Project Report - ORIE4741

Company Bankruptcy Prediction

Name	NetID
Christopher Hales	cmh342
Mathias de Bang	md794

12.05.2021

1 Introduction

Our project seeks to predict company bankruptcy based on data collected on bankruptcies from the Taiwan Economic Journal between 1999 and 2009.

Our objective is to answer whether it is possible to accurately predict whether a company will go bankrupt given the financial information that we have about all of the companies in the data set. Bankruptcy prediction is crucial in the financial industry for individuals and financial institutions make better assessments when it comes to credit and loans. Generally, the costs of these decisions involve a lot of resources for the institutions, and the large bankruptcies could have an affect on the rest of the market and economy as a whole. Therefore, the value of being able to accurately and efficiently predict which companies will end up bankrupt can not be overstated.

Fundamentally, these decisions have traditionally relied entirely on financial theory, but the increase in the amount of available data makes the application of machine learning enticing. The data set we have selected can make it possible to understand whether patterns in the financial data would enable us to predict which companies are at risk of bankruptcy (which will also be of interest to the company itself or other potential stakeholders).

Some specific questions of interest involve how we balance the ratios of false positive and negatives in our predictions to achieve the best results and most valuable predictions. We are mainly interested in identifying high-risk companies, which means we must consider the value of identifying a company with a high bankruptcy risk compared to falsely categorizing a company as being at the risk of bankruptcy (false positives). We also hope to gain insights into the most valuable predictors of bankruptcy. This could add to the wider understanding of financial statement analysis.

2 Data and Features

The data set that we are working with includes 6819 observations and 95 features about the financial and managerial accounting information for companies in Taiwan, including features such as the debt ratio percentage (liabilities / total assets), accounts receivable turnover, cash flow to sales, the ratio of gross profit to sales, and others, and of course if the company ended in bankruptcy or not. In all, it contains the quantifiable data about all of the financial information for each company recorded. The data was collected by the Taiwan Economic Journal for a decade: from 1999 - 2009. A description of the data shows that about 3.22% of companies went bankrupt during this time period.

Our data set does not contain any Null values, but there are two columns that do not accurately represent our data: 'Liability-Assets Flag' and 'Net Income Flag'. The 'Liability-Assets Flag' column has the Boolean value 0 for all but 8 of the rows, whereas the 'Net Income Flag' column has the Boolean value 1 for all entries.

Our next plan was to create a correlation matrix of all of our data. We did so over all of features in the data, and it was subsequently too small and convoluted to reasonably understand and interpret. As a result, we created a Series of all columns and their respective correlation to the Boolean column 'Bankrupt?' (returning 1 if bankrupt, 0 else), and then returned the ten features with the highest correlation coefficients. The correlation heatmap for all of the features is presented on the next page. These columns are as follows:

Debt ratio % (0.2502), ***Current Liability to Assets*** (0.194), ***Borrowing dependency*** (0.1765), ***Current Liability to Current Assets*** (0.1713), ***Liability to Equity*** (0.1668), ***Current Liabilities/Equity*** (0.154), ***Current Liability to Equity*** (0.1538), ***Liability-Assets Flag*** (0.1392), ***Total expense/Assets*** (0.139), ***Equity to Long-term Liability*** (0.139).

Noticing identical correlation values, further exploration of the columns shows that the columns Current Liabilities/Equity and Current Liability to Equity are identical columns, so we can ignore one of them. Thus, the top variables that we will be looking at are 'Debt ratio %', 'Current Liability to Assets', 'Borrowing dependency', 'Current Liability to Current Assets', 'Liability to Equity', 'Current Liabilities/Equity', 'Liability-Assets Flag', 'Total expense/Assets', 'Equity to Long-term Liability'.

The correlation heatmap for only these specific variables is also included on the next page.

