The neural encoding of visual stimuli during different behavioural states

Felipe Martín
Candidate Number: 260774
Supervisor: Dr. Benjamin Evans

September 1, 2023

UNIVERSITY
OF SUSSEX

Abstract

This study investigates how neuronal responses in a large population of neurons encode visual input differently during various behavioral states in mice, specifically during periods of arousal and non-arousal. Two-photon imaging techniques were used to collect neural activity data from hundreds of neurons in the early visual system of the mouse. We developed a Support Vector Machine (SVM) model to predict visual stimuli from neuronal activity evaluating the results in different scenarios. The model's performance was evaluated using cross-validation techniques.

The results indicate that the model shows variable performance in different experiments, with some configurations achieving more than 80% accuracy. Additionally, it was observed that the model's accuracy is better correlated with the product of the number of recordings and the number of neurons per experiment. A t-test analysis revealed a statistically significant difference in accuracy between the 'Arousal' and 'Non-Arousal' groups, favoring the 'Arousal' group.

These findings suggest that the state of arousal could play a significant role in the model's efficacy and open new avenues for future research. The results also support the notion that there are significant differences between the 'Arousal' and 'Non-Arousal' data, and that models trained specifically on one of these sets have difficulty generalizing to the other.

Contents

List of Figures

Chapter 1

Introduction

Visual perception is a complex process involving the integration of sensory and cognitive information in the brain. Several investigations have shown that neuronal activity in the visual cortex is strongly influenced by the behaviour of the individual [1], [2]. In particular, it has been shown that movement can influence the activity of neurons in the early stages of visual processing [3]. However, there are still many unanswered questions about how different behavioural states affect the neural coding of visual stimuli.

In this study, we aim to investigate whether the neural responses of a large neural population in the early visual system encode visual input differently during different behavioural states, specifically when the animal is running versus when it is not running. To achieve this, we will develop and train a machine learning model to predict the visual stimulus from the neural activity.

By addressing this question, we hope to gain a better understanding of how the brain processes visual information during different behavioural states. This research may have implications for understanding the neural basis of perception, how contextual variables affect the representation of sensory information in neural populations, and may also contribute to the development of more accurate machine learning models for predicting visual stimuli from neural activity.

1.1 Objectives

The main goal of the project is to understand whether neural responses of a large neural population encode visual input differently during different behavioural states, specifically during arousal and non-arousal periods.

The project will address the following questions:

(a) Can we predict the visual input from the neural responses more accurately during periods of arousal compared to periods of non-arousal? (b) How accurate is the prediction during periods of arousal if we train the model exclusively with data collected during non-arousal periods? (c) How do models trained with data collected during periods of arousal differ from those trained during periods of non-arousal? To achieve this, neural activity data from hundreds of neurons in the early visual system of the mouse will be collected using two-photon imaging. These data will be gathered while the mouse views moving gratings in different directions. A machine learning model will be developed to predict the visual stimulus based on the activity of the neuronal population. We will then assess whether the model's predictions are more accurate during periods of arousal.

Additionally, we will investigate how models trained with data collected during arousal differ from those trained with data collected during non-arousal periods.

1.1.1 General Objectives:

- To investigate how different behavioural states influence the neural coding of visual stimuli in a large neural population.
- To develop a machine learning model capable of predicting the visual stimulus from the activity of this neural population.
- Evaluate the accuracy of visual stimulus predictions during different behavioural states, specifically during arousal and non-arousal periods.
- To compare and analyse the differences between models trained with data collected during arousal and non-arousal states.

1.1.2 Specific Objectives:

- Preprocess and clean the neural and behavioural data for further analysis.
- Develop a machine learning model using the neural activity and behavioural data to predict the visual stimulus.
- Evaluate the accuracy of the prediction model during different behavioural states, using measures of error and cross-validation.
- Compare trained models with data collected during arousal and non-arousal states, analysing differences in neural coding characteristics and prediction accuracy.
- Interpret the results and discuss the implications of the relationship between behaviour and neural coding of visual stimuli during different behavioural states.
- Identify possible practical applications of the findings in the field of neuroscience and machine learning.

