CORPORATE CREDIT RATING USING MACHINE LEARNING TECHNIQUES

A Project Report Submitted

in Partial Fulfilment of the Requirements

for the Degree of

BACHELOR OF TECHNOLOGY

in

Mathematics and Computing

by

Aditya Divekar

(Roll No. 140123003)

Sarthak Agarwal

(Roll No. 140123031)

to the

DEPARTMENT OF MATHEMATICS

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

GUWAHATI - 781039, INDIA

April 2018

CERTIFICATE

This is to certify that the work contained in this project report entitled "**Corporate Credit Rating Prediction Using Machine Learning Techniques**" submitted by **Aditya Divekar** (**Roll No.: 140123003**) and **Sarthak Agarwal** (**Roll No.: 140123031**), to the Department of Mathematics, Indian Institute of Technology Guwahati towards partial requirement of **Bachelor of Technology** in Mathematics and Computing has been carried out by them under my supervision.

It is also certified that, along with literature survey, **empirical analysis has been done** by the student under the project.

Turnitin Similarity: 15%

Guwahati - 781 039
April 2018

(Dr. N. Selvaraju)
Project Supervisor

ABSTRACT

Credit ratings are useful in various ways to retail investors, institutional investors and banks for investment management purposes. They mainly prove useful in financial risk management as it enables the investor to gauge the risk involved in the investment. Credit ratings are available for a variety of settings from corporate ratings to sovereign ratings, and individual consumer ratings. Usually higher ratings correspond to lesser risk, and lower ratings to higher risk, allowing investors to match their investments with their risk preferences. In this paper, we used Machine Learning techniques to predict the corporate credit ratings. We also attempted to identify the important financial factors that influence the credit ratings for the dataset of US companies. Historically, this data has been proprietary and hence is not available freely. The motivation of this paper is to develop efficient credit ratings models which can be used freely by investors and corporations, and eliminate any human biases involved. We obtained prediction accuracies of 90% for Neural Networks using a feed forward network, and accuracies to the upward of 87% for RF and KNC. Also, feature selection techniques were used to rank the variables in order of their contribution. These results were used to interpret the dictating factors for the US market.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 What are Corporate Credit Ratings?

The term Corporate Credit Ratings refers to the ratings that are accorded to companies in the Stock Market. Credit Ratings in itself may refer to Sovereign Credit Ratings - given to countries, Consumer Credit Ratings - given to individuals for their credit worthiness, Bank Credit ratings, and others. In this paper, we attempted to study the Corporate Credit Ratings and their methodology. These are issued by private firms like Moody's Investor Services, Standard & Poor's Financial Services LLC, and Fitch Ratings Inc which together hold more than 95% of this industry share. These ratings reflect the confidence level that a company will be able to fulfill its financial obligations.

Credit ratings broadly divide companies in two categories

(i) Investment bracket (ii) Non-Investment bracket

Exact classifications may vary according to the scale of different issuers, but the above brackets are widely static. These ratings give investors a yardstick measure of the risk involved in investing per company and of how well a company's derivatives and bonds will perform. These are consulted by investors for investment purposes and deciding their Return vs. Risk profile, or by lenders to gauge the credit profile of borrowing entity. Credit ratings are quite significant to a borrower as having lower a rating entails higher interest rates on their derivative instruments to offset the distrust leveled by the rating. The rating not only looks at "probability of default" but also at "loss given default", which leads to further granular ratings between the

above two categories.

1.2 Methodology of Credit Ratings

The rating methodology of the issuers is quite similar on a broad scale, though discrepancies exist whereby companies may be classified differently. There are no exact probabilities of default attached to each rating class; just an ordering of the expected probability levels.

Credit Ratings*

		Moody's	S&P's	Fitch
Investment Grade	Strongest	Aaa	AAA	AAA
		Aa	AA	AA
		A	A	A
		Baa	BBB	BBB
Non Investment Grade		Ba	BB	BB
		B	B	B
		Caa	CCC	CCC
		Ca	CC	CC
		C	C	C
	Weakest	D	D	D

*These credit ratings are reflective of obligations with long-term maturities.

Figure 1.1: S&P Credit Rating Scale

The initial approach to the underlying rating methodology can be broadly explained in two steps.

1. Business Risk : Involves study in respect of the company's sector advantage, performance and standing compared to competitors and other related factors.

2. Financial Risk : Involves an evaluation of the company on its financial numbers/ratios of profitability, liquidity, YoY growth, debt - equity ratios, and other financial forecasts.

The exact methodology is not publicly available due to its proprietary nature.

However, incidents like the 2008 sub-prime mortgage crisis raised questions on the authenticity of these ratings. The awarding of high ratings to the poor performing CDOs and other debt instruments were instrumental in their own way of causing the financial crisis in 2008. As the crisis went down, the issuers downgraded their high investment grade ratings to low non-investment grade ratings in short spans, raising doubt over their credibility. It is hence critical that there exist an unbiased system which can classify and award company ratings backed solely by credible research. The the historical financial performance of the company and the expected performance in the future prove to be the major catalysts in deciding the credit ratings of the companies. We aim to capture these relationships using Machine learning models, and use the recently developed efficient Deep Learning techniques in classifying companies in rating classes. We do a comparative analysis of Deep Learning with other Classification techniques : Support Vector Machines (SVM), KNeighbors and Random Forests.

