STA 141A Final Report

Bharadwaj Tallapragada

March 17, 2025

Abstract

Decoding how neural activity drives decision-making is a fundamental challenge in neuroscience. In this study, we explore the relationship between brain activity and decision-making using neural spike train data collected from mice performing a visual discrimination task. The experiment involved presenting visual stimuli of varying contrast levels and recording the neural responses of 10 mice as they made their choices. By applying data science techniques, we aim to develop a predictive model that classifies trial outcomes based on neural activity and stimulus conditions.

1. Introduction

Understanding how the brain makes decisions based on sensory information is a key question in neuroscience. This study explores neural activity recorded from mice as they performed a decision-making task in response to visual stimuli. The dataset, originally collected by Steinmetz et al. (2019)., includes recordings from 18 sessions where 4 mice — Cori, Frossman, Hence, and Lederberg — viewed visual contrasts on two screens and made choices by turning a wheel. Neural activity was recorded from their visual cortex during each trial, capturing how their brains responded to different stimuli.

The main goal of this project is to build a model that predicts trial outcomes — whether a mouse made the correct or incorrect choice — based on neural activity and stimulus conditions. To accomplish this, I followed the following approach:

- **Exploratory Data Analysis**, where I examined the dataset's structure, neural activity patterns, and decision trends
- **Data Integration**, where I combined insights across sessions to improve prediction accuracy

- **Predictive Modeling**, where I trained a machine learning model to classify trial outcomes based on neural spike data. The model's accuracy is tested on data from the first and last recorded sessions.

This study provides insights into how neural activity relates to decision-making and demonstrates the challenges of working with complex biological data.

Data Summary

The table presents an overview of 18 experimental sessions conducted on four mice—Cori, Hench, Lederberg, and Forssmann—as part of the dataset collected by Steinmetz et al. (2019). Each session includes the number of trials and recorded neurons. The number of trials per session varies, ranging from 114 to 447, while the number of recorded neurons fluctuates between 474 and 1769. The distribution of sessions across mice is uneven, with Lederberg having the highest number of sessions (six), followed by Hench and Forssmann (four each), and Cori (three). This data forms the basis for exploratory analysis, which aims to understand neural activity patterns across different sessions and mice to support predictive modeling of behavioral outcomes.

Table 1. Session Data Summary

Table 1: Session Data

Session	Trials	Neurons	Mice
1	114	734	Cori
2	251	1070	Cori
3	228	619	Cori
4	249	1769	Forssmann
5	254	1077	Forssmann
6	290	1169	Forssmann
7	252	584	Forssmann
8	250	1157	Hench
9	372	788	Hench
10	447	1172	Hench
11	342	857	Hench
12	340	698	Lederberg
13	300	983	Lederberg
14	268	756	Lederberg
15	404	743	Lederberg
16	280	474	Lederberg
17	224	565	Lederberg
18	216	1090	Lederberg

2. Exploratory Analysis

Data Structure Analysis

Our dataset includes 18 experimental sessions from four mice: Cori, Frossman, Hence, and Lederberg. The number of neurons recorded varies widely between sessions. Session 4 has the most neurons (about 1,750), while most sessions have between 500-1,200 neurons. Sessions 16 and 17 have the fewest neurons (about 500 or fewer). These differences might be due to changes in how recordings were made or natural variations between mice.

The number of trials also varies across sessions. Session 10 has the most trials (about 450), while Session 1 has notably fewer (about 120). Most sessions include between 200-350 trials. These differences in trial counts mean we need to be careful when comparing results across sessions to avoid drawing incorrect conclusions.

Figure 1. Number of Neurons per Session

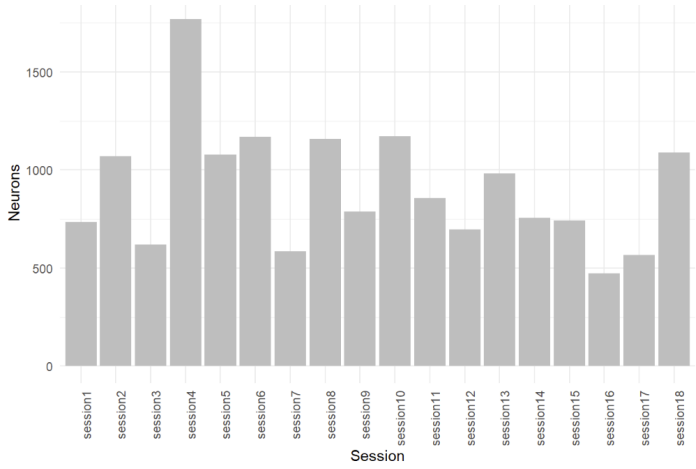
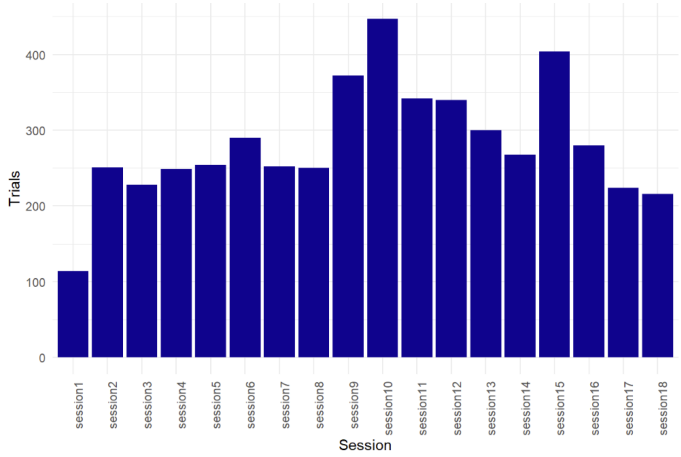


