Final Exam – Mukhtar A. Yusuf

12/11/2021

1. Objectives

The overall objective is to assess the role of foreign investors' inherent risk preferences against investment gains that could be explained by individuals' choice of specific decision factors and how they influence the propensity to engage in FDI based on their effects and predictive capacity.

Problem Statement

Understanding the complexity of FDI decision factors by analyzing their effects and using them to make FDI prediction

2. Approach

Overview of packages and dataset

- ❖ I used several relevant R packages in analyzing the dataset, most notable amongst them are "DPLYR, Class, lattice, ISLR, and proc". The packages are uploaded onto R from their various libraries. A total of …packages were required to run the codes.

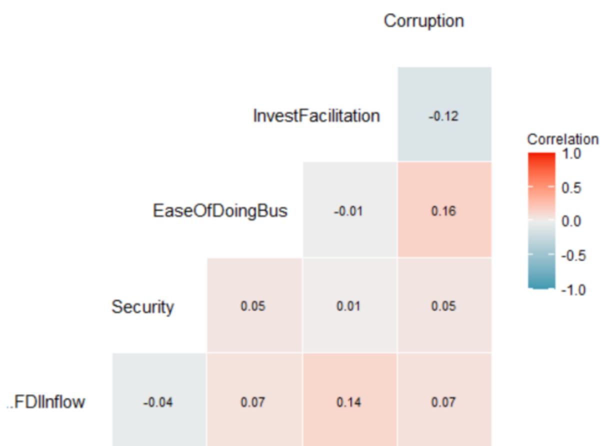
```
FinalProject<-read.csv("MukhtarMLProject.csv")
```

- ❖ The original dataset was a dataset gathered from a real online survey on Qualtrics from 260 foreign investors. The selected variables data were extracted from SPSS onto R in the form of excel.csv.
- ❖ There are no missing values in the dataset, hence, there was no need for us to transform them into numeric or impute the data points at any points. At the end of it, we have five columns in the data frame which include the vector (Foreign Direct Investment inflow) and four predictors all mean to ascertain their effect on vectors for investment decisions.

	ï..FDIInflow <int>	Security <dbl>	EaseOfDoingBus <dbl>	InvestFacilitation <dbl>	Corruption <dbl>
1	7	0.08251019	1.0681966	2.8	3.142857

Correlation

We plotted a correlation table among the various variables. There seems to be a considerable positive correlation between the FDI inflow decision and investment facilitation, and Corruption variables. Surprisingly, the Ease of doing business is also negatively correlated with investment facilitation services. Most pairwise correlations between predictors are generally low. (Santos, 2021)



Data normalization

The dataset was normalized using the method "range" for an efficient outcome

```
  ï..FDIInflow       Security       EaseOfDoingBus   InvestFacilitation   Corruption
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.000      Min.   :0.0000
 1st Qu.:0.2500   1st Qu.:0.4761   1st Qu.:0.5139   1st Qu.:0.650      1st Qu.:0.4286
```

Analyses and Outcomes

1. Linear Regression
2. Naive Bayes to make a prediction
3. Decision Tree for FDI decision

Linear Regression

- We created a linear model for all the variables against the FDI Inflow and displays a plot of the points using a Linear Model code "lm" and summarized the outcome.

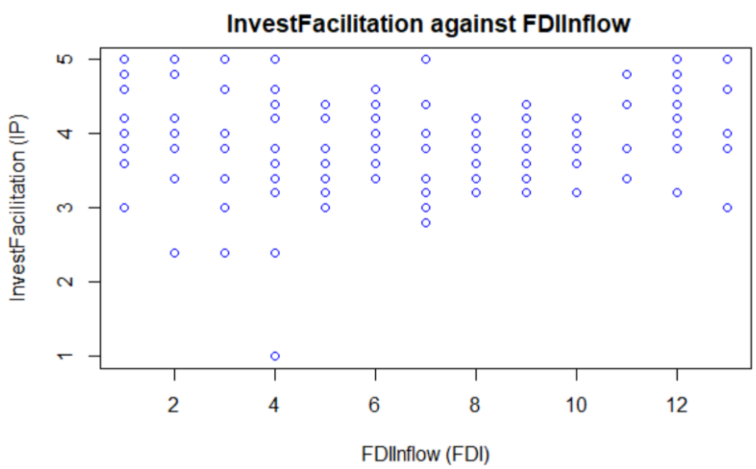
```
Modela = lm(FinalProject$ï..FDIInflow ~ FinalProject$EaseOfDoingBus +
FinalProject$InvestFacilitation + FinalProject$Corruption + FinalProject$Security, data =
FinalProject)
summary(Modela)
```