Chapter 2

Literature Review - Machine Learning Methods

Machine Learning is a field of computer science which gives a computer the ability to think on its own and work without being actually programmed. There are typically two kinds of machine learning tasks -

1. Supervised Learning - Both the input variables and the desired outputs are supplied to the computer. The supervised learning algorithm produces an inferred function after analyzing the training data which is used to map new examples.

2. Unsupervised Learning - In this kind of learning, the input data has no corresponding output variables. The goal is to model the underlying distribution to learn more about the data.

To predict the credit ratings, we use the supervised learning methods as we already have the historical ratings of the companies. We utilize the following machine learning techniques

2.1 Support Vector Machines

Support vector machines are supervised machine learning models which use associated learning algorithms. The SVM training algorithm builds a model on the training set, each belonging to one or other of two categories. It is a non-probabilistic classifier which assigns unseen data points to one of the categories.

A data point is viewed as a p dimensional vector. The goal of the problem is to form a $m-1$ dimensional hyperplane which can separate the data points into two categories depending on their classes. Now, there can exist multiple hyperplanes that perform this task. The hyperplane that maximizes the margin is considered the best hyperplane. If Such a hyperplane exists, it is termed as maximum-margin hyperplane. It often happens that the data is not linearly separable in the m space. This problem is tackled by transforming the data points to higher dimension space, which allows for easier separation. Kernel functions $k(x, y)$ are chosen intelligently to define variables in the m space so that it is easy to compute dot products in terms of variables in original lower dimensional space.

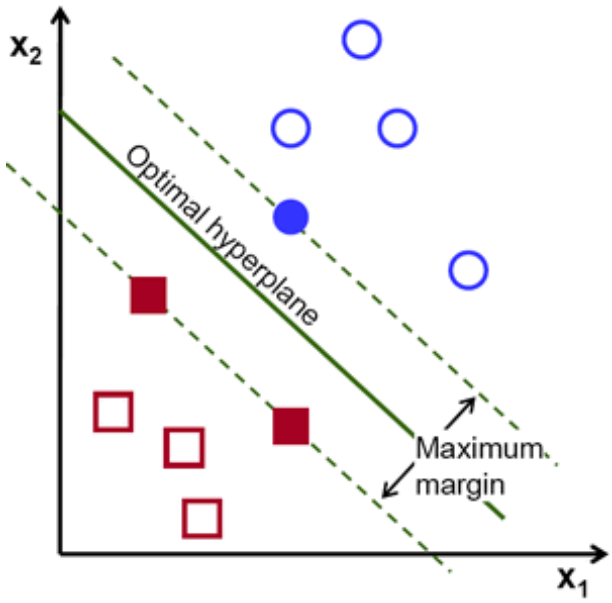


Figure 2.1: Optimal Hyperplane for Support Vector Machine

The following optimization problem is solved in support vector technique:

$$\begin{aligned} \underset{w\in H, b\in R, \eth_i \in R}{\text{minimize}} \quad & \frac{1}{2} w^T w + C \sum_{i=1}^{l} \eth_i \\ \text{subject to} \quad & y_i(w^T \varphi(x_i) + b) \geq 1 - \eth_i, \eth_i \geq 0, i = 1, ..., l, \end{aligned}$$

where $w \in R_d$ is the weight vector, $C \in R_+$ is the regularization constant, and the mapping function φ projects the training data into a suitable feature space H. The credit rating prediction problem is a multi class classification problem as the ratings provided by the institutes are divided in many classes. So we use a multi class SVM rather than a binary SVM.

The above discussed idea works only if the classifier tries to classify data points in just two classes. For a multiclass problem, we use a one vs rest classifier where each class is selected once and the rest others are combined to form a single class. The problem now is reduced to a binary classification problem where the earlier logic works easily.

2.1.1 Hyper-parameters Used

The following hyper-parameters are tuned to optimize the performance of the classifier:

1. **C** - It is the penalty parameter of the error term with a default value of 1. It specifies how much we want to avoid misclassifying. The larger the value of C, the smaller is the margin of the hyperplane.

2. **Gamma** - It is a coefficient for the kernel used in the support vector classifier. A large value of gamma implies that the model has high bis and low variance and vice versa.

3. **Kernel** - It specifies the type of kernel used in the model. Different types of kernels are linear, polynomial, sigmoid, rbf etc.

2.2 Neural Networks

An Artificial Neural Network (ANN) consists of multiple interconnected perceptrons (or neurons) which work together towards solving problems. The learning in ANNs is similar to the natural human learning process in the sense that adjustments are made between the synaptic connections depending on the outcome (or the reward). A Neural Network (NN) consists of stacks of layers containing above mentioned neurons which adapt the coefficients of their activation function using gradient descent on a specified loss function. They receive input, process using activation function, and then propagate the output to the next layer, or as a final result depending on its layer position in the network. The structure of a NN involves an input layer, some

hidden layers, and an output layer stacked together. The following figure shows an example network.

