## The University of Texas at Dallas



Assignment on

**TAIWANESE BANKRUPTCY PREDICTION**

**--- Project Final Report**

*Course Title: Business Analytics with R*

*Course Code: BUAN 6356.004*

**Prepared for:**

**Zhe Zhang, PhD**

**Prepared by:**

Allen Daniel Baskaran – AXB220222

Venkat Bhargav Nisthala – BVN220000

Ankita Shinde – ASS220032

Aravind Velaga – AXV230044

Piper Wang – YXW230012

**Date: 4th December 2023**

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| TOPIC | PAGE NUMBER |
| INTRODUCTION | 3 |
| PRE-PROCESSING | 3 |
| ANALYSIS | 4 |
| MODELS AND DISCOVERIES | 6 |
| LOGISTIC REGRESSION ANALYSIS | 7 |
| DECISION TREE ANALYSIS | 18 |
| NEURAL NETWORK ANALYSIS | 23 |
| CONCLUSION | 28 |

1. **Introduction**

Bankruptcy or business failure can have a negative impact both on the enterprise itself and the global economy. Business practitioners, investors, governments, and academic researchers have long studied ways to identify the potential risk of business failure to reduce the economic loss caused by bankruptcy. In short, bankruptcy prediction is very important for many related financial institutions. In general, the aim is to predict the likelihood that a firm may go bankrupt. Financial institutions need effective prediction models to make appropriate lending decisions.

1. **Pre-processing:**

We first need to pre-process the data for the analysis phase

*#Loading the data*

banking.df <- read.csv("data.csv")

head(banking.df)

**Missing Values:**

#We can find the missing values using the below command

any(is.na(banking.df))

No value is missing in the dataframe

**Data partition for validation:**

set.seed(1, sample.kind = "Rounding")

validation\_index <- createDataPartition(y = banking.df$Bankrupt, times = 1, p = 0.2, list = FALSE)

validation <- banking.df[validation\_index,]

banking.df <- banking.df[-validation\_index,]

**Identifying & Removing Predictors:**

nzv <- nearZeroVar(df)

nzv[!nzv %**in**% 1] *#to avoid delete the variable of interest*

## Variable Format Changes:

## banking.df$Bankrupt <- factor(df$Bankrupt, labels = c("non\_bankruptcy", "bankruptcy"))

## validation$Bankrupt <- factor(validation$Bankrupt, labels = c("non\_bankruptcy", "bankruptcy"))

## Train and test sets:

To train the models, test, and optimize we are using 80% of the data for training the model and 20% for testing

train\_index <- createDataPartition(y = banking.df$Bankrupt, times = 1, p = 0.8, list = FALSE)

train\_set <- banking.df[train\_index,]

test\_set <- banking.df[-train\_index,]

**B. Analysis**

Simple analysis

summary(banking.df[,1:12])

Bankrupt. ROA.C..before.interest.and.depreciation.before.interest

Min. :0.000 Min. :0.024

1st Qu.:0.000 1st Qu.:0.476

Median :0.000 Median :0.503

Mean :0.032 Mean :0.506

3rd Qu.:0.000 3rd Qu.:0.536

Max. :1.000 Max. :1.000

ROA.A..before.interest.and...after.tax ROA.B..before.interest.and.depreciation.after.tax

Min. :0.000 Min. :0.034

1st Qu.:0.536 1st Qu.:0.527

Median :0.560 Median :0.552

Mean :0.559 Mean :0.554

3rd Qu.:0.590 3rd Qu.:0.585

Max. :0.985 Max. :0.958

Operating.Gross.Margin Realized.Sales.Gross.Margin Operating.Profit.Rate Pre.tax.net.Interest.Rate

Min. :0.000 Min. :0.000 Min. :0.000 Min. :0.000

1st Qu.:0.601 1st Qu.:0.601 1st Qu.:0.999 1st Qu.:0.797

Median :0.606 Median :0.606 Median :0.999 Median :0.797

Mean :0.608 Mean :0.608 Mean :0.999 Mean :0.797

3rd Qu.:0.614 3rd Qu.:0.614 3rd Qu.:0.999 3rd Qu.:0.798

Max. :1.000 Max. :1.000 Max. :1.000 Max. :0.851

After.tax.net.Interest.Rate Non.industry.income.and.expenditure.revenue

Min. :0.000 Min. :0.000

1st Qu.:0.809 1st Qu.:0.303

Median :0.809 Median :0.304

Mean :0.809 Mean :0.304

3rd Qu.:0.809 3rd Qu.:0.304

Max. :0.864 Max. :1.000

Continuous.interest.rate..after.tax. Operating.Expense.Rate

Min. :0.000 Min. :0.00e+00

1st Qu.:0.782 1st Qu.:0.00e+00

Median :0.782 Median :0.00e+00

Mean :0.781 Mean :1.98e+09

3rd Qu.:0.782 3rd Qu.:4.07e+09

Max. :0.829 Max. :9.99e+09

## Distributions and Relationships

This graph shows the high prevalence in the data, only 3% of the companies in the data have a bankruptcy and the other 97% have no bankruptcy

## 

## C. Models and Discoveries

## We plan to use all three models: Decision Tree, Neural Network, and Logistic Regression models. Then we will compare the accuracy statistics of the three possible models to summarize our discoveries and predictions for Taiwanese bankruptcies with relatively more important variables from the dataset. For the Decision Tree model, we will be practicing different splits to get a result that has relatively higher accuracy. For the Logistic Regression model, we will use stepwise variable selection technique. formulas to change the distribution of the variable dataset to get multiple models. For the Neural Network model, we will manual balancing of the dataset to overcome biased predictions.

## D. Logistic Regression Analysis Logistic Regression is a statistical technique that is commonly used for binary classification. It is a form of regression analysis that is particularly effective in predicting the likelihood of an outcome that can only have two possible values, typically 0 and 1, or true and false. Despite its name, logistic regression is actually a classification algorithm rather than a regression algorithm.

## We read the csv file into a dataframe using the below code.

banking.df <- read.csv("Bank.csv")

head(banking.df)

We split the data with 50% in the training data set and 50% of the data in the validation data set.

train.index <- sample(c(1:dim(banking.df)[1]), dim(banking.df)[1]\*0.5)

train.df <- banking.df[train.index, ]

valid.df <- banking.df[-train.index, ]

Regression Model:

logit.reg <- glm(Bankrupt. ~ ., data = train.df, family = "binomial"); summary(logit.reg)

Output:

Call:

glm(formula = Bankrupt. ~ ., family = "binomial", data = train.df)

Coefficients: (4 not defined because of singularities)

