NJIT CS 636

DATA ANALYTICS WITH R

LIVERPOOL ION-SWITCHING DATASET PREDICTION

PROJECT GROUP 9

KOMALDEEP KAUR BHATIA (kb488)

CHANDNI MANDAVIYA (csm45)

SOURAV HELAPALYA ASWATHNARAYAN (sh667)

TWINKLE CHAURASIA (tc449)

Data Preprocessing

In any Machine Learning process, Data Preprocessing is that step in which the data gets transformed, or Encoded, to bring it to such a state that now the machine can easily parse it. In other words, the features of the data can now be easily interpreted by the algorithm.

Here the data is having noise in the form of signal drift, so we tried to remove signal drift manually and then pass the below clean data to our models to get the accuracy.

To clean the data, we divided the data manually to several batches based on similarities between the signal classes.

And from the paper here. We got to know that the data is synthesized. Also "electrophysiological" noise and drift were added. Drift is a signal bias causing the signal to no longer be a horizontal line like batches 2, 7, 8, 9, 10 above.

We tried to remove the parabolic drifts from the data in the classes 7,8,9,10 manually.

Below is the code to transform into clean data.

```r
library(data.table)
library(dplyr)
data <- read.csv("../input/liverpool-ion-switching/train.csv",header=TRUE)
test_data <- read.csv("../input/liverpool-ion-switching/test.csv",header=TRUE)
train2 = copy(data)
a<-500000
b<-600000
train2[a:b,2] <-  train2[a:b, 2] - (3*(train2$time[a:b] - 50)/10)
f <- function(x,low,high,mid){
  return(-((-low+high)/625)*(x-mid)**2+high -low)
}

# CLEAN TRAIN BATCH 7
batch <- 7
a <- 500000*(batch-1)
b <- 500000*batch
train2[a:b,2] <-  train2[a:b, 2] - f(data$time[a:b], -1.817,3.186,325)
# CLEAN TRAIN BATCH 8
batch <- 8
a <- 500000*(batch-1)
b <- 500000*batch
train2[a:b,2] <-  train2[a:b, 2] - f(data$time[a:b],-0.094,4.936,375)
# CLEAN TRAIN BATCH 9
batch = 9;
a <- 500000*(batch-1)
b <- 500000*batch
train2[a:b,2] <-  train2[a:b, 2] - f(data$time[a:b],1.715,6.689,425)
# CLEAN TRAIN BATCH 10
batch = 10;
a <- 500000*(batch-1)
b <- 500000*batch
train2[a:b,2] <-  train2[a:b, 2] - f(data$time[a:b],3.361,8.45,475)
# Training batch 1 and 2(1 Slow Open Channel)
batch <- 1
a <- 500000*(batch-1)
b <- 500000*batch
batch <- 2
c <- 500000*(batch-1)
d <- 500000*batch
abc <- c(train2$signal[a:b],train2$signal[c:d])
X_train <- c()
```

Console

CLEANING THE DATA

MODELING PROCEDURE

By observing the data is divided into classes, hence it is multiclass Classification. Therefore, we tried with different algorithms such as KNN, k means, Random forest, decision tree, LSTM and Naive Bayes. But since the data has multiple classes, the best accuracy we got was in random forest after cleaning and removing the drift from data as shown above.