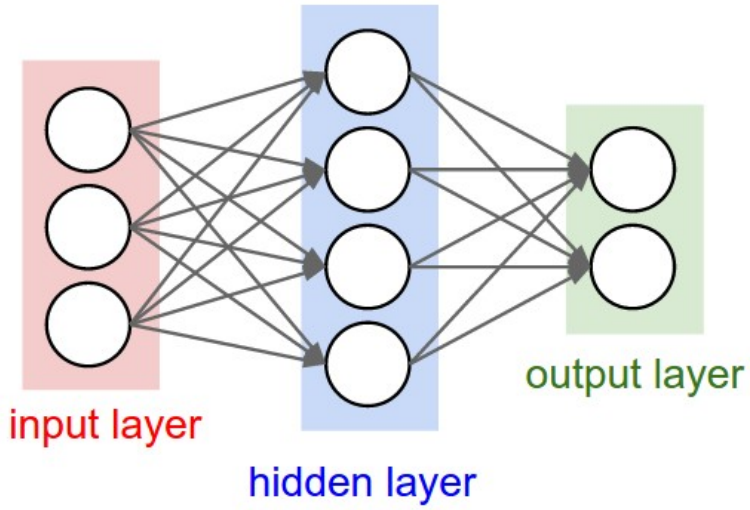


Figure 2.2: Structure of a Neural Network Model

More generally, neural networks can be thought of as a chain of functions that takes some input and calculates some desired output. The power of neural networks lies in the fact that chaining together a variety of simpler functions makes it possible to represent more complicated functions in an easily trainable, parameter-efficient way.

The learning process of neural networks involves iterating through the training data multiple times over epochs. Each iteration involves seeing all the training examples, usually in the form of batches, and each batch leads to an update of the activation function weights using gradient descent. During this training phase, the neurons adjust weights so as to perform optimally on out-of-sample data; mimicking the human brain. This type of learning is also called "connectionist learning," due to the chained interdependence of the neurons. Neural networks are quite robust as they can effectively filter noise from true data, and have the ability to learn complex unseen patterns.

Neural networks with multiple hidden layers are generally termed as Deep Neural Networks. An example would be

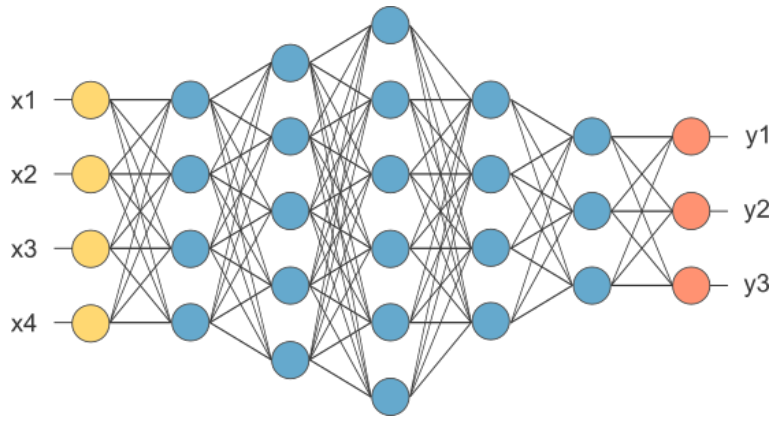


Figure 2.3: Structure of a Deep Neural Network Model

2.2.1 Hyper-parameters Used

There are two types of hyper-parameters in neural networks.

1. **Model hyper-parameters**
2. **Training hyper-parameters**

The hyper-parameters for the **Model** are tuned to create an optimal neural network. They are:

1. **hidden units** - The number of hidden layers in the network, and the number of nodes per layer.
2. **weight decay** - Reduces the over-fitting in the network, by penalizing high values of network weights.
3. **weight initialization** - The initialization method to use for the network weights. This affects the local minimum of the loss found by the training algorithm. Commonly used choices are initialization with 0, and with random numbers with Gaussian distribution.

The hyper-parameters for the **Training** the model are tuned to train the neural network optimally. They are:

1. **learning rate** - Determines how large the update to the gradient should be towards the gradient descent. General purpose values lie in the range 1 to 10^{-6}.

2. **loss function** - Measures the error in the network's output for a training example compared to the true output. The Euclidean norm is a commonly used for this purpose.

3. **batch size** - Number of training examples taken together in a batch. Smaller batches speed up gradient computation, but the gradient value update for largest batches is more efficient as it is closer to the true gradient.

4. **epochs** - Number of gradient descent iterations over the entire training data.

2.3 KNeighbors Classifier

KNeighbors Classifier is a simple machine learning algorithm where all the computations are performed only during the classification phase. Thus it is an instance-based learning algorithm.

In this scheme, data points in the closer neighborhood of the point in reference should contribute in comparison to the points which are far. One way to do this is to assign weights to the points such that the closer points have more weight. This can be achieved by using a inverse distance weight scheme.

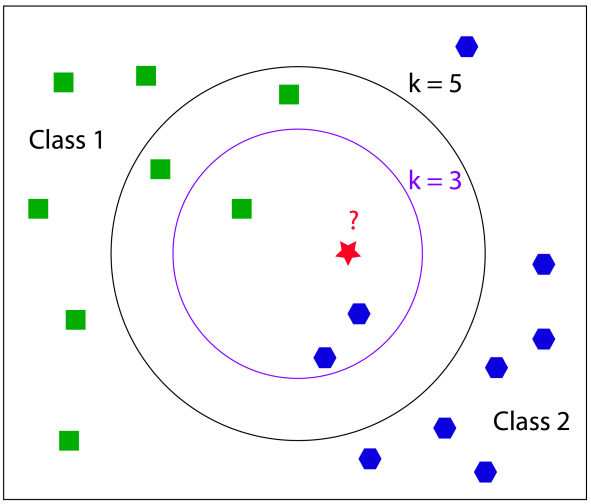


Figure 2.4: KNeighbors Classification

Algorithm

The training examples are pairs of d-dimensional feature vectors along with their class labels. In the training phase, the input vector is stored with their class labels.

