Classification of Polish Bankruptcy Data Using Support Vector Machines, Random Forests and Neural Networks

Ben Denis Shaffer
University of Michigan

Georgios Spyrou
University of Michigan

Noah Gale
University of Michigan

Contributions of each member: For the general coding and analysis part, all of the three members we worked together in order to clean the data properly, pick the best training and test data sets, perform a dimension reduction and adjust our data in an ideal formation for the further analysis. After this task, each of us focused on one specific main topic of our project. Specifically, Ben Denis Shaffer focused into the analysis of Random Forests, Georgios Spyrou mainly focused on the construction and analysis of the Support Vector Machines and Noah Gale to the analysis of the Neural Networks. Although this is true, after our individual work, we all worked together for all of the tasks on the project as a proof-reading method and improvement of the whole result.

Classification and Prediction of Bankruptcy for Polish Companies

Abstract

For this project we analyzed a dataset about bankruptcies of polish firms. Our goal was to classify the firms as bankrupt or not bankrupt based on their financial data, in a time span of five years. Specifically, we build classifiers using Random Forests, nonlinear Support Vector Machines, and one layer Neural Networks. For the SVMs we used the Gaussian and the Laplacian kernels. Overall our best model was the Laplacian SVM. The performance was evaluated based on the False Positive, False Negative, and Mean error rates on a testing set. Our main finding was that the Random Forests achieved the lowest FNR, SVM achieved the lowest FPR, and Neural Networks were somewhere in between.

Introduction

Bankruptcy prediction is often used to assess the financial stability of a company, and the probability of its continued existence in future market conditions. A corporate bankruptcy can have far-reaching consequences for economic actors as varied as investors, creditors, suppliers, competitors, government entities, and customers. A bankruptcy can create large economic costs for these actors, and so the topic of corporate bankruptcy has been researched in various forms across disciplines in order to understand the causes and signs of a bankruptcy. Econometric and Financial models often use past data and complex models with years of implied theoretical knowledge and empirical observations pertinent to their fields. This can lead to issues, as a corporate bankruptcy can be caused by a firm's inability to raise capital at a crucial point during an economic or sector-specific recession. There are further difficulties with such a model, because while some companies do declare bankruptcy, others simply persist as distressed companies for the studied period, causing an imbalance in the data. Each year, there are often many more companies that survive than companies that declare bankruptcy. Thus it's likely that a model would incorrectly predict far more companies going bankrupt in the observed period than those that actually do go bankrupt.

Description of Data and areas of interest

The objective of our analysis is to build a classification model for bankruptcy of firms based on their financial information. The data we are using can be found on the UCI Machine Learning Repository[1].

Data

Specifically, the data concerns a study made for Polish companies and their probability of bankruptcy according to `64` different financial indicators, such as net profit, gross profit, and working capital, total assets and more. Therefore, by using these `64` variables we are trying to evaluate the financial condition of each company and draw conclusions about its future in the economic sector (specifically we are focusing on the manufacturing sector), while our main goal is the prediction of a possible bankruptcy for the company or not. The period in which we are focusing our research is separated in five different years. The first year regards economic and financial data for `7027` companies, from which the `271` companies ended up to be bankrupt while the rest `6756` avoid the bankruptcy, in the next five years. For the second year we are examining `10173` companies and we end up with a total of `400` bankrupted companies and `9773` that did not

bankrupt, in the next `4` years. The third year contains `10503` firms with `495` of them going bankrupt and `10008` of them not going bankrupt in the next three years. For the fourth year our interest is focusing on `9792` companies and we end up with `515` companies as bankrupted and `9277` as not bankrupted, in the next two years. Finally, the fifth year includes a total of `5910` companies from which `410` ended bankrupt and `5500` did not, in the next year. Our main concern in this project is to try and analyze the aforementioned data that we have for these five years, and by using the total of `43405` different instances of companies and the 64 financial indicators, deduce into some useful findings and results about the future of each instance. For the above-mentioned purpose we found interesting and challenging the fact that we have to construct several different classification models and thus try and conclude to the optimal one, considering always what we are trying to "predict" and what the restrictions of each model are. Because of the fact that we are talking about the cases of bankruptcy versus not-bankruptcy which is a binary-classification case we can try and test many different models and thus try to optimize our results – both in terms of numerical outputs (mean errors etc.) , as well as , in terms of visualization.

Areas of Interest

We are interested in controlling the expression of error types within our models, since it's possible that economic factors may find one error preferable to the other. A creditor attempting to assess the ability of a firm to pay back its debt would likely feel comfortable with incorrectly predicting a large number of firms going bankrupt, and avoiding all of them, so that it can more accurately lend to firms that will not go bankrupt and cost the creditor their investment. A short-seller attempting to bet against the economic well-being of corporations would likely feel comfortable with incorrectly predicting a larger number of companies that will not go bankrupt, as long as they can more surely predict the companies that will in order to maximize their returns.

Methodology

We begin our analysis with data cleaning and dimension reduction using PCA. Specifically, we have to address the missingness and redundancy of information in the variables that we used as predictors. Once we obtained the clean data set with reduced dimensions we began to visually explore the class structure of the data. Visual exploration led us to the decision to not use classification methods such as LDA/QDA, Logistic Regression, or K-NN. Instead we decided to explore how well the data can be classified with the use of Random Forests, Support Vector Machines, and Neural Networks.

Methodology: Missingness

The original dataset has `43405` rows and `66` columns. Some of the columns are ratios and linear combinations of ratios of other columns. Furthermore there are a total of `41322` missing values which account for about `1.44%` of the data. The following code tells us more about where the missing values are.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   1.000   0.952   1.000  41.000
```

We can see that on average every row has `1` missing value. This means that we can't simply remove rows with missing values. Furthermore we observe that there is a row with `41` missing value.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     8.0   114.5   626.1   134.0 18980.0
```

Next we observe that on average every column has `626` missing values and one column has `18980` missing values which accounts for about `44%` or the rows. It is obvious that these extreme values for the number of missing values in a row or column are outliers and that the distributions for missingness could possibly be modeled via a Poisson distribution. Of importance to our analysis however was the question of whether missingness of the data was related to the bankruptcy. Thus we came up with the following approach to dealing with the missing value and proceeding with analysis. We introduced two tolerance measures: 1) First is the tolerance for how many missing values a column has. For example, if a column has more than `200` missing values we would not impute the missing values, but instead drop the variable. 2) The second tolerance measure was the rate of bankruptcy within the missing values of a variable. If there were many more bankruptcies relative to non-bankruptcies within the missing values, we deemed the missingness to be non-random and transformed the variable to an indicator/factor variable. In the data `5.06%` of the firms went bankrupt. We set the second tolerance measure to twice this rate.

