Learning Analytics for MOOC

Exploratory Data Analysis Document and Classification

Group No. : DS707-2017-10
ChellaPriyadharshini M (**MT2016041**)
Daminee Sao (**MT2016045**)
Jyotsana (**MT2016068**)
Kanika Narang (**MT2016069**)
Tehreem Ansari (**MT2016145**)

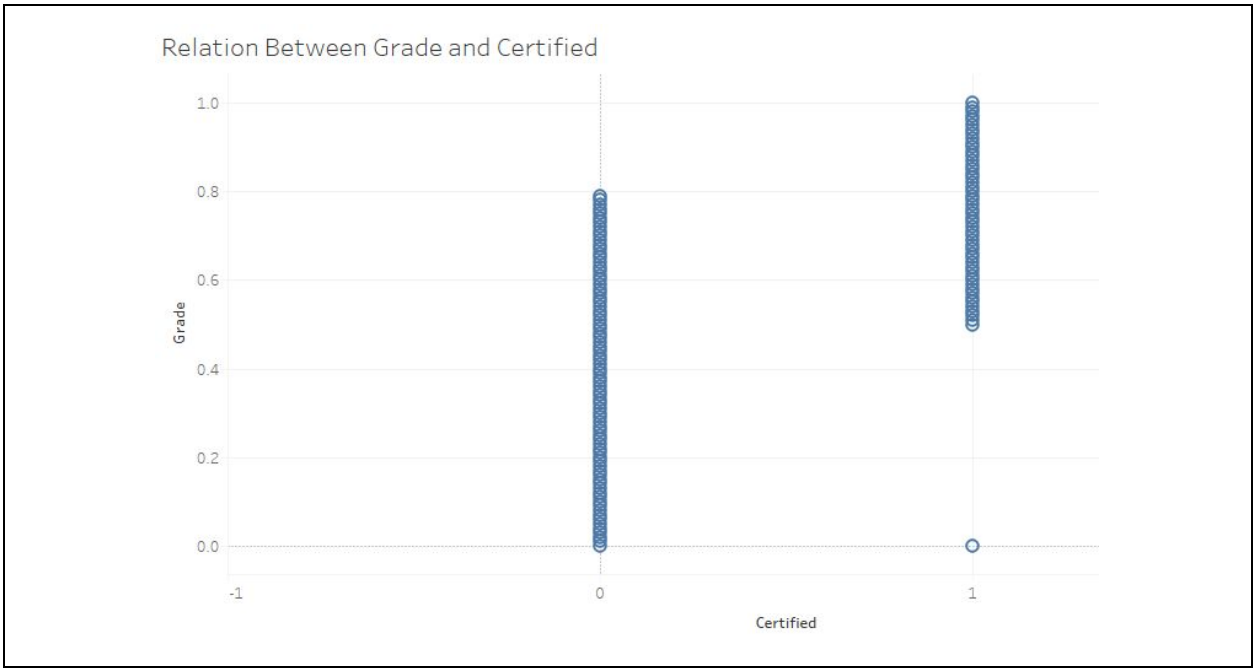
Done By : Kanika Narang (MT2016069)

Description

Analysis is done for **'Instructors'** as stakeholder. From Below analysis instructor can clearly judge which student should get certified. How students are interacting with the course and how it affects the grade of the students. It will help the instructor to improve the course and impart quality knowledge.

Tableau visualization [(click Here to View the tableau report)]

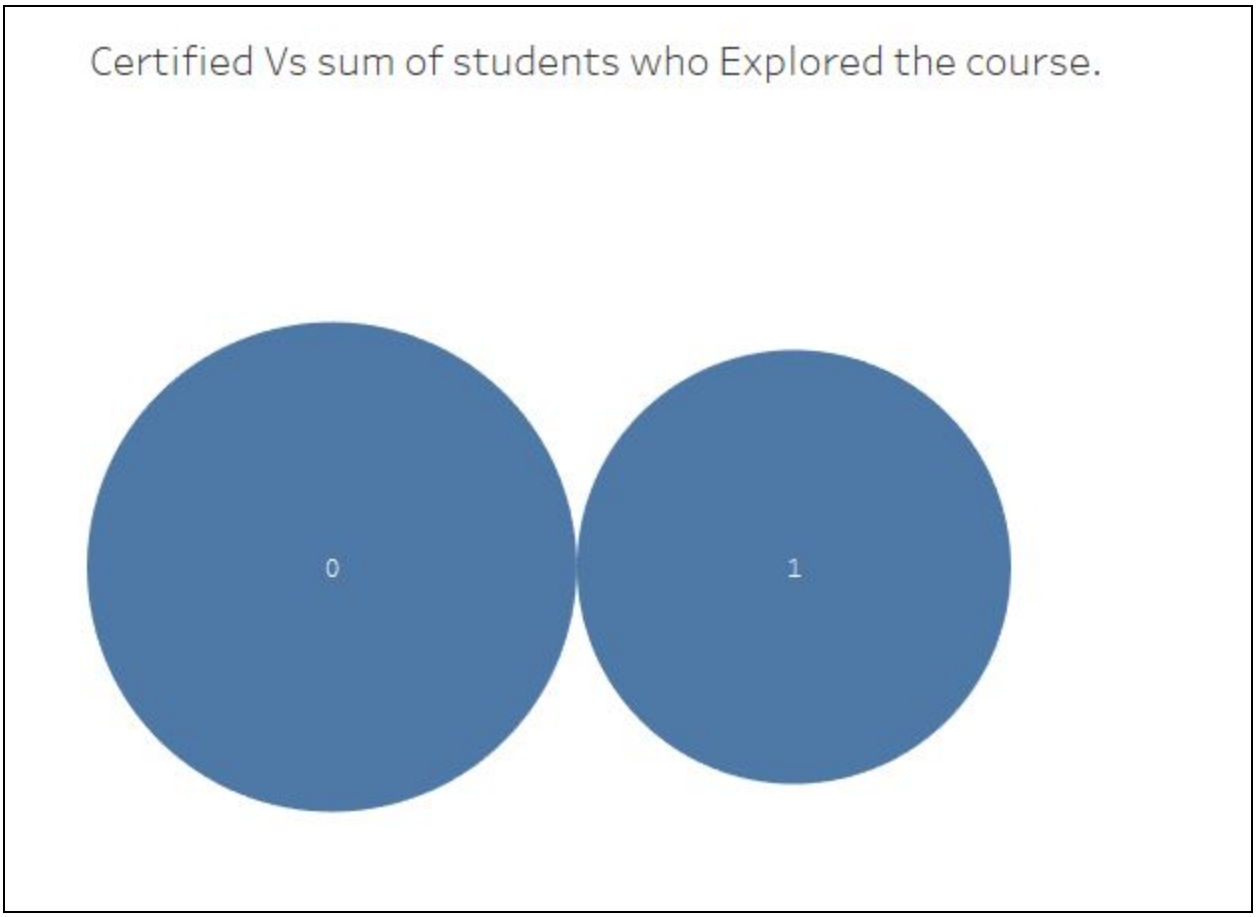
1. **Grade Vs Certified :** This helped us explore that granting of certificate does not only depends on Grade. There may be other factors responsible in granting the certificate. Below graph shows that the students who are granted the certificate must have grade greater than 0.5, but that does not mean that every student having grade greater than 0.5 will get certified.



2. **Certified Vs Number of students who Explored the course:** This plot shows the number of students getting certified who explored the course is a little less than the students who didn't get the certified even after exploring the course.

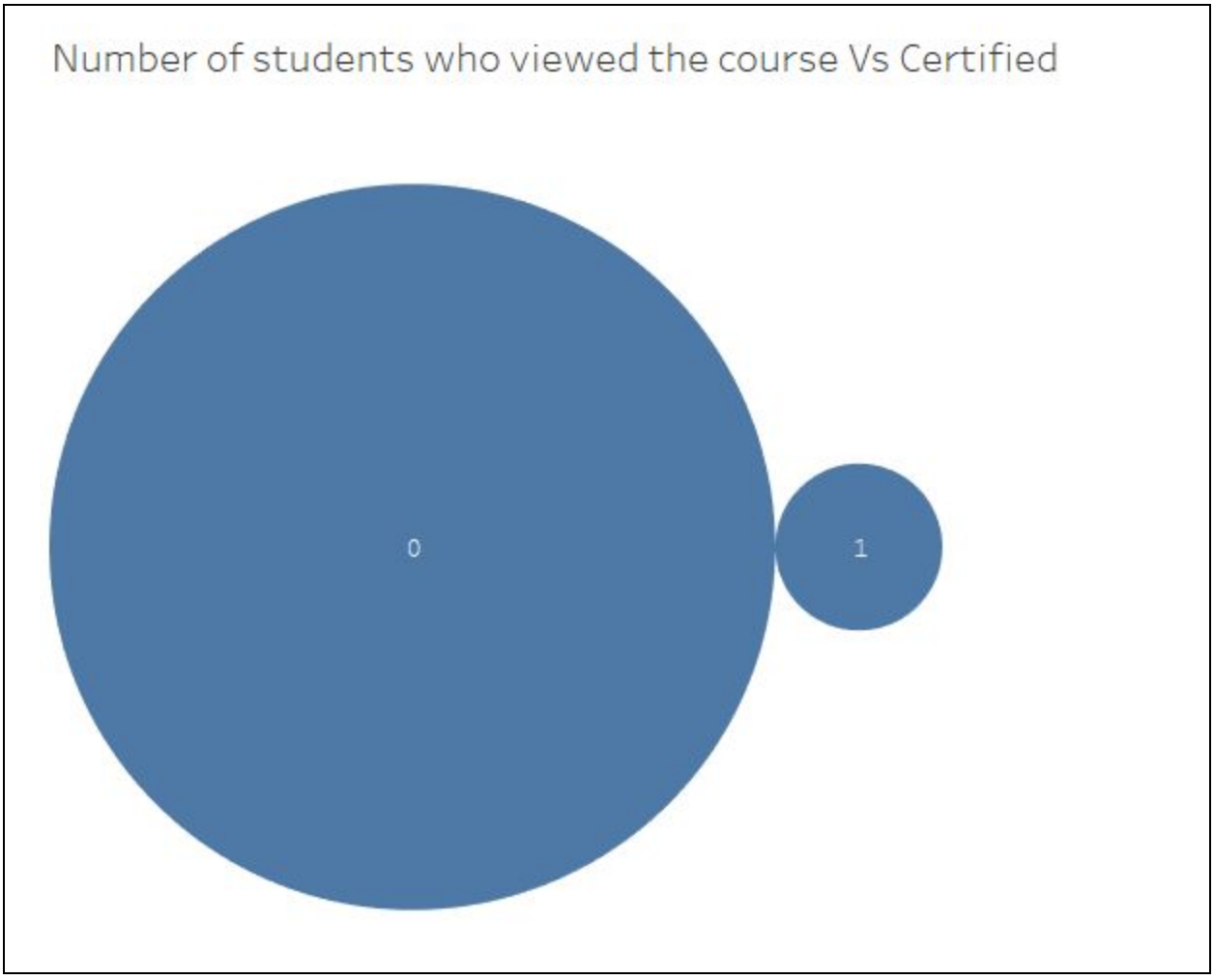
No. students	Certified	Not certified
Explored	14,233	18,122

Below the text shows the category of getting certified (1: certified and 0: Not certified) and size shows the number of students who explored the course.



3. **Certified Vs Number of students who Viewed the course:** This graph shows the number of students getting certified who viewed the course is much less than the students who didn't get the certified even after viewing the course.

No. students	Certified	Not certified
Explored	14,889	281,431

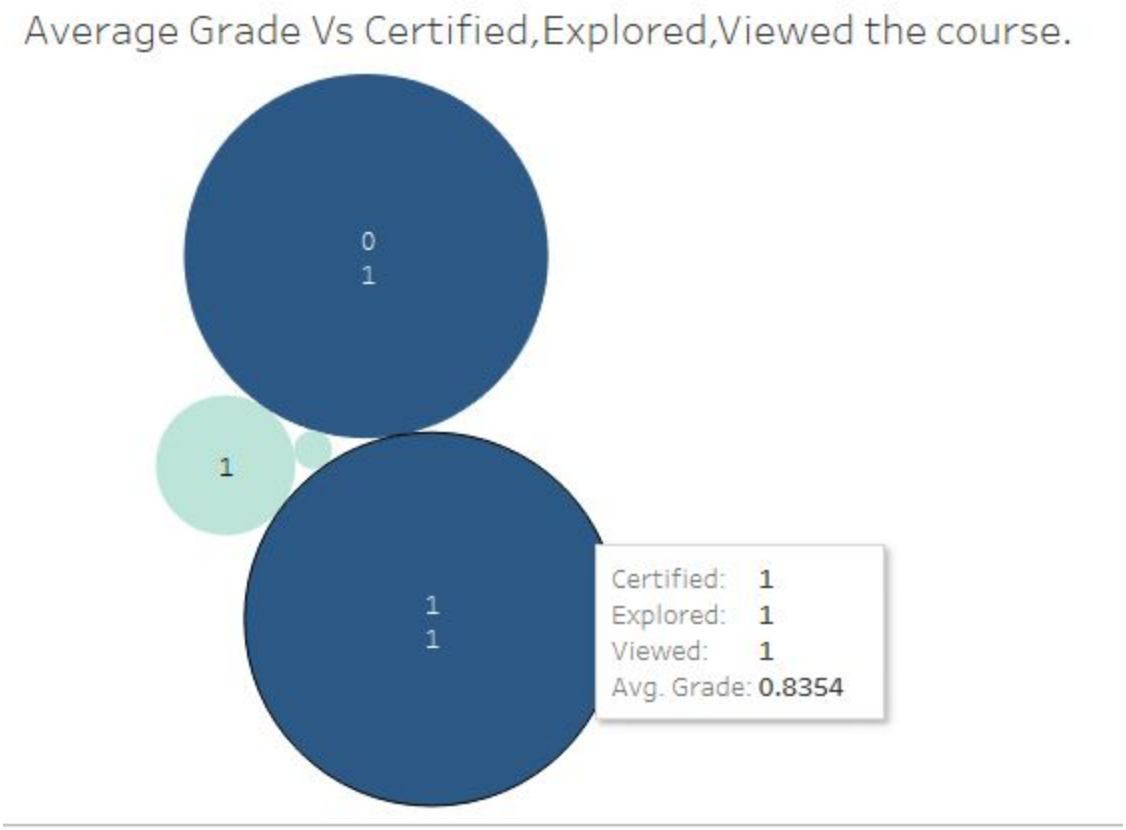


Below the text shows the category of getting certified (1: certified and 0: Not certified) and size shows the number of students who explored the course.

4. **Average Grade Vs Viewed,Explored and certified the course:** This graph shows the average grade the student gets if he has viewed, explored the course.

Viewed	Explored	Certified	Avg Grade
1	0	1	0.794
1	1	1	0.835
1	1	0	0.117
1	0	0	0.0088

Inference from this is student gets more grade if student has explored and viewed the course and thus get certified.



