Company Bankruptcy Prediction Proposal

Reed Grimm
North Carolina State University
Raleigh, North Carolina, USA
rsgrimm@ncsu.edu

Apurva Sandeep Sonavane
North Carolina State University
Raleigh, North Carolina, USA
asonava@ncsu.edu

Michał Stępień
North Carolina State University
Raleigh, North Carolina, USA
mstepie@ncsu.edu

Dave Balaji Talari
North Carolina State University
Raleigh, North Carolina, USA
dtalari@ncsu.edu

1 INTRODUCTION AND BACKGROUND

1.1 Problem Statement

The financial health of a company is a topic of great concern both within and outside that business, with the most adverse outcome of poor financial health being bankruptcy. This negatively impacts business employees and investors directly, but ripple effects can felt in the broader community and across financial markets and economies. When companies declare bankruptcy en masse, it can result in overall economic decline.

Thus, the ability to assess the present and future financial health of businesses is highly desirable. Knowledge of which financial metrics are indicators or predictors of financial health is also highly important, as it would allow for more nuanced analysis of how changes in those metrics would impact the overall business financial stability. Currently, it is not always readily apparent when a business is on the decline or at risk of going bankrupt.

To address these problems, we analysed financial ratios across many companies and whether those companies eventually declared bankruptcy. Other research has shown that bankruptcy can be fairly accurately predicted from such financial data.

Keywords: Bankruptcy, Feature Selection, Model Selection, Random Forest, Pearson Coefficient, RFECV

1.2 Related Work

As bankruptcy prediction is a topic of broad interest, there have been many previous attempts to use financial data to predict bankruptcy, dating back to the 1960s with the use of traditional statistical methods. A few papers in particular discuss the various methods and models used in these attempts and compare their results.

Qu, Quan, Lei, and Shi gives a historical overview of methods used on various datasets and discusses their effectiveness of bankruptcy prediction in those separate contexts.

Barboza, Kimura, and Altman compared many traditional statistical methods and machine learning models (including neural networks [NN], support-vector machines [SVM], boosting, bagging, and random forest [RF]) on the same dataset and their performances.

Liang et al. similarly explored predicting bankruptcy with the same dataset as the one used in this study, only including additional corporate government indicators.

2 METHODOLOGY

2.1 Approach

Feature selection is going to be a big part of the scope of this project in this report. Since we have 96 attributes (including the class to predict), trimming the attributes down to decrease model complexity becomes one of the most primary concerns. We plan to focus most of our attention on feature selection. There are two methods of feature selection, namely:

1. Feature Elimination
2. Feature Extraction

Within feature elimination we have several techniques that we explored:

- Filter method
- Recursive Feature Elimination using Cross validation
- Embedded method (Binomial Logistic Regression with Lasso Regularization)
- Permutation Feature Importance

(1) **Pearson coefficient for correlation amongst features:** In the filter method, we manually eliminated features using Pearson correlation coefficient. The Pearson correlation calculates the linear correlation by using the below formula:

$$\rho_{xy} = \frac{Cov(x,y)}{\sigma_x \sigma_y} \tag{1}$$

It is the co-variance of two variables scaled down through division by the product of their respective standard deviations. Pearson correlation is in the range of +1 to -1 due to the scaling. +1 means perfect positive correlation, in which both the entities increase together in alignment, -1 means a perfect negative correlation, which means that one increases while the other decreases. We have used the Pearson correlation to trim down on our feature space. If it is not related to the Y variable that is to be predicted, we remove it. We have also used it to drop features from the X space if they are highly correlated amongst each other. The novelty about using this approach to us was that we compared Pearson to the other coefficients and wanted to conduct exploratory data analysis manually before we started using algorithms.

(2) **Recursive Feature Elimination using Cross Validation:** We also perform Recursive Feature Elimination with Cross Validation. Recursive Feature elimination is an interpretative method of feature elimination. It starts by taking all of the

features at once, ranking the features in order of importance, and then building a model on top of it to measure the accuracy. It then removes one feature at a time starting with the lowest rank and keeps iterating through the loop above. It uses accuracy as a metric to score the model in our case. After the cycle is completed, it chooses the model with the highest accuracy. We can obtain everything, from the optimal number of features to select, to the ranks of the features. To tune this model, we can use various classifiers and cross validation parameters. This usually takes an extremely long time to run depending on the size of the feature space.
It is a wrapper method. In this case, the wrapper section of this algorithm is the feature eliminator, the core of this remains a classifier that is used multiple times to look for the most optimal solution.
The novelty in this approach is that we will use recursive feature selection, in combination with the filter approach using random forest classifier. Besides this, we are using the SVC classifier without filter to compare the results that both the models will achieve.