It is very important to mention that we applied this approach only to the `14` variables with the greatest number of missing values. In other words, columns with a relatively little number of missing values were not subject to our method of evaluating whether values were missing at random or not, because there wouldn't be sufficient information to make such a claim. Lastly, we used the Predictive Mean Matching (PMM) method for multiple imputation implemented in the `mice` package.

PMM is a method based on linear regression with a slight modification. The method works as follows:

First regress:

$$\hat{Y}_{obs} = X_{obs}\hat{\beta}$$

$$\hat{Y}_{mis} = X_{mis}\beta^*$$

Where β^* is sampled from the sampling distribution of $\hat{\beta}$.

Then compute:

$$\hat{Y}_{mis,i} = |\hat{Y}_{obs} - \hat{Y}_{mis,i}|$$

And out of $\hat{Y}_{mis,i}$ pick at random one of the smallest three differences.

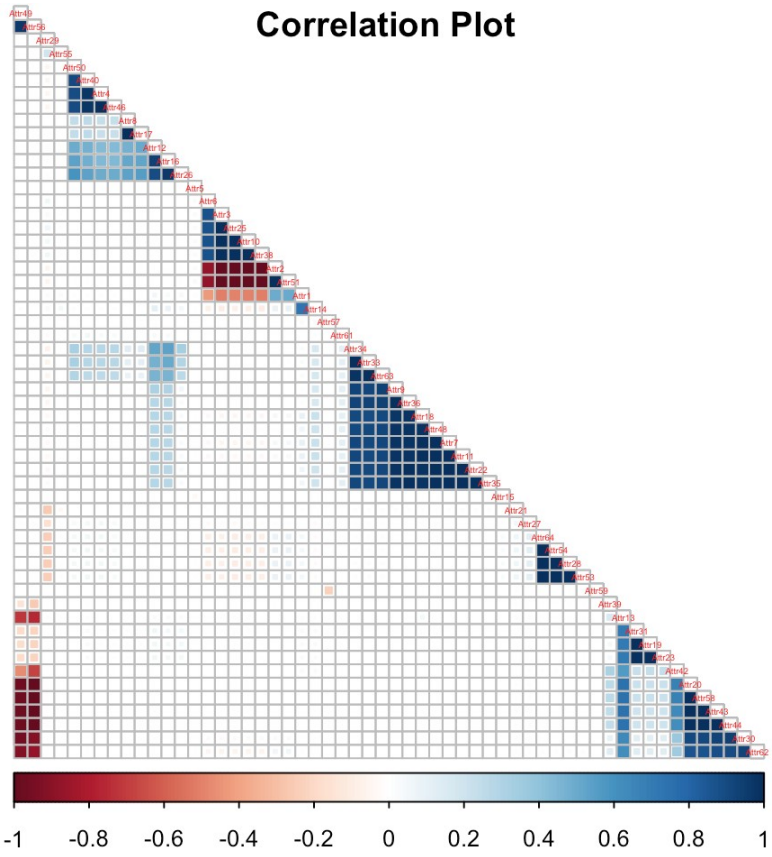
This algorithm doesn't allow for extrapolation outside of the ranges of the variables being imputed which is why we chose to use it in our analysis.

The following function `reduce_impute_polish` implements our approach and produces a clean data-set.

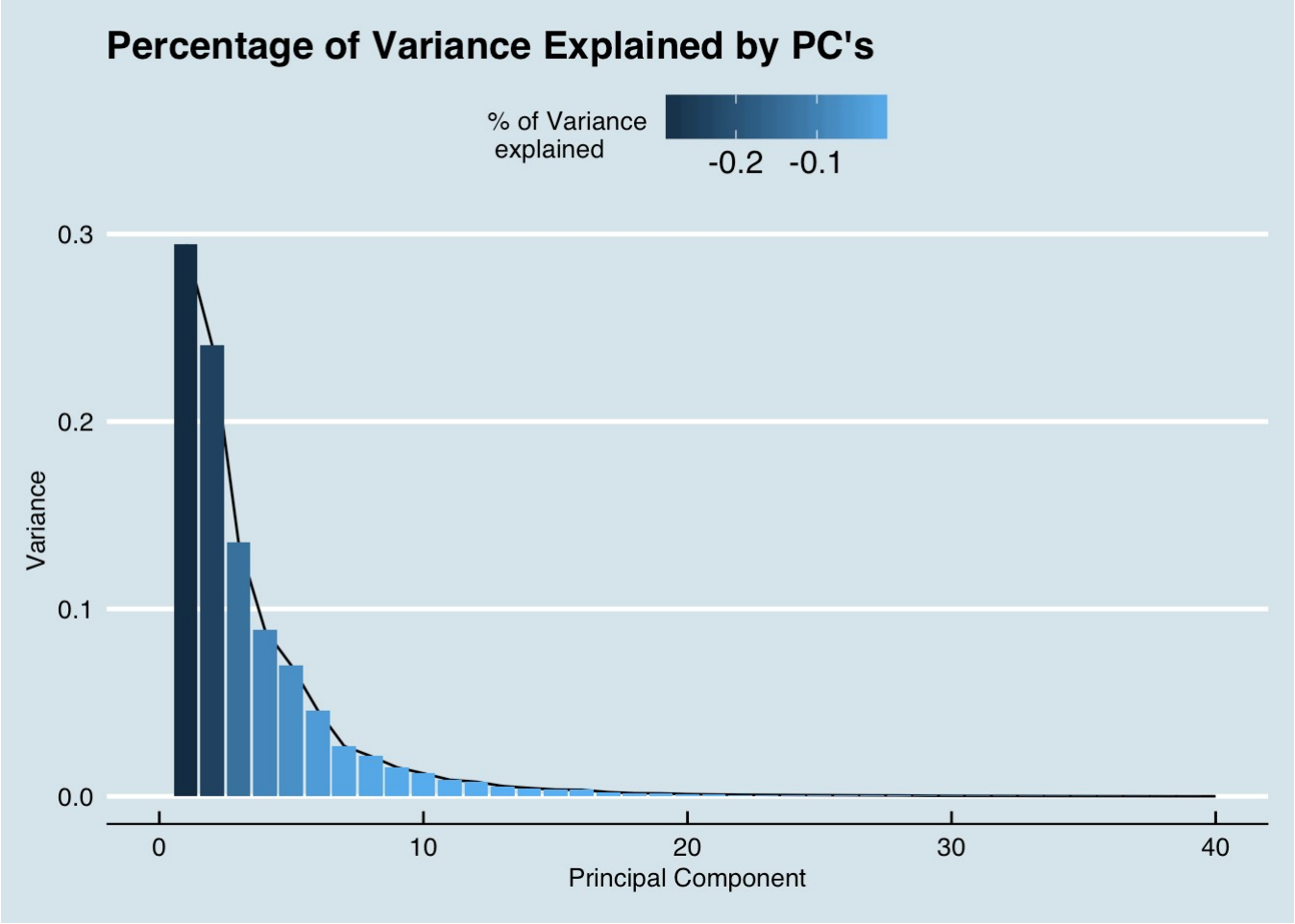
The result is a data-set with `42987` rows, `58` columns, and no missing values, which means that `8` variables were dropped and `418` rows removed. `418` rows that were removed were the ones where we still had missing values after the `pmm` imputation procedure.

