Progress Report
On
University of Liverpool - Ion Switching

Data Analytics in R - Final Project
Spring 2020

Submitted by : Group 7

Darshan Shah (ds962)
Gagan Shrivastava
Jasneek Singh Chugh (jc2433)
Rohit Phadke (rp879)
Ashish Kumar

INTRODUCTION

A single ion channel is a pore in the cell membrane that can assume an “open” state in which ions (such as potassium or sodium) can pass through. Ion channels are “gated”, i.e. they open in response to a specific stimulus, such as a change in membrane potential (voltage-gated ion channels) or the binding of a neurotransmitter (ligand-gated ion channels). Ion channels are present in the membranes of all excitable cells. Their functions include establishing a resting membrane potential, shaping action potentials and other electrical signals by gating the flow of ions across the cell membrane, controlling the flow of ions across secretory and epithelial cells, and regulating cell volume.

When ion channels open, they pass electric currents. Existing methods of detecting these state changes are slow and laborious. Humans must supervise the analysis, which imparts considerable bias, in addition to being tedious. These difficulties limit the volume of ion channel current analysis that can be used in research.

The task is to create a model to predict the number of open channels based on electrophysiological signal data, at each timestamp.

Overview of the data:

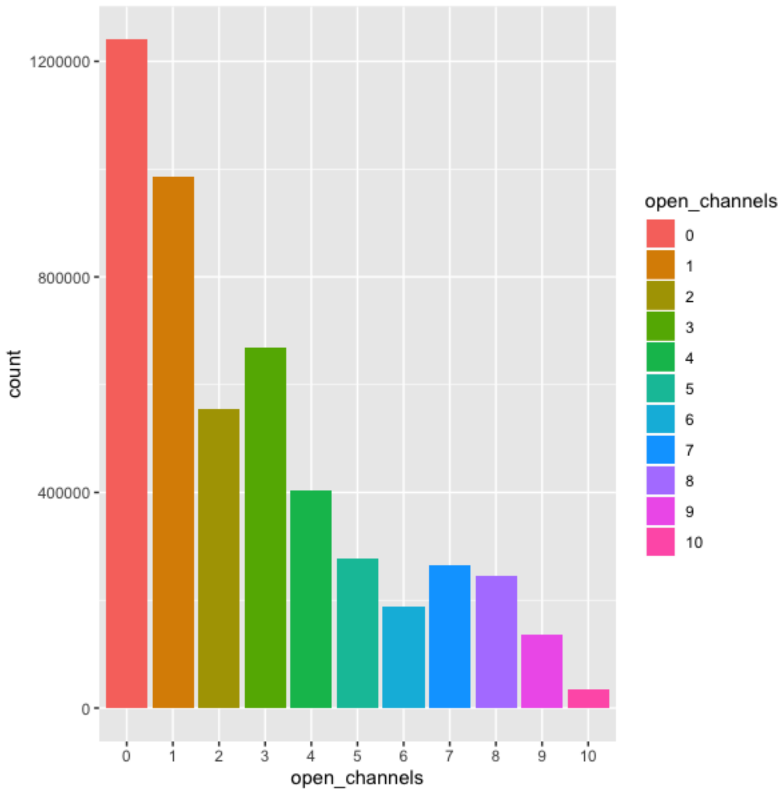
The data was recorded in the batches of 50 seconds. Therefore, each 500,000 rows are in one batch. The training data has 10 batches, total 5,000,000 and the test data has 4 batches, total 2,000,000 records.

We have used the cleaned data, i.e. data without the drift. The drift was removed from the data using the Kalman Filtering.

Link for the cleaned data: https://www.kaggle.com/michaln/kalman-filter-on-clean-data

