Mouse_EDA

XihaoCao

2022/4/6

```
knitr::opts_chunk$set(cache = T)
knitr::opts_chunk$set(echo = F)
knitr::opts_chunk$set(warning = F, message = F)
```

Introduction

Our client is Cruz-Martin, Alberto from the Center for Systems Neuroscience at Boston University. Our client is mainly interested in decoding how mice's brain circuits control their behaviors, and he wants to understand the contribution of specific cell types and their connections to the visual processing and perception of a mouse. Our client wants us to help him predict the mice's behaviors by their cell activation status in this project.

Our client has conducted three types of experiments: Zero Maze, Opposite Sex, and Direct Interaction on multiple mice. In the Zero Maze experiment, mice are placed in an elevated zero maze to explore for 10 minutes, they can either be in the closed arm (anxiolytic) or the open arm (anxiogenic), and the location of the mice is recorded. In the Opposite Sex experiment, mice were placed in a social chamber for 10 minutes and allowed the mice to explore two cups containing a male and female of the same strain. Which cup the experiment mouse interacts with is recorded. In the Direct Interaction experiment, each mouse is placed in a social chamber for 5 minutes and is allowed to interact with a novel mouse of the same strain freely. Then whether the experiment mouse is performing a social behavior is recorded. In all three experiments, the behaviors of mice are considered binary.

Our group focuses on the Zero Maze experiment and conducts the following analysis.

Data and Methods

Our client carried out the Zero Maze experiment on 13 mice; the data of each mouse are stored in separate data sets. Each experiment's time length is 10 minutes, and the behaviors of the mouse and information of each cell are recorded every 10 HZ and stored as a row. The following graph shows the basic structure of the data sets.

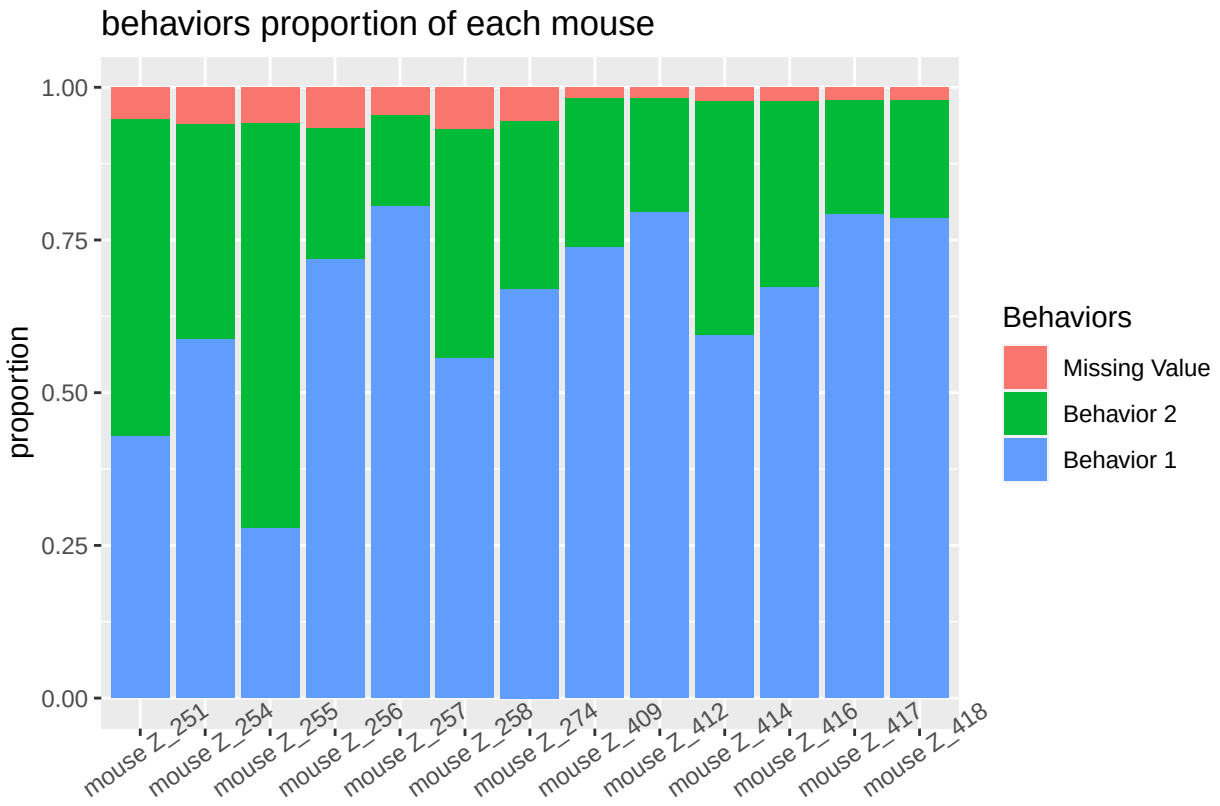
```
## Image. Width: 1812 pix Height: 402 pix Depth: 1 Colour channels: 4
```

Our client records whether the mouse is in the open armor in the close arm separately using two columns, a value of 1 in a column representing the mouse is in the corresponding arm, 0 representing not. When the mouse transits from one arm to the other and its body is in the middle of the two arms, the values of the two columns are both recorded as 0. However, after discussing with our client, we consider the transition situation the edge case and no longer include them in the analysis. Thus the behaviors of mice in this experiment are assumed binary; they either stay in the open arm or the closed arm.