Methodology: Dimension Reduction

Due to the nature of the variables we suspected that there was a lot of redundant information and thus decided to apply Principal Component Analysis to reduce the dimensions. The redundancy can be seen on the correlation plot where dark blue regions are highly positively correlated, and dark red regions are highly negatively correlated. We also note that for the purpose our analysis we don't concern ourselves with the interpretability of models, but rather the classification performance. By applying the PCA to our data we wish to extract the Principal Components which are by design uncorrelated, and most importantly help reduce the dimentionaity while optimally preserving maximum variation contained in the data.

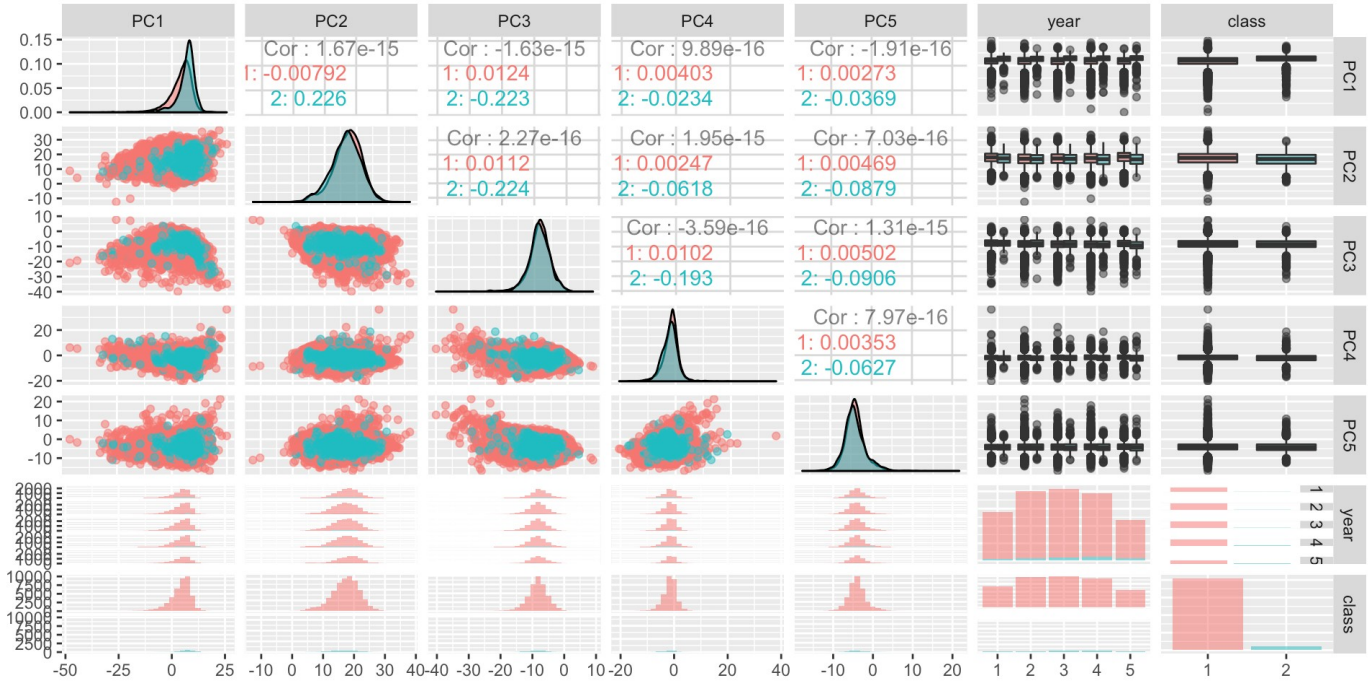


In order to visualize the data we are analysing we introduce a transformation by first squaring the data and then taking the logarithm, which results in a nice scale. We have alse experimented with different transformations such as standardization, however the earlier proposed transfromations nicely centeres the data, which lead us to make this choice. After applying the transformation we proceed to apply PCA.



We can see that the first 5 PCs explain about `83.1%` of the the variance in the data. Thus we will use these as covariates in our analysis instead of the `49` numerical variables that we have in the cleaned data-set. The factor loadings for these PCs did not have any obvious interpretation.

After the dimensional reduction procedure we now have a data-set with `5` continuous covariates, `6` indicator variables for the variables missing not randomly, the response, as well as the year within which the bankruptcy occurred. Finally we can visualize the data to explore the structure of the classes.



From this plot we can see that in all of the dimensions the two classes overlap very strongly, and it is hard to imagine that a linear boundary would separate the two classes well. In fact it appears that the two classes come from the same distribution, and many more extreme values are observed for firms that don't go bankrupt simply because there are many more such cases. Because of this strong overlap between the regions we decided to test classification methods that are non-linear, or can be formulated as non-linear.