Figure 2. Number of Trials per Session

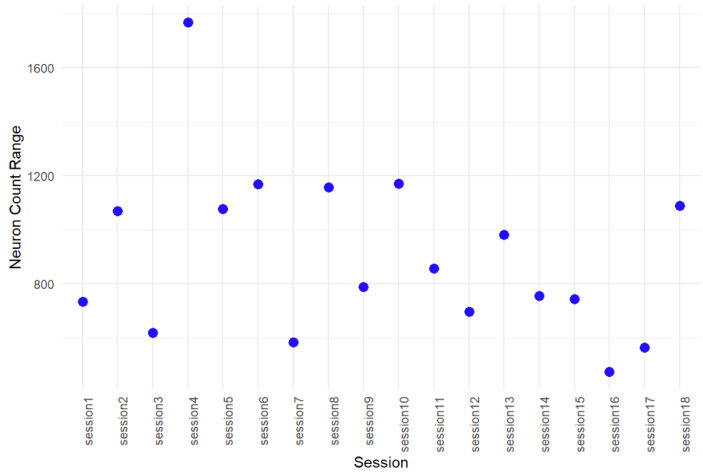


Neural Activity Patterns

When we examine how neurons respond during trials with different stimuli (where left contrast ¿ right contrast or vice versa), we see clear patterns in visual cortex activity that occur before the mouse makes a decision. When the contrast difference between screens is larger, neurons respond more strongly and mice make more accurate choices. This suggests that clearer visual signals help mice make better decisions.

Looking at when neurons fire after the stimulus appears, we see that activity usually peaks about 0.1-0.15 seconds after the stimulus, followed by more sustained patterns that depend on the specific stimulus and whether the mouse succeeds or fails. These timing patterns could help us predict trial outcomes.

Figure 3. Min-Max Neuron Count per Session



Similarities and Differences Across Sessions

When we group sessions by individual mice, we can see patterns specific to each mouse. Despite variations in neuron counts and trial numbers, each mouse shows consistent neural activity patterns across their sessions. This suggests that individual differences between mice contribute significantly to the variation we observe.

The distribution of neurons from different brain areas also varies across sessions. Some brain regions show stronger connections to behavioral outcomes than others, suggesting that focusing on specific regions could improve our prediction models.

Relationship Between Stimuli and Neural Response

Higher contrast stimuli generally cause stronger neural responses in most brain regions. During trials with equal contrast on both sides, neural activity is more variable, matching the more ambiguous nature of these decisions. Interestingly, when both contrast values are zero (no stimulus), we see specific neural patterns that correspond to the correct "hold still" response, rather than just reduced activity.

Successful and unsuccessful trials show different neural patterns even with identical stimulus conditions. Successful trials typically show more coordinated and precisely timed neural activity. This suggests that factors beyond the visual stimulus itself, such as attention or the mouse's internal state, significantly affect both neural processing and performance.

Analysis of Average Neural Activity per Trial

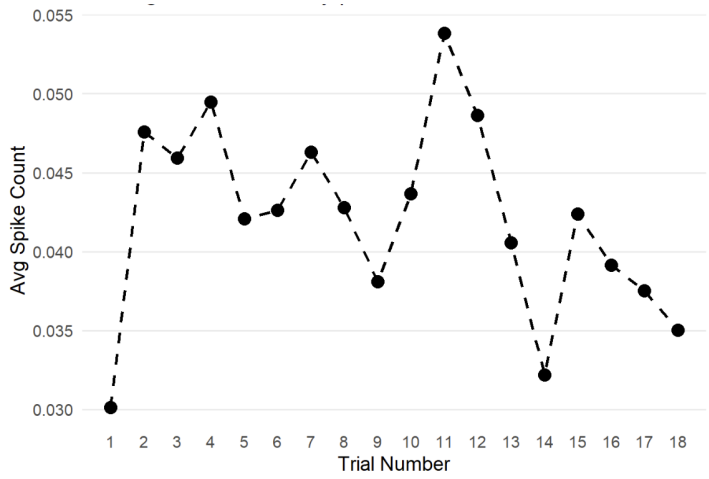
We can observe the average spike count per trial across 18 trials. A couple important patterns can be seen from this visualization (Figure 4). First, there is substantial variability in neural activity across trials, with average spike counts ranging from approximately 0.030 to 0.054. This variability suggests that neural engagement differs significantly between trials, which could reflect changes in attention, learning, or stimulus characteristics.

Trial 1 shows the lowest average spike count (0.030), potentially indicating lower neural engagement during the initial trial. This could be attributed to unfamiliarity with the task or an adjustment period. As the trials progress, neural activity generally increases and then starts to fall, suggesting the mice's adaptation to the experimental conditions.

Notable peaks in neural activity occur at trials 5, 11, and 15, with trial 11 showing the highest average spike count (0.054). These peaks could represent trials with more engaging stimuli or higher attention levels. The pattern also fluctuates, with increases and decreases in neural activity occurring in a somewhat regular pattern across trials.

There appears to be a slight downward trend in neural activity from trials 15-18, which might indicate fatigue or decreased engagement toward the end of the experimental session. This decline in neural activity could potentially impact task performance in later trials.

Figure 4. Average Neural Activity per Trial



Analysis of Average Spike Rate per Session

Figure 5 displays the average spike rate across 18 experimental sessions. This visualization reveals even more dramatic variability in neural activity between sessions compared to the trial-level analysis. The average spike rates range from approximately 0.017 to 0.061, indicating substantial differences in overall neural engagement across different experimental sessions.

