STA 141A Final Project

Rohan Pillay

ID: 920674736

March 17, 2025

GitHub Link

Abstract:

Being able to understand how neural activity is a driving force in decision making is very important in the field of neuroscience. This project looks at how brain activity and decision making are related as neural activity is captured in the form of spike trains from mice performing certain tasks. We look at the recorded neural responses data of 10 mice as they are presented with visual prompts. Our main goal is to develop a predictive model to be able to forecast trial outcomes based on observed patterns in neural activity.

1. Introduction

As mentioned, being able to understand how neural activity in the brain affects decision making is vital in neuroscience and neural development. The study being conducted explores neural activity from mice who need to perform a decision making task in response to contrasting visual stimuli, resulting in a successful decision or an unsuccessful one. The dataset we look at is from a study done by Steinmetz et al. (2019). The part of the dataset focused on in this project is recordings from 18 sessions where 4 mice (Cori, Forssmann, Hench, and Lederberg) made decisions by turning a wheel based on visual stimuli on two screens. Electrodes were placed into the mice, and neural activity was recorded from their visual cortex through all the trials.

The main goal for this project is to try to build a model that can accurately predict trial outcomes. In other words we want to try to be able to predict the mice's behavior and whether they will make the correct or incorrect decisions.

Data Summary

This table provides us with a complete overview of the 18 different sessions conducted on the four mice (Cori, Forssmann, Hench, and Lederberg). This is from the dataset collected by Steinmetz et al. (2019). We see that each of the sessions include the number of trials, neurons, brain areas, and the mean average firing rate. The distribution of sessions between the four different mice are uneven as Lederberg has six sessions (highest), followed by both Forssmann and Hench with four sessions each, and lastly Cori with three sessions (lowest). The number of trials per session vary between 114 and 447. The number of neurons varies between 474 and 1769. The number of brain areas ranges from 5 to 15, and the mean average firing rate varies between 0.0166 and 0.0614. Using this data, we will move into the exploratory analysis where we will try to understand the neural patterns of the different mice across the sessions. This in turn will help us with being able to predict behavioral outcomes.

Table 1: Session Data Summary

session	mouse	num_trials	num_neurons	num_brain_areas	mean_firing_rate
session1	Cori	114	734	8	0.0384910
session2	Cori	251	1070	5	0.0316408
session3	Cori	228	619	11	0.0558484
session4	Forssmann	249	1769	11	0.0210389
session5	Forssmann	254	1077	10	0.0278957
session6	Forssmann	290	1169	5	0.0165797
session7	Forssmann	252	584	8	0.0353844
session8	Hench	250	1157	15	0.0413892
session9	Hench	372	788	12	0.0396842
session10	Hench	447	1172	13	0.0296884
session11	Hench	342	857	6	0.0312622
session12	Lederberg	340	698	12	0.0415816
session13	Lederberg	300	983	15	0.0614420
session14	Lederberg	268	756	10	0.0251652
session15	Lederberg	404	743	8	0.0366021
session16	Lederberg	280	474	6	0.0262594
session17	Lederberg	224	565	6	0.0291453
session18	Lederberg	216	1090	10	0.0274110

2. Exploratory Analysis

Data Structure Analysis

As mentioned, the dataset we are looking at has 18 experimental sessions with four mice in particular being studied - Cori, Forssmann, Hench, and Lederberg. The number of trials across each of the sessions varies. Session 1 has the fewest trials recorded with about 110-120 and Session 10 has the most trials with about 450. When looking at figure 1, we can also deduce that most sessions have between about 200-300 trials. Along with this, we see Cori generally had fewer trials with one low potential outlier during session 1, Forssmann had an average number of trials, Hench's trials were relatively higher, and Lederberg also had an average number of trials with one high potential outlier during session 15.

Along with the trials, the number of neurons recorded varied quite a bit through the sessions. Session 16 had the fewest number of neurons recorded with fewer than 500, while session 4 had the highest with just over 1750. Most of the sessions had between about 500-1100 neurons recorded. Generally when looking at each mice individually, the number of neurons recorded were relatively similar, each had a couple sessions where they were in the average, and one maybe a little lower or higher. The one standout obviously is session 4 when Forssmann had well over the average number of neurons recorded.

Figure 1: Number of Trials per Session

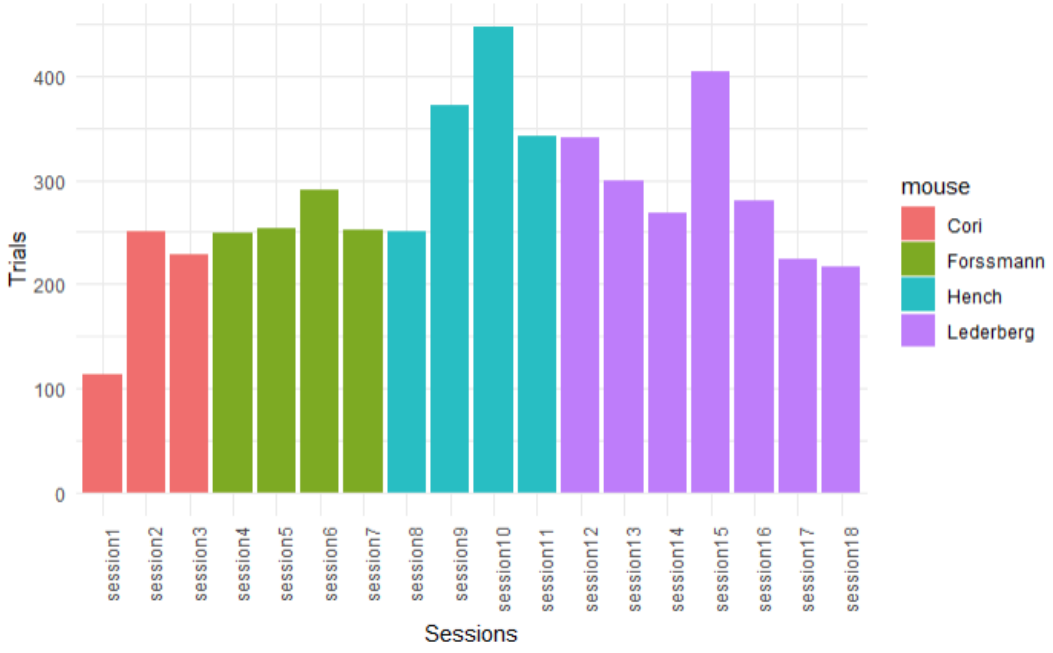
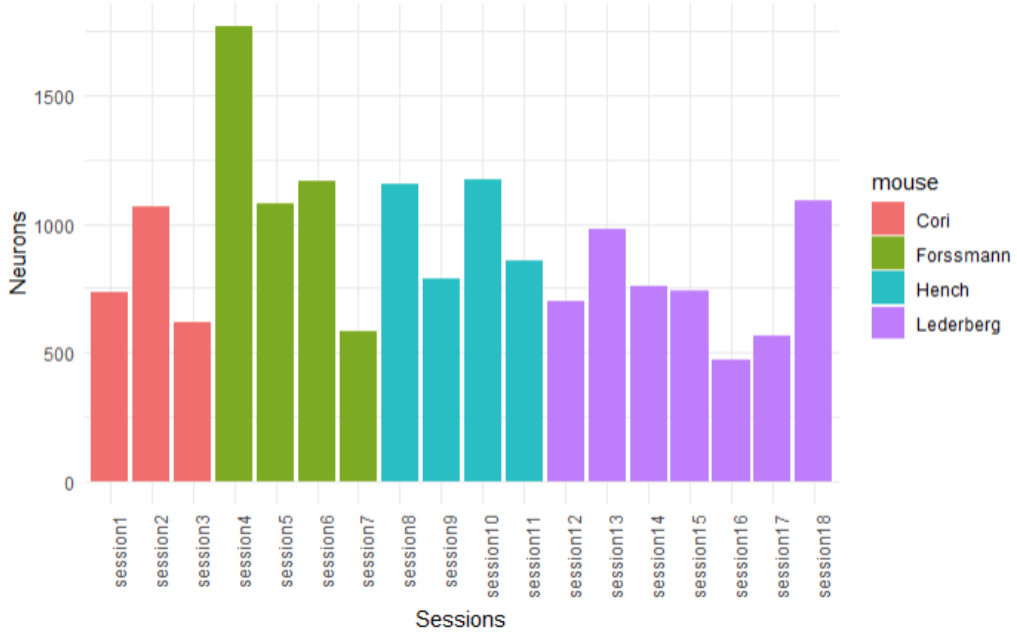


