hospital_readmission_code.R

Abhishek

Mon Sep 30 16:34:28 2019

```
#Hospital Readmission Prediction
#Name: Abhishek Patil

setwd("C:/Users/Abhishek/Desktop/Hospital Readmission Prediction/Challenge")
options(repr.matrix.max.cols=50, repr.matrix.max.rows=100)
options(warn=-1)

#Libraries
library(data.table)
library(xgboost)
library(Matrix)
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(dummies)
```

```
## dummies-1.5.6 provided by Decision Patterns
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
train <- read.csv('challengetraining_data.csv')

#Data Preprocessing includes dropping columns, deleting some rows, changing column types from categoric

#Data Summary before processing.
#Most of the preprocessing steps are based on the results of this summary.
summary(train)
```

```
##   encounter_id        patient_nbr                     race
##  Min.   :    12522   Min.   :      135   ?                : 1813
##  1st Qu.: 85078676   1st Qu.: 23403584   AfricanAmerican:15413
##  Median :152440233   Median : 45531958   Asian           :  513
##  Mean   :165201161   Mean   : 54309594   Caucasian       :60842
##  3rd Qu.:230147157   3rd Qu.: 87504566   Hispanic        : 1617
##  Max.   :443867222   Max.   :189502619   Other           : 1216
##
##              gender           age              weight
##  Female         :43752   [70-80):20890   ?        :78844
##  Male           :37660   [60-70):18059   [75-100) : 1075
```

```
##  Unknown/Invalid:    2   [80-90):13800   [50-75)  :  718
##                          [50-60):13736   [100-125):  509
##                          [40-50): 7708   [125-150):  120
##                          [30-40): 3019   [25-50)  :   70
##                          (Other): 4202   (Other)  :   78
##  admission_type_id discharge_disposition_id admission_source_id
##  Min.   :1.000     Min.   : 1.00            Min.   : 1.000
##  1st Qu.:1.000     1st Qu.: 1.00            1st Qu.: 1.000
##  Median :1.000     Median : 1.00            Median : 7.000
##  Mean   :2.021     Mean   : 3.71            Mean   : 5.754
##  3rd Qu.:3.000     3rd Qu.: 3.00            3rd Qu.: 7.000
##  Max.   :8.000     Max.   :28.00            Max.   :25.000
##
##  time_in_hospital   payer_code                medical_specialty
##  Min.   : 1.000   ?      :32231   ?                     :39935
##  1st Qu.: 2.000   MC     :25945   InternalMedicine      :11774
##  Median : 4.000   HM     : 4990   Emergency/Trauma      : 6073
##  Mean   : 4.398   SP     : 3979   Family/GeneralPractice: 5924
##  3rd Qu.: 6.000   BC     : 3720   Cardiology            : 4255
##  Max.   :14.000   MD     : 2801   Surgery-General       : 2491
##                   (Other): 7748   (Other)               :10962
##  num_lab_procedures num_procedures  num_medications number_outpatient
##  Min.   :  1.00     Min.   :0.000   Min.   : 1.00   Min.   : 0.0000
##  1st Qu.: 31.00     1st Qu.:0.000   1st Qu.:10.00   1st Qu.: 0.0000
##  Median : 44.00     Median :1.000   Median :15.00   Median : 0.0000
##  Mean   : 43.16     Mean   :1.339   Mean   :16.03   Mean   : 0.3657
##  3rd Qu.: 57.00     3rd Qu.:2.000   3rd Qu.:20.00   3rd Qu.: 0.0000
##  Max.   :132.00     Max.   :6.000   Max.   :81.00   Max.   :42.0000
##
##  number_emergency  number_inpatient      diag_1          diag_2
##  Min.   : 0.0000   Min.   : 0.0000   428    : 5423   276    : 5422
##  1st Qu.: 0.0000   1st Qu.: 0.0000   414    : 5209   428    : 5318
##  Median : 0.0000   Median : 0.0000   786    : 3225   250    : 4881
##  Mean   : 0.1994   Mean   : 0.6358   410    : 2892   427    : 4008
##  3rd Qu.: 0.0000   3rd Qu.: 1.0000   486    : 2834   401    : 2978
##  Max.   :76.0000   Max.   :21.0000   427    : 2214   599    : 2664
##                                      (Other):59617   (Other):56143
##      diag_3      number_diagnoses max_glu_serum A1Cresult
##  250    : 9290   Min.   : 1.000   >200: 1196    >7  : 3047
##  401    : 6610   1st Qu.: 6.000   >300: 1011    >8  : 6594
##  276    : 4163   Median : 8.000   None:77166    None:67762
##  428    : 3670   Mean   : 7.422   Norm: 2041    Norm: 4011
##  427    : 3155   3rd Qu.: 9.000
##  414    : 2923   Max.   :16.000
##  (Other):51603
##   metformin     repaglinide    nateglinide    chlorpropamide
##  Down  :  457   Down  :   35   Down  :    9   Down  :    1
##  No    :65501   No    :80187   No    :80844   No    :81340
##  Steady:14615   Steady: 1096   Steady:  540   Steady:   67
##  Up    :  841   Up    :   96   Up    :   21   Up    :    6
##
##
##
##  glimepiride    acetohexamide   glipizide      glyburide
```

```
##  Down  :  160   No    :81413   Down  :  439   Down  :  450
##  No    :77279   Steady:    1   No    :71156   No    :72929
##  Steady: 3700                  Steady: 9211   Steady: 7378
##  Up    :  275                  Up    :  608   Up    :  657
##
##
##
##  tolbutamide    pioglitazone   rosiglitazone    acarbose
##  No    :81398   Down  :   99   Down  :   62   Down  :    3
##  Steady:   16   No    :75508   No    :76314   No    :81164
##                 Steady: 5616   Steady: 4891   Steady:  238
##                 Up    :  191   Up    :  147   Up    :    9
##
##
##
##    miglitol     troglitazone    tolazamide    examide    citoglipton
##  Down  :    4   No    :81412   No    :81388   No:81414   No:81414
##  No    :81382   Steady:    2   Steady:   26
##  Steady:   27
##  Up    :    1
##
##
##
##    insulin      glyburide.metformin glipizide.metformin
##  Down  : 9841   Down  :    3        No    :81405
##  No    :37803   No    :80854        Steady:    9
##  Steady:24682   Steady:  549
##  Up    : 9088   Up    :    8
##
##
##
##  glimepiride.pioglitazone metformin.rosiglitazone metformin.pioglitazone
##  No    :81413             No    :81412            No    :81413
##  Steady:    1             Steady:    2            Steady:    1
##
##
##
##
##
##  change     diabetesMed readmitted
##  Ch:37656   No :18681   N:72328
##  No:43758   Yes:62733   Y: 9086
##
##
##
##
##
```

```
#Defining a function for preprocessing
preprocessing <- function(train)
  {
  #Dropping the ID columns
  train$encounter_id <- NULL
  train$patient_nbr <- NULL
```

```r
#Dealing with Special Characters (Replacing "?" with NA values)
train[train == "?"] <- NA

#Converting Race to numeric
train$race <- as.numeric(as.factor(train$race))

#Converting Age ranges into numeric values
train$age <- ifelse(train$age == "[0-10)",  0, train$age)
train$age <- ifelse(train$age == "[10-20)", 1, train$age)
train$age <- ifelse(train$age == "[20-30)", 2, train$age)
train$age <- ifelse(train$age == "[30-40)", 3, train$age)
train$age <- ifelse(train$age == "[40-50)", 4, train$age)
train$age <- ifelse(train$age == "[50-60)", 5, train$age)
train$age <- ifelse(train$age == "[60-70)", 6, train$age)
train$age <- ifelse(train$age == "[70-80)", 7, train$age)
train$age <- ifelse(train$age == "[80-90)", 8, train$age)
train$age <- ifelse(train$age == "[90-100)", 9, train$age)
train$age <- as.numeric(train$age)

#Converting Gender to numeric