Chapter 2

Literature Review

Recent studies have focused on developing machine learning models to predict visual input from neural activity in both humans and animals [1], [4], [5], [6], [7], [8]. However, it is unclear whether these models perform differently when the subject/animal is in different behavioural states. An interesting aspect of the studies is that the authors suggest that from the prediction model, it is possible to reverse engineer and draw conclusions about the function and organisation of the visual cortex [7], [8].

In [1] focuses on the analysis of neuronal selectivity and decoding of visual information in the mouse brain. They use a multinomial logistic regression model to predict the neuronal response to different visual stimuli during periods of activity and rest. In addition, the study also examines how neural activity during movement affects the accuracy of visual decoding.

In [4] First, they recorded the neuronal activity of thousands of neurons in multiple brain regions of the mice using two-photon imaging. These recordings provided information about the activity of neurons over time during spontaneous behaviours. Then they used principal component analysis (PCA) to reduce the dimensionality of the neural data and then applied a classification algorithm known as support vector machines (SVM) to predict neural activity during spontaneous behaviours in mice.

In [5] the researchers used functional magnetic resonance imaging (fMRI) data to record participants' brain responses while they were presented with visual stimuli. They then used a deep convolutional neural network (CNN) architecture to analyse and extract relevant features from the visual stimuli and predict the corresponding brain responses.

In [6] They used functional magnetic resonance imaging (fMRI) data to record participants' brain responses (superior visual cortex) while they were presented with visual stimuli. The study compared the performance of the hierarchical models developed with other modelling approaches, such as linear models and direct feed-forward neural networks. Different performance metrics were evaluated to determine which model provided the most accurate predictions of neural responses in the superior visual cortex and the results showed that the optimised hierarchical models significantly outperformed other modelling approaches in terms of prediction accuracy of neural responses in the superior visual cortex.

In [7] The researchers used functional magnetic resonance imaging (fMRI) data to record

participants' brain responses while they were presented with visual stimuli. This data was used to train deep convolutional neural networks in a supervised learning approach.

Once the deep neural networks were trained, the learned representations in different layers of the network were analysed to determine how complexity evolved along the ventral stream. Specific visual features were examined and the ability of the deep layers to represent and discriminate different categories of objects was assessed.

The results of the study showed that as visual representations propagated through the deeper layers of the neural network, an increase in the complexity of the neural representations was observed. This suggests that the ventral brain stream follows a gradient of complexity in visual processing and object representation. The authors suggest that these models may be a useful tool for understanding the organisation and function of the visual cortex.

In [8] compares visual object recognition behaviour in humans, monkeys and artificial deep neural networks. The authors find interesting similarities and differences between the three groups, suggesting that neural network models can be a useful tool for studying human visual perception.

Chapter 3

Methodology

3.1 General flow

The general flow (Figure 3.1) starts with data collection, which are stored on a website from where they are downloaded and proceed to the preprocessing and transformation stage. After completing this stage, the data are stored again in a compressed format, and then move on to the stages of data analysis and machine learning.

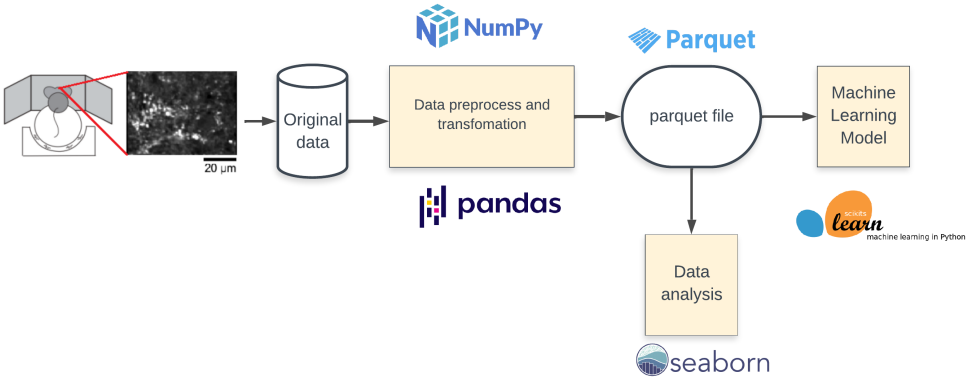

Figure 3.1: General flow

In the preprocessing and transformation stage, raw data undergo a series of steps to clean, structure, and adequately prepare them for analysis and modeling.

These steps include:

Data Cleaning: Missing or inconsistent data that could affect the quality of subsequent analyses are removed.