Figure 2: Number of Neurons per Session

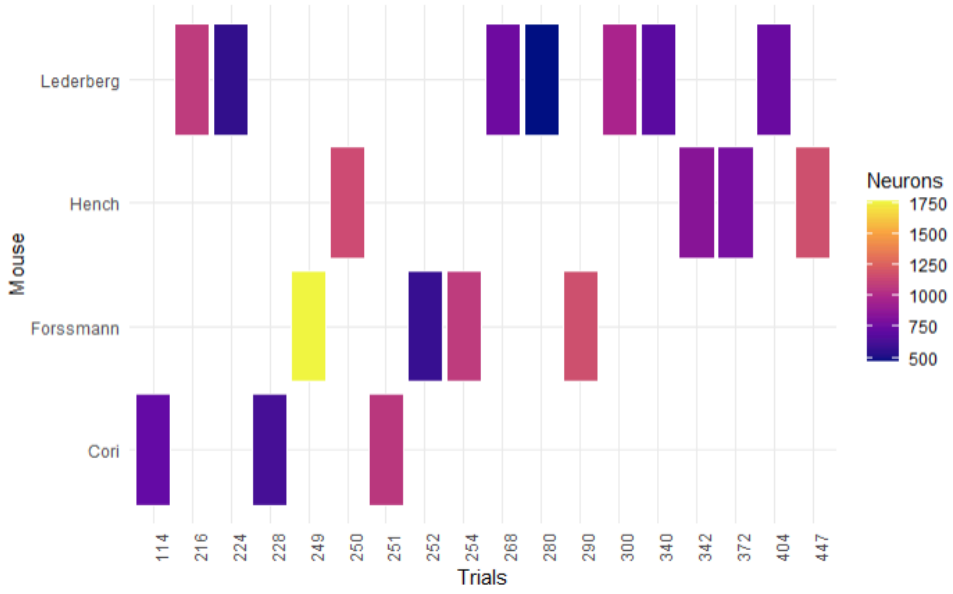


Neural Activity Across Trials

To examine patterns in visual cortex activity that happens right before the mice make a decision, we can look at how well the neurons respond during trials. When there is more difference between the visual stimuli the mice look at, in turn as well, the neurons for each mouse will respond accordingly and strongly. This helps the mice make decisions more accurately.

Looking at the counts of Neurons recorded over Trials, we see that generally most of the time, the neurons recorded were about average on the heat map scale (about 500-1000). Forssmann, seemed to be the one mouse with much higher neurons recorded on average with one of counts being about 1750 and the other being about 1250.

Figure 3: Number of Neurons Recorded Over Certain Trials (heatmap)



Changes Across Sessions

When grouping the sessions by each mouse individually, one can visibly see patterns that correlate to each mouse. Each mice showed consistent neural patterns throughout the sessions despite the mice having different neuron counts, trial numbers, etc. This allows us to say that the variation that we observe is due to unique differences between each of the mice.

When looking at the neurons for the mice from different brain regions, we again notice the variability through the sessions. Certain brain regions of the mice showed much stronger connections to behavioral results. Thus, focusing on the specific regions that show the stronger connections can help us improve our prediction models to better accurately predict behavioral outcomes.

Stimuli Conditions

Stimuli with higher contrast tend to generally cause stronger neural responses in most of the brain regions. When the contrast values are zero meaning no stimulus, there are still certain neural patterns that correlate with the correct response of hold still. It isn't just reduced neural activity when no stimulus occurs which is surprising. One would think that without stimulus the mice would just have reduced activity, but instead we see unique patterns. When looking at equal contrast on either side, expectedly, neural activity is much more variable.

There are quite different neural patterns between a successful and unsuccessful trial. This stays true even with completely identical stimulus conditions. Successful trials normally showed more precise neural activity, suggesting that factors beyond just the visual stimulus at hand play a part in the mice's behavior.

Analyzing the Average Neural Activity per Trial

When we look at the mean spike rates per trial throughout the sessions, we see a lot of variability. The average spike count ranges from about 0.025 to 0.045. With such variability, it suggests that the engagement of each of the mice's neurons differs significantly between each trial. This could mean multiple things from changes in the visual stimuli, to changes in attention or even learning for the mice. Being able to understand if the mice throughout the trials learn and adapt is key to predicting their behaviors.

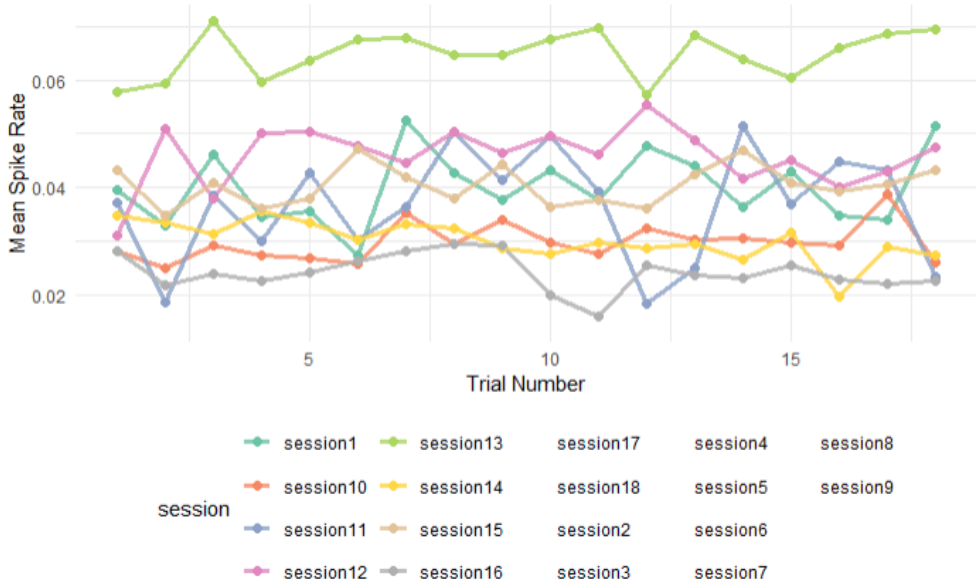
Looking at the figure 4, we see that throughout the trials, the mean spike rate was higher than the average for session 18. This could be due to enhanced learning, where the mice have adapted and now are producing more neurons. It also could have been because the visual stimuli caused the mice to release more neurons.

During the early trials (1-5), several of the sessions showed increased variability, and many sessions appeared to be stabilizing their activity patterns during these trials. In the middle trials (6-10), notably for trials 8-10, many of the sessions showed a visible dip in neural activity. Along with this, there was generally less variability between the sessions

in this trial range. In the later trials (11-15), several of the sessions showed an increase in neural activity around trials 12-14, and a couple sessions like sessions 13, maintained their elevated activity. In the final trials (16-18), many of the sessions showed an upward trend including session 1 (strong increase) and session 12 to name a few.

From this we can deduce that throughout the trials, the mice across the sessions show what could be adaptability as there were gradual changes in the spike rate.