Estimate

(Intercept) 47557704976050623086592.0000000

ROA.C..before.interest.and.depreciation.before.interest 3861177575451011.0000000

ROA.A..before.interest.and...after.tax 5597256931598609.0000000

ROA.B..before.interest.and.depreciation.after.tax -5442148194487207.0000000

Operating.Gross.Margin -7930611129466464256.0000000

Realized.Sales.Gross.Margin -29700448783638668.0000000

Operating.Profit.Rate -392526385968989208576.0000000

Pre.tax.net.Interest.Rate 327810415790435139584.0000000

After.tax.net.Interest.Rate 506706031062821376.0000000

Non.industry.income.and.expenditure.revenue -187553744622750433280.0000000

Continuous.interest.rate..after.tax. -514524648887278144.0000000

Operating.Expense.Rate -11214.4186360

Research.and.development.expense.rate -5246.8923922

Cash.flow.rate -6267490763761008.0000000

Interest.bearing.debt.interest.rate -422557.1948020

Tax.rate..A. -1220499124395565.0000000

Net.Value.Per.Share..B. -20916153873557568.0000000

Net.Value.Per.Share..A. -27245384584554972.0000000

Net.Value.Per.Share..C. 45964561399667392.0000000

Persistent.EPS.in.the.Last.Four.Seasons -7220964394468634.0000000

Cash.Flow.Per.Share 2258197413247646.0000000

Revenue.Per.Share..Yuan... 32351571.3982751

Operating.Profit.Per.Share..Yuan... 15157257993696300.0000000

Per.Share.Net.profit.before.tax..Yuan... -6091275751915917.0000000

Realized.Sales.Gross.Profit.Growth.Rate 4411468853613372.5000000

Operating.Profit.Growth.Rate -2949408840625904.5000000

After.tax.Net.Profit.Growth.Rate 2267458637022825.5000000

Regular.Net.Profit.Growth.Rate 2634246550222861.0000000

Continuous.Net.Profit.Growth.Rate 561849137790832.3125000

Total.Asset.Growth.Rate 24520.3092485

Net.Value.Growth.Rate -8854561855454414.0000000

Total.Asset.Return.Growth.Rate.Ratio 1860910118402148.2500000

Cash.Reinvestment.. 11164837556014784.0000000

Current.Ratio 118977584290392.3281250

Quick.Ratio -83461.2464943

Interest.Expense.Ratio -18091749968040844.0000000

Total.debt.Total.net.worth 2434197.5016291

Debt.ratio.. 7981503288198739.0000000

Net.worth.Assets NA

Long.term.fund.suitability.ratio..A. -2346507633751327.5000000

Borrowing.dependency 22760784278716012.0000000

Contingent.liabilities.Net.worth -38867481354902184.0000000

Operating.profit.Paid.in.capital -7966712837872639.0000000

Net.profit.before.tax.Paid.in.capital 7789976003566904.0000000

Inventory.and.accounts.receivable.Net.value 31316519134876684.0000000

Total.Asset.Turnover -2047999338166265.2500000

Accounts.Receivable.Turnover -500365.3818581

Average.Collection.Days -464795.8367555

Inventory.Turnover.Rate..times. -5429.9963648

Fixed.Assets.Turnover.Frequency 35240.1319482

Net.Worth.Turnover.Rate..times. -826814374168451.8750000

Revenue.per.person 1486152.9806758

Operating.profit.per.person -1343630799523870.2500000

Allocation.rate.per.person -163416.3397742

Working.Capital.to.Total.Assets -62995337513853029187584.0000000

Quick.Assets.Total.Assets 660696681707818.7500000

Current.Assets.Total.Assets 17253931795360035897344.0000000

Cash.Total.Assets -731993108245517.8750000

Quick.Assets.Current.Liability -451729.8353641

Cash.Current.Liability -112709.8072649

Current.Liability.to.Assets -56191700127919579856896.0000000

Operating.Funds.to.Liability 2027267920880872.5000000

Inventory.Working.Capital -99309446175291.3593750

Inventory.Current.Liability -130283.1768921

Current.Liabilities.Liability 1061367969566675.2500000

Working.Capital.Equity -72714565171619296.0000000

Current.Liabilities.Equity -67423124095290208.0000000

Long.term.Liability.to.Current.Assets 23467.1595624

Retained.Earnings.to.Total.Assets 10962601432150526.0000000

Total.income.Total.expense 6198671169639676.0000000

Total.expense.Assets -1328083830716495.2500000

Current.Asset.Turnover.Rate 3770.5570255

Quick.Asset.Turnover.Rate 13801.2836560

Working.capitcal.Turnover.Rate -428298182248112512.0000000

Cash.Turnover.Rate -10809.3593744

Cash.Flow.to.Sales 260178659096119968.0000000

Fixed.Assets.to.Assets 461036.4887741

Current.Liability.to.Liability NA

Current.Liability.to.Equity NA

Equity.to.Long.term.Liability -1437338555312922.7500000

Cash.Flow.to.Total.Assets 1600826024447828.0000000

Cash.Flow.to.Liability -5730013147368155.0000000

CFO.to.Assets -2370924638147149.5000000

Cash.Flow.to.Equity -10294863353318250.0000000

Current.Liability.to.Current.Assets -7536151787212027.0000000

Liability.Assets.Flag 2949244392313608.5000000

Net.Income.to.Total.Assets -11326146970284826.0000000

Total.assets.to.GNP.price -80112.0153634

No.credit.Interval 1068193246712751.7500000

Gross.Profit.to.Sales 7957395228922233856.0000000

Net.Income.to.Stockholder.s.Equity -30386297449031172.0000000

Liability.to.Equity -26717164528461832.0000000

Degree.of.Financial.Leverage..DFL. -1329086942772221.0000000

Interest.Coverage.Ratio..Interest.expense.to.EBIT. -328727816948963.5000000

Net.Income.Flag NA

Equity.to.Liability -7527715704739385.0000000

Std. Error z value

(Intercept) 2135677811292377.5000000 22268202

ROA.C..before.interest.and.depreciation.before.interest 163126591.6695676 23669823