train <- train[!is.na(train$gender), ] #Dropping rows with NA values in Gender (2 rows of Unknown/Inv
train$gender <- as.numeric(as.factor(train$gender))

##Converting Weight to numeric
train$weight <- ifelse(train$weight == "[0-25)", 0, train$weight)
train$weight <- ifelse(train$weight == "[25-50)", 1, train$weight)
train$weight <- ifelse(train$weight == "[50-75)", 2, train$weight)
train$weight <- ifelse(train$weight == "[75-100)", 3, train$weight)
train$weight <- ifelse(train$weight == "[100-125)", 4, train$weight)
train$weight <- ifelse(train$weight == "[125-150)", 5, train$weight)
train$weight <- ifelse(train$weight == "[150-175)", 6, train$weight)
train$weight <- ifelse(train$weight == "[175-200)", 7, train$weight)
train$weight <- ifelse(train$weight == ">200", 8, train$weight)
train$weight <- as.numeric(train$weight)

#Converting the following columns to numeric/factors as applicable
train$admission_type_id <- as.numeric(as.factor(train$admission_type_id))
train$discharge_disposition_id <- as.numeric(as.factor(train$discharge_disposition_id))
train$admission_source_id <- as.numeric(as.factor(train$admission_source_id))
train$time_in_hospital <- as.numeric(train$time_in_hospital)
train$payer_code <- as.numeric(as.factor(train$payer_code))
train$medical_specialty <- as.numeric(as.factor(train$medical_specialty))
train$num_lab_procedures <- as.numeric(train$num_lab_procedures)
train$num_procedures <- as.numeric(train$num_procedures)
train$num_medications <- as.numeric(train$num_medications)
train$number_outpatient <- as.numeric(train$number_outpatient)
train$number_emergency <- as.numeric(train$number_emergency)
train$number_inpatient <- as.numeric(train$number_inpatient)
train$diag_1 <- as.numeric(as.factor(train$diag_1))
train$diag_2 <- as.numeric(as.factor(train$diag_2))
train$diag_3 <- as.numeric(as.factor(train$diag_3))
train$number_diagnoses <- as.numeric(train$number_diagnoses)
```

```r
#Converting max_glu_serum to numeric
train$max_glu_serum <- ifelse(train$max_glu_serum == "None",  0, train$max_glu_serum)
train$max_glu_serum <- ifelse(train$max_glu_serum == "Norm",  1, train$max_glu_serum)
train$max_glu_serum <- ifelse(train$max_glu_serum == ">200",  2, train$max_glu_serum)
train$max_glu_serum <- ifelse(train$max_glu_serum == ">300",  3, train$max_glu_serum)
train$max_glu_serum <- as.numeric(train$max_glu_serum)

#Converting A1Cresult to numeric
train$A1Cresult <- ifelse(train$A1Cresult == "None",  0, train$A1Cresult)
train$A1Cresult <- ifelse(train$A1Cresult == "Norm",  1, train$A1Cresult)
train$A1Cresult <- ifelse(train$A1Cresult == ">7",    2, train$A1Cresult)
train$A1Cresult <- ifelse(train$A1Cresult == ">8",    3, train$A1Cresult)
train$A1Cresult <- as.numeric(train$A1Cresult);

#Columns with over half of the data missing
drops <- c("weight", "payer_code", "medical_specialty")
train <- train[ , !(names(train) %in% drops)]

#Columns having the same value throughout
drops <- c("examide", "citoglipton")
train <- train[ , !(names(train) %in% drops)]

#Columns with very imbalanced categories
drops <- c("chlorpropamide", "acetohexamide", "tolbutamide", "acarbose", "miglitol", "troglitazone",
           "glimepiride.pioglitazone", "metformin.rosiglitazone", "metformin.pioglitazone", "nateglini
train <- train[ , !(names(train) %in% drops)]

#Columns with Numeric and String values
#Can be converted to numeric. Reference: (https://en.wikipedia.org/wiki/List_of_ICD-9_codes)
#Due to limitation of time, dropping it.
