ECE 681 Pattern Classification and Recognition
Bankruptcy Prediction Project
Course Project Final Report
Professor Stacy Tantum, Ph.D.
Zihan Liu (zl188)

I. Problem Statement

Understanding the financial state of a company creates trust about the reliability of a company and supports many investments and customer decisions [1]. Companies rely on such information to manage risks and take precautions within the company [1]. Banks and other institutions assess such information to decide the credibility of a firm. The financial conditions of firms and corporates not only concern business partners and financial institutions, but also impact the society and country's economy as a whole.

The business failure and bankruptcy can be devastating to the society and impose significant loss. In addition to direct financial loss on its investors and shareholders, many indirect costs are also associated with corporate bankruptcy. Human and labor loss is also one of them [7]. The consequences of bankruptcy on workers include pension losses, psychological and social costs [7]. The reduce in wages of workers can lead to personal bankruptcy, that increases the country's unemployment rate and reduce the nation's purchasing power [7]. Therefore, it is not only at the business partners and investor's interests, but also at both employee's and policy maker's interests to be able to evaluate the financial health of businesses. The ability to forecast bankruptcy is vital to help minimize the social and economic loss associated with bankruptcies.

1.1 Previous Work

Extensive research into bankruptcy prediction can be traced back to 1932 [3]. Early bankruptcy prediction models are mostly statistical based [3]. The most common statistical tool is the multiple discriminant analysis which was first used by Altman in 1968 known as Zeta Model [3]. The Zeta model returns a single number, the z-score to present the likelihood of a company going bankrupt in the next two years [4]. With the advancement of computer technique, modern bankruptcy prediction mostly uses computer-based techniques that allows the model to be more robust. Artificial Neural Network is one of them. A comparison was made in 1992 by Tam and Kiang and they reported that Neural Network applies to bankruptcy prediction as it outperforms other tested techniques [10]. Later, Support Vector Machine [8] was also applied to the subject [8]. Van Gestel (2003) implemented the Least Square Support Vector Machine and later Shin et al (2005) tested another Support Vector Machine against Neural Networks [10]. The SVM classifier was acknowledged as a more simple and accurate technique than Neural Networks in bankruptcy prediction [10]. The last decades, most research was focused on improving and comparing existing models.

II. Data Overview

Due to the popularity of the bankruptcy prediction, this area was heavily researched over the years. However, the research result can be affected by different econometric features studied and sample distribution. In this project, I wanted to analyze the bankruptcy data of polish companies acquired from University of California Irvine Machine Learning Repository to predict bankruptcy [9]. I wanted to focus on analyzing the strengths and drawbacks of different feature subsets and different classifiers on predicting bankruptcy and determining how different models are suited under different circumstances. This project can provide valuable insights to companies, employees, financial institutions and policy makers to determine the financial conditions of a company and its long-term sustainability in the market.

2.1 Data Description

The data was collected from Emerging Markets Information Service (EMIS), which is a database containing information on emerging markets around the world [9]. The dataset contains statistics from both bankrupt and still operating companies [9]. The bankrupt companies were analyzed from 2000 to 2012, while the still operating companies' information were collected in the period of 2007 to 2013 [9]. This dataset is very apt for our research purpose since it contains 64 important financial and economic attributes that can be used to determine the financial health of the company. There are adequate number of features to select from and analyze. The dataset contains five different cases summarized in five consecutive years. The first-year data contains financial rate from the first year of the forecasting period and corresponding class label that indicates bankruptcy status after 5 years [9]. The second-year data contains financial rate from the second year of the forecasting period and corresponding class label that indicates bankruptcy status after 4 years, respectively [9]. In this project, the first-year data was primary used for exploration and analysis.

2.2 Data Exploration

To understand the dataset, the number of data points and data types were determined and were summarized in the Table 1 below. The proportion of data in each class was visualized below in Figure 1.

Data	Total Points	Non-Bankrupt Instances	Bankrupt Instances	Data Type
1st Year	7027	6756	271	Object
2nd Year	20173	9773	400	Object
3rd Year	10503	10008	495	Object
4th Year	9792	9227	515	Object
5th Year	5910	5500	410	Object

Table 1 Dataset Summary from year 1 to year 5

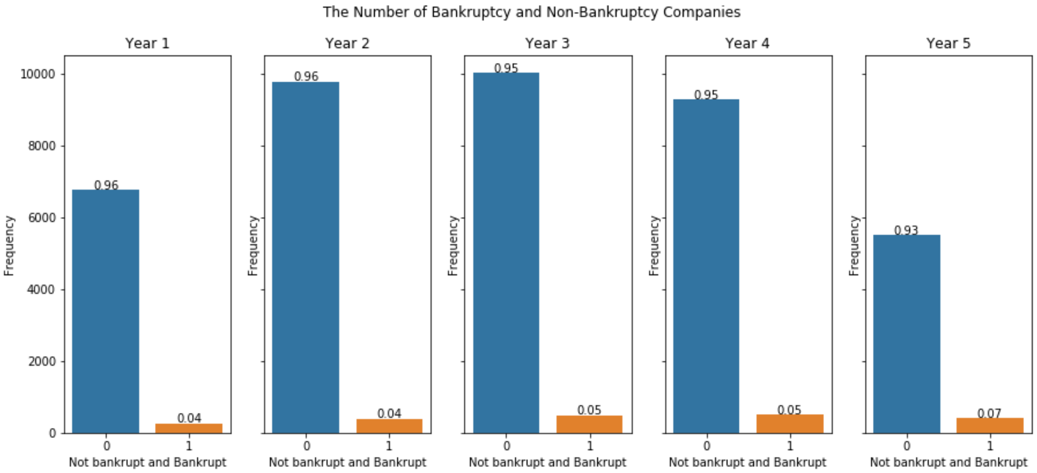

Figure 1 The proportion of bankrupt and non-bankrupt companies from year 1 to year 5

The number non-bankrupt companies weight significantly more than bankrupt companies. The dataset has imbalanced data distribution. To further understand the dataset quality, the heatmap of missing values in each year data was generated. The heatmap of missing values in year 1 data is shown below in Figure 2.

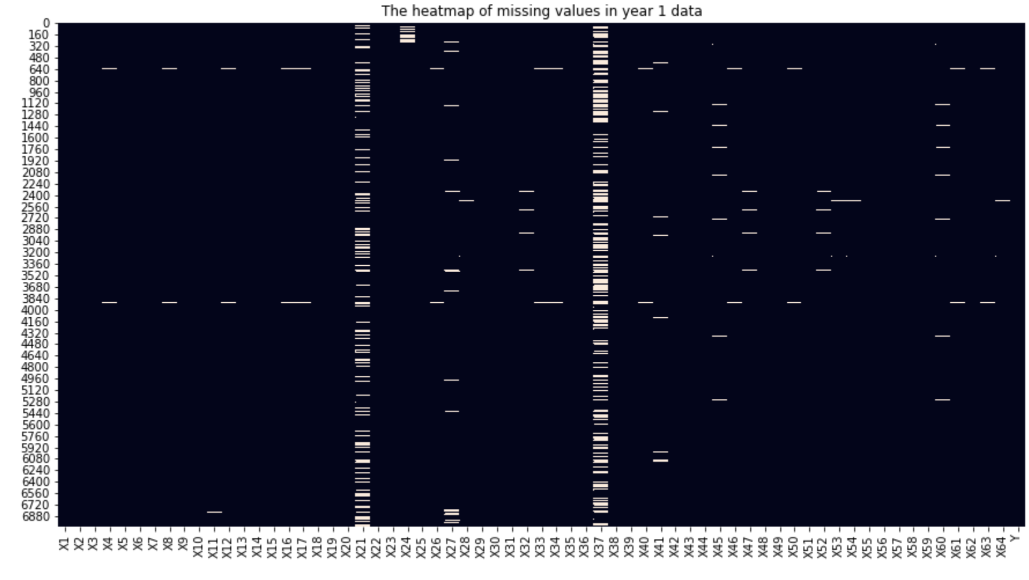

Figure 2 The heatmap of missing data in year 1

The occurrence of missing value is common in this data set and may have a significant impact on the results to be obtained. Specifically, we observe that X21(sales(n)/sales(n-1)) and X37 (current assets - inventories) / long-term liabilities) have outstanding amount of missing values. Other attributes have missing value with variation in occurrence.

