Bankruptcy Prediction for Polish Companies

Table of Contents

Problem Setting **3**

Data Description **3**

3. Data Exploration **5**

4. Data Mining Tasks **7**

5. Data Mining Models/Methods **8**

6. Performance Evaluation **11**

7. Project Results: **18**

8. Conclusion **19**

References **19**

1. Problem Setting

Bankruptcy prediction is important since early warnings can help stakeholders like managers, investors and even public policy makers to proactively minimize the impact of firms declaring bankruptcy.

Early prediction of bankruptcy can help preserve scarce resources by guiding managers and investors when firms are in danger of undergoing bankruptcy so corrective action can be taken. Furthermore, this system can also prove to be a guide for policy makers who can identify any underlying systemic issues or particular issues affecting a sector.

A major challenge in bankruptcy prediction is identifying variables that can be used in prediction; since the problem is diversified the inputs vary from individual financial ratios of a firm, its contributions, indicators reflecting the industry its working in and even the global economy.

The question is how reliable our built models are for predicting bankruptcies and can they be scaled to be used in the future.

2. Data Description

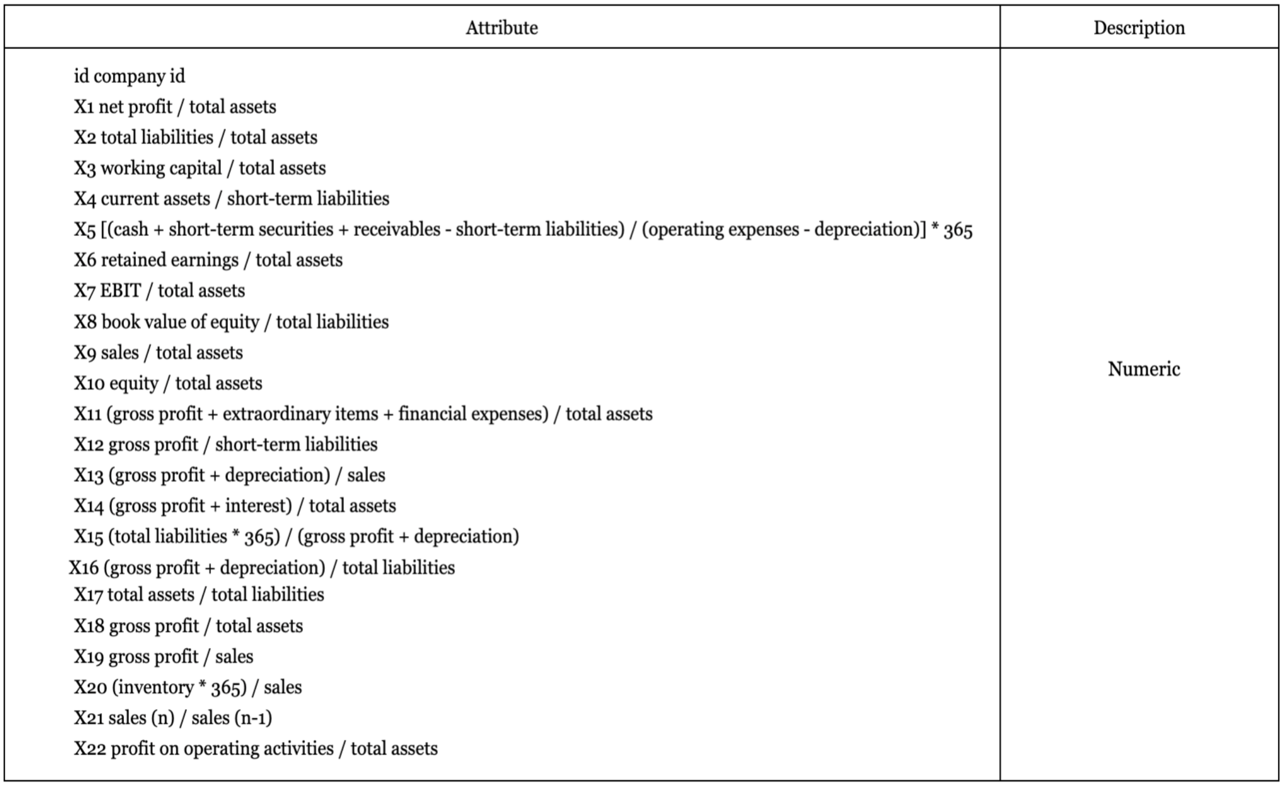
The data is sourced from UCI Machine Learning Repository and was collected from Emerging Markets Information Service. The bankrupt companies were analyzed in the period 2000-2012, while the still operating companies were evaluated from 2007 to 2013. We used the data from ‘Year 3’ for this project.

The dataset comprises 10,503 company instances, 64 attributes and 1 target variable ‘class’. Out of 10,503 instances, 495 firms have class values of 1(bankruptcy) and 10,008 firms did not go bankrupt.

Each feature is constructed using two or more accounting ratios. The synthetic features are formulated by performing arithmetic operations on the core indicators.

The purpose of the synthetic features is to combine the indicators into complex features and increase statistical significance.

The table below gives a description of some of the variables. The complete description of all variables can be found on the UCI Machine Learning Repository page.