drops <- c("diag_1", "diag_2", "diag_3")
train <- train[ , !(names(train) %in% drops)]

#Converting change to numeric
train$change <- as.character(train$change)
train$change [train$change  == "Ch"] <- 1
train$change [train$change  == "No"] <- 0
train$change  <- as.numeric(train$change)

#Converting diabetesMed to numeric
train$diabetesMed  <- as.character(train$diabetesMed)
train$diabetesMed [train$diabetesMed  == "Yes"] <- 1
train$diabetesMed [train$diabetesMed  == "No"] <- 0
train$diabetesMed  <- as.numeric(train$diabetesMed)

#Converting metformin, repaglinide, glimepiride, glipizide, glyburide, pioglitazone, rosiglitazone, i
train$metformin <- as.character(train$metformin)
train$repaglinide <- as.character(train$repaglinide)
train$glimepiride <- as.character(train$glimepiride)
train$glipizide <- as.character(train$glipizide)
train$glyburide <- as.character(train$glyburide)
train$pioglitazone <- as.character(train$pioglitazone)
train$rosiglitazone <- as.character(train$rosiglitazone)
```

```
  train$insulin <- as.character(train$insulin)
  train[train == "Down"] <- -1
  train[train == "No"] <- 0
  train[train == "Steady"] <- 1
  train[train == "Up"] <- 2
  train$metformin <- as.integer(train$metformin)
  train$repaglinide <- as.numeric(train$repaglinide)
  train$glimepiride <- as.numeric(train$glimepiride)
  train$glipizide <- as.numeric(train$glipizide)
  train$glyburide <- as.numeric(train$glyburide)
  train$pioglitazone <- as.numeric(train$pioglitazone)
  train$rosiglitazone <- as.numeric(train$rosiglitazone)
  train$insulin <- as.numeric(train$insulin)

  return(train)
}

#Calling the defined function for data preprocessing
train <- preprocessing(train)

#Converting readmitted to numeric
train$readmitted <- as.character(train$readmitted)
train$readmitted[train$readmitted == "Y"] <- 1
train$readmitted[train$readmitted == "N"] <- 0
train$readmitted <- as.numeric(train$readmitted)

#Data Summary after processing
summary(train)
```

```
##       race           gender           age         admission_type_id
##  Min.   :2.000   Min.   :1.000   Min.   : 0.000   Min.   :1.000
##  1st Qu.:4.000   1st Qu.:1.000   1st Qu.: 6.000   1st Qu.:1.000
##  Median :4.000   Median :1.000   Median : 7.000   Median :1.000
##  Mean   :3.657   Mean   :1.463   Mean   : 7.099   Mean   :2.021
##  3rd Qu.:4.000   3rd Qu.:2.000   3rd Qu.: 8.000   3rd Qu.:3.000
##  Max.   :6.000   Max.   :3.000   Max.   :10.000   Max.   :8.000
##  NA's   :1813
##  discharge_disposition_id admission_source_id time_in_hospital
##  Min.   : 1.000           Min.   : 1.000      Min.   : 1.000
##  1st Qu.: 1.000           1st Qu.: 1.000      1st Qu.: 2.000
##  Median : 1.000           Median : 7.000      Median : 4.000
##  Mean   : 3.673           Mean   : 5.547      Mean   : 4.398
##  3rd Qu.: 3.000           3rd Qu.: 7.000      3rd Qu.: 6.000
##  Max.   :26.000           Max.   :17.000      Max.   :14.000
##
##  num_lab_procedures num_procedures  num_medications number_outpatient
##  Min.   :  1.00     Min.   :0.000   Min.   : 1.00   Min.   : 0.0000
##  1st Qu.: 31.00     1st Qu.:0.000   1st Qu.:10.00   1st Qu.: 0.0000
##  Median : 44.00     Median :1.000   Median :15.00   Median : 0.0000
##  Mean   : 43.16     Mean   :1.339   Mean   :16.03   Mean   : 0.3657
##  3rd Qu.: 57.00     3rd Qu.:2.000   3rd Qu.:20.00   3rd Qu.: 0.0000
##  Max.   :132.00     Max.   :6.000   Max.   :81.00   Max.   :42.0000
##
##  number_emergency  number_inpatient  number_diagnoses max_glu_serum
```

```