Dark Blue color shows certified (1)
Light Blue color shows Not certified (0)
Size shows Avg Grade.
First text shows Explored
Second text shows Viewed

Classification

Problem Statement 1: Classifying which student will get certified or not.

Stakeholders- Instructor: Instructor could now the estimated number of students who end up getting certified, thus according to the predicted performance of students, instructor can modify the course's difficulty level. For example. If predicted number of students who will get certified is less, then instructor can make content more interactive and understandable.

Data Preparation for classification

1. Converting column **'certified'** to factor because it is label column.

```
> #data preparation for classification
> # column certified is an integer column so need to convert it to factor

> class(student_data_dataverse$certified)
[1] "integer"
> student_data_dataverse['certified']<- as.factor(student_data_dataverse$
certified)
> class(student_data_dataverse$certified)
[1] "factor"
```

2. Removing column **'YoB'.**

```
> # Since we have calculated the column age so YOB not needed
> # hence dropping YOB also column last_event_DI contains 166085
> student_data_dataverse<- student_data_dataverse[-11]
```

3. Removing rows with NA's in column **'last_event_DI'.**

```
> # removing th rows with NA in column last_event_DI
> row_no<- which(is.na(student_data_dataverse$last_event_DI))
> student_data_dataverse <- student_data_dataverse[- row_no, ]
```

4. Replacing the NA's in **'grade'** column with 0 since, the rows with NA's in 'grade' column also have 0 in the column 'certified' thus it means students with NA grade are not certified.

```
> # Since we can see that all the NA's in grade column is because these s
tudents have not been certified
> # this could because of many reasons like student din't appeared for ce
rtification.
> sum(is.na(student_data_dataverse$grade))
[1] 42383
> sum(is.na(student_data_dataverse$grade) & student_data_dataverse$certif
ied ==0)
[1] 42383
> # Thus replacing NA's in grade column with 0.
> student_data_dataverse$grade<- ifelse(is.na(student_data_dataverse$grad
e),0,student_data_dataverse$grade)
```

5. Converting column '**userid_DI**' to character because random forest does not support factor with more than 53 categories.

```
> # converting userid_DI to character because random forest does not supp
ort
> # factor with more than 53 categories.
> student_data_dataverse$userid_DI <- as.character(student_data_dataverse
$userid_DI)
```

6. Converting **'start_time_DI**' and **'last_event_DI'** to date.

```
> # converting start_time_DI to date
> student_data_dataverse$start_time_DI<- as.Date(student_data_dataverse$s
tart_time_DI)
> # converting last_event_DI to date
> student_data_dataverse$last_event_DI<- as.Date(student_data_dataverse$l
ast_event_DI)
> str(student_data_dataverse)
```

Code for doing classification

I will be using Random forest with number of trees = 100 and label column is 'certified'

1. Loading Data and Libraries:

```
student_data_dataverse<- read.csv('/home/kanika/Documents/3rd_Sem/gi
student_data_dataverse<- student_data_dataverse[-c(1,2,3)]
source('helper_functions.R')
library(randomForest)
library(e1071)
library(caret)
library(ggplot2)
```

2. Splitting Data into Training and Testing data using 60-40 rule.

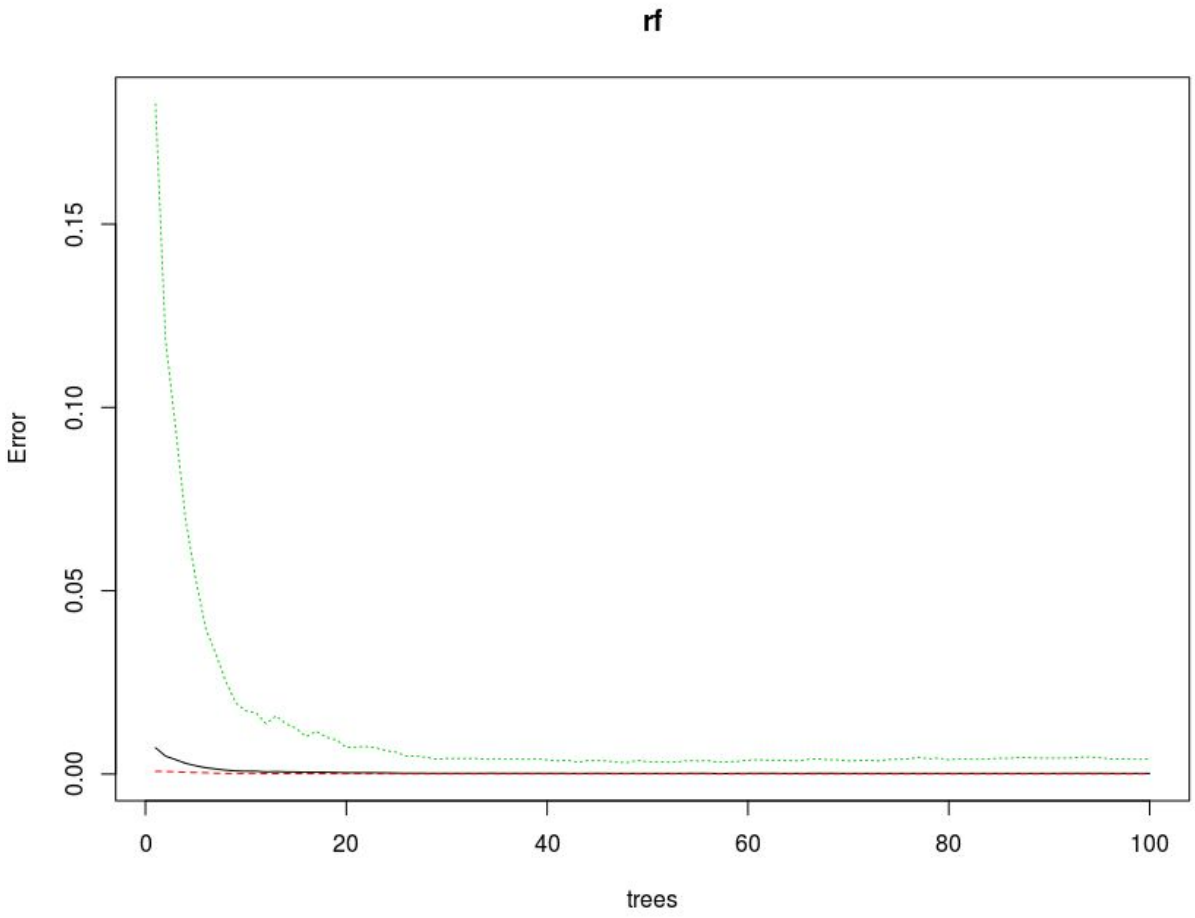
```
> # i am spliting data into training and testing with 60-40 rule
> #Create data for training
> sample.ind = sample(2,
+                     nrow(student_data_dataverse),
+                     replace = T,
+                     prob = c(0.6,0.4))
> data.dev = student_data_dataverse[sample.ind==1,]
> data.val = student_data_dataverse[sample.ind==2,]
```

3. Let's Look the proportion of students certified in the training and testing datasets. (0 means Not certified and 1 means certified)

```
> # looking at the split percentage for the certified students
> # Original Data
> table(student_data_dataverse$certified)/nrow(student_data_dataverse)

        0         1
0.9642882 0.0357118
> # Training Data
> table(data.dev$certified)/nrow(data.dev)

         0          1
0.96430699 0.03569301
> # Testing Data
> table(data.val$certified)/nrow(data.val)

         0          1
0.96426002 0.03573998
```

4. Fitting the random forest model.

```
> #Fit Random Forest Model
> rf = randomForest(certified ~ .,
+                   ntree = 100,
+                   data = data.dev[,-5])
> plot(rf)
> print(rf)

Call:
 randomForest(formula = certified ~ ., data = data.dev[, -5],      ntree
= 100)
               Type of random forest: classification
                     Number of trees: 100
No. of variables tried at each split: 4

        OOB estimate of  error rate: 0.02%
Confusion matrix:
       0    1  class.error
0 241672   13 5.378902e-05
1     37 8863 4.157303e-03
```



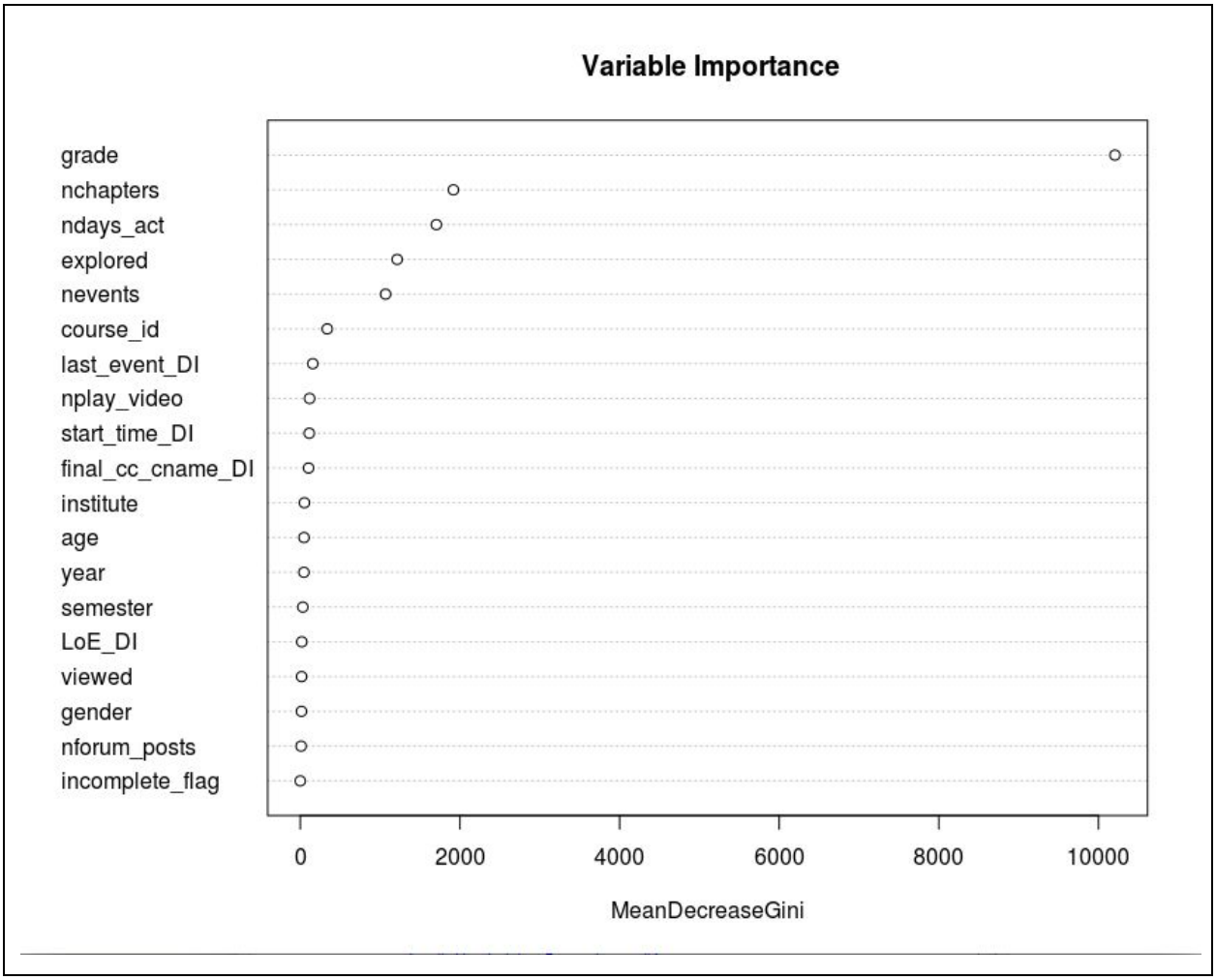
The plot shows that after the 25 trees there is not much changes in terms of error. It fluctuates a bit but not to a large degree. Black curve shows OOB , green curve shows o class of certified (Not certified) and red curve shows the 1 class of certified (certified).

The confusion matrix shows that there are 13 false positive and 37 false negatives. It's OOB estimate of error rate is 0.02% .

5. Plotting variable importance: This plot indicates what variables had the greatest impact in the classification model.

According to the below graph we can see that 'grade' is the most important attribute (or variable) according to the Mean decrease gini values. The mean decrease in Gini coefficient is a measure of how each variable contributes to the homogeneity of the nodes and leaves in the resulting random forest. Each time a particular variable is used to split a node, the Gini coefficient for the child nodes are calculated and compared to that of the original node.

```
> #Variable Importance
> var.imp = data.frame(importance(rf,
+                                 type=2))
> # make row names as columns
> var.imp$Variables = row.names(var.imp)
> print(var.imp[order(var.imp$MeanDecreaseGini,decreasing = T),])
                  MeanDecreaseGini         Variables
grade                 1.020678e+04             grade
nchapters             1.918302e+03         nchapters
ndays_act             1.705175e+03         ndays_act
explored              1.213775e+03          explored
nevents               1.068839e+03           nevents
course_id             3.380197e+02         course_id
last_event_DI         1.568406e+02     last_event_DI
nplay_video           1.167271e+02       nplay_video
start_time_DI         1.125564e+02     start_time_DI
final_cc_cname_DI     1.037807e+02 final_cc_cname_DI
institute             5.216411e+01         institute
age                   4.698230e+01               age
year                  4.592897e+01              year
semester              3.216788e+01          semester
LoE_DI                1.926298e+01            LoE_DI
viewed                1.659971e+01            viewed
gender                1.527788e+01            gender
nforum_posts          1.055042e+01      nforum_posts
incomplete_flag       6.319683e-04   incomplete_flag
```



6. Now it's time to see how the model works with data it has not seen before – making predictions on the test data.

```
> # Now predicting on test data
> # Predicting response variable
> data.val$predicted.response <- predict(rf ,data.val)
>
> # Create Confusion Matrix
> print(
+   confusionMatrix(data=data.val$predicted.response,
+                   reference=data.val$certified,
+                   positive='1'))
Confusion Matrix and Statistics

          Reference
Prediction      0      1
         0 160343     10
         1      4   5979

               Accuracy : 0.9999
                 95% CI : (0.9999, 1)
    No Information Rate : 0.964
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9988
 Mcnemar's Test P-Value : 0.1814

            Sensitivity : 0.99833
            Specificity : 0.99998
         Pos Pred Value : 0.99933
         Neg Pred Value : 0.99994
             Prevalence : 0.03601
         Detection Rate : 0.03595
   Detection Prevalence : 0.03597
      Balanced Accuracy : 0.99915

       'Positive' Class : 1
```