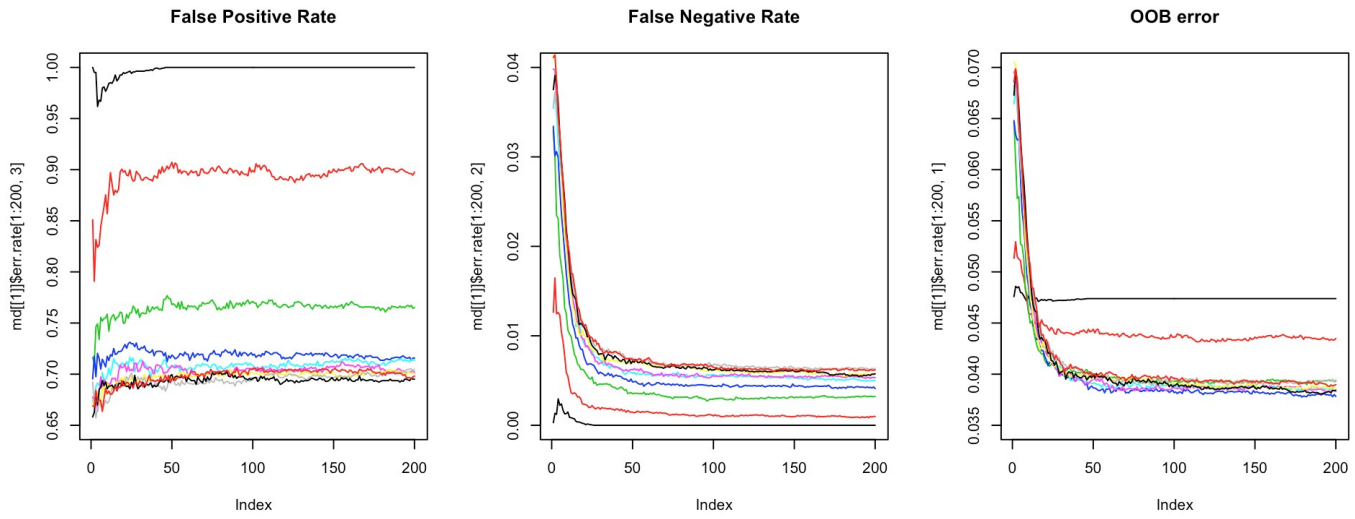
Models

Random Forests

For all of the methods, including Random Forests, that we are using we split that data into a training set for learning the models, and a test set to evaluate its performance. We take `2/3` of the data for training, and the remained for testing. We take a random split of these proportions from each year to preserve the proportion of observation of the years.

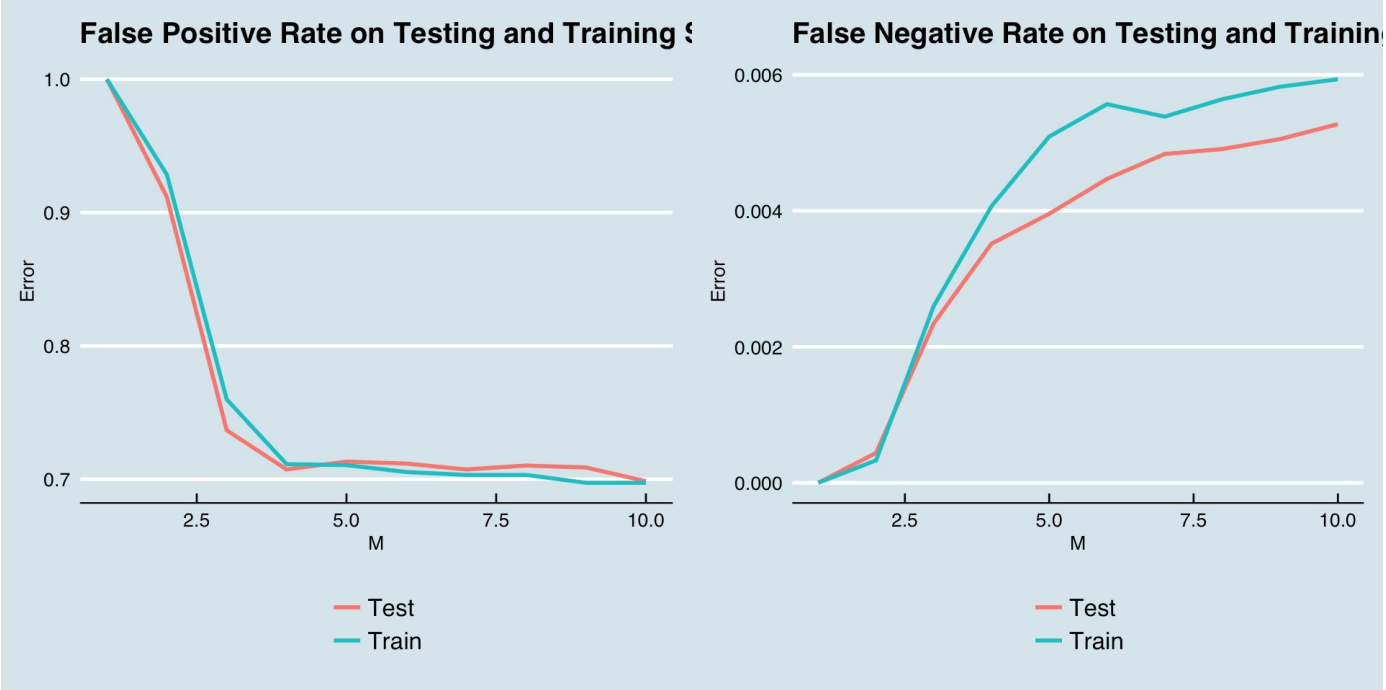
The procedure for fitting the random forests involves fitting many decision trees on a random subset of covariates of size `m`. The size parameter is the parameter that controls the complexity of the model. Another tuning parameter is the number of trees fit and combined into a forest. We don't use this parameter for tuning the model because it turned out to have little if any effect on the classification performance.

We vary the subset parameter from `1` to `10` and compare the error rates for the models trained with various complexities. For each of the trained models we plot and observe how the False Negative Rate, False Positive Rate, and the OOB Rate evolve over the growth of the forest.