##  Min.   : 0.0000   Min.   : 0.0000   Min.   : 1.000   Min.   :0.0000
##  1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 6.000   1st Qu.:0.0000
##  Median : 0.0000   Median : 0.0000   Median : 8.000   Median :0.0000
##  Mean   : 0.1994   Mean   : 0.6358   Mean   : 7.422   Mean   :0.1398
##  3rd Qu.: 0.0000   3rd Qu.: 1.0000   3rd Qu.: 9.000   3rd Qu.:0.0000
##  Max.   :76.0000   Max.   :21.0000   Max.   :16.000   Max.   :4.0000
##
##    A1Cresult        metformin        repaglinide        glimepiride
##  Min.   :0.0000   Min.   :-1.0000   Min.   :-1.00000   Min.   :-1.00000
##  1st Qu.:0.0000   1st Qu.: 0.0000   1st Qu.: 0.00000   1st Qu.: 0.00000
##  Median :0.0000   Median : 0.0000   Median : 0.00000   Median : 0.00000
##  Mean   :0.3965   Mean   : 0.1946   Mean   : 0.01539   Mean   : 0.05024
##  3rd Qu.:0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.00000   3rd Qu.: 0.00000
##  Max.   :4.0000   Max.   : 2.0000   Max.   : 2.00000   Max.   : 2.00000
##
##    glipizide         glyburide        pioglitazone      rosiglitazone
##  Min.   :-1.0000   Min.   :-1.0000   Min.   :-1.00000   Min.   :-1.00000
##  1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 0.00000   1st Qu.: 0.00000
##  Median : 0.0000   Median : 0.0000   Median : 0.00000   Median : 0.00000
##  Mean   : 0.1227   Mean   : 0.1012   Mean   : 0.07246   Mean   : 0.06293
##  3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.00000   3rd Qu.: 0.00000
##  Max.   : 2.0000   Max.   : 2.0000   Max.   : 2.00000   Max.   : 2.00000
##
##     insulin            change        diabetesMed       readmitted
##  Min.   :-1.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.: 0.0000   1st Qu.:0.0000   1st Qu.:1.0000   1st Qu.:0.0000
##  Median : 0.0000   Median :0.0000   Median :1.0000   Median :0.0000
##  Mean   : 0.4055   Mean   :0.4625   Mean   :0.7705   Mean   :0.1116
##  3rd Qu.: 1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   : 2.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##
```

```
df <- train

#Train-Test Split
set.seed(888)
train.index <- sample(nrow(df), nrow(df)*0.7)
train.df <- df[train.index,]
valid.df <- df[-train.index,]

X_train <- train.df
X_test <- valid.df
y_train <- train.df$readmitted
y_test <- valid.df$readmitted

X_train$readmitted = NULL
X_test$readmitted = NULL

#### XGBoost Classifier ####
X_train <- as.matrix(X_train)
X_test <- as.matrix(X_test)
y_train <- as.matrix(y_train)
y_test <- as.matrix(y_test)
```

```
dtrain <- xgb.DMatrix(data = X_train,label = y_train)
dtest <- xgb.DMatrix(data = X_test,label=y_test)

#Since it is an imbalanced dataset, considering AUC as the evaluation metric.

params <- list(
  booster = "gbtree",
  objective = "binary:logistic",
  max_depth = 3,
  eta = 0.4,
  eval_metric = "auc"
)

xgbcv <- xgb.cv( params = params,
                 data = dtrain,
                 nrounds = 200,
                 nfold = 10,
                 stratified = T,
                 print_every_n = 20,
                 early_stopping_rounds = 10
)
```

```
## [1]  train-auc:0.632905+0.000756 test-auc:0.632612+0.006765
## Multiple eval metrics are present. Will use test_auc for early stopping.
## Will train until test_auc hasn't improved in 10 rounds.
##
## [21] train-auc:0.682825+0.000853 test-auc:0.667658+0.006269
## Stopping. Best iteration:
## [27] train-auc:0.688099+0.001063 test-auc:0.668556+0.005922
```

```
xgb1 <- xgb.train (
  params = params,
  data = dtrain,
  watchlist = list(val=dtest,train=dtrain),
  print_every_n = 10,
  nrounds = 200,
  early_stopping_rounds = 10,
  seed = 100
)
```

