**Predicting Earning Manipulations by Indian Firms using Machine Learning Algorithms**

# Case Study

Data Mining – IDS 575

Ashima Horra – UIN:654087182

# Predicting Earning Manipulations by Indian Firms using Machine Learning Algorithms

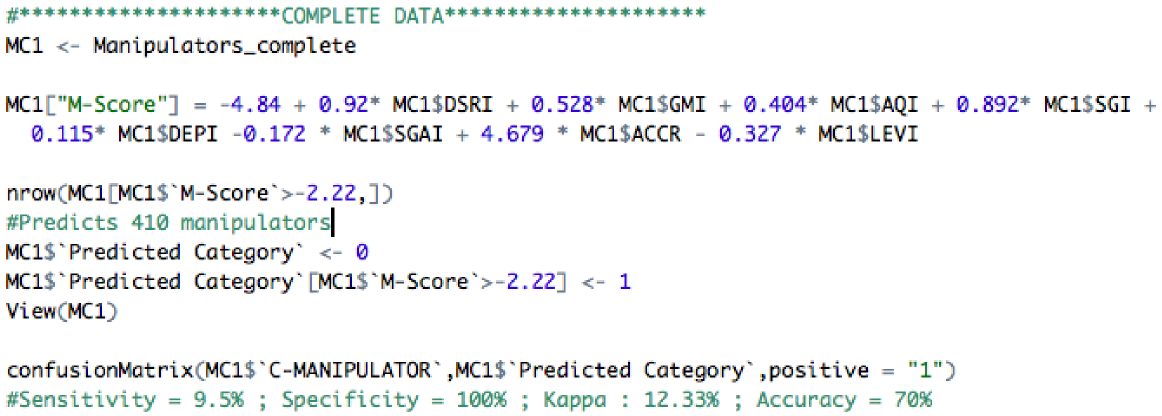
## Beneish Model

The Beneish model was developed on US data in 1999 and uses the 8 financial ratios.

$$\textbf{M-Score} = -4.84 + 0.92 \times DSRI + 0.528 \times GMI + 0.404 \times AQI + 0.892 \times SGI + 0.115 \times DEPI - 0.172 \times SGAI + 4.679 \times TATA - 0.327 \times LVGI$$

*If M-Score is > -2.22 the company is likely to be a manipulator*
*If M-Score is < -2.22 the company is unlikely to be a manipulator*

The model isn't a very good model for Indian data as the data is very unbalanced and this model doesn't give good results when run on Indian data. It predicts all the manipulators; however, it classifies a lot of non-manipulators as manipulators. It has a recall of only 9.5%



```
#*********************COMPLETE DATA*********************
MC1 <- Manipulators_complete

MC1["M-Score"] = -4.84 + 0.92* MC1$DSRI + 0.528* MC1$GMI + 0.404* MC1$AQI + 0.892* MC1$SGI +
  0.115* MC1$DEPI -0.172 * MC1$SGAI + 4.679 * MC1$ACCR - 0.327 * MC1$LEVI

nrow(MC1[MC1$`M-Score`>-2.22,])
#Predicts 410 manipulators
MC1$`Predicted Category` <- 0
MC1$`Predicted Category`[MC1$`M-Score`>-2.22] <- 1
View(MC1)

confusionMatrix(MC1$`C-MANIPULATOR`,MC1$`Predicted Category`,positive = "1")
#Sensitivity = 9.5% ; Specificity = 100% ; Kappa : 12.33% ; Accuracy = 70%
```

## Unbalanced data problems and solutions

While looking at earnings manipulator data, we see that the number of manipulators is very less as compared to the non-manipulators. Only 4-5% of manipulators. This kind of data is unbalanced and leads to a lot of problems in classification problems.