Result

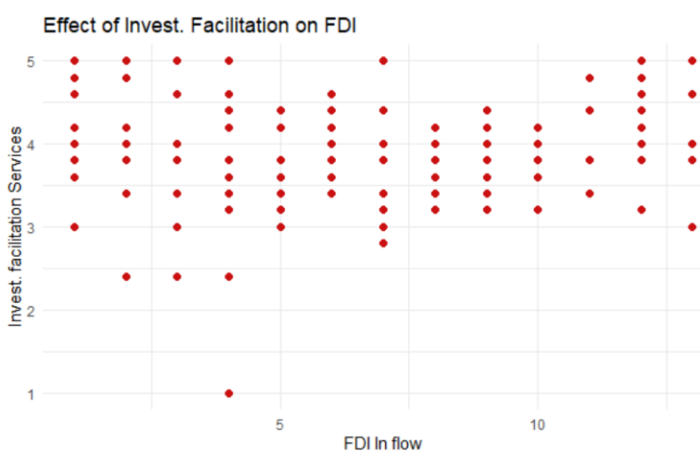
```
Call:
lm(formula = FinalProject$ï..FDIInflow ~ FinalProject$EaseOfDoingBus +
    FinalProject$InvestFacilitation + FinalProject$Corruption +
    FinalProject$Security, data = FinalProject)

Residuals:
    Min      1Q  Median      3Q     Max
-6.4034 -2.3982  0.0278  2.1127  7.0927

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                       0.4920     2.7276   0.180    0.857
FinalProject$EaseOfDoingBus       0.2119     0.2161   0.981    0.328
FinalProject$InvestFacilitation   0.8768     0.3774   2.323    0.021 *
FinalProject$Corruption           0.7494     0.5775   1.298    0.196
FinalProject$Security            -0.1548     0.1997  -0.775    0.439
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.182 on 245 degrees of freedom
Multiple R-squared:  0.03244,   Adjusted R-squared:  0.01664
F-statistic: 2.053 on 4 and 245 DF,  p-value: 0.08759
```

- The result indicates that high beta estimates of 0.877 and 0.75 for investment facilitation services and corruption, respectively. However, only investment facilitation has a significant effect (P-value) of 0.021 and with the highest t-value of 2.323.
- The Multiple R-squared is 3.2% on 245 degrees of freedom because we used very few variables from the original dataset

We plotted only investment facilitation services being the variables that proved to have a significant effect on FDI inflow.




```
ggplot(FinalProject) +
  aes(x = ï..FDIInflow, y = InvestFacilitation) + geom_point(shape = "circle", size = 2.25, colour =
"#CA1010") + labs( x = "FDI In flow",y = "Invest. facilitation Services", title = "Effect of Invest.
Facilitation on FDI") + theme_minimal()
```



Naive Bayes classifier

- We now use Naive Bayes on Ease of Doing business, investment facilitation, Corruption, and security select variables to predict Foreign Direct Investment Inflows.
- We first divide the data set into 80% training and 20% testing and validation

```
FinalProject_Index_Train<-createDataPartition(FinalProject$ï..FDIInflow, p=0.8, list=FALSE)
Train <-FinalProject[FinalProject_Index_Train,]
Test <-FinalProject[-FinalProject_Index_Train,]
```

We now run the Naive Bayes classifier model to predict FDI status on the test set

```
FinalProject_nb_model <-naiveBayes(ï..FDIInflow ~ EaseOfDoingBus + InvestFacilitation + Corruption +
Security, data = Train)
FinalProject_nb_model
```

Result

- ❖ The first part of the output above shows the ratios of default (yes) and default (no) in the training set (called a priori probabilities), followed by a table giving for each target class, mean, and standard deviation of the (sub-)variable. Classes 1,2,10, 11, and 13 have significant priori probabilities

```
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
         1          2          3          4          5          6          7          8 
0.04455446 0.04950495 0.07920792 0.14356436 0.07425743 0.09900990 0.09900990 0.12376238 
         9         10         11         12         13 
0.10891089 0.03960396 0.02475248 0.08910891 0.02475248 

Conditional probabilities:
    EaseOfDoingBus
Y          [,1]      [,2]
  1  -0.4670210 0.6990742
  2  -0.2444398 1.3590611
```

