Business Intelligence

Machine Learning models

Contents

[Investigation of Data: 2](#_Toc4790240)

[Logistic Regression: 2](#_Toc4790241)

[Decision Trees: 6](#_Toc4790242)

[Comparative Analysis: 10](#_Toc4790243)

[Recommendations: 10](#_Toc4790244)

# Investigation of Data:

Large data set with various data types had be used for developing a prediction model. Inbuilt functions in R were used to detect the missing data ad the rows in which the data was missing. Later two predictions models were developed i.e.; Logistic Regression and Decision Trees. Data types of the variables had to be changes for the Decision Trees. These models were trained, tested and later used for prediction. Each of the model has its own pros and cons which is detailed further in the report. The dependent or response variable was ‘writeoff’ which was regressed with other variables.

# Logistic Regression:

Description:

Logistic Regression is a powerful statistical way of modeling a binomial outcome with one or more explanatory variables. It measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution. The dependent variable ‘writeoff’ and other variables are used form the data set to develop a logistic regression model. Accuracy of the model is tested using the “fitted values”.

Code:

library(tidyr)

library(dplyr)

library(ggplot2)

library(corrgram)

library(gridExtra)

library(Deducer)

library(caret)

library(pscl)

library(plyr)

library(tree)

library(ISLR)

library(randomForest)

library(e1071)

loanslog <- read.csv("Loans.csv", header = T, stringsAsFactors = F)

head(loanslog)

print(loanslog)

glimpse(loanslog)

table(loanslog$writeoff)

str(loanslog)

#TrainingModel

loanslog$writeoff <- as.factor(loanslog$writeoff)

def <- ifelse(trainlog$writeoff == 'yes',1,2)

glm1 <- glm(writeoff~., data = trainlog, family = binomial)

summary(glm1)

plot(glm1$fitted.values)

names(glm1)

tblog <- table(glm1$fitted.values>0,def)

sum(diag(tblog))/sum(tblog)

#Prediction

loanspredlog <- read.csv("Loans\_test.csv", header = T, stringsAsFactors = F)

fitted.results <- predict(glm1,newdata=loanspredlog,type='response')

fitted.results <- ifelse(fitted.results > 0.5,1,0)

table(fitted.results)

writeoffLog <- ifelse(fitted.results == '1','yes','no')

predLogistic <- data.frame(loanspredlog,writeoffLog)

predLogistic

write.csv(predLogistic, 'Prediction\_LogRegression.csv')

Pros and Cons:

**Pros:**

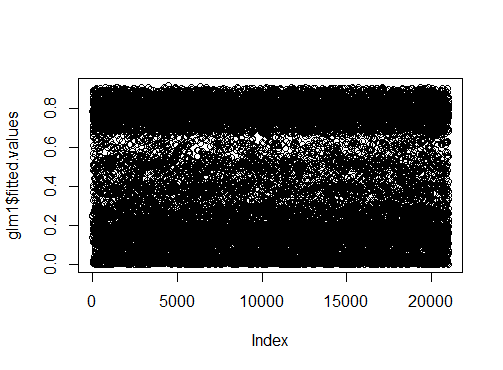
* Convenient probability scores for observations
* Efficient implementations available across tools
* Multi-collinearity is not really an issue and can be countered with L2 regularization to an extent
* Wide spread industry comfort for logistic regression solutions [ oh that’s important too!]

**Cons:**

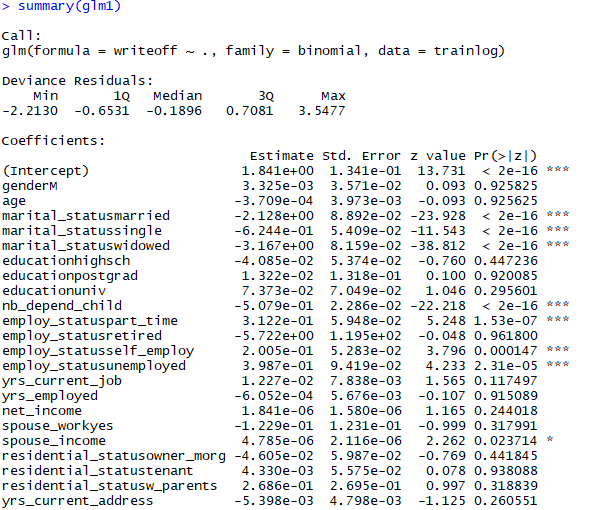
* Doesn’t perform well when feature space is too large
* Doesn’t handle large number of categorical features/variables well
* Relies on transformations for non-linear features
* Relies on entire data [ Not a very serious drawback I’d say]

