Mouse Project Final Report

Yujia Wang, Yuyang Li, Yuchen Liu

5/12/2022

Introduction

Our client is Alberto Cruz-Martín, a Ph.D. in the Department of Anatomy & Neurobiology at Boston University School of Medicine. He studies the neural circuit mechanisms of social behavior. In this project, Alberto Cruz-Martín wants us to use his research data to find the relationships between cells activation and mice behavior. He did three experiments on the mice to provide the data for us. First is the elevated zero mazes. Mice are placed in the Elevated Zero Maze to explore for 10 minutes. They can either be in the closed arm (anxiolytic) or the open arm (anxiogenic). Then is the opposite sex. Mice are placed in a social chamber for 10 minutes and allowed two explore two cups containing a male and female of the same strain. Finally is the direct interaction. Mice are placed in a social chamber for 5 minutes and are allowed to freely interact with a novel mouse of the same strain. We did data cleaning, and EDA, and used different models to analyze the dataset in Python.

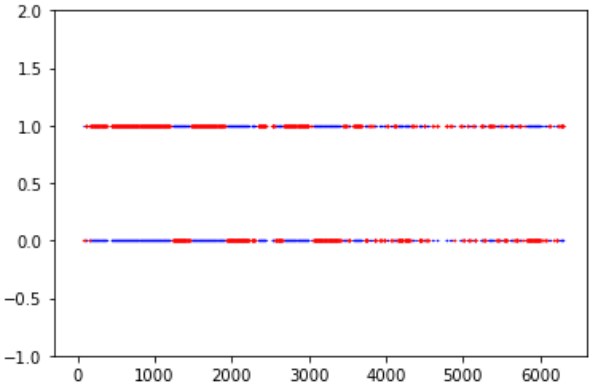
Data and Method

Row Data

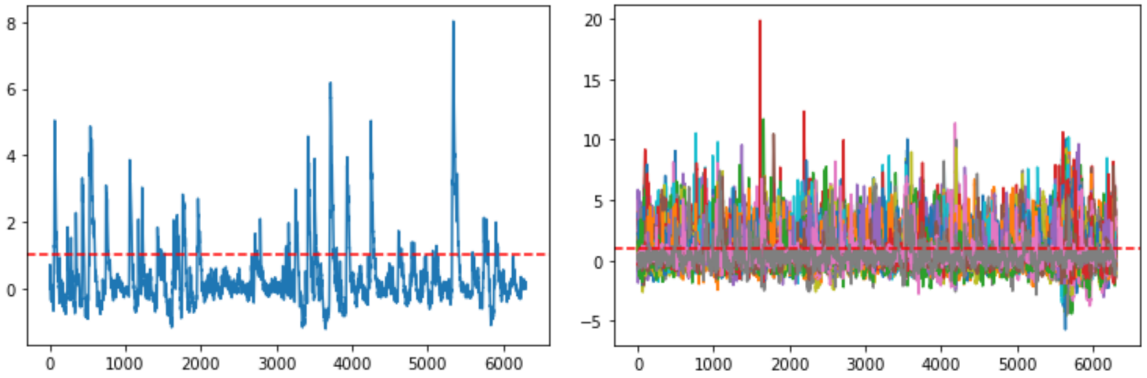
Our analysis and model were developed based on a zero-maze folder and it contains 13 mice in total. The columns of z score files represent the cells from the mouse and the number of chosen cells are different among mice. In behavior files, all of them consist of two columns, representing two behaviors respectively. And the number of rows in the z score file and behavior file of one mouse are the same, representing temporal time.