- ❖ The first part of the output above shows the ratios of default (yes) and default (no) in the training set (called a priori probabilities), followed by a table giving for each target class, mean, and standard deviation of the (sub-)variable. Also, note that the Naive Bayes algorithm assumes a normal distribution for the independent variables. By the rule of the use of categorical predictors (the independent variables have been converted to categorical), we now have the conditional probabilities p(X|Y) for every 13 classes of attribute level given the default status.

```
          Length Class  Mode     
apriori   13     table  numeric  
tables     4     -none- list     
levels    13     -none- character
isnumeric  4     -none- logical  
call       4     -none- call     
```

- ❖ Next, we Predicted the default status of the test data set and showed the confusion matrix of the classifier using the library "gmodels".

```
FinalProject_Predicted_Test_labels <-predict(FinalProject_nb_model, Test)
library(gmodels)
```

```
CrossTable(x=Test$ï..FDIInflow, y=FinalProject_Predicted_Test_labels, prop.chisq = FALSE)
```

Result

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  48


                  | FinalProject_Predicted_Test_labels
Test$ï..FDIInflow |          1 |          2 |          4 |          6 |          7 |          8 |
9 |         10 |         12 | Row Total |
------------------|-----------|-----------|-----------|-----------|-----------|-----------|-
---------|-----------|-----------|-----------|
                1 |          0 |          0 |          1 |          0 |          0 |          0 |
0 |          0 |          1 |          2 |
                  |      0.000 |      0.000 |      0.500 |      0.000 |      0.000 |      0.000 |
```

- It is sometimes useful to output the raw prediction probabilities rather than the predicted class. To do that, we use the raw option in the model.

```r
#Make predictions and return probability of each class
FinalProject_Predicted_Test_labels <-predict(FinalProject_nb_model,Test, type = "raw")
#show the first few values
head(FinalProject_Predicted_Test_labels)
```

Confusion matrix

- Next, we created a confusion matrix, it is useful to create a confusion matrix to determine the performance of the classification algorithm. A confusion matrix is a simple table displaying the number of true positives/negatives and false-positive/negatives, or in other words how often the algorithm correctly or incorrectly predicted the outcome. The confusion Matrix function is very helpful as not only does it display a confusion matrix, it calculates many relevant statistics alongside (Tricks, 2021)

```r
set.seed(123)
data <- data.frame(FinalProject = sample(c("True","False"), 250, replace = TRUE),
                        FinalProject_Predicted_Test_labels = sample(c("True","False"), 250,
replace = TRUE)
                        )
table(data$FinalProject_Predicted_Test_labels, data$FinalProject)
```

Result

```
Confusion Matrix and Statistics

          Reference
Prediction False True
     False    58   54
     True     69   69

               Accuracy : 0.508
                 95% CI : (0.4443, 0.5716)
    No Information Rate : 0.508
    P-Value [Acc > NIR] : 0.5253

                  Kappa : 0.0176

 Mcnemar's Test P-Value : 0.2068

            Sensitivity : 0.5610
            Specificity : 0.4567
         Pos Pred Value : 0.5000
         Neg Pred Value : 0.5179
             Prevalence : 0.4920
         Detection Rate : 0.2760
   Detection Prevalence : 0.5520
      Balanced Accuracy : 0.5088

       'Positive' Class : True
```

- Our results indicate that we misclassified a total of 127 cases out of 250. 69 as False Positives, and 58 as False Negatives. Interestingly, we classified a total of 123 cases of which 69 is True Positive and 54 is True Negative giving us an accuracy of 0.5253
- We have the confidence Interval of 95% that there is a 0.572 probability that the select predictors predicted the Foreign Direct Investment inflow based on the investment decision taken by the investor.
- Rows