Figure 1: Correlation Heatmap for All 96 Features

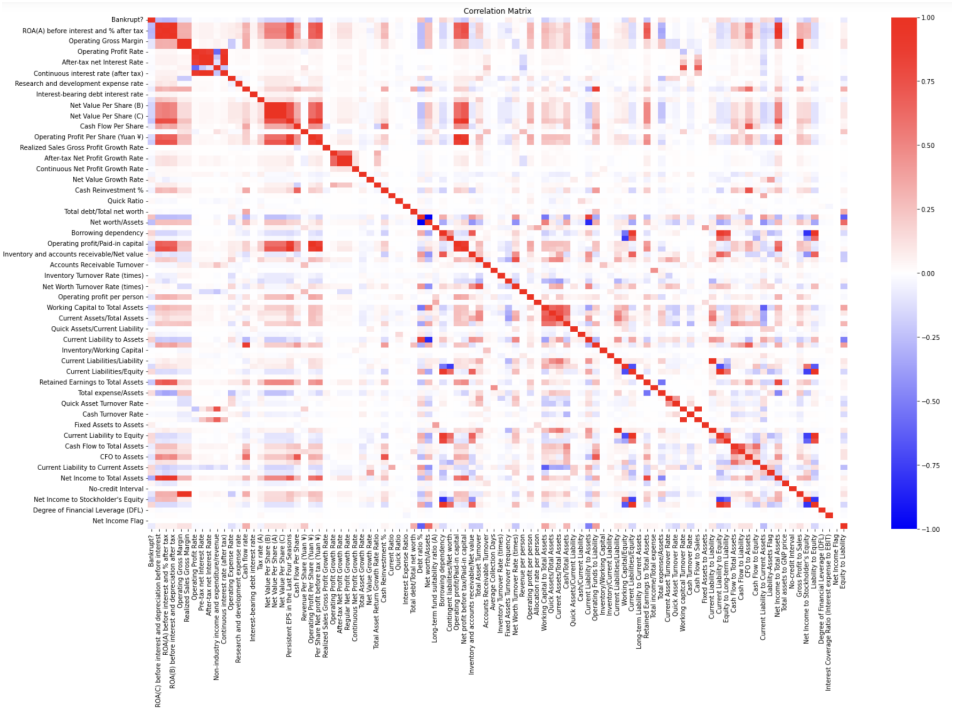
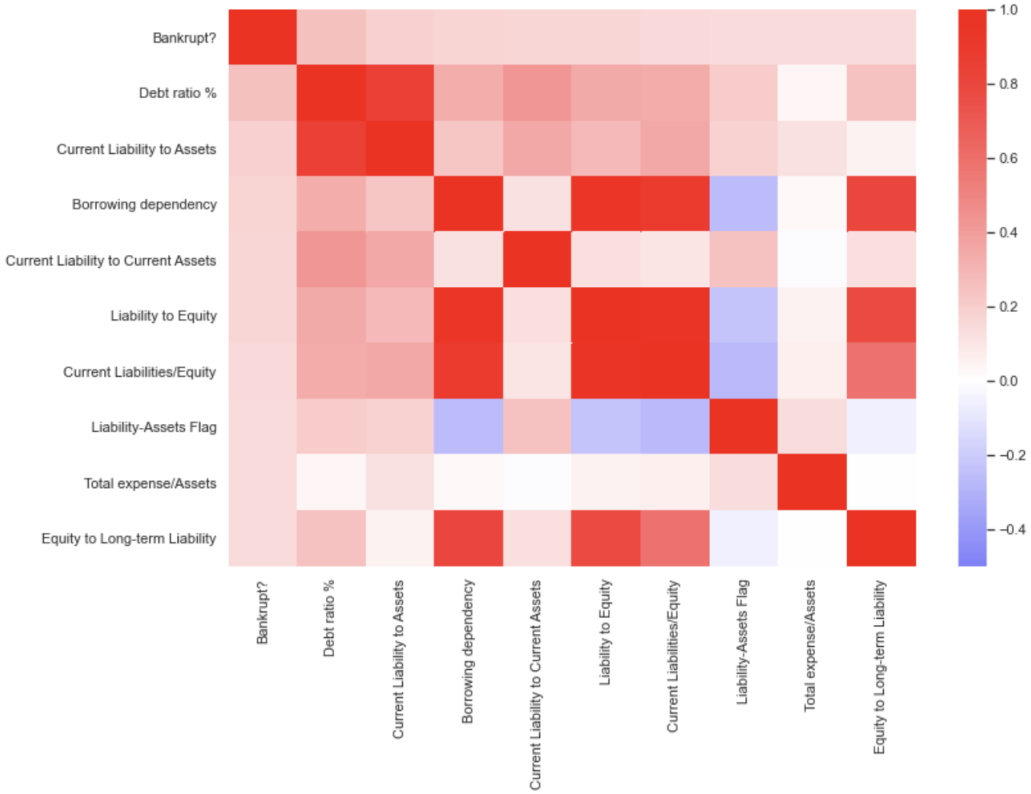


