COMP 5411 FA

Fall 2019

Final Project Report

Project Title: Bankruptcy prediction using Supervised Learning methods.

Group Members:

Name	Student No.	Section
Nishchay Trivedi	1106924	FA
Nidhi Patel	1095526	FA
Praneeth Ramesh	1115125	FA

Abstract

Background: Predicting the bankruptcy of a company can have many socio-economic factors. Bankruptcy early warning systems have become tools of key importance in order to guarantee the stability of the economy. Predicting bankruptcy of a company can not only help assess the financial conditions of a company but, also help in forecasting the future of a company.

Objective: This study uses different classifiers to perform a comparative analysis of the performance, sensitivity and specificity from a dataset which consists of information collected from Polish companies during the years 2007 - 2013. We classify the companies if they are likely to get bankrupt or not based on the given information. The models have been trained with a subset of the dataset for training and testing. We calculate the accuracy, sensitivity and specificity with the test dataset using Support Vector Machine, Random Forests and Naive Bayes classifiers.

Methods: To predict the classification of a given company whether it is bankrupt or not, three different classifying methods are used with stratified cross validation of 5 and 10 folds. 1) SVM with three different kernels; Linear, Radial and Polynomial 2) Random Forest and 3) Naive Bayes using Gaussian NB. We imputed the missing values in the dataset using Weighted - KNN. Training and Testing were performed using stratified cross validation with both 5 and 10-fold. Overall prediction accuracy, specificity and sensitivity are calculated for the dataset using the classifiers SVM, Random Forests and Naive Bayes. Class Imbalance issue was resolved by using Synthetic Minority Oversampling Technique (SMOTE).

Results: Classifying the dataset using SVM with Linear Kernel and Radial Kernel we get an accuracy of around 70% while with Polynomial Kernel we get the best accuracy of 86.55% and 89.44% for 5-fold and 10-fold. We are also getting similar results within the kernels for both 5-fold and 10-fold stratified cross validation.

Sensitivity and Specificity are around 70% for all Linear and Radial while the polynomial kernel has a higher sensitivity of 90.06% and 93.29% for 5-fold and 10-fold. Our random forest performed best with 150 number of trees with an accuracy percent of 94.5% with high sensitivity. The accuracy measures where similar for both 5-fold and 10-fold stratified cross validation. We also performed our analysis with Gaussian Naive Bayes classifier but, it performs poorly compared with SVM and Random Forest with an accuracy of 72% and average specificity and sensitivity. All these analyses were performed after using SMOTE to deal with class imbalance. The prediction model based on Random Forests performs the best consistently with an accuracy of 95% in both 5-fold and 10-fold stratified cross validation.

Conclusion:

We have performed a comparative analysis for predicting bankruptcy of companies using three different types of classifiers; SVM, Random Forests and Naive Bayes. We performed the analysis for the dataset after imputing the missing values in the dataset using Weighted KNN and solved the imbalance in the dataset using SMOTE and split the dataset with 5-fold and 10-fold stratified cross validation.

Introduction

Problem Definition

Predicting the bankruptcy of a company has many socio-economic factors, entire communities, business men, investors and government policies can be influenced by the running of a small or big company. Bankruptcy early warning systems have become tools of key importance in order to guarantee the stability of the economy, as a consequence of their potential to avoid losses to stockholders, creditors, managers and other interested parties.

Predicting bankruptcy of a company can help assess financial conditions of the company while also help forecasting the future of a company in the market. It is a vast area of finance and econometrics that combines expert knowledge about the market and historical data of successful and unsuccessful companies. The aim of predicting the future of a company is to develop a predictive model that combines various econometric measures that lets us foresee the financial condition for the coming years, and for the decision makers to take preventive measures and course correct to get the company back in the right track.

In our project, we have analyzed a dataset that contains bankruptcy data of Polish companies. The dataset contains bankrupt companies analyzed from 2000 – 2012, while the companies that are not bankrupt were analyzed from 2007 - 2013. In this dataset, 5 different classification cases are distinguished based on the forecasting period. We are using the attributes in the data to classify a given company as either bankrupt or not.

Significance of the Problem

The economic meltdown in the past decade, has increased the need for tools to predict bankruptcy in order to avoid such devastating events in the future. Bankruptcy of companies affects the market in multiple fronts, and hence, the need to predict bankruptcy among companies by monitoring multiple variables take on an added significance.

Over the years many researchers have to tried to predict bankruptcy of a company by proposing methods based on statistical hypothesis, statistical modelling and much more recently on Artificial Intelligence but, the economic indicators of a company are generally proposed by domain experts and combining them into a usable model is difficult. Another issue when dealing with Bankruptcy is that historical observations used to train and test a model are imbalanced as there are lot more successful companies that those that are bankrupt thereby, creating an illusion which may make the model predict a company as successful even when some of them have distressing factors.

Literature Review

Since the late 1960s, many approaches to bankruptcy prediction systems have been proposed. Most of them share the common feature of relying on multivariate statistical/intelligent techniques whose input variables are mainly financial descriptors of the credit applicant. Although several bankruptcy predictions models have been developed which rely on market information and experts' decision analysis, the available evidence suggests that these alternative approaches have not significantly outperformed multivariate-based techniques. The majority of research efforts during the last three decades have been devoted to test the accuracy of several kinds of classifiers, proposed by researchers in the fields of statistics or artificial intelligence.

