Using Classification Learning to Predict Bankruptcy

Davis Gilmore and Walker Browne

I. Introduction

When deciding what dataset to choose for our project, our group came across a dataset made up of financial data from the Taiwan Stock Exchange intended for the creation of a bankruptcy prediction model. Among the other datasets we were considering, we deemed this to have the most significant real-world application, as being able to notify a company of impending bankruptcy could be invaluable to their financial well-being. Bankruptcy in this instance is defined by the Taiwan Stock Exchange as a company's inability to pay outstanding debts due to illiquidity. Our overall goal was to use a variety of classification algorithms to achieve a substantial increase in accuracy from the ZeroR accuracy of 49.9%, as well as identify key attributes in predicting bankruptcy.

After examining the attributes in the dataset, we hypothesized that the attributes involving debt and liabilities (quickRatio, liabilityAssetsFlag, etc) would be among the most predictive in the dataset. We also believed it would be likely that bankrupt companies would tend to have higher values for attributes involving debt and potentially lower values for attributes involving profit and net worth. Our findings and analysis involve experimentation with PART, IBK, Naive Bayes, and J48.

II. Dataset

We obtained our dataset from Kaggle (https://www.kaggle.com/fedesoriano/company-bankruptcy-prediction). Our dataset consists of 96 attributes and 6819 instances. The set is made up of financial data gathered from the Taiwan Economic Journal from the years 1999-2009. When examining the attributes, we found that 93 were numeric and 3 were nominal, with a majority of the numeric attributes represented as ratios. Some of the key attributes we examined were inventoryOverWorkingCapital, liabilityAssetsFlag, and totalLiabilityEquityRatio.

Attribute	Description	Attribute Type
1. ROACbeforeInterest	Return on assets before interest and depreciation	numeric
2. ROAAbeforeInterest	Return on assets before interest and after tax	numeric

3. ROABbeforeInterest	Return on assets before interest and depreciation, after tax	numeric
4. operatingGrossMargin	Gross Profit/Net Sales	numeric
5. realizedSalesGrossMargin	Realized Gross Profit/Net Sales	numeric
6. operatingProfitRate	Operating Income/Net Sales	numeric
7. preTaxNetInterestRate	Pre-Tax Income/Net Sales	numeric
8. afterTaxNetInterestRate	Net Income/Net Sales	numeric
9. netNonOperatingIncomeRatio	Non-industry income and expenditure/revenue	numeric
10. continuousInterestRate	Net Loss/Net Sales	numeric
11. operatingExpenseRate	Operating Expenses/Net Sales	numeric
12. researchAndDevExpenseRate	R&D Expenses/Net Sales	numeric
13. cashFlowRate	Operating Cash Flow/Current Liabilities	numeric
14. interestBearingDebtInterestRate	Interest-bearing Debt/Equity	numeric
15. taxRate	Effective Tax Rate at time recorded	numeric
16. netValuePerShareB	Book Value Per Share (B)	numeric
17. netValuePerShareA	Book Value Per Share (A)	numeric
18. netValuePerShareC	Book Value Per Share (C)	numeric
19. persistentEPS	Earnings per-Share - Net Income	numeric
20. cashFlowPerShare	Cash Flow Allocated Per Share	numeric
21. revenuePerShare	Sales per Share (Measured in	numeric

