Course Project

Kesar Sidhu

2025-03-01

Title

Exploring Neural Signals: Applying Predictive Modeling to Trial Outcomes Using Spike Trains

Abstract

This project aims to bridge the gap between neuroscience and data science by exploring the relationship between neural signals and decision-making in mice. Using the data from Steinmetz et al. (2019), we can analyze the recorded neural activity record from a contrast-oriented experiment. Our approach to this project includes exploratory data analysis, data integration across various trials, and applying machine learning to predict trial outcomes. The goal is to design a predictive model that uses spike trains and contrast stimuli to reliably predict whether the decision outcome will be success or failure.

Introduction

Understanding how the brain makes decisions has been a long-standing question within the field of neuroscience. To address this question, we can analyze neural activity in mice performing a visual contrast-oriented task. In the experiment, the mice are presented with two contrasts (left and right) on a screen and must choose the correct side based on the contrast differences using a wheel controlled by their forepaws. The decision outcomes are classified as either success (1) or failure (-1), with success determined by the following rules:

- If the left contrast is greater than the right, success occurs when the mouse turns the wheel to the right; failure occurs if the mouse turns it the wrong way.
- If the right contrast is greater than the left, success occurs when the mouse turns the wheel to the left; failure occurs if the mouse turns it the wrong way.
- If both contrasts are zero, success occurs if the mouse holds the wheel still; failure occurs if it moves the wheel.
- If the left and right contrasts are equal but non-zero, the correct side is chosen randomly (50% chance).

By analyzing their neural activity in response to the contrast levels and their corresponding actions (turning the wheel or holding still), we hope to understand how the neural signals (spike trains) in the mice correspond to their decision-making process.

The main sections in this project include: - Exploratory Data Analysis: We will explore the main characteristics of the data, such as the data structure, comparing the neural activity between and across experiments, understand the homogenity and heterogenity across sessions and mice, alongside running hypothesis tests to compare the neural activity across different sessions and mice. - Data Integration: - Predictive Modeling: - Discussion:

Importing Libraries and Data

```
##  [1] "session1.rds"  "session10.rds" "session11.rds" "session12.rds"
##  [5] "session13.rds" "session14.rds" "session15.rds" "session16.rds"
##  [9] "session17.rds" "session18.rds" "session2.rds"  "session3.rds"
## [13] "session4.rds"  "session5.rds"  "session6.rds"  "session7.rds"
## [17] "session8.rds"  "session9.rds"
```

```
##    sesssion     mouse neuron_num trial_num success contrast_left_avg
## 1 Session 1      Cori        734       114      69         0.2960526
## 2 Session 2      Cori       1070       251     159         0.3177291
## 3 Session 3      Cori        619       228     151         0.2357456
## 4 Session 4 Forssmann       1769       249     166         0.3343373
## 5 Session 5 Forssmann       1077       254     168         0.3385827
## 6 Session 6 Forssmann       1169       290     215         0.3534483
##   contrast_right_avg spikes_avg
## 1          0.4298246  1.5396410
## 2          0.3515936  1.2656328
## 3          0.3684211  2.2339370
## 4          0.3222892  0.8415550
## 5          0.3562992  1.1158292
## 6          0.3250000  0.6631899
```

Exploratory Data Analysis

The goal of this section is to explore the main characteristics of the dataset. This includes understanding the data structure, analyzing neural activity within and across experiments, and examining the homogeneity and heterogeneity across sessions and mice. Additionally, hypothesis tests will be conducted to compare neural activity across different sessions and subjects.

Data Structure Overview

```
## [1] "contrast_left"  "contrast_right" "feedback_type"  "mouse_name"
## [5] "brain_area"     "date_exp"       "spks"           "time"
```

```
##                Length Class  Mode
## contrast_left  114    -none- numeric
## contrast_right 114    -none- numeric
## feedback_type  114    -none- numeric
## mouse_name       1    -none- character
## brain_area     734    -none- character
## date_exp         1    -none- character
## spks           114    -none- list
## time           114    -none- list
```

Our imported data features 18 session, each containing 8 variables of different lengths and data types as shown above. For instance the feedback type shows a length of 114 as a numeric vector, while the mouse_name is a character type with a length of 1.

The metadata for this dataset includes: - Contrast_left: contrast level of the left side of the screen in the experiment - Contrast_right: contrast level of the right side of the screen in the experiment - Feedback_type: success or failure if the mouse made the correct decision - Mouse_name: name of the mouse - Brain_area: which part of the brain each neuron corresponds to - Date_exp: date of the experiment - Spks: spike trains for each neuron (matrix where row = neuron, column = time bin) - time: timestamps for each trial

```
## [1] 734  40
```

```
## [1] 734
```

```
##  [1] 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [39] 0 0
```

We can look more deeply into the spike dimensions corresponding to the neurons from a session. Here we see that the spike matrix of neuron 1 from session 1 has 734 trials with time binds of length 40. More closely we that the vector length for brain area is 734 as well. This means that each trial is associated with a specific region of the brain.

To picture the spike train, we can look at the spike train of trial 6 from neuron 1 in session 1. Here we can see the 1's and 0s representing the feedback type of success or failure.