Figure 2: Correlation Heatmap for 9 Highest Correlated Features with Bankruptcy



The different histograms the distribution of data for each of the 9 columns are presented below.

Figure 3: Debt Ratio %

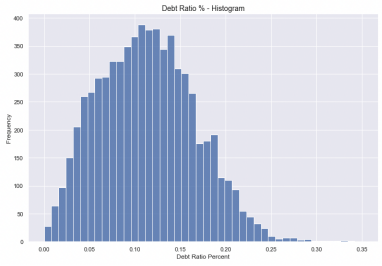


Figure 4: Current Liabilities/Equity

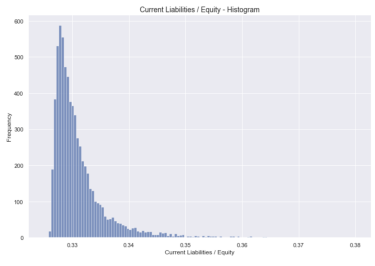


Figure 5: Borrowing Dependency

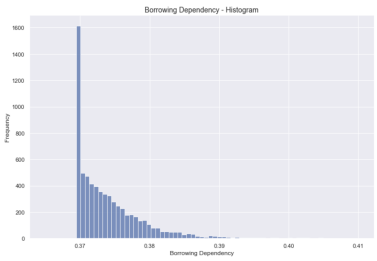


Figure 6: Current Liabilities to Current Assets

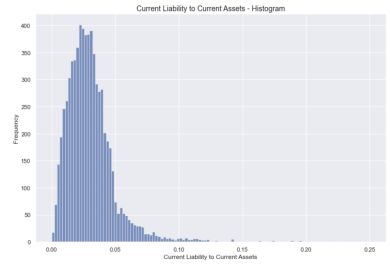


Figure 7: Liability to Equity

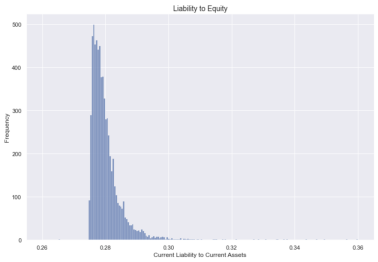


Figure 8: Current Liabilities/Equity

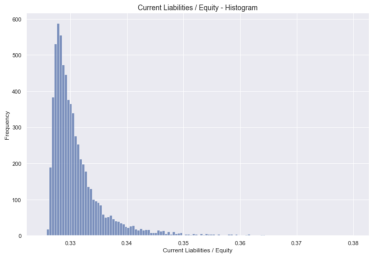


Figure 9: Liability-Assets Flag

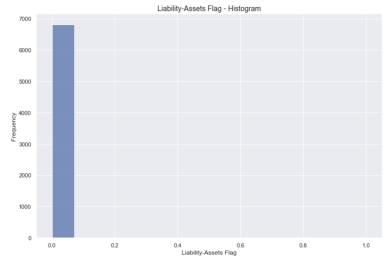


Figure 10: Total expense/Asset

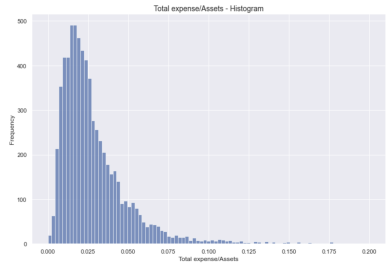
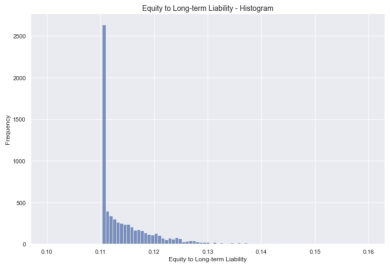