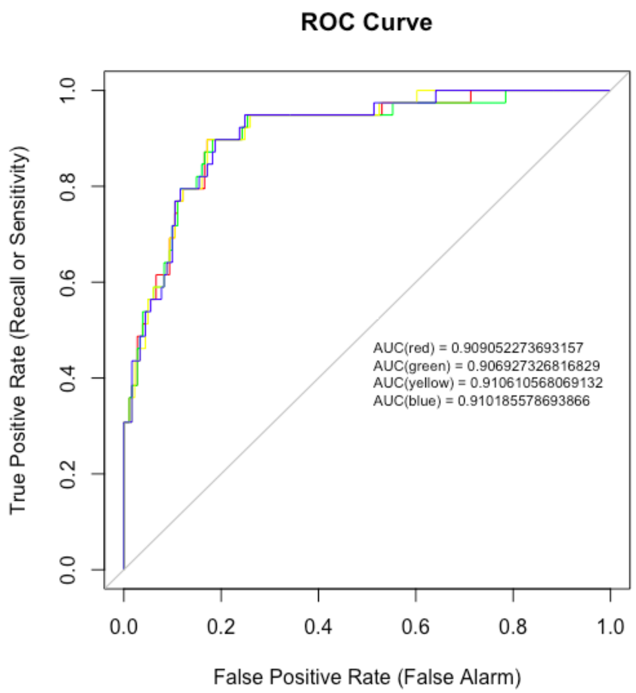
1. Classification problems tend to be biased towards the majority class. The features of the minority class are treated as noise and are often ignored. Thus, there is a high probability of misclassification of the minority class as compared to the majority class
2. A model with 96% accuracy would look like a good model but in a case where there are 5% manipulators in a dataset, it would mean that only a very small percentage of the manipulators are getting predicted. This completely defeats the purpose.

There are few things that can be done to deal with the problems with unbalanced data:

1. We can balance the unbalanced data. There are a lot of ways to do so, some of which are:
    a) Random Under-Sampling – Randomly eliminating some instances of the majority class to get it in a comparable ratio with the minority class.
    b) Random Over-Sampling – Randomly over sampling the minority class by replicating some values of the minority class.
    c) Informed Over Sampling: Synthetic Minority Over-Sampling Technique: Over sampling the minority class in such a way that the new instances are derived from the existing minority instances and are close to each other.
2. Chose a different measure to evaluate the model. Accuracy isn't the best measure to evaluate model performance in case of unbalanced datasets. Instead, measure like recall, specificity, precision, false alarm, F-Score, ROC curves or auc should be used.

## Logistic regression model on sample data

The model created using all the eight financial ratios shows that there are three insignificant variables. We use backward variable selection to select variables and use the auc and AIC statistic to decide the model.



Here, we compare 4 models based on there are user ROC curves and find that the yellow and the blue models are performing better than the rest. We then perform repetitive cross validation to get the best sensitivity, specificity, f-score and kappa statistic to decide the final model. The model with the following five features is selected: DSRI, GMI, AQI, SGI, ACCR

We calculate the threshold of the model using the minimum distance from point (1,0) and the minimum Youden's index. The threshold in this case comes out to 0.13. So, all the instances with probability greater than 0.13 are predicted as manipulators (1) and the ones with probability below 0.13 are predicted as non -manipulators (0)

We use repetitive cross validation (training 70% and test-30%) to evaluate how the model performs. We run the cross validation 10 times to get a better idea of how the model performs.
*We don't use k-fold cross validation since the dataset is very small and causes problems creating stratifies k folds.
*We use repetitive cross validation since the 30% test in just one iteration could be a biased estimate of the dataset. The data is shuffled properly after every iteration so make sure we get unbiased test sets and take the average of all 10 iteration evaluation measures to finally evaluate the model.

To evaluate the model we use sensitivity, F-score and kappa statistic as the evaluation measures.
Sensitivity or recall – Percentage of true positives out of those predicted positive.
Specificity - Percentage of true negatives out of those predicted negatives
F-Score- Balanced estimate of the model's sensitivity and precision
Cohen's **kappa** coefficient– Kappa statistic is used as an accuracy measure of unbalanced data sets.