Figure 4: Mean Spike Rates across Trials



Analyzing the Mean Neural Firing Rate per Session

When looking at figure 5, we see that it shows us the mean neural firing rates across the 18 sessions. This graph again shows us the constant variability in the neural activity between the sessions. The average spike rates range from about 0.016 to 0.061, telling us that there are considerable differences in the overall neural engagement for the mice throughout the 18 sessions.

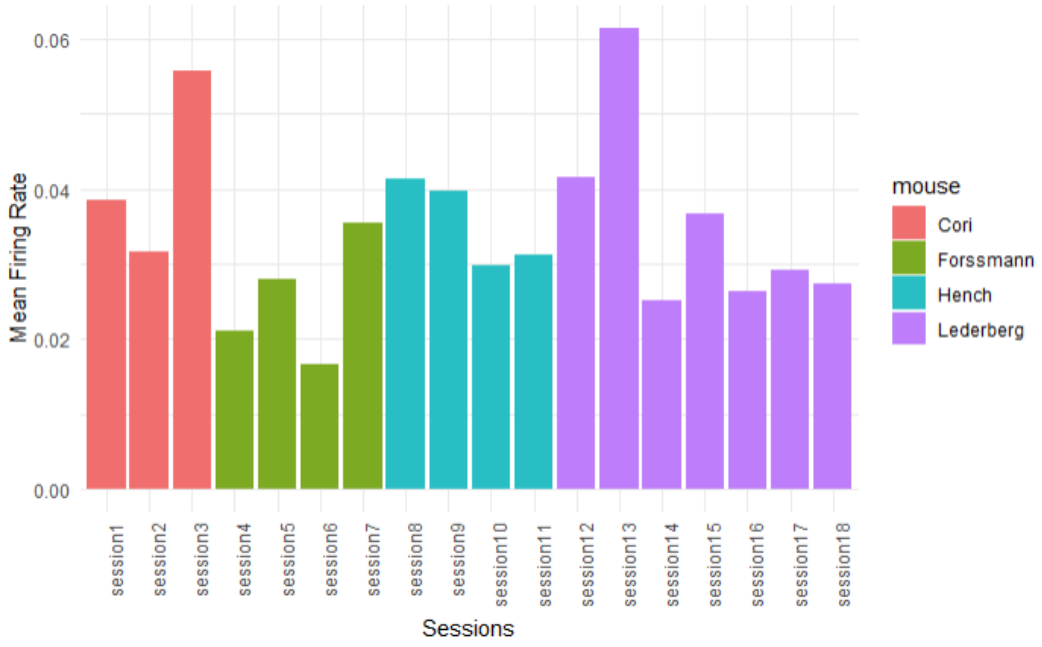
Session 13 stands out as it has the highest mean firing rate (0.061), with session 3 being not far behind (mean firing rate of 0.056). What this could tell us is that during these sessions, the visual stimuli could be more engaging or different than before, the mice could be in a higher state of attentiveness, the electrodes that read the neural

activity are in better places, etc. On the other hand, the lowest mean firing rate (0.017) was from Session 6 with session 4 being close by as well (mean firing rate 0.021). This could reflect the opposite, where the mice had less attentiveness or suboptimal electrode placements.

Out of the four mice, Lederberg was the one with the highest mean firing rate (0.061) and Forssmann was the one with the least (0.017). Forssmann's mean firing rate was on average the lowest out of the four, while Cori or Lederberg had the most. This could give us valuable information and insights on the mice. Cori and Lederberg tend to have higher mean neural firing rates, meaning they could be more attentive mice, and Forssmann could be a little less attentive mouse in comparison to the others. One most also take into account other factors here, such as the stimuli provided to each of the mice through the sessions and electrode placements to make sure these assumptions are viable.

Due to there being a lack of a clear trend across the sessions, this could mean that this variability being observed isn't due to standardized changes over time such as learning effects, but instead is due to sessions specific aspects.

Figure 5: Mean Neural Firing Rate per Session



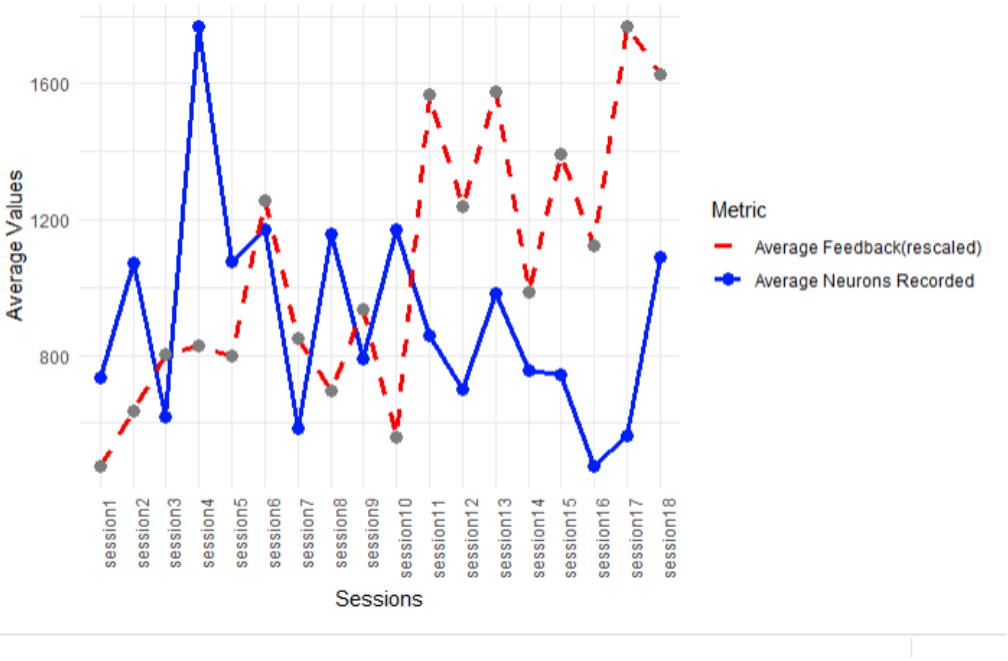
3. Data Integration

Shared Patterns Across Sessions

When looking at data integration, we take a look at the shared patterns across the sessions. When looking at figure six, we see that there is a disconnect between average feedback (scaled) and average neurons recorded. They both tend to fluctuate quite a bit and never follow a similar pattern. Due to this, the number of neurons recorded may not be a very strong predictor of performance outcomes for the mice.

What we can say from this graph is that for average feedback, the later trials tended to have a much higher average feedback which tells us that the learning effects of the mice should be taken into account. Along with this, the average neurons recorded tended to follow a more random fluctuation style, while the average feedback, as the sessions progressed increased.

Figure 6: Average Feedback and Average Neurons Recorded



Neural Activity Trends

We can now look at the average firing rate between session 1 and session 18. This will show us if the mice evolve or adapt as the sessions go on, and we can also look at the average firing rate through all the sessions. There are two key questions at hand here:

1. How does the average firing rate change over the trials?
2. How can we use the neural firing patterns to see if the mice adapt or evolve over the trials?

In Session 1 we see that the overall firing rate is higher in comparison to session 18. We also see a U-shape trend in session 1. This U-shape increases initially then decreases. In session 18 we see that there is a lower but more consistent trend and decreases as the trials go on.

Looking at the neural activity for the trials over all sessions, although harder to tell, based on previous assessments, session 3 and session 18 have much higher average firing rates across the trials compared to the other sessions. Along with that, a lot of the sessions have average firing rates between about 0.03 and 0.05.

This shows us that with experience, the average firing rate will fluctuate less and tend to be more efficient. This emphasizes how important trial position in each session is. The trials given at the beginning for session 1 (trials about 1-50) could be different from the trials given at the same time for session 18, and potentially some of the other sessions as well.