The typology by Blazy and Combier (1997) suggests that major causes of bankruptcy could be accidental like malfeasance, death of a leader or fraud, market problems due to market share, lack of customers or financial threats or macroeconomic factors like increased competition, declining demand, and strategy. However, the failure of a company cannot be attributed only to the financial documents but, other parameters should be taken into account as well.

In "Ensemble Boosted Trees with Synthetic Features Generation in Application to Bankruptcy Prediction" the author proposes a novel approach for Bankruptcy Prediction using Extreme Gradient Boosting for learning an ensemble of decision trees. To reflect higher order statistics and impose a prior knowledge about data representation, the authors have introduced a new concept called synthetic features. Synthetic features is a combination of the econometric measures using arithmetic operations. Each synthetic feature can be seen as a single regression model.

How the proposed project matches/differs with previous work

In the previous work they have analyzed and predicted bankruptcy using Ensemble Boosted Trees and by introducing a new concept called synthetic features. In this project, we are analyzing the bankruptcy of companies using SVM, Naive Bayes and Random Forests algorithms and produce a comparative analysis between the various models.

What has been achieved in this project?

We have performed a comparative analysis of three different supervised learning methods namely Support Vector Machines, Random Forest and Naïve Bayes. We perform our analysis on a dataset that has financial information of Polish Companies between the years 2007 – 2013, and after training the model we try to predict using the given data if a company is bankrupt or not.

Data

Data source

We got the dataset from UCI machine learning repository. Originally, the data was collected from Emerging Markets Information Service. The dataset is about bankruptcy prediction of Polish companies. The bankrupt companies were analyzed in the period 2000-2012, while the still operating companies were evaluated from 2007 to 2013. This data set can be found here.

Data description

This dataset contains 64 features and more than 43K instances. Attributes are considered based on financial factors like profits, sales, assets, inventory etc. Every attribute in the datasets are of continuous type. There are no categorical data in our dataset. And the dataset is multivariate that means analysis is based on more than two variables per observation. Based on the collected data it is distinguished in five classification cases. The attributes can be described as below:

1. net profit / total assets
2. total liabilities / total assets
3. working capital / total assets
4. current assets / short-term liabilities
5. [(cash + short-term securities + receivables - short-term liabilities) / (operating expenses - depreciation)] * 365
6. retained earnings / total assets
7. EBIT / total assets
8. book value of equity / total liabilities
9. sales / total assets
10. equity / total assets
11. (gross profit + extraordinary items + financial expenses) / total assets
12. gross profit / short-term liabilities
13. (gross profit + depreciation) / sales
14. (gross profit + interest) / total assets
15. (total liabilities * 365) / (gross profit + depreciation)
16. (gross profit + depreciation) / total liabilities
17. total assets / total liabilities
18. gross profit / total assets
19. gross profit / sales
20. (inventory * 365) / sales
21. sales (n) / sales (n-1)
22. profit on operating activities / total assets
23. net profit / sales
24. gross profit (in 3 years) / total assets
25. (equity - share capital) / total assets
26. (net profit + depreciation) / total liabilities
27. profit on operating activities / financial expenses
28. working capital / fixed assets
29. logarithm of total assets
30. (total liabilities - cash) / sales
31. (gross profit + interest) / sales
32. (current liabilities * 365) / cost of products sold
33. operating expenses / short-term liabilities
34. operating expenses / total liabilities
35. profit on sales / total assets
36. total sales / total assets
37. (current assets - inventories) / long-term liabilities
38. constant capital / total assets
39. profit on sales / sales
40. (current assets - inventory - receivables) / short-term liabilities
41. total liabilities / ((profit on operating activities + depreciation) * (12/365))
42. profit on operating activities / sales
43. rotation receivables + inventory turnover in days

44. (receivables * 365) / sales
45. net profit / inventory
46. (current assets - inventory) / short-term liabilities
47. (inventory * 365) / cost of products sold
48. EBITDA (profit on operating activities - depreciation) / total assets
49. EBITDA (profit on operating activities - depreciation) / sales
50. current assets / total liabilities
51. short-term liabilities / total assets
52. (short-term liabilities * 365) / cost of products sold)
53. equity / fixed assets
54. constant capital / fixed assets
55. working capital
56. (sales - cost of products sold) / sales
57. (current assets - inventory - short-term liabilities) / (sales - gross profit - depreciation)
58. total costs /total sales
59. long-term liabilities / equity
60. sales / inventory
61. sales / receivables
62. (short-term liabilities *365) / sales
63. sales / short-term liabilities
64. sales / fixed assets

Table 1: The table below shows the data for every forecasting period

	First year	Second year	Third year	Fourth year	Fifth year
Bankrupted	271	400	495	515	410
Survived	6756	9773	10008	9227	5500
Total	7027	10173	10503	9792	5910

Methods & Tools

Preprocessing of the data:

Techniques that are used prior to the application of data mining method is known as Data Preprocessing. We had multiple missing values in the dataset. First, we normalized the dataset. For normalization, we have used Z-Score Normalization in which the values of each feature in the data set has zero mean and unit variance. We calculated the Z-Score standardization by determining the distribution mean and standard deviation of each feature and subtracting the mean from each feature and dividing the result by the standard deviation.