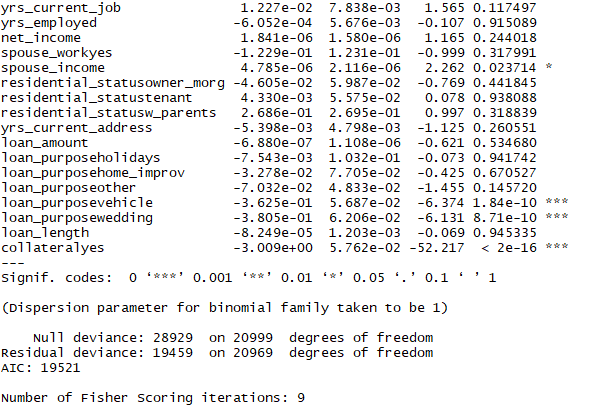
Results:

1)Plot

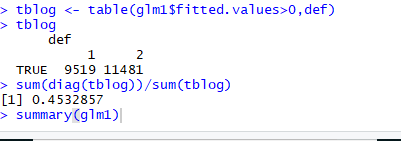
LogisticRegression

2)Summary





3)Table and Accuracy



# Decision Trees:

Description:

Decision trees are diagrams that attempt to display the range of possible outcomes and subsequent decisions made after an initial decision. Decision trees provide a framework to consider the probability, which can help you analyze a decision to make the most informed decision possible. A training model is developed based on the response variable ‘writeoff’ regressing with the other variables in training data set to obtain the decision trees. The training model is then applied on the testing model to measure the accuracy of the model developed. The model is then applied on the prediction data set to predict the loans defaulters.

**Code:**

library(tidyr)

library(dplyr)

library(ggplot2)

library(corrgram)

library(gridExtra)

library(Deducer)

library(caret)

library(pscl)

library(plyr)

library(tree)

library(ISLR)

library(randomForest)

#Model Building

loans <- read.csv("Loans.csv", header = T, stringsAsFactors = F)

head(loans)

print(loans)

glimpse(loans)

table(loans$writeoff)

str(loans)

#Data Type conversion

#Factor

loans$writeoff <- as.factor(loans$writeoff)

loans$gender <- as.factor(loans$gender)

loans$marital\_status <- as.factor(loans$marital\_status)

loans$education <- as.factor(loans$education)

loans$employ\_status <- as.factor(loans$employ\_status)

loans$spouse\_work <- as.factor(loans$spouse\_work)

loans$residential\_status <- as.factor(loans$residential\_status)

loans$loan\_purpose <- as.factor(loans$loan\_purpose)

loans$collateral <- as.factor(loans$collateral)

#Numeric

loans$age <- as.numeric(loans$age)

loans$nb\_depend\_child <- as.numeric(loans$nb\_depend\_child)

loans$yrs\_current\_job <- as.numeric(loans$yrs\_current\_job)

loans$yrs\_employed <- as.numeric(loans$yrs\_employed)

loans$net\_income <- as.numeric(loans$net\_income)

loans$spouse\_income <- as.numeric(loans$spouse\_income)

loans$yrs\_current\_address <- as.numeric(loans$yrs\_current\_address)

loans$loan\_amount <- as.numeric(loans$loan\_amount)

loans$loan\_length <- as.numeric(loans$loan\_length)

#looking for missing data

is.na(loans)

loans[!complete.cases(loans),]

#Splitting the DataFrame to training and testing dataFrames

set.seed(77)

n=nrow(loans)

ind<-sample(1:n,21000)

train<-loans[ind,]

head(train)

dim(train)

test<-loans[-ind,]

#Training the model with 'writeoff' as response variable using train data

tree.train=tree(writeoff~.,data=train)

plot(tree.train)

text(tree.train,pretty=0)

tree.train

summary(tree.train)

#Testing the training model for accuracy using testing data

tree.test = predict(tree.train,test, "class")

tree.test

tb<-table(tree.test, test$writeoff) #confusion matrix

tb

acc<-sum(diag(tb))/sum(tb) #accuracy

acc #accuracy is 85.16%

#Using the training model to predict the 'writeoffs':

loanpred <- read.csv("Loans\_test.csv", header = T, stringsAsFactors = F)

dim(loanpred)