Transformation and Normalization: Transformations such as changing the structure of data, changing scales, applying mathematical functions, or normalization are carried out to ensure the data are in a coherent and comparable format.

Aggregation and Summarization: Data are aggregated at a higher level to summarize the information and reduce complexity.

Once the data have been preprocessed and transformed, a file in .parquet format will be generated. This format is known for its efficiency in terms of storage and processing, making it suitable for large volumes of data [9].

The next step in the flow is data analysis and the creation of the machine learning model. In this stage, the preprocessed data are used to extract valuable information, patterns, and trends. This may involve the application of various exploratory analysis techniques, visualization, and descriptive statistics to better understand the nature of the data.

The construction of the machine learning model involves selecting the appropriate algorithm, training it with the data, and tuning its parameters to achieve optimal performance. The model is trained in different scenarios: with the entirety of the data, with records in an arousal state, and in a non-arousal state. Each of these scenarios requires special treatment.

Each stage is essential to ensure the quality and validity of the final results, as the accuracy of a model largely depends on the quality of the data and the appropriate processing that is carried out.

3.2 Requirements

For data processing, analysis, visualisation and implementation of the machine learning models we will use the python programming language. Mainly, the libraries Pandas and Numpy for processing and analysis, Matplotlib and Seaborn for data visualisation and Scikit-learn for the development of the machine learning models. Also, other visualisations will be explored with the PowerBI tool.

3.3 Code repository

All python code that is developed will be stored in the following GitHub repository

`https://github.com/FelipeMartin9999/The_neural_encoding_of_visual_stimuli_during_different_behavioural_states`

3.4 Data Collection

The pre-processed data used in this research proposal are available at `https://rdr.ucl.ac.uk/collections/Arousal_modulates_retinal_output/4934931/3` (Schroder et al., 2020);

3.4.1 General procedure for data collection

Measurements of retinal activity were performed in the intact and awake brain using images of synaptic terminals of retinal axons in the superior colliculus (SC). Expression of a calcium indicator (SyGCaMP6f) in synaptic terminals was achieved by specifically targeting retinal ganglion cells (figure 3.2 A and B). For imaging, a two-photon technique was used through an implant that accessed the posterior region of the superficial SC without damaging the brain (Figures 3.2 D and E). The mice were placed on a treadmill to stabilise their movement (Figure 3.2 C). Using this technique, the receptive fields of retinal synaptic terminals could be measured non-invasively in the intact brain.

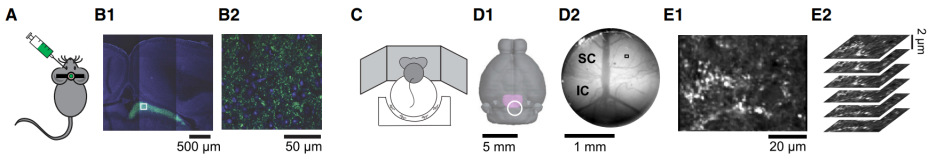

Figure 3.2: Procedure for data collection. Source: Schröder et al. (2020)

(A) The calcium indicator SyGCaMP6f is injected into one eye of the mice used in the experiment. This indicator allows the detection of neuronal activity through changes in calcium levels.

(B) Confocal microscopy images are obtained showing SyGCaMP6f expression at axonal synapses of retinal ganglion cells in the contralateral SC. This allows visualisation of the structures and location of retinal synapses.

(C) Mice are placed in a fixed head configuration on a treadmill surrounded by three monitors. This configuration allows the movement of the mice to be controlled and visual stimuli to be presented in a controlled manner.

(D) An implant is placed in the posterior region of the CS using a brain-friendly technique. The implant allows two-photon imaging of retinal synapses.

(E) Two-photon imaging sessions are performed to capture the activity of retinal synapses in the CS. During the imaging session, multiple planes are acquired with a 2-mm spacing in order to track the synapses even during brain movements.

These steps allow retinal activity data to be obtained in the SC using SyGCaMP6f expression and two-photon imaging in awake and moving mice in a controlled environment.

3.4.2 Experimental setup and visual stimuli

The mouse was fixed on its head by means of a support that did not obstruct its field of vision. For the two-photon technique, the mouse could run freely on a Styrofoam ball suspended in the air, the rotation of which was measured by two optical computer mice. For electrophysiology, the mouse could run freely on a Styrofoam wheel, whose rotation was measured by a rotary encoder.