The logistic model generated has an average Sensitivity of 0.5505 ; Average F-Score of 0.635124, an average Kappa coefficient of 0.5783 and an average specificity is 0.970982

## Strategy to deploy the logistic regression model:

The model's threshold is 0.13 so MCA technologies should use the flowing strategy:

After running the logistic regression model on the set of firms in the dataset, the firms which have a predicted probability of more than 0.13 should be predicted as manipulators.
Prob (Manipulator = 1) > 0.13 => Manipulator

And, the firms which have a predicted probability of less than 0.13 are predicted as non-manipulators.
Prob (Manipulator = 1) < 0.13 => Non-Manipulator

## M-Score – Manipulator Score

Based on the logistic model developed on the sample data the following M-Score should be used:

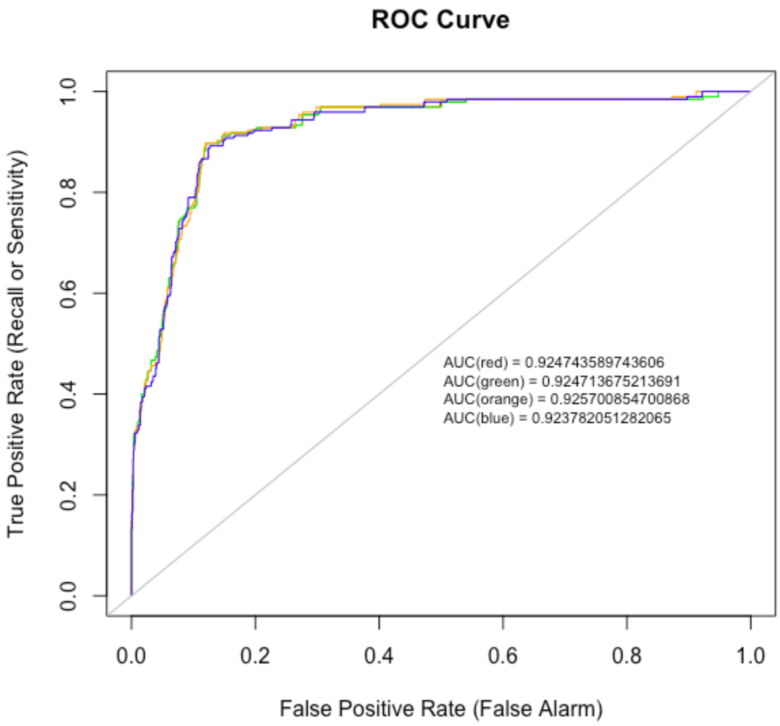
**M-Score = -8.0325 + 1.01 DSRI + 1.188 GMI + 0.522 AQI + 2.351 SGI + 9.2331 ACCR**

*If M-Score is less than -1.9 then the company is unlikely to be a manipulator*
*If M-Score is greater than -1.9 then the company is likely to be a manipulator*

## Logistic regression model on complete dataset

To perform logistic regression on the complete data we first balance the dataset using SMOTE (Synthetic Minority Over-Sampling Technique). The final data set has 17% positives (which is comparable to the sample dataset)

We compare the four models as we did in case of sample data and chose the model with the largest area under the ROC curve. The model with the following six financial measures has been used: DSRI + GMI + AQI + SGI + DEPI + ACCR