ROA.A..before.interest.and...after.tax 130547763.3545838 42875165

ROA.B..before.interest.and.depreciation.after.tax 208984228.1744182 -26040952

Operating.Gross.Margin 554128785352.2565918 -14311856

Realized.Sales.Gross.Margin 2784756018.5642214 -10665368

Operating.Profit.Rate 20036964823823.3164062 -19590112

Pre.tax.net.Interest.Rate 16735636435601.8671875 19587568

After.tax.net.Interest.Rate 11354788978.8006401 44624874

Non.industry.income.and.expenditure.revenue 9578919355270.9218750 -19579844

Continuous.interest.rate..after.tax. 12577784688.3336525 -40907414

Operating.Expense.Rate 0.0004038 -27773664

Research.and.development.expense.rate 0.0004638 -11312543

Cash.flow.rate 197548210.8925543 -31726386

Interest.bearing.debt.interest.rate 0.0113905 -37097399

Tax.rate..A. 9861350.9736392 -123765915

Net.Value.Per.Share..B. 1373896906.5497797 -15223962

Net.Value.Per.Share..A. 2310192001.3973317 -11793559

Net.Value.Per.Share..C. 1864996867.3070951 24645919

Persistent.EPS.in.the.Last.Four.Seasons 269225884.3347601 -26821212

Cash.Flow.Per.Share 208033829.7603141 10854953

Revenue.Per.Share..Yuan... 0.9029875 35827264

Operating.Profit.Per.Share..Yuan... 871218054.5829204 17397778

Per.Share.Net.profit.before.tax..Yuan... 247782192.9922526 -24583186

Realized.Sales.Gross.Profit.Growth.Rate 509198106.6567418 8663561

Operating.Profit.Growth.Rate 429501793.5781869 -6867047

After.tax.Net.Profit.Growth.Rate 716253062.7123406 3165723

Regular.Net.Profit.Growth.Rate 708095000.8515679 3720188

Continuous.Net.Profit.Growth.Rate 83139895.0264956 6757876

Total.Asset.Growth.Rate 0.0004322 56727583

Net.Value.Growth.Rate 1936373327.9313061 -4572756

Total.Asset.Return.Growth.Rate.Ratio 397620375.8533697 4680118

Cash.Reinvestment.. 167285770.8315005 66741107

Current.Ratio 105590459.3656072 1126783

Quick.Ratio 0.0092197 -9052538

Interest.Expense.Ratio 339040836.5751566 -53361566

Total.debt.Total.net.worth 0.0416932 58383531

Debt.ratio.. 132299468.2177572 60329066

Net.worth.Assets NA NA

Long.term.fund.suitability.ratio..A. 53137529.9067462 -44159140

Borrowing.dependency 437608262.8600204 52011779

Contingent.liabilities.Net.worth 652875326.1550131 -59532777

Operating.profit.Paid.in.capital 873485624.9456029 -9120600

Net.profit.before.tax.Paid.in.capital 349495680.7148021 22289191

Inventory.and.accounts.receivable.Net.value 558234237.7610515 56099245

Total.Asset.Turnover 28677940.9089630 -71413751

Accounts.Receivable.Turnover 0.0061063 -81941983

Average.Collection.Days 0.0054994 -84517793

Inventory.Turnover.Rate..times. 0.0003787 -14340104

Fixed.Assets.Turnover.Frequency 0.0005302 66468912

Net.Worth.Turnover.Rate..times. 80005525.9135108 -10334466

Revenue.per.person 0.0462687 32120050

Operating.profit.per.person 50466006.3738801 -26624473

Allocation.rate.per.person 0.0031903 -51223443

Working.Capital.to.Total.Assets 2839893577597782.0000000 -22182288

Quick.Assets.Total.Assets 12378572.2370668 53374224

Current.Assets.Total.Assets 777824693620457.5000000 22182289

Cash.Total.Assets 14902791.8631590 -49117851

Quick.Assets.Current.Liability 0.0082975 -54441890

Cash.Current.Liability 0.0020870 -54005352

Current.Liability.to.Assets 2533178546765165.5000000 -22182290

Operating.Funds.to.Liability 96002098.9084444 21116912

Inventory.Working.Capital 115134756.7909931 -862550

Inventory.Current.Liability 0.0021405 -60864478

Current.Liabilities.Liability 19637654.1553033 54047595

Working.Capital.Equity 729634724.2158189 -99658860

Current.Liabilities.Equity 1876270206.7946739 -35934656

Long.term.Liability.to.Current.Assets 0.0019144 12258491

Retained.Earnings.to.Total.Assets 97548723.1884274 112380778

Total.income.Total.expense 90957600.0584913 68149019

Total.expense.Assets 87775490.0127947 -15130463

Current.Asset.Turnover.Rate 0.0005024 7505731

Quick.Asset.Turnover.Rate 0.0004118 33516933

Working.capitcal.Turnover.Rate 5957001666.4122562 -71898281

Cash.Turnover.Rate 0.0004211 -25666790

Cash.Flow.to.Sales 7030794218.5404320 37005586

Fixed.Assets.to.Assets 0.0099685 46249325

Current.Liability.to.Liability NA NA

Current.Liability.to.Equity NA NA

Equity.to.Long.term.Liability 431907423.0448223 -3327886

Cash.Flow.to.Total.Assets 57579622.2190747 27801954

Cash.Flow.to.Liability 74518946.9388520 -76893373

CFO.to.Assets 70971647.7735062 -33406645

Cash.Flow.to.Equity 166995278.2389673 -61647631

Current.Liability.to.Current.Assets 86800874.8963427 -86821150

Liability.Assets.Flag 65930421.2807740 44732679

Net.Income.to.Total.Assets 175497177.7266171 -64537488

Total.assets.to.GNP.price 0.0035127 -22806238

No.credit.Interval 81422415.4033442 13119154

Gross.Profit.to.Sales 554087718537.1064453 14361255

Net.Income.to.Stockholder.s.Equity 437967014.3314694 -69380333

Liability.to.Equity 1979112128.5257914 -13499571

Degree.of.Financial.Leverage..DFL. 57295193.4039361 -23197180

Interest.Coverage.Ratio..Interest.expense.to.EBIT. 103092911.4355536 -3188656

Net.Income.Flag NA NA

Equity.to.Liability 57385290.1623956 -131178490

Pr(>|z|)

(Intercept) <0.0000000000000002 \*\*\*

ROA.C..before.interest.and.depreciation.before.interest <0.0000000000000002 \*\*\*