EDA

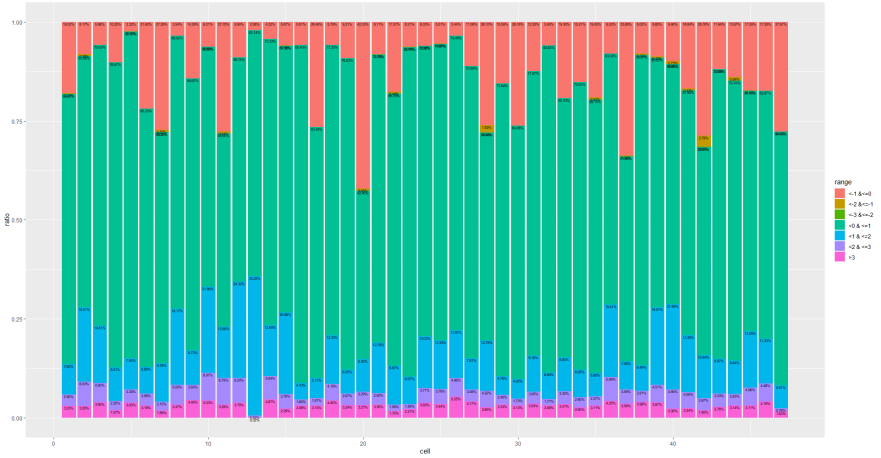
We first got an initial understanding of the data by using EDA on one mouse.



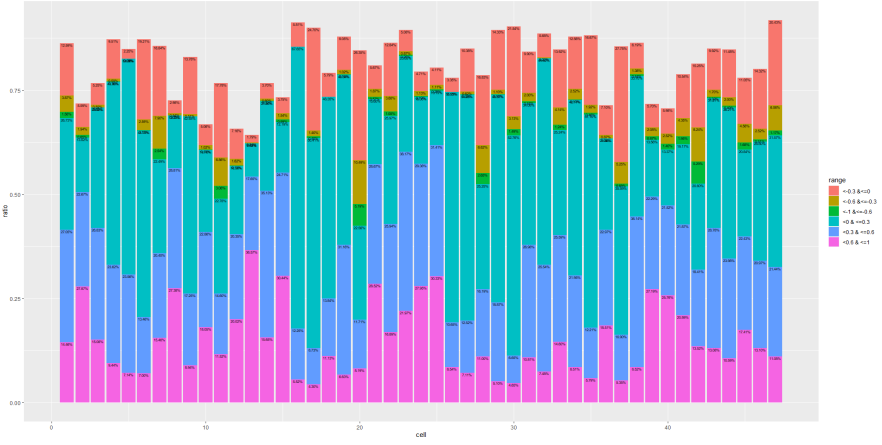
According to the plot of behavior data above, red points represent behavior 0, and blue points represent behavior 1. The X-axis is the time of observation, and y=1 means mice are displaying this behavior at this temporal time. We found that 2 rows of many behavior data are 0, which means researchers failed to track mice, so we deleted them.



The plot above on the left is z-scores of one cell over time with y=1 as a baseline, and the plot above on the right is z-scores of 47 cells over time with y=1 as a baseline. We can see the pattern of each cell over time in the mouse.



According to the plot of z-scores data above, the x-axis is 47 cells, the y-axis is the ratio of their sample values within a certain range. It was clear that most values of neural activations are between -1 and 1, especially between 0 and 1.



Then we further observed the values between -1 and 1 and found out that most values of neural activations are from 0 to 0.3.

Method

Data Pre-processing

Firstly we deleted temporal time points whose values were 0 either in behavior 0 or behavior 0 since those time points were missing observation. Then we took the first 80% observations of the two data sets as the training set and remained 20% as the test set.

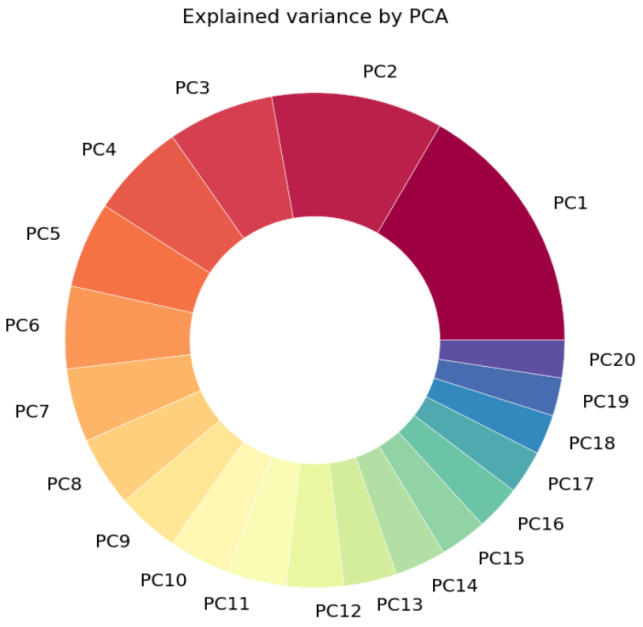
Logistic Regression (Baseline Model)