```
               1          2          3          4          5           6          7
[1,] 0.012707195 0.10667584 0.09936510 0.55143899 0.03993762 0.005383761 0.13922021
[2,] 0.085990667 0.01203632 0.03841155 0.07736411 0.05251685 0.231223446 0.06266761
[3,] 0.068605817 0.06598634 0.06895985 0.24067328 0.01747661 0.047600512 0.04545547
[4,] 0.007971585 0.03291661 0.11106826 0.14239066 0.04443764 0.046565772 0.04351140
[5,] 0.005223793 0.02583836 0.05747524 0.16048161 0.15289880 0.112087242 0.05266614
[6,] 0.025644311 0.02915543 0.09296148 0.32056595 0.22605694 0.066814865 0.04411609
                8           9          10         11          12          13
[1,] 3.773758e-05 0.002856895 0.006204834 0.00152367 0.004877408 0.029770736
[2,] 4.609641e-02 0.162915247 0.088242387 0.01916609 0.092448988 0.030920314
[3,] 1.759180e-07 0.009099320 0.001749284 0.03716447 0.270704275 0.126524594
[4,] 1.364153e-05 0.047827416 0.001439325 0.24967161 0.265805494 0.006380601
[5,] 6.919233e-02 0.159617341 0.140300026 0.03641062 0.025087042 0.002721462
[6,] 1.409264e-03 0.090172581 0.042152019 0.03872901 0.012086962 0.010135096
```

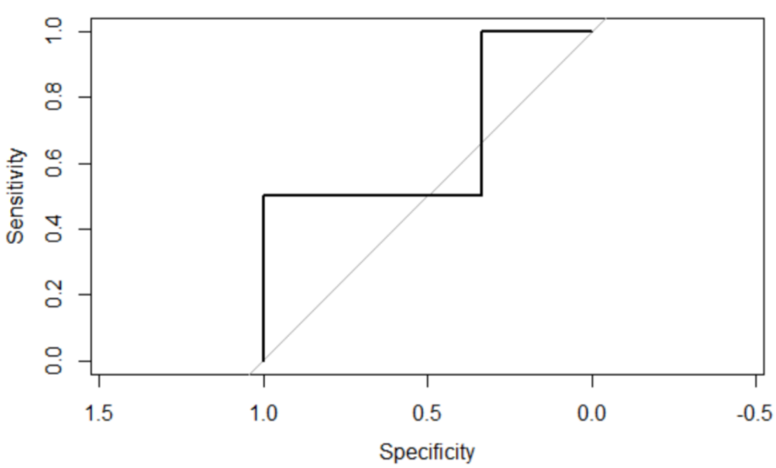
ROC Curves

- We can now output the ROC curves. we should remember that ROC curves plot sensitivity (true positive rate) versus (1 - specificity), which is (1 - TNR) or false positive rate
- We now move on to Passing the column of the predicted probabilities, that column contains the probability associated with 'yes'

```
roc(Test$ï..FDIInflow, FinalProject_Predicted_Test_labels[, 2])
plot.roc(Test$ï..FDIInflow, FinalProject_Predicted_Test_labels[, 2])
```

- ROC Curve Results

```
Call:
roc.default(response = Test$ï..FDIInflow, predictor =
FinalProject_Predicted_Test_labels[,     2])

Data: FinalProject_Predicted_Test_labels[, 2] in 3 controls
(Test$ï..FDIInflow 1) < 2 cases (Test$ï..FDIInflow 2).
Area under the curve: 0.6667
```



The AUC is 1. The ROC curve is also plotted, though note that the X-Axis is Specificity (True Negative Rate), rather than 1-Specificity (False Positive Rate). This function can also be thought of as a plot of the FDI as a function of the Type I Error of the decision rule. The area under the curve is 0.667. (Displayrr, 2018)

- Box-Cox Transformation
  We first illustrated the transformation of data using the Box-Cox transformation approach by Creating a Box-Cox Transformation Model

```
FinalProject_Box_Cox_Transform<-preProcess(FinalProject,method = "BoxCox")
FinalProject_Box_Cox_Transform
```

Result

Data transformation, and particularly the Box-Cox power transformation, is one of these remedial actions that may help to make data normal. The Lambda value indicates the power to which all data should be raised. To do this, the Box-Cox power transformation searches from Lambda = -5 to Lamba = +5 until the best value is found. The Box-Cox transformation tries to improve the normality of the residuals. Since that is the assumption of ANOVA as well. (StackStats, 2021)

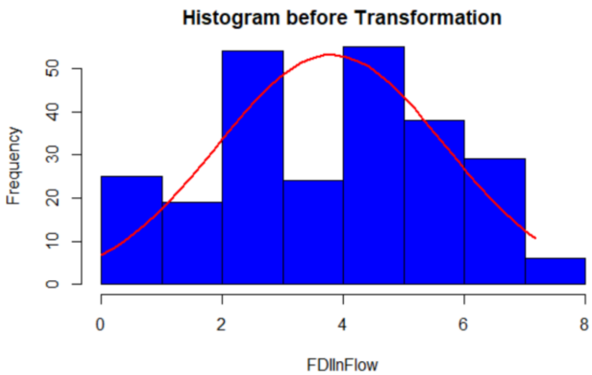
The lower and upper confidence levels (CLs) show that the best results for normality were reached with Lambda values between 0.7 and 1.8, the best value is 2.