The mouse was surrounded by three computer screens placed at right angles, covering approximately 270 degrees of visual angle. In some experiments, Fresnel lenses and scattering films were used to compensate for the reduction in luminance and contrast at steeper viewing angles. The eye contralateral to the recording site was illuminated with an infrared LED for eye tracking.

The visual stimuli were presented as moving sinusoidal grids covering all screens and displaying lines at angles between 0 and 330 degrees (Figure 3.3). These gratings had a spatial frequency of 0.08 cycles/deg and a temporal frequency of 2 Hz. Gratings were presented for 2 s separated by a gray screen for 3-6 s. Checkerboard images with white, black and mean gray squares with an edge length of 10 visual degrees (in one dataset 4 visual degrees) were also presented to map receptive fields.

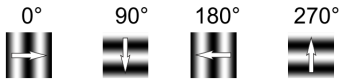

Figure 3.3: Grating angles

Spontaneous activity was recorded both on grey screens and in the dark. For the dark recordings, all screens were turned off and other light sources were eliminated in the experimental setup.

In addition, estimates of photoisomerisation per cone per second (R/cone/s) with changes in pupil size are provided, based on measurements of monitor spread, monitor irradiance, transmission of prior media to photoreceptors, retinal area, cone collection area and mouse pupil size.

3.4.3 Methodological limitations

It is important to consider that the two-photon imaging technique used in this methodology has limitations in terms of spatial and temporal resolution, as well as possible biases introduced by the technique itself. These limitations must be considered when interpreting the results and may influence the accuracy and interpretation of the data obtained.

3.5 Dataset

The dataset consists of 28 experiments, each with a different date. Each of the experiments contains the following information obtained from the process described in the previous section.

Calcium traces and neuron properties

- **dff (delta-F-over-F):** Calcium traces of all ROIs (Region of Interest) for all experiments (stimulus presentations) performed during one imaging session

- **timestamps:** Time [s] stamps for calcium traces
- **planes:** Imaging plane of each ROI
- **ids:** ID of each ROI
- **xyz:** 3D position of ROIs in imaging volume
- **isGad:** 1 if ROI is GAD+, -1 if ROI GAD-, NaN if not known
- **delay:** [s] Temporal delay for each plane relative to plane

Behaviour

- **speed:** Trace of running speed of animal
- **running.timestamps:** Time [s] stamps of running trace
- **diameter:** Pupil size
- **xyPos:** Center position of pupil
- **eye.timestamps:** Time [s] stamps for pupil size and position

Stimuli

- **intervals:** [On- and offset times [s] of gratings
- **gratingID:** ID of grating in each trial
- **directions:** Direction [angles] of movement of grating; NaN if blank was shown
- **sparseNoise.times:** Time [s] of each stimulus frame (map of visual sparse noise pattern)
- **sparseNoiseID:** ID of sparse noise frame
- **sparseNoiseArea.edges:** Edges (left, right, bottom, top) of sparse noise frames within visual field
- **map:** Map of visual sparse noise pattern for each stimulus frame; -1 if black, 1 if white, 0 if gray
- **gratings.intervals:** [s] Start and end of gratings experiment
- **grayScreen.intervals:** [s] Start and end of gray screen presentation
- **sparseNoise.intervals:** [s] Start and end of sparse noise experiment

Results of further analyses

- **largePupil:** It is a matrix of dimension nGratings, nRepetitions. The value is True if the trial was categorized as large pupil trial (pupil is mostly larger than median size)

3.6 Level of arousal

As outlined in the objectives, we aim to investigate how the neuronal encoding of visual stimuli is influenced in a state of arousal versus a non-arousal state. To differentiate between a state of arousal and non-arousal, we use pupil size, as it is one of the physiological measures that has been used to assess the level of "arousal" or activation in an individual [10]. It has been observed that the pupil dilates (increases in size) in response to a variety of emotional, cognitive, and sensory stimuli that generate an increase in arousal levels [11]. Changes in pupil dilation can also indicate reactions to sensory stimuli. For example, the pupil may dilate in response to more attractive or interesting visual stimuli [12]. The change in pupil size is controlled through the autonomic nervous system and, more specifically, via the interaction between the sympathetic and parasympathetic nervous systems [10].