	Yuan)	
22. operatingProfitPerShare	Operating Income Per Share (Measured in Yuan)	numeric
23. preTaxPerShareNetProfit	Pretax Income Per Share (Yuan)	numeric
24. realizedSalesGrossProfitGrowthRate	Realized Gross Profit From Sales	numeric
25. operatingProfitGrowthRate	Operating Income Growth	numeric
26. afterTaxNetProfitGrowthRate	Net Income Growth	numeric
27. regularNetProfitGrowthRate	Continuing Operating Income after Tax-Growth	numeric
28. continuousNetProfitGrowthRate	Net Income excluding Disposal Gain or Loss Growth	numeric
29. totalAssetGrowthRate	Total Growth on Assets	numeric
30. netValueGrowthRate	Total Equity Growth	numeric
31. totalAssetReturnGrowthRateRatio	Return On Total Asset Growth	numeric
32. cashReinvestmentRatio	Cash Reinvestment Ratio	numeric
33. currentRatio	Current Ratio: Ability to pay liabilities within 1 Year	numeric
34. quickRatio	Ability to pay liabilities within Short-Term	numeric
35. interestExpenseRatio	Interest Expenses/Total Revenue	numeric
36. totalLiabilityEquityRatio	Total Debt/Total Net Worth	numeric
37. debtRatio	Liability/Total Assets	numeric
38. netWorthbyAssets	Equity/Total Assets	numeric

39. longTermFundSuitability	(Long-term Liability+Equity)/Fixed Assets	numeric
40. borrowingDependency	The cost of interest-bearing debt	numeric
41. contingentLiabilitiesOverNetWorth	Contingent Liability/Equity	numeric
42. operatingProfit	Operating Income/Capital	numeric
43. netProfitPreTax	Pretax Income/Capital	numeric
44. inventoryAndAccountsReceivable	(Inventory+Accounts Receivable)/Equity	numeric
45. totalAssetTurnover	Rate of Asset Turnover within 1 year	numeric
46. accountsReceivableTurnover	Rate of Account Receivable Turnover within 1 year	numeric
47. averageCollectionDays	Days Receivable Outstanding	numeric
48. inventoryTurnoverRate	Rate of Turnover for Inventory within 1 year	numeric
49. fixedAssetsTurnover	Frequency of turnover for Fixed Assets	numeric
50. netWorthTurnoverRate	Equity Turnover within 1 year	numeric
51. revenuePerPerson	Sales per Employee	numeric
52. operatingProfitPerPerson	Operation Income Per Employee	numeric
53. allocationRatePerPerson	Fixed Assets Per Employee	numeric
54. workingCapitalToTotalAssets	Working Capital/Total Assets	numeric
55. quickAssetsOverTotalAssets	Quick Assets/Total Assets	numeric

56. currentAssetsOverTotalAssets	Current Assets/Total Assets	numeric
57. cashOverCurrentLiability	Cash/Current Liabilities	numeric
58. quickAssetsOverCurrentLiability	Quick Assets/Current Liabilities	numeric
59. cashOverCurrentLiability	Capital/Current Liabilities	numeric
60. currentLiabilityToAssets	Current Liabilities/Total Assets	numeric
61. operatingFundsToLiability	Operating Funds/Liability	numeric
62. inventoryOverWorkingCapital	Inventory/Working Capital	numeric
63. inventoryOverCurrentLiability	Inventory/Current Liability	numeric
64. currentLiabilitiesOverLiability	Current Liabilities/Total Liabilities	numeric
65. workingCapitalOverEquity	Working Capital/Equity	numeric
66. currentLiabilitiesOverEquity	Current Liabilities/Equity	numeric
67. longTermLiabilityToCurrentAssets	Long-term Liability/Current Assets	numeric
68. retainedEarningsToTotalAssets	Retained Earnings/Total Assets	numeric
69. totalIncomeOverTotalExpense	Total Income/Total Expense	numeric
70. totalExepenseOverAssets	Total Expense/Assets	numeric
71. currentAssetTurnoverRate	Current Assets/Sales	numeric
72. quickAssetTurnoverR	Quick Assets/Sales	numeric