Attribute	Description
id company id X1 net profit / total assets X2 total liabilities / total assets X3 working capital / total assets X4 current assets / short-term liabilities X5 [(cash + short-term securities + receivables - short-term liabilities) / (operating expenses - depreciation)] * 365 X6 retained earnings / total assets X7 EBIT / total assets X8 book value of equity / total liabilities X9 sales / total assets X10 equity / total assets X11 (gross profit + extraordinary items + financial expenses) / total assets X12 gross profit / short-term liabilities X13 (gross profit + depreciation) / sales X14 (gross profit + interest) / total assets X15 (total liabilities * 365) / (gross profit + depreciation) X16 (gross profit + depreciation) / total liabilities X17 total assets / total liabilities X18 gross profit / total assets X19 gross profit / sales X20 (inventory * 365) / sales X21 sales (n) / sales (n-1) X22 profit on operating activities / total assets	Numeric

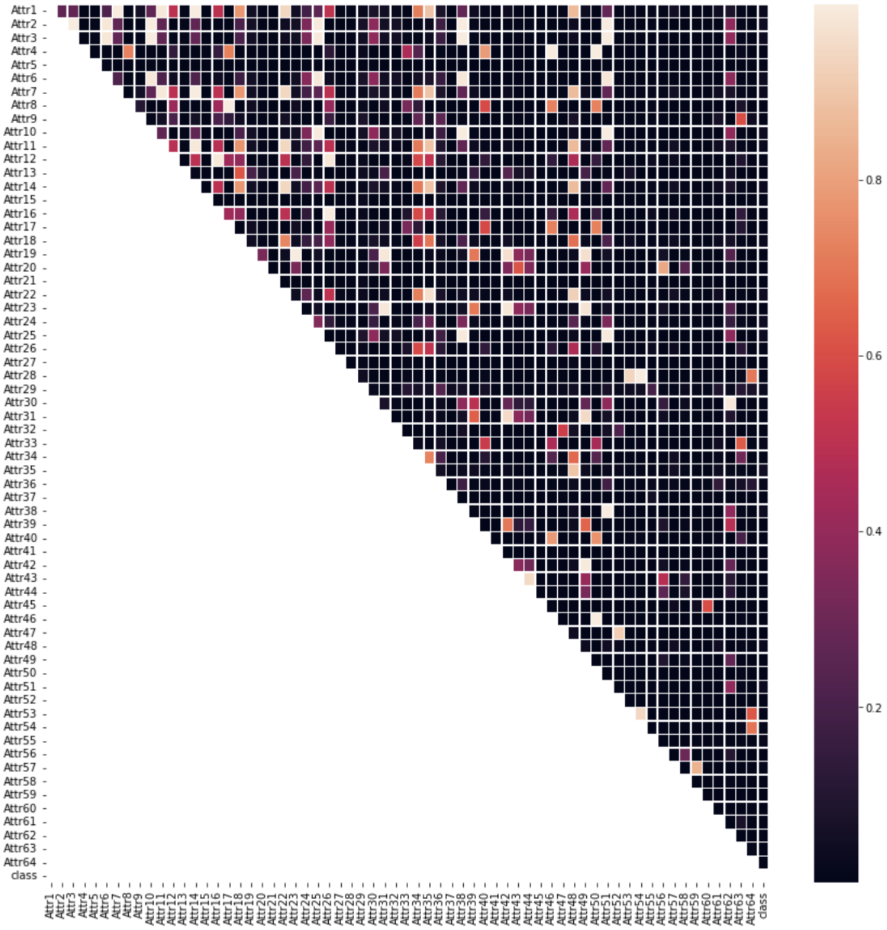
X23 net profit / sales X24 gross profit (in 3 years) / total assets X25 (equity - share capital) / total assets X26 (net profit + depreciation) / total liabilities X27 profit on operating activities / financial expenses X28 working capital / fixed assets X29 logarithm of total assets X30 (total liabilities - cash) / sales X31 (gross profit + interest) / sales X32 (current liabilities * 365) / cost of products sold X33 operating expenses / short-term liabilities X34 operating expenses / total liabilities X35 profit on sales / total assets X36 total sales / total assets X37 (current assets - inventories) / long-term liabilities X38 constant capital / total assets X39 profit on sales / sales X40 (current assets - inventory - receivables) / short-term liabilities X41 total liabilities / ((profit on operating activities + depreciation) * (12/365)) X42 profit on operating activities / sales X43 rotation receivables + inventory turnover in days X44 (receivables * 365) / sales X45 net profit / inventory X46 (current assets - inventory) / short-term liabilities X47 (inventory * 365) / cost of products sold X48 EBITDA (profit on operating activities - depreciation) / total assets X49 EBITDA (profit on operating activities - depreciation) / sales	Numeric
X50 current assets / total liabilities X51 short-term liabilities / total assets X52 (short-term liabilities * 365) / cost of products sold) X53 equity / fixed assets X54 constant capital / fixed assets X55 working capital X56 (sales - cost of products sold) / sales X57 (current assets - inventory - short-term liabilities) / (sales - gross profit - depreciation) X58 total costs /total sales X59 long-term liabilities / equity X60 sales / inventory X61 sales / receivables X62 (short-term liabilities * 365) / sales X63 sales / short-term liabilities X64 sales / fixed assets	Numeric
isBankrupted if it is bankrupt, then the value is 1, otherwise 0	Boolean