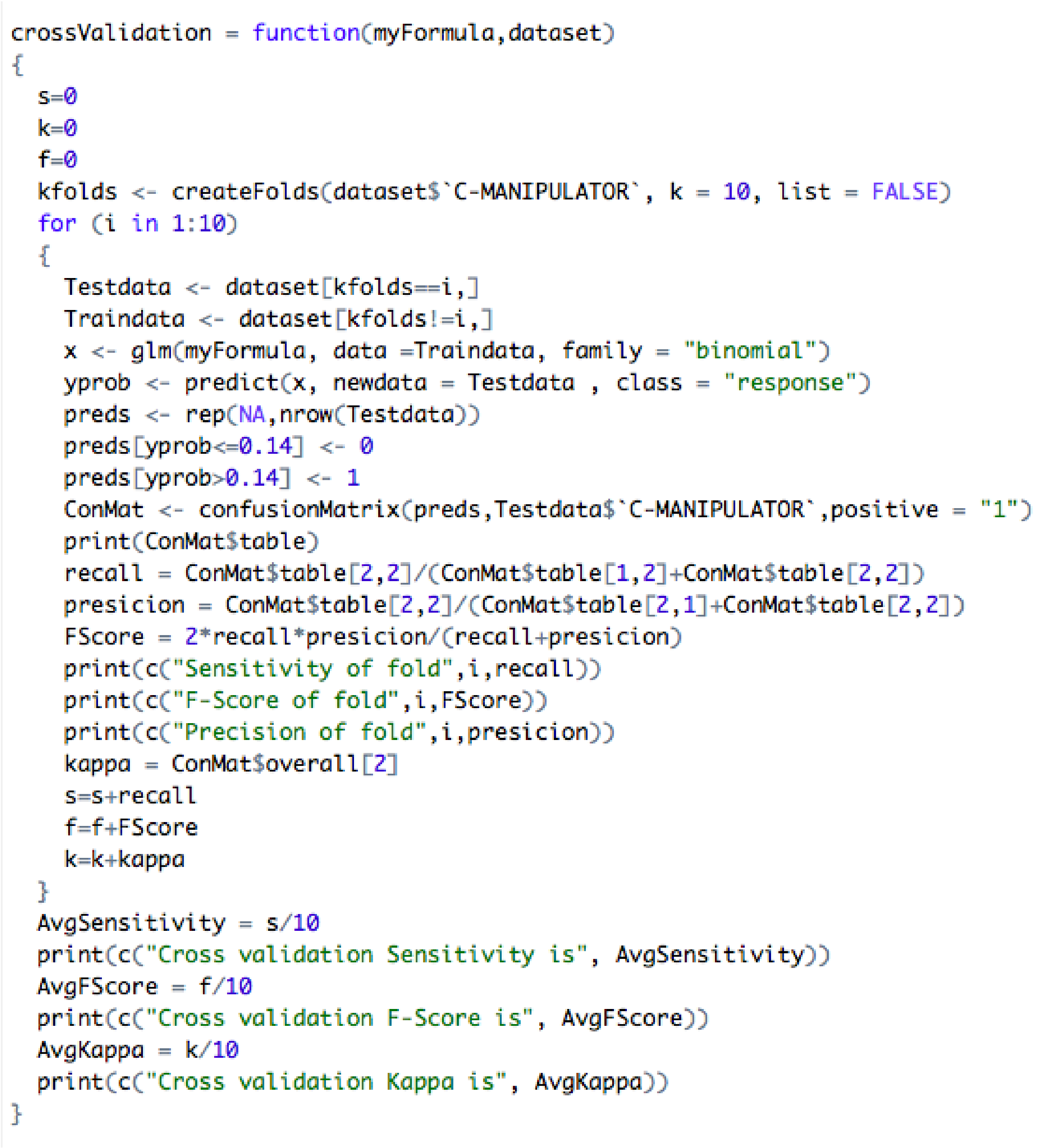


The threshold according to the minimum distance for this model is 0.14. So, all the instances with probability greater than 0.14 are predicted as manipulators (1) and the ones with probability below 0.14 are predicted as non-manipulators (0)

We use 10-fold cross validation to evaluate the performance of the model. We chose to use 10-fold cross validation since the size of the data set is sufficiently large to be able to do so and it gives a better estimate (with lower variance) of the overall model. While doing cross validation, we estimate the Kappa coefficient, sensitivity, specificity and the F-score of the model.