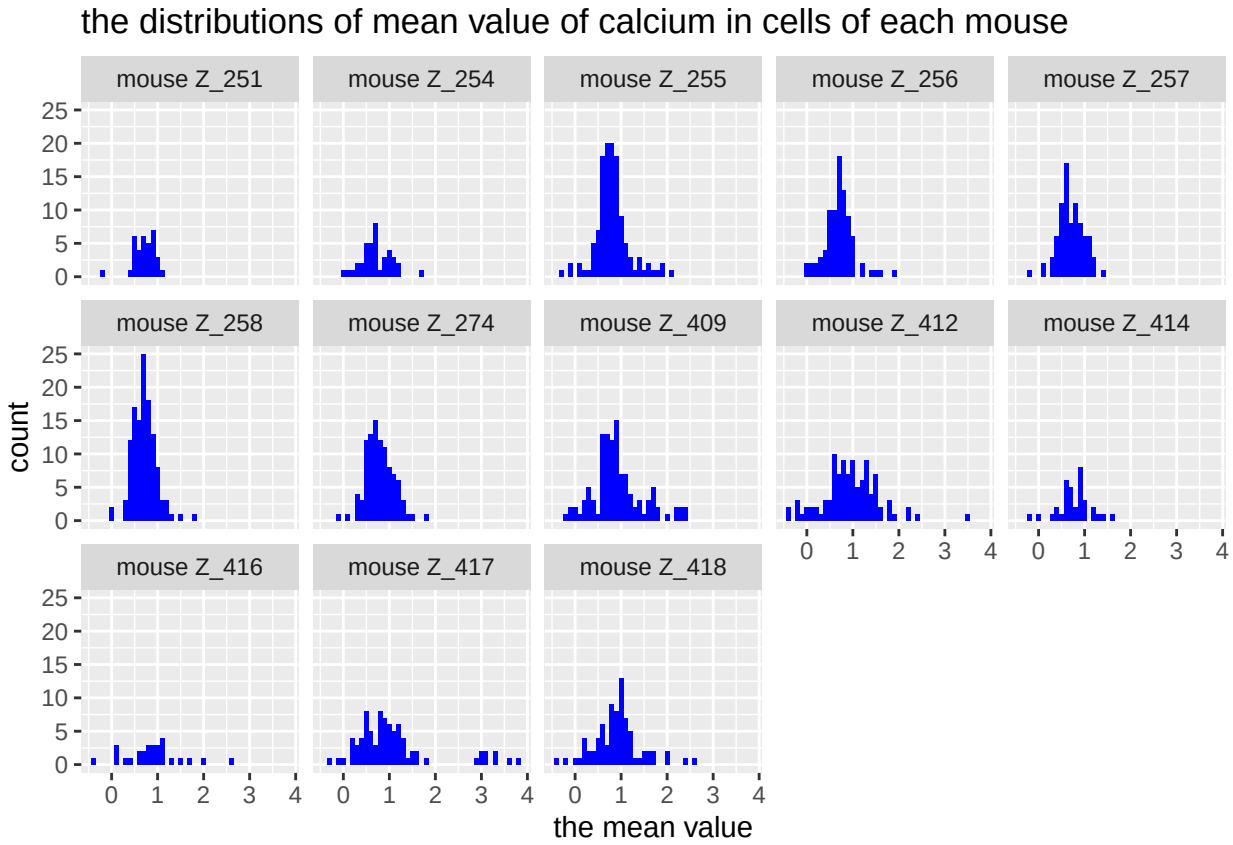
Each mouse cell's calcium flow amount is recorded to quantify whether the cell is on fire; the values are all positive. Meanwhile, the collection of cells recorded for each mouse is different; for example, 112 cells of mouse Z409 are recorded, and only 28 cells of mouse Z416 are recorded.

EDA

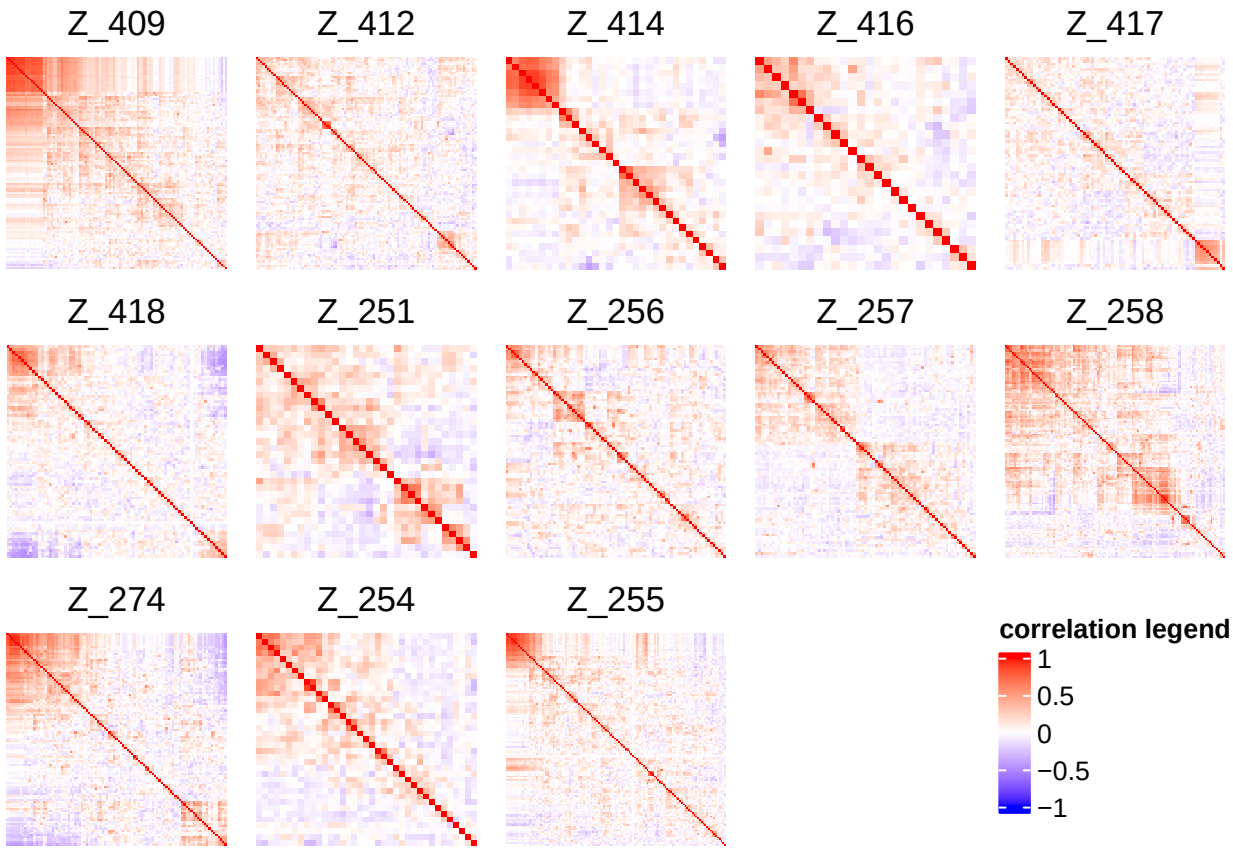
Then we explore the proportion of two types of behaviors of each mouse. We can see that most mice conduct behavior 1 at least 50% of the time except for mice Z251 and Z255. Mouse Z257, Z417, and Z418 even conduct behavior 1 about 75%. We also notice the existence of missing values where both behaviors are recorded as 0, but there are not many of them.



This graph shows us the distribution of the mean value of calcium transition in cells of each mouse. As we can see, most cells have an average value between 0 to 2. Moreover, most of them are t distributed except for the mice Z_416 and Z_251, the two mice that have fewer data (25 cells and 34 cells, respectively).



In this heat map, we plot 13 mice. The heat map illustrates the correlations of cells of each mouse, as we can see that there exist some strong correlations between certain cells. Thus we need to check correlations in detail before constructing any models.