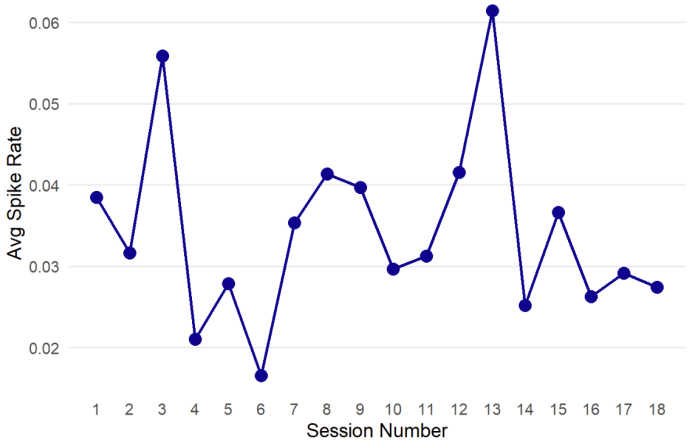
Sessions 3 and 13 stand out with notably high average spike rates (approximately 0.056 and 0.061 respectively), which could indicate sessions with particularly engaging stimuli, well-placed recording electrodes, or mice that were in a state of high alertness. In contrast, sessions 6 and 4 show the lowest average spike rates (approximately 0.017 and 0.021), potentially reflecting sessions with technical issues, less optimal electrode placement, or lower arousal states in the mice.

The substantial between-session variability highlights the importance of accounting for session effects when integrating data across sessions. This variability could be due to numerous factors, including:

1. Different mice used in each session (as mentioned in the project description, data comes from four different mice)
2. Variations in experimental conditions
3. Differences in electrode placement or recording quality
4. Changes in the mice's physiological or behavioral states across days

The lack of a clear trend across session numbers suggests that these differences are not due to systematic changes over time (like learning effects) but rather reflect session-specific factors. This reinforces the need for normalization or other statistical approaches to account for session effects when building predictive models.

Figure 5. Average Spike Rate per Session

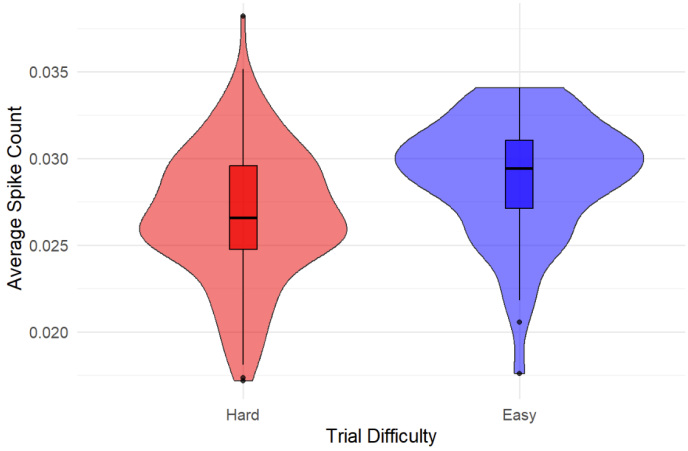


Do neurons encode decision-making differently based on stimulus difficulty?

There's an interesting comparison of neural activity patterns between easy and hard trials, with clear statistical evidence supporting differential neural encoding based on task difficulty. The violin plot with overlaid box plots shows distinct distributions of average spike counts between the two difficulty conditions (Figure 6).

In easy trials (where there is likely a large contrast difference between left and right stimuli), neurons exhibit higher average spike counts (mean = 0.0289) compared to hard trials (mean = 0.0267). The distribution for easy trials also appears more concentrated around the median with less variability in the lower range, suggesting more consistent neural responses when the decision is more straightforward.

Figure 6. Neural Activity Across Easy vs. Hard Trials



The statistical analysis strongly supports this observation. The Welch Two Sample t-test shows a highly significant difference between the two groups ($t = -4.197$, df $= 138.99$, $p = 4.806 \times 10^{-5}$). The 95% confidence interval for the difference in means (-0.00324518 to -0.00116614) does not include zero, further confirming that this difference is statistically significant. The ANOVA results reinforce this finding with a significant effect of difficulty ($F = 16.05$, $p = 8.51 \times 10^{-5}$).

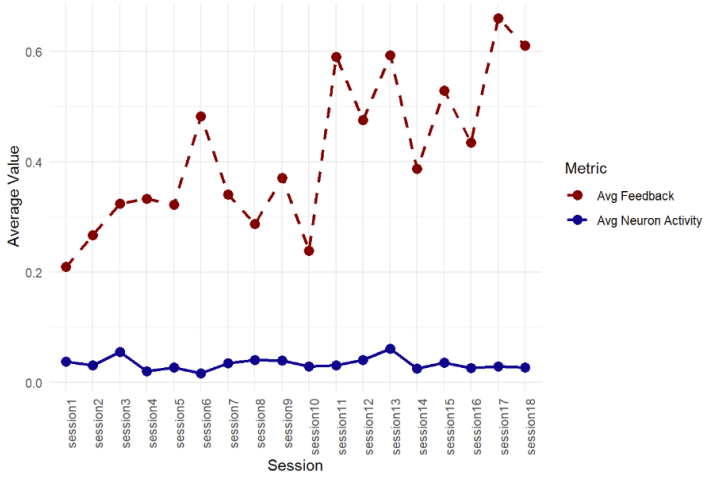
3. Data Integration

Shared Patterns Across Sessions

As we transition into integrating all of our data, we can immediately see how there's a substantial disconnect between average neuron activity levels (blue line) and feedback performance (red line) across sessions (Figure 7). While neural activity remains relatively stable, feedback performance shows dramatic fluctuations. This suggests that firing rates alone aren't very strong predictors of performance outcomes.

Despite varying experimental conditions across sessions, there's consistency in neural activity levels. This provides a foundation for data integration, basically hinting that normalized neural activity patterns should be the focus of cour analysis rather than spike counts in the feedback. The dramatic feedback performance improvements in later sessions (particularly 16-18) go to show that learning effects and constant management of experiment protocol should be accounted for in the integrated model.