```
Created from 250 samples and 3 variables

Pre-processing:
  - Box-Cox transformation (3)
  - ignored (0)

Lambda estimates for Box-Cox transformation:
0.7, 1.8, 2
```

Transformation

- We now apply transformation using:

```
FinalProject_Transformed=predict(FinalProject_Box_Cox_Transform, FinalProject)
y <-FinalProject_Transformed$ï..FDIInflow
h<-hist(y, breaks=10, col="blue", xlab="FDIInFlow",
   main="Histogram before Transformation")
xfit<-seq(min(y),max(y),length=40)
yfit<-dnorm(xfit,mean=mean(y),sd=sd(y))
yfit <- yfit*diff(h$mids[1:2])*length(y)
lines(xfit, yfit, col="red", lwd=2)
```



Result

The data before transformation seemingly assumes a relative normal frequency distribution of FDI inflow.

Alternatively, we used Hyper-tuning

- Hyper tuning: Hyperparameter tuning in the ML model can largely affect its predictive performance, thus it is important to set a suitable hyperparameter for the model. Traditionally, hyperparameter tuning in the ML model is usually performed by a trial-and-error process. Depending on how many hyperparameters exist in the ML model, this process can be very exhausting (dhikaaurel, 2021)s
- We used hyper tuning to analyze the dataset by dividing the data into 80% tests and the remaining for training before plotting.

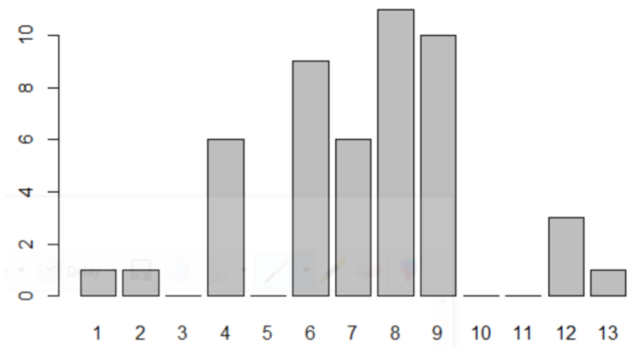
```
Index_Train<-createDataPartition(FinalProject$ï..FDIInflow, p=0.8, list= FALSE)
Train <-FinalProject[Index_Train,]
Test  <-FinalProject[-Index_Train,]
```

```
nb_model <-train(ï..FDIInflow ~ EaseOfDoingBus + InvestFacilitation + Corruption + Security, data =
Train, preProc = c("BoxCox", "center", "scale"))
# Predict the default status of test dataset
Predicted_Test_labels <-predict(FinalProject_nb_model,Test)
summary(Predicted_Test_labels)
plot(Predicted_Test_labels)
```

Result

The result predicted that 80% of the used tested dataset in 13 categories, categories 3,5,10, and 11 are zero while category 8 has the highest number of 11.

1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	0	6	0	9	6	11	10	0	0	3	1



- ❖ We now created a confusion matrix

```r
set.seed(123)
data <- data.frame(FinalProject = sample(c("True","False"), 250, replace =
TRUE),
Predicted_Test_labels = sample(c("True","False"), 250, replace = TRUE)
)
library(caret)
confusionMatrix(as.factor(data$Predicted_Test_labels),
as.factor(data$FinalProject), positive = "True")
```

- ❖
- ❖ Result

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|

Total Observations in Table:  48

                  | Predicted_Test_labels
Test$ï..FDIInflow |         2 |         4 |         6 |         7 |         8 |         9 |        10 |        12 | Row Total |
------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
                1 |         1 |         0 |         2 |         0 |         0 |         0 |         0 |         0 |         3 |
                  |     0.333 |     0.000 |     0.667 |     0.000 |     0.000 |     0.000 |     0.000 |     0.000 |     0.062 |
                  |     0.500 |     0.000 |     0.182 |     0.000 |     0.000 |     0.000 |     0.000 |     0.000 |           |
                  |     0.021 |     0.000 |     0.042 |     0.000 |     0.000 |     0.000 |     0.000 |     0.000 |           |

 -----------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
               13 |         0 |         0 |         0 |         0 |         0 |         0 |         0 |         1 |         1 |
                  |     0.000 |     0.000 |     0.000 |     0.000 |     0.000 |     0.000 |     0.000 |     1.000 |     0.021 |
                  |     0.000 |     0.000 |     0.000 |     0.000 |     0.000 |     0.000 |     0.000 |     0.200 |           |
                  |     0.000 |     0.000 |     0.000 |     0.000 |     0.000 |     0.000 |     0.000 |     0.021 |           |
 -----------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
     Column Total |         2 |         6 |        11 |         3 |        12 |         8 |         1 |         5 |        48 |
                  |     0.042 |     0.125 |     0.229 |     0.062 |     0.250 |     0.167 |     0.021 |     0.104 |           |
 -----------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
```