Then, we imputed missing values in the dataset by using Weighted KNN imputation method. In Weighted KNN, each nearest k value is assigned a weight using the kernel function. A common implementation of the kernel function is the inverse distance function, where the inverse of the distance is used as weight to impute using weighted KNN. To achieve this task, we have used function named “KNNImputer”. The function KNNImputer is from library “missingpy”. We have taken value of K as 5.

Validation Method on the dataset:

The validation method that we have used in our project is stratified K fold Cross Validation technique to maintain the ratio of target class values. Cross-validation is a statistical method used to estimate the skill of machine learning models. It helps in an appropriate way to select an apt model for the classification. The values which we used for stratified K-fold cross-validation are 5 and 10.

We have used a class called “StratifiedKFold” from library “sklearn.model_selection”. In this class, we have taken parameter “n_splits” as 5 and 10.

Method to avoid Class Imbalance:

The dataset we have used is highly imbalanced as majority of values in target class are 0 (not bankrupt). We have used Synthetic Minority Oversampling Technique (SMOTE) to deal with class imbalance. SMOTE adds some synthetic instances from minority class which are created by using an algorithm. For implementing SMOTE, we have used a class called “SMOTE” from library “imblearn.over_sampling”. We have applied it on training data in each fold by using method “fit_sample” before applying any supervised learning methods.

Supervised Learning Methods applied on the dataset:

The supervised learning methodologies that we have implemented in our project are Support Vector Machine, Random Forest and Naïve Bayes Algorithms.

- **Support Vector Machine:**
  The Support Vector Machine (SVM) is a supervised learning technique that classifies the data points into two classes based on a hyper plane. As our dataset is based on bankruptcy and the class variable for a company is either bankrupt or not bankrupt, we tried to implement this algorithm on the dataset.

  To implement this method, we used “SVC” class which is known as support vector class from library “svm” that is contained by Scikit-Learn in python. We trained our training data by using “fit” method and made predictions on the testing data by using “predict” method. For SVM, we tried three different kernels on our dataset which are linear, polynomial and gaussian.

    - Linear Kernel: Linear kernel classifies data that are linearly separable. The linear kernel equation for predicting a new input is considered as the dot product of the input data value which is x and the support vector value which is xi given by:

      $$f(x) = B(0) + sum(ai * (x,xi))$$

      To implement Linear Kernel, we set parameter kernel as “linear” in SVC.

    - Polynomial Kernel: Polynomial kernel represents the similarity of vectors in a feature space over polynomials of the original variables, allowing learning of non-linear models. It can be written as:

K(x,xi) = 1 + sum(x * xi)^d

To implement Linear Kernel, we set parameter kernel as “poly” in SVC.

- Gaussian Kernel: Gaussian kernel, is a kernel that is in the form of a radial basis function. It is also known as Radial Basis Function kernel. It can be written as:

  K(x,xi) = exp(-gamma * sum((x — xi²))

  To implement Linear Kernel, we set parameter kernel as “rbf” in SVC.

- **Random Forest:**

  Random forests are an ensemble learning method for classification. In Random Forests we use multiple decision trees and based on majority voting we choose class label. The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners. This bootstrapping procedure leads to better model performance because it decreases the variance of the model, without increasing the bias.

  We have implemented Random Forest by using class “RandomForestClassifier” which is from “ensemble” library. In our code, we have performed hyper parameter tuning to choose number of trees in random forests. To achieve this task, we used “RandomizedSearchCV” class which is from library “model_selection”. Both the libraries are contained by contained by Scikit-Learn in python. We trained our training data by using “fit” method and made predictions on the testing data by using “predict” method. In this, we have taken parameters in RandomForestClassifier which we got after hyper parameter tuning.

- **Naïve Bayes:**

  The naïve Bayes algorithm is based on Bayes theorem. Here every attributes of a data set are assumed as an independent variable. The Bayes theorem is indicated by the below equation:

  $$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

  Where, p(a|b) is defined as probability of a such that the probability of the event b is known. It is also called as Posterior probability. p(b|a) is the probability of event b to happen based on the already happened event a’s probability. It is also called as Likelihood. P(a) and P(b) are the independent probabilities of a and b events. As P(b) is independent of class variable we can remove that in calculation.

  We are using Gaussian Naïve Bayes Classifier for our classification as the dataset that we have used is contained with continuous values in each feature and the Gaussian Naïve Bayes Classifier is especially for the features which have continuous values.

  $$p(x = v \mid C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(v-\mu_k)^2}{2\sigma_k^2}}$$

  Where, μ_k is mean and σ_k is standard deviation of values in feature x with class k.

  To implement the Gaussian Naïve Bayes, we have used “GaussianNB” class which is from “naive_bayes” library that is also contained by Scikit-Learn. We trained our training data by using “fit” method and made predictions on the testing data by using “predict” method.

Results:

Predicting Bankruptcy

Initially our dataset was divided in five different years based on different forecast periods. For each year we used three different classification methods and analyzed the results by measuring accuracy, sensitivity and specificity. A comparative analysis for all five years is shown for different classifiers and validation methods in below figures. Figure 1 shows the result of SVM with 'poly' kernel with 5-fold stratified cross validation while Figure 2 shows the result for 10 folds.