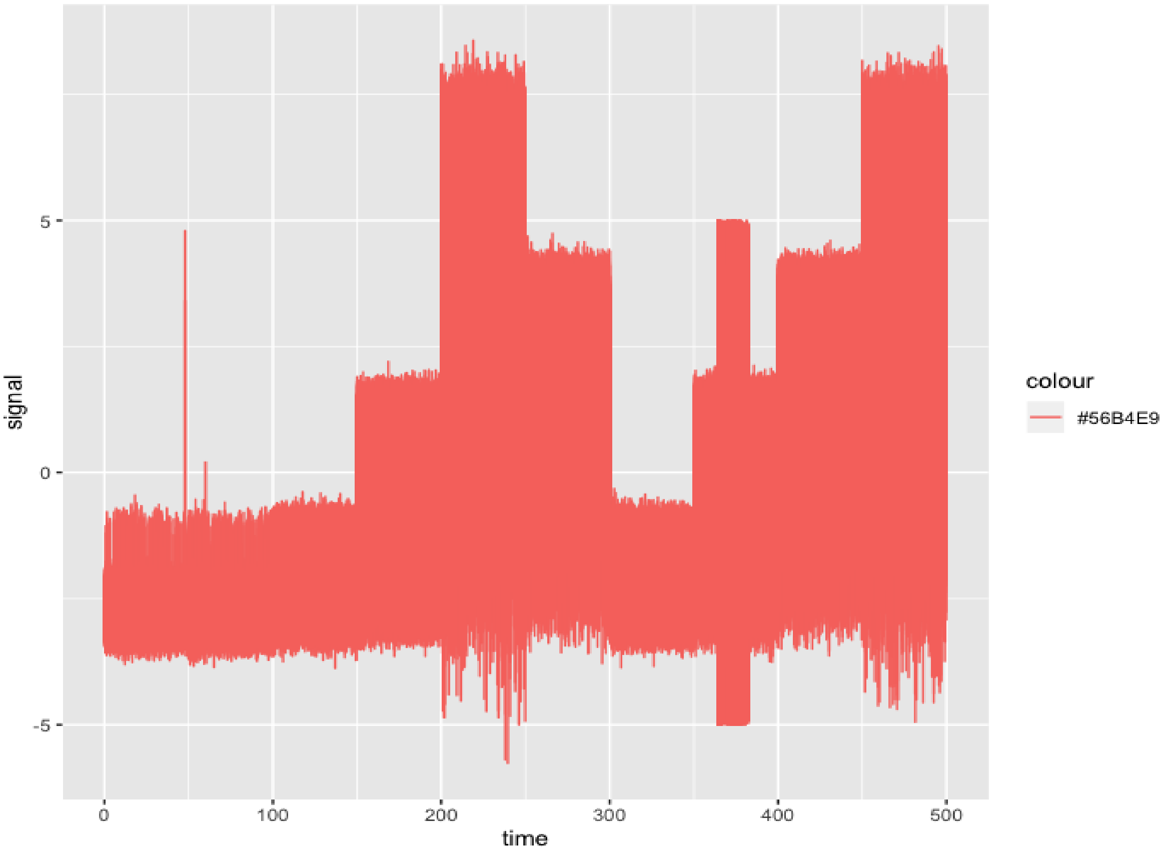
```
#Loading required packages and Libraries
library(data.table)
library(dplyr)
library(ggplot2)
library('scales')
library('grid')
library('gridExtra')
library('RColorBrewer')
library('corrplot')
library('ggridges')
library(caTools)
library(randomForest)
library(glmnet)
library(tidyverse)
library(caret)
library(rpart)

> setwd("/Users/jasneekchugh/Desktop/DataScience/R-Programming/Ion-Switching")

> #reading the required files
> trainData <- fread('data/train_cleaned.csv', sep = ",", header=T)
> testData<- fread('data/test_cleaned.csv', sep = ",", header=T)
> sampleSubmission<- fread('data/sample_submission.csv', sep = ",", header=T)

#summary of the data
> #The data was recorded in batches of 50 seconds. Therefore, there are 500,000 rows per batch.
> str(trainData)
Classes ‘data.table’ and 'data.frame':5000000 obs. of  3 variables:
 $ time        : num  1e-04 2e-04 3e-04 4e-04 5e-04 6e-04 7e-04 8e-04 9e-04 1e-03 ...
 $ signal      : num  -2.76 -2.85 -2.42 -3.13 -3.14 ...
 $ open_channels: int  0 0 0 0 0 0 0 0 0 0 ...
 - attr(*, ".internal.selfref")=<externalptr>
```

```
> head(trainData)#Train data has 10 batches
   time  signal open_channels
1: 1e-04 -2.7607           0
2: 2e-04 -2.8480           0
3: 3e-04 -2.4243           0
4: 4e-04 -3.1300           0
5: 5e-04 -3.1449           0
6: 6e-04 -2.6499           0

> dim(trainData)
[1] 5000000       3

> table(trainData$open_channels) #0-10, 11 levels. Prediction will 11 possible values

      0       1       2       3       4       5       6       7       8       9      10 
1240152  985865  553924  668609  403410  277877  188112  265015  245183  136120 
35733

> head(testData) #Test data has 4 batches
       time  signal
1: 500.0001 -2.6513
2: 500.0002 -2.8466
3: 500.0003 -2.8538
4: 500.0004 -2.4438
5: 500.0005 -2.6125
6: 500.0006 -2.5692

> dim(testData)#2000000, 2
[1] 2000000       2

> #reformatting
> trainData <- trainData %>%
+   mutate(open_channels = factor(open_channels)) %>%
+   mutate(signal= as.numeric(signal))

> #EDA
> ggplot(data=trainData)+geom_bar(aes(x=open_channels, fill= open_channels))
```