In addition, we wanted to understand the correlation between different features by generating a heatmap of attributes. Figure 3 displays the year 1 attributes correlation heatmap. Each square shows the correlation between the variables on each axis. Correlation ranges from -1 to 1. We observe that there are strong correlations between many attributes, for example X42 (profit on operating activities/sales) and X13 ((gross profit + depreciation)/sales) and etc. Such correlation information can help us determine relevant features in the later process.

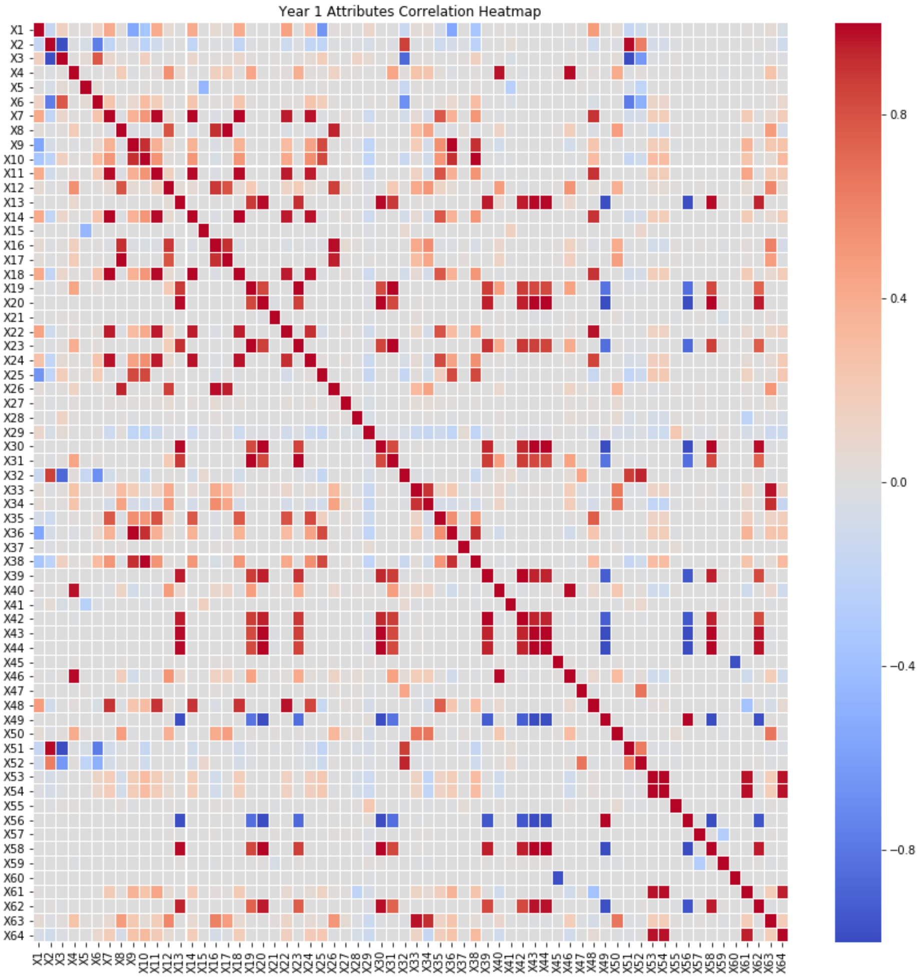

Figure 3 Heatmap of attributes correlation in year 1

Lastly, the raw data was visualized using t-SNE [8], which is a technique for the visualization of high-dimensional data. In Figure 4, we observe the non-linearity separation between the two classes. Therefore, we can infer that non-linear classifiers should outperform linear classifiers.

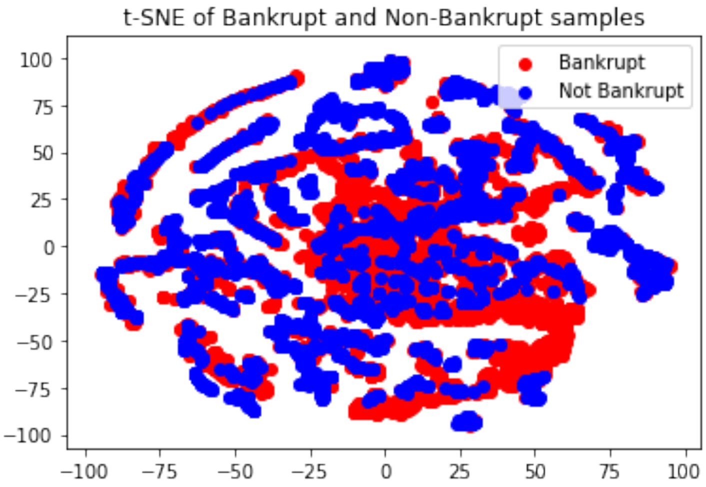

Figure 4 t-SNE visualization of Bankrupt and Non-Bankrupt samples

III. Approaches

3.1 Data Preprocessing

The data preparation stage was comprised of imputing missing values, resampling dataset to address imbalanced classes, normalizing data, removing outliers, and dimensionality reduction and feature selection. Three data sets were prepared following the process shown in Figure 5.

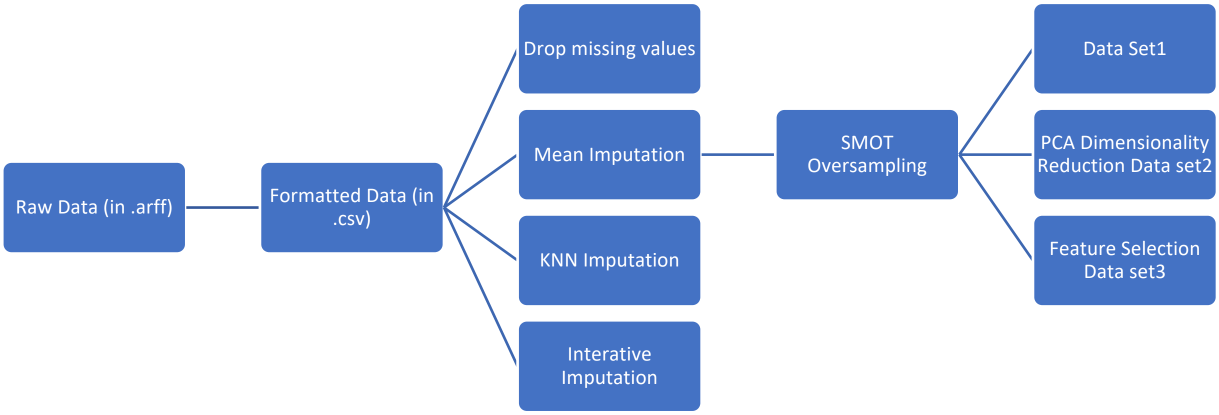

Figure 5 Data preprocessing flow chart

3.1.1 Missing Values Imputation

As we observed the frequent occurrence of missing values in our dataset, I explored several different approaches of imputing missing values and compared their model performance. The first approach implemented was to drop all missing terms. However, as we discussed, in some attributes, missing values make up the majority of the sample size. Simply dropping missing values may result in losing important information about the dataset. The second approach implemented was replacing all missing values with the mean value in the column. The third approach is the K-NN approach [12]. The algorithm uses 'feature similarity' to predict the missing value using the nearest-neighbor row or column. The last approach used is called the iterative imputation. The iterative imputation models [12] each feature with missing values as a function of other features and uses that estimate for imputation. To evaluate the quality of different approaches of handling missing values, a function was created to report the out-of-sample Mean Absolute Error (MAE) score [11]. The Mean Absolute Error is given by