PCA

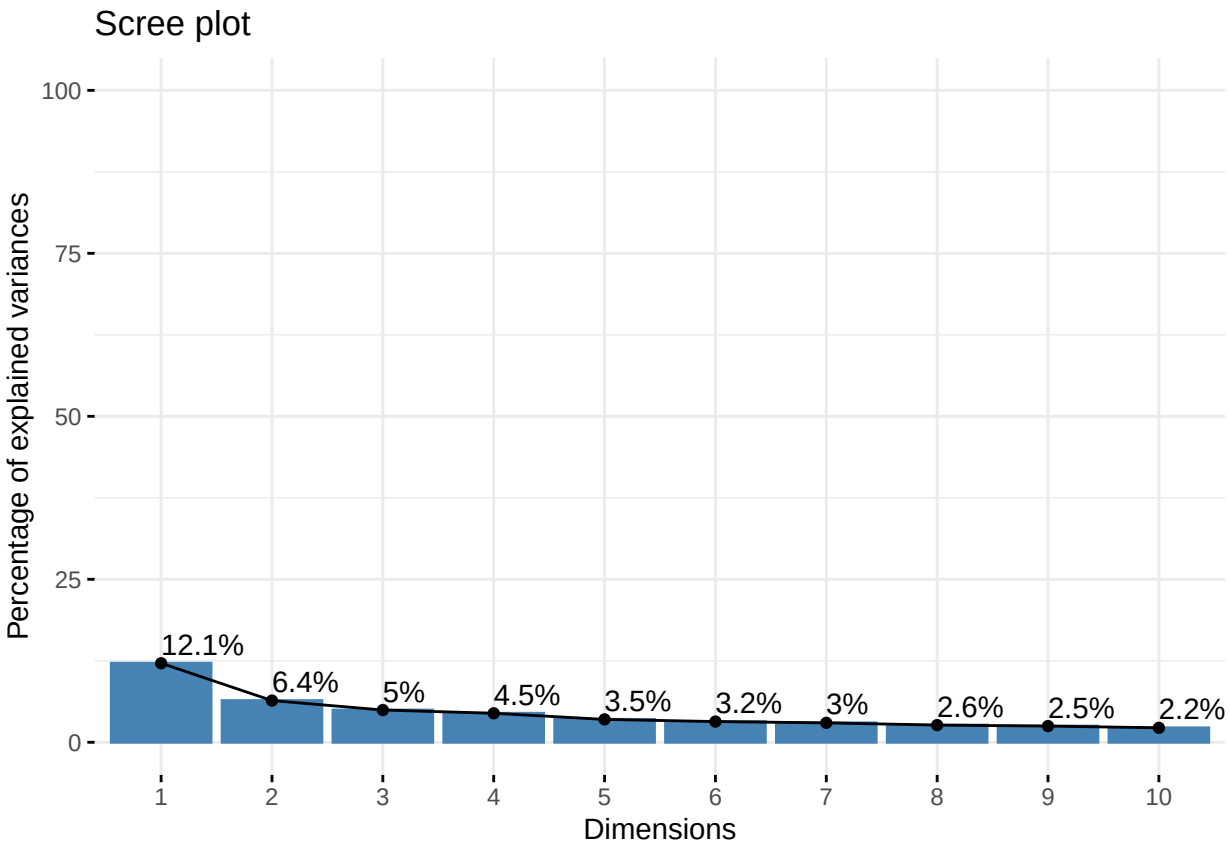
According to the correlation map, we found there is a high correlation between variables, so we considered conducting PCA. At first, we calculated VIF values. Large VIF suggests predictors almost completely explained by the other variables in the equation. The VIF values we calculate for Mouse 255 of the Zero Maze study are as follows:

```
##        X1        X2        X3        X4        X5        X6        X7        X8
##  4.443693 32.799587 11.122254  3.406167  5.946836  2.626223  5.854133  4.427542
##        X9       X10       X11       X12       X13       X14       X15       X16
## 10.108495 18.600526  5.731881 30.105783 40.978332 17.056840  3.807708  5.079167
##       X17       X18       X19       X20
## 16.882012  3.652280 14.019159  7.706484
```

We find strong colinearity in our predictors, so we conduct PCA and then choose the first 20 PCs as new predictors.

Training model

The scree plot shows the tenth dimension only explained the 2.2% information of the data.



Then we Randomly half-by-half split the data into train set and test set and fit logistic regression again using the 20 principle components. Then we calculate the VIF values again, and we found there is no strong colinearity.

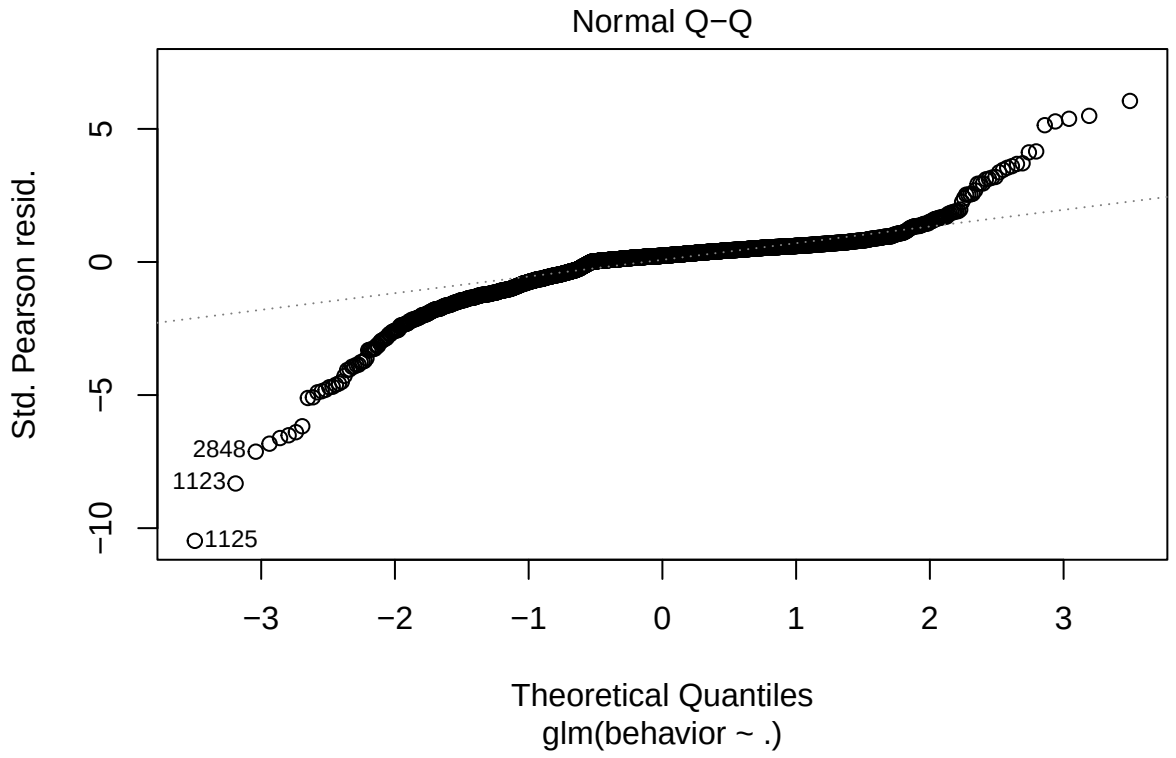
```
##    Dim.1    Dim.2    Dim.3    Dim.4    Dim.5    Dim.6    Dim.7    Dim.8
## 1.208163 1.507070 1.529415 1.468538 1.569593 1.411640 1.474581 1.241545
##    Dim.9   Dim.10   Dim.11   Dim.12   Dim.13   Dim.14   Dim.15   Dim.16
## 1.477846 1.226444 1.364005 1.160606 1.255761 1.149925 1.177338 1.251072
##   Dim.17   Dim.18   Dim.19   Dim.20
## 1.124744 1.066233 1.163989 1.146291
```

Then we did the prediction on the test set using the threshold of 0.5 and got a result of about 0.83 accuracies (0.81, 0.85), 0.8 recall, and 0.9 specificities. The ROC plot also shows that the best threshold may be around 0.3, where we can have both high recall and specificity.

```
## Confusion Matrix and Statistics
##
##    p
##        0    1
##   0  462  159
##   1  170 1328
##
##                Accuracy : 0.8447
##                  95% CI : (0.8286, 0.8599)
##     No Information Rate : 0.7017
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6272
```

```
##
##  Mcnemar's Test P-Value : 0.5814
##
##             Sensitivity : 0.7310
##             Specificity : 0.8931
##          Pos Pred Value : 0.7440
##          Neg Pred Value : 0.8865
##              Prevalence : 0.2983
##          Detection Rate : 0.2180
##    Detection Prevalence : 0.2931
##       Balanced Accuracy : 0.8120
##
##        'Positive' Class : 0
##
```

The QQ plot shows that the residual follows a normal distribution.