```
## [1]  val-auc:0.633137    train-auc:0.632560
## Multiple eval metrics are present. Will use train_auc for early stopping.
## Will train until train_auc hasn't improved in 10 rounds.
##
## [11] val-auc:0.654852    train-auc:0.664354
## [21] val-auc:0.665033    train-auc:0.680847
## [31] val-auc:0.666258    train-auc:0.689207
## [41] val-auc:0.665006    train-auc:0.694311
## [51] val-auc:0.667223    train-auc:0.699320
## [61] val-auc:0.666226    train-auc:0.703754
## [71] val-auc:0.666322    train-auc:0.708023
## [81] val-auc:0.665463    train-auc:0.710764
## [91] val-auc:0.664629    train-auc:0.712724
## [101]    val-auc:0.663771    train-auc:0.715645
## [111]    val-auc:0.665319    train-auc:0.718856
```

```
## [121]    val-auc:0.663694    train-auc:0.721602
## [131]    val-auc:0.663487    train-auc:0.723715
## [141]    val-auc:0.663619    train-auc:0.726468
## [151]    val-auc:0.662627    train-auc:0.727665
## [161]    val-auc:0.662413    train-auc:0.730061
## [171]    val-auc:0.661782    train-auc:0.732111
## [181]    val-auc:0.661592    train-auc:0.735737
## [191]    val-auc:0.661509    train-auc:0.738210
## [200]    val-auc:0.661231    train-auc:0.739805
```

```
#Evaluation

#Training Accuracy
xgbpred_train <- predict (xgb1,dtrain)
#Threshold was set according to the accuracy score used
xgbpred_train <- ifelse (xgbpred_train > 0.12,1,0)
myroc <- roc(y_train, xgbpred_train)
cat("Training Accuracy: ", auc(myroc))
```

```
## Training Accuracy:  0.6719187
```

```
#Testing Accuracy
xgbpred_test <- predict (xgb1,dtest)

#Threshold was set according to the accuracy score used
xgbpred_test <- ifelse (xgbpred_test > 0.12,1,0)
myroc <- roc(y_test, xgbpred_test)
cat("Testing Accuracy: ", auc(myroc))
```

```
## Testing Accuracy:  0.6170448
```

```
###### Final Model ######
#Train on the whole data
X_train <- df
y_train <- df$readmitted

X_train$readmitted = NULL

X_train <- as.matrix(X_train)
y_train <- as.matrix(y_train)

dtrain_whole <- xgb.DMatrix(data = X_train,label = y_train)

xgbpred <- predict (xgb1, dtrain_whole)
#Threshold was set according to the accuracy score used
xgbpred <- ifelse (xgbpred > 0.12,1,0)
myroc <- roc(y_train, xgbpred)
cat("Final Model Accuracy: ", auc(myroc))
```

```
## Final Model Accuracy:  0.6557635
```

```
#Feature Importances
#mat <- xgb.importance (feature_names = colnames(X_train),model = xgb1)

#The plot shows the top 10 important features for this model.
#xgb.plot.importance (importance_matrix = mat[1:15])
```

```r
#Commenting the code for the plot as the markdown had problems displaying the plot
#The plot gives some interesting insights. Variables like number_inpatient, nu_lab_procedures, num_medi
```

```r
#Prediction

#Reading the test file
test <- read.csv('challengetest_data.csv')

#Creating a new dataframe for probabilities
predicted_probability <- data.frame("encounter_id" = test$encounter_id)

#Calling the preprocessing function
test <- preprocessing(test)

#Creating a matrix for XGB
dtest_final <- xgb.DMatrix(data = as.matrix(test))

#Using the XGB model to predict probability
xgbpred_final_test <- predict (xgb1, dtest_final)

#Adding a column of probability to the new dataframe
predicted_probability$predicted_probability <- xgbpred_final_test

#Writing to a CSV file
write.csv(predicted_probability, file = "patil_abhishek.csv")

#The accuracy is not great but certainly better than a random guess.

#Some of the things I would have loved to try out but couldn't due to limited time:
#1. EDA to visualize the patterns among the variables and their relationship with the dependent variabl
#2. Correlation Plot Analysis, Hypothesis testing.
#3. Detailed Feature Engineering (Using Dummy Variables, dealing with missing values, etc.)
#4. Try out different models with Grid Search to compare performance.

#This was part of a challenge that was to be completed in 3 hours. Hence, this was just a preliminary i
#Any comments on what could be improved in this are appreciated.Thanks.
```