We can first look at if sessions later in the experiment fire more or less than earlier sessions. This can help us understand if the neural activity changes over time. To tackle this question, since we have 18 sessions, selecting one early and one later session could work. In our case, we'll select session 1 and session 10.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.094   1.309   1.469   1.540   1.764   2.339
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.814   1.059   1.191   1.188   1.327   1.646
```

To approach this problem I decided calculate the average number of spikes per trial across all the neurons in the session. This will give us an idea of how active the neurons are in each session. Calculating the average number of spikes was done by creating a function and using sapply to apply the mean to each neuron in the session. After performing this operation for Sessions 1 and 10, we could calculate the summary statistics. Session 1's mean number of spikes was 1.540 with median of 1.469, whereas session 10 had a mean of 1.188 and median of 1.191. This suggests that session shows higher neural activity, since session 10 had both a lower mean and median value.

```
## 
##  Wilcoxon rank sum test with continuity correction
## 
## data:  s1_avg_spikes and s10_avg_spikes
## W = 43489, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

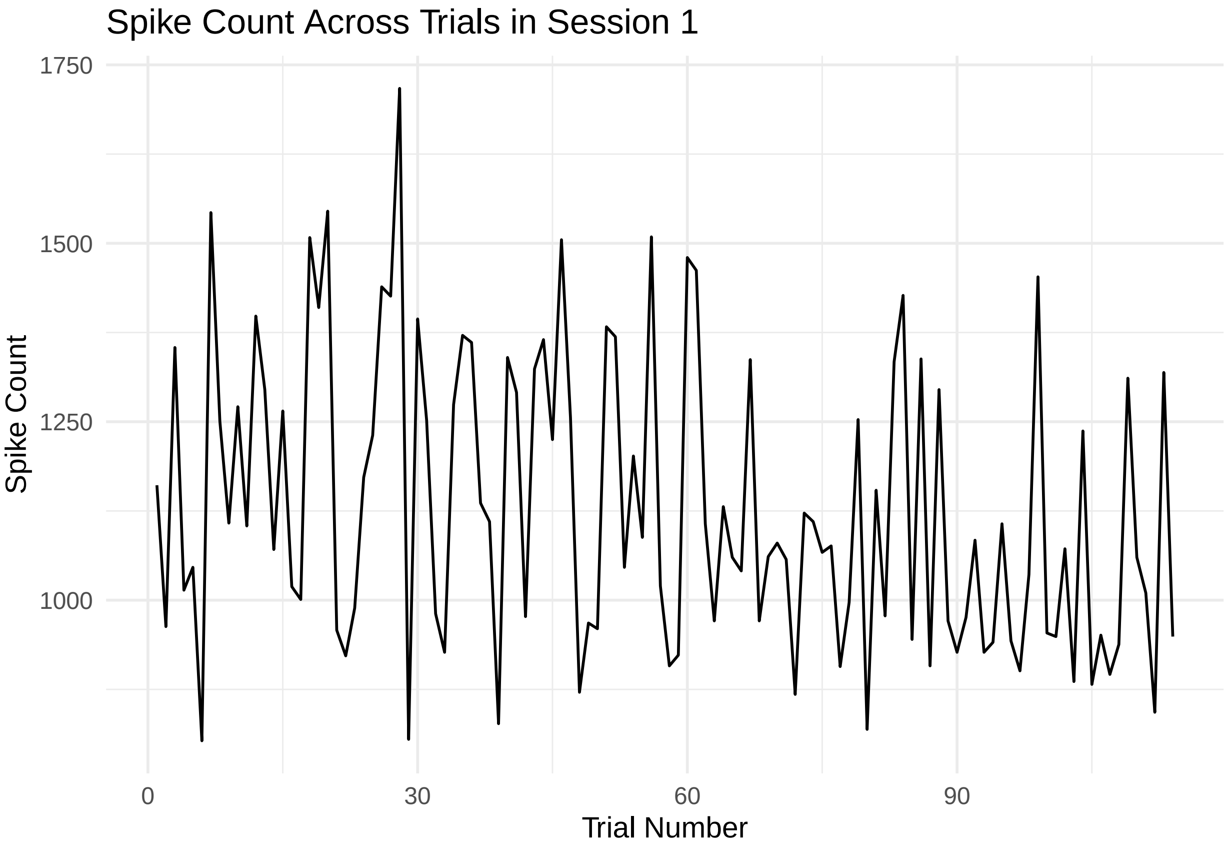
To test if this decrease is statistically significant, we can perform the Wilcoxon rank sum test. This test is used to compare the means of two groups, in this case the number of spikes in session 1 and session 10. The null hypothesis is that the two groups have the same mean. If p-value is less than 0.05, we can conclude that the average number of spikes from Session 1 is statsically different from Session 10. In this case, our p-value is less than 2.2e-16, which is less than 0.05. Thus we can reject the null and conclude that average number of spikes in session 1 is statistically different from session 10