Figure 7: Neural Activity over Trials (Session 1 and Session 18)

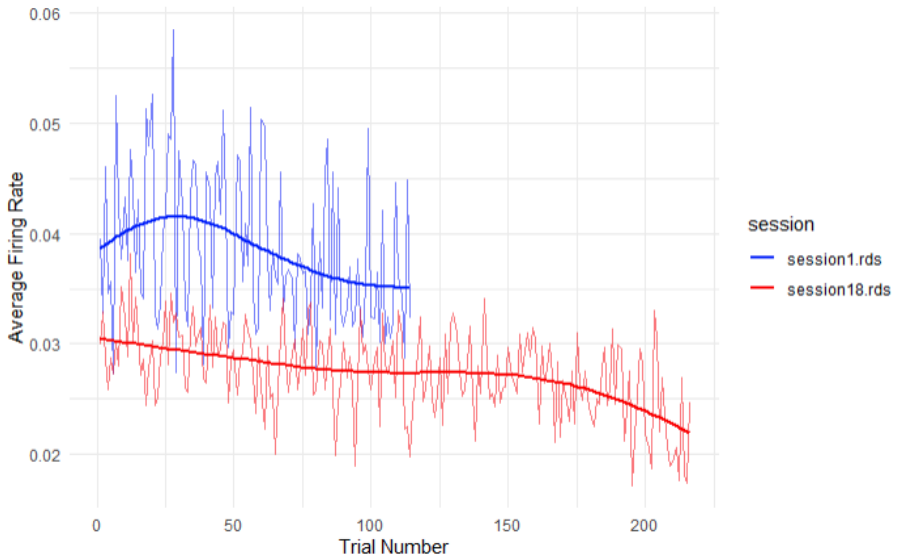
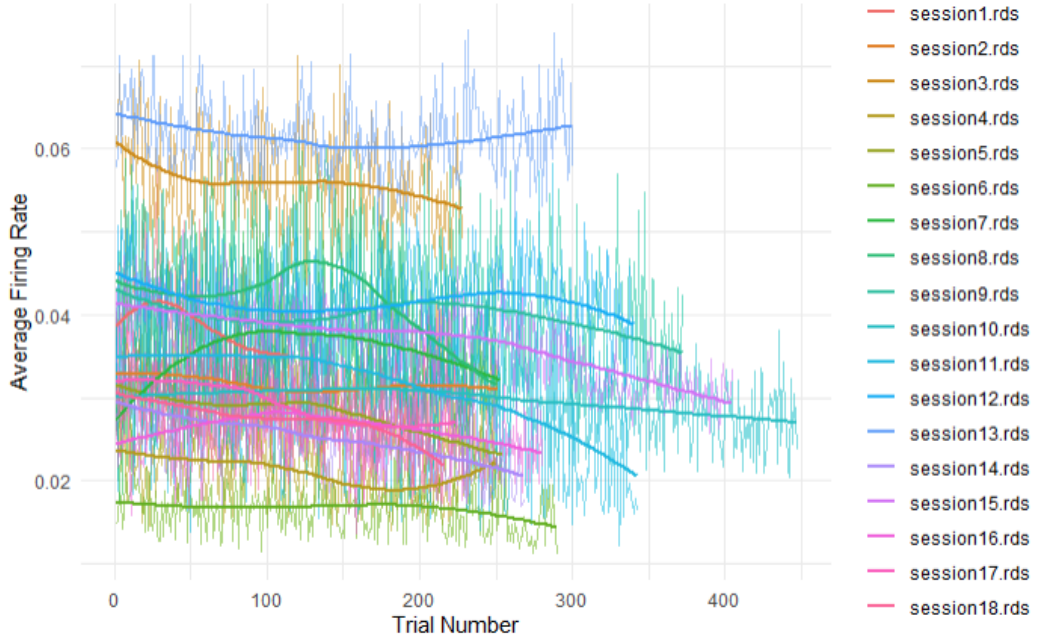


Figure 8: Neural Activity over Trials (All Sessions)



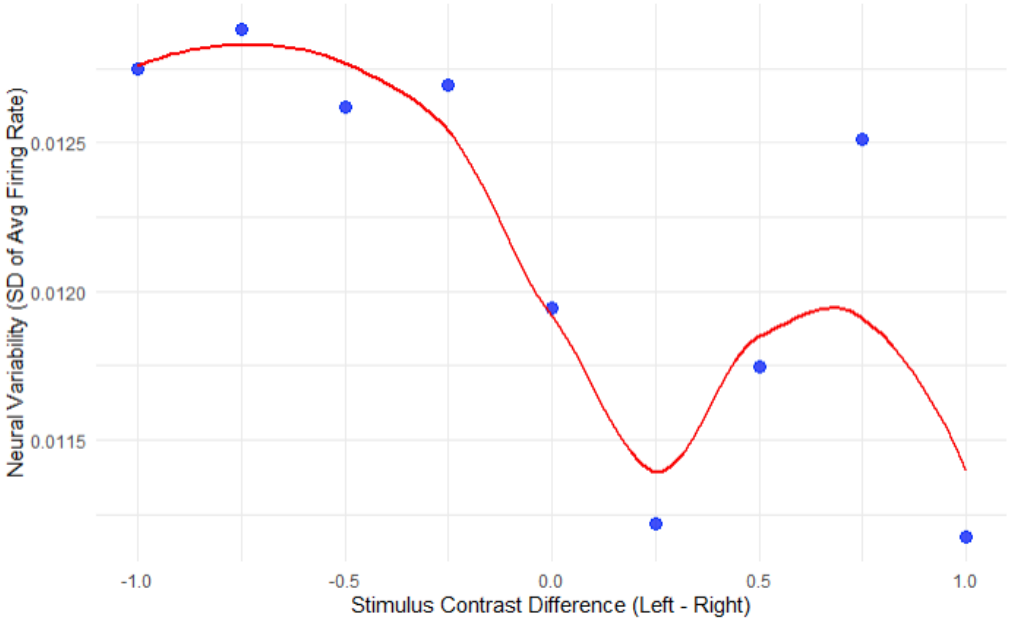
Differences Across Sessions Based on Stimulus Contrast

When looking at these trends, it questions how neural variability relates to stimulus contrast differences. In figure 9, we see a U shaped relationship meaning that neural variability follows a non-linear relationship across different contrast differences. This tells us that neural responses are much more variable when there is a clear difference between the left and right stimuli, and less variable when the stimulus contrasts are similar.

Neural variability reaches its minimum at a stimulus contrast difference of approximately 0.25 to 0.3, which is slightly right of the center. Along with this the curve isn't symmetrical around zero, suggesting that the left side (left stimulus has lower contrast than the right stimulus) shows higher overall variability than the right side.

This pattern suggests that neurons may be the most consistent when there is a moderate difference favoring the right side, as the left side, when contrast differences are larger, shows much more variability. This could indicate a potential bias in the mice's neural system or could reflect how the mice's brain processes different contrast differences.

Figure 9: Neural variability vs Stimulus Contrast Difference



PCA for Neural Activity

When looking at our PC plots and breakdown, we see that PC1 explains 39.02% of the variance, PC2 explains 33.94% of the variance, and PC3 explains 27.04% of the variance. This shows that the distribution of variance across the three components is relatively balanced and another thing to note is PC1 and PC2 together account for over half of the total variance (72.96%).

For PC1 we see two feedback types which are -1 (Incorrect) and 1(Correct). We also see with the confidence ellipses for both feedbacks have significant overlap and that feedback types are distributed across parallel lines. This could mean that feedback type isn't the main determinant for line pattern in this graph. There is also a clear separation between successful (1) and unsuccessful (-1) trials.

For PC2 we see the breakdown of the four mice (Cori, Forssmann, Hench, and Lederberg). There isn't a dominant pattern, or no single mouse dominates any region of the graph which indicates that the mouse being tested doesn't explain the primary structure.