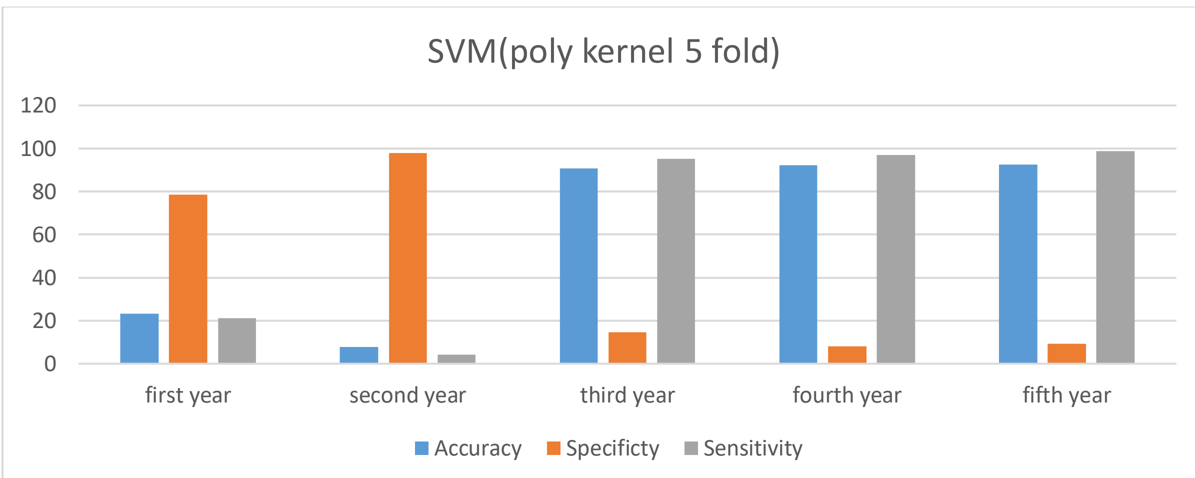


Figure 1

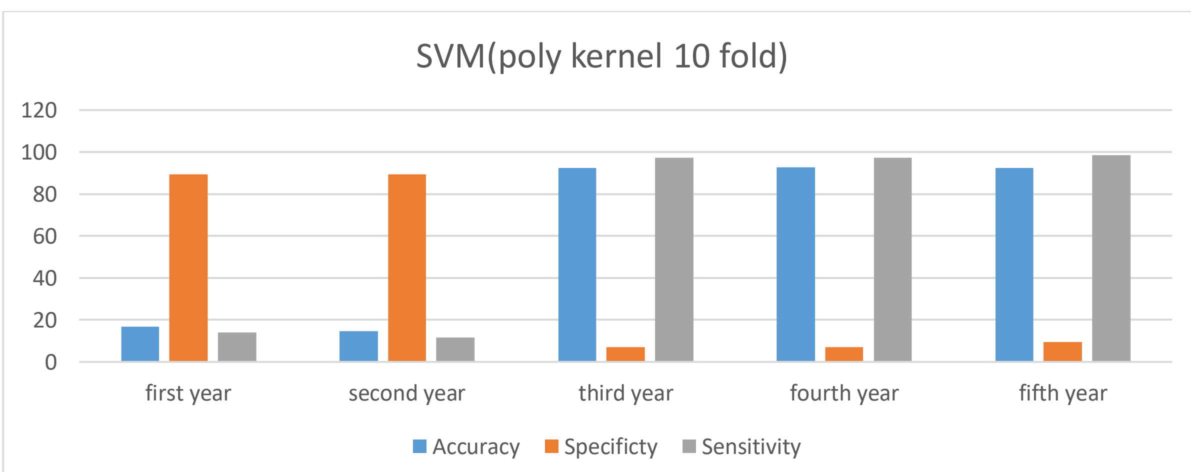


Figure 2

Figure 3 shows the result of Random forest with 150 trees with 5-fold stratified cross validation while Figure 4 shows the result for 10 folds.