We can see that as more trees are combined into a forest the FPR increases and stabilizes, while the opposite is true for the FNR and the overall model accuracy. We also note that as complexity of the model increases the FPR decreases and stabilizes at around `0.70` and the FNR increases and stabilizes around `0.006` . Thus, the random forests are over all accurate, but that is due to the fact that the data has many more observations of non-bankrupt firms. When we look at how well the models classify observations as bankrupt when they are bankrupt, we see that an error is made `70%` of the time at best. This is a very poor result, which we will be able to mitigate by using SVM and NN.

Nonetheless we also look at how well the models of various complexities generalize to the testing set, in terms of the FPR and the FNR. Again, we note that for models of all complexities the overall error rate is about `95%`.



We can see that the models do equally poorly on the testing set as on the training set when it comes to the FPR. For larger complexity of the model the classification of the testing set is a little bit better but not significantly. The FNR worsens with complexity and in fact the FNR is somewhat better on the testing set, but gain not in any significant way.

There is an obvious trade-off between the Type I and Type II errors when fitting the random forests. If this model was to be used we would prefer to use a model with `m` set to `5` or `6` because increasing the complexity beyond doesn't improve the FPR in any noticeable way, but the FNR continues to deteriorate. Overall however random forests did not perform well on our data, and hence we move onto SVM and NN.

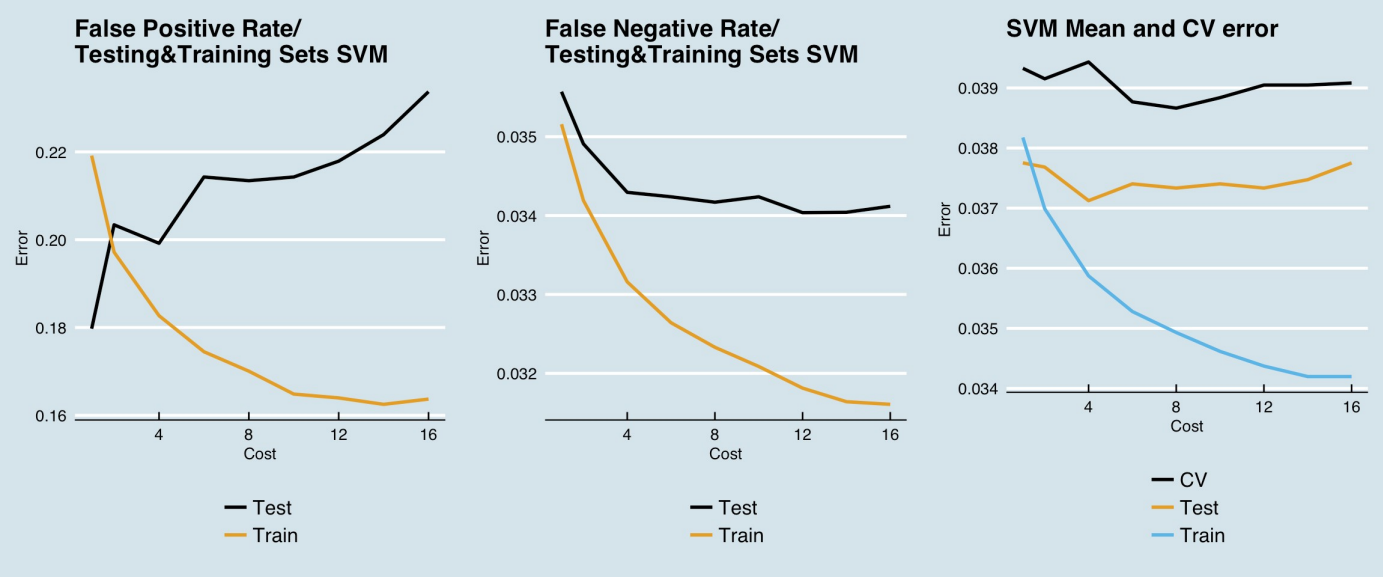
Support Vector Machines

Support Vector Machines is a supervised statistical learning model that finds an optimal separation boundary between classes. For our data it's clear that there is not linear boundary that can separate the data well. One of the main strengths of SVMs is that it can use as a nonlinear formulation when a kernel is applied to the data. In other words, we can use a kernel function to define similarity between observations in a non-Euclidian space. For our analysis we are using two different type of kernel functions: the Gaussian radial basis, and Laplace kernel functions. We are using these as implement in the `kernlab` package, where the following formulation is used:

This is the formula for the Gaussian kernel:

$$k(x,y) = exp\left(- \frac{||x-y||^2}{2\sigma^2} \right)$$

The complexity of the SVM models is determined by the cost parameter which defines the amount of slackness allowed in the model fitting. For each of the kernels we construct a SVM at different values of the cost parameter. In the plot below you can see the error rates as a function of the cost parameter for SVMs with the Gaussian kernel, on both the training and testing sets.