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}$$

I split and fitted the first-year data set to a Random Forest Model[8] and calculated the Mean Absolute Error (MAE) [11] for each method with the predicted bankruptcy result and actual result. The second approach, imputing missing values with the mean values from the column achieved the minimum error score and this method was chosen to impute missing values.

Mean Absolute Error from dropping columns with missing values	0.04595
Mean Absolute Error from mean imputation	0.02844
Mean Absolute Error from KNN imputation	0.03997
Mean Absolute Error from iterative imputation	0.02881

Table 2 The mean absolute error for each missing value imputation method

3.1.2 Imbalanced Data Set Resampling

In addition to the significant amount of missing values in the dataset, we also observe that the data set contains imbalanced classes. The number of Non-Bankrupt companies is significantly more than the Bankrupt companies. I explored the Synthetic Minority Oversampling Technique (SMOT) [5] that uses a nearest neighbor's algorithm to generate new and synthetic data. SMOT [5] takes a sample from the dataset and consider its k nearest neighbors in the feature space. To create a synthetic data point, SMOT takes the vector between one of those k neighbors and the current data point [5]. It multiplies this vector by a random number x which lies between 0 and 1 and adds the current data point to create the new and synthetic data point [5]. Table 3 shows the number of samples in each class before and after SMOT resampling. After resampling process, the number of Bankrupt data is as the same as the number of Non-Bankrupt data.

	1st year dataset	2nd year dataset	3rd year dataset	4th year dataset	5th year dataset
Original Bankrupt sample size	271	400	495	515	410
Original Non- Bankrupt sample size	6756	9773	10008	9277	5500

Applied SMOT number of Bankrupt data	6756	9773	10008	9277	5500
Applied SMOT number of Non-Bankrupt data	6756	9773	10008	9277	5500

Table 3 The number of sampling points in each class before and after SMOT

3.1.3 Normalization and Outlier removal

All data were first normalized using z-scaling [6]

$$z = \frac{x - \mu}{\sigma}$$

, where μ is the mean of population and σ is the standard deviation of the population. Then, for each column, if any data are 3 standard deviation from the mean value in the column, it was identified as outlier and was removed from the data set [14]. The Figure 6 shows the pair plot of the first ten features in the data set before data normalization and outlier removal. We identified that some features, such as X4 is highly skewed and some features contains noticeable outliers, such as X3.