In the classification phase, hyper-parameter k is chosen by the user. When an unlabeled vector is given as an input, it is assigned a label which is most frequent amongst it's k closest neighbors. Euclidean distance is generally used as a the distance metric. Sometimes, Hamming distance is also used.

Parameter Selection

The choice of K is depended on the type of the data used. Choosing a larger value of k reduces the noise effect on classification but on the same time, makes the boundaries between the class less distinguishable. A good value of k can be selected by using convenetional techniques such as hyper-parameter optimization.

2.3.1 Hyper-parameters Used

The following hyper-parameters are tuned to optimize the performance of the classifier:

1. **neighbors** - It refers to the number of neighbors to be checked during the model execution.

2. **Algorithm** - It refers to the algorithm used to compute the nearest neighbors. BallTree, KDTree and Brute are the different algorithms that can be used.

3. **Metric** - It refers to the distance metric to be used to calculate distances.

2.4 Random Forest

Random Forest is an ensemble learning algorithm that uses multiple decision trees at training time. Then for prediction, it outputs the mode class for classification and mean value for regression, derived from the predictions of the different random forests. So to understand how random forests work, we need to first explore the working of decision trees.

Decision Tree Learning

The goal of the Decision Tree algorithm is to predict a target value based of some input variables. A decision tree is drawn upside down with it's root at the top. Every interior node corresponds to some specific input feature. Edges of the tree correspond to all the possible values a variable can take. The leaves denote all possible values

of the target variable when given values of the input variables denoted by the path from the root up to the leaf.

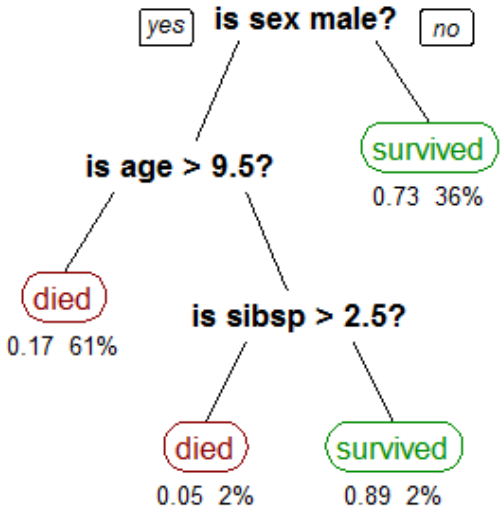


Figure 2.5: Decision Tree

The learning takes place using specific algorithms such as ID3, C4.5 , CART etc. All these algorithms follow a top down approach for constructing the decision tree; choosing the variable that gives the *best* split of the item set. The definition of the *best* split depends on the metric used. The most common metrics used are as follows:

1. Gini Impurity - Gives a measure of how often a random chosen element from the item set will get an incorrect label if the labeling was selected randomly, and the distribution of labels mimics that in the subset.

2. Information Gain - Based on the entropy concept. It is defined as the difference between entropy of the parent and the weighted sum of the entropy of the children.

The problem with decision trees is that they tend to over fit the training set *i.e.*. tend to develop low bias and very high variance. Random forest average the outputs of multiple decision trees, trained on different parts of the same training data to reduce the variance. This introduces some bias in the model but gives a great boost to the performance of the model.

2.4.1 Hyper-parameters Used

The following hyper-parameters are tuned to optimize the performance of the decision tree:

1. **n_estimators** - The number of trees in the forest.
2. **criterion** - The criteria for measuring the quality of the data split. The options available are Gini impurity and Entropy for the measuring of information gain.
3. **max_features** - Specifies the number of features that should be looked at to decide the best split. More hyper-parameters are available, but they are not relevant in our case.

Chapter 3

Literature Review - Credit Ratings Prediction

As seen before, credit rating prediction is used for predicting the rating of a company given information mainly regarding its financial strength and many other factors such as the state of the market, the country and its competition. The process involves assigning a grade $\omega \in \Omega$. The rating scale may vary, for example the scale for S&P is $AAA, AA, A, BBB, BB, B, CCC, CC, C, D$, a total of 10 values[5].

Many different methods involving Machine Learning, AI and Statistical techniques have been used in prior studies for finding models for credit ratings prediction. These studies also varied greatly in the features used, the time period used, the set of rating classes Ω.