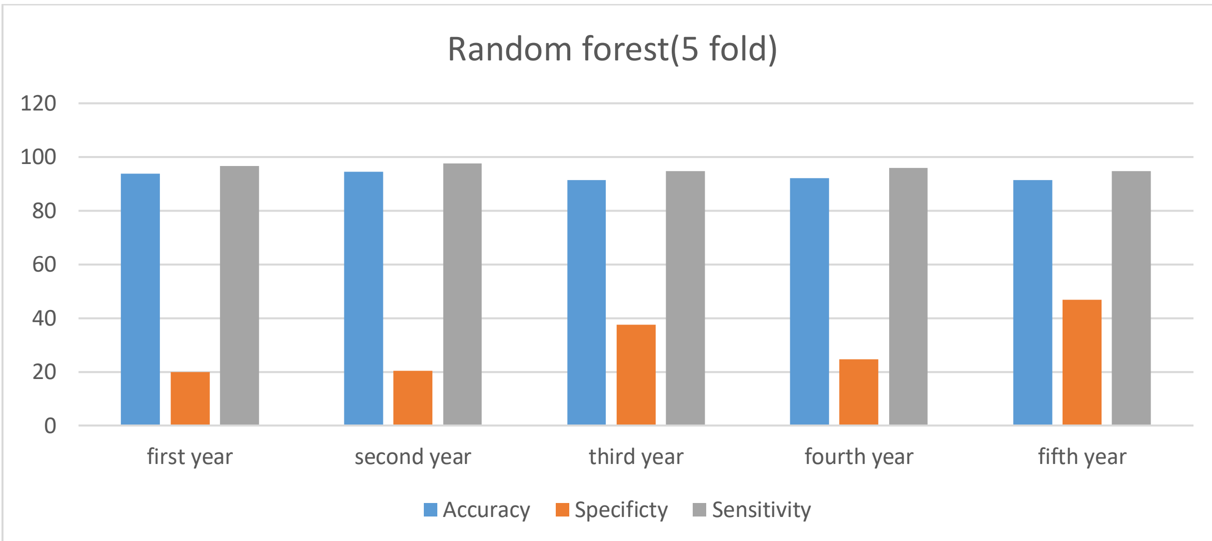


Figure 3

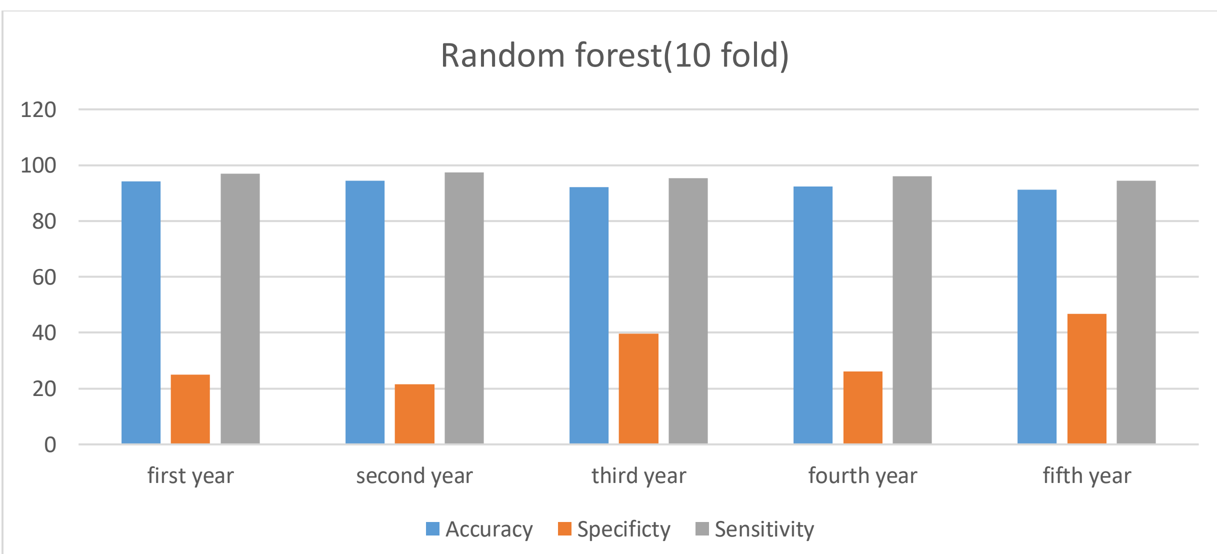


Figure 4

Figure 5 shows the result of Naïve Bayes with Gaussian version using 5-fold stratified cross validation while Figure 6 shows the result for 10 folds.