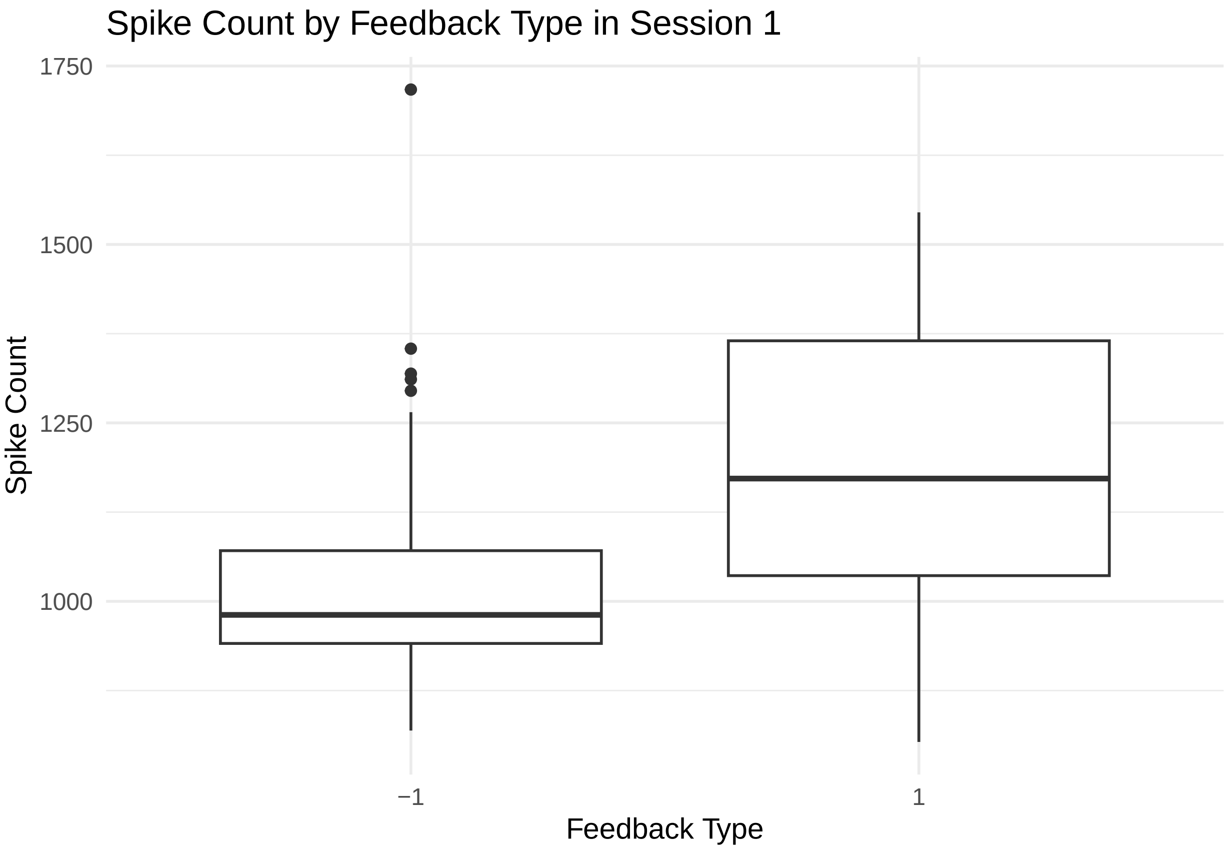
Overall, we determined there's a change in neural activity over time, and an indication that mice respond differently to later stages of the experiment.