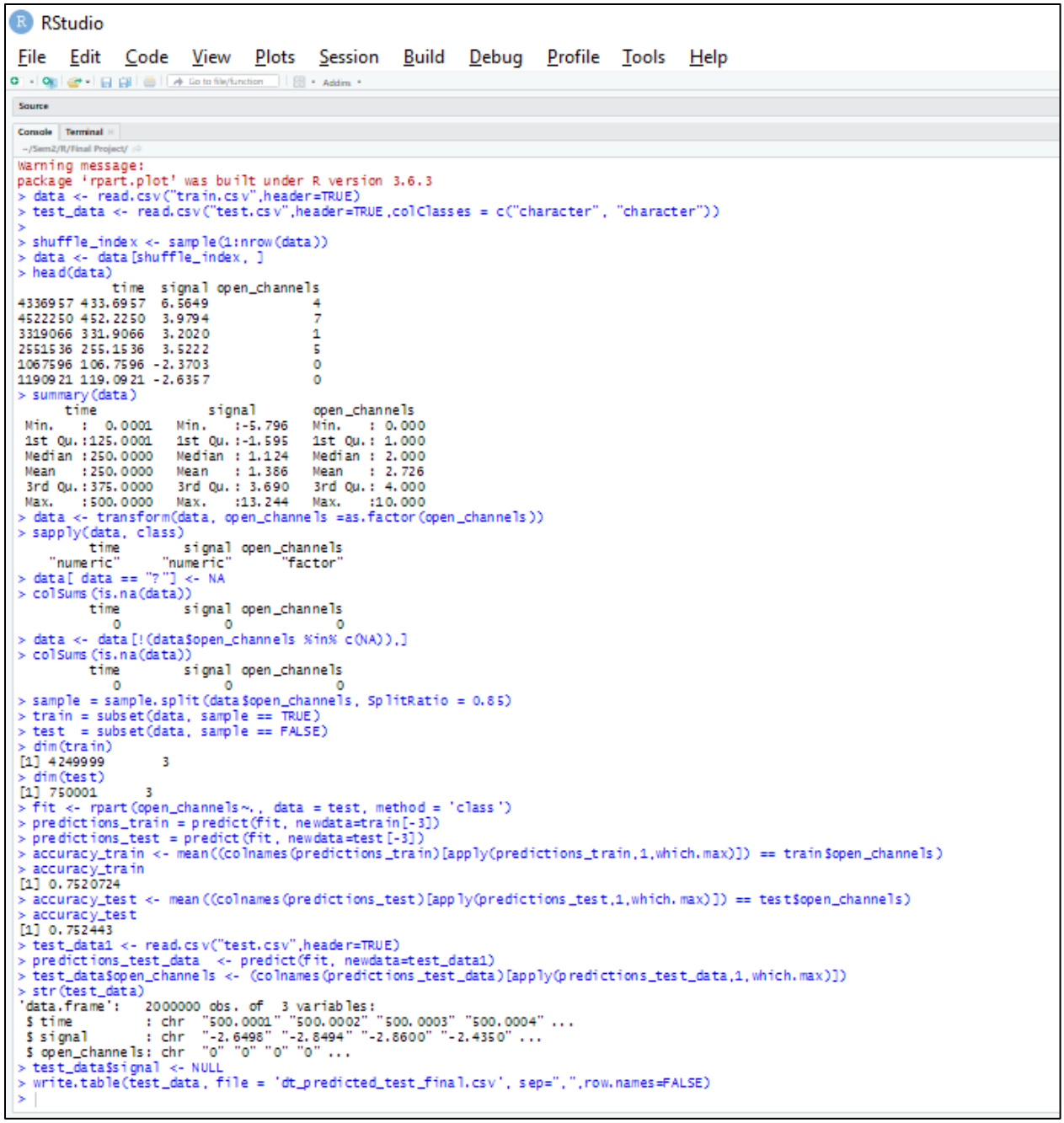
Also, Random Forest models are fast training models and are scalable.

The advantages of random forests include:

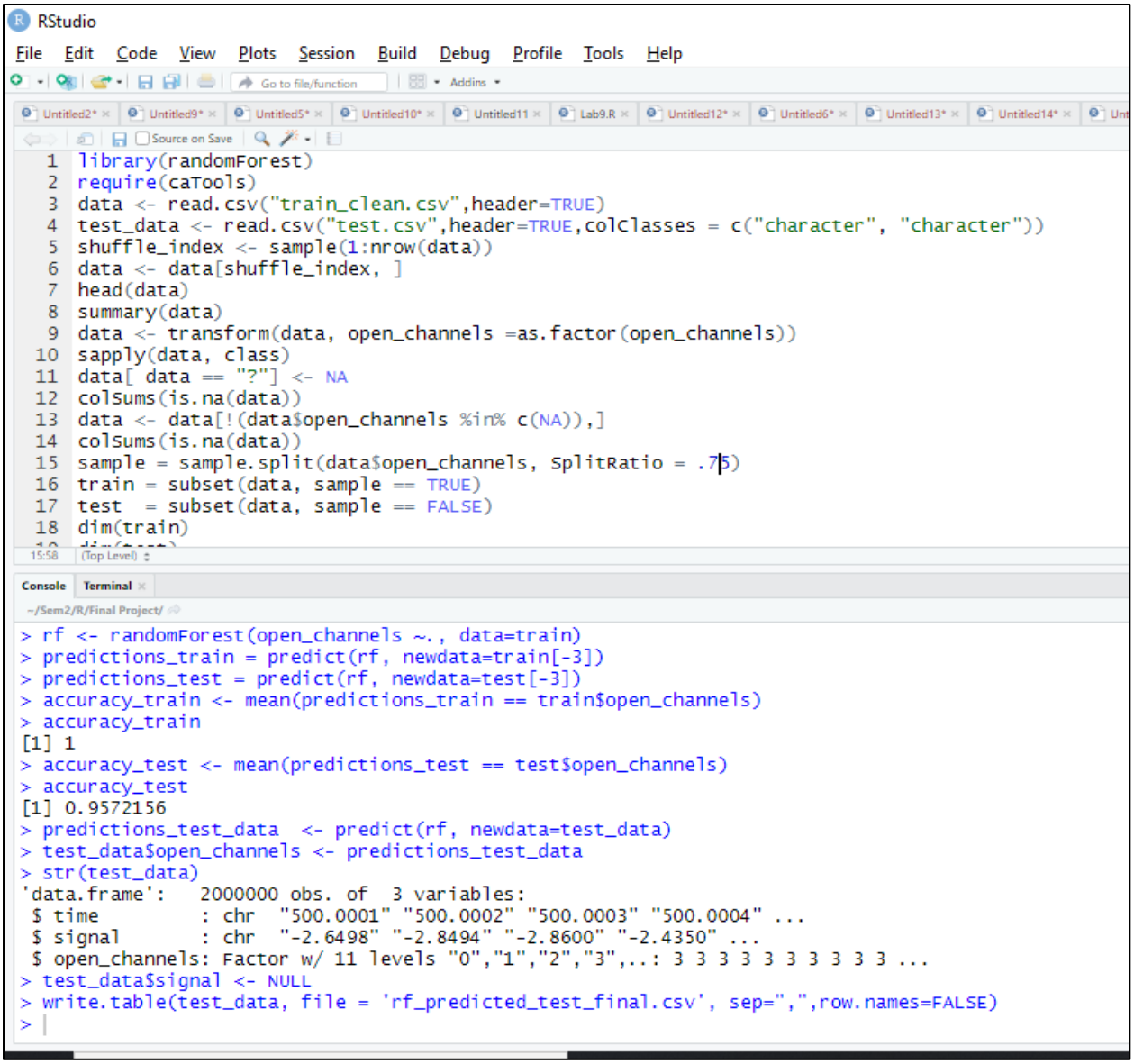
- The predictive performance can compete with the best supervised learning algorithms.
- They provide a reliable feature importance estimate.
- They offer efficient estimates of the test error without incurring the cost of repeated model training associated with cross-validation.

Below are the Models we tried to get better accuracy.

- Decision Tree
- Random Forest.
- K-Means
- Naïve Bayes
- LSTM



DECISION TREE IMPLEMENTATION



RANDOM FOREST IMPLEMENTATION

```
#############K-Means#############################
#loading the require library
library(VIM)
library(tidyverse)
library(factoextra)
#loading the dataset
train <- read.csv('train.csv/train.csv')
test<-read.csv('test.csv/test.csv')

#to check if the data has any missing values or not
#aggr(train)

#distance <- get_dist(train)
k2 <- kmeans(train, centers = 5, nstart = 25)
str(k2)
# fviz_cluster(k2, data = train)

#to find optimal k
set.seed(123)

# function to compute total within-cluster sum of square
wss <- function(k) {
  kmeans(train, k, nstart = 10 )$tot.withinss
}

#k=4 is the elbow point
set.seed(123)
model <- kmeans(train, 4, nstart = 25)

#prediction
fitted(model)
```

K-MEANS IMPLEMENTATION

In [27]:
```
model <- keras_model_sequential()
```

In [39]:
```
model %>%
 layer_dense(units = 1, activation = 'softmax',
 input_shape = 2)
```