ROA.A..before.interest.and...after.tax <0.0000000000000002 \*\*\*

ROA.B..before.interest.and.depreciation.after.tax <0.0000000000000002 \*\*\*

Operating.Gross.Margin <0.0000000000000002 \*\*\*

Realized.Sales.Gross.Margin <0.0000000000000002 \*\*\*

Operating.Profit.Rate <0.0000000000000002 \*\*\*

Pre.tax.net.Interest.Rate <0.0000000000000002 \*\*\*

After.tax.net.Interest.Rate <0.0000000000000002 \*\*\*

Non.industry.income.and.expenditure.revenue <0.0000000000000002 \*\*\*

Continuous.interest.rate..after.tax. <0.0000000000000002 \*\*\*

Operating.Expense.Rate <0.0000000000000002 \*\*\*

Research.and.development.expense.rate <0.0000000000000002 \*\*\*

Cash.flow.rate <0.0000000000000002 \*\*\*

Interest.bearing.debt.interest.rate <0.0000000000000002 \*\*\*

Tax.rate..A. <0.0000000000000002 \*\*\*

Net.Value.Per.Share..B. <0.0000000000000002 \*\*\*

Net.Value.Per.Share..A. <0.0000000000000002 \*\*\*

Net.Value.Per.Share..C. <0.0000000000000002 \*\*\*

Persistent.EPS.in.the.Last.Four.Seasons <0.0000000000000002 \*\*\*

Cash.Flow.Per.Share <0.0000000000000002 \*\*\*

Revenue.Per.Share..Yuan... <0.0000000000000002 \*\*\*

Operating.Profit.Per.Share..Yuan... <0.0000000000000002 \*\*\*

Per.Share.Net.profit.before.tax..Yuan... <0.0000000000000002 \*\*\*

Realized.Sales.Gross.Profit.Growth.Rate <0.0000000000000002 \*\*\*

Operating.Profit.Growth.Rate <0.0000000000000002 \*\*\*

After.tax.Net.Profit.Growth.Rate <0.0000000000000002 \*\*\*

Regular.Net.Profit.Growth.Rate <0.0000000000000002 \*\*\*

Continuous.Net.Profit.Growth.Rate <0.0000000000000002 \*\*\*

Total.Asset.Growth.Rate <0.0000000000000002 \*\*\*

Net.Value.Growth.Rate <0.0000000000000002 \*\*\*

Total.Asset.Return.Growth.Rate.Ratio <0.0000000000000002 \*\*\*

Cash.Reinvestment.. <0.0000000000000002 \*\*\*

Current.Ratio <0.0000000000000002 \*\*\*

Quick.Ratio <0.0000000000000002 \*\*\*

Interest.Expense.Ratio <0.0000000000000002 \*\*\*

Total.debt.Total.net.worth <0.0000000000000002 \*\*\*

Debt.ratio.. <0.0000000000000002 \*\*\*

Net.worth.Assets NA

Long.term.fund.suitability.ratio..A. <0.0000000000000002 \*\*\*

Borrowing.dependency <0.0000000000000002 \*\*\*

Contingent.liabilities.Net.worth <0.0000000000000002 \*\*\*

Operating.profit.Paid.in.capital <0.0000000000000002 \*\*\*

Net.profit.before.tax.Paid.in.capital <0.0000000000000002 \*\*\*

Inventory.and.accounts.receivable.Net.value <0.0000000000000002 \*\*\*

Total.Asset.Turnover <0.0000000000000002 \*\*\*

Accounts.Receivable.Turnover <0.0000000000000002 \*\*\*

Average.Collection.Days <0.0000000000000002 \*\*\*

Inventory.Turnover.Rate..times. <0.0000000000000002 \*\*\*

Fixed.Assets.Turnover.Frequency <0.0000000000000002 \*\*\*

Net.Worth.Turnover.Rate..times. <0.0000000000000002 \*\*\*

Revenue.per.person <0.0000000000000002 \*\*\*

Operating.profit.per.person <0.0000000000000002 \*\*\*

Allocation.rate.per.person <0.0000000000000002 \*\*\*

Working.Capital.to.Total.Assets <0.0000000000000002 \*\*\*

Quick.Assets.Total.Assets <0.0000000000000002 \*\*\*

Current.Assets.Total.Assets <0.0000000000000002 \*\*\*

Cash.Total.Assets <0.0000000000000002 \*\*\*

Quick.Assets.Current.Liability <0.0000000000000002 \*\*\*

Cash.Current.Liability <0.0000000000000002 \*\*\*

Current.Liability.to.Assets <0.0000000000000002 \*\*\*

Operating.Funds.to.Liability <0.0000000000000002 \*\*\*

Inventory.Working.Capital <0.0000000000000002 \*\*\*

Inventory.Current.Liability <0.0000000000000002 \*\*\*

Current.Liabilities.Liability <0.0000000000000002 \*\*\*

Working.Capital.Equity <0.0000000000000002 \*\*\*

Current.Liabilities.Equity <0.0000000000000002 \*\*\*

Long.term.Liability.to.Current.Assets <0.0000000000000002 \*\*\*

Retained.Earnings.to.Total.Assets <0.0000000000000002 \*\*\*

Total.income.Total.expense <0.0000000000000002 \*\*\*

Total.expense.Assets <0.0000000000000002 \*\*\*

Current.Asset.Turnover.Rate <0.0000000000000002 \*\*\*

Quick.Asset.Turnover.Rate <0.0000000000000002 \*\*\*

Working.capitcal.Turnover.Rate <0.0000000000000002 \*\*\*

Cash.Turnover.Rate <0.0000000000000002 \*\*\*

Cash.Flow.to.Sales <0.0000000000000002 \*\*\*

Fixed.Assets.to.Assets <0.0000000000000002 \*\*\*

Current.Liability.to.Liability NA

Current.Liability.to.Equity NA

Equity.to.Long.term.Liability <0.0000000000000002 \*\*\*

Cash.Flow.to.Total.Assets <0.0000000000000002 \*\*\*

Cash.Flow.to.Liability <0.0000000000000002 \*\*\*

CFO.to.Assets <0.0000000000000002 \*\*\*

Cash.Flow.to.Equity <0.0000000000000002 \*\*\*

Current.Liability.to.Current.Assets <0.0000000000000002 \*\*\*

Liability.Assets.Flag <0.0000000000000002 \*\*\*

Net.Income.to.Total.Assets <0.0000000000000002 \*\*\*

Total.assets.to.GNP.price <0.0000000000000002 \*\*\*

No.credit.Interval <0.0000000000000002 \*\*\*

Gross.Profit.to.Sales <0.0000000000000002 \*\*\*

Net.Income.to.Stockholder.s.Equity <0.0000000000000002 \*\*\*

Liability.to.Equity <0.0000000000000002 \*\*\*

Degree.of.Financial.Leverage..DFL. <0.0000000000000002 \*\*\*

Interest.Coverage.Ratio..Interest.expense.to.EBIT. <0.0000000000000002 \*\*\*

Net.Income.Flag NA

Equity.to.Liability <0.0000000000000002 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 930.67 on 3408 degrees of freedom

Residual deviance: 6776.21 on 3317 degrees of freedom

AIC: 6960.2

Number of Fisher Scoring iterations: 17

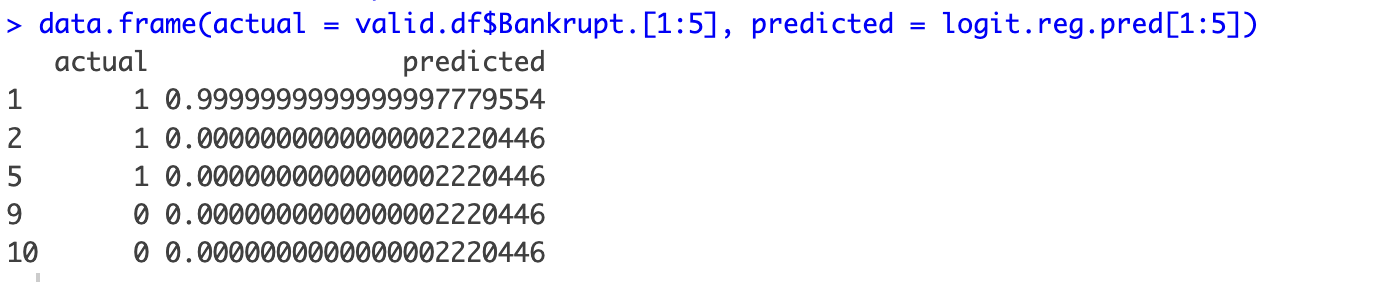
Using logistic regression, we are predicting the probabilities of Y=1 with the validation dataset

logit.reg.pred <- predict(logit.reg, valid.df, type = "response")

first 5 actual and predicted records

data.frame(actual = valid.df$Bankrupt.[1:5], predicted = logit.reg.pred[1:5])

Output: The actual values are from the dataset and predicted values are from the logistic regression model.