Firstly, we tried Logistic Regression as our baseline model. We took z scores of the training set as input x and behavior of the straining set as output y to fit the model. For input, we considered each cell(column) to be an independent variable and developed a Logistic Model with those variables and behaviors at the same time point.

With the training set, we obtained parameters of our Logistic Regression model, then we tested the prediction accuracy over the test data set.

PCA

Since a correlation exists between each cell, we wanted to try some methods that could reduce the dimension of input, which could possibly lessen the influence of correlation on our prediction accuracy. In our trial, we utilized Principle Component Analysis(PCA) to do dimension reduction. After PCA we found the variance ratio of the first principle component was only 17%, therefore, we decided to choose the top 20 components as our new inputs to fit the model. The following figure shows the variance ratio of each component after PCA of mouse 257.



Recurrent Neural Network(RNN)

Considering the time relation of observations and it was time-series data, we developed an RNN model including the LSTM layer. In the LSTM layer, the next unit contains information about former units, which means the value of former time points can influence the output after time. In our model, we tried different time lags like 3,5,10, etc. After testing we finally decided to use three-time points in the training set as one input and the behavior of the fourth point as the outcome.

The introduction of our model layer is following:

Layer 1: LSTM layer with 'tanh' activation.

Layers 2 & 3: Dense layer with 'relu' activation.

Layer 4: Dense layer with 'sigmoid' activation.

More details:

optimizer='adam', loss function='mean_squared_error'

epochs(iteration) = 300, batch_size=32

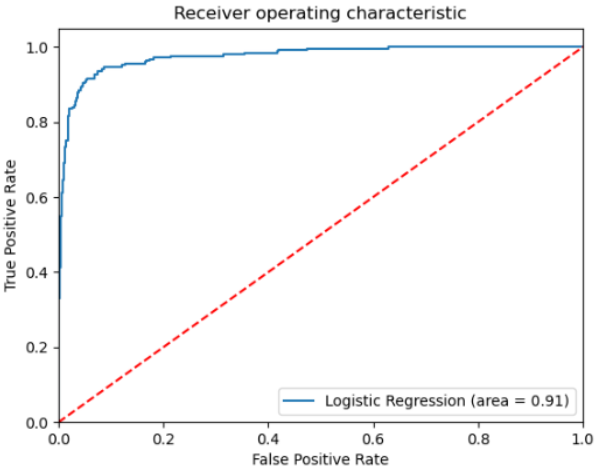
Result

Logistic Regression (Baseline Model)

Take mouse 257 as an example. Since there is a huge amount of temporal time without observations for two behaviors, if we include these data into the model, the accuracy of prediction would be influenced to a low value, so we removed them. Then we got an accuracy of 89.98%. Below are the intercept and coefficients after fitting the logistic regression model.