The earliest studies involved using traditional statistical methods. These involve Linear Regression (LR), Ordered Logistic Regression (OLR), Multiple Discriminant Analysis (MDA) and Ordered Prohibit Model. Studies [7] have shown that the OLR and OPM perform better on average, partly in fact as they take the class ordering in consideration [8] [9]. In the light of recent advancements, AI techniques have also been employed in learning and then predicting credit ratings using models [6]. These allow the learning of much more complex relationships, and learn from data themselves. Machine Learning has also been used extensively in credit rating predictions. First the model is trained on data for which the results are known, *i.e.* where the ratings of companies are known, and then the trained model is used for rating companies in the future time periods. Previous studies have shown that Neural Networks preform significantly better than statistical methods [2]. Hajek [6] com-

pared the subclasses of Neural Networks and found that Probabilistic NNs and RBF (Radial Basis Function) NNs performed better than Multi Layer Perceptrons. Also, Support Vector Machines have proven to give good results as they allow specifying different kernel functions [6]. Huang et al. and Kyoung-jae Kim [10] independently studied the adoption of Multi Class SVMs for building prediction models of credit rating.

Feature Selection is also a topic of great interest in this area. Determining the most important features to be used in the model and filtering out the other features for reducing noise in the model, and increasing accuracy. Also, feature selection provides an important insight into the behavior of markets. ASSD showed that total assets and total debt were more important in US model implying that the US bond raters give more weight-age to size of the company, whereas the importance of operating profit margins in Taiwan model show that the Taiwan raters focus more on the profitability of the company.

Chapter 4

Methodology

4.1 Data Processing

4.1.1 Data sources

The data used for the study can be divided into two parts. The Historical Credit Ratings which were obtained from the Thomson Reuters Eikon Database. And the Historical Fundamental Data was obtained from the website `simfin.com`. All the learning models were built using the Python Scikit-learn [12] library, and the Keras [3] library was used for building the Neural Network models.

4.1.2 Data description

The dataset consists of 1430 ratings of 400 companies considered over a period of 6 years from 2010 to 2016. Due to unavailability of complete fundamental data, the dataset was trimmed to contain 1222 rows. The data for each company consists of quarterly values of 49 financial quantities over the specified period from 2010 to 2016.

4.1.3 Training dataset

The features used in the data are listed below. A total of 98 features are used, where data from 49 financial quantities for the rating year and the year previous to it are used for each training example.

Some of these features are:

Table 4.1: List of features

Total Equity_curr	Current Liabilities_prev
Total Equity_prev	Preferred Equity_curr
Retained Earnings_curr	Minorities_curr
Equity Before Minorities_curr	Long Term Debt_curr
Equity Before Minorities_prev	Intangible Assets_curr
Retained Earnings_prev	Cash and Cash Equivalents_prev
Dividends_curr	Accounts Payable_curr
Debt to Assets Ratio_prev	Net Change in PP&E & Intangibles_prev
Debt to Assets Ratio_curr	Operating Margin_prev
Total Assets_curr	Return on Assets_prev

The **_curr** suffix denotes the data for the same year, whereas **_prev** denotes data from the previous year.

4.2 Cross Validation

Cross Validation is used in each of the methods for optimizing the training of the model. These optimized models are constructed for KNeighbors Classifier, SVM, Neural Network and Random Forest which are then used for testing on the out-of-sample date for accuracy. The technique of grid search is used for searching the best parameters for the models. Also, the generated results are reported after 7-fold cross validation for higher consistency.

4.3 Classification of Data

The Credit ratings are categorized into 5 coarser credit categories ('AA', 'A', 'BBB', 'BB', 'C'). The composition of these classes is shown in the following table. It can be observed that, roughly, a bell curve distribution exists which is indicative of the true distribution of these ratings.

Table 4.2: Ratings Classification

Ratings	Number of Companies	%
AA	27	2.2
A	163	13.3
BBB	548	44.84
BB	246	20.13
C	238	19.47

Chapter 5

Study

5.1 Performance Measurement Metrics

Different metrics are used in evaluating the performance of the models. The metrics are as follows:

1. **Precision Score** - It measures how precise are the predictions made by the model. It is the ability of the model to not label a negative sample as positive. Mathematically it is defined as:

$$\frac{tp}{tp + fp}$$

   Here tp denotes true positives and fp denotes false positives.

2. **Recall Score** - The recall is intuitively the capacity of the classifier to discover all the positive examples. Numerically it is characterized as:

$$\frac{tp}{tp + fn}$$

3. **F1 Score** - It is the weighted average of the precision and recall scores. It has a range of $0 - 1$ with 1 as the best F1 score and 0 the worst. Mathematically it is defines as:

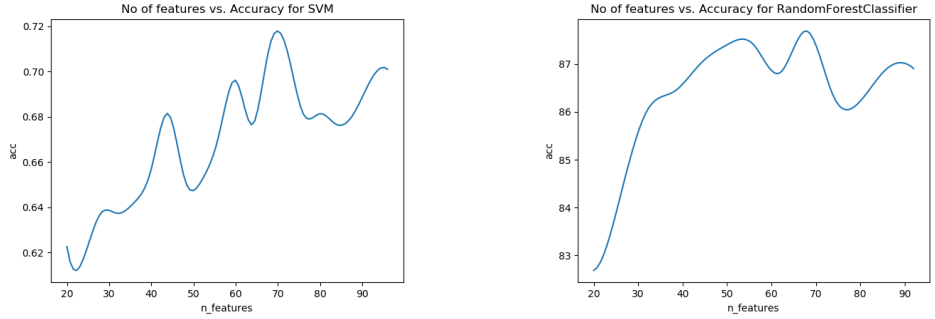
$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

5.2 Feature Selection

Univariate feature selection is performed based on univariate statistical tests to select the best features amongst all. Features are selected according to their percentile of the highest score.