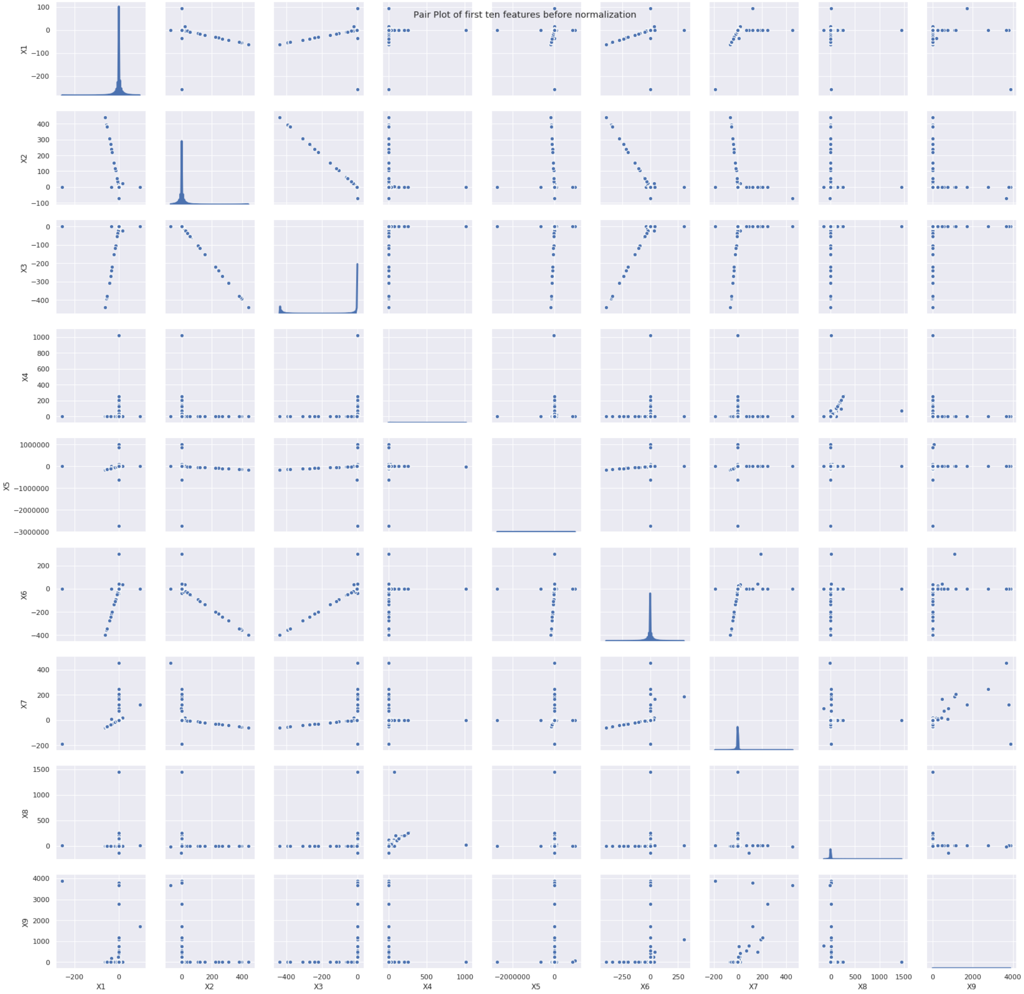

Figure 6 Pair Plot of first ten features before normalization

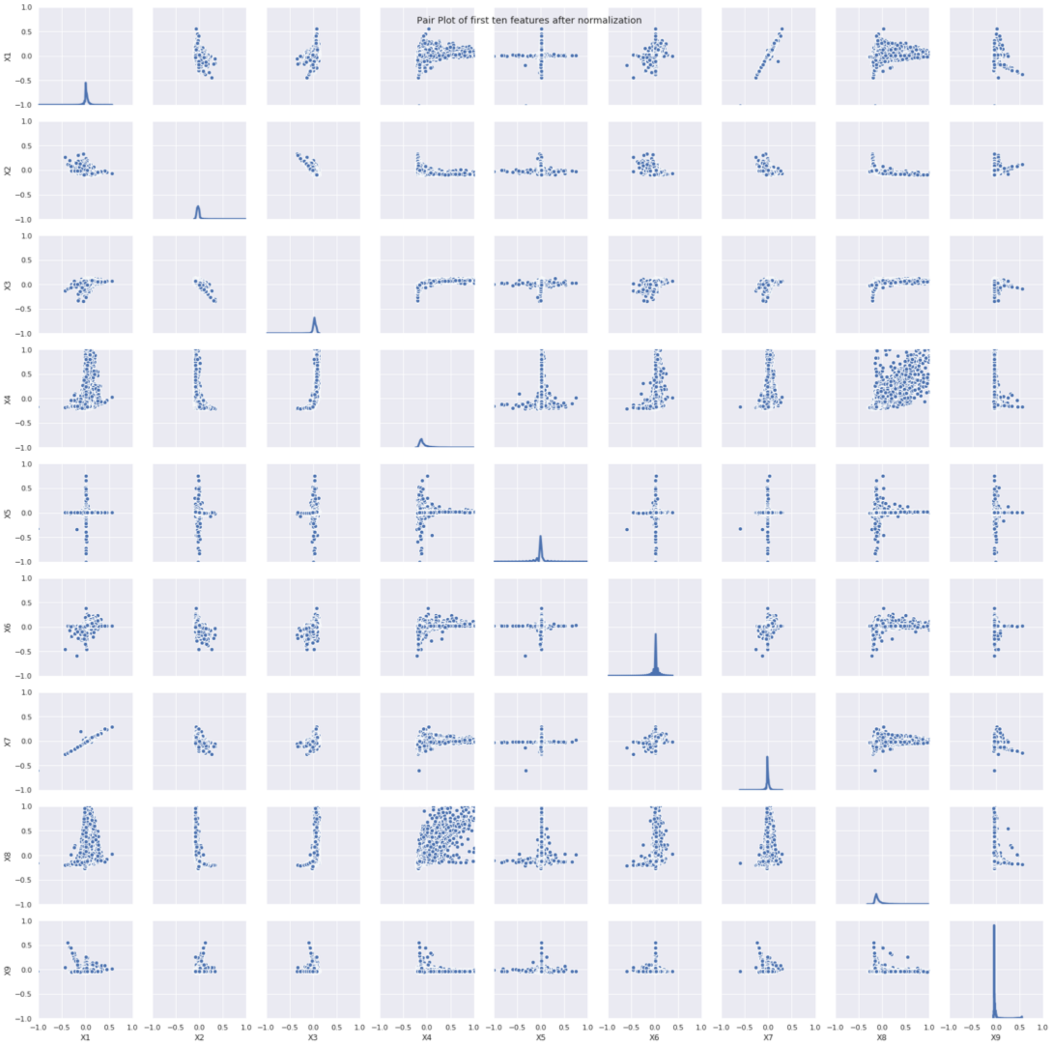

Figure 7 Pair Plot of first ten features after normalization

Figure 7 displays the pair plot of first ten features after normalization. We observe that the data is noticeably less skewed and noisy.

3.1.4 Dimensionality Reduction and Feature Selection

Given a total of 64 number of features in the data set, I explored both dimensionality reduction and feature selection on the data set. Principle component Analysis [8] was first applied. Figure 8 shows the explained and cumulative variance ratio with different number of principle components. Reading from the cumulative explained variance ratio, with keeping 20 principle components and above, we can explain more than 90% of the full variance. A data set with 20 principle components were prepared for prediction to compare with full 64 features data set. Though, later principle component axes still contribute to the full variance, it may worth to trade off some variance to gain computation efficiency if two data sets achieve similar model performance.

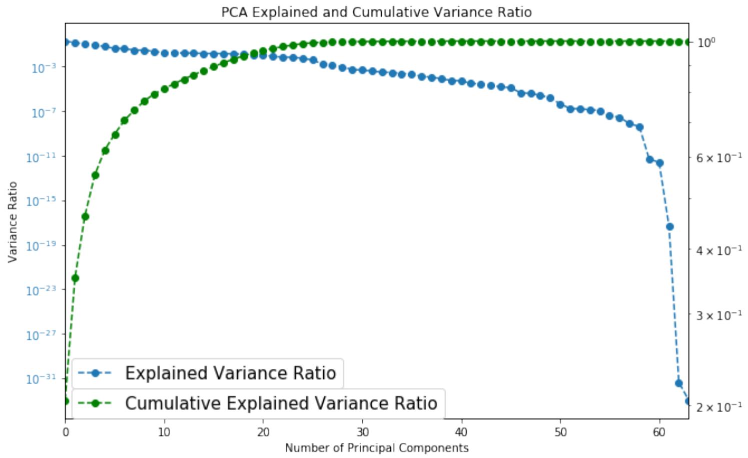

Figure 8 PCA explained and cumulative variance ratio

In addition, I explored the recursive feature selection [20]. Due to the large quantity of both features and observations in the data set, I reduced the sample size by randomly selecting 1000 observations. Recursive feature selection was applied to the training set by repeatedly creating models and keeping aside the worst performing feature at each iteration [20]. It constructs the next model with left features until all features are exhausted [20]. It then ranks feature based on the order of their elimination [20]. I plotted the number of features in the model along their cross-validated test score [21]. Figure 9 shows that the training set achieves the highest accuracy when 56 features are selected.

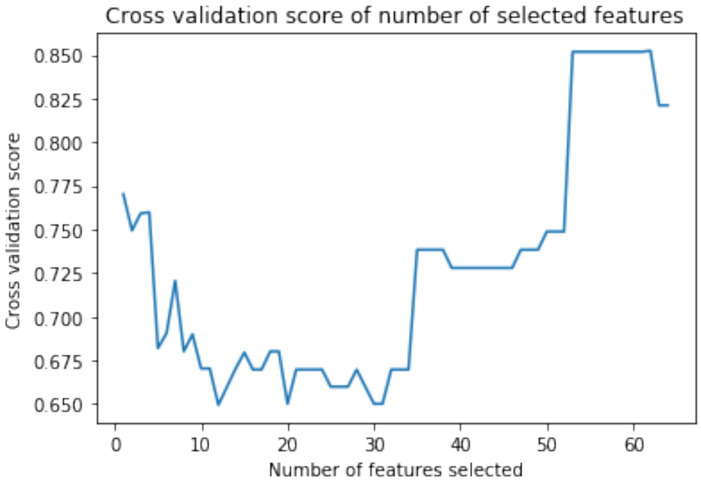

Figure 9 Cross validation score of number of selected features

After feature preprocessing, three data sets were prepared in total. One of which contains all 64 features, another one was kept with 20 principle components and the other one contains 56 features after step-forward feature selection. I will use all three feature sets for classification and understand the variance and bias trade off with dimensionality reduction and feature selection.

3.2 Classifiers

In this project, I considered three main classifiers for training our data sets, K-Nearest Neighbor, Support Vector Machine and Logistic Regression classifiers. From our t-SNE plot in Figure 4, we observed non-linear relationship between the two classifiers. Thus, non-linear classifiers were primarily chosen for classification, including K-Nearest Neighbor and Support Vector Machine. Both classifiers work well with large number of instances. Logistic Regression model was chosen as a linear benchmark classifier to compare with the other two, with the reasonable inference that KNN and SVM should outperform Logistic Regression.