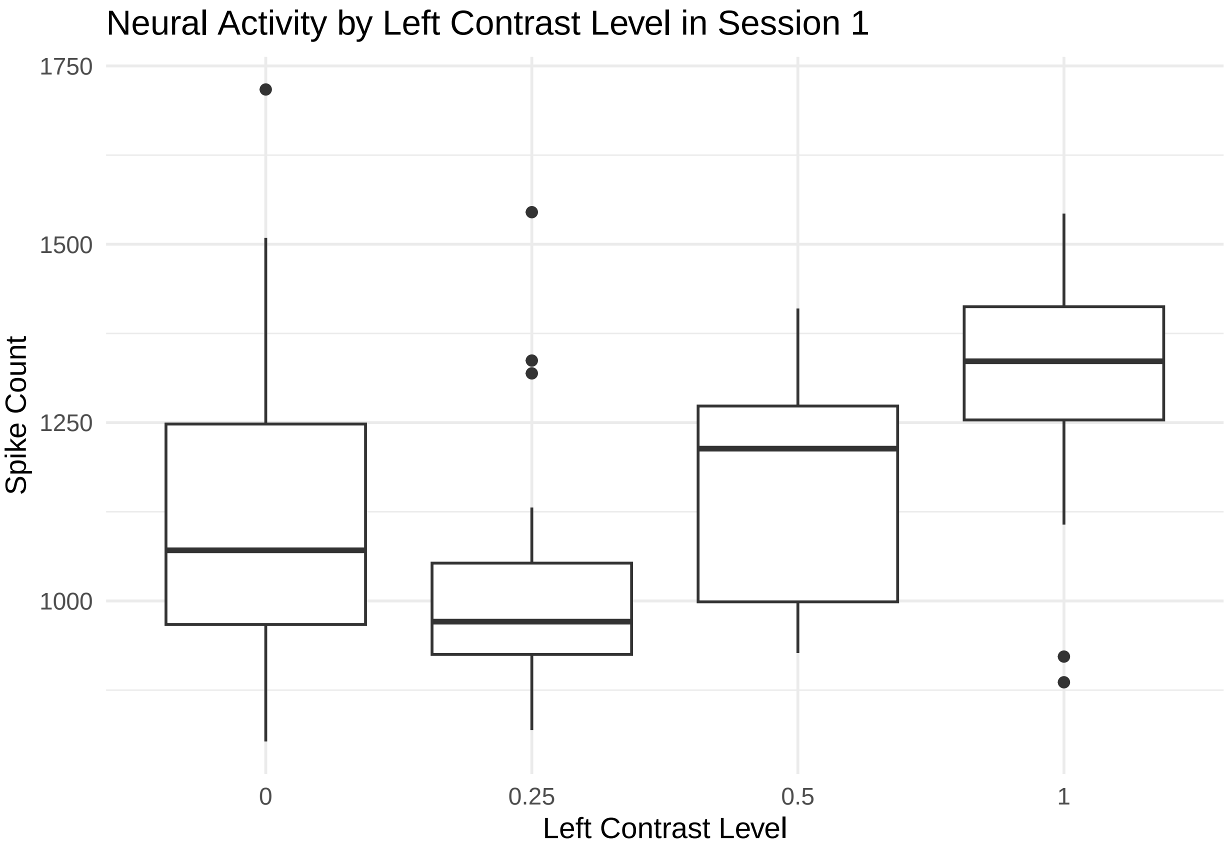
Neural Activity Across Experiments

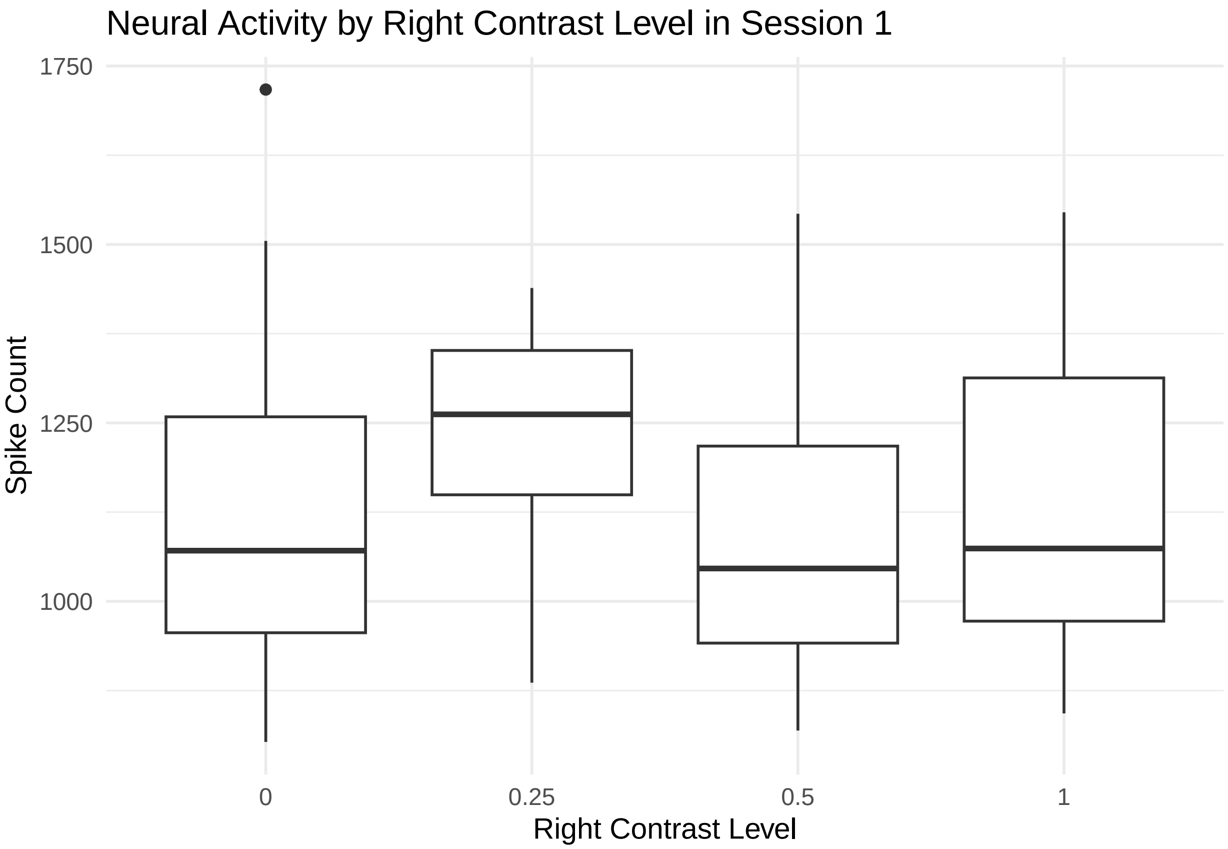
```
##     trial spike_count feedback contrast_left contrast_right
## 1       1        1161        1          0.00           0.50
## 2       2         963        1          0.00           0.00
## 3       3        1354       -1          0.50           1.00
## 4       4        1014       -1          0.00           0.00
## 5       5        1046       -1          0.00           0.00
## 6       6         803        1          0.00           0.00
## 7       7        1543        1          1.00           0.50
## 8        8        1251        1          0.50           0.00
## 9        9        1108        1          0.00           0.00
## 10      10        1271        1          0.50           0.25
## 11      11        1104        1          0.50           0.00
## 12      12        1398        1          0.00           1.00
## 13      13        1295       -1          1.00           1.00
## 14      14        1071       -1          0.00           0.00
## 15      15        1265       -1          0.00           0.00
## 16      16        1019       -1          0.00           0.00
## 17      17        1001        1          0.00           0.00
## 18      18        1508        1          0.00           0.50
## 19      19        1410        1          0.50           0.25
## 20      20        1545        1          0.25           1.00
## 21      21         958        1          0.00           0.00
## 22      22         922       -1          1.00           0.50
## 23      23         989       -1          0.00           0.00
## 24      24        1172        1          0.00           0.00
## 25      25        1231        1          0.00           0.25
## 26      26        1439        1          1.00           0.25
## 27      27        1426        1          0.00           0.50
## 28      28        1717       -1          0.00           0.00
## 29      29         805        1          0.00           0.00
## 30      30        1394        1          0.00           0.50
## 31      31        1252        1          0.50           0.00
## 32      32         981       -1          0.00           0.00
## 33      33         927        1          0.00           0.00
## 34      34        1274        1          0.50           0.50
## 35      35        1371        1          0.00           0.25
## 36      36        1361        1          0.00           1.00
## 37      37        1136        1          0.00           0.50
## 38      38        1110        1          0.25           1.00
## 39      39         827        1          0.00           0.00
## 40      40        1340        1          1.00           0.00
## 41      41        1291        1          0.00           0.50
## 42      42         977       -1          0.00           0.50
## 43      43        1324        1          1.00           0.00
## 44      44        1365        1          1.00           0.25
## 45      45        1225        1          0.50           1.00
## 46      46        1505        1          1.00           0.00
## 47      47        1256       -1          1.00           1.00
## 48      48         871        1          0.00           0.00
## 49      49         968       -1          0.25           1.00
## 50      50         960        1          0.25           1.00
## 51      51        1383        1          0.50           0.00
## 52      52        1369        1          1.00           0.00
## 53      53        1046       -1          0.25           0.50
## 54      54        1202       -1          0.50           0.00
## 55      55        1088        1          0.00           0.50
## 56      56        1509        1          0.00           1.00
## 57      57        1020       -1          0.25           1.00
## 58      58         908       -1          0.25           1.00
## 59      59         923       -1          0.25           1.00
## 60      