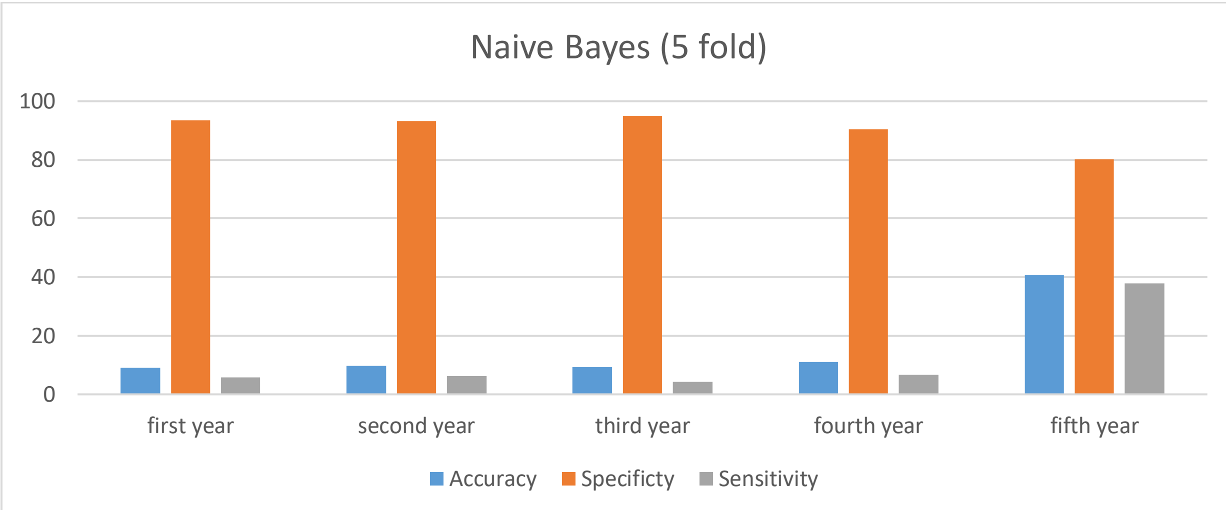


Figure 5

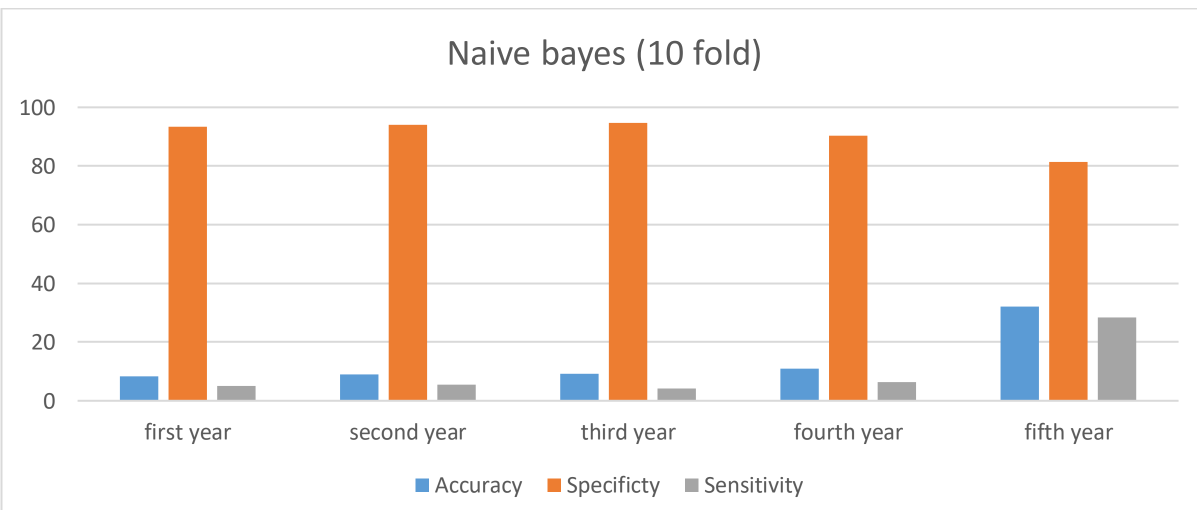


Figure 6

Later we combined all year dataset in one as all_years.csv and analysed the results using the same methods we used for each year. For this dataset too, we used 5 and 10-fold stratified cross validation. The results of each classifier are summarised in below tables.

Table 2 Results of SVM with poly kernel

	Accuracy	Sensitivity	Specificity
5 fold	86.55	90.06	17.31
10 fold	89.44	93.29	13.34

Table 3 Results of Random forest with 150 number of trees

	Accuracy	Sensitivity	Specificity
5 fold	94.5	97	46.1
10 fold	94.55	96.9	46.4

Table 4 Results of Naive Bayes with Gaussian version

	Accuracy	Sensitivity	Specificity
5 fold	71.6	72	64.2
10 fold	72.1	72.5	63.9

The graph below shows the difference with different kernels used with SVM (Figure 7)