As for PC3, it is another visualization of the same data.

While all the mice show similar overall patterns, there are subtle differences across the lines and graphs which could indicate the variation of neural coding and neurons released in each mouse. Along with that, the neural responses for each mouse are consistently organized by the contrast in stimuli.

Table 2: PCA Breakdown

```
Importance of components:
                            PC1    PC2    PC3
Standard deviation     1.0819 1.0090 0.9007
Proportion of Variance 0.3902 0.3394 0.2704
Cumulative Proportion  0.3902 0.7296 1.0000
```

Figure 10: PCA of Mouse Trial Data (PC1)

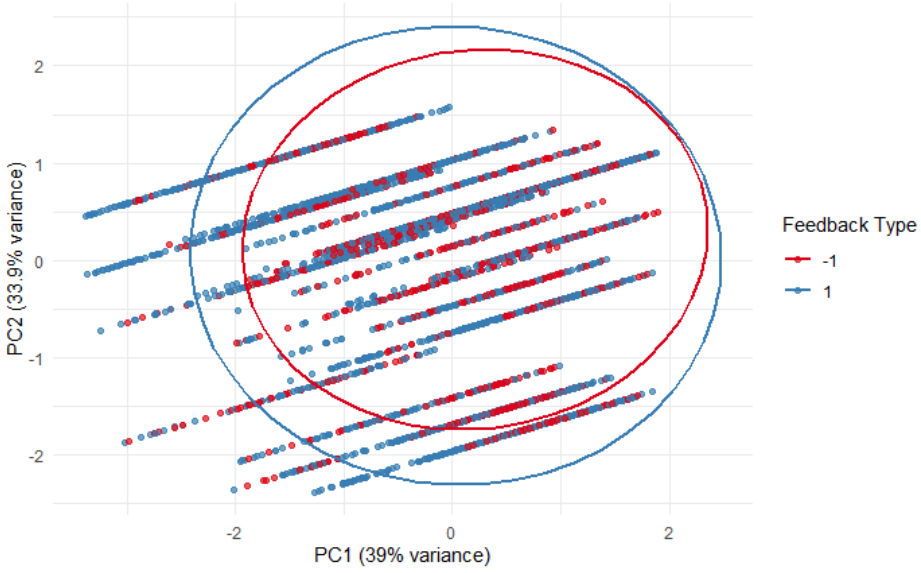


Figure 11: PCA plot by Mouse (PC2)

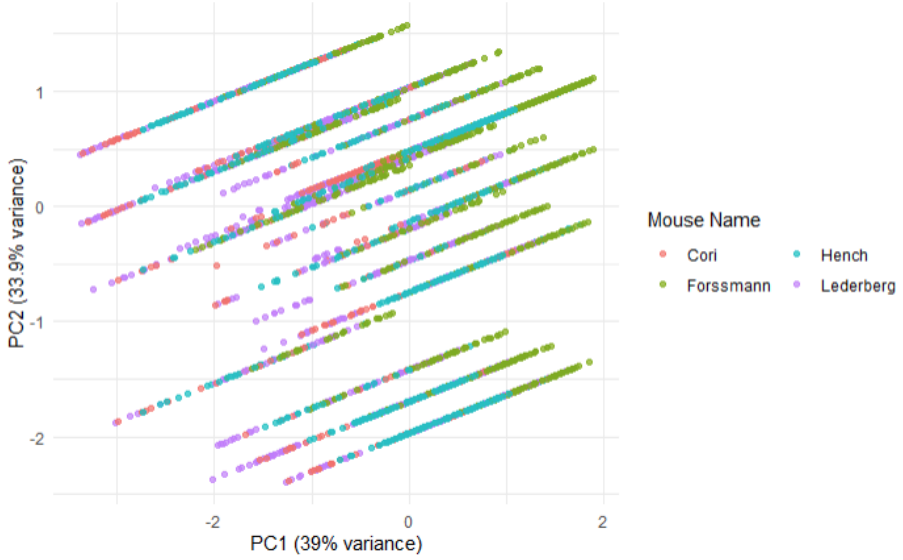
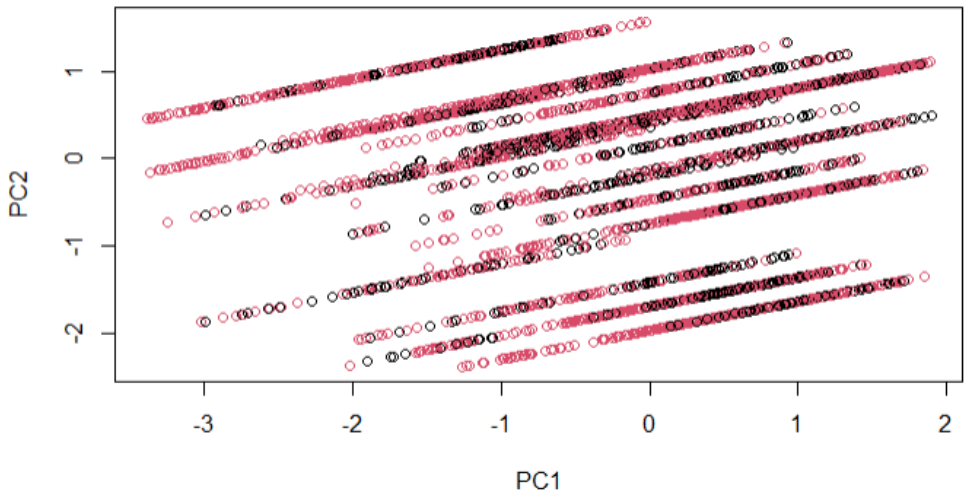


Figure 12: Simple PCA Plot (PC3)



4. Predictive Modeling

Model Performance Analysis

For the training model, I combined the data from the sessions to split it up into two sets: training (80%) and validation (20%). I decided to use a linear regression prediction model.

From the overall confusion matrix, we see that our accuracy is decent at 71.03%. Out of 1015 trials our model predicted 721 correctly. The prevalence being 0.289 indicates that 28.9% of the data is “Incorrect”. The problem or the one flaw of the model is that indeed although the accuracy is high, the model fails to detect incorrect cases.

When modifying the linear regression model, I was able to get it to correctly predict incorrect guesses (it wouldn’t just guess everything as correct), but unfortunately, I wasn’t able to get the accuracy as high as it is with this linear regression prediction model. Along with that, I also tried completely switching from a linear regression model to a random forest model. When doing so, again, my model didn’t just predict everything as correct, but similarly, I was only able to get the accuracy to about a 60%, which is why I chose this linear regression model as the accuracy is indeed higher.

Table 3: Prediction Model Overall Confusion Matrix

```
Confusion Matrix and Statistics

          Reference
Prediction  Incorrect Correct
  Incorrect         0       0
  Correct         294     721

               Accuracy : 0.7103
                 95% CI : (0.6814, 0.7381)
    No Information Rate : 0.7103
    P-Value [Acc > NIR] : 0.5157

                  Kappa : 0

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.0000
            Specificity : 1.0000
         Pos Pred Value :    NaN
         Neg Pred Value : 0.7103
             Prevalence : 0.2897
         Detection Rate : 0.0000
   Detection Prevalence : 0.0000
      Balanced Accuracy : 0.5000

       'Positive' Class : Incorrect
```

5. Prediction Performance on Test Sets

With the official test sets, I evaluated the logistic regression model on test set 1 and test set 2.

For test set 1, the logistic regression model achieved an accuracy of 72%. The confusion matrix shows that the prediction model correctly classified 72 out of 100 trials with 28 false negatives, but zero false positives. The prevalence rate of 0.28 indicates that only 28% of my data belongs to the “Incorrect” class. This shows a significant class imbalance. Along with this we see that the model has a sensitivity of 0 meaning that it fails to identify the “Incorrect” class and the model appears to predict everything as “Correct” with a specificity of 1.