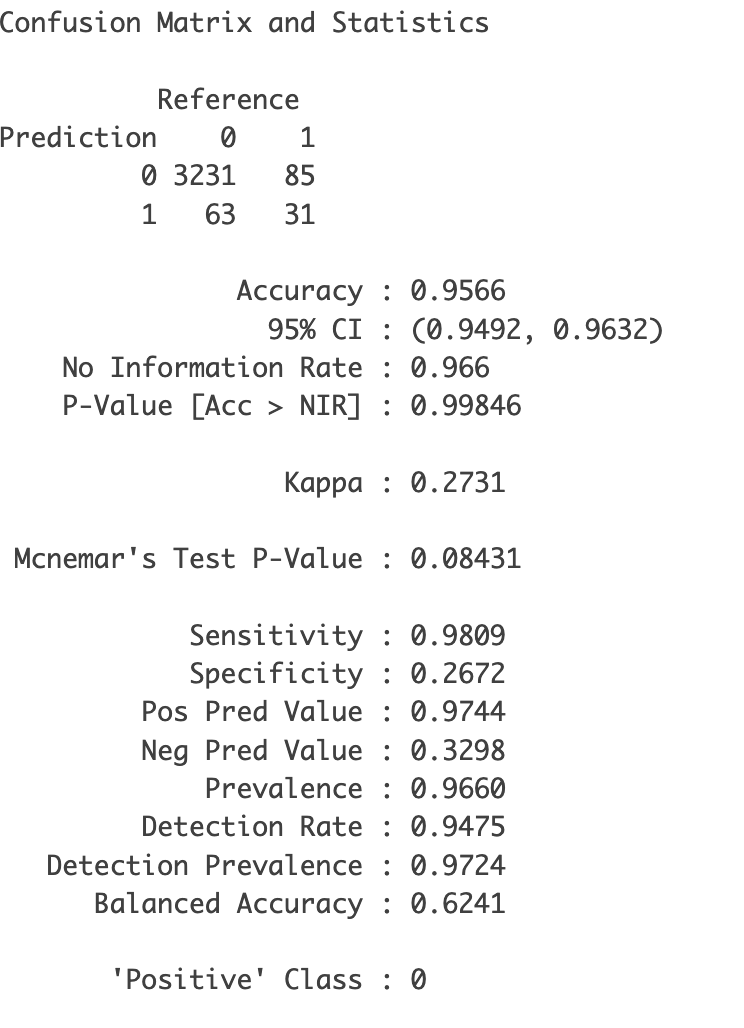


From the logistic model we only get the probability of Y = 1 so we have to make the data categorical using a filter. In this case the best filter was 0.5.

0.5 above as 1 and 0.5 below as 0

logit.reg.pred.classes <- ifelse(logit.reg.pred > 0.5, 1, 0)

confusionMatrix(as.factor(logit.reg.pred.classes), as.factor(valid.df$Bankrupt.))



From the confusion matrix, we get an accuracy of 95.66%. 3231 companies have been predicted to be non bankrupt and the model is right. 31 companies have been predicted to be bankrupt and it matches the actual model.

In logistic regression, we can use sequential variable selection techniques. There are 3 model.

**Forward Selection:** It considers all the models obtained by adding one more explanatory variable(X) to the current model.

**Backward Elimination:**

The initial model contains all possible explanatory variables and removes them as the model progresses.

**Stepwise:**

The constant mean model with no explanatory variables is the starting model. We do one step of forward selection and one step of backward selection. The model which has the lowest AIC is selected. In this project, stepwise selection has been used to find the best model with the best explanatory variables.

R code for stepwise selection.

full.logit.reg <- glm(Bankrupt. ~ ., data = train.df, family = "binomial")

empty.logit.reg <- glm(Bankrupt. ~ 1,data = train.df, family= "binomial")

summary(empty.logit.reg)

stepwise = step(empty.logit.reg,scope=list(lower=formula(empty.logit.reg),upper=formula(full.logit.reg)), direction="both",trace=1)

formula(stepwise)

The final explanatory variables after the stepwise selection are:

Bankrupt. ~ Persistent.EPS.in.the.Last.Four.Seasons + Cash.Total.Assets +

Equity.to.Liability + Cash.Flow.to.Liability + Total.Asset.Turnover +

Contingent.liabilities.Net.worth + Cash.Flow.to.Total.Assets +

Net.Income.to.Total.Assets + Interest.Expense.Ratio + Fixed.Assets.Turnover.Frequency +

Current.Liabilities.Liability + Inventory.Turnover.Rate..times. +

Per.Share.Net.profit.before.tax..Yuan... + Net.Value.Per.Share..B. +

Net.Value.Per.Share..C. + Retained.Earnings.to.Total.Assets +

Cash.Turnover.Rate + Interest.bearing.debt.interest.rate + Equity.to.Long.term.Liability

The AIC = 539.88 (lowest)

building a model with the new variables selected by the model.

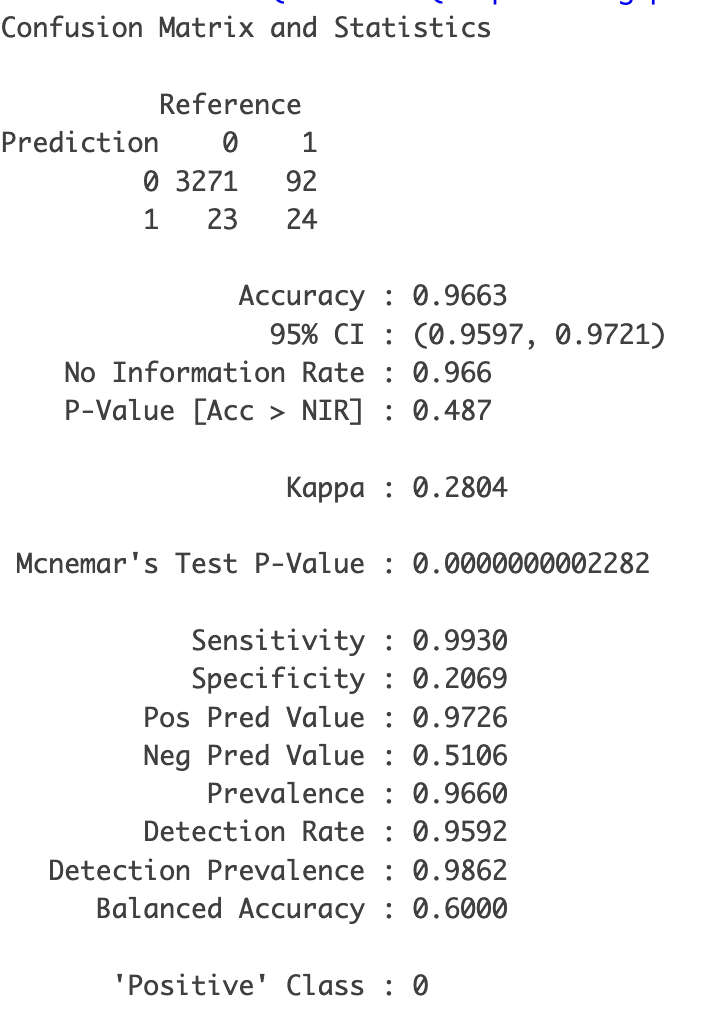
R code for the model.

stepwise.reg.pred <- predict(stepwise, valid.df, type = "response")

stepwise.reg.pred.classes <- ifelse(stepwise.reg.pred > 0.4, 1, 0)

confusionMatrix(as.factor(stepwise.reg.pred.classes), as.factor(valid.df$Bankrupt.))

The cut of variable of 0.5 has provided the highest accuracy rate.