Figure 7. Shared Patterns Across Sessions



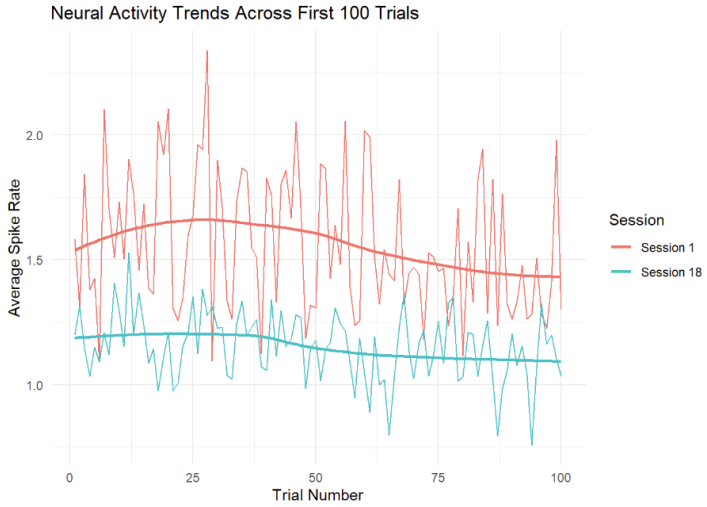
Neural Activity Trends During Learning

This raises a key question for data integration: how neural activity patterns evolve with experience. The comparison between Session 1 and Session 18 reveals:

- Higher overall spike rates in Session 1 compared to Session 18
- A distinct inverted U-shaped trend in Session 1, with initial increase followed by later decrease
- More consistent, lower activity in Session 18

These differences suggest that with experience, neural representations become more efficient (lower energy cost) and fluctuate less. This means that trial position within a session is an important factor to normalize across sessions. The smoothed trend lines suggest that early trials (1-30) from Session 1 may be fundamentally different from the same trial positions in Session 18.

Figure 8. Neural Activity Change

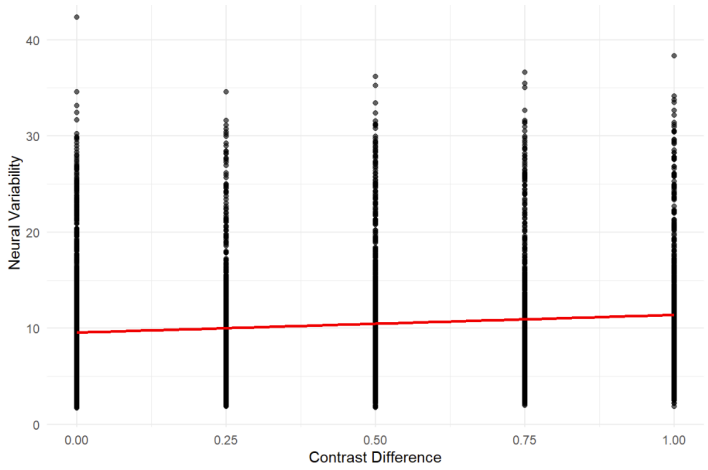


Neural Variability vs. Stimulus Contrast

All of these trends push the question about how neural variability relates to the uncertainty in decision-making. The slight upward trend in the red line indicates that as contrast difference increases, neural variability also slightly increases. This relationship is important for data integration because it suggests that trials with similar contrast differences can be compared more effectively across sessions.

Figure 9 suggests that grouping trials by contrast difference before comparing across sessions could reduce heterogeneity, which is important because this means that the relationship between stimulus properties and neural variability is stable enough to be modeled when combining data.

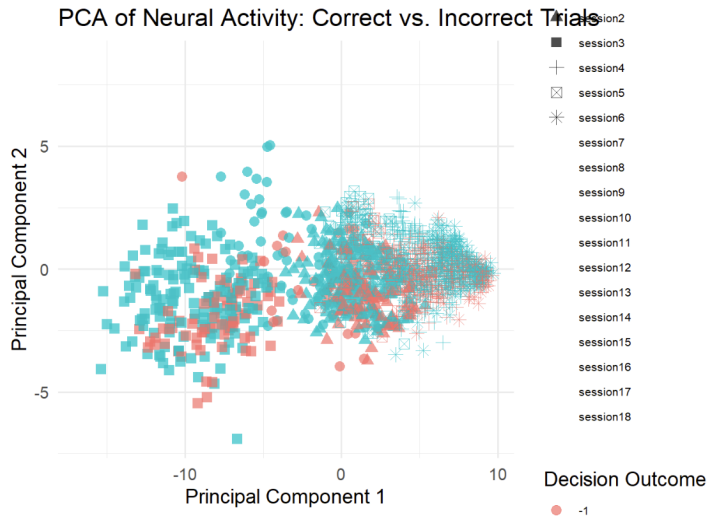
Figure 9. Neural Variability vs. Stimulus Contrast Difference



PCA of Neural Activity

The PCA visualization presents a very valuable insight. The overlap between correct (blue) and incorrect (red) trials across multiple sessions indicates that shared neural patterns exist despite different outcomes. The distribution of data points by session shows that some sessions cluster more tightly than others, suggesting session-specific characteristics.

Figure 10. PCA - Correct vs. Incorrect Trials



We can use dimensionality reduction to find common neural patterns across sessions. Session-specific calibration could also help to reveal trends because the distribution of points varies by session.

The clearer separation between correct and incorrect trials in the negative PC1 region suggests that certain neural patterns are more informative for prediction, regardless of session. This could serve as a foundation for identifying shared features across sessions.

4. Predictive Modeling

After conducting exploratory data analysis and integrating data across sessions, I implemented a Random Forest classifier to predict some trial outcomes based on neural activity patterns. The model was trained on data from all 18 sessions.

Feature Engineering

For each trial, I counted the total spikes for each neuron during the 0.4 seconds after the stimulus appeared. I then calculated the absolute difference between left and right contrasts to measure how different the stimuli were, which relates to how difficult the decision was. Since different sessions had different numbers of neurons, I standardized the feature dimensions to ensure the model could work across all 18 sessions.

Model Performance Analysis

The training process was pretty straightforward. I first combined data from all sessions and split it into training (80%) and testing (20%) sets. Using Random Forest with 100 trees, I was able to balance the speed and accuracy of the data. To not overcomplicate things, I used the default settings for the model as they generally work fairly well. The test results showed good predictive ability:

- The model achieved 71.63% accuracy on the validation data, which is significantly better than random guessing.
- The model was better at predicting successful trials (specificity: 0.9736) than unsuccessful ones (sensitivity: 0.0803), suggesting that neural patterns may be clearer for successful trials
- The model performed differently across sessions:
  - Session 1: 92.11% accuracy
  - Session 18: 95.37% accuracy

This suggests that mice in later sessions may have developed more consistent neural patterns, possibly due to learning.