Model for the whole dataset

Finally, based on the model building in the training set and test by test set, we fit the model again in the whole dataset.

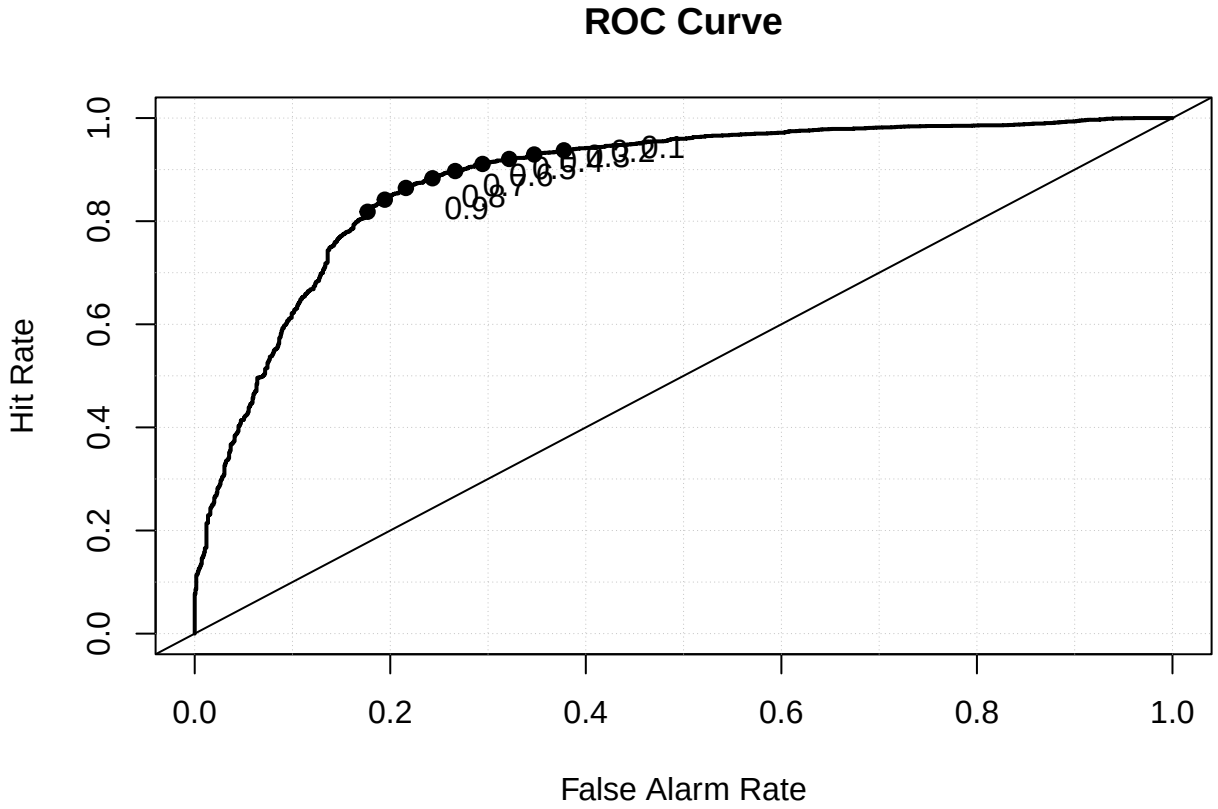
```
##
## Call:
## glm(formula = behavior ~ ., family = "binomial", data = new_predictors1)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -3.2308  -0.4357   0.3118   0.6522   3.1049
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.42835    0.05539  25.786  < 2e-16 ***
## Dim.1       -0.12589    0.01220 -10.317  < 2e-16 ***
## Dim.2        0.19036    0.02455   7.754 8.88e-15 ***
## Dim.3        0.26253    0.02606  10.075  < 2e-16 ***
## Dim.4       -0.12443    0.02910  -4.276 1.90e-05 ***
## Dim.5        0.76711    0.03514  21.830  < 2e-16 ***
## Dim.6       -0.40561    0.03277 -12.376  < 2e-16 ***
## Dim.7       -0.24334    0.03134  -7.764 8.24e-15 ***
## Dim.8        0.39169    0.02877  13.614  < 2e-16 ***
## Dim.9        0.12280    0.03444   3.565 0.000364 ***
## Dim.10       0.22221    0.03562   6.238 4.42e-10 ***
## Dim.11       0.33747    0.03580   9.427  < 2e-16 ***
## Dim.12       0.15524    0.03040   5.106 3.28e-07 ***
## Dim.13       0.48362    0.03400  14.223  < 2e-16 ***
## Dim.14       0.04326    0.03325   1.301 0.193205
## Dim.15       0.02575    0.03400   0.757 0.448877
## Dim.16      -0.28052    0.03960  -7.083 1.41e-12 ***
## Dim.17      -0.22576    0.03353  -6.733 1.67e-11 ***
## Dim.18      -0.03125    0.03342  -0.935 0.349739
## Dim.19      -0.28814    0.03842  -7.500 6.37e-14 ***
## Dim.20       0.14868    0.03965   3.750 0.000177 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5140.9  on 4237  degrees of freedom
## Residual deviance: 3311.6  on 4217  degrees of freedom
## AIC: 3353.6
##
## Number of Fisher Scoring iterations: 6
```