(3) **Embedded method (Binomial Logistic Regression with Lasso Regularization)** Logistic regression is used for classification especially when the data has binary classes to predict. We apply the regularization factor of L1 i.e. Lasso regularization. When the original error is the same with or without the inclusion of an attribute, but the regularization factor increases, in turn increasing the error overall, the model decides that the feature is an unnecessary addition. It is eliminated by making its coefficient equal to 0. This encourages a more generalised, less over-fitted and a simpler model.
Usually people use L1 regularizer for reducing the coefficients to zero. In our case, since it was a classification problem, we decided to use Logistic regression. We had to specifically look for a way to combine logistic regression with Lasso for our specific problem statement.

(4) **Permutation Feature Importance** Permutation Feature Importance uses an estimator like Random Forest and it fits the model entirely on the training data set. On the test data set, which is different from the training data set, we shuffle the values of a single feature, and observe the reduction in the scoring metric. This process is followed for every single feature. This metric referred to is usually model accuracy. The sklearn function fetches an importance mean for every feature.This is the difference between the baseline metric and the new accuracy after the feature value shuffle. If we get high positive differences for a feature, the feature is relevant. In this approach, it finds the correlation of a feature to the classification label using a novel approach. If the shuffled values of a parameter still yield a good result, this means that the feature we shuffled on is irrelevant and can be discarded, it does not really make any impact on the result.
Now talking of Feature Extraction, we have used **PCA**

(5) **Principal Component Analysis** Principal component analysis transforms the feature set in a linear manner to create another view on the dataset that captures the most variance in the data. This is especially useful when we want to cut down our features that do not contribute as much, but we still want them to serve their function. Even after we cut off a few features, the new features are the combinations of the old ones, we are still retaining the valuable parts of our old variables.
All of our other methods focus on feature elimination. PCA reduces the dimensionality while avoiding loss of information. It reduces the interpretability but retains the information. The novelty that we are going to perform on PCA is that we are going to run it in combination with stratified and SMOTE sampling methods after all of the feature elimination is performed.

(6) **Standard Scaler** Standard Scaler ensures that all of the values will have a mean of 0 and a variance of 1. Essentially, it performs z-score normalization of the data. The novelty of this approach is that we are going to compare Standard Scaler to all the other Scaler functions and non-linear transformations, to find out which transformation yields the best accuracy.

(7) **MinMax Scaler** MinMax Scaler ensures that all of the values will be in the range of 0 and 1 for all the features. This however, does not mean that they are not sensitive to outliers. They just push all of their inliners to be within the range of [0, 0.005]. The novelty of MinMax Scalers is that we have not used them in class and all of our group members have only ever used Standard Scalers before this

(8) **Max Absolute Scaler** Max Absolute Scaler just scales the features to be between -1 and 1. This novelty of this approach is that we are we have not used this function in class and we are going to compare all the other scalers to this.

(9) **Robust Scaler**.
Robust Scaler scales the data according to the Inter-Quartile Range, hence it is robust to outliers unlike previously explained methodologies. This method uses the median and the interquartile range, hence, outliers do not affect this.

(10) **Power Transformer**
The novelty of this approach is that this is used to when there is a lot of varied variance across the features in the data set. When we want a normalized bell curve(Gaussian) across the data set, we use the monotonic power transformer function. Unless specified otherwise, we will get a unit variance and zero mean transformation. Within the two methods allowed, we use Yeo-Johnson, this supports both positive and negative values. The novelty in this approach is that this method knows how to handle constantly changing variances within the dataset by transforming the dataset to a gaussian distribution.

(11) **Quantile Transformer** It is used to obtain a uniform or a normal distribution out of the feature set. This is robust to outliers as, for every attribute, the transformation tends to spread out the most commonly occurring values. To begin with, an approximation of the cumulative distributive function is applied to a feature to obtain a uniform distribution. These values are then modified to be in the preferred distribution through the associated quantile function. This is a non-linear transformation and it changes the correlation between different features if they are scaled similarly. We

have used the Gaussian and uniform distribution to be the output distribution in our project. The novelty in this approach is that Quantile transformers are non-linear in nature and they can transform the data to a custom distribution while handling outliers.