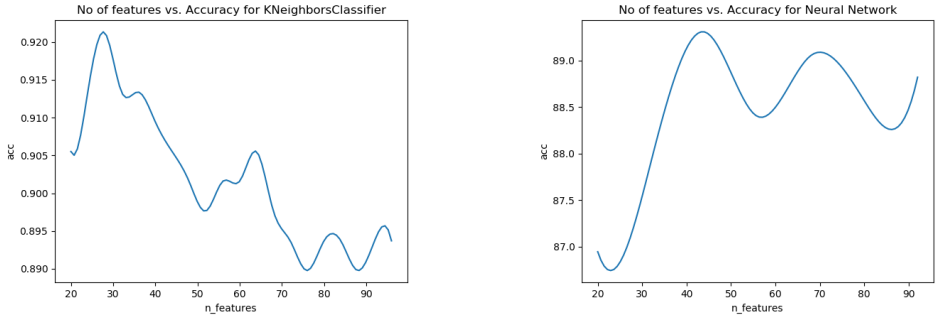
5.2.1 Dimensional analysis

Model accuracy was checked for different dimensions of the training set for four different models. The output is shown below:



(a) SVM (b) Random Forest Classifier

Figure 5.1: Feature selection for SVM and RFC



(a) KNeighbours Classifier (b) Neural Network

Figure 5.2: Feature selection for KNC and NN

5.2.2 Top Features Selected

The following table shows the top 30 features sorted according to their importance score

Table 5.1: Feature selection scores using ANOVA f-value

Features	**Percentile**	**Features**	**Percentile**
Total Equity_curr	1.00000	Total Liabilities_curr	0.57928
Total Equity_prev	0.98814	Total Liabilities_prev	0.56341
Retained Earnings_curr	0.96691	Cash Operating Activities_prev	0.55853
Equity Before Minorities_curr	0.96629	Net Profit_curr	0.51935
Equity Before Minorities_prev	0.94975	Receivables_curr	0.51696
Retained Earnings_prev	0.88775	Cash Operating Activities_curr	0.49606
Dividends_curr	0.87365	Receivables_prev	0.47601
Debt to Assets Ratio_prev	0.85161	Current Assets_curr	0.47530
Debt to Assets Ratio_curr	0.82343	Treasury Stock_curr	0.44122
Total Assets_curr	0.76314	EBITDA_prev	0.42770
Dividends_prev	0.75279	EBITDA_curr	0.42623
Total Assets_prev	0.74847	Treasury Stock_prev	0.42199
EBIT_prev	0.65097	Current Assets_prev	0.41889
EBIT_curr	0.61276	Short term debt_curr	0.39697
Net Profit_prev	0.59885	Short term debt_prev	0.39188

5.3 Principle Component Analysis

Principle Component Analysis (PCA) is a statistical method used to convert a set of partially correlated variables to a set of linearly uncorrelated variables using orthogonal transformations. The variables outputted are in a sorted in such a way that the first component has the highest variance and followed by other components in decreasing order. PCA is used for dimension reduction of the data which originally contained 98 features.

The following graph shows the amount of variance captured by different number of features for the data:

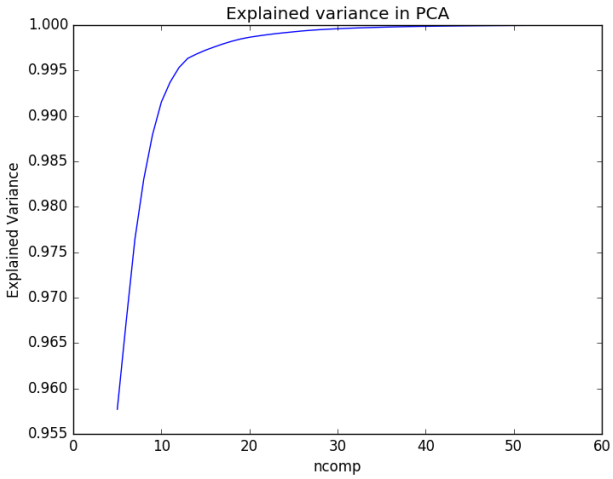
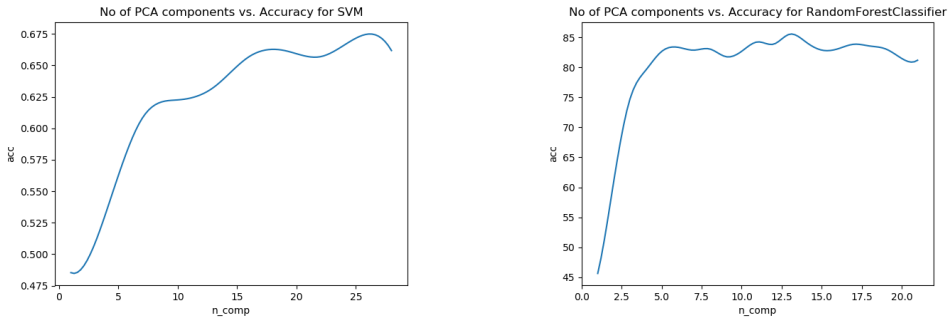