The dataset provides two categories of pupil size for each interval in which a visual stimulus is displayed. The categories are small pupil and large pupil.

3.7 Data pre-process and transformations

In this stage of the process, the original data stored in folders (Figure 3.4) in files with a .npy extension (Figure 3.5) are converted into a relational table structure [13]. This conversion is essential for enabling the efficient and standardized handling of large volumes of data, thereby facilitating more effective manipulation, querying, and analysis

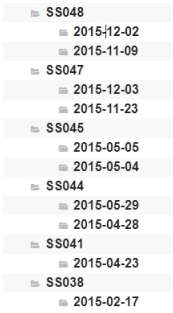

Figure 3.4: Data source folders

The choice of a relational table structure is justified for several reasons. First, this type of structure organizes data into rows and columns, simplifying the representation and understanding of the information. Additionally, relational databases provide robust mechanisms to ensure data integrity, apply constraints, and establish relationships between tables, which is crucial for maintaining data coherence and quality [13][14].

The use of tools and libraries like Seaborn and Scikit-learn is also related to the transformation of data into a relational table structure. Seaborn is a data visualization library built on Matplotlib that offers advanced capabilities for creating visually appealing and informative

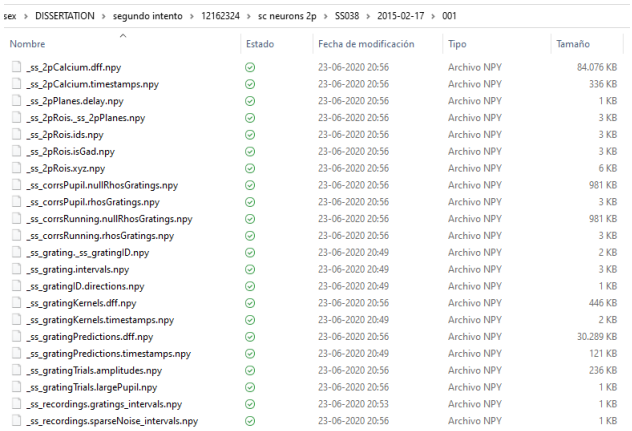

Figure 3.5: Files per folder

statistical visualizations from tabular data [15]. On the other hand, Scikit-learn is one of the most widely used machine learning libraries, offering a broad range of algorithms and tools for data analysis and modeling [16].

Adapting these libraries to work with data in a relational table format allows for more effective utilization of their functionalities. For example, Seaborn can generate graphs and visualizations based on the tabular structure, while Scikit-learn can apply machine learning algorithms to the transformed data. This facilitates the exploration and exploitation of valuable information in the context of data analysis.

To process this data, a python code was created to automate the following steps:

1. That goes through each folder and transforms each file into a pandas dataframe. A column indicating the date of the experiment is added to each dataframe. Finally, all the experiments are concatenated to create a single dataframe per parameter.
2. The delay between different planes is added.
3. Intervals and grating values are associated with the dff values.
4. New categorical parameters are added: running/not running and small pupil/large pupil.
5. An interpolation of the pupil size and speed is performed so that the timings of both align with the times when dff is measured.
6. All parameters are combined to create a single table as a pandas dataframe.
7. The table is exported as a file in .parquet format.

Finally, we perform transformations and join the different parameter data frames to obtain a single data frame with all the necessary columns to be used as inputs in our machine learning models. For example, the dff table must be joined with the time interval table and the gratingsID table to obtain the angle of the grid that was displayed at each time interval with its associated dff.

3.7.1 Data Cleaning

To ensure our dataset's quality and relevance, we undertook a rigorous data cleaning process. Originally, the dataset had 140,627,800 rows, encompassing 23 experiments. However, a large portion of these records lacked associated visual stimuli, making them irrelevant for our analysis. After filtering out these irrelevant entries, the dataset was reduced to a more manageable 18,678,279 rows.

This not only improved the dataset's quality but also optimized the efficiency of our algorithms. The reduced row count led to faster run times, allowing us to perform our analyses more quickly.

In addition to row reduction, we removed extraneous columns that did not contribute to our specific analytical objectives. This simplified the structure of the dataset and further sped up data processing.

We also addressed the issue of missing values in the dataset, particularly in the 'dff'. These were identified and filtered out, ensuring that our subsequent analyses would be based on complete and reliable data.