(12) **Normalizer** This function normalizes the data one row at a time. Unlike other scalers, it does not go feature wise. Normalize samples individually to unit norm. The novelty in this method is the way the normalization functions row-wise in case of non-zero inputs.

(13) **SMOTE** SMOTE is Synthetic Minority Oversampling Technique. This upsamples the minority class for balancing the dataset. It does so by creating synthetic datapoints that are arbitrarily close in the feature space of the class being upsampled. The approach it uses is novel and we wanted to explore this further.

(14) **Stratified sampling** Stratified sampling ensures that we take equal amounts of data from every class. This does not create new samples but samples the underrepresented class set with repetition. Stratified sampling ensures that the training runs on only real datapoints.

2.2 Rationale

(1) **Pearson coefficient for correlation amongst features:**
Pearson coefficient is used for detecting linear relationships between variables. It shows us the correlation without the causation. Since we have 96 numerical attributes, many of them are highly similar to each other:
Example: ' Net profit before tax/Paid-in capital', ' Per Share Net profit before tax (Yuan ¥)', This is further explained in the experiment section. We decided to eliminate the intercorrelated features from our dataset. Besides, high intercorrelation, we also use this method to remove features that are not linearly dependent on the class predicted. We are still exploring this particular space of the problem, for now, our non-linear dependencies are taken care of by the random forest classifier that we ran for Recursive Feature Elimination. Hence, we did not use Spearman rank correlation on the data that was filtered through Pearson because, a random forest was run on this data.

(2) **Recursive Feature Elimination using Cross Validation:**
We chose recursive feature elimination using cross validation because it is the most comprehensive of all the approaches presented to us in wrapper methods.
In both forward and backward selection methods, we need to specify the number of attributes to be selected in the subset from the set of features. In variance threshold, they just use the variance again to limit the features. We have already done that using Pearson's correlation coefficient.
In RFECV, the method not only gives us the most optimal number of features using cross validation, but it also gives us the features to be selected ranked according to their importance. We have the option to use stratified K fold sampling. Our data set is imbalanced and this option really helps, since it shuffles all of the data around to ensure that every fold is balanced in the way the classes are represented within the fold equally.

(3) **Embedded method (Logistic Regression with Lasso Regularization):**
We specifically learnt that Lasso is a strict regularizer as compared to ridge. It removes more attributes and turns the coefficients to 0. In terms of regression, Logistic is the only one that we can use, since the others are not used for classification. Lasso specifically has a solver algorithm that supports binary classes called "liblinear". Hence, it was the ideal choice for our data set. Logistic regression is also specifically useful when the attributes are ratio, interval, nominal, or ordinal. Within our 96 attributes, we for sure have ratio and interval attributes and this model will flexibly fit on our given dataset. L1 regularizer is the main component that forces the coefficients to be 0 whenever it detects insignificance. We only use the coefficients in this case, not the prediction.

(4) **Permutation Feature Importance** We are trying to find a full proof method to eliminate irrelevant features from our data set since we have 95 columns. We tried using RFECV during our midterm report, for the same functionality. RFECV takes an extremely long time to run as it eliminates features from the entire pool of features. We wanted to compare the performance of RFECV to Permuation feature importance. Permutation Feature importance does not take as long to run and it almost uses the same methodology, but it gives a higher accuracy.

(5) **Principal Component Analysis** We are looking to reduce the dimensionality of our feature set. After trying to eliminate the features through so many interpretable methods, when we want to further cut down on the dimensionality beyond this, we try to remove the insignifcant parts of our existing features that have been selected using PCA.

(6) **Standard Scaler** We used standard scaler initially because it is one of the standard go-to ways to scale the dataset.

(7) **MinMax Scaler** We wanted to compare the performance of Standard Scalers with MinMax scalers as they compress the value to different ranges.

(8) **Max Absolute Scaler** To retain the sparsity of our dataset, we explored MaxAbsolute scaler.

(9) **Robust Scaler**. We used Robust Scaler because this scaler takes care of our possible outliers responsibly.

(10) **Power Transformer** Since all of our features have different ranges and variance, we decided to explore the power transformer function that smooths the positive and negative values using Yeo-Johnson into a gaussian curve.

(11) **Quantile Transformer** We used this transformation as it allows us to map our dataset into a distribution of our choice (gaussian and uniform) after initially mapping it onto a cumulative distributive function. This also handles outliers and it handles non-linearities.

