title: “Prediction of student score” author: “Vishal Jain” date: “04.12.2021”

## Introduction

Presented here is the explanation of the chosen predicted model to predict the students’ score in assessments along with the R code to achieve that.

## Machine Learning Model Chosen: Support Vector Machines with Linear Kernel

Loading the required libraries

library(dplyr)

## Warning: package 'dplyr' was built under R version 4.0.5

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.0.5

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.6 v stringr 1.4.0  
## v tidyr 1.1.4 v forcats 0.5.1  
## v readr 2.1.0

## Warning: package 'ggplot2' was built under R version 4.0.5

## Warning: package 'tibble' was built under R version 4.0.5

## Warning: package 'tidyr' was built under R version 4.0.5

## Warning: package 'readr' was built under R version 4.0.5

## Warning: package 'forcats' was built under R version 4.0.5

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(caret)

## Warning: package 'caret' was built under R version 4.0.5

## Loading required package: lattice

## Warning: package 'lattice' was built under R version 4.0.5

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

library(doSNOW)

## Warning: package 'doSNOW' was built under R version 4.0.5

## Loading required package: foreach

## Warning: package 'foreach' was built under R version 4.0.5

##   
## Attaching package: 'foreach'

## The following objects are masked from 'package:purrr':  
##   
## accumulate, when

## Loading required package: iterators

## Warning: package 'iterators' was built under R version 4.0.5

## Loading required package: snow

## Warning: package 'snow' was built under R version 4.0.5

library(xgboost)

## Warning: package 'xgboost' was built under R version 4.0.5

##   
## Attaching package: 'xgboost'

## The following object is masked from 'package:dplyr':  
##   
## slice

library(ggplot2)

Importing the studentInfo and studentAssessment csv files from the OULAD dataset

#Reading the csv files   
student.info <- read.csv("studentInfo.csv")  
student.assessment <- read.csv("studentAssessment.csv")

The first order of business is merging the relevant variables from both files into a single dataframe.

#Merging the studentInfo.csv and relevant variables from studentAssessment.csv to get more info about the students in the same dataframe  
mergedInfo <- merge(x=student.assessment, y=student.info[,c("id\_student","gender", "age\_band", "studied\_credits")], by="id\_student")

The na.omit() functio can be used to get rid of any rows with missing data

#Omitting any rows with NA values   
mergedInfo <- na.omit(mergedInfo)  
  
  
#Converting any character variables in the merged dataframe to factors  
mergedInfo <-   
 mergedInfo %>%  
 mutate\_if(is.character, as.factor)

It needs to be checked if there are any variables with little or no variability as this can affect the model

nearZeroVar(mergedInfo, saveMetrics = TRUE)

## freqRatio percentUnique zeroVar nzv  
## id\_student 1.045455 1.127566e+01 FALSE FALSE  
## id\_assessment 1.273973 9.078091e-02 FALSE FALSE  
## date\_submitted 1.172365 1.492090e-01 FALSE FALSE  
## is\_banked 57.368658 9.657544e-04 FALSE TRUE  
## score 1.547896 4.877059e-02 FALSE FALSE  
## gender 1.343385 9.657544e-04 FALSE FALSE  
## age\_band 2.296618 1.448632e-03 FALSE FALSE  
## studied\_credits 2.885048 2.510961e-02 FALSE FALSE

We can see that the nzv field is TRUE for the ‘is\_banked’ varibale, so we need to get rid of it

#Resaving the dataframe as the same name but without the is-banked variable using select() function  
mergedInfo <-  
 mergedInfo %>%  
 select(-is\_banked)

# Setting a seed to ensure the reproducibility  
set.seed(2020)

Creating a trainIndex object that indexes 70% of the data, to follow the widely used 70-30 pattern for training and testing respectively:

trainIndex <- createDataPartition(mergedInfo$score,  
 p = .7,  
 list = FALSE,  
 times = 1)  
  
score.train <- mergedInfo[trainIndex,]  
score.test <- mergedInfo[-trainIndex,]

trainControl function can be used to control the training function later on. Here a 10-fold Cross Validation process is used which is repeated 3 times to increase the accuracy of the model:

train.control <- trainControl(number = 10,  
 repeats = 1,  
 method = "repeatedcv",  
 verboseIter = TRUE)

To make the training process faster, the doSNOW package has been used to train the model parallelly. Here, a cluster with 8 processes has been created, but this number depends on the number of cores/threads on the computer the code is being run on:

cl <- makeCluster(8, type = "SOCK")  
registerDoSNOW(cl)

Training the model using the previously set up control parameters and using the svmLinear model which has been selected by trial and error compared to a number of models including xgbTree, ranger etc.

forestFit <- train(score ~ .,  
 data = score.train,  
 method = "svmLinear",  
 trControl = train.control,  
 verbose = TRUE)

## Aggregating results  
## Fitting final model on full training set

#Clearing the cluster created to free up the memory  
stopCluster(cl)  
  
#Examining the model  
forestFit

## Support Vector Machines with Linear Kernel   
##   
## 144967 samples  
## 6 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 1 times)   
## Summary of sample sizes: 130470, 130471, 130470, 130470, 130470, 130471, ...   
## Resampling results:  
##   
## RMSE Rsquared MAE   
## 19.23126 0.01532217 14.40299  
##   
## Tuning parameter 'C' was held constant at a value of 1

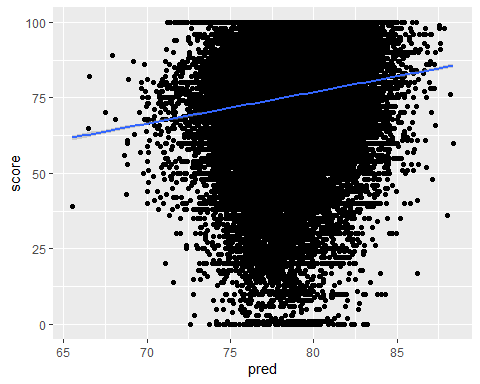
Not the model can be used to predict the scores of the students in the testing module to check the accuracy of the created module:

preds <- predict(forestFit, score.test)  
  
#Creating a new object for the testing data with predicted values included  
score.test.augmented <-  
 score.test %>%  
 mutate(pred = predict(forestFit, score.test),  
 obs = score)  
  
#Transforming this new object into a data frame  
defaultSummary(as.data.frame(score.test.augmented))

## RMSE Rsquared MAE   
## 19.10577461 0.01737013 14.32796812

A scatterplot can be drawn using the predictions against the actual scores in the the testing module.

ggplot(score.test.augmented, aes(x=pred,y=score)) +  
 geom\_point() +  
 geom\_smooth(method='lm', formula= y~x)



## Conclusion

As of right now, this is the best model that has been found with the Rsquared and RMSE values on the testing data being 0.01767867 and 19.10235243 respectively