3. Data Exploration

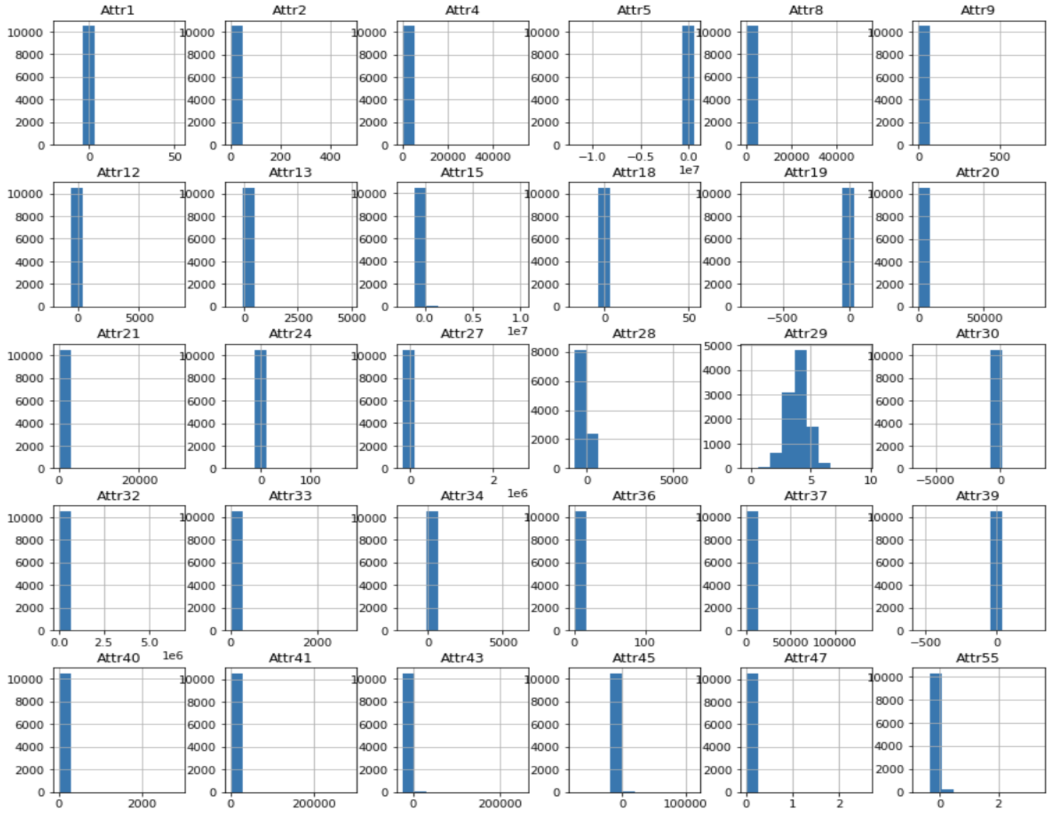
We have outlined below the process of exploration of the dataset and data mining before building models.

The correlation heatmap below was made to check the correlation between variables. Dimension reduction was done by removing variables that had a high correlation (<0.9). This method was utilized to speed up the computation of the models with minimum loss of information from reducing variables.

In the heatmap, the lighter colors indicate variables that have a higher pairwise correlation. As evident from the image, most variables are not highly correlated.



Next, distribution plots for the variables were made to check for the need to normalize the data prior to model generation. As seen, the data is mostly skewed so normalization is needed.



4. Data Mining Tasks

1) **Data reduction**

The first step was to check the data for missing values. Variables with a high percentage of missing values (>85%) were deleted.

Data reduction was done by removing variables that had high pairwise correlation as described in the previous section.

2) **Data transformation**

The data was transformed into type ‘float 64’ for ease. All input variables were already numeric so this was a simple process to undertake.

3) **Missing data imputation**

For variables that had missing values and were retained, mean imputation was performed. Since the variables are arithmetic ratios of financial indicators, domain knowledge could not be used to fill in these missing values.

4) **Data Partitioning**

Upon checking the data was found to be imbalance, with the minority '1' for the bankrupt variable occurring in only 4.71% of the data. The data was split into train and test sets using the sklearn 'train test split'. Oversampling was then performed on the train data set to ensure accurate prediction in the models generated.

The dataset was also normalized to minimize the scale difference of the variables.

5. Data Mining Models/Methods

Five classification data mining models were built using train data and each of them was tested and compared based on the performance result using test data. The five models are explained below:

1) Support Vector Machine (SVM)

The support vector machine (SVM) is a data classification algorithm that assigns new data elements to labeled categories; usually binary and sometimes multiclass. To predict whether a firm will go bankrupt or not, we have built an SVM since our question is also binary.

Advantages :

- Produce accurate and robust predictions
- Mostly unaffected by noise in the data
- Less prone to overfitting

Disadvantages :

- There is no probabilistic explanation for the classification.
- Not suitable for large datasets.