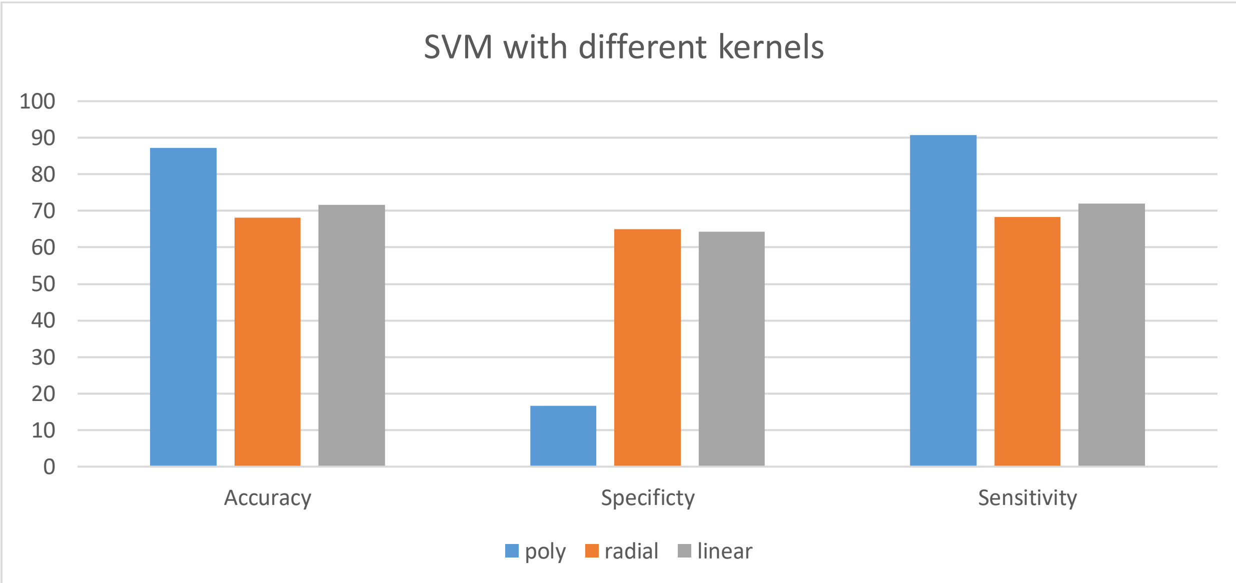
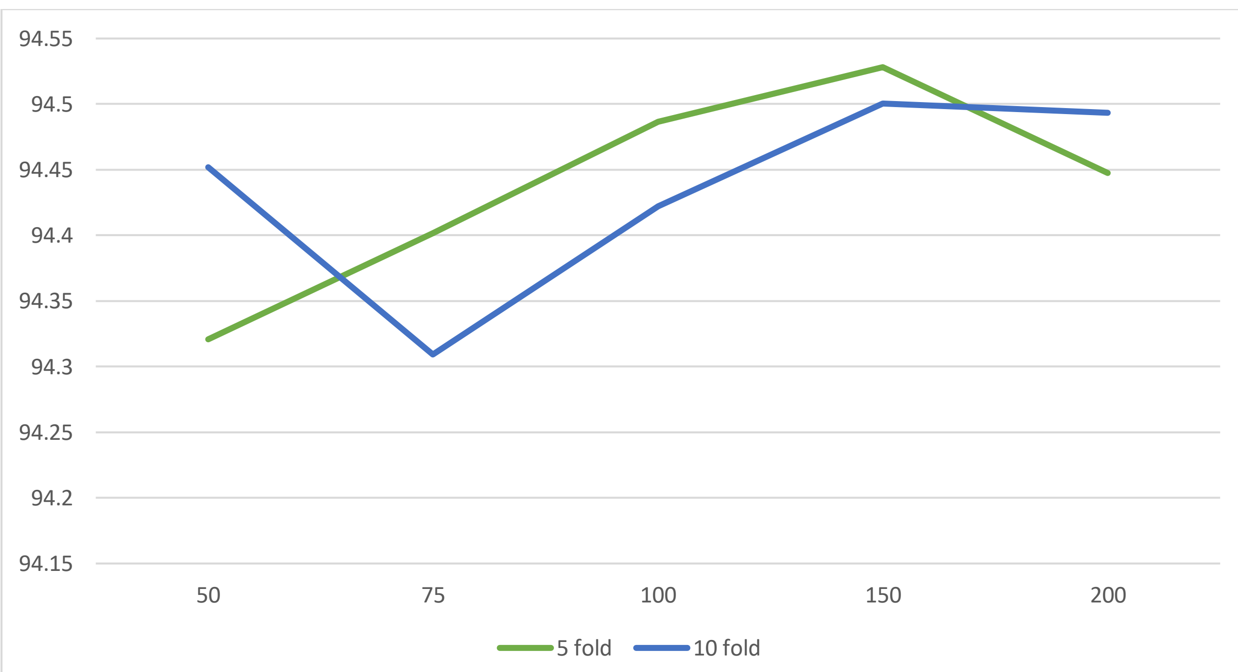


Figure 7

For random forests, we used hyper parameter tuning to find the best number of trees to achieve maximum accuracy, the results of the hyper-parameter tuning are presented below.



From the results it is clear that Naïve Bayes is performs very poor on this data, the reason behind the poor performance in predicting if a given company is bankrupt or not even though the specificity of the model very high can be analyzed using a confusion matrix as described below. For our dataset we extracted the confusion matrix and observed the below values.

True Positive: 250

True Negative: 40

False Positive: 15

False Negative: 8000

The calculated Specificity for our dataset with Naïve Bayes is very high because we have very few instances of *True Positive* compared to *False Negative.* The Sensitivity will be low for our dataset because we comparatively have more numbers of instances with *False Negative* compared to *True Negative*. Accuracy also remains low because the number of *True Negative* and *True Positive* are very less which are used to determine the accuracy.

On the positive end of the spectrum, Random forest performs really well compared to the other two classifiers. Random Forest not only produces the best accuracy but, also is computationally faster. While SVM in comparison has an accuracy percentage of about 90% with average sensitivity and specificity.

In the first iteration of our project, we decided to combine all of the prediction year's data and then run supervised learning methods to it but, after further analysis we came to a conclusion that the data is divided into different forecasting period and combining those individual year prediction data would make the data non homogeneous. We ran the supervised learning algorithms on each individual year to achieve our results.

Discussion and Conclusion:

Summary of What has Been Done:

We analyzed a dataset that consists of financial information of Polish companies during the years 2007 - 2013. We are performing pre-processing on the dataset to impute the missing values in the dataset using Weighted-KNN. We are also resolving the imbalance that is present in the dataset because of the presence of more instances with not bankrupt status compared to those that are bankrupt by implementing Synthetic Minority Oversampling Technique (SMOTE). We use stratified cross validation technique of 5 fold and 10 fold to split the dataset into training and testing. We then perform a comparative analysis by building models using various classifiers to test our dataset to predict the possibility of bankruptcy based on the available data. We first implement SVM with three different kernels; Linear, Radial and Polynomial kernels to calculate the accuracy, specificity and sensitivity of the predictions. We then find the number of tress required to generate random forests using a simulation and generate accuracy, specificity and sensitivity of the model and finally we implement Gaussian Naive Bayes to calculate the values of accuracy, specificity and sensitivity of the model. All the models generated for the purpose of classification has been implemented using SMOTE and 5 and 10 fold cross validation.