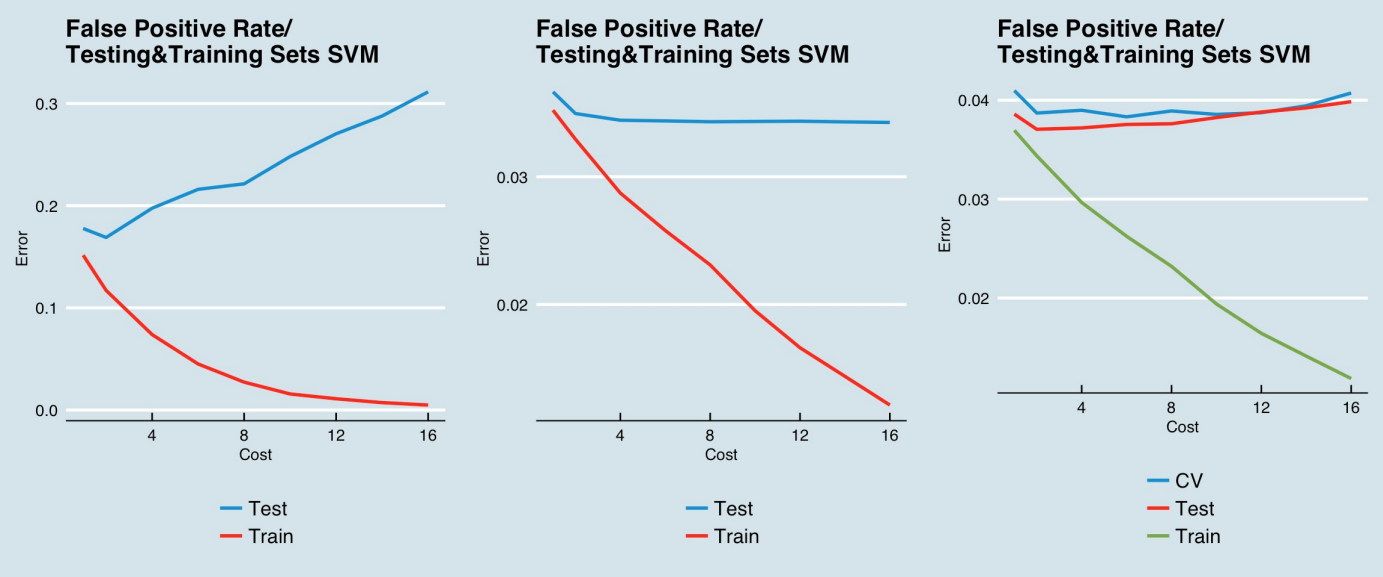


We observe that FPR is in general is much lower than what we saw with the random forest models, however the FNR is somewhat larger. As expected the training error decreases as the model becomes more complex, however we can see that over fitting begins to occur at low model complexity. Specifically, the test FPR begins to increase immediately and the FNR doesn't appear to improve beyond cost of `8` . Based on these results our optimal Gaussian SVM is the one with cost set at `6` , which is also when the CV error stops improving. For this model we have the FNR at about `0.034` and the FPR at about `0.21` , which is a very significant improvement over the `0.70` rate for the random forest. For this model we have `3269` support vector points.

Laplacian kernel:

$$k(x,y) = exp\left(- \frac{||x-y||}{\sigma} \right)$$

Below you can see the same error plots for the SVMs fit with the Laplacian kernel. The over fitting is perhaps even more evident with this kernel specification, and thus our choice for the best Laplacian SVM is the one with the cost parameter set at two `2` . This yields a FPR of about `0.17` , a FNR of `0.035` and a CV error about `0.037` , which is a marginal improvement over the Gaussian SVM with cost set at `6` . For this model we have `7253` support vectors.



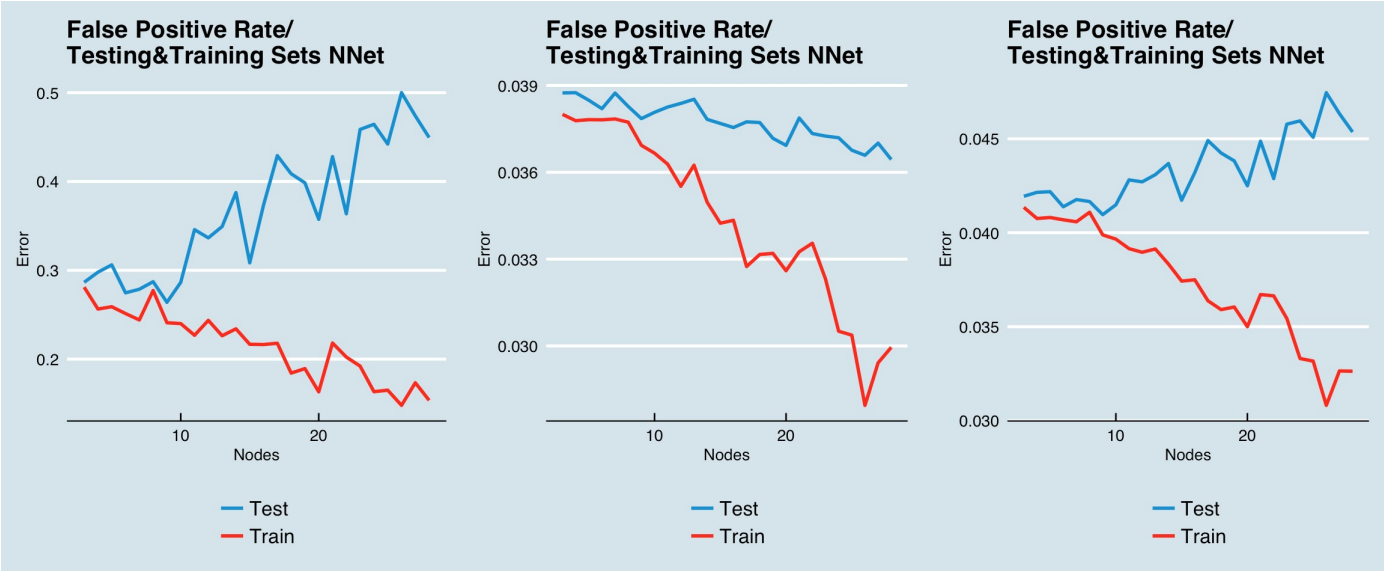
Overall we see that SVM make a significant improvement over the random forests in terms of the FPR, and a small increase in the FNR, which leads us to prefer these models overall. We now move onto Neural Networks and explore how well they do with our data.