#Data type Conversion

#Factor

loanpred$gender <- as.factor(loanpred$gender)

loanpred$marital\_status <- as.factor(loanpred$marital\_status)

loanpred$education <- as.factor(loanpred$education)

loanpred$employ\_status <- as.factor(loanpred$employ\_status)

loanpred$spouse\_work <- as.factor(loanpred$spouse\_work)

loanpred$residential\_status <- as.factor(loanpred$residential\_status)

loanpred$loan\_purpose <- as.factor(loanpred$loan\_purpose)

loanpred$collateral <- as.factor(loanpred$collateral)

#Numeric

loanpred$age <- as.numeric(loanpred$age)

loanpred$nb\_depend\_child <- as.numeric(loanpred$nb\_depend\_child)

loanpred$yrs\_current\_job <- as.numeric(loanpred$yrs\_current\_job)

loanpred$yrs\_employed <- as.numeric(loanpred$yrs\_employed)

loanpred$net\_income <- as.numeric(loanpred$net\_income)

loanpred$spouse\_income <- as.numeric(loanpred$spouse\_income)

loanpred$yrs\_current\_address <- as.numeric(loanpred$yrs\_current\_address)

loanpred$loan\_amount <- as.numeric(loanpred$loan\_amount)

loanpred$loan\_length <- as.numeric(loanpred$loan\_length)

n<-length(loanpred)

wr<-numeric(n)

for(i in n){

writeoff=predict(tree.train,loanpred, "class")

writeoff[wr]<-writeoff

}

Prediction\_df<-data.frame(loanpred,writeoff)

head(Prediction\_df)

table(writeoff)

Prediction\_df

dim(Prediction\_df)

#writing the dataframe to csv

write.csv(Prediction\_df, 'Prediction\_DeciosionTrees.csv')

Pros and Cons:

Pros:

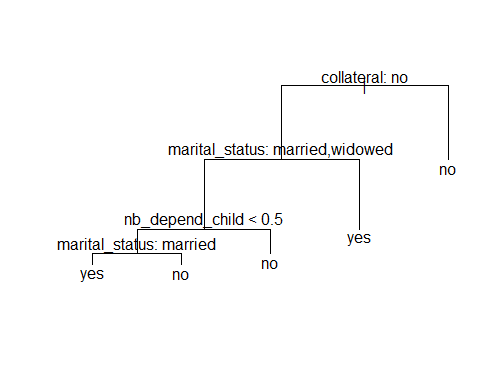
* Interpretable
* Irrelevant values are handled easily
* Handles missing value data
* Compact and fast testing time

Cons:

* Only axis-aligned split of data
* Greedy as it may not find the best tree out of the many possible ones

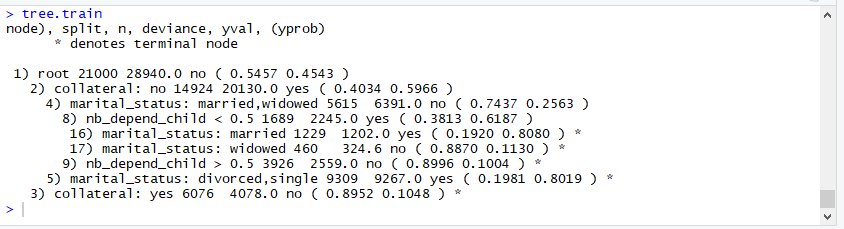
Results:

1)Plots

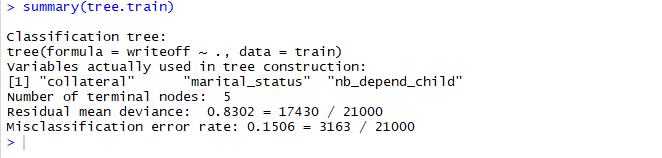


Decision Trees

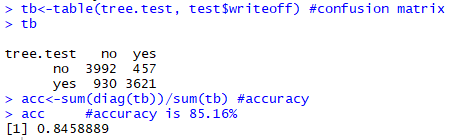
2)Model



3)Summary



4)Table and Accuracy



# Comparative Analysis:

Based on the accuracy and the variables used by the algorithms, it can be inferred that Decision Trees provide better classification of the variables used and better accuracy to predict the defaulters. It is visually easy to understand the tree classification, which helps to analyze the effect of each variable on the response variable.

# Recommendations:

ABSA Bank can use decision trees to predict the defaulters as the accuracy is higher. Based on the models developed, the customers probability of defaulting is predicted as ‘writeoff’ is either ‘yes’ or ‘no’.