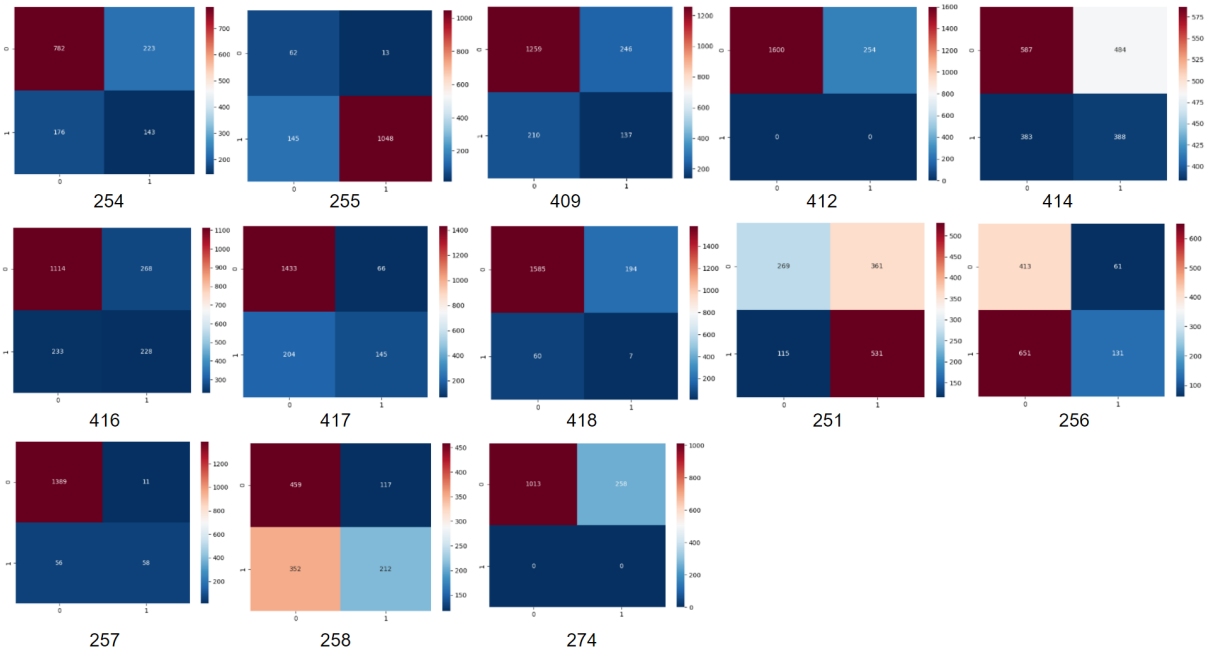
```
In [60]: print(model_logistic.intercept_, model_logistic.coef_)
[-5.98084378] [[ 0.24073423  0.24505155  0.16871795 -0.26646477  0.18191644  0.57832499
  -0.58167727  0.04711633  1.11034486  0.25867836  0.59368533  1.08252171
   0.30819175 -0.33589995 -0.04950964 -0.29360361  0.4820166   0.85239863
   0.08555266  0.05765893  0.90630021 -0.09962573  0.59783959  0.17848784
  -0.42709141 -1.41058601 -0.71322802 -0.0917473   0.12817653 -0.20902145
  -0.13834352 -0.01837179  0.04908175 -0.04180085 -1.31622424  0.02250423
   0.01659086  0.01967559  0.24615041  0.44950952 -0.17112492  0.89127981
   0.2998736  -0.55490263  1.09957348  0.56611178  0.37584697 -0.14563915
   0.01445961 -1.25897833 -0.1093008  -0.20154598 -0.4496369   0.00328724
  -0.56356914 -0.17273688 -0.21443481  0.35264315  0.09668005 -1.09039291
   1.1113909  -0.03556359 -0.45819712 -0.42117468  0.24489128 -0.70626264
  -0.2240025  -2.2865977   0.99600425 -0.80213821 -1.67453849 -0.33884669
```

To check the model, we looked at the ROC curve. According to the plot below, we can see it's a good classifier, which stays as far away from the red line.

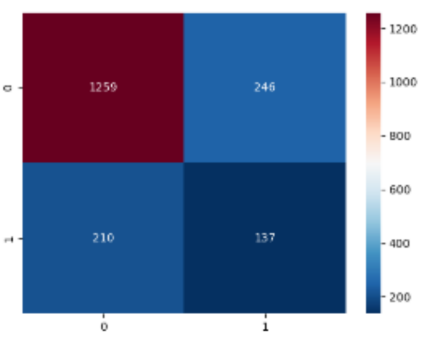


RNN

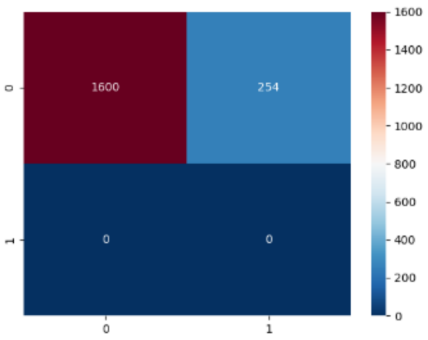
After using PCA to reduce the dimension of the original data, we put every mouse into our RNN model. Below is the confusion matrix of all mice, Acting on different mice, the prediction effect of RNN is different, and the effect of predicting 0 and predicting 1 is good or bad.