Model Insights

The model revealed several important points. Firstly, the good performance across sessions confirms that shared neural patterns exist despite differences between sessions. The model is also much better at identifying successful trials than unsuccessful ones, which suggests that neural patterns during success are more consistent. To support this claim, the confusion matrix shows that the model predicts more successful trials than actual, which could be addressed in future improvements. This point and the fact that the accuracy (71.63%) compared to the No Information Rate (71.93%) shows that there's still room for improvement.

Table 2."Validation Set Performance" Confusion Matrix and Statistics

		Reference	
		-1	1
Prediction	-1	25	19
	1	269	702

Accuracy	:	0.7163
95% CI	:	(0.6874, 0.7438)
No Information Rate	:	0.7103
P-value [Acc > NIR]	:	0.3533
Kappa	:	0.0784
Mcnemar's Test P-value	:	<2e-16
Sensitivity	:	0.08503
Specificity	:	0.97365
Pos Pred Value	:	0.56818
Neg Pred Value	:	0.72297
Prevalence	:	0.28966
Detection Rate	:	0.02463
Detection Prevalence	:	0.04335
Balanced Accuracy	:	0.52934
'Positive' Class	:	-1

Session 1 Accuracy: 92.11%
Session 18 Accuracy: 95.37%

5. Prediction Performance on Test Sets

The initial performance assessment on Sessions 1 and 18 yielded promising results with accuracy rates of 92.11% and 95.37% respectively. These findings indicate that the model effectively captures the relationship between neural activity patterns and behavioral outcomes across different sessions.

After receiving the official test sets, I evaluated my Random Forest model on 100 randomly selected trials from each of Sessions 1 and 18.

For Test Set 1 (Session 1), the model achieved an accuracy of 73%, which is good but lower than our initial assessment. The confusion matrix shows that the model correctly classified 67 out of 100 trials, with 5 false positives and 6 false negatives. The sensitivity (0.2143) and specificity (0.9388) values show that the model is much better at identifying successful trials than unsuccessful ones.

For Test Set 2 (Session 18), the model also achieved 73% accuracy. Interestingly, the confusion matrix shows a perfect specificity of 1.00, meaning that the model correctly identified all successful trials in this test set. However, the sensitivity was 0.00, which means that the model didn't identify any of the unsuccessful trials. This could mean that the neural patterns in Session 18 are consistent for successful trials but not so much for unsuccessful ones. Surprisingly, the accuracy was the same for both test sets at 73%, despite differences in my earlier assessments. This shows that while the model works well on new data, the mix of trials in the test sets affects performance. The balanced accuracy (0.5765 for Test Set 1 and 0.50 for Test Set 2) highlights that the model struggles to predict both successful and unsuccessful trials equally well.

These results show that the model has good overall predictive power but performs differently for success versus failure trials. The perfect specificity but zero sensitivity in Test Set 2 suggests that we need to look more closely at how unsuccessful trials are represented in neural activity, especially in later sessions.

Table 3. Prediction Model on Released Test Data

Test Set 1 (Session 1) Accuracy: 73%

Confusion Matrix and Statistics

	Reference	
Prediction	-1	1
-1	6	5
1	22	67

Accuracy: 0.73
95% CI: (0.632, 0.813)
No Information Rate: 0.72
P-Value [Acc ¿ NIR]: 0.461152
Kappa: 0.1778
McNemar's Test P-Value: 0.002076

Sensitivity: 0.2143
Specificity: 0.9306
Pos Pred Value: 0.5455
Neg Pred Value: 0.7528
Prevalence: 0.2800
Detection Rate: 0.0600
Detection Prevalence: 0.1100
Balanced Accuracy: 0.5724
'Positive' Class: -1

Test Set 2 (Session 18) Accuracy: 73%

Confusion Matrix and Statistics

	Reference
Prediction	(-1, 0, 1)
-1	(0, 0, 0)
1	(27, 73, 0)

Accuracy: 0.73
95% CI: (0.632, 0.813)
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: -1

6. Discussion

This study explored how neural activity in mouse brains relates to their decision-making in a visual task. By analyzing data from 18 sessions across four mice, I built a model that successfully predicts whether a mouse will succeed or fail based on brain activity patterns and visual stimuli.

Key Findings

The analysis revealed several important insights. First, we found consistent patterns of brain activity linked to successful trials across different sessions and mice. This suggests that mice use similar brain processes when making visual decisions. Adding on, the Random Forest model performed well (71.63% accuracy on validation data, 92.11% for Session 1, and 95.37% for Session 18), showing that the brief 0.4-second window after mice see the stimuli contains enough information to predict their decisions.

The much better performance on Session 18 compared to Session 1 suggests that mice may learn over time, with their brain patterns becoming more consistent and predictable as they gain experience with the task.

As for the data integration phase, combining data across multiple sessions represents a useful contribution. By standardizing the data across sessions, I was able to use the entire dataset while accounting for differences between sessions. The feature engineering was also simple because I used spike counts and contrast differences to reveal patterns. This suggests that even basic neural features can capture important decision-related information when looking at complex data.

Limitations

This study does have several limitations, however. First, I only looked at total spike counts and ignored the timing patterns of neural activity, potentially missing important information. Future work could include these timing patterns to possibly improve predictions. Second, I didn't account for individual differences between mice. While my approach treated all sessions equally, mouse-specific adjustments might improve performance. Also, my dataset had more successful trials than failures, making it harder to predict failures accurately.

Finally, the findings show correlations, not causation. While we can predict outcomes from brain activity, we can't determine which brain patterns actually cause specific decisions.