(12) **Normalizer** For our dataset, normalizer is a good choice because we want to compare row wise normalizations to column wise.

(13) **SMOTE** Our data is imbalanced and we wanted to test which upsampling method is better. SMOTE generates more instances of the minority class, this is a data augmentation technique to ensure prediction accuracy. The number of companies that are about to go bankrupt will always be a

minority. To increase the number of instances of the 'Bankrupt?' true value, we utilise SMOTE.

(14) **Stratified** The shortcoming of SMOTE is that it creates a synthetic data. Stratified sampling fetches lesser data, but it is always true data. To check the performance of both of these techniques against one another, we have included both.

3 EXPERIMENT

3.1 Data set

The data used for the project was collected from the Taiwan Economic Journal for the years 1999 to 2009. This dataset contains 6819 samples and 96 attributes, 95 of which are financial ratios.

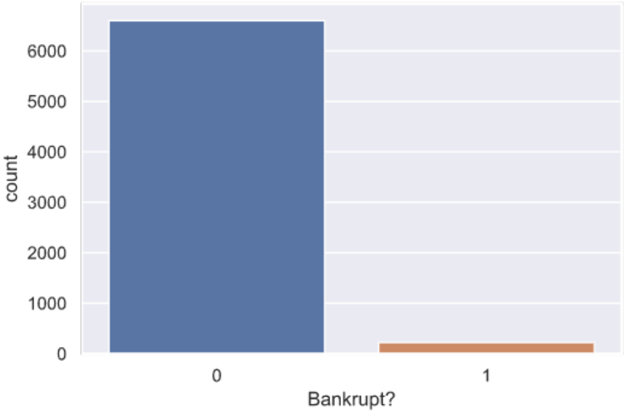

Figure 1: Data imbalance with regards to bankruptcy. Category 0 represents non-bankrupt companies, while category 1 bankrupt ones.

3.2 Hypotheses

The main goal of our project is to determine whether we can build a classifier that predicts a company bankruptcy status based on the financial dataset we use with the emphasis on maximizing macro average F-1 score and bankruptcy-class recall. Based on that we formulate our main hypothesis: ***Is it possible to train a machine learning classifier that predicts a company bankruptcy status with satisfactory F-1 score based on its financial data?*** Previous efforts to predict bankruptcy have achieved best performances of Type-I and Type-II around 20% each, which, for a binary-class problem, corresponds to a macro-F1 score of around 80%.

3.3 Experimental Design

To investigate our hypothesis we employ the methods described in section 2. We use Python to implement all the methods, and make use of Jupyter Notebook to build the code and present the findings. We import the dataset in csv file into the Jupyter Notebook using Pandas library, and store it as a Pandas DataFrame object.

The first data characteristic we analyze is the missing values in the dataset. Next, we look into the ratio of bankrupt to non-bankrupt companies. Subsequently, we investigate the variance of all the features in the dataset, with the plan to drop the ones that have 0 variance (measured in standard deviations). A feature that has 0 variance takes exactly the same value for every sample. Therefore, it is not predictive. After that we scale the feature set. In the project we consider scaling type as one of hyperparameters to tune, as we cannot be sure which one would be best. We used the following 7 types of scaling (using the corresponding sklearn objects): z-score normalization (using StandarScaler), scaling values between 0 and 1 (using MinMaxScaler), scaling values between -1 and 1 (using MaxAbsScaler), scaling values using Inter-Quartile Range (using RobustScaler), applying non-linear transformation to map features to a uniform distribution and Gaussian distribution (using QuantileTransformer with parameters output_distribution equal to 'uniform' and 'normal', respectively), applying power-like transformation to make the data more Gaussian-like (using PowerTransformer).

The next step in feature selection is correlation analysis. We only consider Pearson correlation coefficient to determine the correlation among the columns of the dataset. The calculation is handled by Pandas library. First we look into the pairwise correlation between every feature with the target (bankruptcy status). We decide to drop the features that are very weakly correlated with the target (we chose the threshold of 0.01 for Person correlation coefficient). Weak correlation with the target is an indicator of low predictive characteristic of a feature. Next we investigate the correlation between each and every pair of features that have been left in the dataset. We decide to drop the highly correlated features (we chose the threshold of 0.9 for the absolute value of Person correlation coefficient). The reasoning behind this is that highly correlated featured tend to convey similar information. Therefore, it is not ideal to keep such features, because they increase the dimensionality of the data without a strong influence on the trend.