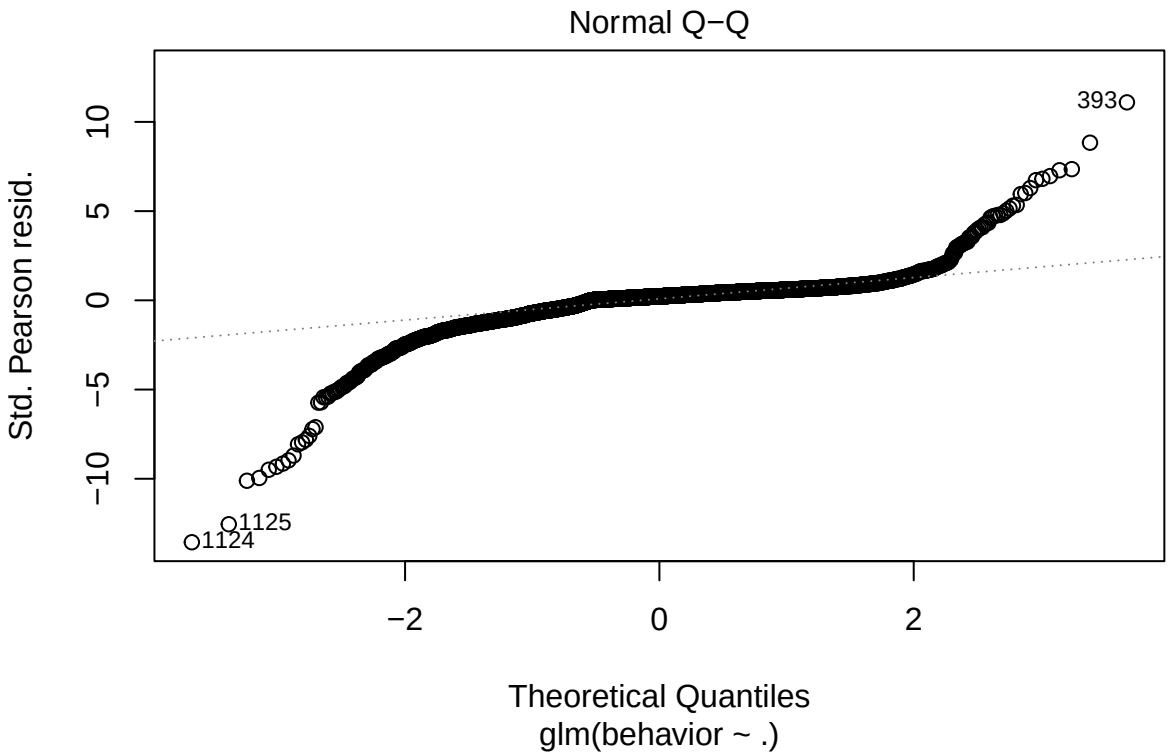
The confusion matrix shows the accuracy of the model for the whole dataset is 0.85 (0.8376, 0.8594) and with a Sensitivity of 0.74 and Specificity of 0.88.

```
## Confusion Matrix and Statistics
##
##    p
##        0    1
##   0  916  334
##   1  307 2681
##
##                Accuracy : 0.8487
##                  95% CI : (0.8376, 0.8594)
##     No Information Rate : 0.7114
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.634
##
##  Mcnemar's Test P-Value : 0.3044
```

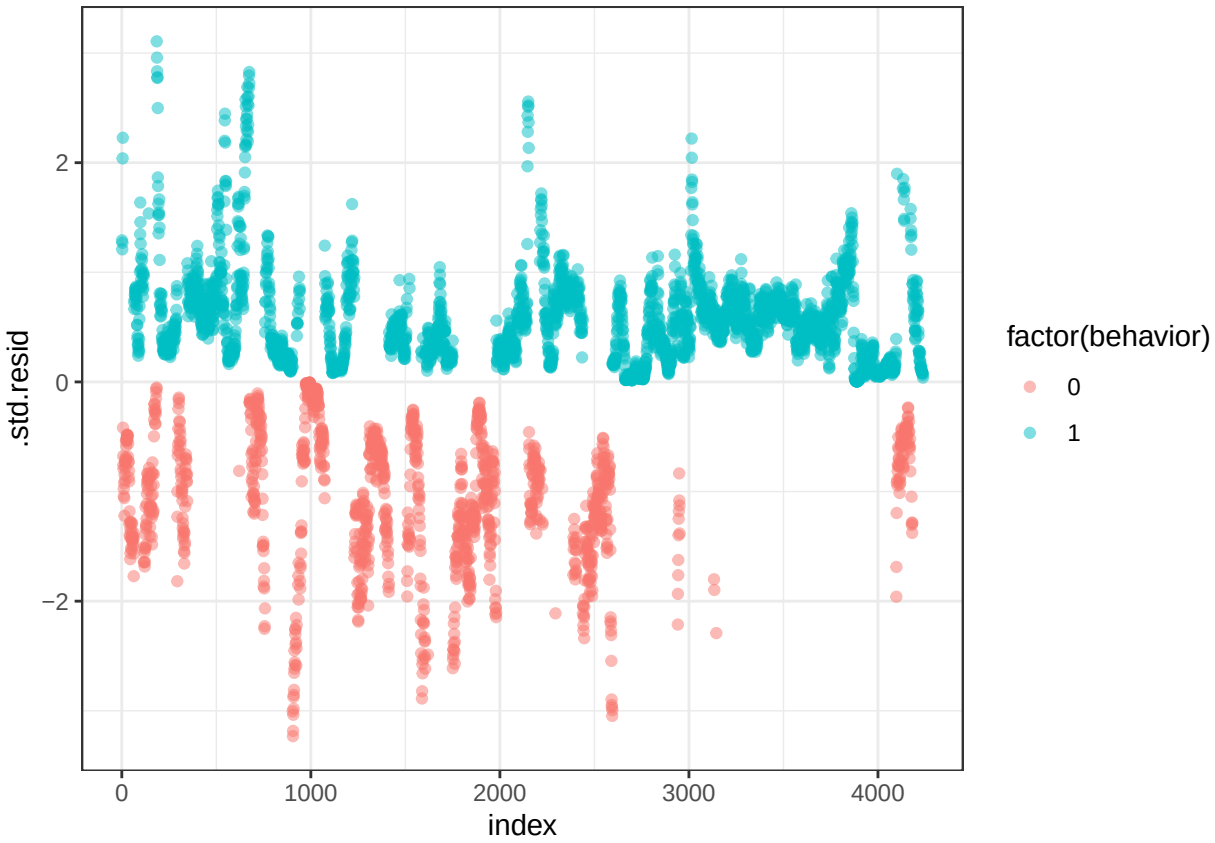
```
##
##             Sensitivity : 0.7490
##             Specificity : 0.8892
##          Pos Pred Value : 0.7328
##          Neg Pred Value : 0.8973
##              Prevalence : 0.2886
##          Detection Rate : 0.2161
##    Detection Prevalence : 0.2950
##       Balanced Accuracy : 0.8191
##
##        'Positive' Class : 0
##
```