ate		
73. workingCapitalTurnoverRate	Working Capital/Sales	numeric
74. cashTurnoverRate	Cash/Sales	numeric
75. cashFlowToSales	Cash Flow/Sales	numeric
76. fixedAssetstoAssets	Fixed Assets/Total Assets	numeric
77. currentLiabilityToLiability	Current Liabilities/Total Liabilities	numeric
78. currentLiabilityToEquity	Current Liabilities/Equity	numeric
79. equityToLongTermLiability	Equity/Long Term Liability	numeric
80. cashFlowToTotalAssets	Cash Flow/Total Assets	numeric
81. cashFlowToLiability	Cash Flow/Total Liabilities	numeric
82. cfoToAssets	Cash Flow/Total Assets	numeric
83. cashFlowToEquity	Cash Flow/Equity	numeric
84. currentLiabilitytoCurrentAssets	Current Liabilities/Current Assets	numeric
85. liabilityAssetsFlag	If total liabilities are greater than total assets or not	Nominal (1: Total liability exceeds total assets 0: Total assets exceeds total liabilities)
86. netIncometoTotalAssets	Net Income/Total Assets	numeric
87. totalAssetstoGNP	Total Assets/Gross Net Profit	numeric
88. noCreditInterval	How many days can a company operate without having to open lines of credit: Runway	numeric
89. grossProfitToSales	Gross Profit/Sales	numeric

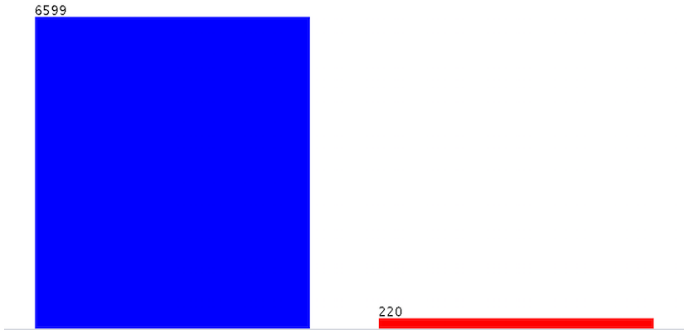
90. netIncomeToStockHoldersEquity	Net Income/Stockholders Equity	numeric
91. liabilityToEquity	Total Liability/Equity	numeric
92. degreeOfFinancialLeverage	Earnings Per Share/Operating Income	numeric
93. interestCoverageRatio	Interest Expense/Earnings before Interest and Taxes	numeric
94. netIncomeFlag	Is Net Income negative for the last two years?	Nominal (1: if net income is negative for the last two years. 0: otherwise)
95. equityToLiability	Equity/Total Liabilities	numeric
96. bankrupt	Has the company gone bankrupt?	Nominal (1: Bankrupt, 0: Not Bankrupt)

III. Data Preparation

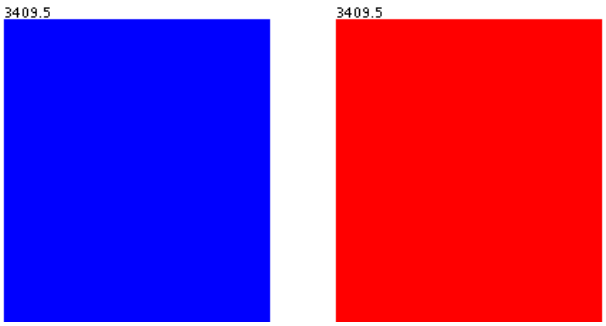
1. Loading Into Weka: We ran into some trouble once we had downloaded the .zip file containing the dataset and attempted to open it in VsCode or Excel. The .zip file contained two .csv files, one with the actual dataset and the other with a giant block of encrypted characters. What was interesting was, when we attempted to open the normal dataset, a similar large block of encrypted text appeared. After several rounds of trial and error, we decided to use an online tool to unzip the files for us to see if this would get rid of the issue. Once we downloaded the unzipped .csv files and opened the dataset in Excel, the data appeared perfectly. We manually changed the 1st attribute listed "bankrupt" to the class attribute, converted the file into .arff format, and successfully opened the dataset in Weka after several "index out of range" errors.