Neural Networks

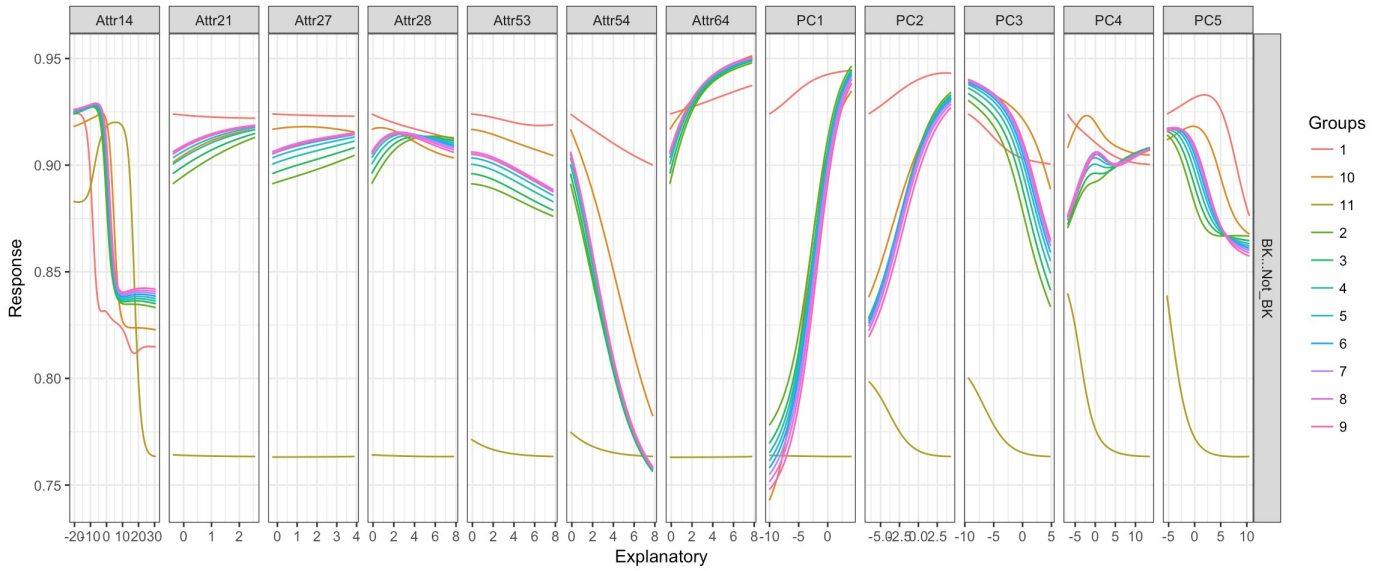
We used Neural Networks (NNets), and focused on nodes arrayed in a single layer. The PCA data we created was used on Neural Networks ranging from three to twenty-eight nodes, in order to test the efficacy of NNet nodes on bankruptcy prediction. Neural Networks used on all 64 variables were not able to converge at acceptable threshold levels.

While using R to create our models, we were forced to increase the threshold of our neural networks from 0.01 to 0.1. Using smaller thresholds would prevent the Neural Networks from converging. We also note that using multiple layers of nodes in our Neural Nets did not produce much difference in error rates, though they have not been included in the study.

As we can see from the plots of the Mean Error Rates, False Negative, and False Positive rates, the best Neural Network we can use is likely the model with 8 nodes, pictured below. More nodes will cause a divergence in the error test rates for the Training and Testing Sets after around 7 nodes in the hidden layer, which suggests over fitting of the training model.



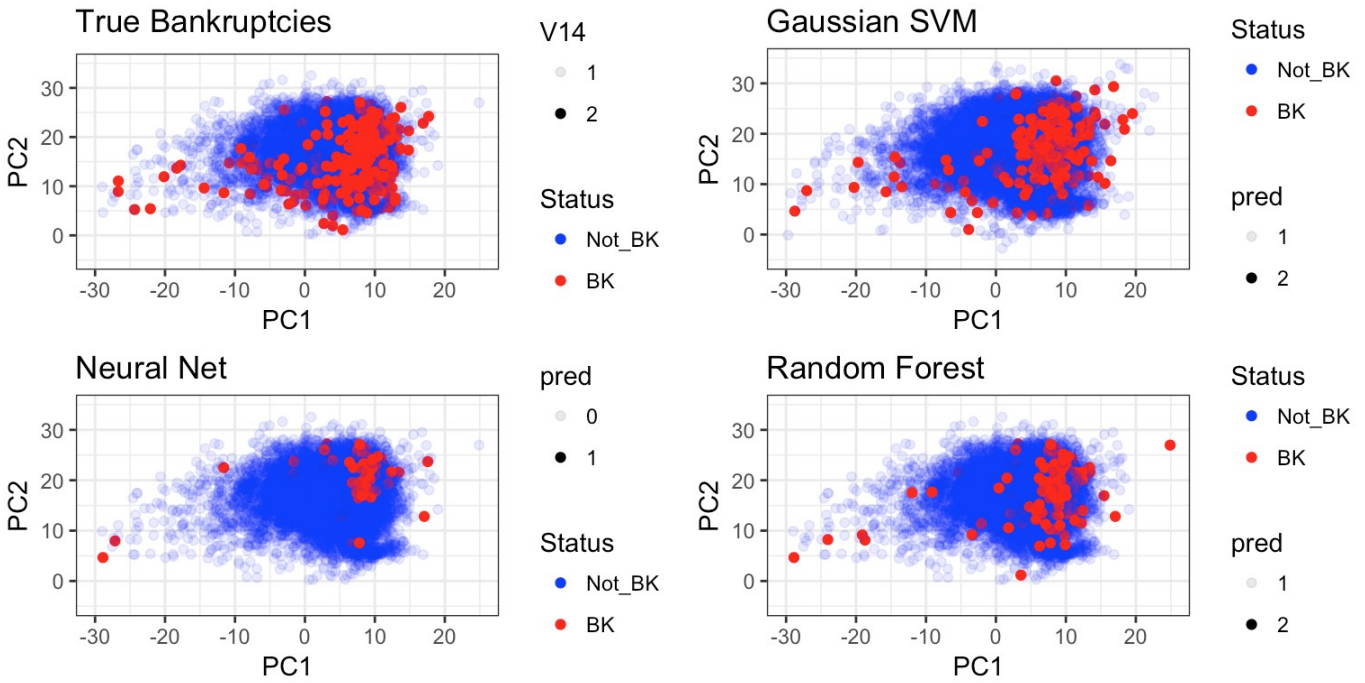
We then use Lek's Profile Method [2] to perform sensitivity analysis on the most effective Neural Network, with ten groups for the quantile values at which to hold other explanatory variables constant. The NeuralNetworkTools and NNet packages in R allow us to use Lek's Profile method quickly.
Lek's Profile Method [5], as explained by Gevrey et. al. has each variable analyzed in turn relative to the other variables set at predetermined values, often the mean, (which we used.). The process is done by creating a matrix covering the range of all input variables. In practice, each variables is split up into a number of equal intervals, (we use 10, in this case,) between the minimum and maximum possible values. All variables except 1 are set initially at their minimum values, and progressively increased by our set quantile values (0.1) as they are graphed for the variable in question's range. We then graph all ten curves created by Lek's Method to determine the contribution to the Neural Net of each variable, and each variable's sensitivity to change.