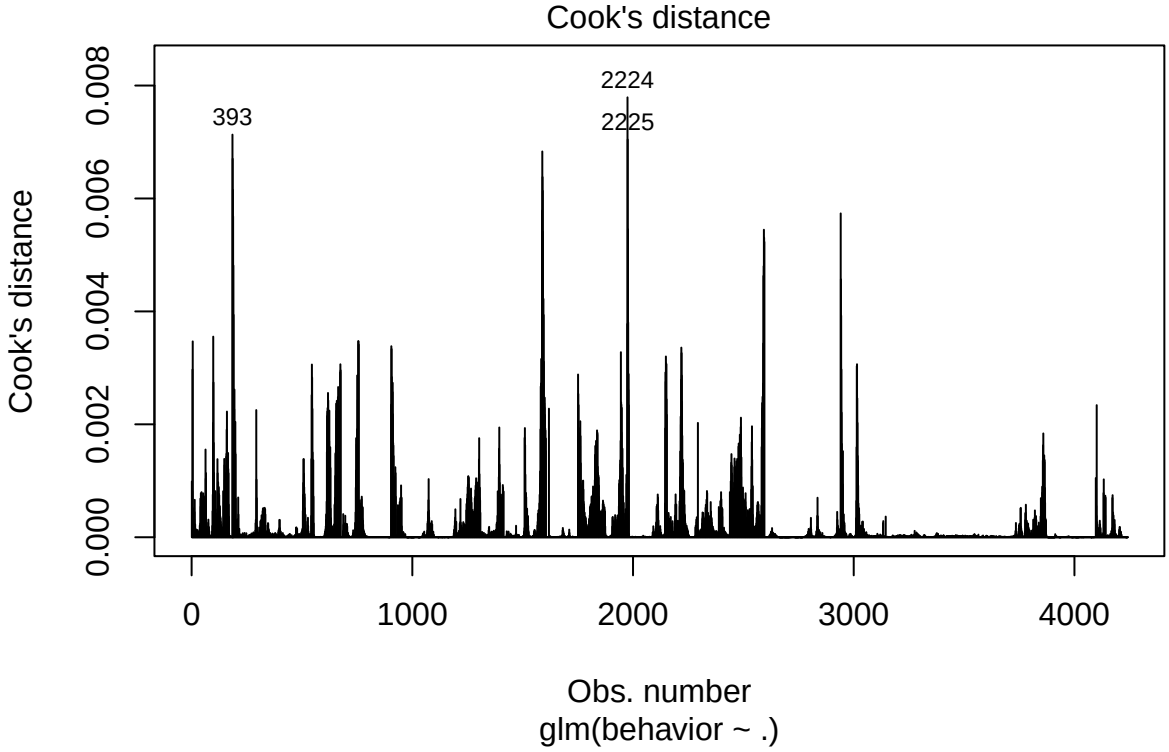
We also check the normality of the residual, and the residual follows a normal distribution.



Then we plot the residual and we found that the model has better performance of detecting B1 than B0.



Influential values which are extreme individual data points that can alter the quality of the logistic regression model.The most extreme values in the data can be examined by visualizing the Cook's distance values. Here we label the top 3 largest values.



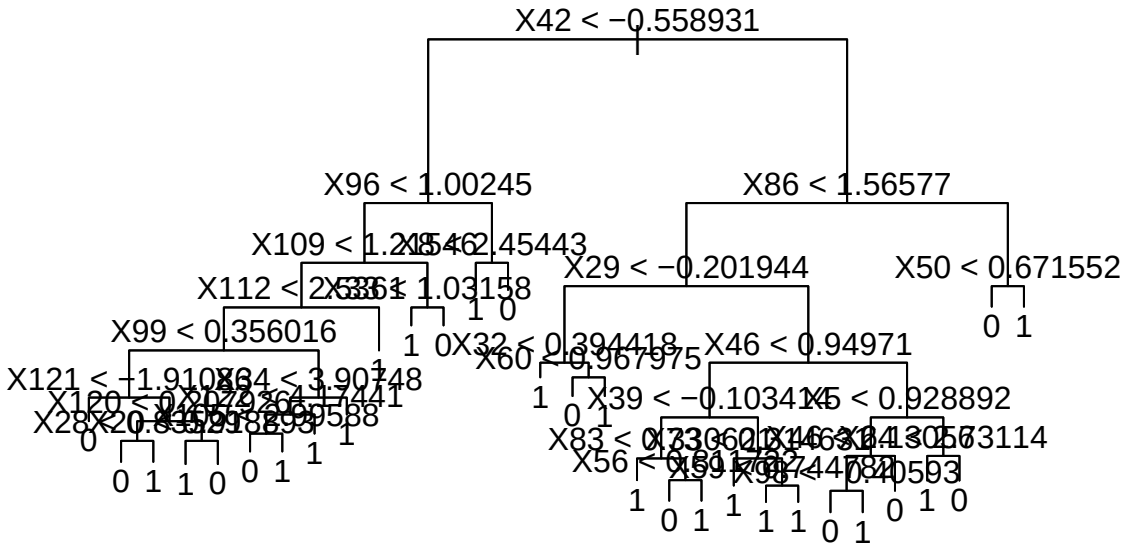
Tree-Based Method

We extract 20 PC to fit the logistic regression, while the price for that is that we lose some useful signal information and noise since the top 20 PC only explained an accumulative 61% of the data, and we only get the 83% accuracy. Considering this data includes many cells, we do not need to make dimensionality reduction before the random forest—the tree-based method without distribution restrictions.

Binary Decision tree

Therefore, we use a classification tree to improve our accuracy of prediction. We put all variables in the training set and fit a binary decision tree.

```
##
## Classification tree:
## tree(formula = behavior ~ ., data = rf[train, ])
## Variables actually used in tree construction:
##  [1] "X42"  "X96"  "X109" "X112" "X99"  "X121" "X120" "X28"  "X20"  "X34"
## [11] "X122" "X105" "X33"  "X8"   "X86"  "X29"  "X32"  "X60"  "X46"  "X39"
## [21] "X83"  "X56"  "X73"  "X59"  "X5"   "X98"  "X64"  "X50"
## Number of terminal nodes:  30
## Residual mean deviance:  0.3309 = 691.2 / 2089
## Misclassification error rate: 0.05805 = 123 / 2119
```



The predicted result shown by the confusion matrix, the accuracy is 0.89(0.88, 0.90). And the Sensitivity is 0.840, and the Specificity is 0.9164.

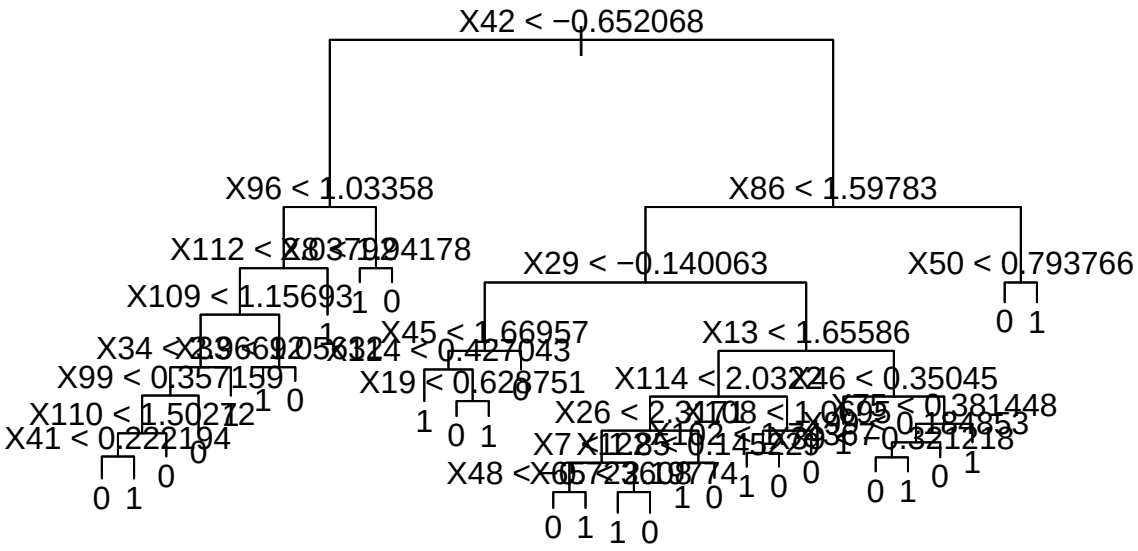
```
## Confusion Matrix and Statistics
##
##
## p      0    1
##   0  482   87
##   1  117 1433
##
##                Accuracy : 0.9037
##                  95% CI : (0.8904, 0.916)
##     No Information Rate : 0.7173
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.759
##
##  Mcnemar's Test P-Value : 0.04232
##
##             Sensitivity : 0.8047
##             Specificity : 0.9428
##          Pos Pred Value : 0.8471
##          Neg Pred Value : 0.9245
##              Prevalence : 0.2827
##          Detection Rate : 0.2275
##    Detection Prevalence : 0.2685
##       Balanced Accuracy : 0.8737
```

```
##
##        'Positive' Class : 0
##
```