The model developed on the complete data has a Cross Validation Sensitivity is 0.39447 ; Cross validation F-Score of 0.51112, a Cross validation Kappa of 0.4602 and a cross validation specificity of 0.973333

The performance of this model has reduced when compared to the model generated on the sample data.

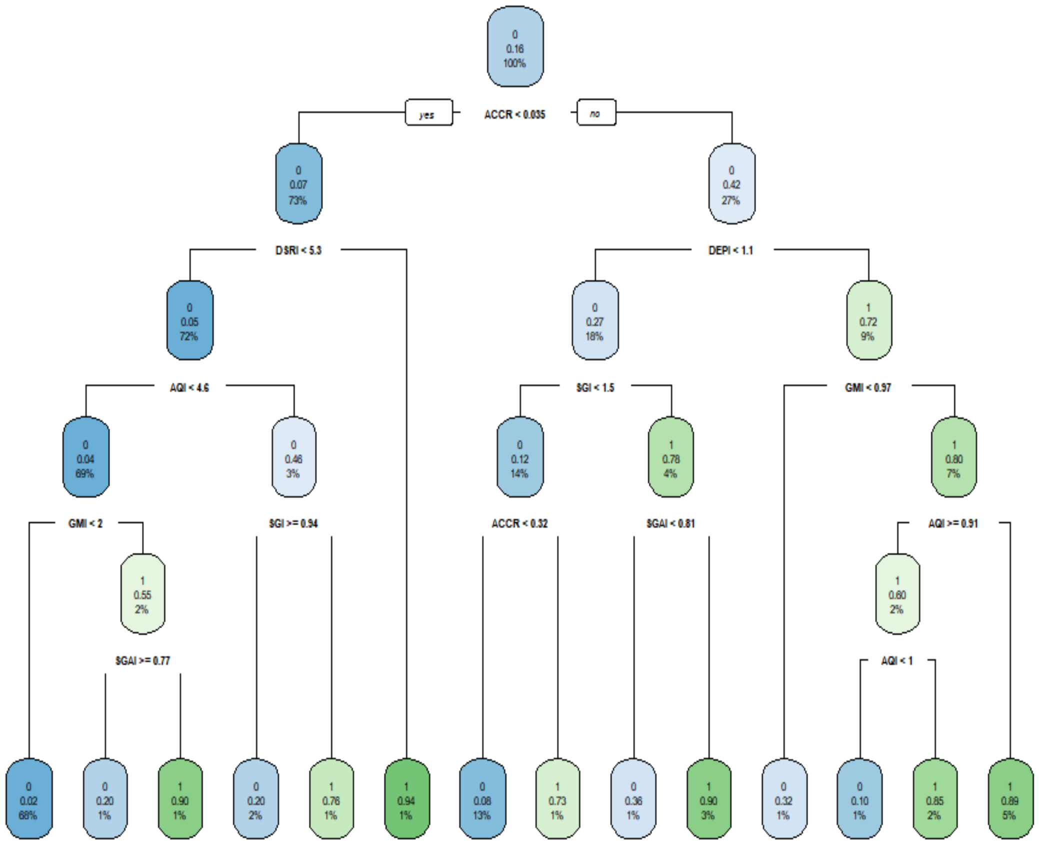


```
crossValidation = function(myFormula,dataset)
{
  s=0
  k=0
  f=0
  kfolds <- createFolds(dataset$`C-MANIPULATOR`, k = 10, list = FALSE)
  for (i in 1:10)
  {
    Testdata <- dataset[kfolds==i,]
    Traindata <- dataset[kfolds!=i,]
    x <- glm(myFormula, data =Traindata, family = "binomial")
    yprob <- predict(x, newdata = Testdata , class = "response")
    preds <- rep(NA,nrow(Testdata))
    preds[yprob<=0.14] <- 0
    preds[yprob>0.14] <- 1
    ConMat <- confusionMatrix(preds,Testdata$`C-MANIPULATOR`,positive = "1")
    print(ConMat$table)
    recall = ConMat$table[2,2]/(ConMat$table[1,2]+ConMat$table[2,2])
    presicion = ConMat$table[2,2]/(ConMat$table[2,1]+ConMat$table[2,2])
    FScore = 2*recall*presicion/(recall+presicion)
    print(c("Sensitivity of fold",i,recall))
    print(c("F-Score of fold",i,FScore))
    print(c("Precision of fold",i,presicion))
    kappa = ConMat$overall[2]
    s=s+recall
    f=f+FScore
    k=k+kappa
  }
  AvgSensitivity = s/10
  print(c("Cross validation Sensitivity is", AvgSensitivity))
  AvgFScore = f/10
  print(c("Cross validation F-Score is", AvgFScore))
  AvgKappa = k/10
  print(c("Cross validation Kappa is", AvgKappa))
}
```