**Comparing the full model to the stepwise model:**

The accuracy of the stepwise model is higher than the full model. Also, logistic regression is a greate technique for variable selection for any dataset. It involves fitting all possible subset models and identifying the ones that best satisfy some model fitting criterion.

## E. Decision tree Analysis

## A decision tree is a powerful and popular tool in machine learning and data analysis. It is a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility.

require(mosaic)

library(reshape2)

library(ggplot2)

library(rpart)

library(rpart.plot)

library(caret)

require(caret)

library(pROC)

bankruptcy.df <- read.csv("data (version 1) - variable segregation.csv")

bankruptcy.df1 <- bankruptcy.df[c(1:9,96)]

summary(bankruptcy.df1)

sum(is.na.data.frame(bankruptcy.df1))

**Spliting training & validation dataset**

# partition

set.seed(1)

train.index <- sample(c(1:dim(bankruptcy.df1)[1]), dim(bankruptcy.df1)[1]\*0.5)

train.df <- bankruptcy.df1[train.index, ]

valid.df <- bankruptcy.df1[-train.index, ]

**Building default decision tree**

default.ct <- rpart(Bankrupt. ~ ., data = train.df ,method = "class")

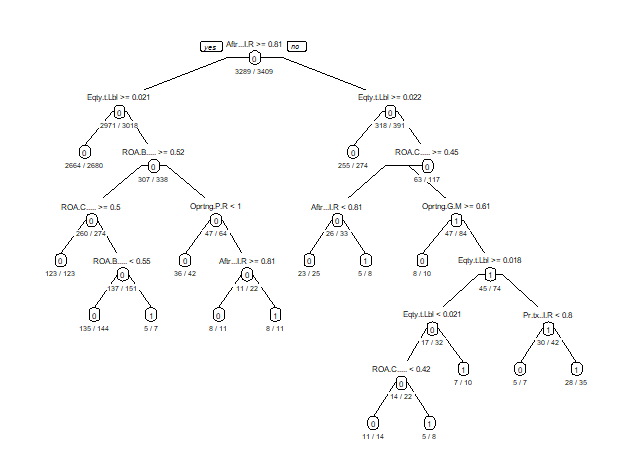
# plot tree

prp(default.ct, type = 1, extra = 2, under = TRUE, split.font = 1, varlen = 10)

# count number of leaves

length(default.ct$frame$var[default.ct$frame$var == "<leaf>"])

*16*



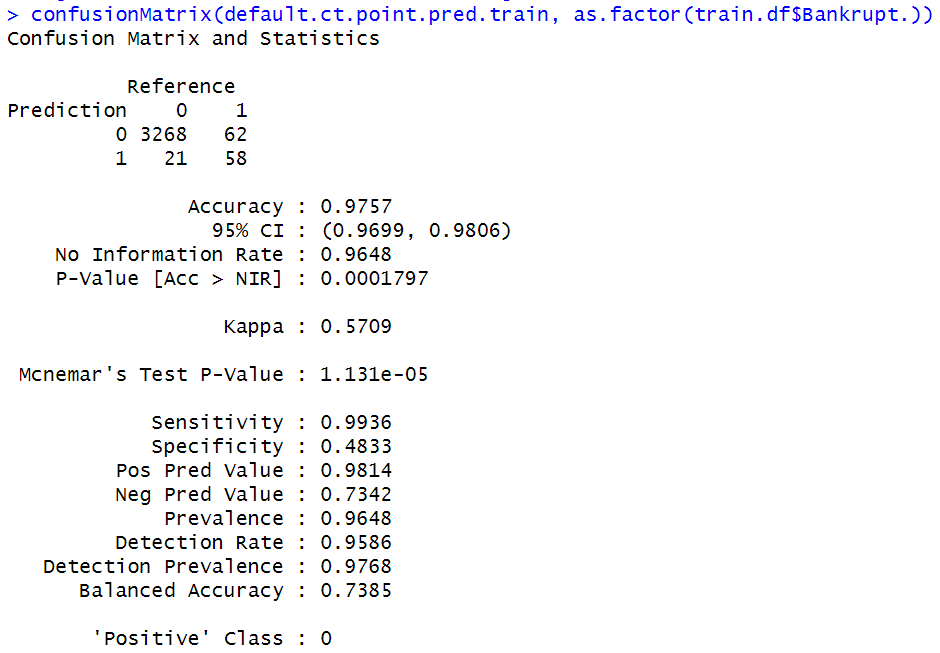
**Precision data on the default decision tree**

# classify records in the training data.

default.ct.point.pred.train <- predict(default.ct,train.df,type = "class")

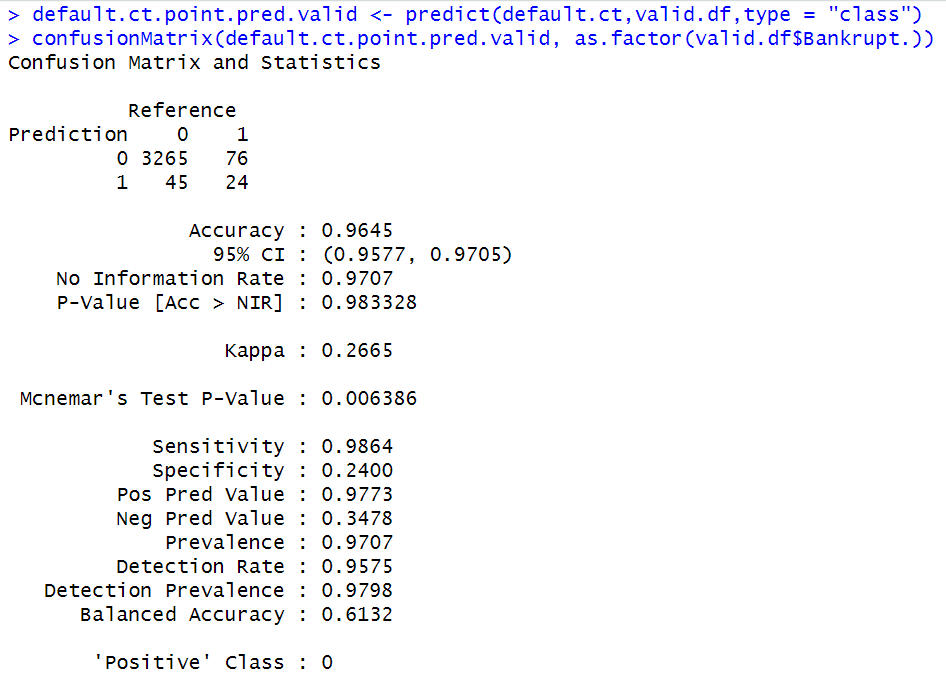
# generate confusion matrix for training data

confusionMatrix(default.ct.point.pred.train, as.factor(train.df$Bankrupt.))



default.ct.point.pred.valid <- predict(default.ct,valid.df,type = "class")