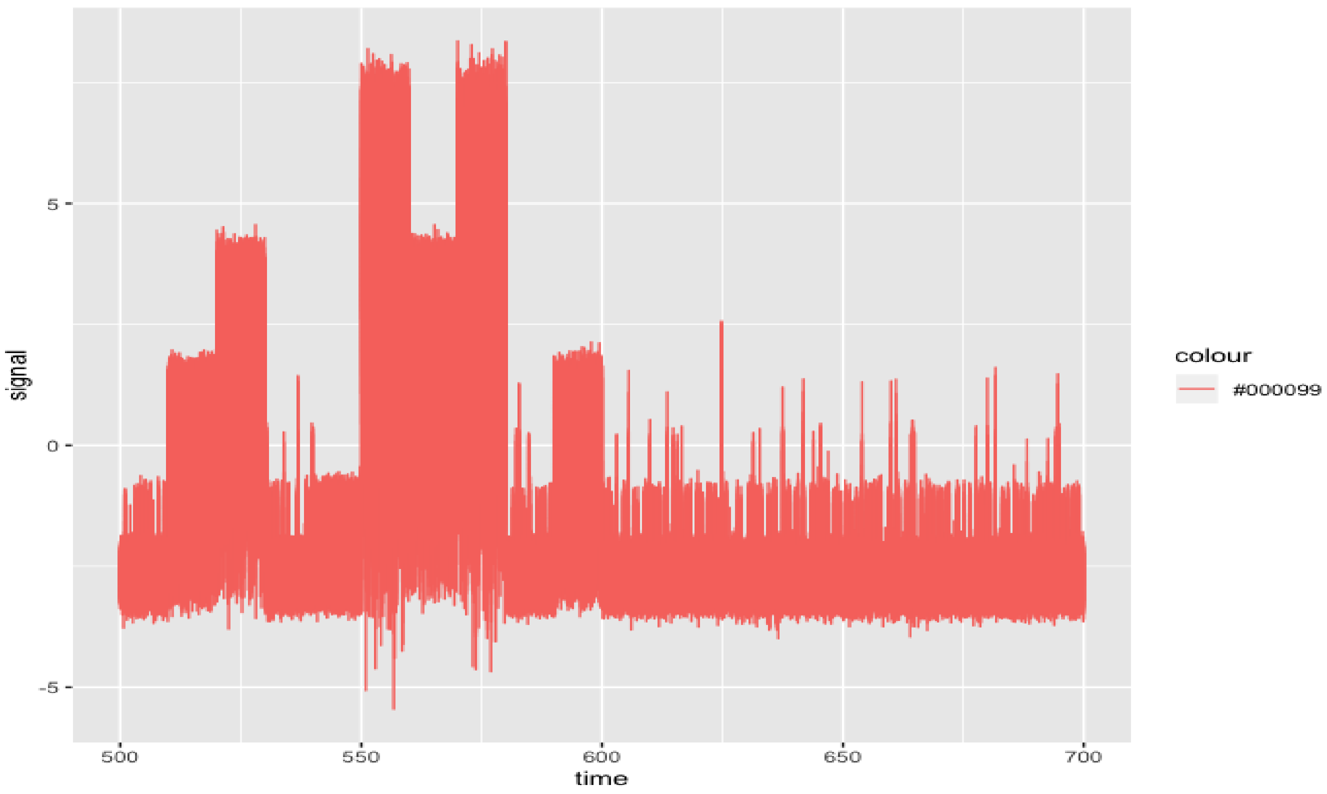


#Time VS Signal
#Train Data

```
> ggplot(data=trainData)+geom_line(aes(x=time,y=signal, color="#56B4E9"))
```



```
#Test Data
> ggplot(data=testData)+geom_line(aes(x=time,y=signal, color="#000099"))
```