Similarly to test set 1, the logistic regression model achieved a decent accuracy at first glance with 73%. Again though, we see the same problems as the model I chose unfortunately doesn’t seem to predict incorrect instances and only identifies correct ones. This could mean that the neural patterns in both sessions or either one are consistent through the successful trials but not as much for unsuccessful ones.

This is called the accuracy paradox, which is a common thing in imbalance datasets. The model tends to default to the majority class, which in this case is the “Correct” trial. The balanced accuracy for both of the test sets goes to show that the model struggles quite a bit to predict successful and unsuccessful trials equally well.

Table 4 and 5: Prediction Model Based on Test data

Confusion Matrix for Test set 1

```
Confusion Matrix and Statistics

          Reference
Prediction  Incorrect Correct
  Incorrect         0       0
  Correct          28      72

               Accuracy : 0.72
                 95% CI : (0.6213, 0.8052)
    No Information Rate : 0.72
    P-Value [Acc > NIR] : 0.5507

                  Kappa : 0

 Mcnemar's Test P-Value : 3.352e-07

            Sensitivity : 0.00
            Specificity : 1.00
         Pos Pred Value :  NaN
         Neg Pred Value : 0.72
             Prevalence : 0.28
         Detection Rate : 0.00
   Detection Prevalence : 0.00
      Balanced Accuracy : 0.50

       'Positive' Class : Incorrect
```

Confusion Matrix for Test set 2

```
Confusion Matrix and Statistics

          Reference
Prediction  Incorrect Correct
  Incorrect         0       0
  Correct          27      73

               Accuracy : 0.73
                 95% CI : (0.632, 0.8139)
    No Information Rate : 0.73
    P-Value [Acc > NIR] : 0.5516

                  Kappa : 0

 Mcnemar's Test P-Value : 5.624e-07

            Sensitivity : 0.00
            Specificity : 1.00
         Pos Pred Value :  NaN
         Neg Pred Value : 0.73
             Prevalence : 0.27
         Detection Rate : 0.00
   Detection Prevalence : 0.00
      Balanced Accuracy : 0.50

       'Positive' Class : Incorrect
```

6. Discussion

This project aimed to explore how neural activity in 4 mices' brains relate to their decision making based on different visual stimuli. By looking at data from 18 sessions across the four mice, the goal was to try to build a model that successfully predicted whether a mouse will either succeed or fail due to their brain activity patterns that change based on the visual tasks they are presented with.

Conclusion:

The analysis conducted was able to reveal several important findings. We were able to find consistent patterns in brain activity, which in turn linked to successful trials across different sessions and different mice as well. This suggested that the mice do use similar brain processes to make decisions and that neural activity in their brain was dependent on the visual stimuli at hand. The logistic regression prediction model we used had a high accuracy with 72% for test set 1 and 73% for test set 2.

Although I was able to achieve a high accuracy, I wasn't able to optimize the model's performance to accurately predict incorrect trials. The model that I was able to come up

with only predicted correct trials. Future work on this topic could significantly improve the results. One major fix would be to switch the model completely into a random forest model. I had tried doing a random forest model, but unfortunately the accuracy wasn't getting higher than about 60% so I decided to stick to the logistic regression model as the accuracy was higher. Some ways to improve the logistic regression model could be to try using different class weights or adjusting the classification threshold.

For the data integration part, by combining the data from all the sessions, it allowed me to make assumptions based on looking at all the data at once, while taking into account the differences between each session. Along with this, splitting the data up by mice in the beginning of the report allowed me to understand which mice had more neural activity where, which mice had more trials, etc. Looking at mean average firing rates and spike counts allowed me to notice significant patterns in the data.

All in all, this study goes to show that neural activity in the visual cortexes of mice hold much valuable information when looking at their decision making. The model's prediction accuracy still shows that there is a strong correlation between the brain activity of mice and their behavior. These discoveries can help us better understand how neural activity in the brain influenced decision making.

7. Appendix

```
---
title: "STA141 Final Project"
author: "Rohan Pillay"
date: "2025-03-11"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

```{r, echo=FALSE}
library(jsonlite)

setwd("C:/Users/rohan/Downloads/STA141AProject/sessions")
rds_files = list.files(pattern = "*.rds")  #only .rds files