In [48]:
```
summary(model)
```

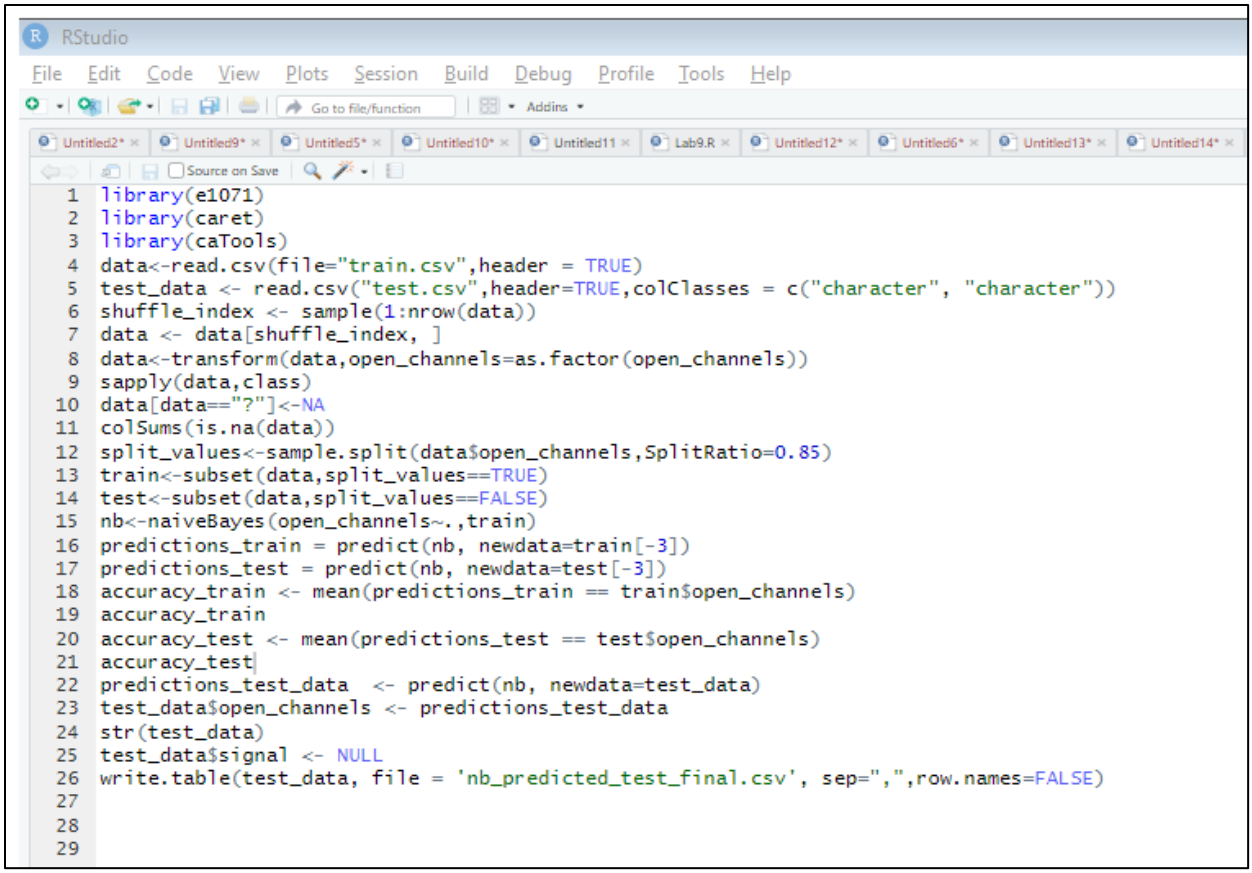
```
Model: "sequential_1"
_________________________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================================
dense_2 (Dense)              (None, 1)                 3
_________________________________________________________________________________
dense_3 (Dense)              (None, 1)                 2
_________________________________________________________________________________
dense_4 (Dense)              (None, 1)                 2
=================================================================================
Total params: 7
Trainable params: 7
Non-trainable params: 0
_________________________________________________________________________________
```