In summary, our data cleaning process accomplished two key objectives: it eliminated irrelevant and incomplete data, and it optimized the dataset for more efficient and accurate analysis.

3.7.2 Final data structure

Figure 3.6 presents the final arrangement of the data, illustrating an example that consists of 3 experiments, each with 3 neurons. The data are grouped by the experiment to which they belong and then by the corresponding Region of Interest (ROI).

experiment	ROI	dff	timestamps	in_interval	interval_ID	grating_ID	grating_Value	pupil_size	pupil_size_category	speed	speed_category
experiment 1	0	dff 1	time 1	True	1	1	0°	0.3	small	0	no running
experiment 1	0	dff 2	time 2	True	1	2	30°	0.4	small	0	no running
experiment 1	1	dff 3	time 1	True	1	1	0°	0.6	small	0	no running
experiment 1	1	dff 4	time 2	True	1	2	30°	1.2	large	0	no running
experiment 1	2	dff 6	time 1	True	1	1	0°	0.6	small	0	no running
experiment 1	2	dff 7	time 2	True	1	2	30°	1.7	large	2.5	running
experiment 2	0	dff 8	time 1	True	1	1	0°	1.4	large	2.5	running
experiment 2	0	dff 9	time 2	True	1	2	30°	1.2	large	2.5	running
experiment 2	1	dff 10	time 1	True	1	1	0°	2.3	large	3.6	running
experiment 2	1	dff 11	time 2	True	1	2	30°	1.2	large	3.4	running
experiment 2	2	dff 12	time 1	True	1	1	0°	0.3	small	3.5	running
experiment 2	2	dff 13	time 2	True	1	2	30°	1.2	large	3.5	running
experiment 3	0	dff 14	time 1	True	1	1	0°	1.2	large	3.5	running
experiment 3	0	dff 15	time 2	True	1	2	30°	1.2	large	0	no running
experiment 3	1	dff 16	time 1	True	1	1	0°	2.3	large	0	no running
experiment 3	1	dff 17	time 2	True	1	2	30°	0.3	small	0	no running
experiment 3	2	dff 18	time 1	True	1	1	0°	1.2	large	0	no running
experiment 3	2	dff 19	time 2	True	1	2	30°	1.2	large	0	no running

Figure 3.6: Final data structure

The selected parameters to be included as columns in the final table are as follows:

- Experiment: Unique identifier for each experiment. Includes the date of the experiment.
- ROI: Unique identifier for the region of interest or neuron.
- dff: Value of dff.
- Timestamps: Time at which the dff measurement was taken.
- In interval: Indicates whether the measurement is within a visual stimulus interval.
- Interval ID: Unique identifier for the visual stimulus interval.
- Grating ID: Unique identifier for the grating value associated with the interval.
- Grating Value: Value in degrees of the grating associated with the interval.
- Pupil size: Pupil size associated with the measurement.
- Pupil size category: Categorization of pupil size associated with the measurement (large/small).
- Speed: Speed associated with the measurement. Not used in this study but made available for future studies.
- Speed category: Categorization of speed associated with the measurement (no running/running). Not used in this study but made available for future studies.

To work on data analysis and in the machine learning model, this table is loaded from the parquet file into a pandas dataframe (Figure 3.7).

experiment	nROIs	dff	timestamps	in_interval	gratingID	gratingValue	interval_ID	tamaño_pupila	speed	pupil_size_category	speed_category
SS038/2015-02-17/001	0	0.077757	55.569	False	NaN	None	NaN	38.312397	0.000000	Large pupil	No Running
SS038/2015-02-17/001	0	0.326822	55.634	False	NaN	None	NaN	38.137302	0.000000	Large pupil	No Running
SS038/2015-02-17/001	0	0.227577	55.700	False	NaN	None	NaN	37.947975	0.000000	Large pupil	No Running
SS038/2015-02-17/001	0	0.224148	55.766	False	NaN	None	NaN	37.911902	0.000000	Large pupil	No Running
SS038/2015-02-17/001	0	0.575342	55.831	False	NaN	None	NaN	38.021345	0.000000	Large pupil	No Running
...	...	...	...	...	...	...	...	...	...	...	...
SS048/2015-12-02/001	291	0.433730	1201.292	True	9.0	240°	195.0	36.707846	8.241876	Large pupil	Running
SS048/2015-12-02/001	291	0.411959	1201.426	True	9.0	240°	195.0	36.544398	7.935835	Large pupil	Running
SS048/2015-12-02/001	291	0.245105	1201.559	True	9.0	240°	195.0	36.392896	7.632079	Large pupil	Running
SS048/2015-12-02/001	291	-0.087540	1201.693	True	9.0	240°	195.0	36.390179	7.326039	Large pupil	Running
SS048/2015-12-02/001	291	NaN	1201.826	True	9.0	240°	195.0	36.586809	7.022283	Large pupil	Running