Implementation :

Grid search was used to finalize the parameters 'C' and gamma and the kernel function for the support vector machine. The results were found as below:

Name	Value
Kernel	Sigmoid
C	11.089139819925428

Gamma	0.0001

2) Logistic Regression

Logistic regression extends the ideas of linear regression to the situation where the response variable Y is categorical. Logistic regression classification model can be used for classifying a new observation, where its class is unknown, into one of the classes based on the values of its predictor variables. Instead of using Y as the dependent variable, we use a function of it which is logit.

Advantages :

- It is easy to implement, interpret and very efficient to train.
- It does not make any assumptions to the distribution of classes in features.
- It is easier to extend the model to multi-class situations which use multinomial regression.

Disadvantages :

- If the number of observations is smaller than the number of predictor variables, there is a problem of overfitting to train data.
- Also one of the major limitations is the assumption of linearity between the dependent variable and predictor variables.

Implementation

The base logistic regression model was executed and the accuracy of model was 68%

3) Naive Bayes Classifier

Naive Bayes classifier is a classification model based on Bayes' theorem and it assumes the independence between predictors. Even this assumption of independence violates the in real world practices, this model usually delivers competitive classification accuracy.

Advantages :

- It is suitable for solving multi-class prediction problems.
- It can perform good classification performance with much less number of train data than other classification methods.

Disadvantages :

- The conditional independence assumption might be far from the real case.
- Zero probability problem : The probability of belonging to a certain class might be zero if the test data has a particular class that is not shown in train data.

4) KNN Classifier

KNN is a machine learning algorithm that is largely used on large-scale data mining tasks. In KNN classifier each point of the data set is plotted on a high-dimensional space where each axis means the value of each predictor variable and points located close to each other have high probability of belonging to the same class. Based on this idea, when a new data(point) is given, the model automatically finds the class of that point based on k number of neighbor data points.

Advantages :

- KNN is a non-parametric data mining model which means there are no assumptions to meet to implement KNN method.
- The model is instance-based learning which means it constantly evolves and adapts to new input data.
- KNN can be used for both classification and regression problems.

Disadvantages :

- Curse of dimensionality : The model struggles to predict the outcome of a new input variable if the number of variables is large.
- As the size of the dataset increases, the speed of the algorithm declines significantly.

Implementation

A grid search was done changing the value of the number of neighbors(k) and the best model was obtained when k = 3.

5) Decision Tree Classifier

The decision tree classifier partitions the data space based on the value of the predictor variable. This data-driven, non-parametric classification model develops in the direction to increase the homogeneity of each terminal node. It classifies new examples based on the logic rules according to the nodes of the decision tree.

Advantages :

- It is inexpensive to construct and easy to interpret the result intuitively.
- It does not require any normalization or standardization of variable due to its robustness to outliers

Disadvantages :

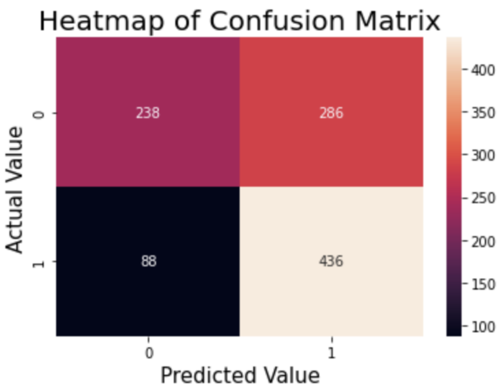
- Construction of decision trees can be volatile based on the change of data which means the logic rules of each node can change every time the data changes.
- Each node has one logic rule for one predictor variable value which means it is likely to miss any relationship between predictor variables that can highly affect the classification performance.

Implementation

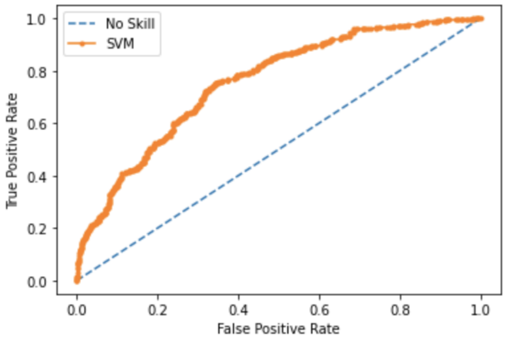
Multiple decision tree models were tested changing the maximum depth of the tree. he best model was obtained when the maximum depth was 2 with the accuracy of 0.68

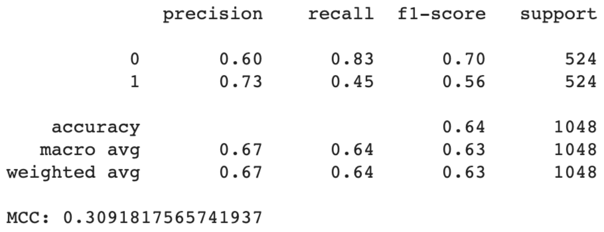
6. Performance Evaluation

1) Support Vector Machine (SVM)



```
No Skill: ROC AUC=0.500
SVM: ROC AUC=0.757
```