After all those feature selection techniques we apply Principle Component Analysis using sklearn. We use elbow rule applied to the scree plot to decide on the number of principle components that will be used.

To prepare the dataset to train and evaluate the classifier, we performed train - validation - test split. We used sklearn's object train_test_split to do that. We decided to split the data in the following ratio: 60:20:20 train:validation:test.

Next on we tackle the problem of class imbalance. Here we try three strategies: stratified sampling, sample weighting, and Synthetic Minority Oversampling Technique (SMOTE). Later on we evaluate which on was the better choice.

Finally, we took many combinations of feature selection and data balancing methods and trained a random forest classifier model based upon each. The parameters of each random forest was the same (aside from sample weighting only being applied to certain configurations): 100 estimators and no maximum depth. Feature selection and data balancing methods were evaluated upon the performance of those models, where performance was evaluated based upon macro F1-score and bankruptcy-class performance. We also assessed the macro F1-score per attribute of each model, to gain a rough estimate of the ability of each configuration to choose only the most critical features.

4 RESULTS

4.1 Results

In this section we present the results we were able to obtain by implementing the methods outlined in section 3.3: Experimental Design. The analysis of the missing values in the dataset results in finding no missing values. Therefore we conclude that methods for handling missing data do not have to be considered further on.

The findings from checking the dataset with regards to the ratio of non-bankrupt to bankrupt companies are shown on Figure 1. We found the ratio of 30:1 for non-bankrupt to bankrupt companies. This signifies heavy data imbalance, which must be addressed to obtain a non-skewed model.

The variance analysis can be seen on Figure 2 and Figure 3. Figure 3 is the magnified view of the tail of the plot in Figure 2, because it is not readable due to the y-axis magnitude. This analysis shows that there is one feature that has zero variance. That feature is *Net Income Flag* and it takes a value of 1 for every single data point in the dataset. We drop it from the feature set for the reasons explained before. After that we are left with 94 features.

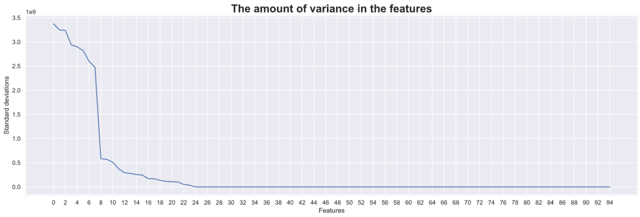

Figure 2: Variance analysis using standard deviation as a unit.

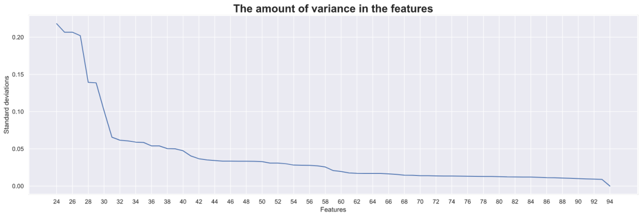

Figure 3: The tail of the variance analysis.

The pairwise correlation between the features and the target is presented on Figure 4. Pearson correlation coefficient has been used to measure the correlation. It can be seen on the plot that there are some features that have Pearson correlation coefficient smaller than 0.01 (23 features exactly). We drop those features. The list of those features: [*’Continuous Net Profit Growth Rate’, ’After-tax net Interest Rate’, ’Pre-tax net Interest Rate’, ’Continuous interest rate (after tax)’, ’Total income/Total expense’, ’Average Collection Days’, ’Operating Expense Rate’, ’No-credit Interval’, ’Interest Coverage Ratio (Interest expense to EBIT)’, ’Accounts Receivable Turnover’, ’ Revenue Per Share (Yuan ¥)’, ’Quick Assets/Current Liability’, ’Working capitcal Turnover Rate’, ’Allocation rate per person’, ’Interest Expense Ratio’, ’Current Ratio’, ’Inventory/Working Capital’, ’Inventory Turnover Rate (times)’, ’Inventory/Current Liability’, ’Long-term Liability to Current Assets’, ’Cash Flow to Sales’, ’Realized Sales Gross Profit Growth Rate’, ’Operating Profit Rate’*]. After this step we are left with 71 features.