3.2.1 K-Nearest Neighbor Classifier

The first classifier that I explored was KNN classifier. The K-Nearest Neighbors Algorithm is a non-parametric method used for classification and regression [8]. In the classification setting, the K-nearest neighbor algorithms essentially forms a majority vote between the K most similar instances to a given “unseen” observation [8]. Similarity is defined according to a distance metric between two data points. The distance we choose here is Euclidean distance given by

$$d(x,x') = \sqrt{(x_1 - x'_1)^2 + (x_2 - x'_2)^2 + \cdots + (x_n - x'_n)^2}$$

KNN runs through the whole dataset computing the distance, d between unseen observation, x and each training observations. It then estimates the conditional probability for each class

$$P(y = j|X = x) = \frac{1}{K}\sum I(y^{(i)} = j)$$

3.2.2 Support Vector Machine Classifier

The second classifier that I applied was Support Vector Machines. Support Vectors are the data points that lie closest to the decision surface [13]. Support Vector Machines uses numerical methods such as quadratic programming or Lagrange multiplier to maximize the margin around the separating hyperplane [13]. “Slack variable”, ε is introduced, which is the penalty for misclassification [8]. In our case, we want to find the maximum margin for linearly Non-separable data with penalty for points on the wrong side of margin boundary. Due to the non-linearity of the data, I want to apply RBF kernel to the SVM. The optimal diving hyperplane is shown as

$$w = \sum_{i \in SV} h_i y_i \phi(x_i)$$

And the offset of the hyperplane is

$$b = \frac{1}{|SV|}\sum_{i \in SV}(y_i - \sum_{j=1}^{N}(h_j y_j \phi(x_j)\phi(x_i))$$

The classification rule is

$$c = sign(< w, \phi(x) > +b)$$

, where x is a list of training vectors, y are respective labels, h is the Lagrangian coefficient and SV is set of Support Vectors. The kernel function is

$$\chi \times \chi \to R \text{ such that } k(x_i, x_j) = \exp\,(-\tfrac{\|x-x'\|^2}{2\sigma^2})$$

and it can replace the dot product above with $k(x_i, x_j)$ [13].

3.2.3 Logistic Regression Classifier

Another classifier that I explored was Logistics Regression. Logistic Regression assumes the log-likelihood can be modeled as a linear function of inputs [19]. The model Logistic Regression model is that

$$log \frac{p(x)}{1 - p(x)} = \beta_0 + x * \beta$$

Solving for p, this gives

$$p(x; b, w) = \frac{1}{1 + e^{-(\beta_0 + x * \beta)}}$$

The decision boundary separating the two predicted class is the solution of $\beta_0 + x * \beta = 0$ [19].

IV. Results

After designing the classifier, I split the 80% data as training data and 20% data as testing data. I applied 10 folds cross validation to all three training data sets, one contains all 64 features, another one after dimensionality reduction, and the other one after feature selection. Hyperparameters were explored using grid search with 10-fold cross validation. Two metrics, the area under the ROC curve and its accuracy were calculated to evaluate the performance of the classifier.

4.1 K-Nearest Neighbor Classifier

Experimenting with choosing different k values, the optimal k value falls in the range of 6 to 15, where we achieved the minimum error rate and highest accuracy. The result of apply KNN classifier when K=10 on preprocessed data set, PCA dimensionality reduced data set, and feature selected data set are shown below, respectively.

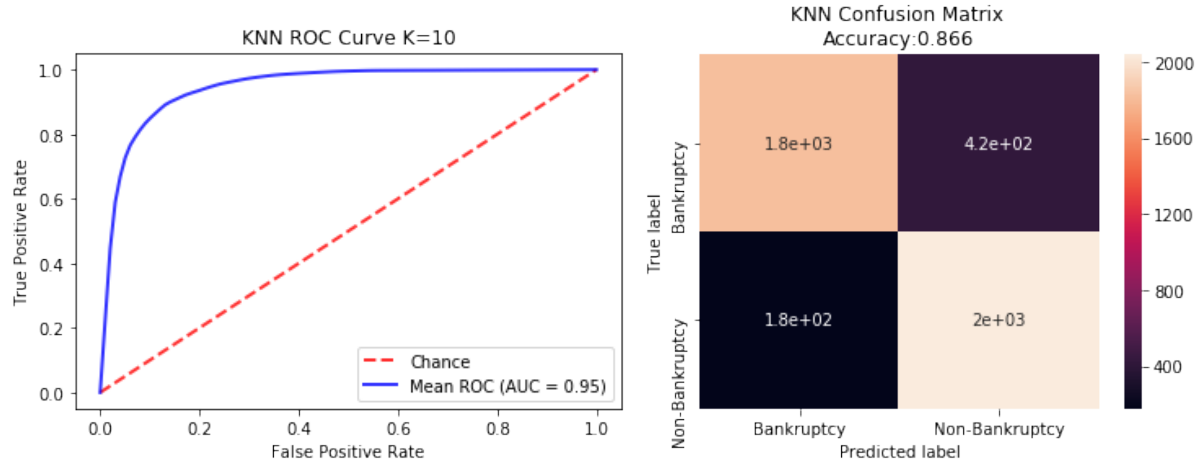

Figure 10 The ROC and confusion matrix of KNN applied on all 64 features data set, respectively

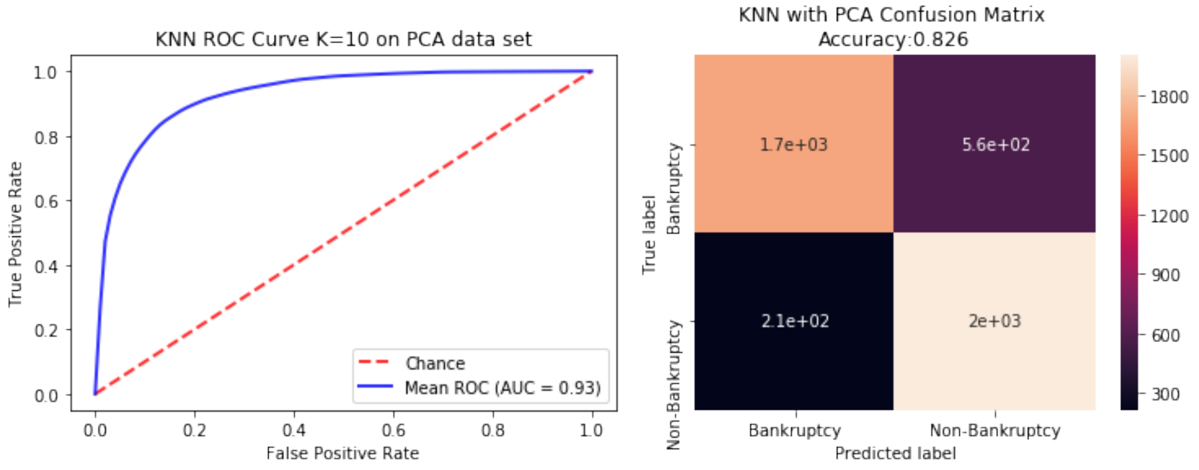

Figure 11 The ROC and confusion matrix of KNN applied on the dimensionality reduced data set, respectively