We have the same result as in the initial confusion matrix

```
Confusion Matrix and Statistics

          Reference
Prediction False True
     False    58   54
     True     69   69

               Accuracy : 0.508
                 95% CI : (0.4443, 0.5716)
    No Information Rate : 0.508
    P-Value [Acc > NIR] : 0.5253

                  Kappa : 0.0176

 Mcnemar's Test P-Value : 0.2068

            Sensitivity : 0.5610
            Specificity : 0.4567
         Pos Pred Value : 0.5000
         Neg Pred Value : 0.5179
             Prevalence : 0.4920
         Detection Rate : 0.2760
   Detection Prevalence : 0.5520
      Balanced Accuracy : 0.5088

       'Positive' Class : True
```

❖ Decision Tree

Decision Trees are versatile Machine Learning algorithms that can perform both classification and regression tasks. They are very powerful algorithms, capable of fitting complex datasets (guru99com, 2021) . Decision trees can produce a set of rules that can be easily interpreted and understood by humans. In a decision tree each internal node represents a "test" on an attribute, each branch represents the outcome of the test, band each leaf node represents one of the potential outputs of the tree.

```
FDI<- rpart(ï..FDIInflow ~ InvestFacilitation, data = Train) # only one independent variable
FDI<- rpart(ï..FDIInflow ~ ., data = Train) # Except one dependent variable, others are all
independent variables
```

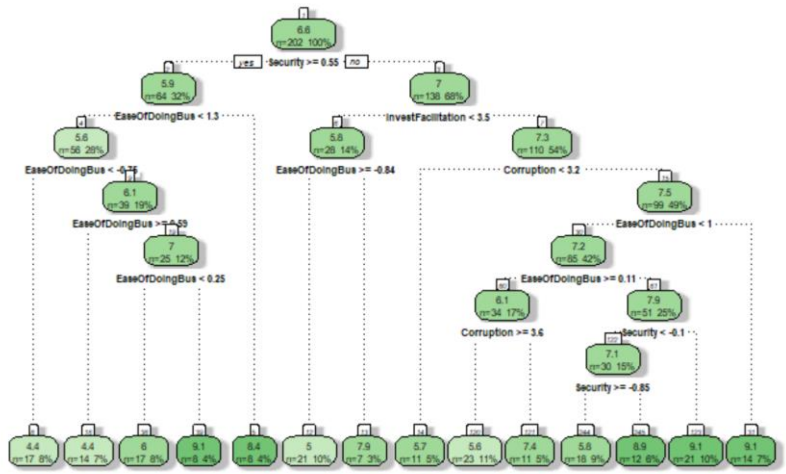
And then a decision tree with the minimal observation of 15

```
PredictFDI<-rpart(ï..FDIInflow ~ EaseOfDoingBus + InvestFacilitation + Corruption + Security, data =
Train, control = rpart.control(minbucket = 15))
fancyRpartPlot(PredictFDI)
```

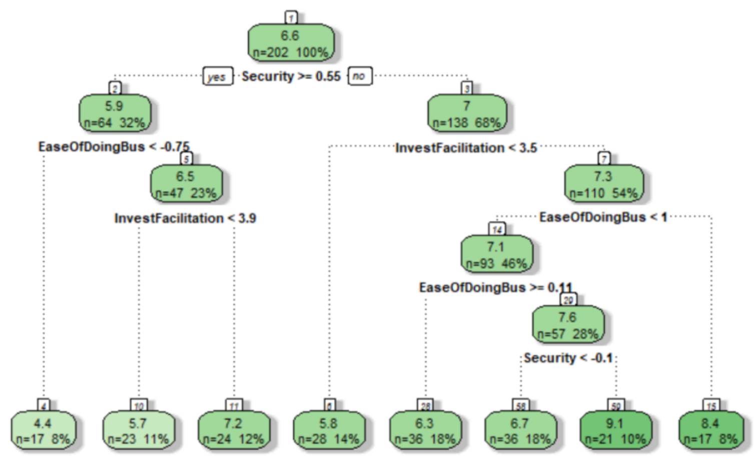
Result

The result indicates that:

1. At the top, it is the overall probability of FDI inflow. It shows the proportion of investment facilitation services that determine FDI decisions. 66% percent of FDI inflow represents investment facilitation.
2. This node asks whether the security factor influences FDI inflow. If yes, then you go down to the root's left child node (depth 2). 32 percent are Ease of doing business with n number of 64 and determining the probability of 59 percent.
3. In the second node, you consider if the EODB is less than 13%. If yes, then the chance of determining FDI inflow is 56 percent.
4. We could keep on going like that to understand what specific factors impact the FDI inflow decisions.
5. We may note that one of the many qualities of Decision Trees is that they require very little data preparation. In particular, they don't require feature scaling or centering. (guru99com, 2021)