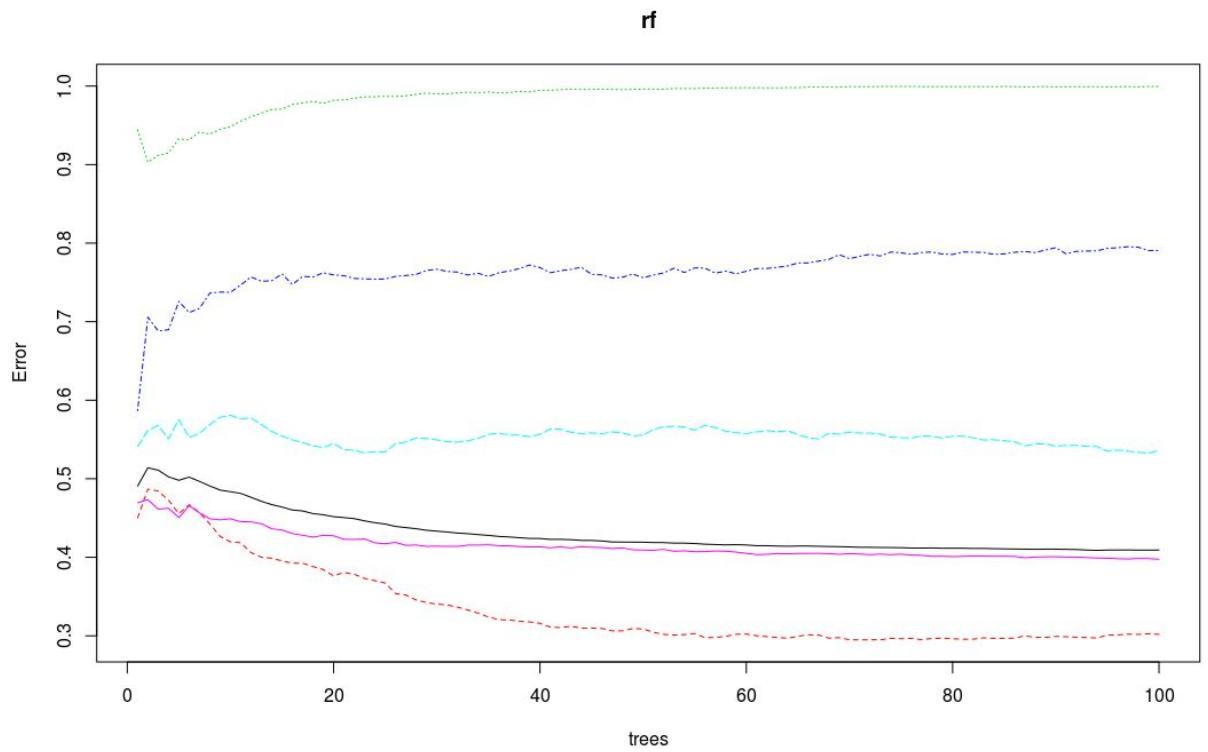
As we can see we get only 4 false negative and 10 false positive. Accuracy is around 99%.

Problem Statement 2: Classifying students LoE (level of education)

Stakeholders- Instructor: instructor could classify students based on their LoE, then instructor could provide course based on education levels and hence improve the credibility of the course.

1. Classification Model is build using random forest with 100 trees, OOB estimate of error is 40.93 % which is not good.

```
> #Fit Random Forest Model
> rf = randomForest(LoE_DI ~ .,
+                   ntree = 100,
+                   data = data.dev[,-5])
> plot(rf)
> print(rf)

Call:
 randomForest(formula = LoE_DI ~ ., data = data.dev[, -5], ntree = 100)
               Type of random forest: classification
                     Number of trees: 100
No. of variables tried at each split: 4

        OOB estimate of  error rate: 40.93%
Confusion matrix:
                    Bachelor's Doctorate Less than Secondary Master's Secondary
Bachelor's               76969         0                  36    18547     14693
Doctorate                 1577         3                   0     4502        62
Less than Secondary        191         0                1239        4      4483
Master's                 28408         2                   2    25025       514
Secondary                24615         0                 358     4619     44842
                    class.error
Bachelor's            0.3018368
Doctorate             0.9995117
Less than Secondary   0.7906033
Master's              0.5361532
Secondary             0.3975603
```

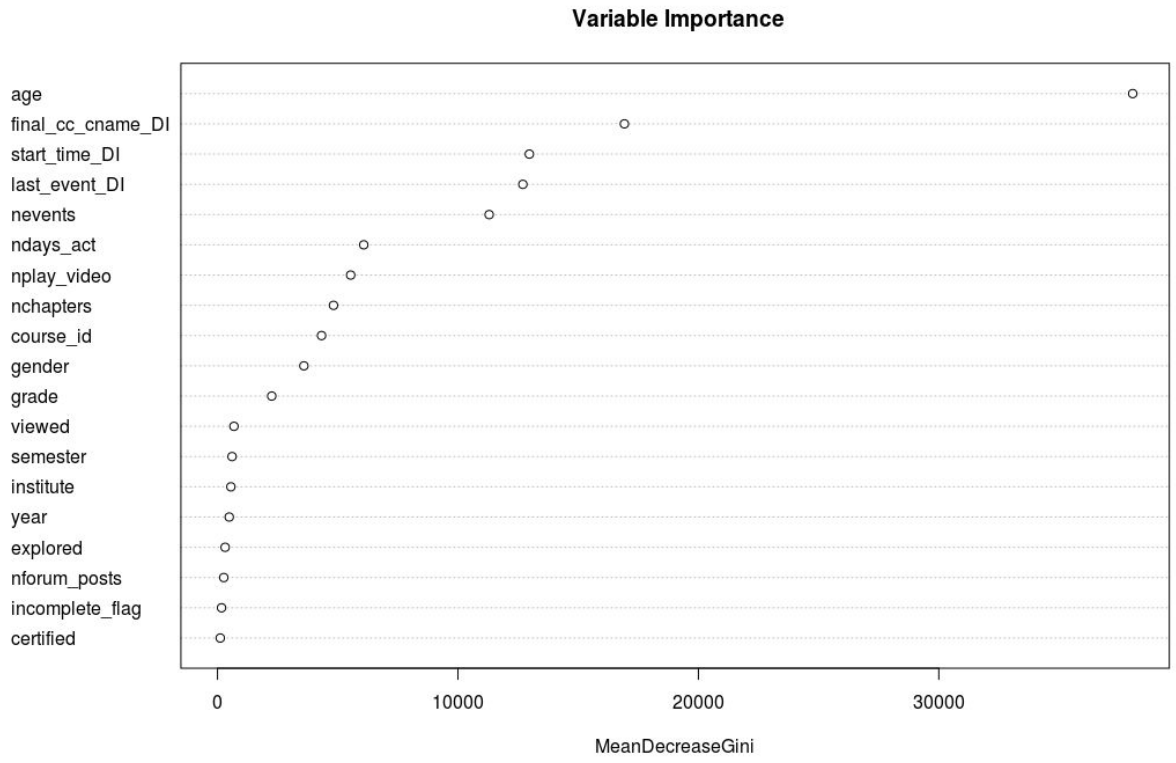


Plot shows the error to classify different categories of LoE_DI.
Black: OOB
Green:Doctorate
Blue: Less than Secondary
Light Blue:Master's
Pink: Secondary
Red: Bachelar's

2. Plotting variable importance: This plot indicates what variables had the greatest impact in the classification model.

It shows age is the most important variable to classify the LoE_DI.

```
> # Variable Importance
> varImpPlot(rf,
+            sort = T,
+            main=" Variable Importance")
> #Variable Importance
> var.imp = data.frame(importance(rf,
+                                 type=2))
>
> # make row names as columns
> var.imp$Variables = row.names(var.imp)
> print(var.imp[order(var.imp$MeanDecreaseGini,decreasing = T),])
                  MeanDecreaseGini         Variables
age                    38063.9405               age
final_cc_cname_DI      16923.2315 final_cc_cname_DI
start_time_DI          12970.6996     start_time_DI
last_event_DI          12700.3830     last_event_DI
nevents                11298.3637           nevents
ndays_act               6082.2337         ndays_act
nplay_video             5539.0008       nplay_video
nchapters               4825.1540         nchapters
course_id               4327.7040         course_id
gender                  3593.3935            gender
grade                   2255.8696             grade
viewed                   684.8114            viewed
semester                 605.6848          semester
institute                556.8323         institute
year                     487.9516              year
explored                 317.4979          explored
nforum_posts             261.2945      nforum_posts
incomplete_flag          168.4753   incomplete_flag
certified                115.2103         certified
```



3. Applying the model on the test data set. Accuracy is around 60%.

```
> # Now predicting on test data
> # Predicting response variable
> data.val$predicted.response <- predict(rf ,data.val)
>
> # Create Confusion Matrix
> print(
+   confusionMatrix(data=data.val$predicted.response,
+                   reference=data.val$LoE_DI,
+                   ))
```

```
Confusion Matrix and Statistics

                    Reference
Prediction           Bachelor's Doctorate Less than Secondary Master's Secondary
  Bachelor's              52252       989                 132    19274     16580
  Doctorate                   0         0                   0        0         0
  Less than Secondary        22         0                 785        2       180
  Master's                11193      3054                   3    16012      2687
  Secondary                9554        50                2949      253     30259

Overall Statistics

               Accuracy : 0.5974
                 95% CI : (0.5951, 0.5998)
    No Information Rate : 0.4393
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3751
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Bachelor's Class: Doctorate Class: Less than Secondary
Sensitivity                     0.7156          0.00000                   0.202895
Specificity                     0.6033          1.00000                   0.998744
Pos Pred Value                  0.5856              NaN                   0.793731
Neg Pred Value                  0.7303          0.97538                   0.981336
Prevalence                      0.4393          0.02462                   0.023275
Detection Rate                  0.3143          0.00000                   0.004722
Detection Prevalence            0.5368          0.00000                   0.005950
Balanced Accuracy               0.6594          0.50000                   0.600819
                     Class: Master's Class: Secondary
Sensitivity                  0.45052           0.6088
Specificity                  0.87040           0.8901
```

Problem Statement 3: Classifying if student have interacted with the course.

Stakeholders- Instructor: If we could classify the student based on interaction then instructor can check at the learning level of students and see if students just apply for certification. It could instructor to improve the course material if interaction is low. Considering column ‘viewed’ as indicator of interaction of the course.

1. Data Preparation: converting the column **‘visited’** into factor.

```
> # converting viewed to factor
> student_data_dataverse$viewed<- as.factor(student_data_dataverse$viewed)
```

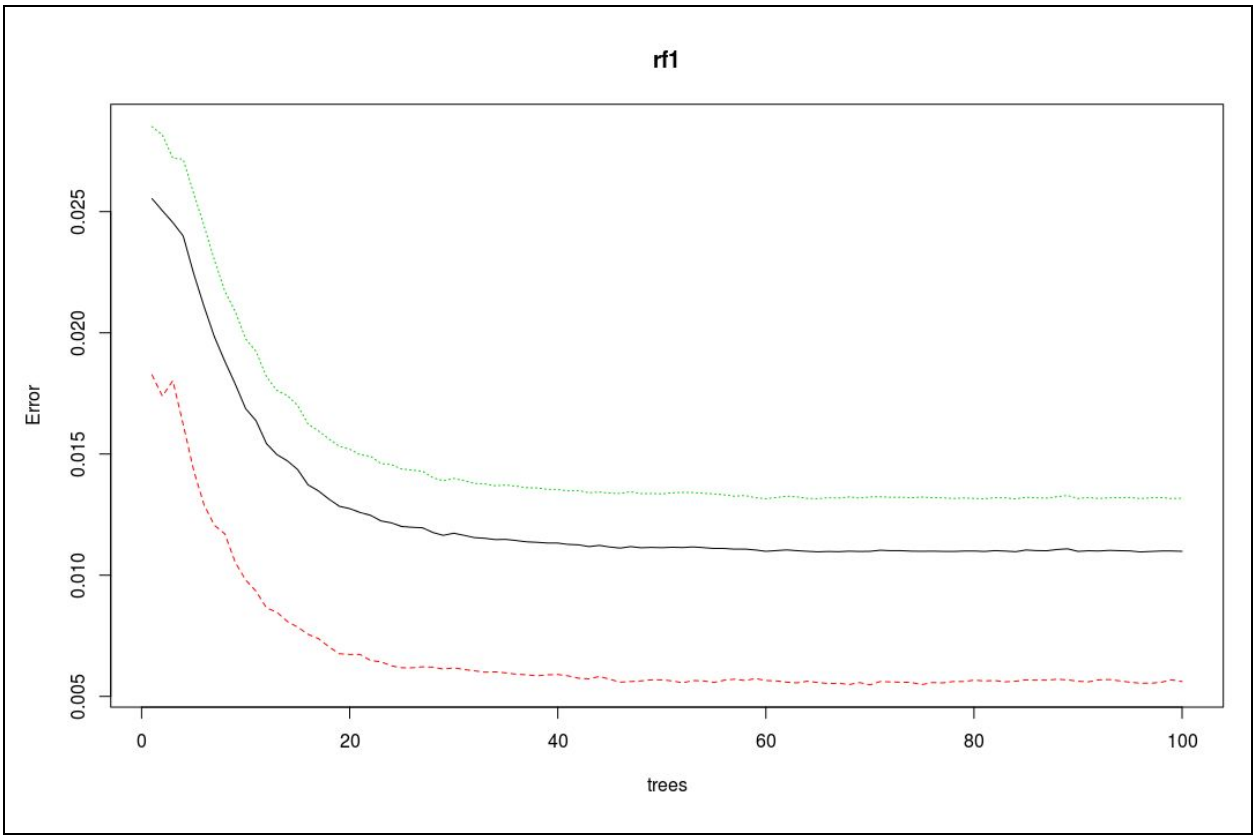
2. Splitting data into train and test data.

```
> # i am spliting data into training and testing with 60-40 rule
> #Create data for training
> sample1.ind = sample(2,
+                      nrow(student_data_dataverse),
+                      replace = T,
+                      prob = c(0.6,0.4))
> data1.dev = student_data_dataverse[sample1.ind==1,]
> data1.val = student_data_dataverse[sample1.ind==2,]
```

3. Fitting the model, we are using random forest with 100 trees, 'viewed' is the label column.

```
> #Fit Random Forest Model
> rf1 = randomForest(viewed ~ .,
+                    ntree = 100,
+                    data = data1.dev[,-5])
> plot(rf1)
> print(rf1)

Call:
 randomForest(formula = viewed ~ ., data = data1.dev[, -5], ntree = 100)
               Type of random forest: classification
                     Number of trees: 100
No. of variables tried at each split: 4

        OOB estimate of  error rate: 1.1%
Confusion matrix:
      0      1 class.error
0 72101    407 0.005613174
1  2347 175792 0.013175105
```