## Decision Tree

We have constructed the decision tree on the full dataset for which we first balance the dataset using SMOTE (Synthetic Minority Over-Sampling Technique). The final data set has 17% positives (which is comparable to the sample dataset). We then use 10-fold cross validation to evaluate the performance of the model (same as logistic regression)

The model developed on the complete data has a Cross Validation Sensitivity is 0.68387; Cross validation F-Score of 0.70790, Cross validation Kappa of 0.65292 and a cross validation specificity of 0.95083



**Prediction rules:**

1. If ACCR < 0.035, DSRI < 5.3 and AQI <4.6 then the customer is not involved in earning manipulation
   - 70% of the total number of observations support this rule
2. If ACCR > 0.035, DEPI > 1.1GMI <0.97 and AQI not between 0.91 and 1 then the customer is involved in earning manipulation
   - 7% of the total number of observations support this rule
3. If ACCR between 0.035,0.32, DEPI < 1.1 and SGI <1.5, then the customer is not involved in earning manipulation
   - 13% of the total number of observations support this rule

Decision trees are performing better compared to Logistic Regression model

## Models created through Machine Learning Algorithms

The Machine Learning algorithms we have used to create the models are random forest and Ada boosting

Similar to decision trees, we have constructed these models on the complete dataset that has been balanced and have then used 10 fold cross validation to evaluate the performance of these models.

**Comparison of Random Forest, Ada Boosting, Decision Trees and Logistic Regression models**

| Model | Sensitivity | Specificity | F-Score | Kappa Statistic |
|---|---|---|---|---|
| **Logistic Regression** | 0.39447 | 0.973333 | 0.51112 | 0.4602 |
| **CART Decision Tree** | 0.68387 | 0.95083 | 0.70790 | 0.65292 |
| **Random Forest** | 0.88061 | 0.98250 | 0.89235 | 0.87213 |
| **Ada Boosting** | 0.48278 | 0.96916 | 0.58403 | 0.52476 |

- Random Forest is performing better than CART Decision Tree, Ada boosting and Logistic Regression model in terms of all the measures i.e. sensitivity, specificity, F-score and Kappa statistic
- Ada boosting is only better than Logistic Regression model, but the specificity of Logistic Regression model is a little better than Ada boosting i.e. Logistic regression model is more accurately able to predict the customers who are not involved in earning manipulation compared to Ada boost
- However, more weightage is given sensitivity since the cost of not being able to predict an earning manipulator is higher than wrongly predicting a non-manipulator as an earning manipulator

Therefore, out of all the models Random Forest gives us the best results followed by CART Decision Tree, Ada boost and Logistic Regression

## Final Recommendation

If the end goal is to predict earning manipulators in the future we would recommend Random Forest, however if the client wants to understand the factors leading to earning manipulations Decision Tree would be a better choice. (Even though Decision Trees is performing poorly compared to Random Forest, Decision Trees are much easier to interpret when compared to Random Forest model)