Rattle 2021-Dec-11 13:04:11 Mukht



Rattle 2021-Dec-11 13:20:55 Mukht

Predict FDI Inflow using Decision Tree

- ❖ We now predicted the vector (FDI Inflow) and produce the required table and the summary of the output. We predict which factors are more likely to determine FDI from the test set. It means we would know among those 250 foreign investors which one will invest or not.

```
library(rpart)
library(rpart.plot)
PredictFDIUnknown<-predict(PredictFDI, Test)
table_FDI <-table(Test$ï..FDIInflow, PredictFDIUnknown)
summary(table_FDI)
table_FDI
```

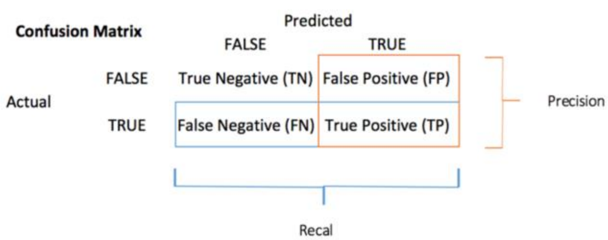
Result

- ❖ The model correctly predicted that there is an insignificant probability of 0.5112 for all combined independent factors that a foreign investor would decide to invest when the degree of freedom is 84.

```
Number of cases in table: 48
Number of factors: 2
Test for independence of all factors:
	Chisq = 82.96, df = 84, p-value = 0.5115
	Chi-squared approximation may be incorrect
    PredictFDIUnknown
     4.41176470588235 5.65217391304348 5.75 6.30555555555556 6.72222222222222 7.25 8.35294117647059 9.09523809523809
  1                 1                0    0                2                0    0                0                0
  2                 0                0    0                0                1    0                1                1
  3                 0                0    1                0                1    0                0                1
  4                 0                0    1                2                3    0                0                0
  5                 1                0    1                1                0    0                0                1
  6                 0                0    0                2                1    0                1                0
  7                 0                1    1                0                1    0                1                1
  8                 0                0    1                0                0    0                1                3
  9                 0                0    0                1                4    1                0                0
  10                0                1    0                1                0    0                0                0
  11                0                0    1                0                0    0                0                0
  12                0                1    0                0                2    1                0                1
  13                0                0    0                1                0    0                0                0
```

Measure performance

- ❖ We computed an accuracy measure for the classification task with the confusion matrix
- ❖ The confusion matrix is a better choice to evaluate the classification performance. The general idea is to count the number of times True instances are classified are False. (guru99com, 2021)



$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy Test

```
accuracy_Test <- sum(diag(table_FDI)) / sum(table_FDI)
print(paste('Accuracy for test', accuracy_Test))
```

Result

```
[1] "Accuracy for test 0.166666666666667"
```

- ❖ performance measurement- confusion matrix: We tried to tune the parameters and see if we could improve the model over the default value. As a reminder, you need to get an accuracy higher than 0.17

```
control <- rpart.control(minsplit = 4,
    minbucket = round(5 / 3),
    maxdepth = 3,
    cp = 0)
tune_fit<-rpart(ï..FDIInflow ~., data = Train, method = "class", control = control)
summary(tune_fit)
```

```
accuracy_tune <- function(fit) {
    PredictFDIUnknown <- predict(FDI, Test, type = 'class')
    table_FDI <- table(FinalProject$ï..FDIInflow, PredictFDIUnknown)
    accuracy_Test <- sum(diag(table_FDI)) / sum(table_FDI)
    accuracy_Test
}
```

Result

This improves and classifies the variable' importance

```
Call:
rpart(formula = ï..FDIInflow ~ ., data = Train, method = "class",
    control = control)
  n= 202

          CP nsplit rel error   xerror       xstd
1 0.02890173      0 1.0000000 1.028902 0.02658238
2 0.01445087      2 0.9421965 1.017341 0.02751194
3 0.01156069      4 0.9132948 1.023121 0.02705421
4 0.00000000      7 0.8786127 1.011561 0.02795626

Variable importance
          Security     EaseOfDoingBus         Corruption InvestFacilitation
                30                 25                 24                 21

Node number 1: 202 observations,    complexity param=0.02890173
  predicted class=4   expected loss=0.8564356  P(node) =1
    class counts:     9    10    16    29    15    20    20    25    22     8     5    18
5
   probabilities: 0.045 0.050 0.079 0.144 0.074 0.099 0.099 0.124 0.109 0.040 0.025 0.089
0.025
```