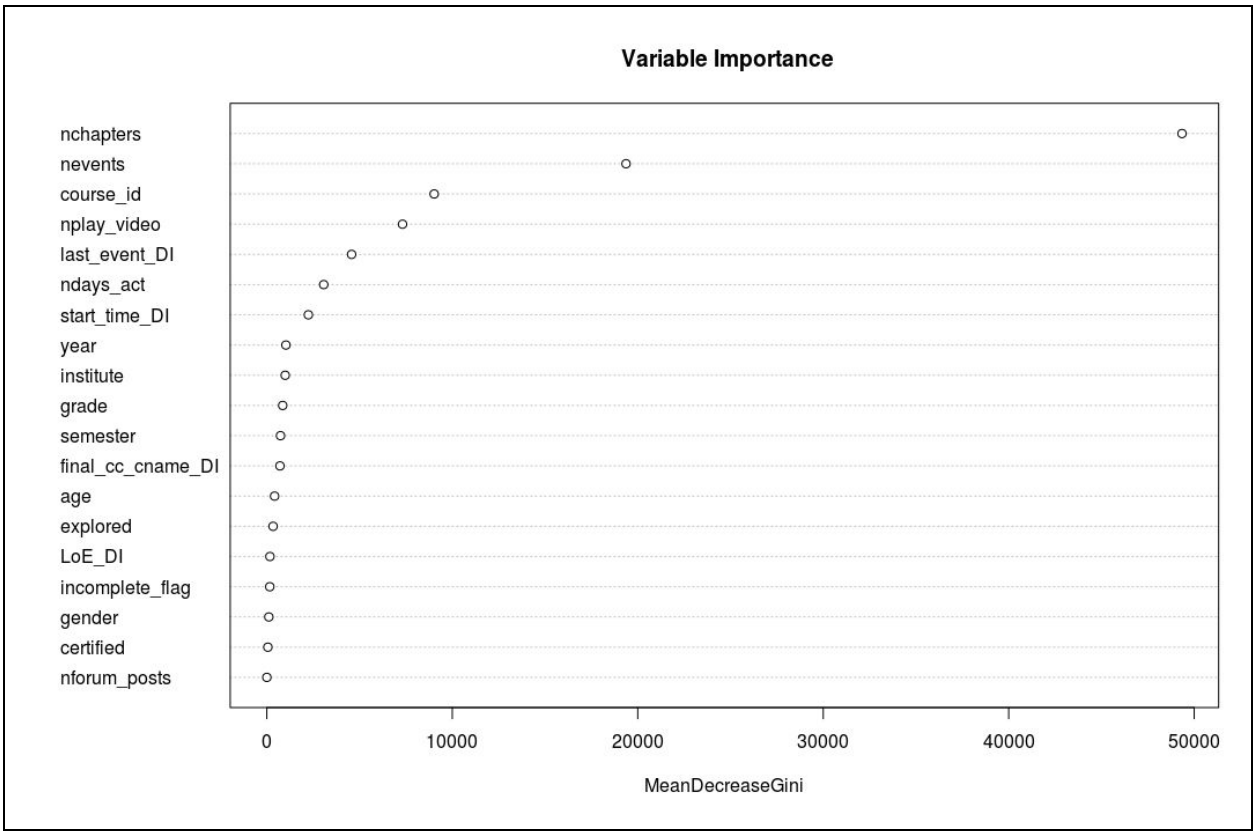
We can see that there are 407 false positives and 2347 false negatives. The OOB estimate of error rate is 1.1% .



The above plot shows that error rate becomes 1.1% while it reaches the 100 trees. Black curve shows OOB. Green curve shows viewed value 1 and red curve shows viewed value 0.

4. Plotting variable importance: This plot indicates what variables had the greatest impact in the classification model. It shows ‘nchapters’ is the most important variable.

```
> # Variable Importance
> varImpPlot(rf1,
+            sort = T,
+            main=" Variable Importance")
> #Variable Importance
> var.imp = data.frame(importance(rf1,
+                                 type=2))
> # make row names as columns
> var.imp$Variables = row.names(var.imp)
> print(var.imp[order(var.imp$MeanDecreaseGini,decreasing = T),])
                 MeanDecreaseGini         Variables
nchapters             49344.984534         nchapters
nevents               19368.611197           nevents
course_id              9023.096416         course_id
nplay_video            7318.425115       nplay_video
last_event_DI          4571.845336     last_event_DI
ndays_act              3068.389894         ndays_act
start_time_DI          2240.060004     start_time_DI
year                   1030.974882              year
institute               991.672662         institute
grade                   857.120246             grade
semester                738.220648          semester
final_cc_cname_DI       709.281893 final_cc_cname_DI
age                     419.789613               age
explored                338.083137          explored
LoE_DI                  169.053164            LoE_DI
incomplete_flag         160.035536   incomplete_flag
gender                  109.000652            gender
certified                56.430234         certified
nforum_posts              2.297111      nforum_posts
>
```



5. Applying the model to test data. Achieved accuracy is around 98%.

```
> # Now predicting on test data
> # Predicting response variable
> data1.val$predicted.response <- predict(rf1 ,data1.val)
>
> # Create Confusion Matrix
> print(
+   confusionMatrix(data=data1.val$predicted.response,
+                   reference=data1.val$viewed,
+                   positive='1'))
Confusion Matrix and Statistics

          Reference
Prediction      0      1
         0  47848   1488
         1    245 116693

               Accuracy : 0.9896
                 95% CI : (0.9891, 0.9901)
    No Information Rate : 0.7108
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9748
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9874
            Specificity : 0.9949
         Pos Pred Value : 0.9979
         Neg Pred Value : 0.9698
             Prevalence : 0.7108
         Detection Rate : 0.7018
   Detection Prevalence : 0.7033
      Balanced Accuracy : 0.9912

       'Positive' Class : 1
```

Done By: Chellapriyadharshini M (MT2016041)

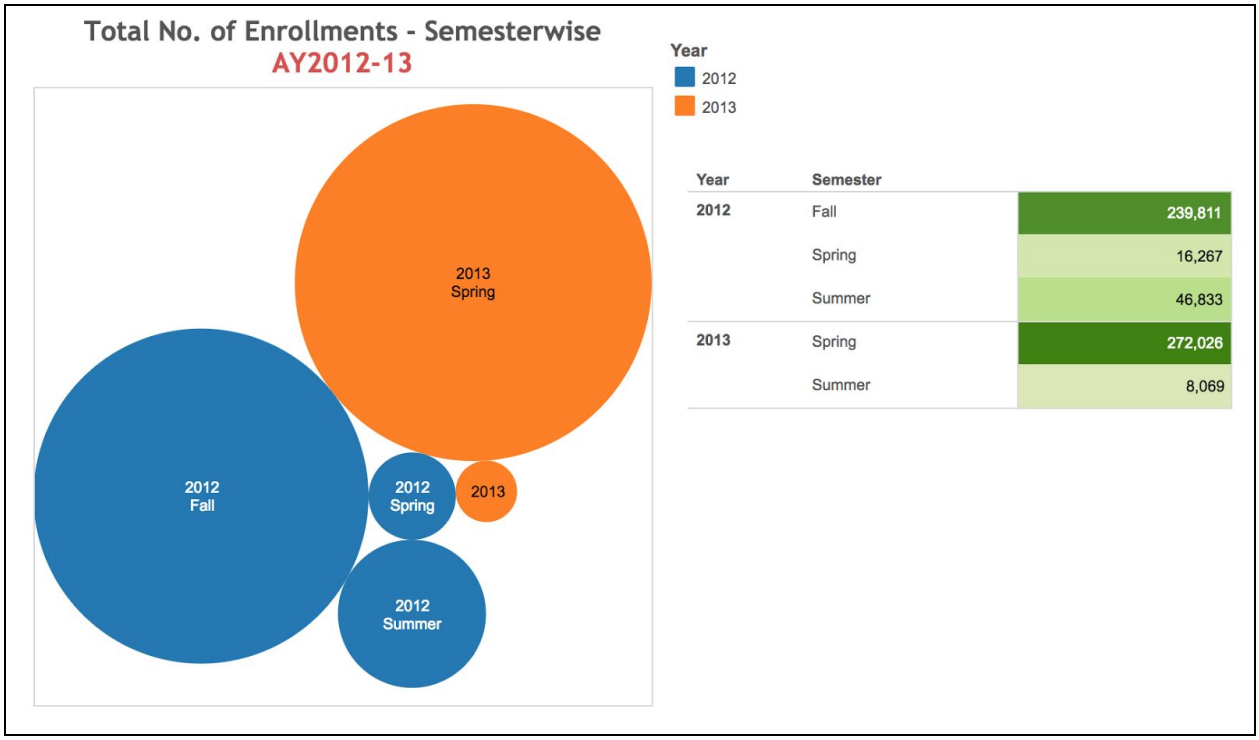
Exploratory Data Analytics

Target User: Institute

EDA Report: Total No. of Enrollments - Semester-wise for AY2012-13

- Description:
  - This report gives the semester-wise student enrollment numbers during the academic year 2012-2013.

- Tableau Visualization:



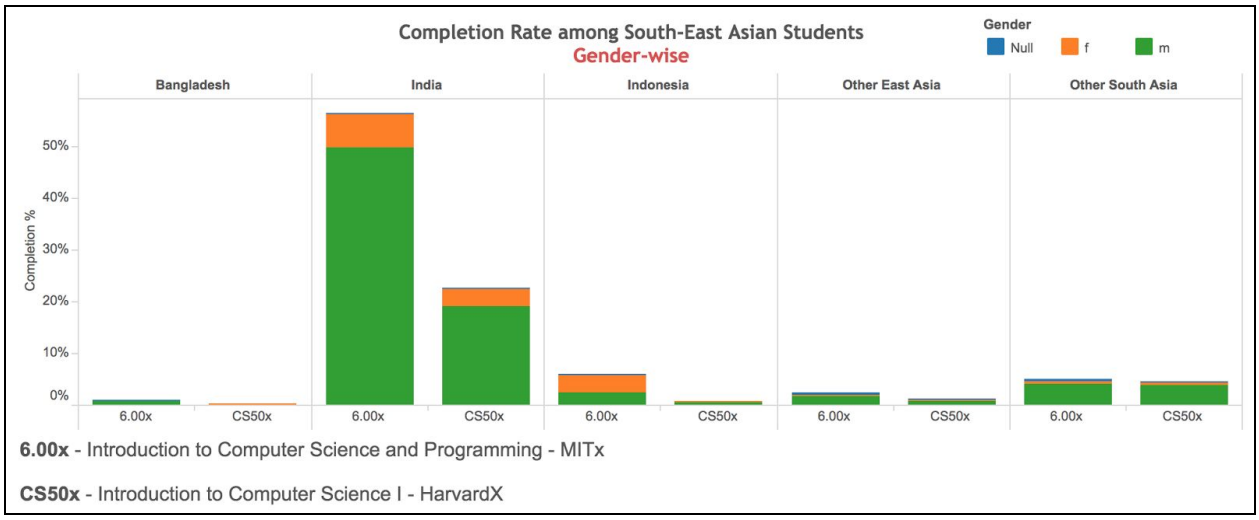
- Inference:
  - From the above we can see that while in 2012 the Fall courses had the most enrollments, in 2013 the Spring courses had the largest no. of students who registered. This will help the Institutes to decide when to introduce important courses.

Target User: Institute, Prospective Student

EDA Report: Comparing “Intro to CS” from MITx and HarvardX

- Description:
  - This report compares the completion rate of “Intro to CS” course offered by MIT and Harvard (among South-East Asian students).

- Tableau Visualization:



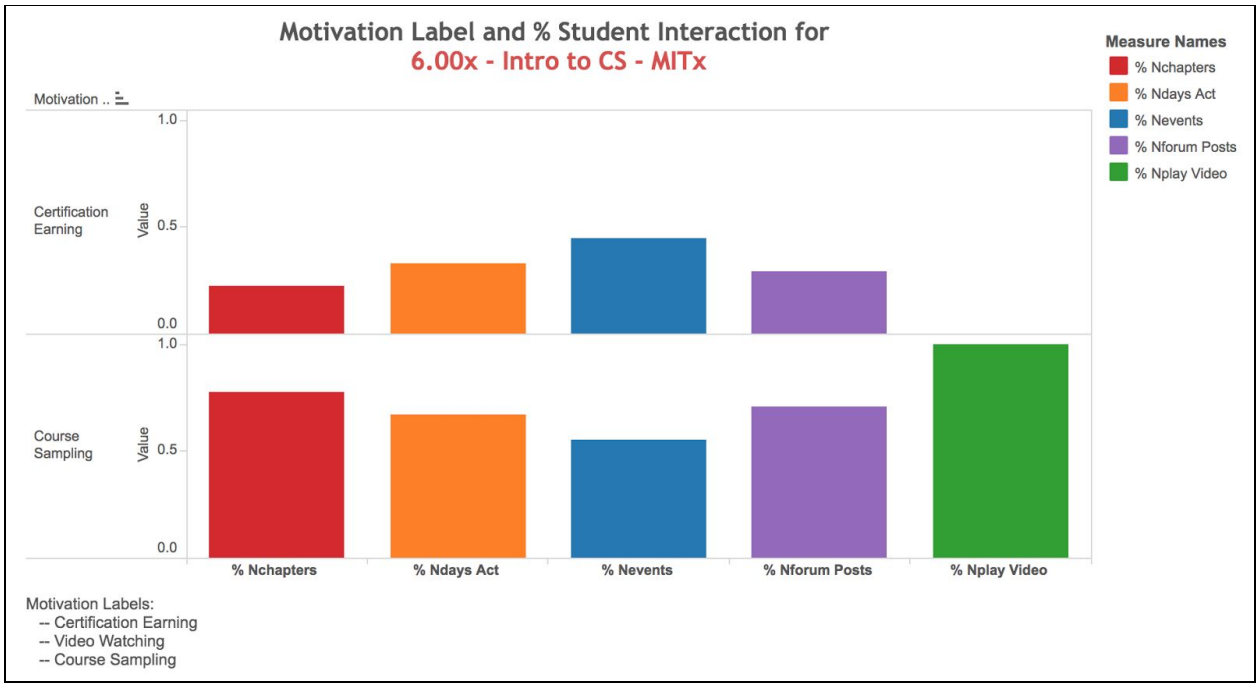
- Inference:
  - From the above we can see that Indian students completion % is way better than their counterparts from other Asian countries. This will help the Institute to take measures to improve their reach in less represented countries. This will also help the Prospective Students to decide if they should take the course from MIT or Harvard, given the similar nature of the two courses.

Target User: Institute

EDA Report: Motivation Label & % Student Interaction

- Description:
  - This report gives Student Interaction % along different Motivation Labels. We have grouped the students into 3 categories based on their level of interaction with the course, as: Certification Earning, Video Watching and Course Sampling.

- Tableau Visualization:



- Inference:
    - From the above report, which is for one specific course (6.00x), we can see that this course does not have any students under the “Video Watching” category. Also among the students who interact with the course materials, the percentage of “Course Sampling” students are more than the percentage of “Certification Earning” students.