60        1480        1          0.00           1.00
## 61      61        1462        1          0.00           1.00
## 62     62        1107        1          0.00           1.00
## 63     63         971       -1          0.25           0.25
## 64     64        1131       -1          0.25           1.00
## 65     65        1060       -1          0.25           1.00
## 66     66        1041       -1          0.25           1.00
## 67     67        1337        1          0.25           1.00
## 68     68         971        1          0.00           0.00
## 69     69        1061        1          0.50           0.00
## 70     70        1080        1          0.00           0.00
## 71     71        1057        1          0.00           1.00
## 72     72         868        1          0.00           0.00
## 73     73        1122        1          0.00           0.25
## 74     74        1110        1          0.00           0.00
## 75     75        1067        1          0.00           0.50
## 76     76        1076        1          0.00           1.00
## 77     77         907       -1          0.25           0.50
## 78     78         997        1          0.00           0.50
## 79     79        1253        1          1.00           0.25
## 80     80         819       -1          0.25           0.50
## 81     81        1154        1          0.00           0.00
## 82     82         978        1          0.50           0.50
## 83     83        1334        1          1.00           0.00
## 84     84        1427        1          1.00           0.00
## 85     85         945       -1          0.00           0.50
## 86     86        1338        1          1.00           0.00
## 87     87         908       -1          0.00           1.00
## 88     88        1295        1          0.50           0.25
## 89     89         971       -1          0.25           1.00
## 90     90         927       -1          0.25           1.00
## 91     91         976       -1          0.25           1.00
## 92     92        1084       -1          0.25           1.00
## 93     93         927        1          0.50           0.25
## 94     94         941       -1          0.50           0.00
## 95     95        1107        1          1.00           0.00
## 96     96         943       -1          0.50           0.00
## 97     97         901        1          0.00           0.50
## 98     98        1036        1          0.25           1.00
## 99     99        1453        1          1.00           0.00
## 100   100         954       -1          0.25           0.00
## 101   101         949       -1          0.50           0.50
## 102   102        1072       -1          0.50           1.00
## 103   103         886       -1          1.00           0.25
## 104   104        1237        1          1.00           0.25
## 105   105         882       -1          0.25           0.50
## 106   106         951        1          0.00           0.00
## 107   107         896        1          0.25           0.00
## 108   108         938       -1          0.00           0.50
## 109   109        1311       -1          0.00           0.25
## 110   110        1060       -1          0.00           0.50
## 111   111        1010       -1          0.25           1.00
## 112   112         843       -1          0.25           1.00
## 113   113        1319       -1          0.25           1.00
## 114   114         949       -1          0.25           1.00
```

Moving onwards to our next section, we can examine more deeply about the neural activity across different experiments. Some areas we can explore neural activity across trials, by feedback type, and by contrast levels. To do this, we need to keep track of the neural activity, "spike count", and how it varies across different trials. To identify trends through visualizations, storing the trial information in a dataframe with its associated spike count, feedback type, and contrast levels will be helpful.