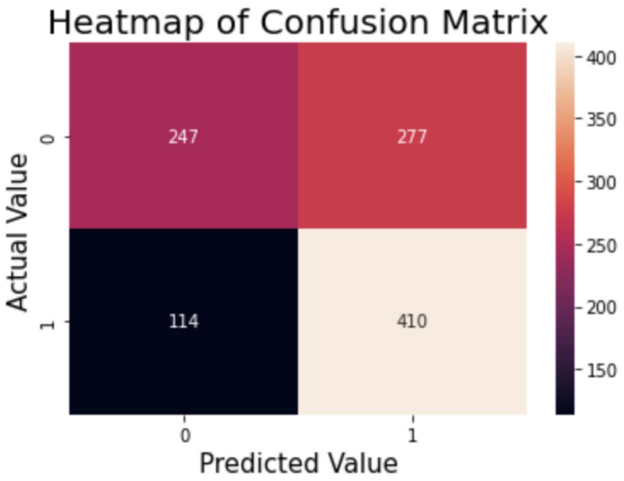


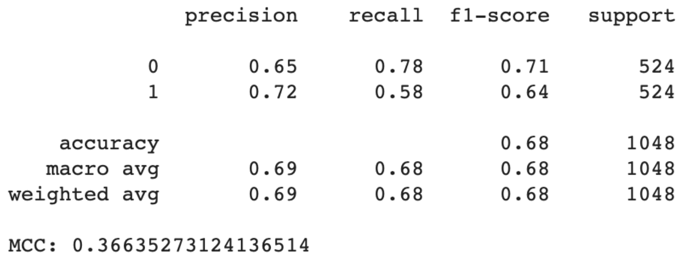
```
              precision    recall  f1-score   support

           0       0.60      0.83      0.70       524
           1       0.73      0.45      0.56       524

    accuracy                           0.64      1048
   macro avg       0.67      0.64      0.63      1048
weighted avg       0.67      0.64      0.63      1048

MCC: 0.3091817565741937
```

The AUC value of 0.757 showed that the model predicted bankrupt cases well. Overall the model showed an accuracy of 64% which is not desirable. The specificity was 83% indicating model could indicate a non-bankrupt case, however the sensitivity was 45% which meant model could not be trained well enough to identify bankrupt cases.

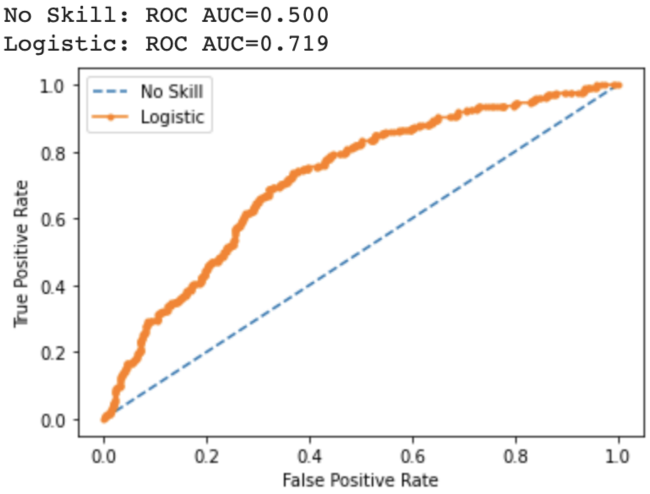
2) Logistic Regression

Our model is implemented with ƛ= 1 and equal weights are given for all the features with L1 regularization. The sensitivity value of 58% indicates it was able to correctly classify the True Positive which is a bankrupt case. The specificity value of 78% indicates it was able to correctly classify the True Negative which is a non-bankrupt case. The ROC curve indicates how sharply the model can discriminate between the bankrupt class and non-bankrupt class with an AUC value of 0.719.





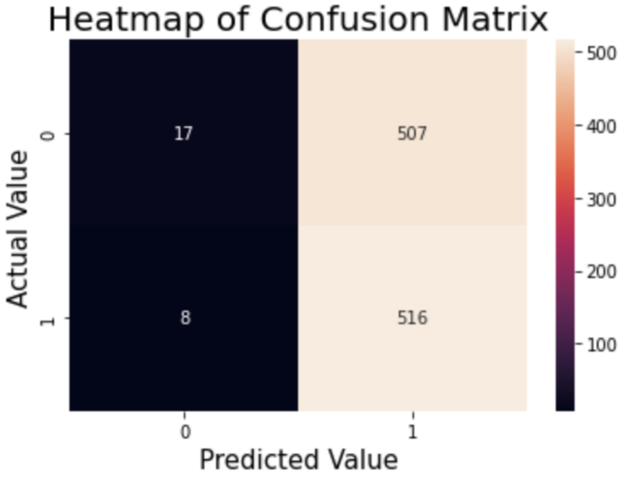
```
              precision    recall  f1-score   support

           0       0.65      0.78      0.71       524
           1       0.72      0.58      0.64       524

    accuracy                           0.68      1048
   macro avg       0.69      0.68      0.68      1048
weighted avg       0.69      0.68      0.68      1048

MCC: 0.36635273124136514
```