In [41]:
```
sgd <- optimizer_sgd(lr = 0.01)
```

In [46]:
```
model %>% compile(
 loss = 'binary_crossentropy',
 optimizer = sgd,
 metrics = 'accuracy'
)
```

In [ ]:
```
## model %>% fit(
 x = traindata,
 y = trainingtarget,
 epochs = 1,
 batch_size = 3,
 validation_split = 0.2,
 verbose = 0
)
```

LSTM IMPLEMENTATION

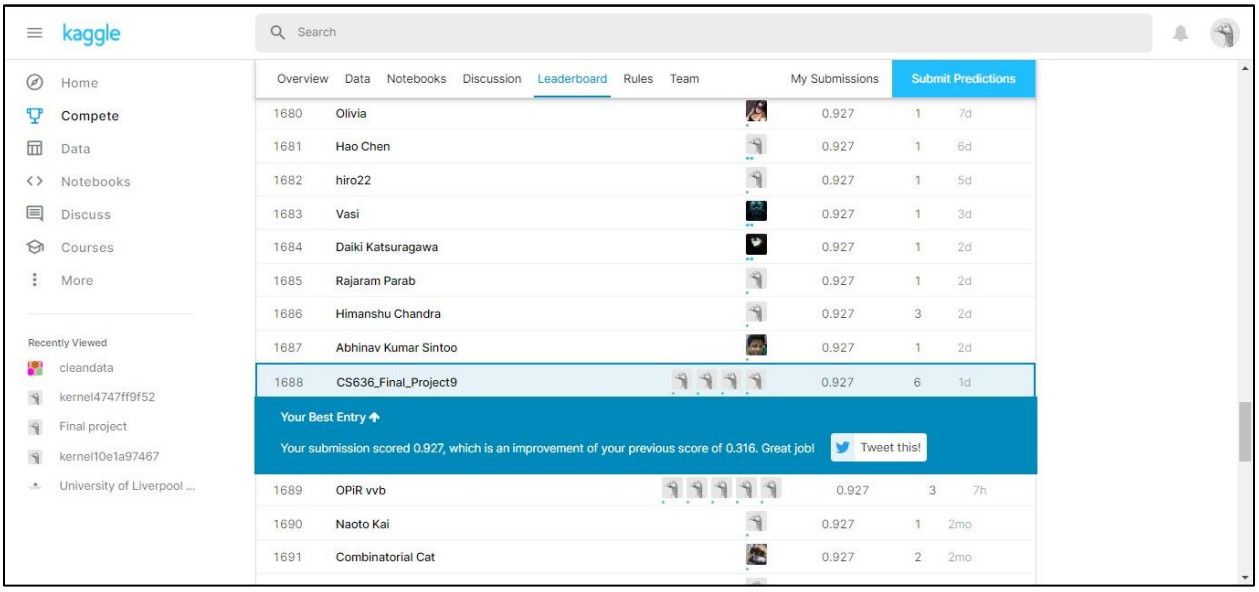


NAÏVE BAYES IMPLEMENTATION

KAGGLE LEADERBOARD SCORE.

Link to my account: https://www.kaggle.com/komaldeepkaurbhatia

Team Name on Kaggle: **CS636_Final_Project9**



KAGGLE LEADERBOARD SCORE

INDIVIDUAL CONTRIBUTIONS

KOMALDEEP KAUR BHATIA (kb488)

Participated in analyzing data and cleaning data for the model. Tried to get the best accuracy using Random Forest algorithm, Decision Tree modelling and XG Boost algorithm. Worked on understanding the distortion of the data and signals. Worked in making the Final report.

CHANDNI MANDAVIYA (csm45)

Tried making model of LSTM and K-Means. Participated in analyzing data and cleaning data for the model. Research on which models will be most fitted for this type of data. Worked in making the Final report.

SOURAV HELAPALYA ASWATHNARAYAN (sh667)

Tired making a model in Decision Trees and k-folds with logistics regression. Helped is developing the code and logic. Tried different methods to find the best accuracy. Participated in analyzing data and cleaning data for the model.

TWINKLE CHAURASIA (tc449)

Tried making the model through the KNN algorithm. Participated in analyzing data and cleaning data for the model. Worked on the Presentation documentation.