Spike Count by Feedback Type in Session 1


Neural Activity by Left Contrast Level in Session 1



Homogeneity and Heterogeneity Across Sessions and Mice

Another thing we can examine in this exploratory section is to examine whether the spiking activity is homogeneous *(constant)* or heterogeneous *(varies significantly)* across different sessions. This can help us determine whether the neural activity is steady or changes over time.

To address this question, we can apply the Kruskal-Wallis test, which evaluates whether the variance in spike counts is equal across different mice.

```
##
##  Kruskal-Wallis rank sum test
##
## data:  spikes_avg by mouse
## Kruskal-Wallis chi-squared = 4.9616, df = 3, p-value = 0.1746
```

After performing the Kruskal-Wallis test, our results are as shown above. The null hypothesis for the Kruskal-Wallis test is that the distribution of average spike counts is the same across all mice. Here our p-value is 0.1746 is greater than 0.05, which means we fail to reject the null hypothesis. In other words, there is not enough evidence that spike counts differ significantly between the mice. So therefore, we can't conclude that different mice show distinct spiking behaviors.

Data Integration

The goal of this section is to combine data from different sessions and trials into a combined format. This will allow us to improve the model performance and conduct a better analysis. Specifically we'll focus on identifying common neural activity patters across sessions, normalizing and standardizing the data, and merging the session data into a single dataset.

Merging Session Data

After testing to see if the distribution of the average spike counts is different across all mice, we concluded that there isn't enough evidence to suggest that spike counts differ significantly between the mice. Hence, we can merge the data from all sessions. This merging process will be a bit different from the intial one at the beginning. That merge was to provide a summary of the data for a trial by trial analysis, whereas this will give a more overall picture using a session-level summary.

```
## tibble [5,081 x 7] (S3: tbl_df/tbl/data.frame)
##  $ session       : Factor w/ 18 levels "session 1","session 2",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ contrast_left : num [1:5081, 1] 0 0 0.5 0 0 0 1 0.5 0 0.5 ...
##  $ contrast_right: num [1:5081, 1] 0.5 0 1 0 0 0 0.5 0 0 0.25 ...
##  $ decision      : Factor w/ 3 levels "left","right",..: 2 3 2 3 3 3 1 1 3 1 ...
##  $ neuron        : int [1:5081] 734 734 734 734 734 734 734 734 734 734 ...
##  $ spikes_avg    : num [1:5081] 1130 1130 1130 1130 1130 ...
##  $ feedback      : Factor w/ 2 levels "-1","1": 2 2 1 1 1 2 2 2 2 2 ...
```

```
## # A tibble: 6 x 7
##   session   contrast_left[,1] contrast_right[,1] decision neuron spikes_avg
##   <fct>                 <dbl>              <dbl> <fct>     <int>      <dbl>
## 1 session 1               0                  0.5 right       734      1130.
## 2 session 1               0                  0   tied        734      1130.
## 3 session 1               0.5                1   right       734      1130.
## 4 session 1               0                  0   tied        734      1130.
## 5 session 1               0                  0   tied        734      1130.
## 6 session 1               0                  0   tied        734      1130.
## # i 1 more variable: feedback <fct>
```

Identifying Shared Patterns Across Sessions

- Extract key features such as spike rates, contrast levels, and feedback types.
- Compare neural responses across sessions to see if certain patterns repeat.

Normalizing and Standardizing Data

The next step in our Data Integration process is to normalize or standardize the data to ensure that the features are on the same scale and don't dominate the model. This can be done using techniques like Z-score for standardization or Min-Max scaling for normalization. To determine which method to use, we can check based on the machine learning algorithms we're implementing. Though we haven't touched the machine learning portion of the project, the algorithms we'll be using are Logistic Regression, Random Forest, and Gradient Boosting. Out of the three, Logistic Regression and Gradient Boosting are sensitive to feature scaling and work best require standardization, while Random Forest doesn't. Thus we'll apply Z-score scaling to our data.

```
##         session      contrast_left.V1    contrast_right.V1   decision   
##  session 10: 447   Min.   :0.0000000   Min.   :0.0000000   left :1762  
##  session 15: 404   1st Qu.:0.0000000   1st Qu.:0.0000000   right:1633  
##  session 9 : 372   Median :0.2500000   Median :0.2500000   tied :1686  
##  session 11: 342   Mean   :0.3418618   Mean   :0.3241488               
##  session 12: 340   3rd Qu.:0.5000000   3rd Qu.:0.5000000               
##  session 13: 300   Max.   :1.0000000   Max.   :1.0000000               
##  (Other)   :2876                                                       
##      neuron         spikes_avg     feedback
##  Min.   : 474.0   Min.   : 497.9   -1:1473
##  1st Qu.: 698.0   1st Qu.: 826.6   1 :3608
##  Median : 857.0   Median :1195.1
##  Mean   : 909.8   Mean   :1207.2
##  3rd Qu.:1090.0   3rd Qu.:1382.8
##  Max.   :1769.0   Max.   :2415.9
##
```