Figure 5.3: Explained Variance by PCA components

5.3.1 Dimensional analysis

Model accuracies were checked for different number of PCA components for the four different learning algorithms. The output is shown below:

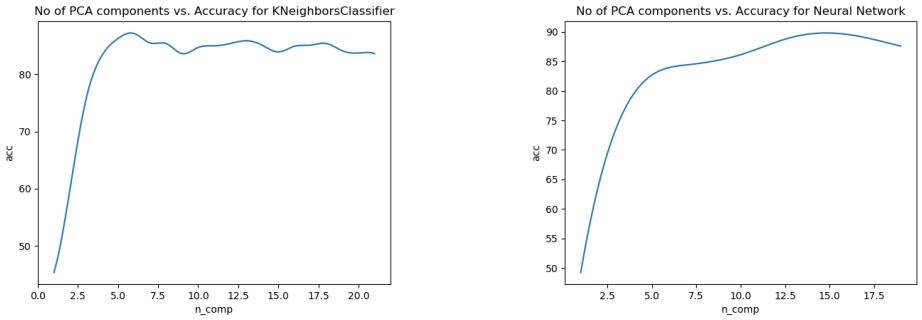


(a) SVM (b) Random Forest Classifier

Figure 5.4: Using PCA with SVM and RFC

5.3.2 Usage in final results

No significant improvement in accuracies was observed using PCA components. This may be attributed to the fact that the intra-class variance is high, given the large differences in magnitude even for same class examples, compared to the inter-class variances and hence PCA retains the intra-class variance. This might lead to a class of the critical inter-class variance which is needed for separation of the classes.



(a) KNeighbours Classifier (b) Neural Network

Figure 5.5: Using PCA with KNC and NN

Hence, the original features were retained for all the subsequent learning algorithms.

5.4 Results

We used different machine learning models to predict the credit ratings on the test dataset. The following model accuracies were obtained:

Table 5.2: Model accuracies

	NN	**SVC**	**RF**	**KNC**
Accuracy	0.8979	0.70098	0.8750	0.9213

The Classification confusion matrices for different models are given below.

5.4.1 Support Vector Machines

Table 5.3: Classification confusion matrix SVM

Actual	**Predicted**				
	0	**1**	**2**	**3**	**4**
0	5	0	0	0	0
1	2	11	6	1	2
2	0	6	83	8	2
3	0	3	17	20	5
4	1	0	3	5	24
	Precision	**Recall**	**F1-Score**	**Support**	
0	0.62	1.00	0.77	5	
1	0.55	0.50	0.52	22	
2	0.76	0.84	0.80	99	
3	0.59	0.44	0.51	45	
4	0.73	0.73	0.73	3	
avg/total	0.69	0.70	0.69	204	

5.4.2 Neural Network

Table 5.4: Classification confusion matrix NN

Actual	Predicted				
	0	**1**	**2**	**3**	**4**
0	4	0	0	0	0
1	1	21	1	0	0
2	0	2	75	1	0
3	0	0	0	32	3
4	1	1	0	1	31

	Precision	**Recall**	**F1-Score**	**Support**
0	1.00	1.00	1.00	4
1	0.55	0.50	0.52	22
2	0.95	0.91	0.93	23
3	0.91	0.86	0.88	35
4	0.94	0.94	0.94	34
avg/total	0.94	0.94	0.94	174

5.4.3 KNeighbors Classifier

Table 5.5: Classification confusion matrix KNN

Actual	Predicted				
	0	**1**	**2**	**3**	**4**
0	5	0	0	0	0
1	0	18	4	0	0
2	0	1	96	1	0
3	0	0	2	34	9
4	0	1	0	4	28

	Precision	**Recall**	**F1-Score**	**Support**
0	1.00	1.00	1.00	5
1	0.55	0.50	0.52	22
2	0.95	0.91	0.93	23
3	0.91	0.86	0.88	35
4	0.94	0.94	0.94	34
avg/total	0.94	0.94	0.94	174

5.4.4 Random Forest

Table 5.6: Classification confusion matrix RF

Actual	Predicted				
	0	**1**	**2**	**3**	**4**
0	3	0	0	0	0
1	0	11	4	0	0
2	0	70	0	0	0
3	0	0	3	24	4
4	0	0	0	4	19

	Precision	**Recall**	**F1-Score**	**Support**
0	1.0	1.00	1.0	3
1	1.00	0.73	0.85	15
2	0.91	1.00	0.95	70
3	0.86	0.77	0.81	31
4	0.83	0.83	0.83	23
avg/total	0.90	0.89	0.89	142

5.5 Comparative Analysis

The following table shows how the accuracies obtained in this study compare with the existing literature.

Table 5.7: Literature review

Study	Method	Research Findings
[1]	OMSVM	Proposed new multi-class classifier OMSVM which is an extension of the SVM binary classifier. Accuracy = 67.98% Classes = 5
[4]	SVM-Fuzzy Clustering	Combined support vector classifier with fuzzy clustering. Accuracy = 73.70% Classes = 4
[8]	SVM / BPN	Introduced support vector machines first time to this problem, and used BPN as benchmark. Accuracy=80% Classes=5
[1]	OMANN	Proposed new multi-class classifier OMANN which is an extension of the ANN binary classifier. Accuracy = 68.34% Classes = 5
[11]	SVM / FFNN	Empirical study of the performance of SVM and FFNN on Korean bond rating data. Accuracy SVM = 67.22% Accuracy FFNN = 59.9 Classes = 5
[13]	RF	Applied RF for variable feature selection and forecasting. Accuracy = 95.5% Classes = 4

Table 5.8: Summary of obtained accuracies

	NN	**SVC**	**RF**	**KNC**
Accuracy	0.8979	0.70098	0.8750	0.9213

The results obtained in this study clearly demonstrate high accuracy compared to the existing methods in the literature for the credit rating prediction problem. In particular, the results of Neural Network and KNeighbors Classifier are quite impressive and provoke further thought towards application.