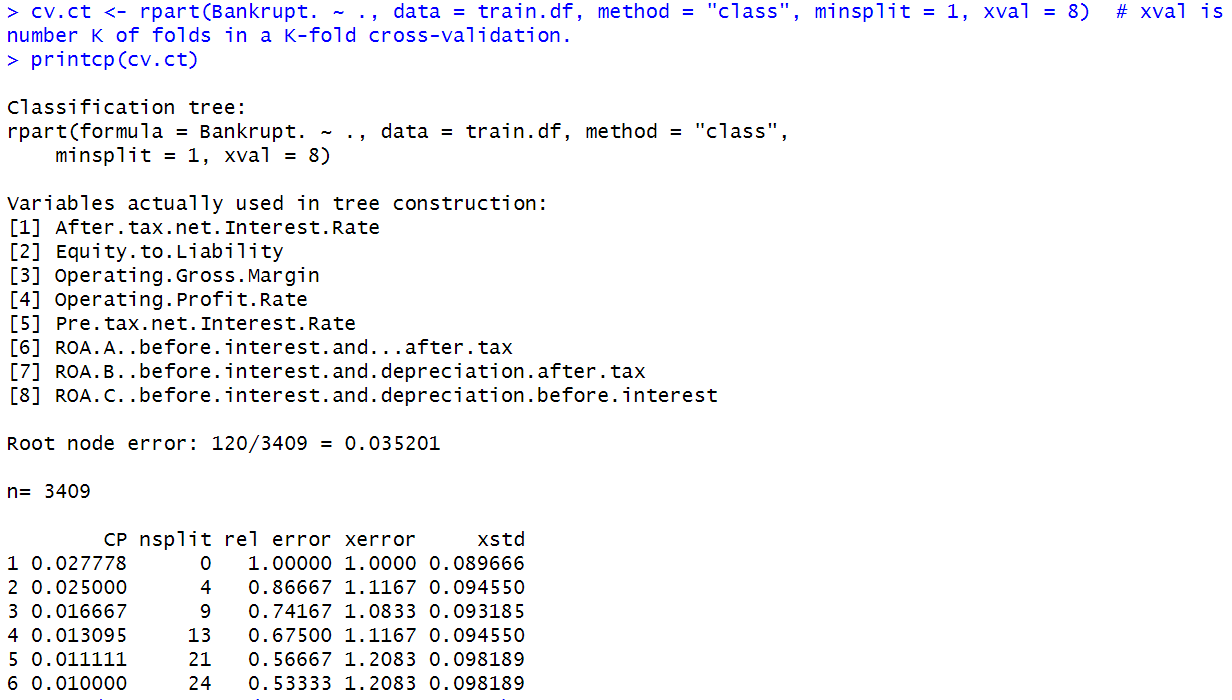
confusionMatrix(default.ct.point.pred.valid, as.factor(valid.df$Bankrupt.))



**Cross-validation procedure**

cv.ct <- rpart(Bankrupt. ~ ., data = train.df, method = "class", minsplit = 1, xval = 8) # xval is number K of folds in a K-fold cross-validation.

printcp(cv.ct)



**Pruned decision tree**

pruned.ct <- prune(cv.ct, cp = 0.016668)

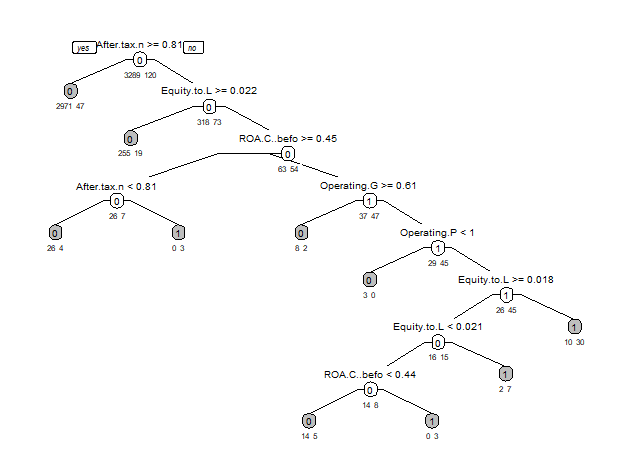
printcp(pruned.ct)

length(pruned.ct$frame$var[pruned.ct$frame$var == "<leaf>"])

*10*

prp(pruned.ct, type = 1, extra = 1, under = TRUE, split.font = 1, varlen = -10,

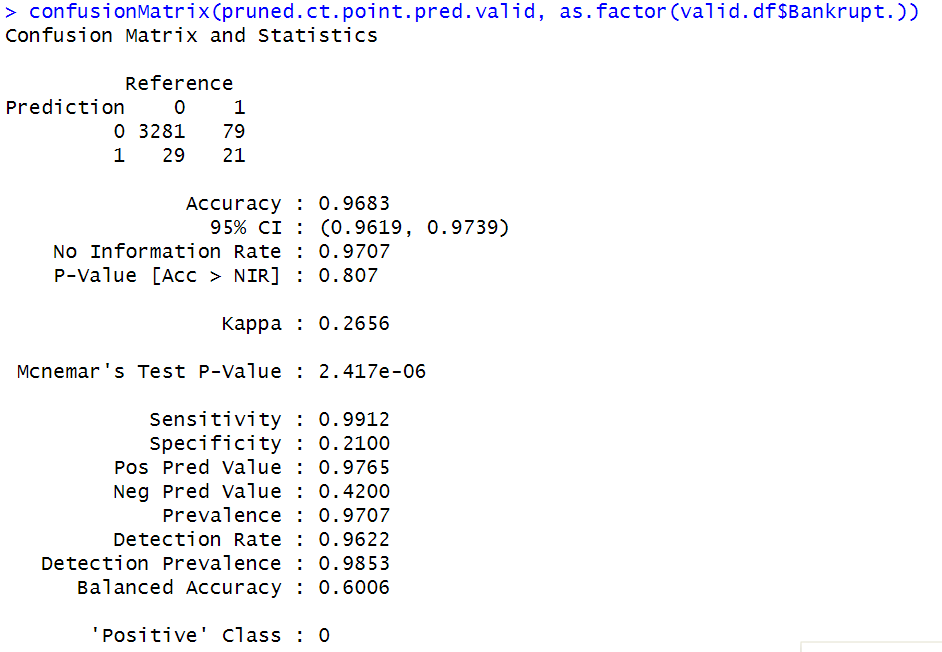
box.col=ifelse(pruned.ct$frame$var == "<leaf>", 'gray', 'white'))



**Precision data on the pruned decision tree**

pruned.ct.point.pred.valid <- predict(pruned.ct,valid.df,type = "class")

confusionMatrix(pruned.ct.point.pred.valid, as.factor(valid.df$Bankrupt.))



* ***Before Post-Pruning:***

The default classification tree has 16 leaves, 15 splits. The accuracy for the validation dataset of this decision tree is 0.9645.

* ***Post-Pruning Method:***

Deciding on using the post-pruning method to avoid overfitting problem, we ran the cross-validation procedure on the dataset. We found that either less than or more than 9 splits, the error rate is increasing. Therefore, we concluded that 9 splits work the best for this model.

* ***After Post-Pruning:***

The new classification tree has 10 leaves, 9 splits. The precision data is better than the default decision tree, in terms of accuracy is 0.9683.

## F. Neural Network Analysis

## A neural network is a computational model inspired by the structure and functioning of the human brain. It is a key component of machine learning, a field of artificial intelligence. Neural networks consist of interconnected nodes, often organized into layers, and are used for various tasks such as pattern recognition, classification, regression, and decision-making.