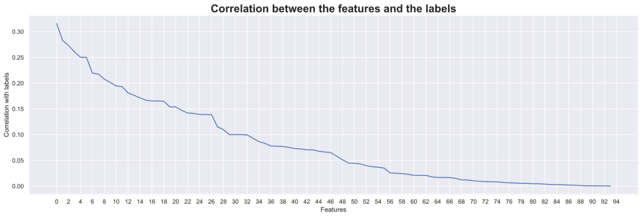

Figure 4: The pairwise correlation between the features and the target as Pearson correlation coefficient.

The pairwise correlation between the features themselves has also been measured using Pearson correlation coefficient. The visualization of the correlation matrix is not included in this report due to the unreadability of the heatmap. It can be viewed appropriately in our Jupyter Notebook in section 3.8.3 *Pairwise correlation between features*. We decided to drop one feature from each pair that has the absolute value of Pearson correlation coefficient larger than 0.9. From each pair, we drop the feature that has lower correlation with the target. We found 16 features to drop. The list of those features: [*’ROA(A) before interest and % after tax’, ’ROA(B) before interest and depreciation after tax’, ’ROA(C) before interest and depreciation before interest’, ’Debt %’, ’Net profit before tax/Paid-in capital’, ’Per Share Net profit before tax (Yuan ¥)’, ’Liability to Equity’, ’Net Value Per Share (B)’, ’Net Value Per Share (C)’, ’Current Liabilities/Equity’, ’Current Liability to Equity’, ’Operating profit/Paid-in capital’, ’Operating Gross Margin’, ’Realized Sales Gross Margin’, ’Regular Net Profit Growth Rate’, ’Current Liabilities/Liability’*]. After dropping those features, we are left with 55 features.

On addition of the variance and correlation analysis results, we apply recursive feature elimination. We used random forest as the estimator for RFECV. We evaluated the method using the accuracy of classification. We started with just 1 feature, and with every run we added 1 feature to the set, finishing at the full feature set of 55 attributes. The accuracy of each run of the algorithm was evaluated based on stratified 10-fold cross validation. The results are shown in Figure 5. The peak performance was achieved with the set of 54 features. However, the variance of accuracy starting at the set of 8 features is very low. This suggests that using all 54 features might be a bad trade-off with the complexity of the model if the accuracy with 8-feature set is very similar. Using accuracy as the metric also does not account for the imbalance of the data.

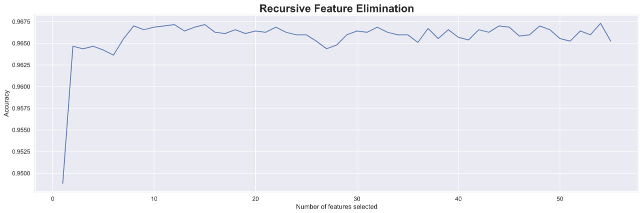

Figure 5: The accuracy of random forest classifier used in recursive feature selection.

We also performed recursive feature elimination on the raw feature set, without the elimination as the result of correlation analysis. For this experiment, we used support vector classifier as the estimator and evaluated the classification accuracy using stratified 15-fold cross validation. The results of this experiment can be seen in Figure 6. We see the peak performance with 14-feature set, however again, the variance of the accuracy is very low. We see a decline of accuracy with more than 50-feature sets, which can imply that the variance and correlation analysis described already may be very appropriate.

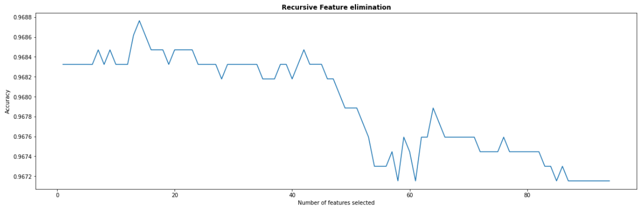

Figure 6: The accuracy of support vector classifier used in recursive feature selection.

For the Logistic regression with L1 regularizer, we obtained a subset of 74 attributes to retain, leaving 20 that can be discarded.

Total Attributes	Selected	Percentage decrease
94	74	21.27%

One method employed to address the high skew of the data was stratified data upsampling. Because non-bankrupt samples outnumber bankrupt samples by a factor of roughly 30, bankrupt samples were replicated by that factor to even the number of samples of each class. This increased the number of bankrupt samples from 220 to 6,599, bringing the total sample count from 6,819 to 13,198. Compare raw data counts in Figure 1 to the upsampled data counts in Figure 7. The SMOTE method of upsampling resulted in the same distribution, though the upsampling scheme was different.