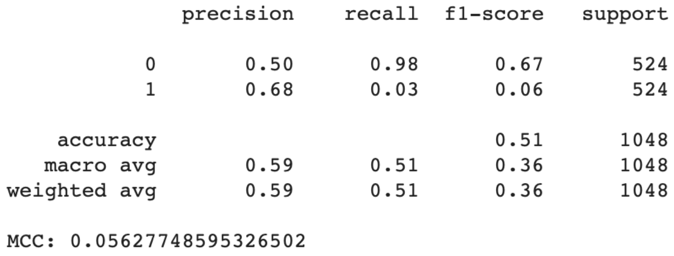


```
No Skill: ROC AUC=0.500
Logistic: ROC AUC=0.719
```

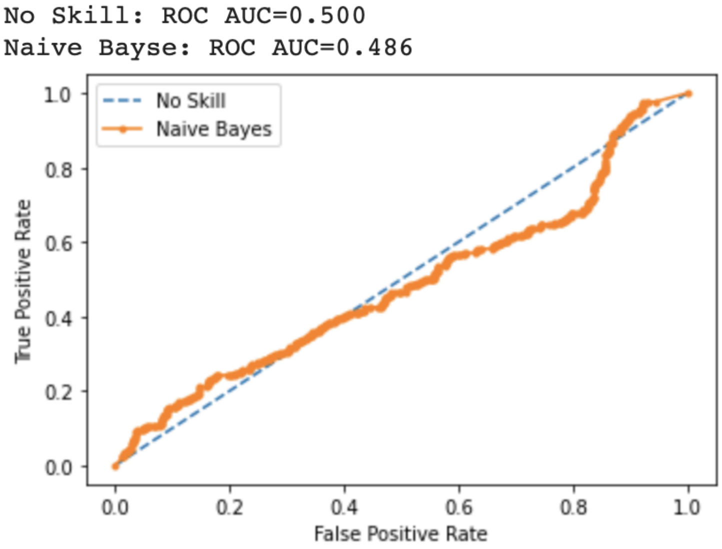
3) Naive Bayes Classifier

It showed poorer performance than the no-model case because the dataset we used wasn't dense enough for Naive Bayes classifier. The best model was obtained when the maximum depth was 2 with the accuracy of 0.51. The sensitivity value of 3% indicates it was able to correctly classify the True Positive which is a bankrupt case. The specificity value of 98% indicates it was able to correctly classify the True Negative which is a non-bankrupt case. The ROC curve shows this model shows worse performance than no-model guess with an AUC value of 0.486.





```
              precision    recall  f1-score   support

           0       0.50      0.98      0.67       524
           1       0.68      0.03      0.06       524

    accuracy                           0.51      1048
   macro avg       0.59      0.51      0.36      1048
weighted avg       0.59      0.51      0.36      1048

MCC: 0.05627748595326502
```

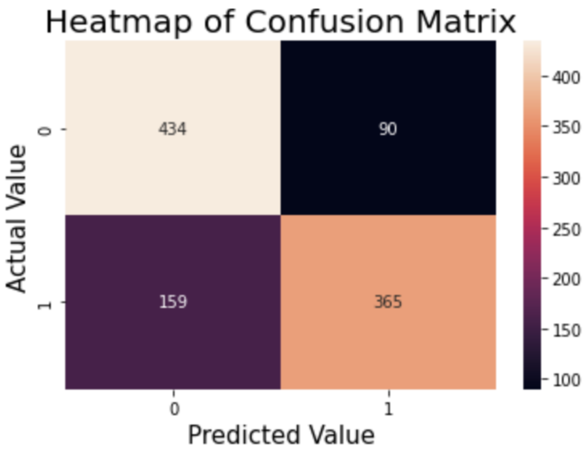


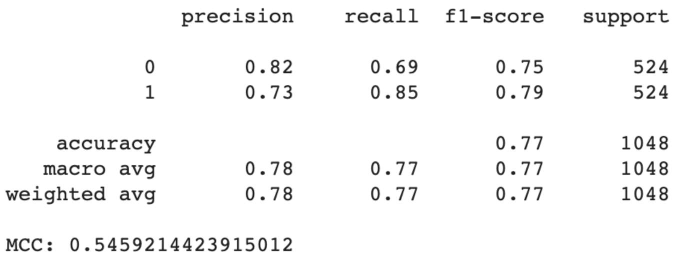
```
No Skill: ROC AUC=0.500
Naive Bayse: ROC AUC=0.486
```

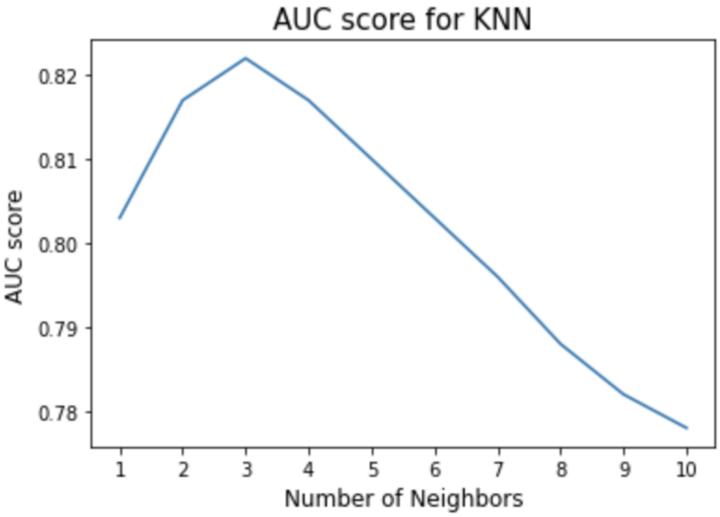
4) KNN Classifier

A grid-search was done changing the number of neighbors in KNN classifier and the best model was obtained when the k=3 with the accuracy of 0.77 with standard Euclidean distance metric. The sensitivity value of 85% indicates it was able to correctly classify the True Positive which is a bankrupt case. The specificity value of 69% indicates it was able to correctly classify the True