2. Unbalanced Data: Once we had converted our dataset to .arff format, we ran ZeroR to find a baseline of accuracy for our model. ZeroR produced 96.7737% accuracy, which immediately raised the flag that we had a severely imbalanced dataset. The class attribute values were imbalanced with a ratio of 6599:220, not bankrupt to bankrupt. In order to balance our data, we decided to use the filter "ClassBalancer" which weighs the class attribute values in the dataset to a 1:1 ratio. Upon applying this filter, we ran ZeroR again and received a more realistic baseline accuracy of 49.9932%.

Class attribute weights before applying ClassBalancer (0, 1):



Class attribute weights after applying ClassBalancer (0, 1):



IV. Data Analysis

1. **PART:** This is a classification algorithm that is similar to PRISM, using a separate and conquer strategy, which is used when dealing with one class. In these types of algorithms, it works to first identify a rule. Then, the algorithm separates out all of the instances it covers. Finally, it will 'conquer' the remaining instances. Overall, this algorithm works to build a partial decision tree in each iteration and makes the "best" leaf into a rule. When this algorithm is tested against ⅔ percentage split, it produces 38 rules. Percentage split is a testing method which in this case, takes 66% of the data, and tests that against the other 34% of the data. I chose to first use the ⅔ percentage split testing method because this method is the ideal strategy for testing the accuracy of a model when a large, diverse dataset is available. Just to compare results, I tried testing the data set against ten fold cross validation even though ten fold cross validation is typically used on smaller data sets. Ten fold cross validation is a testing strategy where it breaks the data into sets of ten, then takes nine of the sets to create a training set, or model, then uses the tenth set to test the model, then tests the model nine more times, which will you give you an accuracy rate for each test, and then averages the percentages. However, when I tested the data using the ⅔ percentage split, I found that the ten fold cross validation actually had more

correctly identified instances when the ClassBalancer filter was implemented. When the ClassBalancer filter was removed, the ⅔ percentage split had a slightly higher accuracy rate. Overall, the rules generated contained rules with one attribute, four attributes, nine attributes, twelve attributes and more. The rules also predicted companies to either go bankrupt, or stay financially successful.

2. **IBK**: IBK or the Nearest Neighbor Algorithm, is a classification algorithm of instance based learning, which is often referred to as lazy learning because no work is necessary until a classification is needed. In the nearest neighbor algorithm, it looks for the k closest instances, based on the new instance and uses the k closest instances in order to predict the class, which in this case would be bankruptcy or not bankruptcy. When comparing the new instances to the existing instances, the algorithm implements a distance metric, commonly the Euclidean Distance function. The limitations of the Euclidean Distance function is that it assumes the attributes have been normalized and are of equal importance. When running this algorithm, you can adjust the k value to whatever you would like. When k = 1, the algorithm would look for the single closest instance to the new one. When k = 3, the algorithm would look for the three closest instances to the new one, and based on what has the majority of the class attributes of those three instances, it will predict that the class attribute will be the same for the new instance. As k increases, accuracy will increase up until a certain point, and then accuracy will start to decrease with each addition to k.

3. **Naive Bayes:** Naive Bayes is another example of a classification algorithm, in which each attribute is assumed to be independent of each other. The Naive Bayes algorithm uses the Bayes Theorem which calculates the probability of the class attribute being predicted, given the value of a specific instance. When you run the Naive Bayes algorithm on a dataset in Weka, the algorithm produces a confusion matrix that represents the correctly and incorrectly classified instances, an overall accuracy score based on those classifications, as well as the mean and standard deviation for each attribute. We initially ran Naive Bayes on our unbalanced dataset with a 80% split, meaning 80% of the data was used to train the model, while the remaining 20 was used to test. Next we applied the “ClassBalancer” filter in the preprocessing tab and ran Naive Bayes again with a 80% split.