Figure 11: Equity to Long-term Liability



3 Model selection and validation

Avoiding over- and underfitting
In order to avoid both over- and underfitting, we need to select the right model, ie. the one that generalizes best to new data. This usually includes testing different models and choosing the optimal values for the model parameters such as a regularization value or number of trees in a tree ensemble model. The goal is to build a model that balances bias and variance in an optimal way where we control the complexity of the model while capturing the patterns in the data. One way to do so is by using a cross-validation, which allows us to test different models and use all the data. The data is split in training and validation sets. The different models we wish to test are then fitted to the training sets and tested on the last part to get an independent performance estimate. For each split we train and test a model for the grid of parameters we wish to optimize over. The advantage of using a cross-validation compared to a simple hold-out test set, is that we are able to use all the data to both test and train our models, while avoiding biasing the performance estimates. After choosing a model, we can get an estimate of the error by testing the model on a completely separate test set.

Performance Metrics
An important part of selecting and validating a classification model as the one in our case is to select one or more

performance metrics. As mentioned in the last section, our data set is imbalanced and only 3.2% of the outcome variable belong to the positive class, meaning only 3.2% of the companies in the data set ended up bankrupt. As a result, this makes Accuracy a very poor performance metric, as a very high score could be obtained with a very poor model. Instead, we will look into using an ROC curve and the F1-measure, which takes the Precision and Recall into account. An additional good and unbiased measure of our performance is just to look at a confusion matrix, which we easily made in our 2-class example.

Based on a number of initial experiments of different models, it was decided to pursue two approaches. The first one is a Logistic Regression model with regularization and the second is an Automated Machine Learning ensemble model made using the H2O.ai framework. Other models that were briefly tested and found not to be performing too well was a Support Vector Machine and GradientBoosting. These models are also tested as part of the H2O AutoML in addition to a number of other models.

3.1 Logistic Regression Model

A logistic regression model is a simple way of predicting the log-odds of a given classification using a linear model. Logistic regression has several advantages that are well suited for our specific problem. Logistic regression provides an intuitive probabilistic interpretation of the coefficients that would allow the model users to easily understand the effects of the features on predictions. Secondly, a logistic regression allows for adjustments using regularization and class weights. In our case we have a large number of features and an imbalance in the number of observations in each outcome class. Adjusting class weights will allow the final model to make a trade-off between false positives and false negatives depending on the 'costs' of making false classifications. Based on the exploratory data analysis, we saw that several features had a correlation with the response variable close to zero. Several of the features also had a significant intercorrelation. It was therefore decided to fit a Logistic Regression model with an elastic net penalty, which includes both an L1 and L2 penalty. Ideally, this gives some sparsity in the features while shrinking highly correlated features.

The figures below show the result of a 5-fold cross-validation, where the hyper-parameters of the regularization strength and L1-ratio are cross-validated. A total of 100 models were tested using an L1 ratio of 0.1, 0.2, ..., 1 and regularization strengths as shown in the figure legend. In this case, a smaller lambda value means a stronger regularization due to the inverse hyper-parameter in scikit-learn.