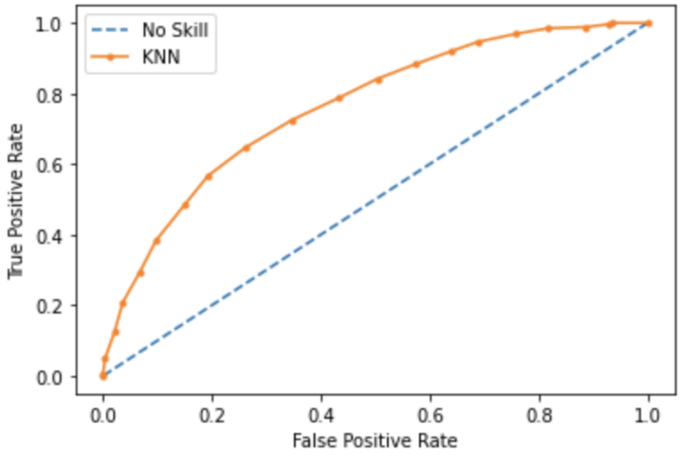
Negative which is a non-bankrupt case. The ROC curve indicates how sharply the model can discriminate between the bankrupt class and non-bankrupt class with an AUC value of 0.76




```
              precision    recall  f1-score   support

           0       0.82      0.69      0.75       524
           1       0.73      0.85      0.79       524

    accuracy                           0.77      1048
   macro avg       0.78      0.77      0.77      1048
weighted avg       0.78      0.77      0.77      1048

MCC: 0.5459214423915012
```

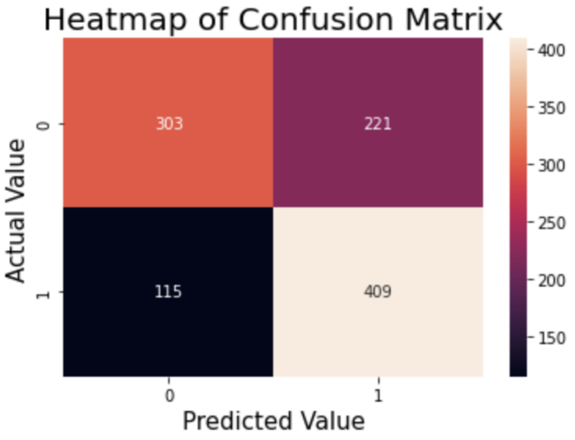


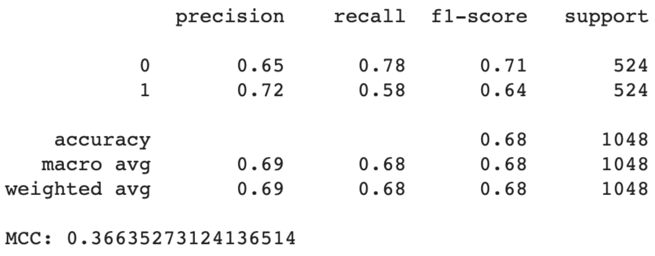
```
No Skill: ROC AUC=0.500
KNN: ROC AUC=0.762
```



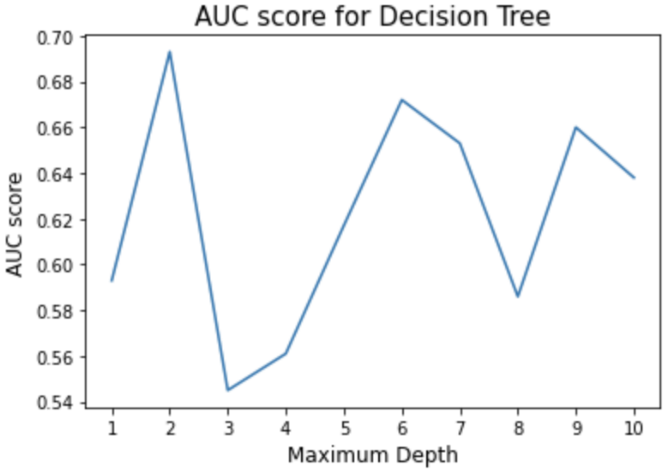
5) Decision Tree Classifier

A grid-search was done changing the maximum depth of the tree model and the best model was obtained when the max_depth = 2, impurity method = Gini with the accuracy of 0.68. The sensitivity value of 58% indicates it was able to correctly classify the True Positive which is a bankrupt case. The specificity value of 78% indicates it was able to correctly classify the True Negative which is a non-bankrupt case. The ROC curve indicates how sharply the model can discriminate between the bankrupt class and non-bankrupt class with an AUC value of 0.693.