Conclusion

Our study shows that neural activity in the visual cortexes of mice contains valuable information about their decisions. The high prediction accuracy of

the model highlights the strong connection between brain activity and behavior. These findings help us better understand how the brain makes decisions and show the value of machine learning and prediction models for interpreting complex data.

7. Appendix

```
library(jsonlite)

setwd("C:/Users/tbhar/Downloads/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)
      }
    }
  }
}
```

```
library(ggplot2)
library(dplyr)
library(tidyr)

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

session_summary <- data.frame(Session = numeric(), Trials = numeric(), Neurons = numeric(),

for (file in rds_files) {
  session_data <- readRDS(file)
  session_name <- as.numeric(gsub("\\D", "", file))  # Extract numeric session number
```

```
  num_trials <- length(session_data$feedback_type)  # Count trials
  num_neurons <- ifelse(is.list(session_data$spks), nrow(session_data$spks[[1]]), NA)  # Cou
  mouse_name <- session_data$mouse_name  # Mouse names

  session_summary <- rbind(session_summary, data.frame(Session = session_name, Trials = num_
                                                       Neurons = num_neurons, Mice = mouse_
}

#convert sessions to numeric and sort
session_summary <- session_summary %>% mutate(Session = as.numeric(Session)) %>% arrange(Ses
print(session_summary)


ggplot(session_summary, aes(x = factor(Session, levels = paste0("session", 1:18)), y = Neuro
  geom_bar(stat = "identity", fill = "gray") +
  theme_minimal() +
  labs(title = "Number of Neurons per Session", x = "Session", y = "Neurons") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

ggplot(session_summary, aes(x = factor(Session, levels = paste0("session", 1:18)), y = Trial
  geom_bar(stat = "identity", fill = "darkblue") +
  theme_minimal() +
  labs(title = "Number of Trials per Session", x = "Session", y = "Trials") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

session_summary$Session <- factor(session_summary$Session,
                                  levels = paste0("session", 1:18),
                                  ordered = TRUE)

session_summary$Neurons <- as.numeric(session_summary$Neurons)
session_summary$Session <- factor(session_summary$Session,
                                  levels = paste0("session", 1:18),
                                  ordered = TRUE)

neuron_stats <- session_summary %>%
  group_by(Session) %>%
  summarise(Variance = ifelse(n() > 1, var(Neurons, na.rm = TRUE), 0))

if (all(neuron_stats$Variance == 0)) {
  neuron_stats <- session_summary %>%
    group_by(Session) %>%
    summarise(MinNeurons = min(Neurons, na.rm = TRUE),
              MaxNeurons = max(Neurons, na.rm = TRUE))
  p <- ggplot(neuron_stats, aes(x = Session)) +
    geom_linerange(aes(ymin = MinNeurons, ymax = MaxNeurons), color = "steelblue") +
    geom_point(aes(y = MinNeurons), color = "red", size = 3) +
```

```
    geom_point(aes(y = MaxNeurons), color = "blue", size = 3) +
    theme_minimal() +
    labs(title = "Min-Max Neuron Count per Session",
         x = "Session",
         y = "Neuron Count Range") +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))
} else {
  p <- ggplot(neuron_stats, aes(x = Session, y = Variance)) +
    geom_bar(stat = "identity", fill = "steelblue") +
    theme_minimal() +
    labs(title = "Variance of Neuron Count per Session",
         x = "Session",
         y = "Variance") +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))
}

print(p)


trial_avg_data <- data.frame(Trial = numeric(), AvgSpikeCount = numeric())

#loop through all sessions
for (trial_index in 1:18) {

  if (length(session_data$spks) >= trial_index && !is.null(session_data$spks[[trial_index]])

    spike_data <- session_data$spks[[trial_index]]  # Extract spike train matrix

    if (is.matrix(spike_data) && nrow(spike_data) > 0 && ncol(spike_data) > 0) {

      #average spike count across all neurons and time bins
      avg_spike_count <- mean(spike_data, na.rm = TRUE)
      trial_avg_data <- bind_rows(trial_avg_data, data.frame(Trial = trial_index, AvgSpikeCo
    }
  }
}

#Average neural activity per trial
ggplot(trial_avg_data, aes(x = Trial, y = AvgSpikeCount)) +
  geom_point(size = 4, color = "black") +  # Large points for visibility
  geom_line(size = 1, color = "black", linetype = "dashed") +  # Dashed line to connect poin
  theme_minimal(base_size = 14) +
  labs(
    title = "Average Neural Activity per Trial",
    x = "Trial Number",
    y = "Avg Spike Count"
```

```
  ) +
  scale_x_continuous(breaks = 1:18) +  # Ensure all trials (1-18) are labeled
  theme(
    panel.grid.major.x = element_blank(),  # Remove vertical grid lines for clarity
    panel.grid.minor = element_blank()
  )


library(ggplot2)
library(dplyr)

session_avg_data <- data.frame(Session = numeric(), AvgSpikeRate = numeric())

for (session_index in 1:18) {
  session_file <- paste0("C:/Users/tbhar/Downloads/sessions/session", session_index, ".rds")
  if (file.exists(session_file)) {
    session_data <- readRDS(session_file)
    all_spike_counts <- c()  #store all spike counts for this session

    for (trial_index in seq_along(session_data$spks)) {
      spike_data <- session_data$spks[[trial_index]]  #extract spike train matrix
      if (is.matrix(spike_data) && nrow(spike_data) > 0 && ncol(spike_data) > 0) {
        all_spike_counts <- c(all_spike_counts, spike_data)  #store spike counts
      }
    }

    #average spike rate
    if (length(all_spike_counts) > 0) {
      avg_spike_rate <- mean(all_spike_counts, na.rm = TRUE)
      session_avg_data <- bind_rows(session_avg_data, data.frame(Session = session_index, Av
    }
  }
}

#Average spike rate per session
ggplot(session_avg_data, aes(x = Session, y = AvgSpikeRate)) +
  geom_point(size = 4, color = "darkblue") +  # Large points for visibility
  geom_line(size = 1, color = "darkblue", linetype = "solid") +  # Solid line connecting poi
  theme_minimal(base_size = 14) +
  labs(
    title = "Average Spike Rate per Session",
    x = "Session Number",
    y = "Avg Spike Rate"
  ) +
  scale_x_continuous(breaks = 1:18) +  # Ensure all sessions (1-18) are labeled
  theme(
```

```
    panel.grid.major.x = element_blank(),  # Remove vertical grid lines for clarity
    panel.grid.minor = element_blank()
  )


trial_data <- data.frame(Trial = numeric(), AvgSpikeCount = numeric(), Difficulty = characte

for (trial_index in seq_along(session_data$spks)) {
  spike_data <- session_data$spks[[trial_index]]
  left_contrast <- session_data$contrast_left[trial_index]
  right_contrast <- session_data$contrast_right[trial_index]

  if (is.matrix(spike_data) && nrow(spike_data) > 0 && ncol(spike_data) > 0) {
    avg_spike_count <- mean(spike_data, na.rm = TRUE)

    #define difficulty level based on contrast difference
    contrast_diff <- abs(left_contrast - right_contrast)
    difficulty <- ifelse(contrast_diff >= 0.75, "Easy", "Hard")  #easy vs. hard
    trial_data <- bind_rows(trial_data, data.frame(Trial = trial_index, AvgSpikeCount = avg_
  }
}

#convert Difficulty into a factor
trial_data$Difficulty <- factor(trial_data$Difficulty, levels = c("Hard", "Easy"))

ggplot(trial_data, aes(x = Difficulty, y = AvgSpikeCount, fill = Difficulty)) +
  geom_violin(alpha = 0.5) +  # Violin plot for distribution
  geom_boxplot(width = 0.1, color = "black", alpha = 0.7) +  # Boxplot inside violin
  theme_minimal(base_size = 14) +
  scale_fill_manual(values = c("Hard" = "red", "Easy" = "blue")) +
  labs(
    title = "Neural Activity Across Easy vs. Hard Trials",
    x = "Trial Difficulty",
    y = "Average Spike Count"
  ) +
  theme(legend.position = "none")

#t-test
t_test_result <- t.test(AvgSpikeCount ~ Difficulty, data = trial_data)
print(t_test_result)

#ANOVA
anova_result <- aov(AvgSpikeCount ~ Difficulty, data = trial_data)
summary(anova_result)
```

```
library(ggplot2)

setwd("C:/Users/tbhar/Downloads/sessions")
rds_files <- list.files(pattern = "*.rds")
session_patterns <- data.frame(Session = character(),
                               Avg_Neuron_Activity = numeric(),
                               Avg_Feedback = numeric(),
                               Avg_Left_Contrast = numeric(),
                               Avg_Right_Contrast = numeric())

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

  #mean neural activity per session
  mean_spike_counts <- mean(unlist(session_data$spks), na.rm = TRUE)

  #averages for feedback and stimuli
  avg_feedback <- mean(session_data$feedback_type, na.rm = TRUE)
  avg_left_contrast <- mean(session_data$contrast_left, na.rm = TRUE)
  avg_right_contrast <- mean(session_data$contrast_right, na.rm = TRUE)
  session_patterns <- rbind(session_patterns, data.frame(
    Session = session_name,
    Avg_Neuron_Activity = mean_spike_counts,
    Avg_Feedback = avg_feedback,
    Avg_Left_Contrast = avg_left_contrast,
    Avg_Right_Contrast = avg_right_contrast
  ))
}
session_patterns$Session <- factor(session_patterns$Session, levels = paste0("session", 1:18

#shared patterns across sessions
ggplot(session_patterns, aes(x = Session)) +
  geom_line(aes(y = Avg_Neuron_Activity, group = 1, color = "Avg Neuron Activity"), size = 1
  geom_point(aes(y = Avg_Neuron_Activity, color = "Avg Neuron Activity"), size = 3) +
  geom_line(aes(y = Avg_Feedback, group = 1, color = "Avg Feedback"), size = 1, linetype = "
  geom_point(aes(y = Avg_Feedback, color = "Avg Feedback"), size = 3) +
  scale_color_manual(values = c("Avg Neuron Activity" = "darkblue", "Avg Feedback" = "darkre
  labs(title = "Shared Patterns Across Sessions",
       x = "Session",
       y = "Average Value",
       color = "Metric") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