Classification

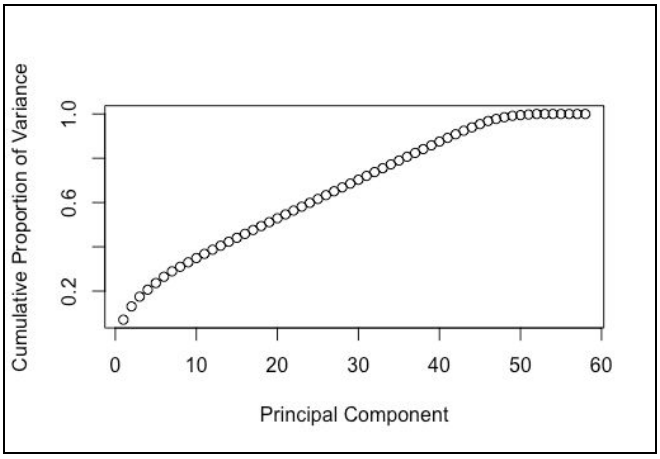
Classification Model: Complete Course/Not (SVM)

- Problem Formulation for Target User - Institute:
    - To predict if a newly enrolled student will complete the course and earn a certificate or not. (Binary Classification)

- Data Preparation:
  - Principal Component Analysis was done in order to identify the attributes that capture the most variance in the dataset.

```
# ************************** Principal Component Analysis ********************************************

# removing the response variable and identifier variables
train_pca <- select(ds1_train, institute, year, semester, age, final_cc_cname_DI, gender, LoE_DI, viewed, explored, nevents, ndays_act,
                    nplay_video, nchapters, nforum_posts)
test_pca <- select(ds1_test, institute, year, semester, age, final_cc_cname_DI, gender, LoE_DI, viewed, explored, nevents, ndays_act,
                   nplay_video, nchapters, nforum_posts)
# convert categorical varibles to numeric variables using one-hot encoding
new_train_pca <- dummy.data.frame(train_pca, names = c("institute", "year", "semester", "final_cc_cname_DI", "gender", "LoE_DI"))
new_test_pca <- dummy.data.frame(test_pca, names = c("institute", "year", "semester", "final_cc_cname_DI", "gender", "LoE_DI"))
# PCA
prin_comp <- prcomp(new_train_pca, scale. = T)
# compute std.dev and variance of each PC
std_dev <- prin_comp$sdev
pr_var <- std_dev^2
# proportion of variance
prop_varex <- pr_var/sum(pr_var)
# cumulative variance plot
plot(cumsum(prop_varex), xlab = "Principal Component", ylab = "Cumulative Proportion of Variance", type = "b")
plot(cumsum(prop_varex), xlim = c(48, 50), ylim = c(0.95, 1.00), xlab = "Principal Component", ylab = "Cumulative Proportion of Variance", type = "b")
# So looking at the plot we take 48 variables out of the 58 variables -
# to capture 98.5% of the variance in the dataset
# add a training set with principal components
```

  - From the cumulative proportion of variance plot, we can see that it is enough to take the first 48 principal components (out of 58) to capture 98.5% of variance.



- R Code for the Classifier:

```
svm_model <- svm(Completed_or_Not~ ., data=train.data, method="C-classification")
svmpredict <- predict(svm_model, test.data[, -1])
```

- Model Evaluation:

```
Confusion Matrix and Statistics

    pred
true      0      1
   0 169581    912
   1   1283   3126

               Accuracy : 0.9875
                 95% CI : (0.9869, 0.988)
    No Information Rate : 0.9769
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7337
 Mcnemar's Test P-Value : 2.848e-15

              Precision : 0.9947
                 Recall : 0.9925
                     F1 : 0.9936
             Prevalence : 0.9769
         Detection Rate : 0.9696
   Detection Prevalence : 0.9748
      Balanced Accuracy : 0.8833

       'Positive' Class : 0
```

- Inference:
    - We are able to predict if a new student will complete the course or not with 98.75% accuracy using SVM. This will help the Institute to follow up on the students' activities so that they complete the course.

Classification Model: Motivation Classification (Decision Tree)

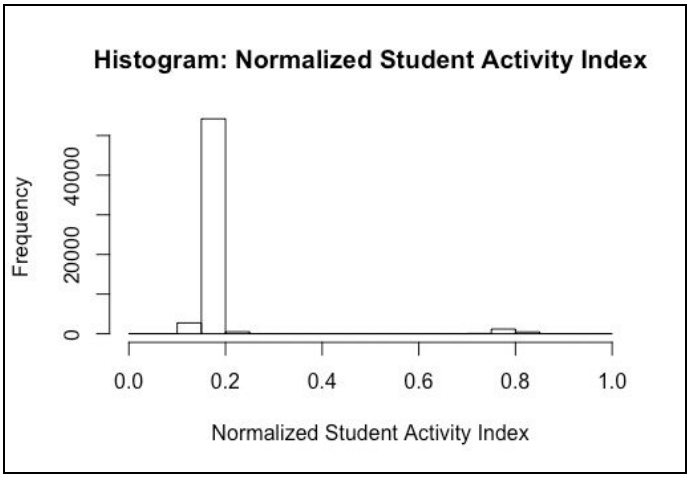
- Problem Formulation for Target User - Institute:
    - This is an extension to the above binary classification.
    - To define a "Motivation Class" by computing an Activity Index for every student. We have defined 3 groups in the Motivation Class: Certification Earning, Video Watching and Course Sampling. For a new student, to predict which class he/she might fall into. (Multi-Class Classification)

- Data Preparation:
    - In this we will do this classification for one particular course: 6.00x-Intro to CS and Programming from MITx.
    - We will calculate the Activity Index for a student using the formula:

- Pa = (k1 * sqrt(f(v)*f(e))/f(c) + (k2 * f(t))

```
# remove rows corresponding to the course CS50x as it has no info on "Student Activity"
dataset <- subset(dataset, !(dataset$course_id == "CS50x"))
# extracting only the rows belonging to course_id 6.00x - Intro to CS & Prog - MITx in Fall 2012
cs_mitx <- subset(dataset, (dataset$course_id == "6.00x" & dataset$year == 2012))
# maximum event activities in the course
e_m = max(cs_mitx$nevents)
# average video watching activities of all learners in the course
v_a = mean(cs_mitx$nplay_video)
# total number of courses in the dataset
tc <- as.numeric(length(unique(dataset$course_id)))
# total enrolled courses per user
enrolled_courses <- count(dataset, userid_DI)
enrolled_courses$n <- as.numeric(enrolled_courses$n)
enrolled_courses$c_value <- enrolled_courses$n/tc
enrolled_courses <- subset(enrolled_courses, (enrolled_courses$userid_DI %in% cs_mitx$userid_DI))

cs_mitx$e_value <- cs_mitx$nevents/e_m
cs_mitx$v_value <- cs_mitx$nplay_video/(v_a*v_a)
cs_mitx <- merge(cs_mitx, enrolled_courses, by = "userid_DI")
# Calculating the Activity Coefficient: Pa
cs_mitx$Pa <- ( ((sqrt((2^cs_mitx$e_value)*(2^cs_mitx$v_value)))/(sigmoid(cs_mitx$c_value)))) + ((2 * cs_mitx$Completed_or_Not))
cs_mitx$normalized_Pa <- (cs_mitx$Pa - min(cs_mitx$Pa))/(max(cs_mitx$Pa) - min(cs_mitx$Pa))

hist(cs_mitx$Pa, xlab = "Student Activity Index", main = "Histogram: Student Activity Index")
hist(cs_mitx$normalized_Pa, xlab = "Normalized Student Activity Index", main = "Histogram: Normalized Student Activity Index")
hist(cs_mitx$normalized_Pa, xlab = "Normalized Student Activity Index", xlim = c(0.1, 0.3), main = "Histogram: Normalized Student Activity Index")
```

- The plot of the Normalized Student Activity Index is as below. Based on this we divide into 3 categories:
  - Normalized Pa < 0.5 - Course Sampling
  - 0.5 <= Normalized Pa < 0.6 - Video Watching
  - Normalized Pa > 0.6 - Certification Earning



- R Code for the Classifier:

```
# Decision Tree
dec_train <- select(cert_earner_train, nevents, ndays_act, nplay_video, nchapters, nforum_posts, Completed_or_Not)
dec_test <- select(cert_earner_test, nevents, ndays_act, nplay_video, nchapters, nforum_posts, Completed_or_Not)

dec_tree_grade <- rpart(Completed_or_Not ~ nevents + ndays_act + nplay_video + nchapters + nforum_posts, data = dec_train,
                        method = "class", control = rpart.control(minsplit=10, minbucket=round(10/3), cp=0.01))
plot(dec_tree_grade, uniform=TRUE, compress=TRUE)
text(dec_tree_grade, use.n=TRUE, all=TRUE, cex=1)
rpart.plot(dec_tree_grade)

# Predict for test data
dtpredict_grade <- predict (dec_tree_grade, dec_test, type='class')
dtconfmat_grade <- table (true = dec_test[, 6], pred=dtpredict_grade)
```

- Model Evaluation:

```
Confusion Matrix and Statistics

    pred
true     0     1
   0 17173    67
   1    66   478

               Accuracy : 0.9925
                 95% CI : (0.9911, 0.9937)
    No Information Rate : 0.9694
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.874
 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9962
            Specificity : 0.8771
         Pos Pred Value : 0.9961
         Neg Pred Value : 0.8787
             Prevalence : 0.9694
         Detection Rate : 0.9656
   Detection Prevalence : 0.9694
      Balanced Accuracy : 0.9366

       'Positive' Class : 0
```

- Inference:
    - With the decision tree model we are able to achieve 99.25% accuracy in predicting the motivation class of the student. This might be used by the Institute to predict the nature of a newly enrolled student and take measures to motivate them towards earning a certificate. At a broad level, whenever a course is launched, among the students enrolled, the institute could predict what % fall under what class and modify course activities accordingly.

Classification Model: Student's Country (Random Forest)

- Problem Formulation for Target User - Institute:
    - To predict the country of domicile of any student given the other attributes including their interaction with the course. (Multi-Class Classification)

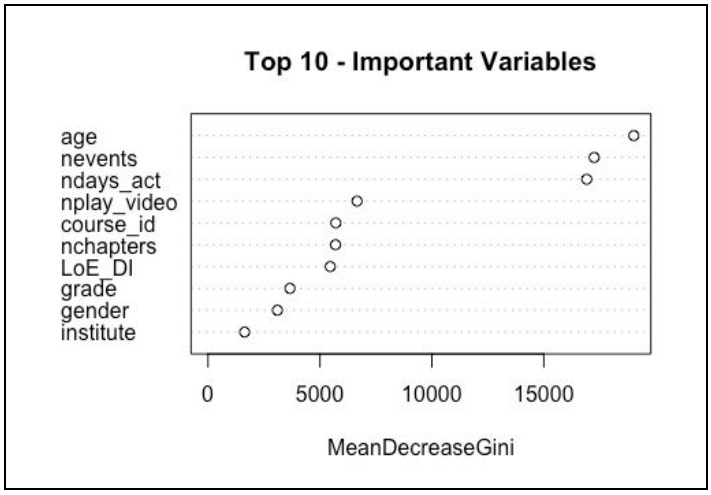
- Data Preparation:

```
dataset_3 <- read.csv("cleaned data/dataverse_cleaned.csv", header = TRUE)

# partitioning into training and test sets
smpsize_3 <- floor(0.70 * nrow(dataset_3))
set.seed(123)
trainindex_3 <- sample(seq_len(nrow(dataset_3)), size = smpsize_3)
dataset_3_train <- dataset_3[trainindex_3, ]
dataset_3_test <- dataset_3[-trainindex_3, ]
```

- R Code for the Classifier:

```
# Random Forest Classifier
rf_train <- select(dataset_3_train, -composite, -start_time_DI, -last_event_DI, -YoB, -userid_DI, -row_id)
rf_test <- select(dataset_3_test, -composite, -start_time_DI, -last_event_DI, -YoB, -userid_DI, -row_id)

rfmodel <- randomForest(final_cc_cname_DI ~ ., data = rf_train, ntree = 100, na.action=na.omit)
rfpredict <- predict(rfmodel, rf_test[, -8])
```

- Model Evaluation:
  - Top-10 Important Attributes:



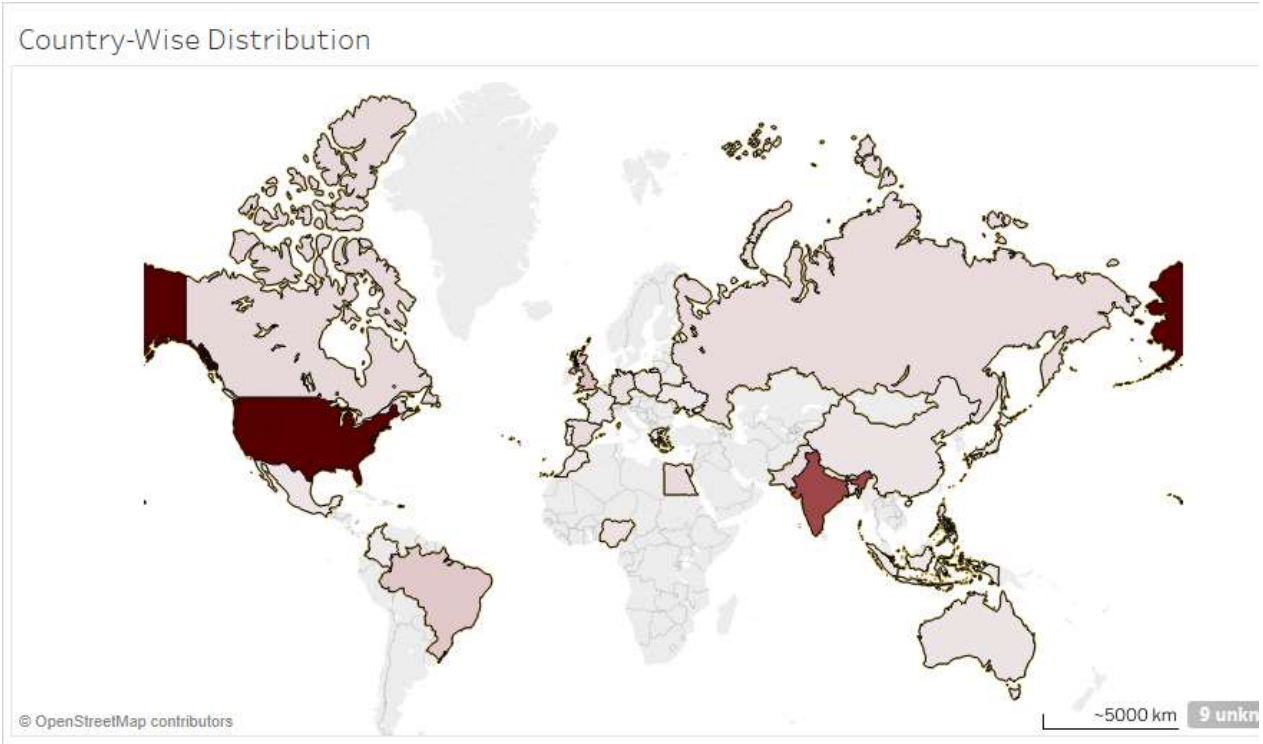

```
Overall Statistics

               Accuracy : 0.4353
                 95% CI : (0.4329, 0.4378)
    No Information Rate : 0.5225
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2788
 Mcnemar's Test P-Value : NA
```

- Inference:
    - We are able to predict the country of origin of the student with accuracy as low as 43.5%. The random forest classifier is not able to generalize well. This might mean the country attribute does not have enough correlation with the other attributes. As this attribute is filled by the Student, it begs the question, if the user is providing fake information here.