4. **J48**: J48 is a classification algorithm in Weka that builds a decision tree that creates rules based on the attributes within the dataset. J48 produces a summary of the correctly and incorrectly classified instances, a list of the rules produced and the amount of instances classified by that rule, as well as a visualizable tree that represents the rules in a more visually digestible way. Because of our extensive list of 96 attributes, we figured that some of the less predictive and redundant attributes would add unwanted noise to the

tree. Therefore, we decided to remove attributes such as operatingProfitRate, regularNetProfitGrowthRate, totalAssetReturnGrowthRateRatio which we deemed non predictive due to the nearly identical mean and standard deviations for these attributes between the bankrupt and non bankrupt instances. We made this decision by examining the mean and standard deviation for each attribute in the Naive Bayes classifier output box. We chose to use a 80% split, meaning 80% of the data was used to train the model, and the remaining 20% was used to test. We chose an 80% split for this dataset, as we've learned in the course that a larger percentage of training data is preferable when working with a large dataset such as ours.

IV: Results

- **PART**
    - PART without ClassBalancer tested on ten fold cross validation generated 27 rules with an accuracy of **96.1725% with 6558 correctly identified instances**
    - PART with ClassBalancer tested on ten fold cross validation generated 38 rules and had an accuracy of **69.4089% with 4732.9904 correctly classified instances**
    - PART without ClassBalancer tested on ⅔ percentage split generated 27 rules with an accuracy of **96.6782% with 2241 correctly identified instances**
    - PART with ClassBalancer tested on ⅔ percentage split generated 38 rules with an accuracy of **66.0328% with 1483.3032 correctly identified instances**
        - Rules Generated by Part (with ClassBalancer tested on ⅔ percentage split):
            - totalIncomeOverTotalExpense > 0.002828: 0 (44.95)
                - This rule is saying that if a company's aggregate income divided by their aggregate expenditures is greater than 0.002828, then the company will not go bankrupt. The 44.95 is saying that this rule is covered by 44.95 instances.
            - currentRatio <= 0.013305 AND operatingProfitGrowthRate <= 0.850845 AND totalExpenseOverAssets > 0.041434 AND currentLiabilityToAssets <= 0.201594: 0 (39.27)
                - This rule is saying that if the current Ratio, which is the ability to pay off liabilities within a year is less than or equal 0.013305 and the rate of operating financial gain is less than or equal to 0.850845 and total expenditures divided by assets is greater than 0.041434 and the ratio of current liabilities to total assets is less than or equal to 0.201594, then the company will not be bankrupt. This rule was covered by 39.27 instances.
            - interestCoverageRatio <= 0.563998: 1 (32.03/1.03)

        - The Interest Coverage Ratio is the earnings before interest and taxes divided by the total interest expenditures and if it is less than or equal to 0.563998, then the company is predicted to go bankrupt.

- **IBK**
    - IBK with ClassBalancer and k =1, had an accuracy of **61.0655% with 1371.7215 correctly identified instances.**
    - IBK with ClassBalancer and k=3, had an accuracy of **66.1246% with 1485.3652 correctly identified instances**
    - IBK with ClassBalancer and k=5, had an accuracy of **70.9768 with 1594.3613 correctly identified instances.**
    - Unlike many datasets we have looked at in class before where we have been running IBK and testing the various k values, the accuracy rate does not reach its peak and then starts its decline until k equals approximately 25.
- **Naive Bayes**
        - Naive Bayes without ClassBalancer had an accuracy of **71.6566% with 1661 correctly classified instances.**
        - Naive Bayes with ClassBalancer applied had an accuracy of **76.2869% with 1713.64 correctly classified instances.**
    - We were intrigued by the mean and standard deviations produced by Naive Bayes for each numeric attribute. We found that these values gave an interesting insight to some of the more predictive attributes in the dataset. We hypothesized that attributes involving debt and liabilities would likely be much higher for instances classified as bankrupt, while inversely some of the attributes involving profitability and total assets would be higher for instances classified as not bankrupt.
        - Quick Ratio: A company's ability to meet its short-term debts/liabilities by utilizing their most liquid assets. A high quick ratio indicates better liquidity and overall financial health, while a low ratio indicates the opposite. The instances classified by Naive Bayes as bankrupt had a mean quick ratio of roughly 4.1 million, while those classified as not bankrupt had a mean quick ratio of about 7.2 million. We inferred that this piece of knowledge supported our hypothesis that companies with a higher quick ratio have better financial health, and are therefore less likely to go bankrupt.
        - Liability Assets Flag: The nominal attribute "liabilityAssetsFlag" is an indicator for if a company's total liabilities exceed their total assets, which we hypothesized would be extremely predictive of bankruptcy. For nominal attributes, Naive Bayes produces a confusion matrix-esque diagram that displays the amount of instances classified for each nominal