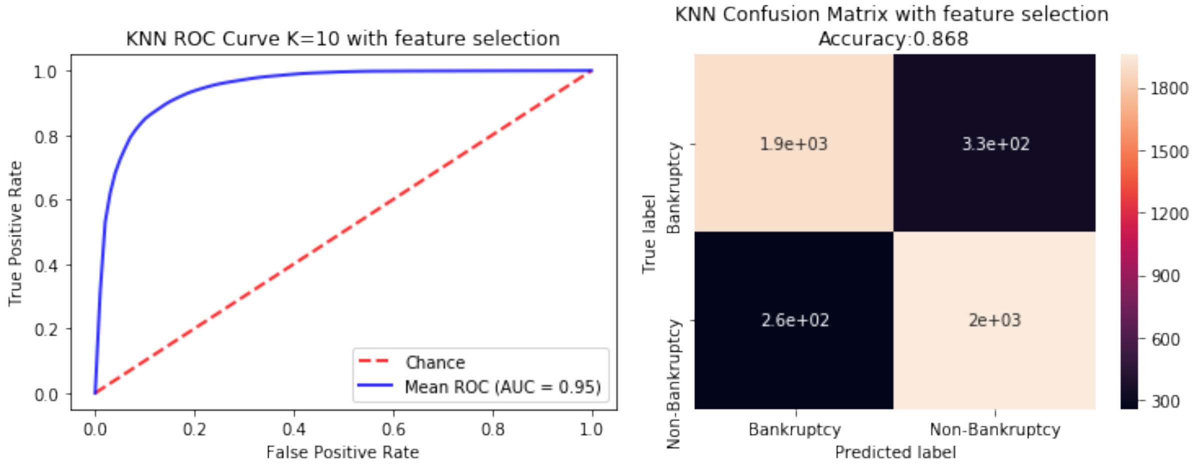

Figure 12 The ROC and confusion matrix of KNN applied on the feature selected data set, respectively

We observe that KNN performs well on both all 64 features data set and feature selected data set, with an accuracy of 0.866 and 0.868 in Figure 10 and 12. It is noticed that KNN performs slightly better on the dimension reduced data set than the full feature data set. However, KNN performs slightly worse on the PCA dataset, with an accuracy of 0.826 in Figure 11. It shows other non-selected principle components still make up an amount of variance in the data set.

4.2 Support Vector Machine Classifier

The second classifier I apply is SVM with RBF kernel [8]

$$k(x_i, x_j) = \exp\left(-\gamma||x - x'||^2\right)$$

With chosen C parameter and gamma parameter. C parameter trades off misclassification of training examples against simplicity of the decision surface [17]. If C value is large, then the model chooses more data points as a support vector and we get the higher variance and lower bias. If C value is small then the model chooses fewer data points as a support vector, resulting in low variance but high bias [17]. The gamma parameter can be viewed as the inverse of the radius of influence of samples selected by the model as support vectors [17]. As gamma increases, the decision boundary will depend on points closer to the decision boundary and vice versa. In this data set, tuning gamma values as 10 and C value as 1 yields the highest model accuracy. The results of apply SVM with RBF kernel on three data sets are shown below, respectively.

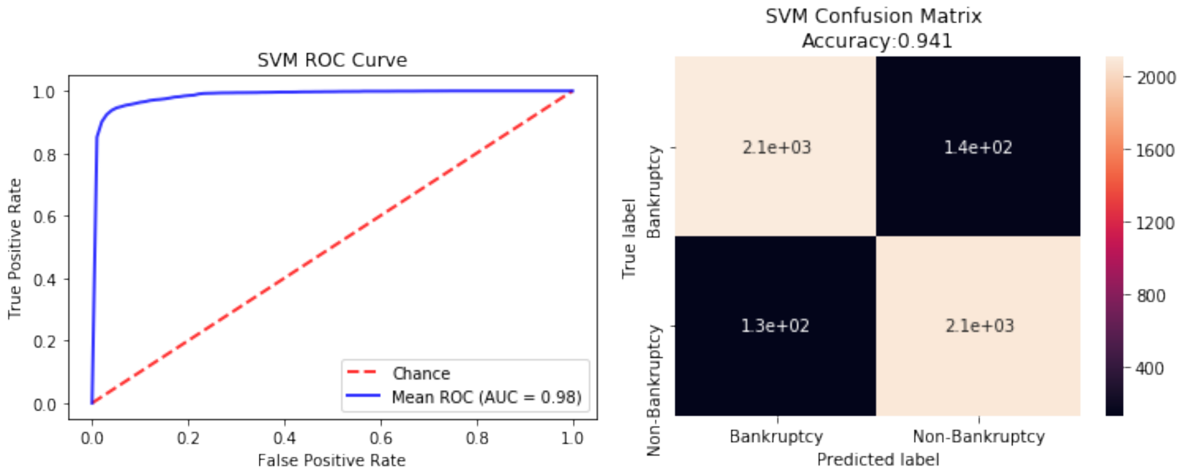

Figure 13 The ROC and confusion matrix of SVM applied on all 64 features data set, respectively

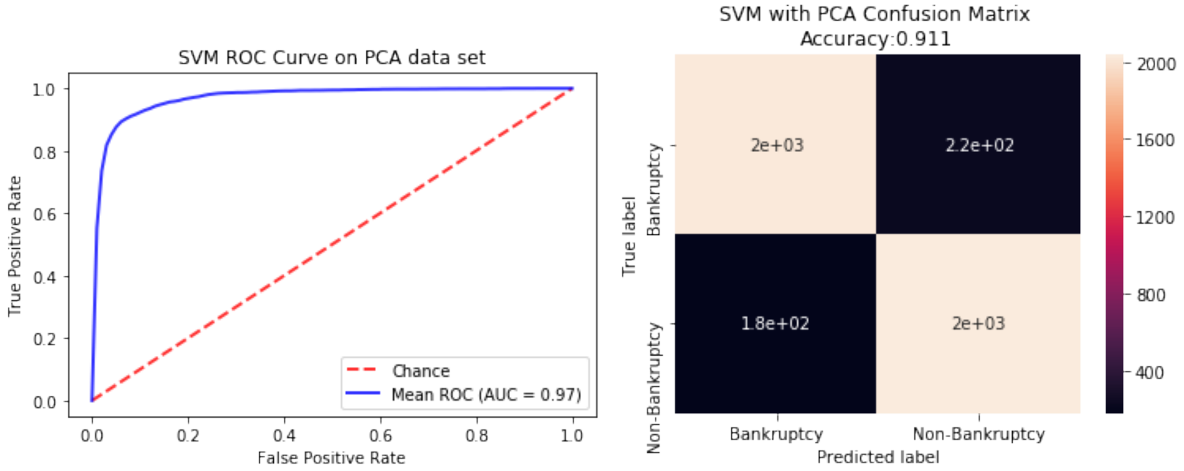

Figure 14 The ROC and confusion matrix of SVM applied the dimensionality reduced data set, respectively

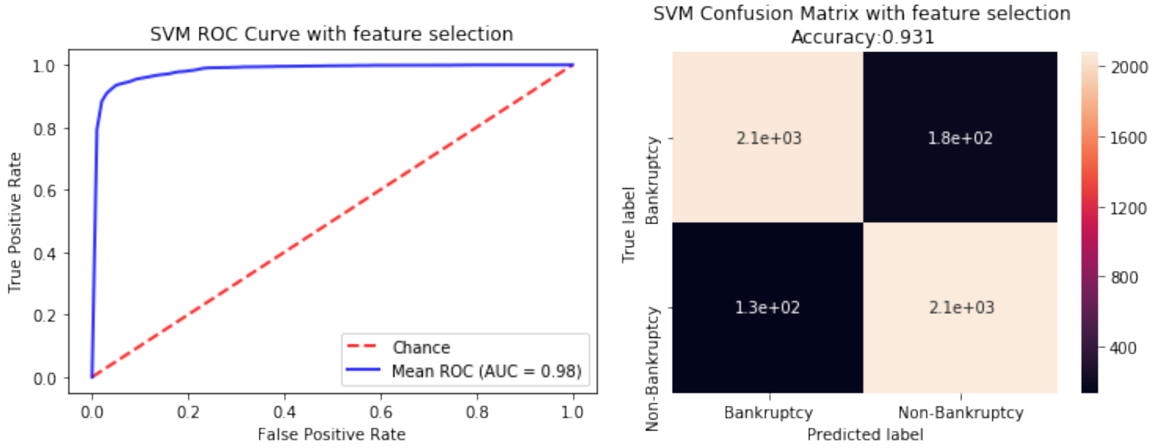

Figure 15 The ROC and confusion matrix of SVM applied on the feature selected data set, respectively