Done By:- Jyotsana (MT2016068)

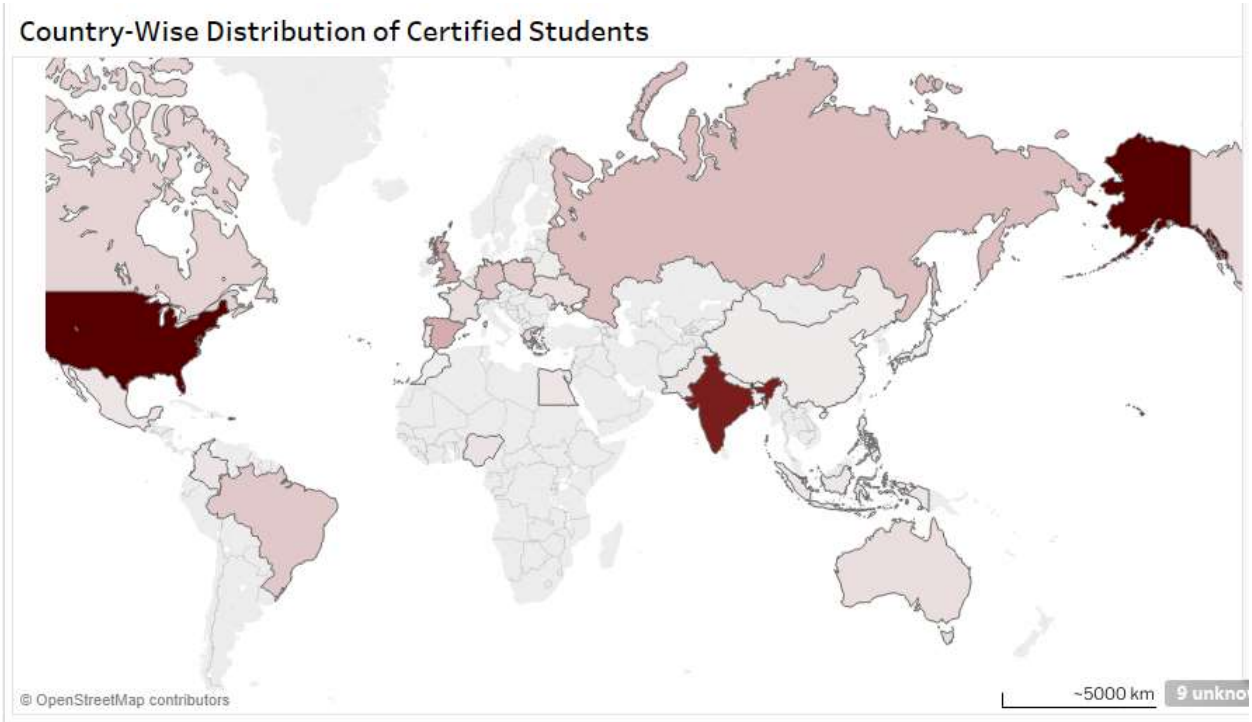
Exploratory Analysis

Country-Wise Distribution of Enrolled Student



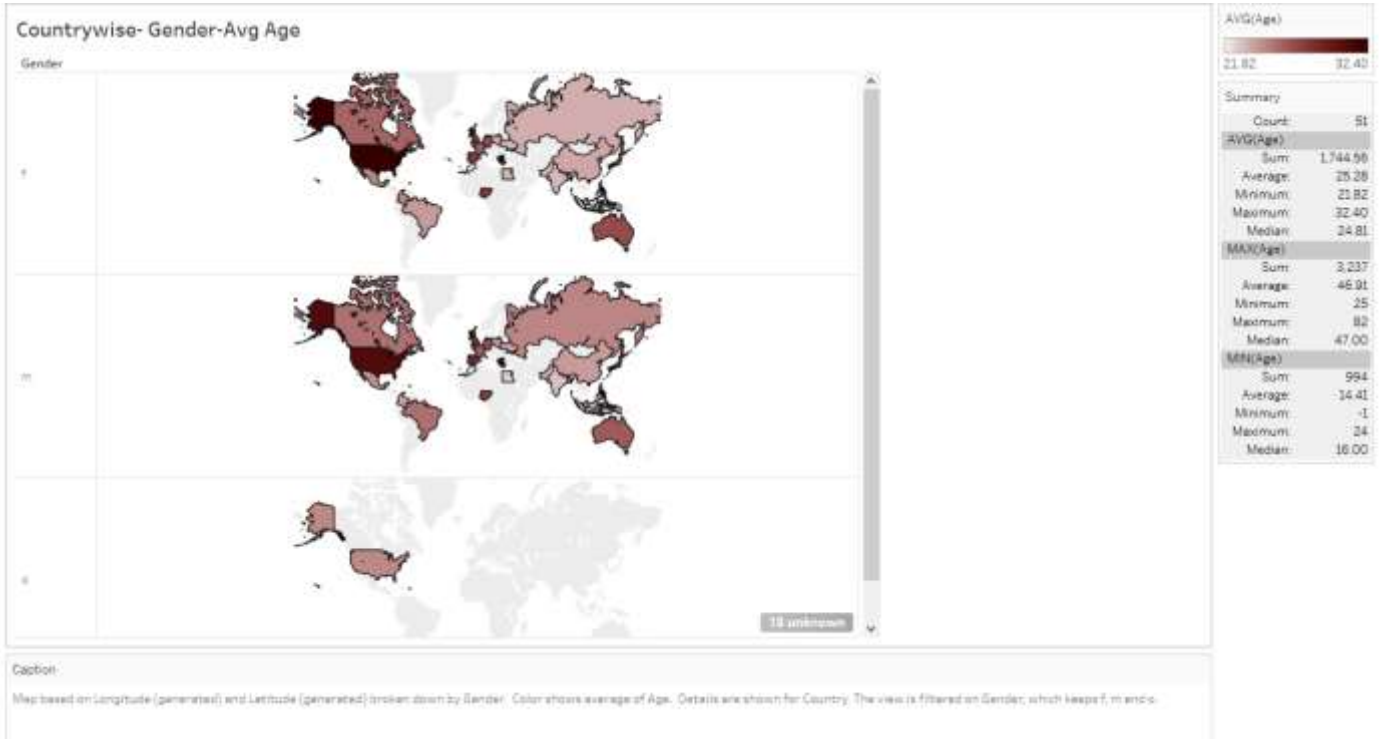
Inference: - Most of the students who enrolled belong to United States, China and India.

Country-Wise Distribution of Certified Students



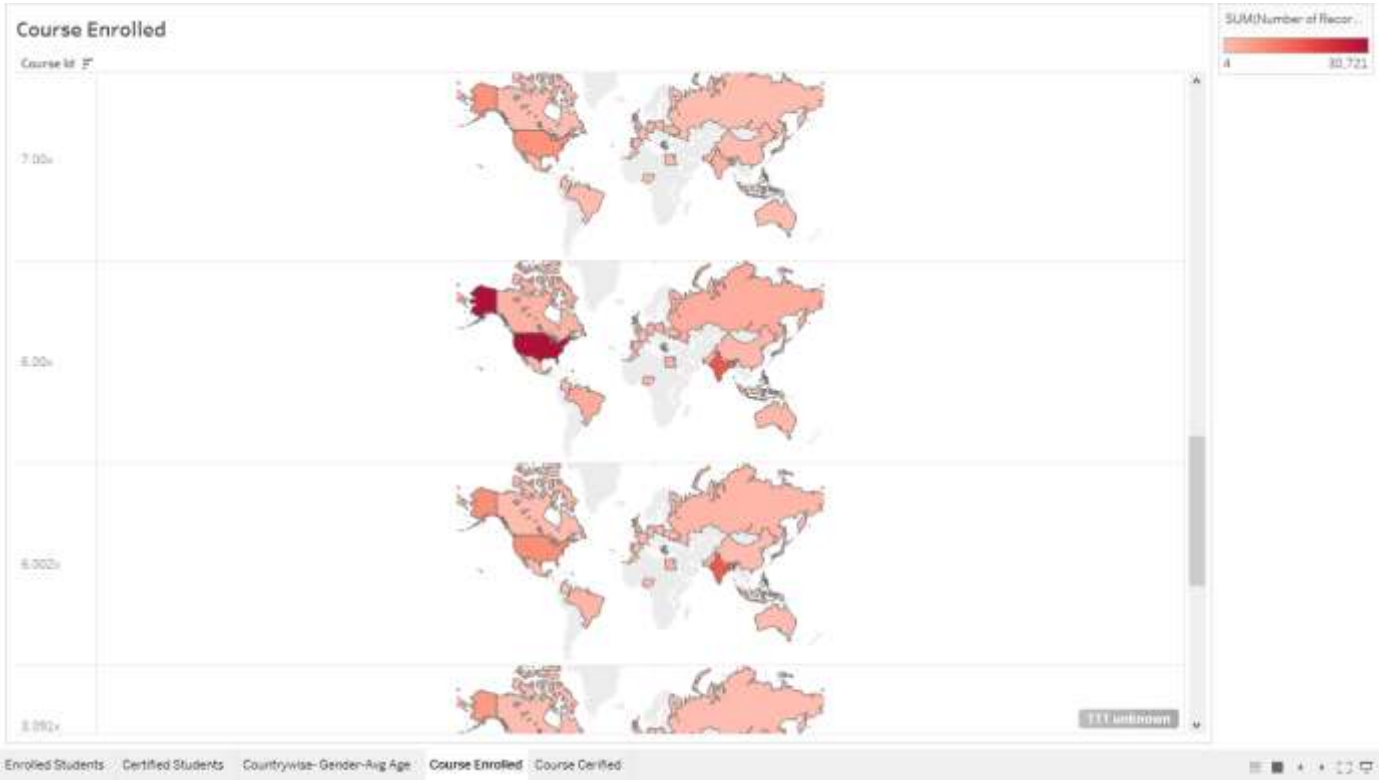
Inference:- We can see that US and Indian students have highest certifications among all. Though Chinese students were among high enrollees they are not that much enthusiastic about getting certification.

Country wise Average age distribution per gender



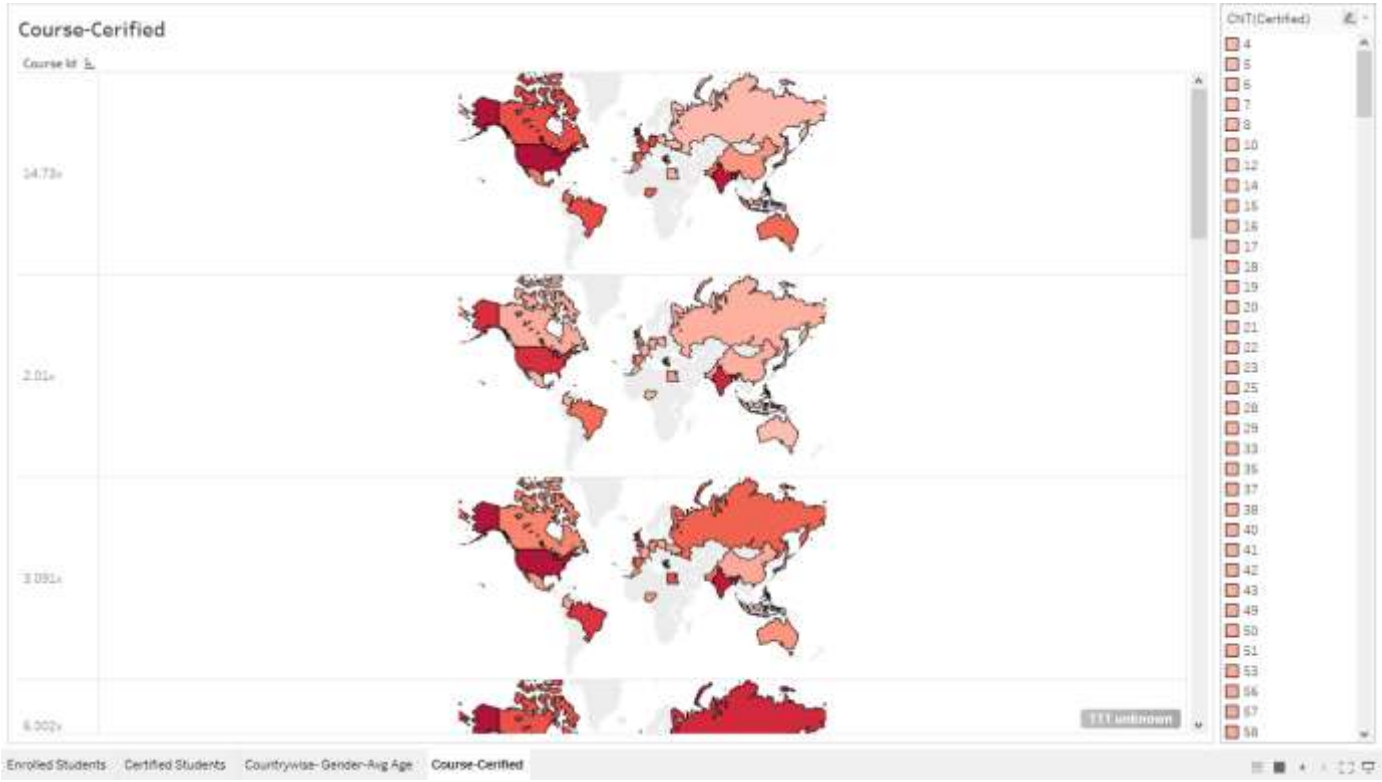
Inference: - US has highest average age among all genders.

Country-wise distribution of enrolled students per course



Inference: - It gives popularity of a course for each country.

Country-wise distribution of certified students per course



Classification Task

Predict Grade of Student

Benefit to Stakeholders

Students: - Predicting the grade based on current activity of a student can gauge their preparation for end of course examination.

Instructors: - Instructors can identify students who are most likely not make through the exam and try to help them.

Edx Platform: - All these in turn will lead to better experience for instructors and students and hence enhance productivity of the platform.

Description

Edx has specified in their website their default overall final grade range according to the percentage grade obtained by a student. Though this can be customized by an instructor who is offering a course, it is quite infrequent.

So, we defined a new attribute called **"Letter Grade"** accordingly: -

%age Grade Range	Letter Grade
0-50	"F"
50-75	"B"
75-100	"A"

R Script:- To convert grade into categorical values

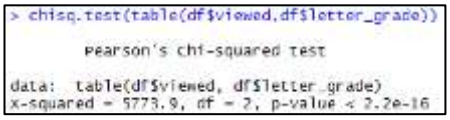
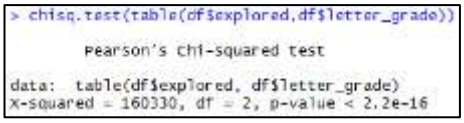
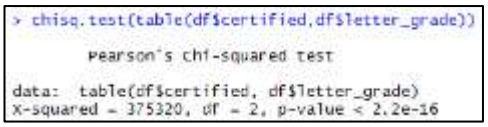
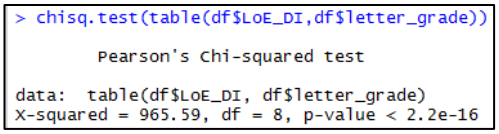
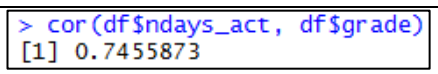
```
  ##Creating Letter grade from grades
df$letter_grade<-cut(df$grade,
                     breaks = c(0,0.50,0.75,1.00),
                     labels = c("F","B","A"),
                     right=FALSE)
df$letter_grade<-as.factor(df$letter_grade)
```

A quick visualization gave following frequency distribution

It shows that we don't have values for 1,739 records out of 404,807 records. So, we can use known values to predict the unknown.