We see in the above sensitivity analysis that several of the variables cause larger changes to our Bankruptcy prediction, with the Principal Components having the largest. Most of the variables follow some form of distribution. The anomalous sensitivities are groups 1 and 10, which correspond to 0.1 and 1, respectively. We could therefore assume that most of the variables are not sensitive when their respective other variables are grouped around the minimum and maximum of the neural net.
The variables Attribute 14, Attribute 53, PC2, PC3, and PC4 are distinctly 'negative' across all groups, generally following an inverse exponential distribution. Each variable contributes the most at, respectively: Attr14(-20), Attr53(0), PC2(~ -5), PC3(-10), PC4(-7)
Attribute 27, Attribute 64, PC1, and PC5 are distinctly 'positive' across all groups, following approximately a Beta(5,1) distribution. Each variable contributes the most at, respectively: Attr27 (4), Attr64 (8), PC1 (5), PC5 (10).
Despite the variation for each variable, the sensitivities are almost entirely between 0.75 and 0.95, which suggests that the neural network nodes are highly responsive to changes in the values of input nodes, except when the node values are at extremes. Based on the analysis, we would assume that PC1, Att54, PC2, PC3 are the most sensitive variables in our Neural Network, and thus relevant for analysis when introduced to newer data.

Results

Below you can see a visual snpashot of how the select models classify companies as bankrupt, and non-bankrupt. The top-left plot is the plot of true, obeserved bankruptcies from the test set. The other three plots show the fitted/classified observation on the test set. Note how remarkably well the SVM model does in replicating the pattern. We know that SVMs have the tendancy to overfit (aformentioned plots), however the results look better then might be expected on the testing set. The Random Forest pattern looks similar, but we note that it fails to capture many of the bankruptcies to for observation with negative values of PC1. i.e. to the left of `0` on the `x-axis` . The Neural Network pattern looks quite different, and perhaps some non-linear boundary and maybe an elliptical boundary. This noticible difference perhaps explains why the Mean Testing Error for the Neural Network showed the worst results out of the select models.



Conclusion

The table below summarizes the results for some of select models in terms of their error performances. Note that the FPR and the FNR shown here correspond to computations made on the testing set. We can infer that the lowest FPR is achieved by the Laplacian SVM with a value of approximately `0.17` for a cost parameter equal to `2` . This means that this SVM is expected to miss-classify `17` bankrupt firms as non-bankrupt, out of a hundred. On the other hand, the lowest FNR was achieved by the Random Forest at `0.005` , which implies that the out of a thousand firms this model is expected to miss-classify `5` non-bankrupt firms as bankrupt, out of a thousand. Nonetheless, the Random Forest had the highest False Positive Rate (FPR) at `0.6988` , which is a very poor performance compared to the SVM or the Single-Layered Neural Network. For the Single-Layered Neural Network has a relatively average FPR at `0.316` , and FNR at `0.0379` . Thus the Neural Network is a makes more balanced classifications compared to the Random Forest. Note though that the Mean Testing and Mean Training errors for the Neural Network is actually marginally greater than the Random Forest and the SVM. The question is which model should be preferred for classifying or predicting firms as bankrupt or non-bankrupt. We suggest that an individual who is more averse to classifying a bankrupt company as non-bankrupt should prefer to use the SVM models, while an individual who is averse to mistaking a non-bankrupt firm as bankrupt should resolve to the use of the Random Forest. Neural Networks appear to have a balanced performance in terms of the FPR and the FNR, however it doesn't excel in either. It is perhaps of interest to explore how well Multi-Layered Neural Networks would do. In addition, it is of interest to explore how dimension reduction techniques other than PCA would impact the results for the kind of modeling we present in this report.

	SVM with Laplace Kernel (Cost = 2)	SVM with Gaussian Kernel (Cost = 4)	SVM with Gaussian Kernel (Cost = 6)	Random Forest	Neural Network
Mean Training Error	0.0344	0.0359	0.0353	0.0387	0.0411
Mean Testing Error	0.0371	0.0371	0.0374	0.0380	0.0417
4-fold CV Error	0.0387	0.0394	0.0388	NA	NA

	SVM with Laplace Kernel (Cost = 2)	SVM with Gaussian Kernel (Cost = 4)	SVM with Gaussian Kernel (Cost = 6)	Random Forest	Neural Network
False Positive Rate	0.1689	0.1992	0.2143	0.7054	0.2872
False Negative Rate	0.0350	0.0343	0.0342	0.0056	0.0383
nSV	7253.0000	3341.0000	3289.0000	NA	NA
mtry	NA	NA	NA	6.0000	NA
Nnodes	NA	NA	NA	NA	8.0000

Bankruptcy prediction is often used to assess the financial stability of a company, and the probability of its continued existence in future market conditions. A corporate bankruptcy can have far-reaching consequences for economic actors as varied as investors, creditors, suppliers, competitors, government entities, and customers. A bankruptcy can create large economic costs for these actors, and so the topic of corporate bankruptcy has been researched in various forms across disciplines in order to understand the causes and signs of a bankruptcy. Econometric and Financial models often use past data and complex models with years of implied theoretical knowledge and empirical observations pertinent to their fields. This can lead to issues, as a corporate bankruptcy can be caused by a firm's inability to raise capital at a crucial point during an economic or sector-specific recession. There are further difficulties with such a model, because while some companies do declare bankruptcy, others simply persist as distressed companies for the studied period, causing an imbalance in the data. Each year, there are often many more companies that survive than companies that declare bankruptcy. Thus it's likely that a model would incorrectly predict far more companies going bankrupt in the observed period than those that actually do go bankrupt.

References

[1] https://archive.ics.uci.edu/ml/datasets/Polish+companies+bankruptcy+data (https://archive.ics.uci.edu/ml/datasets/Polish+companies+bankruptcy+data)

[2] Lek, S., Delacoste, M., Baran, P., Dimopoulos, I., Lauga, J., Aulagnier, S. 1996. Application of neural networks to modelling nonlinear relationships in Ecology. Ecological Modelling. 90:39-52.

[3]Ensemble Boosted Trees with Synthetic Features Generation in Application to Bankruptcy Prediction Maciej Zięba∗ , Sebastian K. Tomczak1 , Jakub M. Tomczak

[4] http://www.stefvanbuuren.nl/publications/2014%20Semicontinuous%20-%20Stat%20Neerl.pdf

[5] Gevrey M, Dimopoulos I, Lek S. 2003. Review and comparison of methods to study the contribution of variables in artificial neural network models. Ecological Modelling. 160:249-264.