Summary of the Results:

After performing our classification using SVM with Linear kernel we generated an accuracy of 71.6%, Sensitivity and Specificity values of 72% and 64%. Classification using SVM and Radial kernel generated similar results to Linear kernel whereas, SVM using Polynomial kernel generated a higher accuracy of 90% and better specificity and sensitivity values.

For Random Forests, the model consistently had a high accuracy percent of 94.5% for both 5-fold and 10-fold stratified cross validation with a sensitivity value of 97% and an average specificity of 47%.

The Gaussian Naïve Bayes performed the most poorly of the three supervised learning algorithms. The Gaussian NB model had an accuracy percent of only 72%, while sensitivity and specificity of the model was on an average about 72% and 64%.

Performing a comparative analysis on three different supervised learning algorithms; SVM, Random Forests and Naïve Bayes has helped us compare the performance of each model under similar circumstances and help give us an idea of what kind of model would work well in predicting bankruptcy in any other dataset. For this particular dataset, Random Forest consistently generated the best accuracy, sensitivity and specificity values.

Implication of the Results:

The implication of our analysis is to help investors and stake holders have an idea that data science can help in predicting with a high level of accuracy if their investment in a company is stable or not. Our analysis of data from Polish Companies has helped us predict if a given company is bankrupt or not using Random Forest with a high level of accuracy. The dataset is balanced and has been pre-processed to help avoid bias while predicting results.

Limitations:

The dataset consists of 64 features that have many details about the company's assets and profits which are details that are not readily available to run analysis on. The same set of features may not be available to help predict if a given company is bankrupt or not in a different dataset. Understanding the features that need to be used to help the classifier consider it as an important feature is a complicated problem which requires the input of domain experts, various stakeholders and data scientists.

Future Work:

We are planning to understand the features in detail and implement various techniques for feature selection and then run the models through the classifiers to improve the accuracy of our models.

References:

1. For dataset: https://archive.ics.uci.edu/ml/datasets/Polish+companies+bankruptcy+data
2. https://www.sciencedirect.com/science/article/abs/pii/S0950705111002413
3. Zieba, M., Tomczak, S. K., & Tomczak, J. M. (2016). Ensemble Boosted Trees with Synthetic Features Generation in Application to Bankruptcy Prediction. Expert Systems with Applications
4. Chen, T., He, T., 2015b. xgboost: eXtreme Gradient Boosting. R package version 0.3-0. Technical Report
5. Cortes, C., Vapnik, V., 1995. Support-vector networks. Machine learning 20, 273–297
6. Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., Herrera, F., 2012. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 42, 463–484

Appendix:

To execute the program, please follow the below instruction.

For our analysis we have used python 3.0+ version. In the environment the previously described library should be installed that can be done using simple command – pip install ‘library_name’.

Please follow the below commands to install libraries.

1) If you don’t have python installed, download python 3.0+ version and install it. Also put it under the path variable.
2) Install NumPy with **pip install numpy.**
3) Install Pandas with **pip install pandas**.
4) Install missingpy with **pip install missingpy**.
5) Install sklearn with **pip install sklearn**.
6) To install library for smote use this command: **pip install imblearn**.

Next, if any IDE is used like PyCharm or Anaconda just open Project.py file and hit the run button, else type the command – python Project.py that will run the code from command prompt.

Our dataset is divided into five years, so if you want to get results for just the first year enter 1 in the prompt question asked for which year dataset you want to take. As mentioned in the below image, if Input is taken as 0 so it will perform on all_years dataset and 1 – 5 it will perform analysis on that respective year’s dataset.

```
print("Accuracy:", nb_acc/k_fold )
print("Specificty:", nb_spe/k_fold )
print("Sensitivity:", nb_sen/k_fold )
```

```
Which year dataset you want to take ( from 1-5 and to get dataset of combined all years enter 0)0
```

For the user input for K fold stratified cross validation, the prompt will ask for K value which is for cross validation to mention how many folds you want to run e.g. if you want 5 - fold cross validation, just enter 5 as shown in below prompt.

```
Which year dataset you want to take ( from 1-5 and to get dataset of combined all years enter 0)0
Enter k value for k-fold cross validation(5 or 10): 5
```

After that it will ask to enter what kind of kernel you want to use for SVM. For polynomial enter ‘poly’, for radial enter ‘rbf’ and for linear enter ‘linear’. Refer below image.

```
Which year dataset you want to take ( from 1-5 or to get dataset of combined all years enter 0)0
Enter k value for k-fold cross validation(5 or 10): 5
For SVM which kernel you want to choose? (for polynomial enter 'poly', for linear enter 'linear', for radial enter ' rbf')poly
```

So as per above given example, it will start performing on the all_years dataset with 5 - fold stratified cross validation for all three classifiers and will show the accuracy, sensitivity and specificity on the console.

For SVM, polynomial kernel gives the best result and only that kernel is kept active in our codebase but, if you want to check the results for linear and radial kernel just uncomment corresponding code and comment other. More details are mentioned in the code.

Note: Before it shows the results, it will give some library warnings, please ignore those warnings.