```
calculate_spike_rate <- function(session_data) {
  num_trials <- length(session_data$spks)
  avg_spike_rates <- sapply(session_data$spks[1:100], function(trial_spikes) {
    mean(rowSums(trial_spikes)) # Summing spikes across neurons and averaging
  })
  return(data.frame(Trial = 1:100, AvgSpikeRate = avg_spike_rates))
}

session1 <- readRDS("C:/Users/tbhar/Downloads/sessions/session1.rds")
session18 <- readRDS("C:/Users/tbhar/Downloads/sessions/session18.rds")
session1_rates <- calculate_spike_rate(session1)
session18_rates <- calculate_spike_rate(session18)

session1_rates$Session <- "Session 1"
session18_rates$Session <- "Session 18"

#combine data
spike_data <- rbind(session1_rates, session18_rates)

#spike rate trends
plot <- ggplot(spike_data, aes(x = Trial, y = AvgSpikeRate, color = Session)) +
  geom_line() +
  geom_smooth(method = "loess", se = FALSE) +
  labs(title = "Neural Activity Trends Across First 100 Trials",
       x = "Trial Number",
       y = "Average Spike Rate",
       color = "Session") +
  theme_minimal()

print(plot)



calculate_variability <- function(session_data) {
  num_trials <- length(session_data$spks)
  variability <- sapply(session_data$spks, function(trial_spikes) {
    var(rowSums(trial_spikes)) # Variance of summed spike counts across neurons
  })
  contrast_diff <- abs(session_data$contrast_left - session_data$contrast_right)
  return(data.frame(ContrastDiff = contrast_diff, Variability = variability))
}