Random Forest

A random forest is the average of many different single trees that only contain different subsets of original predictors. So a random forest will typically be helpful when we have many correlated predictors. We use a random forest model, which can be a universal method for all 13 zero maze data sets. We choose predictor subset size for each data set according to the number of cells. Then we build a reproducible function 'rf' and apply it to all 13 data sets.

```
##
## Classification tree:
## tree(formula = behavior ~ ., data = rf[train, ])
## Variables actually used in tree construction:
##  [1] "X42"  "X96"  "X112" "X109" "X34"  "X99"  "X110" "X41"  "X33"  "X8"
## [11] "X86"  "X29"  "X45"  "X114" "X19"  "X13"  "X26"  "X7"   "X48"  "X65"
## [21] "X122" "X108" "X102" "X46"  "X75"  "X39"  "X50"
## Number of terminal nodes:  30
## Residual mean deviance:  0.3339 = 697.6 / 2089
## Misclassification error rate: 0.05521 = 117 / 2119
```




```
## Confusion Matrix and Statistics
##
##
## p      0    1
```

```
##   0  504  118
##   1  102 1395
##
##                Accuracy : 0.8962
##                  95% CI : (0.8824, 0.9088)
##     No Information Rate : 0.714
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.7478
##
##  Mcnemar's Test P-Value : 0.3119
##
##             Sensitivity : 0.8317
##             Specificity : 0.9220
##          Pos Pred Value : 0.8103
##          Neg Pred Value : 0.9319
##              Prevalence : 0.2860
##          Detection Rate : 0.2378
##    Detection Prevalence : 0.2935
##       Balanced Accuracy : 0.8768
##
##        'Positive' Class : 0
##
```

This is the confusion matrix for Mouse 255, this is one result of our function.

```
## Confusion Matrix and Statistics
##
##
## yhat.rf    0    1
##       0  574   10
##       1   32 1503
##
##                Accuracy : 0.9802
##                  95% CI : (0.9733, 0.9857)
##     No Information Rate : 0.714
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9509
##
##  Mcnemar's Test P-Value : 0.001194
##
##             Sensitivity : 0.9472
##             Specificity : 0.9934
##          Pos Pred Value : 0.9829
##          Neg Pred Value : 0.9792
##              Prevalence : 0.2860
##          Detection Rate : 0.2709
##    Detection Prevalence : 0.2756
##       Balanced Accuracy : 0.9703
##
##        'Positive' Class : 0
##
```

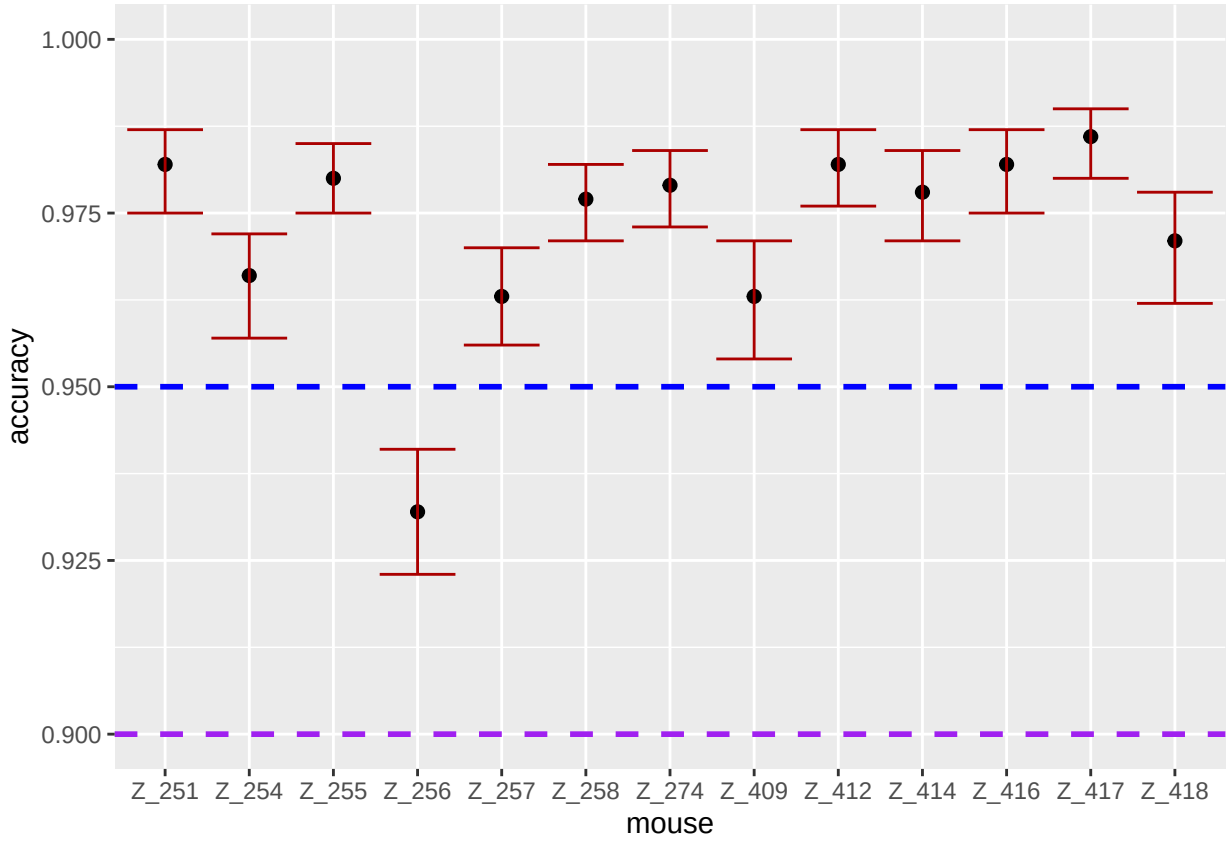
Result for 13 mice in random forest function:

mouse	accuracy	lower_bound	upper_bound	record	specificity	F1
Z_409	0.963	0.954	0.971	0.954	0.970	0.956
Z_412	0.982	0.976	0.987	0.991	0.969	0.986
Z_414	0.978	0.971	0.984	0.942	0.994	0.963
Z_416	0.982	0.975	0.987	0.996	0.935	0.988
Z_417	0.986	0.980	0.990	0.998	0.914	0.992
Z_418	0.971	0.962	0.978	0.993	0.937	0.976
Z_251	0.982	0.975	0.987	0.993	0.956	0.987
Z_254	0.966	0.957	0.972	0.996	0.874	0.976
Z_255	0.980	0.975	0.985	0.999	0.901	0.988
Z_256	0.932	0.923	0.941	0.976	0.863	0.946
Z_257	0.963	0.956	0.970	0.990	0.904	0.974
Z_258	0.977	0.971	0.982	0.995	0.902	0.986
Z_274	0.979	0.973	0.984	0.998	0.896	0.987