Now to determine which features to apply the Z-score scaling, we can look at the summary statistics of the variables. Here we see spikes average having a large spread from 497.9 to 2415.9, so standardizing this would be helpful. Other variables are either categorical or count data, so they don't need to be standardized.

```
##        session      contrast_left.V1    contrast_right.V1  decision
##  session 10: 447   Min.   :0.0000000   Min.   :0.0000000   left :1762
##  session 15: 404   1st Qu.:0.0000000   1st Qu.:0.0000000   right:1633
##  session 9 : 372   Median :0.2500000   Median :0.2500000   tied :1686
##  session 11: 342   Mean   :0.3418618   Mean   :0.3241488
##  session 12: 340   3rd Qu.:0.5000000   3rd Qu.:0.5000000
##  session 13: 300   Max.   :1.0000000   Max.   :1.0000000
##  (Other)   :2876
##      neuron          spikes_avg.V1     feedback
##  Min.   : 474.0   Min.   :-1.6170374   -1:1473
##  1st Qu.: 698.0   1st Qu.:-0.8677296   1 :3608
##  Median : 857.0   Median :-0.0276017
##  Mean   : 909.8   Mean   : 0.0000000
##  3rd Qu.:1090.0   3rd Qu.: 0.4002496
##  Max.   :1769.0   Max.   : 2.7552932
##
```

After applying the scaling, we see that the spikes_avg variable has been standardized. The mean is now 0 and the standard deviation is 1. This will help ensure that the features are on the same scale and don't dominate the model.

Now another technique in the Data Integration process is applying PCA (Principal Component Analysis). This helps reduce the dimensionality of the data. To determine if PCA is necessary, we can look at the correlation between the features *(multicollinearity)*.

```
##                 contrast_left contrast_right   spikes_avg
## contrast_left     1.00000000   -0.045362408  0.028195565
## contrast_right   -0.04536241    1.000000000 -0.004715326
## spikes_avg        0.02819557   -0.004715326  1.000000000
```

After calculating the variance and correlation between the features, we see that the correlation between the features is low. For instance, contrast_left and constrast_right have a correlation of -0.0453 which is very low, so we can conclude that PCA isn't necessary for this dataset.

Predictive Modeling

The goal of this section is to train predictive models to classify trial outcomes (success/failure) based on neural activity and stimulus conditions. Specifically we'll focus on:

- Implementing several models: Logistic Regression, Random Forest, and Graident Boosting (XGBoost)
- Evaluating models performance on training data using cross-validation, precision, recall, and other metrics

Before we start applying the models, lets split the data into training and testing sets. We'll use 80% of the data for training and 20% for testing.

Logistic Regression Model

Logistic Regression: This model is used to predict binary outcomes based on a set of independent variables. Lets establish a logistic regression model and then see its performance on the training data.

```
## [1] "Training Accuracy:  0.711510083620266"
```

Training Accuracy for Logistic Regression was 71%

Random Forest Model

Random Forest: This model is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes as the prediction. Lets establish a baseline model using Random Forest and then apply hyperparameter tuning to optimize the model performance.

Establish a model using Random Forest:

```
## [1] "Training Accuracy:  0.716182980816527"
```

Training Accuracy for Random Forest was 97%

Extreme Gradient Boosting Model (XGBoost)

Extreme Gradient Boosting: This model is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. It's different from normal gradient boosting in that it uses a more regularized model to control overfitting.

Establish a model using XGBoost

```
## [1]  train-logloss:0.629316
## [2]  train-logloss:0.594075
## [3]  train-logloss:0.572322
## [4]  train-logloss:0.557698
## [5]  train-logloss:0.549696
## [6]  train-logloss:0.544386
## [7]  train-logloss:0.540166
## [8]  train-logloss:0.536679
## [9]  train-logloss:0.533946
## [10] train-logloss:0.532143
```

```
## [1] "Tuned Training Accuracy:  0.722085587801279"
```

Training Accuracy for XGBoost was 72%

Overall, after calculating the training accuracy for all the models, we determined that Random Forest was our best model. Thus lets evaluate its performance more deeply in the next section.

Evaluating Training Performance

Now that we have trained the models, we need to assess how well they performed on the training data using different performance metrics. This will help us understand which model is the most effective and should be used on the test data.

- Confusion Matrix: Breakdown of true positives, true negatives, false positives, and false negatives

```
## Confusion Matrix and Statistics
##
```