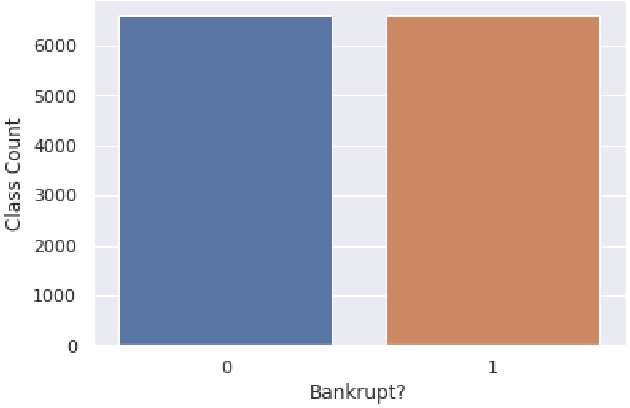

Figure 7: Data class counts after stratified upsampling.

After creating many different methods for selecting features, scaling/transforming data, and balancing data, multiple combinations of these methods were created and used to train RF classifier models. Those models were evaluated primarily upon macro F1-score. Figure 8 shows the macro F1-score of multiple configurations. However, it is worth nothing that model randomness did have in impact on model performance. Due to the close performance of these configurations, it is likely that relative ordering could vary by run.

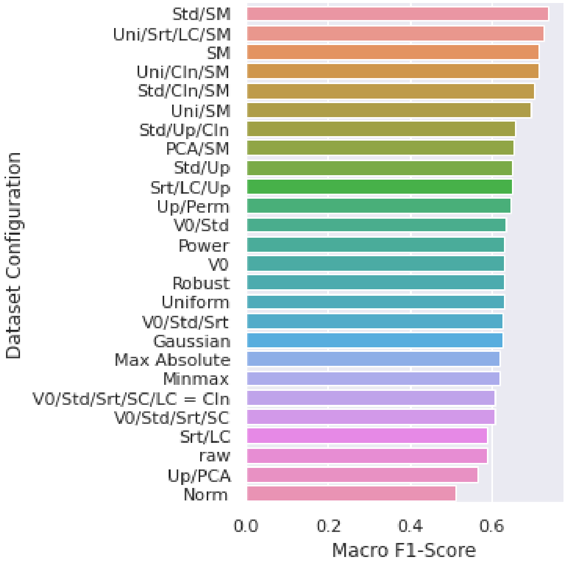

Figure 8: The test macro F1-score of RF models built from various configurations. There are many abbreviations to signify configuration elements: Cln = cleaned (V0,Std,Srt,SC,LC), LC = features with large correlation with other features removed, Perm = features with permutation feature importance < 0 removed, Raw = raw data, SC = features with small correlation with target removed, SM = SMOTE, Srt = features sorted, Std = standard scaler, Uni = uniform, Up = stratified upsampling, V0 = 0-variation features removed

The configuration that resulted in the highest macro F1 score of 0.74 was standard scaled and SMOTEd, and no feature selection method was employed.

4.2 Discussion

It is not surprising that the data configuration that performed best employed no feature selection scheme; at this scale, no data can harm the performance of a model. Unless there is too much data for a model to run in a sufficient time, feature selection is not necessary. However, these findings do reveal which feature selection methods are able to reduce data yet maintain high model performance for this application. Figure 9 reveals the macro F1 score divided by the number of features in the configuration. Taken with the overall macro F1 scores shown in Figure 8, this gives a rough indication of how capable a feature selection scheme is at balancing feature reduction with maintaining high performance. For instance, Uni/Srt/LC/SM configuration was able to achieve a similar F1-score to the top configuration, yet was able to do so with 55 features as

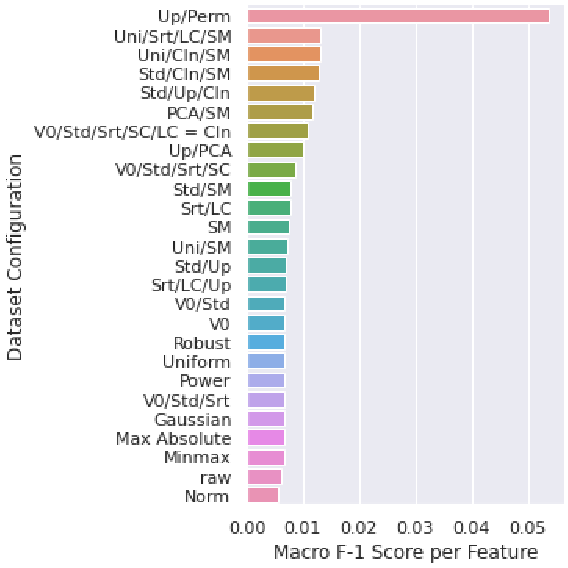

Figure 9: Test macro F1-score divided by number of features.

opposed to 94. This indicates that the removal of highly-correlated features (in conjunction with other data improvement methods) reduces data complexity with minimal information loss, which could prove critical when feature selection is absolutely necessary.