Letter Grade	
A	9,291
B	4,923
F	3,88,854
NA	1,739

Among all the attributes we identified following ones which according to our understanding should have impact on the grade of the student.

Attribute	Description	Type	Correlation/Chi-Square with Grade
Viewed	Anyone who accessed course materials from the "Courseware" tab	Categorical 0/1	Hypothesis whether the activity of viewing the course material is independent of student's grade at .05 significance level.  p-value is less than 0.05 Hence, we reject the null hypothesis.
Explored	Anyone who accessed at least half of the chapters in the courseware	Categorical 0/1	Hypothesis whether the activity of exploring at least half the course material is independent of student's grade at .05 significance level.  p-value is less than 0.05 Hence, we reject the null hypothesis.
Certified	Anyone who earned a certificate	Categorical 0/1	Hypothesis whether earning a certificate for course is independent of student's grade at .05 significance level.  p-value is less than 0.05 Hence, we reject the null hypothesis.
LoE_DI	Highest level of education completed	Categorical	Hypothesis whether level of education is independent of student's grade at .05 significance level.  p-value is less than 0.05 Hence, we reject the null hypothesis.
Nevents	Number of interactions with the course	Numeric	 Strongly Positively Correlated
ndays_act	Number of unique days students	Numeric	 Strongly Positively Correlated

	interacted with the course		

nplay_video	Number of play video events within the course	Numeric	`> cor(df$nplay_video, df$grade)`
			`[1] -0.1768684`
			Weakly Negatively Correlated

Nchapters	Number of chapters (within the courseware) with which the student interacted	Numeric	`> cor(df$nchapters, df$grade)`
			`[1] 0.6896213`
			Strongly Positively Correlated

nforum_post	Number of posts in the Discussion Forum	Numeric	`> cor(df$nforum_posts, df$grade)`
			`[1] 0.106736`
			Weakly Positively Correlated

Splitting the dataset

We remove the records having grades as NA and use it later for prediction. (Validation Set)
All the other records are divided into the ratio 70:30 for training and testing purpose.

```
##Splitting the data into train-test-validation sets
df<-df[-c(10)]

vald<-df[is.na(df$letter_grade),]
df_tt<-df[!is.na(df$letter_grade),]
dt = sort(sample(nrow(df_tt), nrow(df_tt)*.7))
train<-df_tt[dt,]
test<-df_tt[-dt,]
```

Building Decision Tree Model

```
##Building Decision Tree
library(party)
grade.ct<-ctree(letter_grade ~ .,data=train)
plot(grade.ct)
test_pred<-cbind(test[-c(10)],predict(grade.ct,newdata=test[-c(10)]))
colnames(test_pred)[10] <- "letter_grade"

Accuracy<-length(which(test_pred$letter_grade==test$letter_grade))/nrow(test)*100
confusionMatrix(table(test_pred$letter_grade,test$letter_grade))
```

Output:-

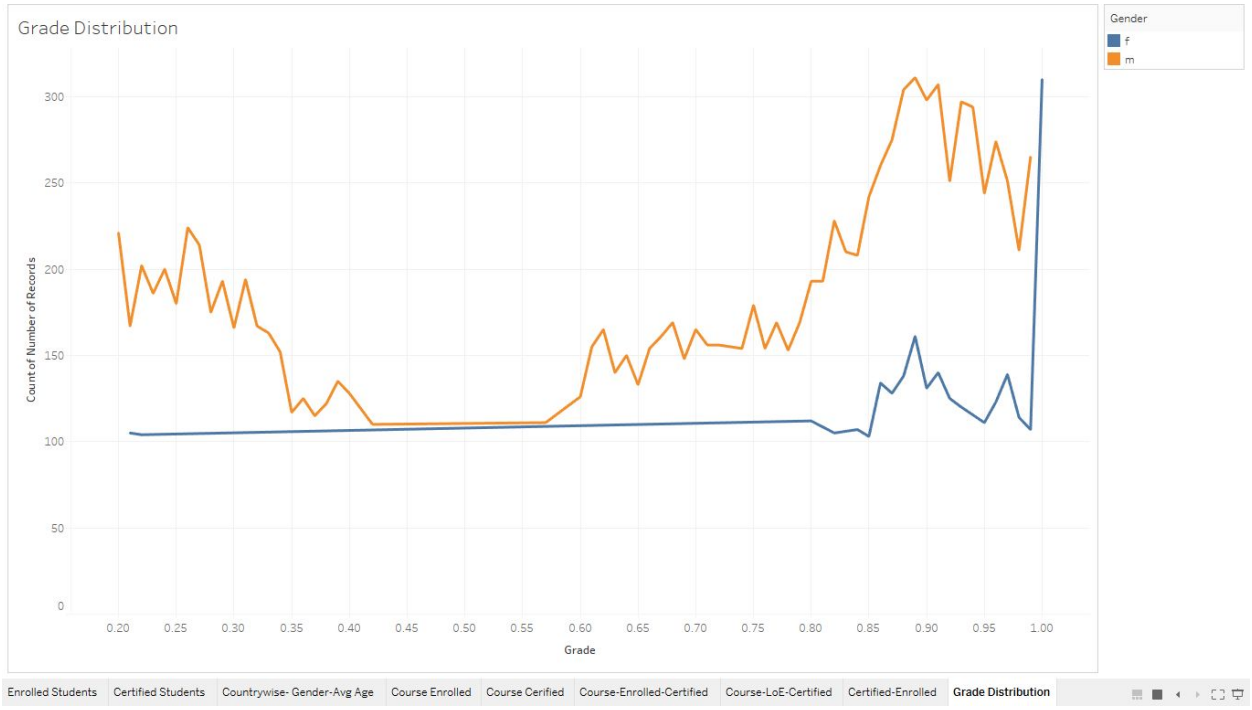
```
Confusion Matrix and Statistics


          F      B      A
  F 116637    285     24
  B      0     34     21
  A      0   1174   2746

Overall Statistics

               Accuracy : 0.9876
                 95% CI : (0.9869, 0.9882)
    No Information Rate : 0.9646
    P-Value [Acc > NIR] : < 2.2e-16
```

The model gave an accuracy %age of 98.76%
We can see that it had most trouble in distinguishing B grade from A and F which should be the case as most of the B values lies at the border of A and F.

Daminee Sao :-

Exploratory Analysis

Grade Distribution- Gender



Inference :- Females have obtained more higher percentage grades as compared to males.

Problem Statement 4: Classifying which student will choose which institute.

Stakeholders- Institute: Institute could now able to estimate number of students who can join the institute, thus according to the predicted performance of students, institue can increase the number of courses offered. For example, the institute can identify what are the other subjects needs to be offered to increase the student count.

Data Preparation for classification

1. **Converting institute as a factor**

```
> #converting institue as a factor
> data$institute <- as.factor(data$institute)
> class(data$institute)
[1] "factor"
```

2. **Select subset of the original data named as filterdata by selecting those fields which can affect the selection of institute.**

```
> library(dplyr)
> filterdata <- select(data,institute,course_id,semester,viewed,explored,certified,final_
cc_cname_DI,LoE_DI,gender,age)
>
>
>
>
```

Code for doing classification

I'm using Decision tree algorithm for classification of the dataset.

1. **Prepare the environment to work he Decision Tree algorithm.**

```
> install.packages('rattle')
  install.packages('rpart.plot')
  install.packages('RColorBrewer')
  library(rattle)
  library(rpart.plot)
  library(RColorBrewer)
```

2. **Splitting Data into Training and Testing data using 70-30 rule.**

```
> index <- 1:nrow(filterdata)
> testindex <- sample(index,trunc(length(index)/3))
> testrecords <- filterdata[testindex,]
> trainrecords <- filterdata[-testindex,]
>
```

3. Construct the classification model using the algorithm on the training records

```
>
> clsdata <- rpart(institute~semester+final_cc_cname_DI+LoE_DI+gender+ age,data= trainrec
ords,method="class")
```

4. Predict the class labels of the test records using the constructed model

```
>
> clspredict <- predict(clsdata,testrecords,type="class")
>
```

5. Cross taulate the predicted classes against the true classes (called as confusion matrix)

```
>
> clscompare <- table(true=testrecords[,7],pred=clspredict)
>
> clscompare
    pred
true HarvardX   MITx
   0    32121 101919
   1     1360   3573
>
```

6. Analysing the confusion matrix

```
>
>
> confusionMatrix(clspredict,testrecords$institute)
Confusion Matrix and Statistics

          Reference
Prediction HarvardX  MITx
  HarvardX    21977 11504
  MITx        37314 68178

               Accuracy : 0.6487
                 95% CI : (0.6462, 0.6512)
    No Information Rate : 0.5734
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.2396
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.3707
            Specificity : 0.8556
         Pos Pred Value : 0.6564
         Neg Pred Value : 0.6463
             Prevalence : 0.4266
         Detection Rate : 0.1581
   Detection Prevalence : 0.2409
      Balanced Accuracy : 0.6131

       'Positive' Class : HarvardX

>
```

From the accuracy measure which is 64.67% we can conclude that the selection of institute is almost similar, does not depends on the country ,gender, course, semester, age.

Done By: Tehreem Ansari (MT2016145)

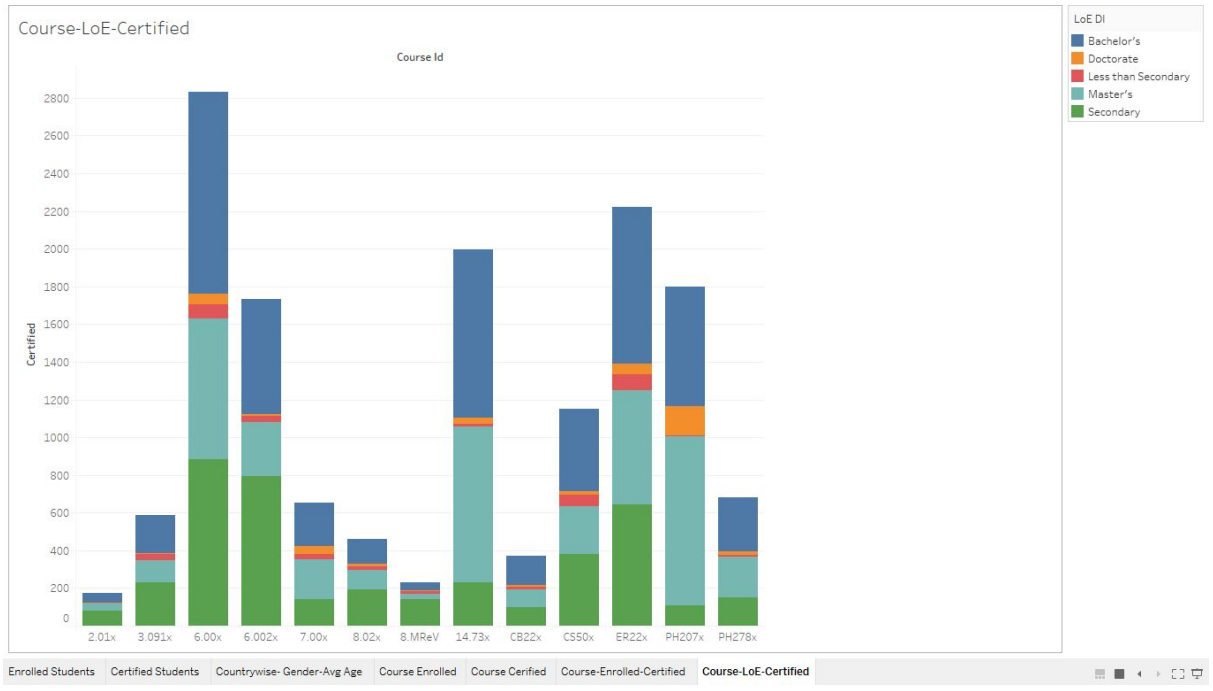
Exploratory Data Analytics

Target User: Institute

EDA Report: Course-LoE in certified

- Description:
    - We try to analyse how the level of education for each course affects the certification.

- Tableau Visualization:



- Inference:

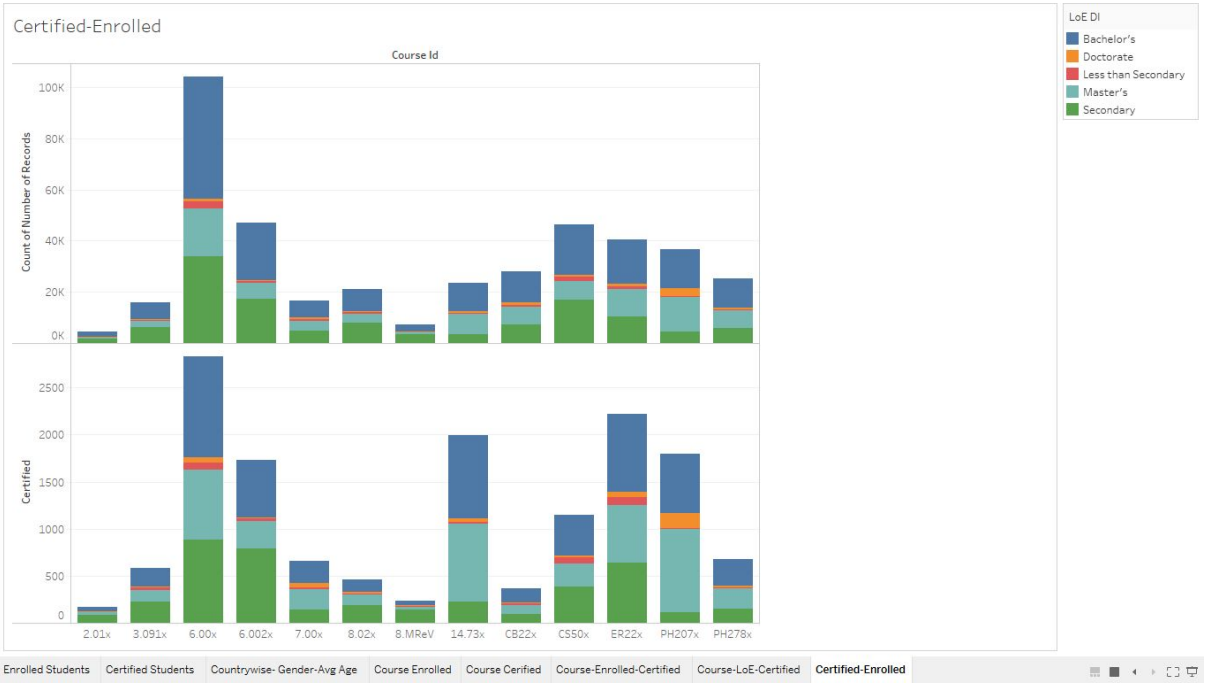
  From the above we can see that maximum certification is achieved by students who have level of education as bachelors or masters. Also, people with less than secondary have the lowest certification.