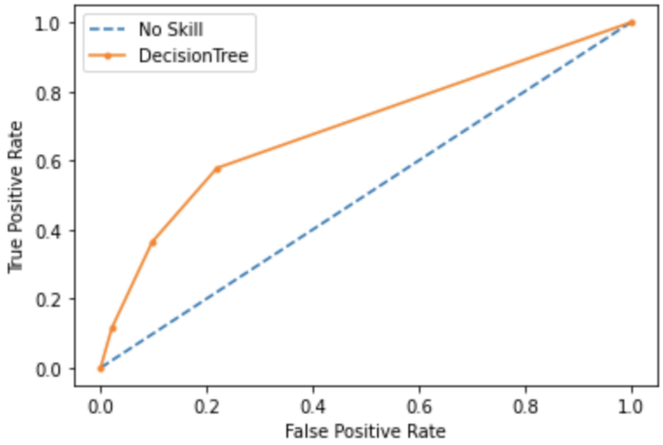




```
              precision    recall  f1-score   support

           0       0.65      0.78      0.71       524
           1       0.72      0.58      0.64       524

    accuracy                           0.68      1048
   macro avg       0.69      0.68      0.68      1048
weighted avg       0.69      0.68      0.68      1048

MCC: 0.36635273124136514
```



```
No Skill: ROC AUC=0.500
Decision Tree: ROC AUC=0.693
```



7. Project Results:

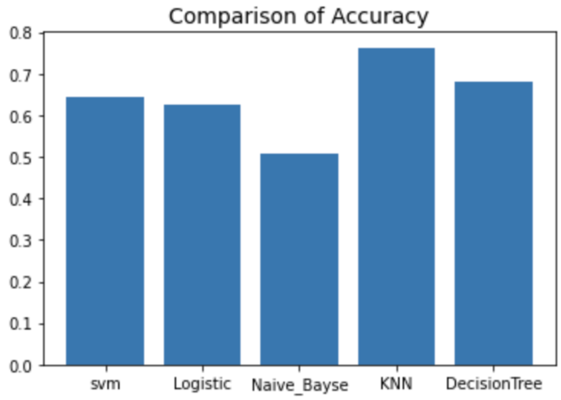
- Accuracy, Precision and Sensitivity were used as performance evaluation metrics to compare the performance between 5 classification models.

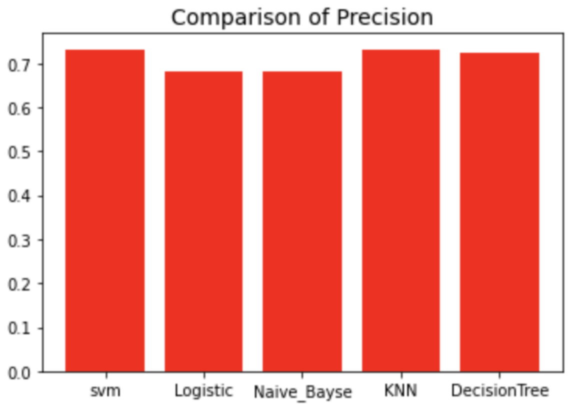
Accuracy = (TP + TN) / (TP +FP + TN + FN)

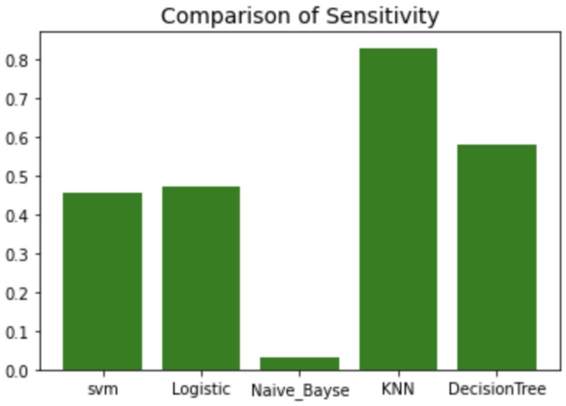
Sensitivity = TP / (TP + FN)

Precision = TP / (TP + FP)

- **KNN** shows the best performance for all 3 indicators. (Accuracy : 0.762 , Sensitivity : 0.828, Precision : 0.732)
- **Naive Bayes** shows the worst performance for all 3 indicators. (Accuracy : 0.508, Sensitivity : 0.032, Precision : 0.68)







8. Conclusion

This project can be used as a baseline for predicting bankruptcy in firms for other countries. Furthermore, using the complete dataset may also result in better performance.

Since the dataset used contained synthetic variables calculated from accounting ratios; future work can have more complex synthetic features with other indicators like macroeconomic performance, lending rates and banking ratios.

Other techniques like boosting, Synthetic Minority Oversampling Technique (SMOTE) and KNN imputation can be used while data mining to improve model performance.

References

Aditya Narvekar, Debashis Guha. Bankruptcy prediction using machine learning and an application to the case of the COVID-19 recession[J]. Data Science in Finance and Economics, 2021, 1(2): 180-195. doi: 10.3934/DSFE.2021010

Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011

Markham, K. (2014, 3 25). *Simple guide to confusion matrix terminology*. Retrieved 2 7, 2019, from Data School: https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/

Atiya, A. F. (2001). Bankruptcy prediction for credit risk using neural networks: a survey and new results. IEEE Transactions on Neural Networks, 12(4), 929:935.