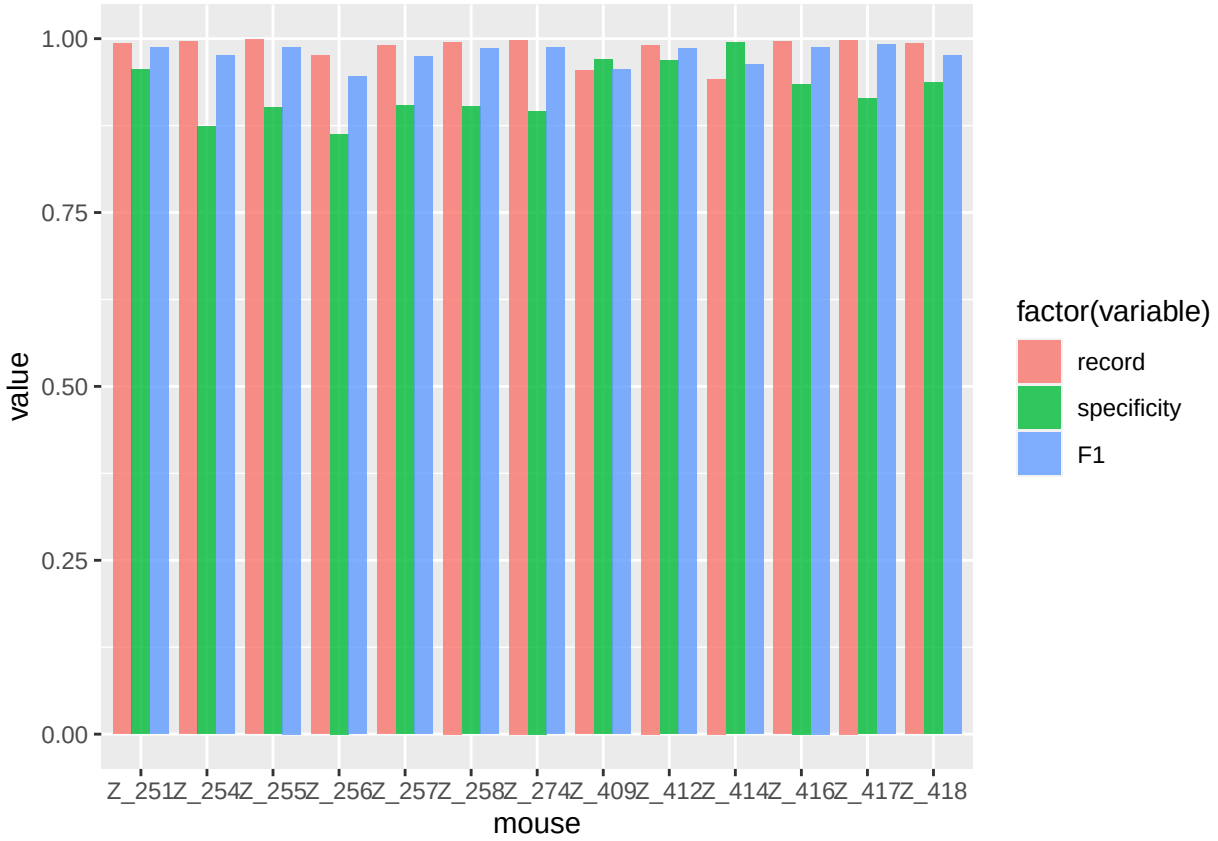
The average accuracy of random forest for 13 mice is 97.23%.

```
##     mouse              accuracy       lower_bound      upper_bound
##  Length:13          Min.   :0.9320   Min.   :0.9230   Min.   :0.9410
##  Class :character   1st Qu.:0.9660   1st Qu.:0.9570   1st Qu.:0.9720
##  Mode  :character   Median :0.9780   Median :0.9710   Median :0.9840
##                     Mean   :0.9724   Mean   :0.9652   Mean   :0.9783
##                     3rd Qu.:0.9820   3rd Qu.:0.9750   3rd Qu.:0.9870
##                     Max.   :0.9860   Max.   :0.9800   Max.   :0.9900
##      record        specificity           F1
##  Min.   :0.9420   Min.   :0.8630   Min.   :0.9460
##  1st Qu.:0.9900   1st Qu.:0.9010   1st Qu.:0.9740
##  Median :0.9930   Median :0.9140   Median :0.9860
##  Mean   :0.9862   Mean   :0.9242   Mean   :0.9773
##  3rd Qu.:0.9960   3rd Qu.:0.9560   3rd Qu.:0.9870
##  Max.   :0.9990   Max.   :0.9940   Max.   :0.9920
```

We plot the accuracy with a 95% confidence interval. The dot in the figure is the accuracy for each mouse, and the red line corresponds 95% confidence interval. The blue dashed line is baseline at 0.95, while the purple dashed line is 0.9. We can see almost mice got an accuracy of more than 95%, and all mice had more than 90% accuracy.



Then we plot the recall, specificity, and F1 of random forest for 13 mice, respectively. These values are generally close to 1, which means our model has excellent performance.



Multilayer Neural Network

We also have built a simple neural network model to predict the mice's behaviors. In the model, we construct two hidden layers, which contain 100 units and 50 units respectively, and both with ReLu activation function. In the output layer, we use the Sigmoid activation function to show the probability of whether the mouse is conducting behavior 1. The following graph shows the model structure of mouse Z409 as an example.

```
## Image. Width: 852 pix Height: 700 pix Depth: 1 Colour channels: 4
```

```
## Image. Width: 1930 pix Height: 752 pix Depth: 1 Colour channels: 4
```

The accuracies of our Neural Network models are very high; most of them are around 97%. We set the epochs to be 50, batch size to be 32, and validation set to be random 20% of the data, and we plot the model accuracy of mouse Z409 as an example below.

Conclusion

There exist correlations between many cells of each mouse, including all the cells as predictors in a model is not wise, dimension reduction is necessary.

Compared with the logistic_PCA model, the random forest classification model not only is easier to visualize and interpret, but also provides a higher prediction accuracy that can even compete with the Neural Network model.

Discussion

For our logistic-PCA model, we select the first 20 principal components as predictors. The choice of this number is not rigorously verified, so 20 may not be the optimal choice. And in the random forest, we pick the predictor subset size to be the square root of the number of all predictors, this is a common convention, and it is not guaranteed to be the optimal choice. In the neural network model, the number of hidden layers and units in each layer are also not verified to be the optimal choice. In the validation process, since there are spatial and time relationships between each row of the data, randomly selecting observations to form the validation set may not be the best choice.