## **R Code:**

library(tidyverse)

library(dplyr)

library(caret)

library(psych)

library(fastDummies)

library(nnet)

library(e1071)

library(kernlab)

library(randomForest)

library(ggplot2)

library(nortest)

library(performanceEstimation)

library(rpart)

library(rpart.plot)

library(reshape)

library(adabag)

library(mboost)

library(neuralnet)

library(gbm)

library(cluster)

library(factoextra)

data <- read.csv("NN-Data.csv")

data

str(data)

#Split trial 2

set.seed(123) # for reproducibility

indices <- createDataPartition(data$Bankrupt, p = 0.7, list = FALSE)

# Create training and initial test sets

train\_data\_initial <- data[indices, ]

test\_data\_initial <- data[-indices, ]

# Manually balance the train set

n <- min(sum(train\_data\_initial$Bankrupt == 0), sum(train\_data\_initial$Bankrupt == 1))

train\_data\_balanced <- rbind(

na.omit(train\_data\_initial[train\_data\_initial$Bankrupt == 0, ])[1:n, ],

na.omit(train\_data\_initial[train\_data\_initial$Bankrupt == 1, ])[1:n, ]

)

# Manually balance the test set

n <- min(sum(test\_data\_initial$Bankrupt == 0), sum(test\_data\_initial$Bankrupt == 1))

test\_data\_balanced <- rbind(

na.omit(test\_data\_initial[test\_data\_initial$Bankrupt == 0, ])[1:n, ],

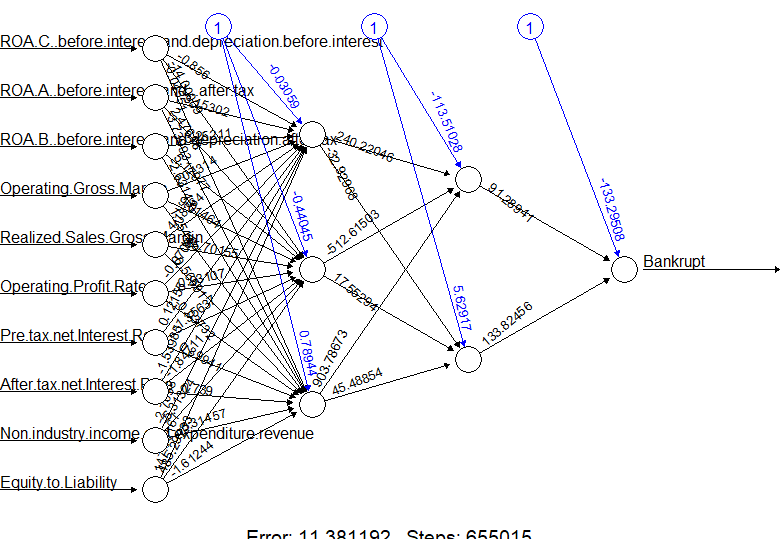
na.omit(test\_data\_initial[test\_data\_initial$Bankrupt == 1, ])[1:n, ]

)

#Neural Network code

nn <- neuralnet(Bankrupt ~ ., data = train\_data\_balanced, linear.output = F, hidden = c(3,2), stepmax = 1e6)

plot(nn, rep="best")



nn.pred <- predict(nn, train\_data\_balanced, type = "response")

nn.pred.classes <- ifelse(nn.pred > 0.5, 1, 0)

Conf\_mat\_train <- confusionMatrix(as.factor(nn.pred.classes), as.factor(train\_data\_balanced$Bankrupt))

Conf\_mat\_train

nn.pred <- predict(nn, test\_data\_balanced, type = "response")

nn.pred.classes <- ifelse(nn.pred > 0.5, 1, 0)

Conf\_mat\_test <- confusionMatrix(as.factor(nn.pred.classes), as.factor(test\_data\_balanced$Bankrupt))

Conf\_mat\_test

**Result:**

#Neural Network code

> nn <- neuralnet(Bankrupt ~ ., data = train\_data\_balanced, linear.output = F, hidden = c(3,2), stepmax = 1e6)

> plot(nn, rep="best")

> nn.pred <- predict(nn, train\_data\_balanced, type = "response")

> nn.pred.classes <- ifelse(nn.pred > 0.5, 1, 0)

> Conf\_mat\_train <- confusionMatrix(as.factor(nn.pred.classes), as.factor(train\_data\_balanced$Bankrupt))

> Conf\_mat\_train

Confusion Matrix and Statistics

Reference

Prediction 0 1

0 147 19

1 15 143

Accuracy : 0.8951

95% CI : (0.8565, 0.9262)

No Information Rate : 0.5

P-Value [Acc > NIR] : <2e-16

Kappa : 0.7901

Mcnemar's Test P-Value : 0.6069

Sensitivity : 0.9074

Specificity : 0.8827

Pos Pred Value : 0.8855

Neg Pred Value : 0.9051

Prevalence : 0.5000

Detection Rate : 0.4537

Detection Prevalence : 0.5123

Balanced Accuracy : 0.8951

'Positive' Class : 0

***Interpretation for Training set:***

* The model shows high accuracy and balanced accuracy on the training set.
* Sensitivity and specificity are high, indicating good performance in correctly identifying positive and negative instances.
* Precision is high, indicating a low rate of false positives.

These results suggest that the model performs well on the training set. However, evaluating the model on a separate validation or test set is important to ensure its performance generalizes to new, unseen data. Overfitting to the training set is a common concern, and assessing performance on a different dataset helps mitigate this risk.

> nn.pred <- predict(nn, test\_data\_balanced, type = "response")

> nn.pred.classes <- ifelse(nn.pred > 0.5, 1, 0)

> Conf\_mat\_test <- confusionMatrix(as.factor(nn.pred.classes), as.factor(test\_data\_balanced$Bankrupt))

> Conf\_mat\_test

Confusion Matrix and Statistics

Reference

Prediction 0 1

0 54 8

1 4 50

Accuracy : 0.8966

95% CI : (0.8263, 0.9454)

No Information Rate : 0.5

P-Value [Acc > NIR] : <2e-16

Kappa : 0.7931

Mcnemar's Test P-Value : 0.3865

Sensitivity : 0.9310

Specificity : 0.8621

Pos Pred Value : 0.8710

Neg Pred Value : 0.9259

Prevalence : 0.5000

Detection Rate : 0.4655

Detection Prevalence : 0.5345

Balanced Accuracy : 0.8966

'Positive' Class : 0

***Interpretation for Test set:***

* The model demonstrates good accuracy and balanced accuracy on the test set.
* Both sensitivity and specificity are high, indicating good performance in correctly identifying positive and negative instances.
* Precision is high, indicating a low rate of false positives.

Overall, the model seems to generalize well to new data, showing robust performance on the test set. This suggests that the model has learned meaningful patterns from the training data that apply to unseen instances.

**G. Conclusion:**

The data set is highly imbalanced so blatantly running any model on that dataset will end up giving us biased predictions and inefficiency. So, it is better to use several balancing methods before training the model.   
  
In Neural Networks we have used manual balancing and were able to achieve 90% accuracy with ~ 91% sensitivity and specificity which helps us provide better predictions in the long run.