value (0 or 1). We found that out of the 3317 instances classified as bankrupt, 94 raised the liability assets flag (94/3317 = .028); while out of the 3409 instances classified as not bankrupt, only 2 raised the liability assets flag (2/3409 = .0006.) We concluded that these findings supported our hypothesis due to the high predictiveness of this attribute, since out of the 96 liability asset flags raised, 94 were bankrupt companies (94/96 = .979/97.9%)

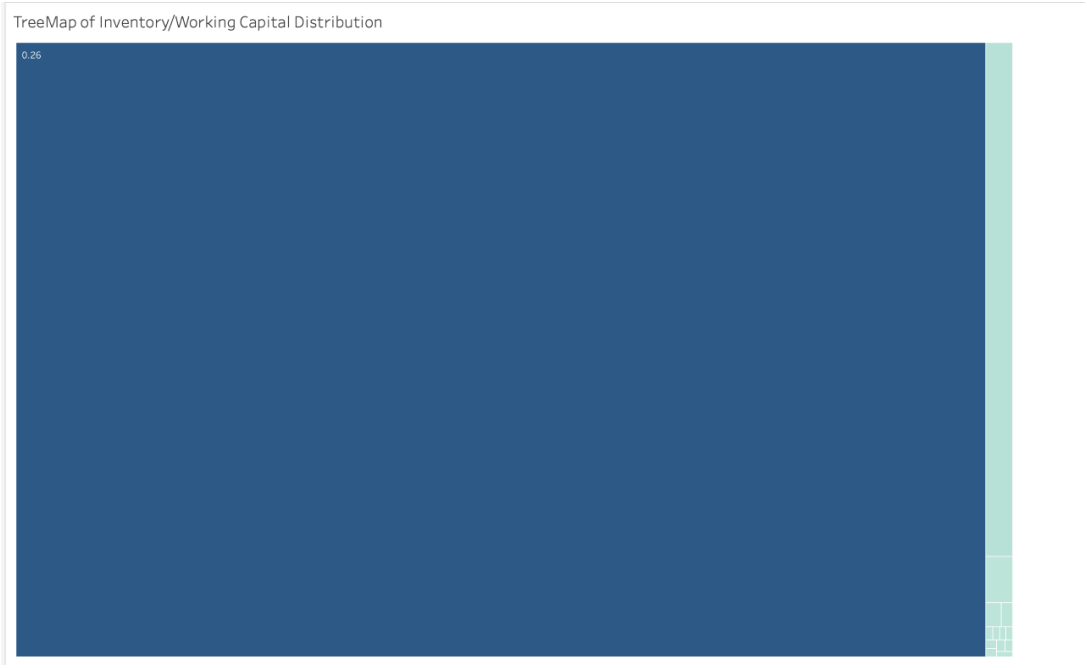
- Cash Over Total Assets: The numeric attribute "cashOverTotalAssets" is the ratio of a company's cash to total assets, a general indicator of high or low liquidity. We hypothesized that companies with a large amount of their cash tied up in long term assets (lower ratio), with a lower amount of cash on hand, were much more likely to be bankrupt than companies with the inverse ratio (higher ratio). The instances classified by Naive Bayes as bankrupt had a mean ratio of .0477 while those classified as not bankrupt had a mean ratio of .1266; nearly 3 times higher than the opposing class. We concluded that these findings supported our initial hypothesis that a lower cash over total assets ratio would be a trustworthy bankruptcy indicator.