for (file in rds_files) {
  session_data = readRDS(file)
```
```

```
  print(paste("Processing:", file))
  print(str(session_data))

  if (is.list(session_data)) {
    for (name in names(session_data)) {
      element = session_data[[name]]

      if (is.data.frame(element) || is.matrix(element)) {
        df = as.data.frame(element)
        write.csv(df, paste0(sub(".rds", "", file), "_", name, ".csv"), row.names = FALSE)
        write_json(df, paste0(sub(".rds", "", file), "_", name, ".json"), pretty = TRUE)
      }
    }
  }
}
```

```{r, warning=FALSE}
library(tidyverse)
library(ggplot2)
library(reshape2)
library(caret)
library(randomForest)
library(Rtsne)
setwd("C:/Users/rohan/Downloads/STA141AProject/sessions")
rds_files <- list.files(pattern = "*.rds")
all_data <- list()

# Loop through each session file
for (file in rds_files) {
  session_data <- readRDS(file)
  df <- data.frame(
    trial = 1:length(session_data$contrast_left),
    contrast_left = session_data$contrast_left,
    contrast_right = session_data$contrast_right,
    feedback_type = session_data$feedback_type,
    mouse_name = session_data$mouse_name,
    date_exp = session_data$date_exp
)
  avg_firing <- sapply(session_data$spks, function(spike_matrix) mean(spike_matrix))
  df$avg_firing <- avg_firing

  all_data[[file]] <- df
```

```
}
combined_data <- bind_rows(all_data, .id = "session")

# Convert categorical variables to factors
combined_data$feedback_type <- as.factor(combined_data$feedback_type)
combined_data$mouse_name <- as.factor(combined_data$mouse_name)
combined_data$session <- as.factor(combined_data$session)
summary(combined_data)
```

```{r, warning=FALSE}
# Exploratory Analysis code
library(tidyverse)
library(knitr)
library(dplyr)
library(gtools)
setwd("C:/Users/rohan/Downloads/STA141AProject/sessions")

rds_files <- list.files(pattern = "*.rds")


session_summaries <- list()


for (file in rds_files) {
  session_data <- readRDS(file)
  session_name <- sub(".rds", "", file)

  num_trials <- length(session_data$contrast_left)
  num_neurons <- nrow(session_data$spks[[1]])
  brain_areas <- unique(session_data$brain_area)
  num_brain_areas <- length(brain_areas)

  contrast_left_dist <- table(session_data$contrast_left)
  contrast_right_dist <- table(session_data$contrast_right)

  feedback_dist <- table(session_data$feedback_type)
  avg_firing_rates <- sapply(session_data$spks, function(spike_matrix) mean(spike_matrix))
  mean_firing_rate <- mean(avg_firing_rates)
  session_summary <- data.frame(
    session = session_name,
    mouse = session_data$mouse_name,
    num_trials = num_trials,
    num_neurons = num_neurons,
```

```
    num_brain_areas = num_brain_areas,
    mean_firing_rate = mean_firing_rate
  )
  session_summaries[[session_name]] <- session_summary
}

summary_df <- bind_rows(session_summaries)
summary_df <- summary_df %>%
  mutate(session = factor(session, levels = mixedsort(summary_df$session))) %>%
  arrange(session)
# Summary Table
kable(summary_df)
write.csv(summary_df, "session_summary.csv", row.names = FALSE)

# Graph for number of trials per session
ggplot(summary_df, aes(x = session, y = num_trials, fill = mouse)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(x = "Sessions", y = "Trials") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

# Graph for number Neurons per session
ggplot(summary_df, aes(x = session, y = num_neurons, fill = mouse)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(x = "Sessions", y = "Neurons") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

# Heat map for Number of Neurons Recorded Over Certain Trials
ggplot(summary_df, aes(x = factor(num_trials), y = mouse, fill = num_neurons)) +
  geom_tile(color = "white", width = 0.9, height = 0.9) +
  scale_fill_viridis_c(option = "C") +
  theme_minimal() +
  labs(x = "Trials", y = "Mouse", fill = "Neurons") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

trial_spike_df <- data.frame()
for (file in rds_files) {
  session_data <- readRDS(file)
  session_name <- sub(".rds", "", file)

  avg_firing_rates <- sapply(session_data$spks, function(spike_matrix) mean(spike_matrix))

  temp_df <- data.frame(
```

```
    trial = 1:length(avg_firing_rates),
    avg_spike_rate = avg_firing_rates,
    session = session_name,
    mouse = session_data$mouse_name
  )

  trial_spike_df <- bind_rows(trial_spike_df, temp_df)
}

trial_spike_df <- trial_spike_df %>%
  filter(trial <= 18)
# Graph for Mean Spike Rates across Trials
ggplot(trial_spike_df, aes(x = trial, y = avg_spike_rate, color = session, group = session)) +
  geom_line(size = 1.2, alpha = 0.8) +
  geom_point(size = 2) +
  theme_minimal() +
  labs(x = "Trial Number", y = "Mean Spike Rate") +
  scale_color_brewer(palette = "Set2") +
  theme(legend.position = "bottom")

# Graph for mean firing rate per session
ggplot(summary_df, aes(x = session, y = mean_firing_rate, fill = mouse)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(x = "Sessions", y = "Mean Firing Rate") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))


# Data Integration
library(ggplot2)
library(dplyr)
library(scales)
library(tidyverse)
library(gtools)
setwd("C:/Users/rohan/Downloads/STA141AProject/sessions")

rds_files <- list.files(pattern = "*.rds")


session_summaries <- list()
session_summary_df <- data.frame()


for (file in rds_files) {
```

```
  session_data <- readRDS(file)
  session_name <- sub(".rds", "", file)


  avg_feedback <- mean(session_data$feedback_type)


  avg_neurons_recorded <- mean(sapply(session_data$spks, nrow))



  temp_df <- data.frame(
    session = session_name,
    mouse = session_data$mouse_name,
    avg_feedback = avg_feedback,
    avg_neurons_recorded = avg_neurons_recorded
  )

  session_summary_df <- bind_rows(session_summary_df, temp_df)
}


session_summary_df <- session_summary_df %>%
  mutate(session = factor(session, levels = mixedsort(session_summary_df$session))) %>%
  arrange(session)
session_summary_df <- session_summary_df %>%
  mutate(scaled_feedback = rescale(avg_feedback, to = range(avg_neurons_recorded)))

# Shared patterns plot
ggplot(session_summary_df, aes(x = session)) +
  geom_line(aes(y = avg_neurons_recorded, color = "Average Neurons Recorded"), size = 1.2,
group = 1) +
  geom_point(aes(y = avg_neurons_recorded, color = "Average Neurons Recorded"), size = 3) +
  geom_line(aes(y = scaled_feedback, color = "Average Feedback(rescaled)"), size = 1.2, group
= 1, linetype = "dashed") +
  geom_point(aes(y = scaled_feedback, color = "Average Feedback (rescaled)"), size = 3) +
  scale_color_manual(values = c("Average Neurons Recorded" = "blue", "Average
Feedback(rescaled)" = "red")) +
  theme_minimal() +
  labs(x = "Sessions", y = "Average Values", color = "Metric") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))


# Temporal Dynamics Analysis: Line plot - How does the avg firing rate change over the trials?
```

```
# This analysis tracks the neural firing to see if the mice adapt over the trials
# Along with that this graph allows us to visualize the average firing rate change across the trials
# Graph for only session 1 and 18 to see potential changes
library(gtools)
filtered_data <- combined_data %>%
  mutate(session_char = as.character(session)) %>%
  filter(session_char %in% c("session1", "session18")) %>%
  arrange(session_char, trial)




ggplot(filtered_data, aes(x = trial, y = avg_firing, color = session_char)) +
  geom_line(alpha = 0.5) +
  geom_smooth(method = "loess", se = FALSE) +
  theme_minimal() +
  labs(x = "Trial Number", y = "Average Firing Rate", color = "Sessions") +
  scale_color_manual(values = c("blue", "red"))


# Graph for all sessions
combined_data <- combined_data %>%
  mutate(session = factor(session, levels = mixedsort(unique(session)))) %>%
  arrange(session, trial)

ggplot(combined_data, aes(x = trial, y = avg_firing, color = session)) +
  geom_line(alpha = 0.5) +
  geom_smooth(method = "loess", se = FALSE) +
  theme_minimal() +
  labs(x = "Trial Number", y = "Average Firing Rate")

# Neural variability vs stimulus contrast difference graph
combined_data <- combined_data %>%
  mutate(contrast_diff = contrast_left - contrast_right)


variability_df <- combined_data %>%
  group_by(contrast_diff) %>%
  summarise(neural_variability = sd(avg_firing, na.rm = TRUE)) %>%
  arrange(contrast_diff)


ggplot(variability_df, aes(x = contrast_diff, y = neural_variability)) +
  geom_point(color = "blue", size = 3, alpha = 0.7) +
```

```
  geom_smooth(method = "loess", se = FALSE, color = "red") +
  theme_minimal() +
  labs(x = "Stimulus Contrast Difference (Left - Right)", y = "Neural Variability (SD of Avg Firing
Rate)")



# Predictive Modeling
library(dplyr)
library(caret)
library(ggplot2)
library(gtools)


setwd("C:/Users/rohan/Downloads/STA141AProject/sessions")
rds_files <- list.files(pattern = "*.rds")


session_summaries <- list()
session_summary_df <- data.frame()


for (file in rds_files) {
  session_data <- readRDS(file)
  session_name <- sub(".rds", "", file)

  num_trials <- length(session_data$contrast_left)
  avg_firing_rates <- sapply(session_data$spks, function(spike_matrix) mean(spike_matrix))
  mean_firing_rate <- mean(avg_firing_rates, na.rm = TRUE)

  session_summary <- data.frame(
    session = session_name,
    mouse = session_data$mouse_name,
    trial = 1:num_trials,
    contrast_left = session_data$contrast_left,
    contrast_right = session_data$contrast_right,
    feedback_type = session_data$feedback_type,
    avg_firing = avg_firing_rates
  )

  session_summaries[[session_name]] <- session_summary
}
```

```
combined_data <- bind_rows(session_summaries)


clean_data <- combined_data %>%
  filter(!is.na(feedback_type)) %>%
  mutate(feedback_type = factor(feedback_type, levels = c(-1, 1), labels = c("Incorrect",
"Correct")))


model_data <- clean_data %>%
  select(feedback_type, avg_firing, contrast_left, contrast_right)


set.seed(20)
train_index <- createDataPartition(model_data$feedback_type, p = 0.8, list = FALSE)
train_data <- model_data[train_index, ]
validation_data <- model_data[-train_index, ]


logit_model <- glm(feedback_type ~ avg_firing + contrast_left + contrast_right,
                   data = train_data, family = "binomial")


validation_data$predicted <- predict(logit_model, newdata = validation_data, type = "response")
validation_data$predicted_label <- ifelse(validation_data$predicted > 0.5, "Correct", "Incorrect")


conf_matrix <- confusionMatrix(factor(validation_data$predicted_label, levels = c("Incorrect",
"Correct")),
                               validation_data$feedback_type)
print("Overall Confusion Matrix:")
print(conf_matrix)



#Logistic Regression: Can neural activity predict feedback type?
# For Model Performance Analysis
set.seed(20)
train_idx <- createDataPartition(combined_data$feedback_type, p = 0.8, list = FALSE)
train_data <- combined_data[train_idx, ]
test_data <- combined_data[-train_idx, ]
train_data$feedback_type <- ifelse(train_data$feedback_type == -1, 0, 1)
test_data$feedback_type <- ifelse(test_data$feedback_type == -1, 0, 1)
logit_model <- glm(feedback_type ~ avg_firing + contrast_left + contrast_right,
```

```
                data = train_data, family = "binomial")
logit_preds <- predict(logit_model, test_data, type = "response")
test_data$logit_pred <- ifelse(logit_preds > 0.5, 1, -1)
logit_acc <- mean(test_data$logit_pred == test_data$feedback_type)
print(paste("Logistic Regression Accuracy:", logit_acc))


# Prediction performance on the test sets
library(dplyr)
library(caret)
library(ggplot2)

setwd("C:/Users/rohan/Downloads/STA141AProject/sessions")
rds_files <- list.files(pattern = "*.rds")


train_data <- data.frame()


for (file in rds_files) {
  session_data <- readRDS(file)
  session_df <- process_test_data(session_data)
  train_data <- rbind(train_data, session_df)
}



test1_data <- readRDS("C:/Users/rohan/Downloads/STA141AProject/test/test1.rds")
test2_data <- readRDS("C:/Users/rohan/Downloads/STA141AProject/test/test2.rds")


process_test_data <- function(test_data) {
  num_trials <- length(test_data$contrast_left)
  avg_firing_rates <- sapply(test_data$spks, function(spike_matrix) mean(spike_matrix))

  test_df <- data.frame(
    contrast_left = test_data$contrast_left,
    contrast_right = test_data$contrast_right,
    feedback_type = factor(test_data$feedback_type, levels = c(-1, 1), labels = c("Incorrect",
"Correct")),
    avg_firing = avg_firing_rates
  )

  return(test_df)
```

```
}

test1_df <- process_test_data(test1_data)
test2_df <- process_test_data(test2_data)



train_data$weights <- ifelse(train_data$feedback_type == "Incorrect", 2, 1)

logit_model <- glm(feedback_type ~ avg_firing + contrast_left + contrast_right,
                   data = train_data,
                   family = "binomial",
                   weights = train_data$weights)  # Add class weights


test1_df$predicted <- predict(logit_model, newdata = test1_df, type = "response")
test2_df$predicted <- predict(logit_model, newdata = test2_df, type = "response")


threshold <- 0.4

test1_df$predicted_label <- ifelse(test1_df$predicted > threshold, "Correct", "Incorrect")
test2_df$predicted_label <- ifelse(test2_df$predicted > threshold, "Correct", "Incorrect")


conf_matrix_test1 <- confusionMatrix(factor(test1_df$predicted_label, levels = c("Incorrect",
"Correct")),
                                     test1_df$feedback_type)
conf_matrix_test2 <- confusionMatrix(factor(test2_df$predicted_label, levels = c("Incorrect",
"Correct")),
                                     test2_df$feedback_type)

test1_accuracy <- mean(test1_df$predicted_label == test1_df$feedback_type) * 100
test2_accuracy <- mean(test2_df$predicted_label == test2_df$feedback_type) * 100


print("Confusion Matrix for Test Set 1:")
print(conf_matrix_test1)

print("Confusion Matrix for Test Set 2:")
print(conf_matrix_test2)