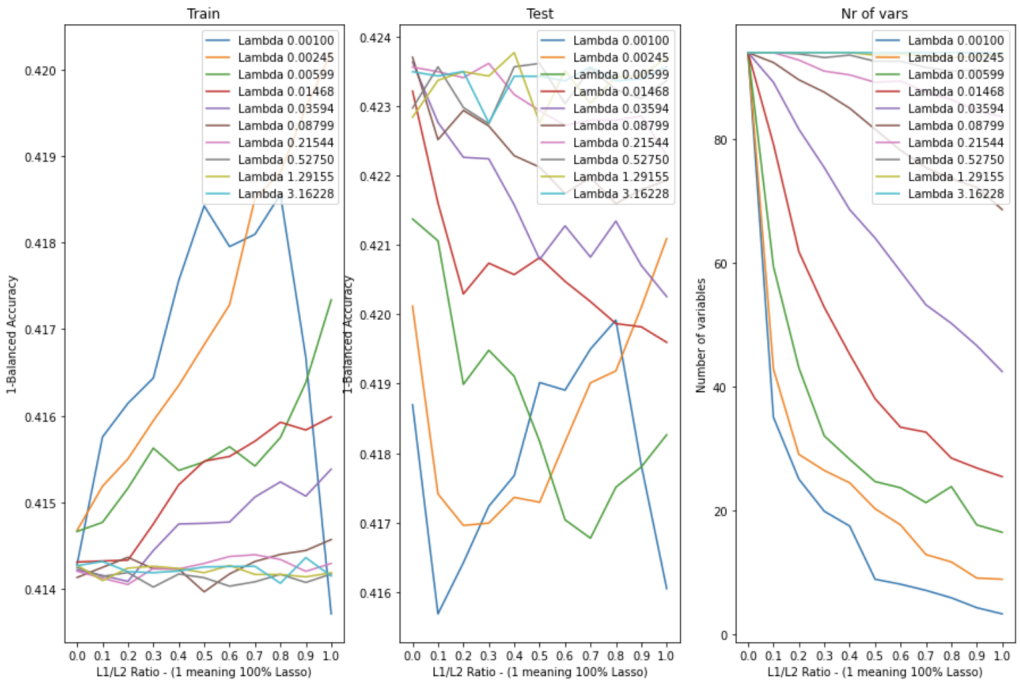


Figure 12: Cross-Validation results for Logistic Regression

Training errors are in general lowest for the most complex models with a small regularization. On the test set, we see that the models with the strongest regularization (blue, orange and green curve) perform best, although the actual difference in the error is relatively small. The test error for these models does not seem to be highly affected by the L1-ratio. The ratio does however determine the number of parameters in the model, as a stronger L1-penalty will drastically reduce the number of variables in the model from 95 to 10 or less for certain models. Based on general principles like the law of parsimony, we often want to choose the least complex model if several models have similar performance. Based on the cross-validation, the optimal hyper-parameters was therefore found to be a penalization of 0.001 with an L1-ratio of 1. This model was among the best-performing while being the least complex.

3.2 Automated Model Selection using H2O AutoML

As an alternative to a regular approach, where several models are tested manually by cross-validation, one can use an automated workflow. In this project, we have also used an automated approach in order to compare the performance of a broad range of automatically generated ensemble models to the logistic regression found in the previous section. To do so, we have used the H2O AutoML package, which is a combined algorithm and hyper-parameter search (CASH) program. This means that the package chooses both an estimator and search for optimal hyper-parameters. Often this results in an ensemble model where a number of different models are combined to make a prediction. H2O includes models such as Generalized Linear Model (GLM), Gradient Boosted Decision Trees, Distributed Random Forest and Support Vector Machines (SVM).

To fit the model, the data is split into a training and a test set using the same random seed as in the Logistic Regression to compare the models. The H2O-algorithm splits the training set into a training and validation test and performs a cross-validation to test different models before recommending a model with the best performance. Below in Table 1 is shown the best performing models in the cross-validation. The top model is a stacked ensemble model. "AllModels" means that the ensemble contains a number of different models, while "BestOfFamily" indicates that a model is build on individual models from the same "family", eg. decision trees.