- **J48**
  - Running J48 with the noisy attributes included in the dataset rendered an accuracy of **70.6552% with 111 leaves**
  - Running J48 with the noisy attributes removed from the dataset rendered an accuracy of **69.6664% with 93 leaves**.
  - While this didn't increase the accuracy of the model, as we had hoped, it reduced the amount of leaves from 111 to 93, making the tree slightly easier to read. Additionally this removed some of the rules that would classify an instance

  Rules Produced by J48 (Noisy attributes removed):
  - If inventoryOverWorkingCapital > .276357 THEN Bankrupt = 1 (266.21)
    - This rule indicates that if a company has a ratio of inventory to working capital greater than .276357, they are bankrupt. This was our best performing rule with 266 instances classified correctly. This rule basically means that if a company has a greater amount of capital tied up in inventory than being actively put to use, the company will have a higher chance of being bankrupt.
    - The visualization below represents the value distribution of instances for the attribute Inventory Over Working Capital. The largest portion of the TreeMap represents a bin of 5654 instances in which the average inventory to working capital ratio was .26. We inferred that this bin likely contains a large portion of the non-bankrupt companies, as The next largest bin is represented by the light blue rectangle that stretches about

80% of the length of the largest bin. This bin represents 328 instances with the average inventory over working capital ratio of .278. This portion of instances is applicable to the rule formed by J48 and likely contains the majority of instances classified as bankrupt by this rule.

-



- If borrowingDependency > .379905 THEN Bankrupt = 1 (98.37)
    - Borrowing dependency is a measure of how dependent a company is on credit, usually indicating low amounts of cash on hand. This rule states that if a company has a borrowing dependency of over .379905, then they are bankrupt. This rule correctly classified 98 instances, making it one of the higher performing rules produced by J48.
        - The visualization below is a tree map that represents the distribution of values for the borrowing dependency attribute. The largest section of the graph represents a bin of 5382 instances with an average borrowing dependency of .36. We inferred that this is likely made up mostly of non-bankrupt companies, those that rely less on borrowing or opening lines of credit to function. The next largest section represents a bin of about 514 instances with an average borrowing dependency of .384. Since these values conveniently fall above the minimum value for the rule produced by J48, we inferred that the majority of the 98 instances classified as bankrupt were contained by this section.

TreeMap of Borrowing Dependency Distribution

- If totalLiabilityEquityRatio > .01093 AND quickRatio <= .00494 THEN Bankrupt = 1 (48)
    - Liability Equity Ratio refers to the ratio of money owed over the amount of money invested by shareholders. Quick ratio is company's ability to meet its short-term debts/liabilities by utilizing their most liquid assets. A high quick ratio indicates better liquidity and overall financial health, while a low ratio indicates the opposite. This rule indicates that if a company has a LE ratio greater than .01093 and a quick ratio greater than or equal to .00494, the company is bankrupt. This rule classified 48 instances correctly, making it a slightly above average rule produced for our dataset. We thought this was an interesting rule because it reinforced our findings regarding quickRatio when running Naive Bayes.

V. Conclusion

We originally hypothesized that attributes involving debt and liabilities would be among the most predictive attributes of bankruptcy. Upon our analysis, we found this to be true for debt-related attributes such as borrowing dependency and liability equity ratio, in which higher values appeared to be quite predictive of bankruptcy. Additionally, we found that instances classified as not-bankrupt had much higher mean values for attributes involving net worth and free cash flow, such as total income over total expense and cash over total assets. With ClassBalancer applied, the highest accuracy we were able to achieve was Naive Bayes with 76.289% and 1713.64 instances classified correctly.

Some real world takeaways from this project for any sprouting entrepreneurs or those interested in personal/corporate finance would be to manage your debts and liabilities with the utmost caution, don't take out lines of credit that you're not sure you'll be able to pay back. Risk can be a good thing and extremely beneficial for a company in some cases, however it must be calculated.