```
```
```

```{r, warning=FALSE}
# PCA Graphs and values for data integration
library(tidyverse)
library(ggplot2)
library(ggfortify)


setwd("C:/Users/rohan/Downloads/STA141AProject/sessions")
rds_files <- list.files(pattern = "*.rds")
all_data <- list()


for (file in rds_files) {
  session_data <- readRDS(file)


  if(is.list(session_data$spks) && length(session_data$spks) > 0) {
    avg_firing <- sapply(session_data$spks, function(spike_matrix) mean(spike_matrix))
  } else {
    warning(paste("Skipping file", file, "due to invalid spike data"))
    next
  }


  n_trials <- length(session_data$contrast_left)
  df <- data.frame(
    trial = 1:n_trials,
    contrast_left = session_data$contrast_left,
    contrast_right = session_data$contrast_right,
    feedback_type = session_data$feedback_type,
    mouse_name = rep(session_data$mouse_name, n_trials),
    date_exp = rep(as.character(session_data$date_exp), n_trials)
  )


  if(length(avg_firing) == n_trials) {
    df$avg_firing <- avg_firing
    all_data[[file]] <- df
  }
}


if(length(all_data) > 0) {
  combined_data <- bind_rows(all_data, .id = "session")
```

```
  combined_data$feedback_type <- as.factor(combined_data$feedback_type)
  combined_data$mouse_name <- as.factor(combined_data$mouse_name)
  combined_data$session <- as.factor(combined_data$session)


  pca_data <- combined_data %>%
    select(contrast_left, contrast_right, avg_firing) %>%
    drop_na()


  if(nrow(pca_data) > 0) {

    pca_result <- prcomp(pca_data, scale. = TRUE)


    pca_summary <- summary(pca_result)
    print(pca_summary)


    pca_df <- as.data.frame(pca_result$x)
    pca_df$feedback_type <- combined_data$feedback_type[match(rownames(pca_df),
rownames(pca_data))]
    pca_df$mouse_name <- combined_data$mouse_name[match(rownames(pca_df),
rownames(pca_data))]


    if(!all(is.na(pca_df$feedback_type))) {

      pca_plot <- ggplot(pca_df, aes(x = PC1, y = PC2, color = feedback_type)) +
        geom_point(alpha = 0.7) +

        tryCatch(
          stat_ellipse(level = 0.95, size = 1),
          error = function(e) geom_point(alpha = 0.7)
        ) +
        labs(
          x = paste0("PC1 (", round(pca_summary$importance[2,1]*100, 1), "% variance)"),
          y = paste0("PC2 (", round(pca_summary$importance[2,2]*100, 1), "% variance)"),
          color = "Feedback Type"
        ) +
        theme_minimal() +
        scale_color_brewer(palette = "Set1", na.value = "gray")
```

```
    print(pca_plot)

    # PCA plot by mouse
    pca_plot_by_mouse <- ggplot(pca_df, aes(x = PC1, y = PC2, color = mouse_name)) +
      geom_point(alpha = 0.7) +
      labs(
        x = paste0("PC1 (", round(pca_summary$importance[2,1]*100, 1), "% variance)"),
        y = paste0("PC2 (", round(pca_summary$importance[2,2]*100, 1), "% variance)"),
        color = "Mouse Name"
      ) +
      theme_minimal() +
      guides(color = guide_legend(ncol = 2))


    print(pca_plot_by_mouse)

    # Simple PCA Plot
    tryCatch({
      biplot <- autoplot(pca_result,
               data = pca_data,
               colour = combined_data$feedback_type[match(rownames(pca_data),
rownames(combined_data))],
               loadings = TRUE,
               loadings.colour = 'blue',
               loadings.label = TRUE,
               loadings.label.size = 3) +
        theme_minimal() +
        labs(title = "PCA Biplot of Mouse Trial Data")
      print(biplot)
    }, error = function(e) {
      message("Could not create biplot: ", e$message)

      plot(pca_result$x[,1], pca_result$x[,2],
           col = as.numeric(combined_data$feedback_type[match(rownames(pca_data),
rownames(combined_data))]),
           xlab = "PC1", ylab = "PC2")
    })
  } else {
    message("No valid feedback_type data for coloring")
  }
} else {
  message("Not enough data for PCA after removing NA values")
```

```
  }
} else {
  message("No valid data found in the RDS files")
}
```