Chapter 6

Conclusion and Future Direction

The feature selection was performed using ANOVA f-value based scoring technique and then sorting the features according to the score. The top 12 features after performing feature selection are tabulated below.

Table 6.1: Top 12 Features

Features	**Type**
Total Equity	Size
Retained Earnings	-
Equity Before Minorities	-
Dividends	Profitability
Debt to Assets Ratio	Leverage
Total Assets	Size
EBIT	Profitability
Net Profit	Profitability

The dataset used is for United States incorporated companies and their credit ratings. The Total Equity and Total Assets variables contribute strong scores among the features. Also, the variables Dividends, EBIT and Net Profit make significant contributions as revealed above. Hence, it can be observed that major emphasis is given to the size of a company in determining the outlook for its credit rating, and profitability also follows suit.

This project applied the techniques of Neural Networks, SVM, Random Forest and KNeighbors Classifier to the problem of corporate credit rating prediction, and it is seen that these methods are indeed quite promising for application in this financial domain. Obtained results show that Neural Networks and KNeighbors Classifier consistently perform well. The performance of KNC is surprising given it

has not been applied before in literature. Further study can be conducted to see how the algorithms scale with more data.

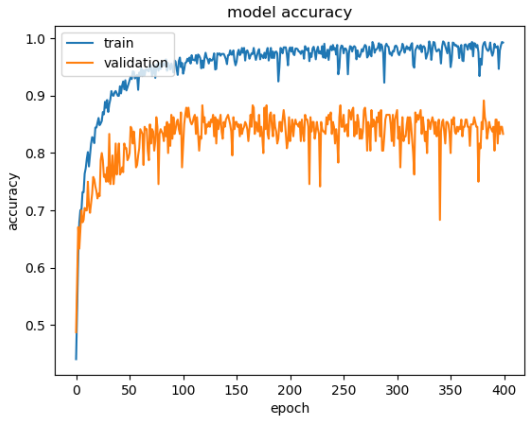


Figure 6.1: Epochs vs Training/Validation Accuracy for NN

As seen in the above figure the Neural Network starts over-fitting after a few epochs, which shows that more training date is required for better training. Also, the excellent performance of the instance-based learning model KNeighbors Classifier in this problem is an interesting topic to dive into, given the novelty of this approach. Since this is not a learning model, the results of this model should be treated differently than the other models.

Bibliography

[1] H Ahn and K.-J Kim. Corporate credit rating using multiclass classification models with order information. *International Journal of Economics and Management Engineering*, 5(12):95–100, 12 2011.

[2] D. Brennan and A. Brabazon. Corporate bond rating using neural networks. *Proceedings of the International Conference on Artificial Intelligence, IC-AI'04*, 1:161–167, 2004.

[3] François Chollet et al. Keras (open source neural network library written in python). 2015.

[4] Xuesong Guo, Zhengwei Zhu, and Jia Shi. A corporate credit rating model using support vector domain combined with fuzzy clustering algorithm. *Mathematical Problems in Engineering*, 2012:1–2, 07 2012.

[5] Petr Hajek and Krzysztof Michalak. Feature selection in corporate credit rating prediction. *Knowledge-Based Systems*, 51(Supplement C):72 – 84, 2013.

[6] Zan Huang, Hsinchun Chen, Chia-Jung Hsu, Wun-Hwa Chen, and Soushan Wu. Credit rating analysis with support vector machines and neural networks: a market comparative study. *Decision Support Systems*, 37(4):543 – 558, 2004.

[7] Ruey-Ching Hwang. Forecasting credit ratings with the varying-coefficient model. *Quantitative Finance*, 13(12):1947–1965, 2013.

[8] Ruey-Ching Hwang, Huimin Chung, and C.K. Chu. Predicting issuer credit ratings using a semiparametric method. *Journal of Empirical Finance*, 17(1):120 – 137, 2010.

[9] RueyChing Hwang, K. F. Cheng, and ChengFew Lee. On multipleclass prediction of issuer credit ratings. *Applied Stochastic Models in Business and Industry*, 25(5):535–550, 09 2008.

[10] Kyoung jae Kim and Hyunchul Ahn. A corporate credit rating model using multi-class support vector machines with an ordinal pairwise partitioning approach. *Computer & Operations Research*, 39(8):1800 – 1811, 2012.

[11] Young-Chan Lee. Application of support vector machines to corporate credit rating prediction. *Expert Systems with Applications*, 33:67–74, 07 2007.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in python (open source machine learning library written in python). *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[13] Hsu-Che Wu and Yu-Ting Wu. Evaluating credit rating prediction by using the kmv model and random forest. *Kybernetes*, 45(10):1637–1651, 2016.