Target User: Institute

EDA Report: Course-LoE in certified

- Description:
    - We try to analyse which category of students have the highest enrollment rate vs the certification rate.

- Tableau Visualization:



- Inference:

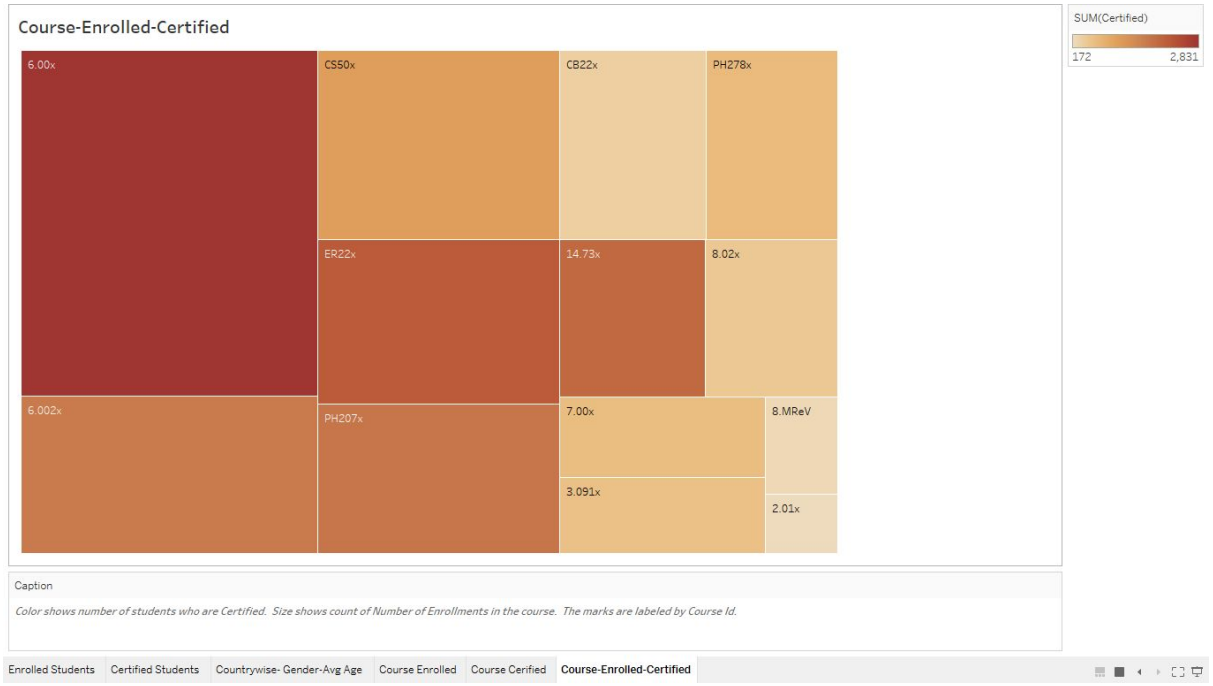
  From the above we can see that maximum students who enroll have level of education as bachelors and/or secondary. However, when compared to the students who achieve a certification, maximum level of education is bachelors or masters.

Target User: Institute

EDA Report: Course-Enrolled in certified

- Description:
    - We try to analyse number of enrollments vs the number of those students who got certificate.

- Tableau visualization:



- Inference:

  The size represents the number of enrollments and the colour shows the percent of students who achieved the certificate. We can see the 6.00x course has the maximum enrollments and also the maximum number of people who got the certification. Similarly, the course 2.01x has less enrollments and thus few certifications.

Classification

Problem Statement 1: Creating a model for predicting if the current student will achieve a certificate or not and then cross checking with test data.

Target User: **Institute and Instructor**:

Goad is to create a model using decision tree on training data to understand which students get a certificate and which do not. The model is then tested and verified to give accuracy measure.

Code for doing classification

1. Reading the data:

```
data<-read.csv("/home/iiitb/DataAnalyticsProject/Data/big_student_clear_third_version.csv", header=TRUE)
```

```
head(data)
```

X	institute	course_id	year	semester	userid_DI	viewed	explored	certified	final_cc_cname_DI	···	grade	sta
4	HarvardX	PH207x	2012	Fall	MHxPC130313697	0	0	0	India	···	0	201
6	HarvardX	PH207x	2012	Fall	MHxPC130237753	1	0	0	United States	···	0	201
7	HarvardX	CS50x	2012	Summer	MHxPC130202970	1	0	0	United States	···	0	201
20	HarvardX	CS50x	2012	Summer	MHxPC130223941	1	0	0	Other Middle East/Central Asia	···	0	201
22	HarvardX	PH207x	2012	Fall	MHxPC130317399	0	0	0	Australia	···	0	201
23	HarvardX	CS50x	2012	Summer	MHxPC130191782	1	0	0	Pakistan	···	0	201

2. Converting the label column to factor

Since we are going to predict the certified class, this is our label class. We need to convert this to factor.

```
class(data$certified)
```

'integer'

```
data['certified']<-as.factor(data$certified)
```

```
class(data$certified)
```

'factor'

3. Loading the required libraries

```
library(dplyr)
library(rpart)
library(ggplot2)
```

4. Selecting the important columns

```
data_filtered<-
select(data,institute,course_id,semester,viewed,explored,certified,final_cc_cname_DI,LoE_DI,gender,
                    grade,start_time_DI,last_event_DI,nevents,ndays_act,nplay_video,nchapters,nforum_pos
ts,incomplete_flag
                    ,age)
```

```
head(data_filtered)
```

institute	course_id	semester	viewed	explored	certified	final_cc_cname_DI	LoE_DI	gender	grade	start_time_DI
HarvardX	PH207x	Fall	0	0	0	India	Bachelor's	m	0	2012-07-24
HarvardX	PH207x	Fall	1	0	0	United States	Secondary	m	0	2012-07-24
HarvardX	CS50x	Summer	1	0	0	United States	Bachelor's	m	0	2012-07-24
HarvardX	CS50x	Summer	1	0	0	Other Middle East/Central Asia	Secondary	m	0	2012-07-24
HarvardX	PH207x	Fall	0	0	0	Australia	Master's	f	0	2012-07-24
HarvardX	CS50x	Summer	1	0	0	Pakistan	Bachelor's	m	0	2012-07-24

Since there are few columns that are not useful, like User_id and year, we will ignore those columns.

5. Dividing the data set into train and test in 70-30 ratio

```
set.seed(101)
# Now Selecting 70% of data as sample from total 'n' rows of the data
sample <- sample.int(n = nrow(data_filtered), size = floor(.70*nrow(data)), replace = F)
train <- data[sample, ]
test  <- data[-sample, ]
```

```
nrow(train)
nrow(test)
```

```
291844
```

```
125077
```

6. Creating the model

```
attach(data_filtered)
dtfit<-rpart(certified~institute+course_id+semester+viewed+explored+final_cc_cname_DI+LoE_DI+
                      grade+start_time_DI+last_event_DI+nevents+ndays_act+nplay_video+nchapte
ts+incomplete_flag
                     +age,data = train, method="class")
dtfit
```

Here, we create the decision tree model.

```
n= 291844

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 291844 10449 0 (0.964196626 0.035803374)
  2) grade< 0.545 281130   177 0 (0.999370398 0.000629602) *
  3) grade>=0.545 10714   442 1 (0.041254433 0.958745567) *
```

7. Model understanding

```
summary(dtfit)
```

```
         CP nsplit  rel error     xerror        xstd
1 0.9407599      0 1.00000000 1.00000000 0.009606063
2 0.0100000      1 0.05924012 0.05924012 0.002378535

Variable importance
    grade ndays_act   nevents nchapters
       50        18        17        15
```

```
Node number 1: 291844 observations,    complexity param=0.9407599
  predicted class=0  expected loss=0.03580337  P(node) =1
    class counts: 281395 10449
   probabilities: 0.964 0.036
  left son=2 (281130 obs) right son=3 (10714 obs)
  Primary splits:
      grade     < 0.545  to the left,  improve=18948.470, (0 missing)
      ndays_act < 27.5   to the left,  improve= 8542.374, (0 missing)
      nchapters < 10.5   to the left,  improve= 8530.945, (0 missing)
      explored  < 0.5    to the left,  improve= 8065.492, (0 missing)
      nevents   < 3345.5 to the left,  improve= 7504.846, (0 missing)
  Surrogate splits:
      ndays_act < 35.5   to the left,  agree=0.977, adj=0.365, (0 split)
      nevents   < 3826.5 to the left,  agree=0.976, adj=0.336, (0 split)
      nchapters < 14.5   to the left,  agree=0.974, adj=0.304, (0 split)

Node number 2: 281130 observations
  predicted class=0  expected loss=0.000629602  P(node) =0.9632886
    class counts: 280953   177
   probabilities: 0.999 0.001

Node number 3: 10714 observations
  predicted class=1  expected loss=0.04125443  P(node) =0.03671139
    class counts:   442 10272
   probabilities: 0.041 0.959
```

8. Predicting the label using the model

```
predicted<-predict(dtfit,test,type="class")
```

9. Creating the confusion matrix

```
table(test$certified)
```

```
     0      1
120637   4440
```

```
library(caret)
print(confusionMatrix(data=predicted, reference=test$certified,positive='1'))
```

```
Confusion Matrix and Statistics

          Reference
Prediction      0      1
         0 120438     77
         1    199   4363

               Accuracy : 0.9978
                 95% CI : (0.9975, 0.998)
    No Information Rate : 0.9645
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : NA
 Mcnemar's Test P-Value : 3.256e-13

            Sensitivity : 0.98266
            Specificity : 0.99835
         Pos Pred Value : 0.95638
         Neg Pred Value : 0.99936
             Prevalence : 0.03550
         Detection Rate : 0.03488
   Detection Prevalence : 0.03647
      Balanced Accuracy : 0.99050

       'Positive' Class : 1
```

Problem Statement 2: Creating a model for predicting if a student will explore the course or not.

On the basis of institute, country, LoE_DI, gender and age, we need to predict if the student will explore the data or not.

1. Loading libraries

```
library(dplyr)
library(rpart)
library(ggplot2)
library(caret)
```

2. Reading the data

```
data<-read.csv("/home/iiitb/DataAnalyticsProject/Data/big_student_clear_third_version.csv", header=TRUE)
```

```
data[,5:12]
```

semester	userid_DI	viewed	explored	certified	final_cc_cname_DI	LoE_DI	gender
Fall	MHxPC130313697	0	0	0	India	Bachelor's	m
Fall	MHxPC130237753	1	0	0	United States	Secondary	m
Summer	MHxPC130202970	1	0	0	United States	Bachelor's	m
Summer	MHxPC130223941	1	0	0	Other Middle East/Central Asia	Secondary	m
Fall	MHxPC130317399	0	0	0	Australia	Master's	f
Summer	MHxPC130191782	1	0	0	Pakistan	Bachelor's	m
Spring	MHxPC130191782	1	0	0	Pakistan	Bachelor's	m
Fall	MHxPC130267000	0	0	0	Other South Asia	Master's	f
Summer	MHxPC130435800	1	0	0	India	Bachelor's	m

3. Cleaning the data

```
#rows_num<-which(is.na(data$gender))
data2<-data[!(is.na(data$gender) | data$gender==""), ]
data2[,5:12]
```

We have some blank spaces and NA's in the gender column. Hence we will delete those rows.

	semester	userid_DI	viewed	explored	certified	final_cc_cname_DI	LoE_DI	gender
1	Fall	MHxPC130313697	0	0	0	India	Bachelor's	m
2	Fall	MHxPC130237753	1	0	0	United States	Secondary	m
3	Summer	MHxPC130202970	1	0	0	United States	Bachelor's	m
4	Summer	MHxPC130223941	1	0	0	Other Middle East/Central Asia	Secondary	m
5	Fall	MHxPC130317399	0	0	0	Australia	Master's	f
6	Summer	MHxPC130191782	1	0	0	Pakistan	Bachelor's	m
7	Spring	MHxPC130191782	1	0	0	Pakistan	Bachelor's	m
8	Fall	MHxPC130267000	0	0	0	Other South Asia	Master's	f
9	Summer	MHxPC130435800	1	0	0	India	Bachelor's	m
10	Fall	MHxPC130284813	0	0	0	United States	Bachelor's	m
11	Summer	MHxPC130235150	1	1	0	India	Bachelor's	m

4. Converting the label column to factor

```
class(data2$explored)
```

'integer'

```
data2$explored<-as.factor(data2$explored)
class(data2$explored)
```

'factor'

5. Selecting the required columns

Since we are trying to predict using only institute, country, LoE_DI, gender and age, we will keep only those columns and ignore the rest.

```
data_filtered<-select(data2,institute,explored,final_cc_cname_DI,LoE_DI,gender,age)
```

6. Separating the dataset into train and test data.

```
set.seed(101)
# Now Selecting 70% of data as sample from total 'n' rows of the data
sample <- sample.int(n = nrow(data_filtered), size = floor(.70*nrow(data)), replace = F)
train <- data[sample, ]
test  <- data[-sample, ]
nrow(train)
nrow(test)
```

7. Creating the model

```
attach(data_filtered)
dtfit <- rpart(explored~institute+final_cc_cname_DI+LoE_DI+gender+ age,data= train,method="class")
dtfit
```

```
n= 291844

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 291844 23615 0 (0.91908348 0.08091652) *
```

8. Understanding of the model

```
summary(dtfit)
```

```
Call:
rpart(formula = explored ~ institute + final_cc_cname_DI + LoE_DI +
    gender + age, data = train, method = "class")
  n= 291844

  CP nsplit rel error xerror xstd
1  0      0         1      0    0

Node number 1: 291844 observations
  predicted class=0  expected loss=0.08091652  P(node) =1
    class counts: 268229 23615
   probabilities: 0.919 0.081
```

9. Predicting on the test data

```
predicted<-predict(dtfit,test,type="class")
```

10. Output of confusion matrix

```
confusionMatrix(data=predicted, reference=test$explored)
```

```
Confusion Matrix and Statistics

          Reference
Prediction      0      1
         0 116337   8740
         1      0      0

               Accuracy : 0.9301
                 95% CI : (0.9287, 0.9315)
    No Information Rate : 0.9301
    P-Value [Acc > NIR] : 0.5028

                  Kappa : NA
 Mcnemar's Test P-Value : <2e-16

          Sensitivity : 1.0000
          Specificity : 0.0000
       Pos Pred Value : 0.9301
       Neg Pred Value :    NaN
           Prevalence : 0.9301
       Detection Rate : 0.9301
Detection Prevalence : 1.0000
   Balanced Accuracy : 0.5000

    'Positive' Class : 0
```