It is worth noting that seven of the top eight configurations employed SMOTE, with a definitive increase in performance in the top six. This highlights the usefulness of performing SMOTE. Furthermore, the following configurations are ones that employed stratified upsampling. Though SMOTE was clearly more effective at increasing the F1 score, balancing the dataset is clearly a critical step in achieving high performance.

Overall, we were able to achieve a top macro F1 score of 74%, which is close to the performance achieved by other efforts to accomplish this task. Liang et. al was able to achieve an F1 score of 80%, though with a vastly expanded dataset.

5 CONCLUSION

Our results show that given a data set that is highly skewed to one class as this one was. There are still a multitude of effective methods that can be utilized to select appropriate features, balance the class variable, and utilize various data transformations to produce models that can perform accurate predictions of bankruptcy using company financial data. In the case of feature selection we explored and employed the use of recursive feature elimination, logistic regression, permutation and permutation feature importance. For balancing the class variable we utilized stratified data upsampling and the smote method of upsampling. We also used PCA to reduce the dimensionality of the data. It became pretty apparent to us early on that evaluating the performance of our final models keeping various combinations of the above mentioned methods would have to be with something other than just an accuracy score due to the initial skewedness of our data and the nature of our data set. We chose to evaluate these using F1 scores. The model that had the highest F1 score turned out to be one which was standard scaled and SMOTEd, with no feature selection method employed. It seems that at current scale feature selection itself didn't have a large effect on results, however removing appropriate features through the various methods described above still didn't have as drastic of an effect on model performance than we initially expected to a certain point. In a more broad sense it seems that with this particular data set there are a few things to keep in mind and a few nuances to also consider. Firstly, skewedness of the class variable would need to be addressed in order to help a model improve the chances of learning for an unrepresented class, feature selection itself wasn't something in our data set that had a large effect on F1 score compared to some models which we failed to use feature selection altogether. This indicates that there is room to remove complexity while not hindering performance which could make a huge difference in a larger more broad data set. Randomness did have a slight effect on model performance and was noted as such. All in all we are pleased with what we were able to accomplish within the scope of our research and hope that our findings help push the needle forward when it comes to company bankruptcy predictions with similar data sets.

Click here to go to our github page

6 MEETING ATTENDANCE

Meeting Schedule: 4/14, 4/21, 4/28, and 4/30 from 12:30 to 1:30 pm

Member	4/14	4/21	4/28	4/30
Reed Grimm	Y	Y	Y	Y
Apurva Sandeep Sonavane	Y	Y	Y	Y
Michał Stępień	Y	Y	Y	Y
Dave Balaji Talari	Y	Y	Y	Y

REFERENCES

[1] Yi Qu, Pei Quan, Minglong Lei, and Yong Shi. 2019. Review of bankruptcy prediction using machine learning and deep learning techniques. Procedia Computer Science 162 (2019), 895–899. DOI:http://dx.doi.org/10.1016/j.procs.2019.12.065

[2] Flavio Barboza, Herbert Kimura, and Edward Altman. 2017. Machine learning models and bankruptcy prediction. Expert Systems with Applications 83 (2017), 405–417. DOI:http://dx.doi.org/10.1016/j.eswa.2017.04.006

[3] Deron Liang, Chia-Chi Lu, Chih-Fong Tsai, and Guan-An Shih. 2016. Financial ratios and corporate governance indicators in bankruptcy prediction: A comprehensive study. European Journal of Operational Research 252, 2 (2016), 561–572. DOI:http://dx.doi.org/10.1016/j.ejor.2016.01.012

[4] Deron Liang and Chih-Fong Tsai. 2021. Company Bankruptcy Prediction. Kaggle (February 2021). https://www.kaggle.com/fedesoriano/company-bankruptcy-prediction