In detail, let's take mouse 409 and 412 that have good accuracy for instance. According to the plot below, for mouse 409, the probability that the RNN model correctly predicts a 0 is much greater than the probability that it correctly predicts a 1. In mouse 412, not even a 1 can be correctly predicted. This is probably due to not having too many 1s in the original data for us to train on. In other words, the y we picked is the result of 0 or 1 when behavior=1, and the data in this row is not as much as 1 in the data with behavior=0. But in short, due to the defects of the data itself, not every mouse can apply this model very well.



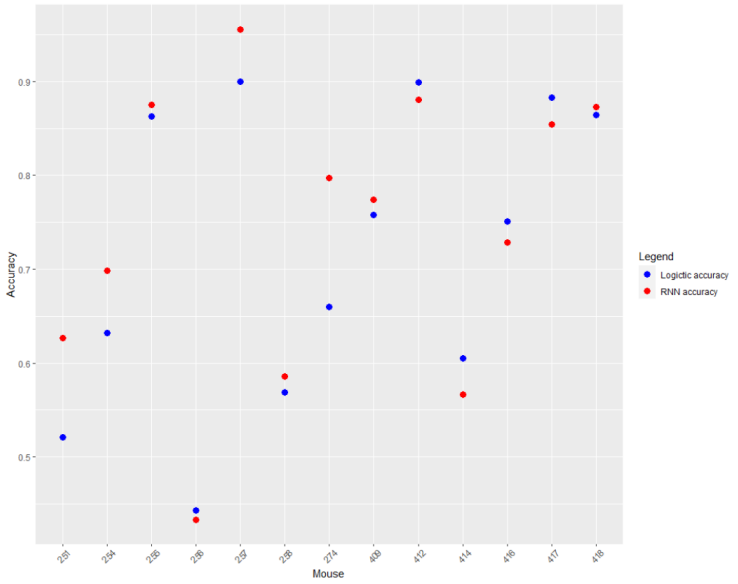
Mouse 409
Accuracy: 77%



Mouse 412
Accuracy: 87%

Comparison of Two Methods

We visualized the difference between the two models. As below, we can see RNN accuracy of 8 mice is higher than logistic accuracy.



As can be seen from the following table, the mice marked in red are improved by the RNN model, which are mouse 409, 418, 251, 257, 258, 274, 254, 255. The rest of the mice suggested using the baseline model for prediction.

Mouse ID	Logistic	RNN	Improvement
608034_409	0.758	0.7738	0.0158
608102_412	0.8988	0.8808	-0.018
608102_414	0.6049	0.5662	-0.0387
608103_416	0.7508	0.7282	-0.0226
608103_417	0.8828	0.854	-0.0288
608103_418	0.8643	0.8727	0.0084
616669_251	0.5207	0.627	0.1063
619539_256	0.4432	0.4331	-0.0101
619539_257	0.8998	0.9557	0.0559
619539_258	0.5687	0.586	0.0173
619541_274	0.6601	0.797	0.1369
619542_254	0.6323	0.6986	0.0663
619542_255	0.8631	0.8754	0.0123

Conclusion

We used both the logistic regression model and the RNN model in this project. After the model update of the RNN, we found that the prediction of the behavior of eight mice: 409, 418, 251, 257, 258, 274, 254, 255 was improved.

However, there are some limitations coming out. First, the original distribution of behavior 0 and 1 is very imbalanced, which will influence our model fitting and prediction. Then, the cells of each mouse are selected randomly (stimulus-selective or neutral). And we didn't consider the correlation between each cell. Also, the lag of time we used is three, and it was chosen without validation. Finally, we haven't found a good method that works for all mice.