we observe that SVM achieves a higher accuracy and AUC score on all three data sets compared with KNN. Out of all three data sets, SVM achieves the highest accuracy and AUC score on the all 64 feature data sets in Figure 13. SVM works well with large dimension sets since the kernel maps data to high dimensional space. SVM also yields high accuracy and AUC score on the PCA reduced data set, shown in Figure 14. We infer that SVM is quite resistant to dimension change. If the prediction needs to be taken place constantly, PCA data set could be recommended to use to trade off some prediction accuracy for computation efficiency. The flexibility of tuning regularization C parameter as well as the kernel parameter help the model to achieve the optimal performance while avoiding under or overfitting on the data set.

4.3 Logistic Regression Classifier

The third classifier that I applied with was logistic regression. I implement the model with L1 regularization, Lasso Regression. Lasso regression adds an absolute value of magnitude of coefficient as penalty term

$$\lambda \sum_{j=1}^{P} |\beta_j|$$

to the loss function [18]. Lasso shrinks some less important feature's coefficient to zero, thus L1 regularization is believed to work well for large feature dimension data set.

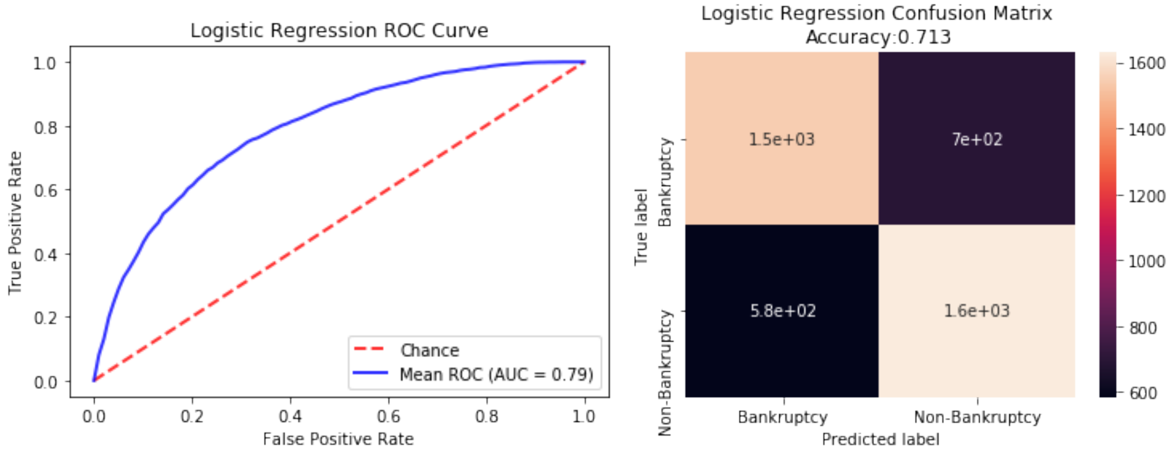

Figure 16 The ROC and confusion matrix of Logistic Regression applied on all 64 features data set, respectively

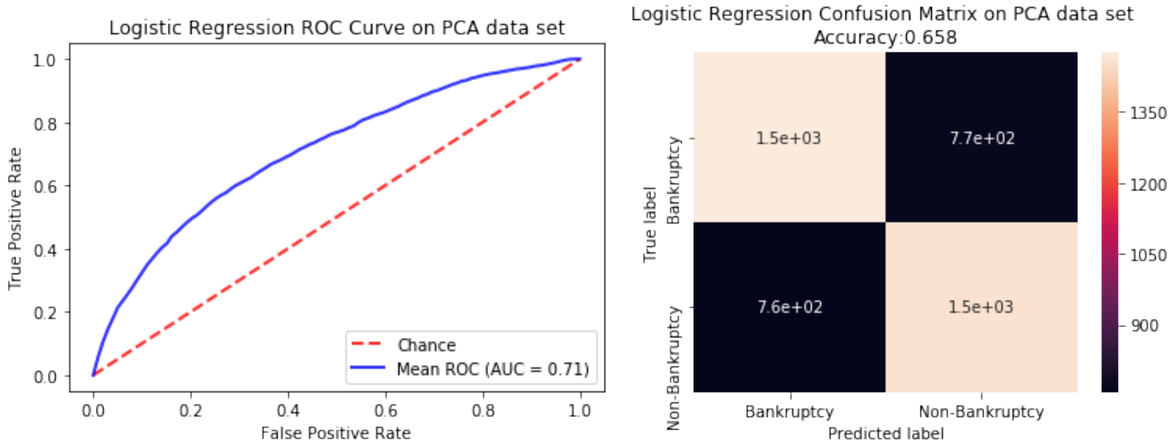

Figure 17 The ROC and confusion matrix of Logistic Regression applied on the dimensionality reduced data set, respectively

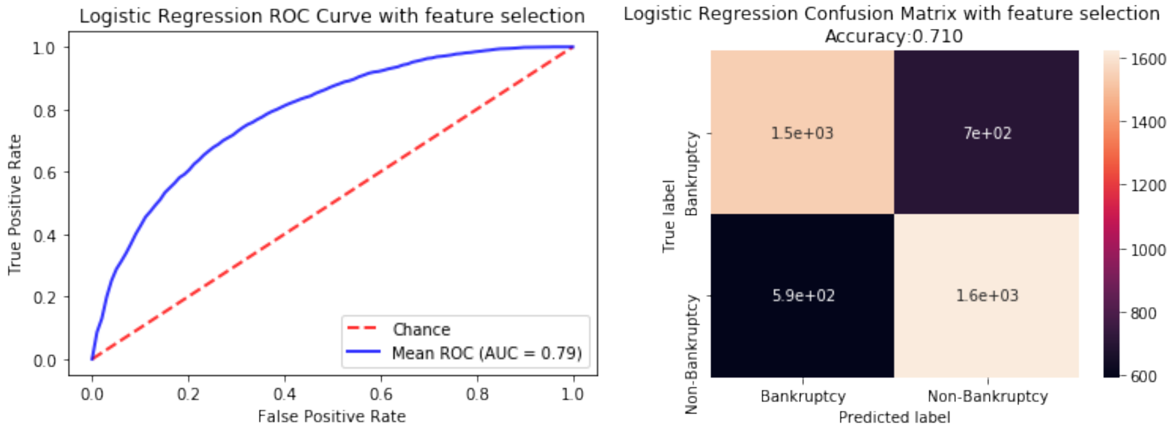

Figure 18 The ROC and confusion matrix of Logistic Regression applied on the feature selected data set, respectively

We observe that logistic regression does not perform as well as the other two classifiers, which is expected since logistic is a linear classifier. Out of three data sets that we trained, we recognize that logistic regression model performs better on featured selected data set and full features data set than dimensionality reduced data set, as shown in Figure 16 and 18, with an accuracy around 0.7. Logistic Regression performs the worst on the PCA reduced data set, as shown in Figure 17. The regularization term in in the logistic regression classifier helps to minimize the contribution of less important terms to our model. Therefore, it is rational to observe that feature selected data set and full features data set achieve very similar results.