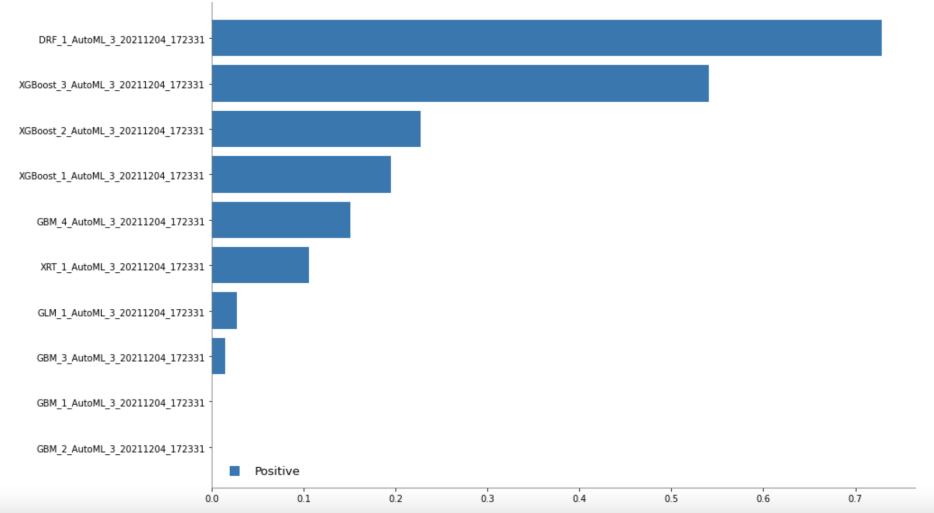
The AutoML uses AUC, or the Area Under the Curve, as a measure of performance. However, it should be noted that this measure also has its drawbacks, which in our case may be particularly important. For instance, AUC ignores the predicted probability values. Instead, it summarizes the test performance over regions of the ROC space in which one would rarely operate because certain cut-off probabilities are simply not interesting (Jorge M. Lobo et al., 2007). The AUC looks at the separation of classes and therefore puts an equal weight on each class. As previously mentioned, this is probably not ideal in our case, as we might want to put more weight on identifying the bankruptcies.

Table 1: AutoML models ranked according to performance

Rank	Model id	AUC	logloss	AUCPR	Mean per class error
1	StackedEnsemble_AllModels_5	0.944	0.0849	0.4451	0.2704
2	StackedEnsemble_BestOfFamily_3	0.942	0.0854	0.4446	0.2780
3	StackedEnsemble_AllModels_2	0.941	0.0855	0.4448	0.2827
4	StackedEnsemble_AllModels_1	0.941	0.0858	0.4346	0.2486
5	StackedEnsemble_BestOfFamily_6	0.941	0.0861	0.4472	0.2757
6	XGBoost_3	0.939	0.0955	0.4412	0.2867

To better understand the model that H20 has found, we can look at the individual models in the ensemble. The figure below shows the weight of the individual models in the ensemble.

Figure 13: Model weight in leading AutoML model



What we see in the figure above is that the ensemble model is mainly build on decision trees. DRF refers to a Distributed Random Forest, while XGBoost and GBM refers to two different gradient boosting algorithm. These are the dominating models in the ensemble. The results of the AutoML model on the test set will be discussed in the next section.

4 Results

After performing the cross-validation to identify the best hyper-parameters, the final model was trained and it's performance was evaluated on the separate test set. This was done for the Logistic regression and the AutoML-model. The final logistic regression model contained only 5 features. The coefficients are shown below.

Table 2: Logistic regression coefficients (L1-penalty, C=0.001)

Feature	Return on Assets (ROA)	Debt ratio %	Net worth/Assets	Working Capital to Total Assets	Net Income to Total Assets
Coefficient Value	-0.29	0.29	-0.29	-0.07	-0.29
exp(Coefficient Value)	0.748264	1.336427	0.748264	0.932394	0.748264

As seen in the table, a higher Debt Ratio % increases the odds of bankruptcy by a multiplicative factor of 1.34. This is the only feature in the model with a positive correlation with the outcome variable. The other coefficients indicate that if a company has a high ROA, Net Worth/asset ratio etc., the company is less likely to go bankrupt.

Comparing the Logistic regression to the AutoML-model confirms how the Logistic Regression provides us with more flexibility to make the trade-off between false positives and false negatives. The logistic model shown in the two most left figures does a decent job of identifying the companies that went bankrupt. Out of 38 companies in the test data, 21 and 36 are identified correctly for the two class weights respectively. However using a higher class weight also comes at the cost of having many false positives. The AutoML-model, which is mainly built on an ensemble of boosted trees does not seem to perform as well. Despite the emphasis on wrongly classified observations in the sequential gradient boosting algorithm, the model doesn't seem to be able to identify the majority of the bankrupt companies and the implementation of the AutoML does not leave us with an easy option to adjust the class weights (although it might in theory be possible).