MODEL:

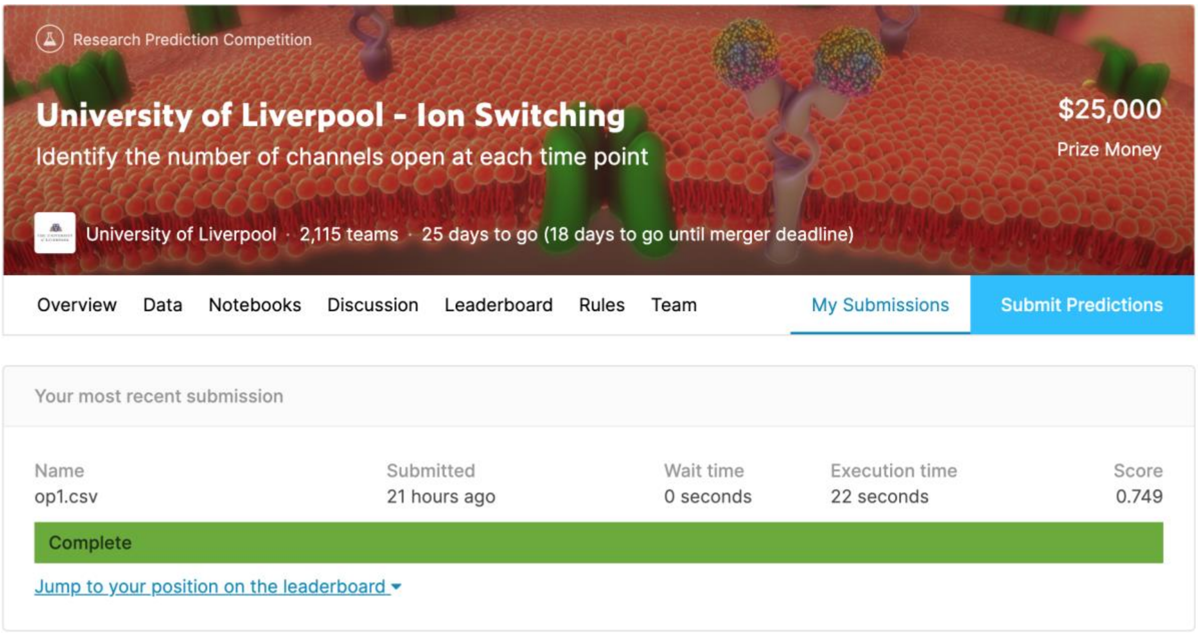
We have used Random Forest model to predict the Open Channels. We have used it with 10% bagging of data. We choose to use Random Forest model because the data set was very heavy and it was unbalanced. The Random Forest model has fast training speed and quick prediction capability and can handle unbalanced data as it tries to minimize the overall error rate.

As model creation and prediction was done separately please refer to the attached R file "Ion-Channel_RF.R" for code.

NEXT STEPS:

- We will implementing the model using Naïve Bayes and Logistic Regression.
- We will be looking for Outliers and also balance the data.
- We will be definitely doing more EDA to make more predictions and increase the accuracy.
- We will be doing hyperparameter tuning and Feature engineering.
- Last, but not the least will be creating more interactive visualizations.

KAGGLE SCREENSHOT:


Research Prediction Competition
University of Liverpool - Ion Switching
Identify the number of channels open at each time point
$25,000
Prize Money
University of Liverpool · 2,115 teams · 25 days to go (18 days to go until merger deadline)
Overview
Data
Notebooks
Discussion
Leaderboard
Rules
Team
My Submissions
Submit Predictions

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
op1.csv	21 hours ago	0 seconds	22 seconds	0.749

Complete

Jump to your position on the leaderboard