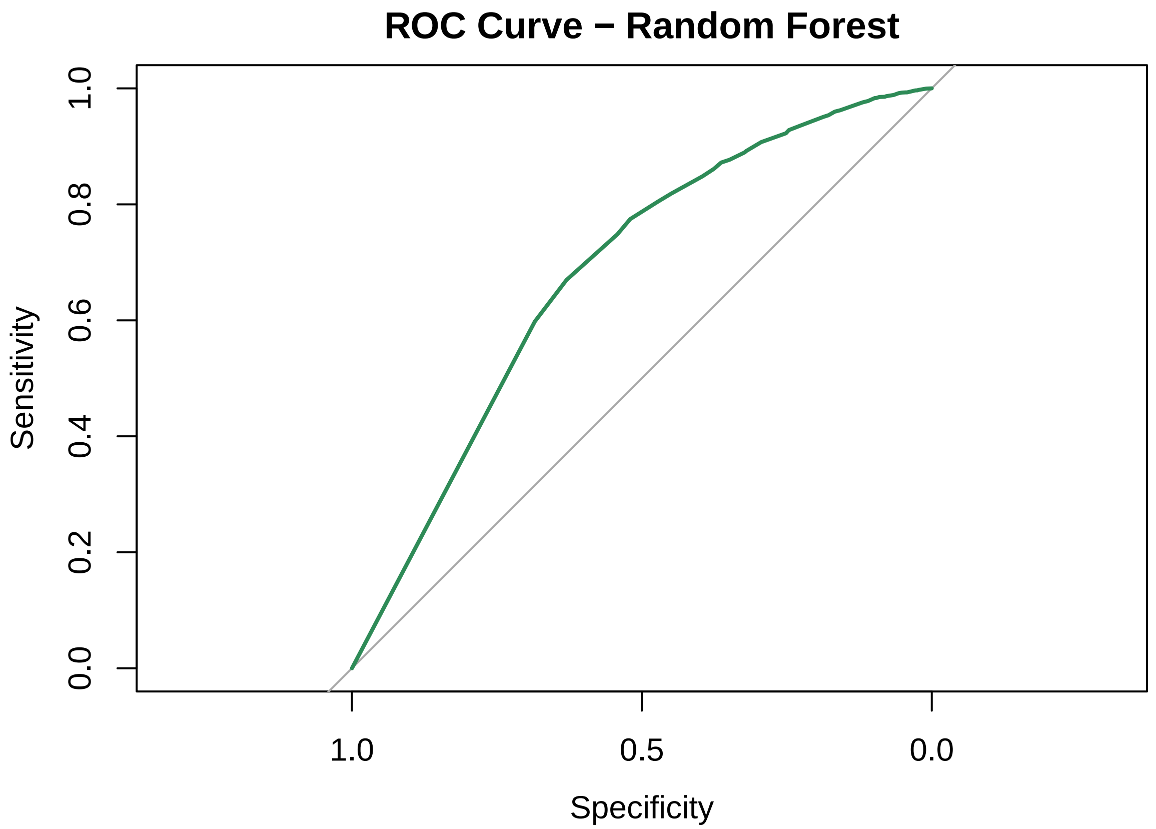
```
##           Reference
## Prediction   -1    1
##         -1   29   10
##         1  1144 2883
##
##                Accuracy : 0.7162
##                  95% CI : (0.7021, 0.73)
##     No Information Rate : 0.7115
##     P-Value [Acc > NIR] : 0.2614
##
##                   Kappa : 0.0298
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.024723
##             Specificity : 0.996543
##          Pos Pred Value : 0.743590
##          Neg Pred Value : 0.715918
##              Prevalence : 0.288490
##          Detection Rate : 0.007132
##    Detection Prevalence : 0.009592
##       Balanced Accuracy : 0.510633
##
##        'Positive' Class : -1
##
```

- ROC Curve and AUC Score: Measuring classification performance across different thresholds

```
## Setting levels: control = -1, case = 1
```

```
## Setting direction: controls < cases
```

```
## [1] "AUC Score: 0.681176069820766"
```



Misclassification Rate: Percentage of incorrect predictions

```
## [1] "Misclassification Rate: 0.283817019183473"
```

Prediction Performance on Test Sets

The goal of this section is to evaluate our best performing model using two distinct test sets, randomly selected from Session 1 18, which will help us assess the model's generalization performance.

Test Set 1: Session 1

```
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## Levels: -1 1
```

```
## Warning: Unknown or uninitialised column: `feedback_type`.
```

```
## [1] "Accuracy:  NaN"
```

Test Set 2: Session 18

```
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20 
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40 
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60 
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80 
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
## Levels: -1 1
```

```
## Warning: Unknown or uninitialised column: `feedback_type`.
```

```
## [1] "Accuracy for Session 18:  NaN"
```

Discussion

What Went Well: The Exploratory Data Analysis (EDA) was a smooth process, and I found it useful to explore the dataset's statistical summaries. These summaries provided key insights into the distribution of variables and potential relationships within the data. The process of training the model also went well. Once I had preprocessed the data and selected relevant features, the training was relatively straightforward, and I saw good performance during the training phase.

Challenges Encountered: The major hurdle I faced occurred during the model evaluation phase, where I encountered an issue that led to model crashes. This happened primarily when trying to evaluate the model and calculate the performance metrics like accuracy, precision, and the confusion matrix. Towards the end of the project, I couldn't troubleshoot this effectively, which caused delays and some frustration.

The issue seemed to arise from the confusion matrix and accuracy calculation. The common error I encountered was "Accuracy: NaN", with the confusion matrix showing no true positives or true negatives, resulting in NaN values for various performance metrics (e.g., sensitivity, specificity).

Possible Causes:

- Mismatch in Factor Levels: The predicted class labels and actual labels might not have had the same factor levels, causing issues when calculating the confusion matrix.
- Empty or Missing Predictions: There might have been missing or empty predictions for certain classes, leading to the failure of performance metric calculations.
- Uninitialized Columns: Some warnings like "Unknown or uninitialized column: feedback_type" suggested that the necessary column might not have been properly handled during the evaluation, causing errors when trying to access it.

Unfortunately, I couldn't resolve these issues in time, but identifying these possible areas would help troubleshoot the problem in future iterations.