Figure 14: LR Class Weight 1:15

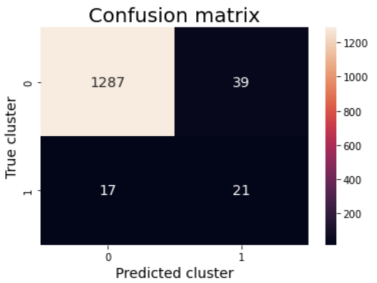


Figure 15: LR Class Weight 1:25

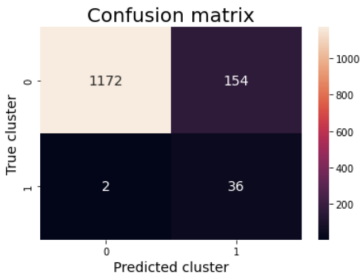
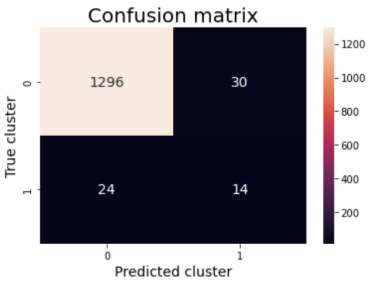
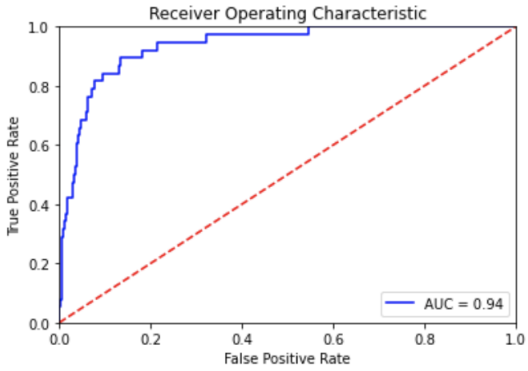


Figure 16: AutoML Model



Looking at the ROC curve below for the Logistic regression model with class weights of 1:25 confirms that the model is actually doing quite well in seperating the classes. When considering the 154 false positives it is important to remember that our of a total of 1326 true negatives, the false positive rate is still 11.6%.

Figure 17: ROC Curve for Logistic Regression with Class Weights 1:25



The exact class weight and classification threshold will ultimately depend on the use of the model. As a financial company, you may be willing to deny financial credit services to a relatively number of customers in order to avoid any clients defaulting on the loans. Based on the expected profits and costs of providing and loan or having a customer default on a loan, the model users would have the opportunity to adjust the model.

5 Limitations and Fairness

An important part of fairness is to consider any potential protected attributes (ie. features on which discrimination is prohibited). None of the data that our bankruptcy data set contained included any sort of protected attributes. All of the data (other than the Boolean 'Bankrupt?' column) were numerical values or ratios between specific financial and managerial accounts that are assumed to be publicly available. There was no data included that identified the properties of the staff, such as gender and race. Location information about each company was also omitted (we were given no coordinates or addresses). As a result, we can confidently conclude that our model is fair with respect to any potential protected attributes.

However, this is not to say that our model is completely fair overall. In the end, we were still facing some false positives and false negatives. If banks or other loan-givers use our model, there is a risk of misidentifying if a specific company might go bankrupt. With a false negative for the predicted bankruptcy, banks might be inclined to invest in the company and eventually lose their investment. With a false positive, banks would deny a company from receiving a loan. This could prove to potentially be disastrous for the company and increase the risk of bankruptcy that would otherwise not exist. If the company was in dire need of the loan to stay in business, a false positive could

end up drastically damaging the company – if not causing bankruptcy, then it could still suffer losses in sales, laying off of staff, and more due to the lack of funds.

6 Conclusion and Future Work

There are many financial aspects that play key roles in a business's potential to go bankrupt. Through our work, we were able to identify the key features that had the highest impact on whether or not the business went bankrupt, and understand the complexities in their relationships with each other. We split the data into training and validation sets and in the machine learning models, we used cross-validation to get estimates of our errors. We found the optimal hyper-parameters to be a penalization of 0.001 with L1-ratio of 1. In a separate model using H2O AutoML, we determined that a higher Debt Ratio % value increases the probability of bankruptcy by a factor of 1.34, and our ensemble relied heavily on building decision trees (DRF). Again, it's important to note we did have a fairly high 11.6% rate of false positives.

Though our model works well in identifying the potential for bankruptcy, more work must be done to limit the incorrect identifications.