4. Summary of Results

a. Linear regression

- There seems to be a considerable positive correlation between the FDI inflow decision and investment facilitation, and Corruption variables. Surprisingly, the Ease of doing business is also negatively correlated with investment facilitation services. Most pairwise correlations between predictors are generally low.
- The result indicates that high beta estimates of 0.877 and 0.75 for investment facilitation services and corruption, respectively. However, only investment facilitation has a significant effect (P-value) of 0.021 and with the highest t-value of 2.323.
- The Multiple R-squared is 3.2% on 245 degrees of freedom because we used very few variables from the original dataset

b. FDI prediction based on Naïve bayes classifier

- Our results indicate that we misclassified a total of 127 cases out of 250. 69 as False Positives, and 58 as False Negatives. Interestingly, we classified a total of 123 cases of which 69 is True Positive and 54 is True Negative giving us an accuracy of 0.5253
- We have the confidence Interval of 95% that there is a 0.572 probability that the select predictors predicted the Foreign Direct Investment inflow based on the investment decision taken by the investor.
- ROC Curve: Area under the curve is 0.667, this function can also be thought of as a plot of the FDI as a function of the Type I Error of the decision rule. Now because the AUC is> 0.5, then the classifier has been able to distinguish between Positive and Negative class points. This also means that the classifier is predicting a random class or constant class for all the data points.
- Box-Cox Transformation: The lower and upper confidence levels (CLs) show that the best results for normality were reached with Lambda values between 0.7 and 1.8, the best value is 2.
- Tested data before transformation: The data before transformation seemingly assumes a relative normal frequency distribution of FDI inflow.

- Hyper-tuning: The result predicted that 80% of the used tested dataset with 13 categories, the classes 3,5,10, and 11 are zero whereas classes 1,2,4,6,7,8,9,12, and 13 are non-zero with category 8 having the highest number of 11. We have the same result as in the initial confusion matrix.

c. FDI prediction based on Decision Tree

- A node shows the proportion of investment facilitation services that determine FDI decision as 66% percent of the overall probability of FDI inflow.
- Another node asks whether the security factor influences FDI inflow, the answer is yes, and it also suggests that 32% of FDI inflow are determined by the Ease of doing business with n number of 64 and with determining the probability of 0.59.
- In a different node, we considered if the EODB is less than 13%. If yes, then the chance of determining FDI inflow is 56 percent, the answer is yes.
- The model correctly predicted that there is an insignificant probability of 0.5112 for all combined independent factors that a foreign investor would decide to invest when the degree of freedom is 84.
- Performance measurement: the accuracy measure for classification tasks with the confusion matrix is relatively low.

5. Conclusions

Our findings indicate that the perception of investment facilitation services has a profound direct positive effect on individual investors' FDI decisions. And that the same factor could be used to effectively predict an investor's decision using Naïve Bayes and decision trees, respectively.

Given the above inferences, I would suggest the following:

- With sufficient capacity building of the Investment Promotion Agency's (IPA) employees, it is envisaged that the employees could perform much better in other areas of services that proved to have weak effects on investor's decisions so that they could attain their full potentials as promoters and facilitators of investment service providers.
- It is now clear from the regression output that a security factor has a negative effect on FDI decision with an insignificant p-value, though. The ease of doing and fighting against corruption must be enhanced to attract the much-needed potential investors in sub-Saharan Africa.
- Finally, the inferences established from the analyses should be used to create knowledge for IP agencies and to utilize the experience for a corporate policy framework that would enhance performance in preparation against the dynamic and evolving FDI competition for economic development.

References

dhikaaurel. (2021). *Genetic Algorithm in R: Hyperparameter Tuning*. Retrieved from https://towardsdatascience.com/genetic-algorithm-in-r-hyperparameter-tuning-5fc6439d2962

Displayrr. (2018). What is a ROC Curve and How to Interpret It. Retrieved from https://www.displayr.com/what-is-a-roc-curve-how-to-interpret-it/

guru99com. (2021). Decision Tree in R | Classification Tree & Code in R with Example. In: guru99com.

Santos, J. D. (2021). RPubs - Search Methods for Hyperparameter Tuning in R. In.

StackStats. (2021). Estimating Lambda for Box Cox transformation for ANOVA. Retrieved from https://stats.stackexchange.com/questions/10095/estimating-lambda-for-box-cox-transformation-for-anova

Tricks, D. (2021, 2021-04-13). Confusion matrix in R: two simple methods - Data Tricks. Retrieved from https://datatricks.co.uk/confusion-matrix-in-r-two-simple-methods