V. Conclusion

In this project, we applied three classifiers, K-Nearest Neighbors, Support Vector Machine and Logistic Regression model to three different pre-processed data sets, full feature data set, dimensionality reduced data set, and feature selected data set. As we observed the non-linear relationship between two classes, Logistic Regression model was used as a benchmark model for our prediction. We expected that other two model should perform better than the Logistic Regression model.

Our classification results reveal that K-Nearest Neighbors and Support Vector Machine did achieve higher accuracy and AUC than Logistic Regression model. The quality of K-Nearest Neighbors model depends on the distance measure among all data points [15]. KNN is likely to perform well when the number of instances or observations is large, but dimension is small. KNN achieves a high accuracy on our data set at roughly 0.86. We also observe that the KNN performs slightly better on feature selected data set than non-feature selected data set, despite that the dimension of feature reduced data set is not much smaller than the full feature data set. This confirms that KNN is indeed more applicable to smaller dimensionality training set. KNN is a great measure when the number of observations is sufficient, and dimension is small. KNN in general is more suitable when sufficient domain knowledge is applicable, or feature selection can be performed [15]. The knowledge supports the selection of a group of relevant features for developing a high performance KNN model.

The SVM Model achieves the highest accuracy among all models trained, with the highest score at 0.94. The speculation is that the SVM model is able to include the kernel trick. The kernel trick transforms data in high dimensionality space and the boundary to classify between two classes is much more obvious [13]. Because of the use of kernel trick, SVM is resistant to dimensionality change. This is also observed in our pre-processed data; we did not see any performance improvement after feature selection. SVM also assumes that data it works with is in a standard range. Normalization of feature vector prior to fitting model is crucial in my experience. Lastly, SVM is also very computation costly, especially with the kernel trick. If top PCA principle components make up the full variance of the data set, PCA can be applied prior to SVM modeling for computation efficiency.

The Logistic Regression model was used as a baseline model for our performance evaluation. As expected, Logistic Regression achieves the minimum accuracy among all classifiers at 0.72. Logistic Regression performs the worst on the PCA dimensionality reduced data set. The speculation is that the feature loses interpretability after dimension reduction and we can’t ensure little or no multicollinearity between selected principle components.

Among three feature sets we pre-processed, feature selected data set and non-feature selected data set achieved very similar results. It is possible that the dimensionality suggested with the feature selection is not much smaller than the raw dimensionality. Since step wise feature selection process is extremely computation heavy, if higher computation machine were available, the feature selection process could be ran on the entire data set to obtain a more optimal number of features. In this particular data set, almost all features contribute to the model. PCA reduced data set in our case performs worse than the other two, meaning the non-selected principle components indeed contribute non-trivial amount to the whole data set variance. Dimensionality reduction is not recommended for this particular data set.

Overall, it is recommended to use SVM with RBF kernel to predict corporate bankruptcy. Data-preprocessing is the key to train model and achieve high accuracy in prediction. I identified some important steps in designing the pipeline, including handling missing values, handling imbalanced sets, data normalization and feature selection. If data set is imbalanced, the model will learn to predict the majority class and the model becomes not promising. Data normalization is necessary for stable convergence of weights during classification. Data visualization is also critical to understand the data set. After all, the classifier can be only properly designed knowing the data set and data distribution. In this project, I identified feature selection was the most challenging process. This may due to the lack of financial or economical domain knowledge. If domain knowledge were applicable, the feature selection can be optimized, and step wise feature selection results can be more interpretable.

Reference

[1] Ahmadi, Zahra, et al. “Towards Bankruptcy Prediction: Deep Sentiment Mining to Detect Financial Distress from Business Management Reports.” *Towards Bankruptcy Prediction: Deep Sentiment Mining to Detect Financial Distress from Business Management Reports - IEEE Conference Publication*, ieeexplore.ieee.org/document/8631483.

[2] Akinfaderin, Wale, and Wale Akinfaderin. “Missing Data Conundrum: Exploration and Imputation Techniques.” *Medium*, IBM Watson Data, 11 Sept. 2017, medium.com/ibm-data-science-experience/missing-data-conundrum-exploration-and-imputation-techniques-9f40abe0fd87.

[3] Alaka, Hafiz A, et al. “Systematic Review of Bankruptcy Prediction Models: Towards a Framework for Tool Selection.” *Expert Systems with Applications*, Elsevier, 26 Oct. 2017, www.sciencedirect.com/science/article/pii/S0957417417307224.

[4] Anjum, and Sanobar. “Business Bankruptcy Prediction Models: A Significant Study of the Altman's Z-Score Model.” *SSRN*, 13 Aug. 2012, papers.ssrn.com/sol3/papers.cfm?abstract_id=2128475.

[5] Chawla, Nitesh, et al. *SMOTE: Synthetic Minority Over-Sampling Technique*. www.cs.cmu.edu/afs/cs/project/jair/pub/volume16/chawla02a-html/chawla2002.html.

[6] Erwin Kreyszig. *Advanced Engineering Mathematics* (Fourth ed.). Wiley. p. 880, eq. 5. ISBN 0-471-02140-7.

[7] Graham, John R, et al. *The Labor Impact of Corporate Bankruptcy: Evidence from Worker-Firm Matched Data*. 2014, www.sole-jole.org/15541.pdf.

[8] James, Gareth, et al. *An Introduction to Statistical Learning: with Applications in R*. Springer, 2017.

[9] Tomczak, Sebastian. *UCI Machine Learning Repository: Polish Companies Bankruptcy Data Data Set*, archive.ics.uci.edu/ml/datasets/Polish companies bankruptcy data.

[10] Wagenmans, Frank. *Machine Learning in Bankruptcy Prediction*. Universiteit Utrecht, 3 July 2017.

[11] Willmott, Cort J.; Matsuura, Kenji. "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance". *Climate Research*. **30**: 79–82. doi:10.3354/cr030079

[12] “Fancyimpute.” *PyPI*, pypi.org/project/fancyimpute/0.0.4/.

[13] Ng, Andrew. “CS229 Lecture Notes.” *Http://cs229.Stanford.edu/Notes/cs229-notes3.Pdf*.

[14] “What Are Outliers in the Data.” *Engineering Statistics Handbook*, www.itl.nist.gov/div898/handbook/prc/section1/prc16.htm.

[15]“Usage of KNN.” *IBM Knowledge Center*, www.ibm.com/support/knowledgecenter/en/SS6NHC/com.ibm.swg.im.dashdb.analytics.doc/doc/r_knn_usage.html.

[16] G. C. Cawley and N. L. C. Talbot, Preventing over-fitting in model selection via Bayesian regularization of the hyper-parameters, Journal of Machine Learning Research, volume 8, pages 841-861, April 2007.

[17] Billard, Aude. *Advanced Machine Learning (SVM, RVM & AdaBoost)*. pdfs.semanticscholar.org/baec/33ac0714d00f9cd87d21b1fd966810112670.pdf.

[18] Bacallado, Sergio. *Lecture 14: Shrinkage*. web.stanford.edu/class/stats202/content/lec14.pdf.

[19] *Logistic Regression*. www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ch12.pdf.

[20] Kaushik, Saurav, and Saurav. “Feature Selection Methods with Example (Variable Selection Methods).” *Analytics Vidhya*, 12 Mar. 2019, www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/.

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