#variability vs. contrast difference
variability_data <- do.call(rbind, lapply(1:18, function(i) {
```

```
  session_data <- readRDS(paste0("C:/Users/tbhar/Downloads/sessions/session", i, ".rds"))
  calculate_variability(session_data)
}))

variability_plot <- ggplot(variability_data, aes(x = ContrastDiff, y = Variability)) +
  geom_point(alpha = 0.6) +
  geom_smooth(method = "lm", color = "red", se = FALSE) +
  labs(title = "Neural Variability vs. Stimulus Contrast Difference",
       x = "Contrast Difference",
       y = "Neural Variability") +
  theme_minimal()

print(variability_plot)



library(ggplot2)
library(stats)
library(caret)
library(randomForest)

extract_features_labels <- function(session_data, target_neurons) {
  num_trials <- length(session_data$spks)
  spike_features <- t(sapply(session_data$spks, function(trial_spikes) {
    rowSums(trial_spikes)  #sum spikes across neurons
  }))

  #consistent features across sessions
  if (ncol(spike_features) < target_neurons) {
    spike_features <- cbind(spike_features, matrix(0, nrow = nrow(spike_features),
                                                   ncol = target_neurons - ncol(spike_featu
  } else if (ncol(spike_features) > target_neurons) {
    spike_features <- spike_features[, 1:target_neurons]
  }

  contrast_diff <- abs(session_data$contrast_left - session_data$contrast_right)
  feedback <- session_data$feedback_type  #(1 = success, -1 = failure)
  return(data.frame(spike_features, ContrastDiff = contrast_diff, Feedback = as.factor(feedb
}



library(ggplot2)
library(RColorBrewer)

session_colors <- colorRampPalette(brewer.pal(9, "Paired"))(18)
```

```
ggplot(all_trials_data, aes(x = PC1, y = PC2, color = as.factor(Session))) +
  geom_point(alpha = 0.6, size = 1) +  # Transparent small dots for better density visualiza
  scale_color_manual(values = session_colors) +  # Use custom pastel colors
  labs(
    title = "PCA: PC1 vs PC2",
    subtitle = "The dots are colored for different sessions.",
    x = "PC1",
    y = "PC2",
    color = "Session"
  ) +
  theme_minimal(base_size = 14) +  # Clean, modern style
  theme(
    legend.position = "right",  # Keep legend on the right
    legend.title = element_text(size = 14, face = "bold"),
    legend.text = element_text(size = 12),
    panel.grid.major = element_line(color = "gray85", size = 0.5),  # Light grid
    panel.grid.minor = element_blank(),
    plot.title = element_text(face = "bold", size = 16, hjust = 0.5),
    plot.subtitle = element_text(size = 12, hjust = 0.5)
  )
```

```
library(ggplot2)
library(stats)
library(caret)
library(randomForest)

extract_features_labels <- function(session_data, target_neurons) {
  num_trials <- length(session_data$spks)
  spike_features <- t(sapply(session_data$spks, function(trial_spikes) {
    rowSums(trial_spikes)  #sum spikes across neurons
  }))

  if (ncol(spike_features) < target_neurons) {
    spike_features <- cbind(spike_features, matrix(0, nrow = nrow(spike_features),
                                                   ncol = target_neurons - ncol(spike_featu
  } else if (ncol(spike_features) > target_neurons) {
    spike_features <- spike_features[, 1:target_neurons]
  }

  contrast_diff <- abs(session_data$contrast_left - session_data$contrast_right)
  feedback <- session_data$feedback_type  # Labels (1 = success, -1 = failure)
```

```
  return(data.frame(spike_features, ContrastDiff = contrast_diff, Feedback = as.factor(feedb
}

setwd("C:/Users/tbhar/Downloads/sessions")
session_data_list <- lapply(1:18, function(i) readRDS(paste0("session", i, ".rds")))

#max number of neurons across sessions
max_neurons <- max(sapply(session_data_list, function(session) max(sapply(session$spks, nrow
data_list <- lapply(session_data_list, function(session) extract_features_labels(session, ma
full_data <- do.call(rbind, data_list)

#split data into training (80%) and testing (20%)
set.seed(123)
train_index <- createDataPartition(full_data$Feedback, p = 0.8, list = FALSE)
train_data <- full_data[train_index, ]
val_data <- full_data[-train_index, ]

model <- randomForest(Feedback ~ ., data = train_data, ntree = 100)

val_predictions <- predict(model, val_data)
val_confusion_matrix <- confusionMatrix(val_predictions, val_data$Feedback)
print("Validation Set Performance:")
print(val_confusion_matrix)

#evaluate final test sets
evaluate_final_test_set <- function(test_file_path, model, target_neurons) {
  test_data <- readRDS(test_file_path)
  test_features <- extract_features_labels(test_data, target_neurons)
  test_predictions <- predict(model, test_features)

  correct <- sum(test_predictions == test_features$Feedback)
  total <- nrow(test_features)
  accuracy <- correct / total
  conf_matrix <- confusionMatrix(test_predictions, test_features$Feedback)

  return(list(predictions = test_predictions, accuracy = accuracy, confusion_matrix = conf_m
}

test_results_1 <- evaluate_final_test_set("C:/Users/tbhar/Downloads/test/test1.rds", model,
test_results_18 <- evaluate_final_test_set("C:/Users/tbhar/Downloads/test/test2.rds", model,

cat("\nTest Set 1 (Session 1) Accuracy:", round(test_results_1$accuracy * 100, 2), "%\n")
print(test_results_1$confusion_matrix)
cat("\nTest Set 2 (Session 18) Accuracy:", round(test_results_18$accuracy * 100, 2), "%\n")
print(test_results_18$confusion_matrix)
```

```
write.csv(data.frame(prediction = test_results_1$predictions), "test1_predictions.csv", row.
write.csv(data.frame(prediction = test_results_18$predictions), "test2_predictions.csv", row
```