RandomForest. Further analysis on this can be done with high volume of data to do the comparison even more intensely.