Figure 3.7: Pandas dataframe

Chapter 4

Exploratory Data Analysis

4.1 General overview

In this section, we provide an overview of the data through exploratory data analysis. The analysis focuses on each experiment, since each experiment will record different neurons and a different number of them, so they must be treated as separate data sets for Machine Learning classifiers.

Each experiment is linked to a variable number of neurons, and the distribution of these quantities across different experiments is graphically presented in Figure 4.1. We note that experiments 3 and 7 have a higher number of neurons compared to the others, 546 and 508 respectively, while experiments 13, 14, and 15 have the lowest numbers of neurons, 62, 70, and 22 respectively.

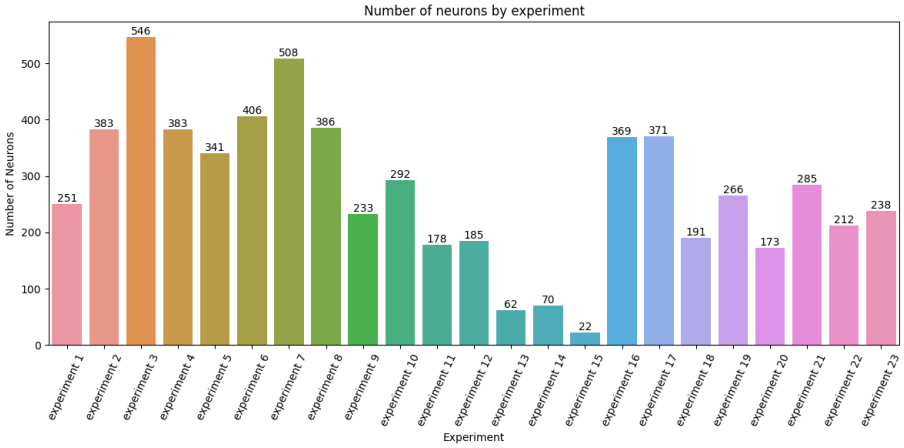

Figure 4.1: Number of neurons by experiment

The number of records per experiment (Figure 4.2) is also variable and depends on the duration of the experiment or the sampling frequency.

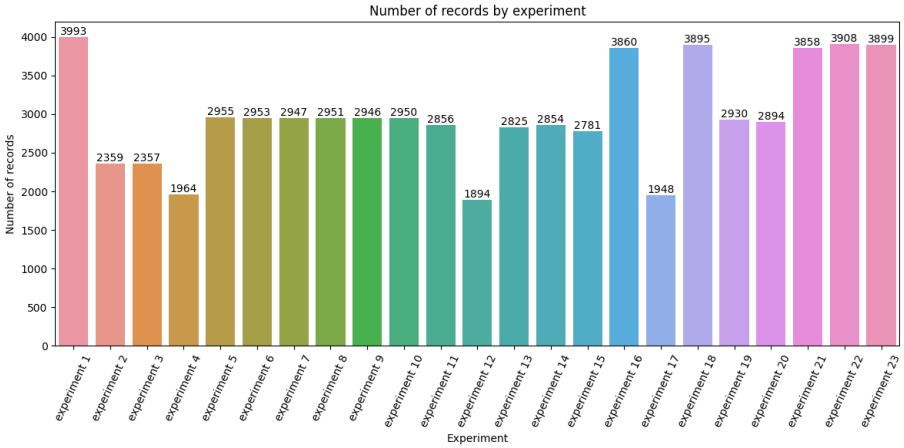

Figure 4.2: Number of records by experiment

Finally, the total number of records for each experiment corresponds to the number of measurements taken during the experiment multiplied by the number of neurons. These are shown in Figure 4.3.

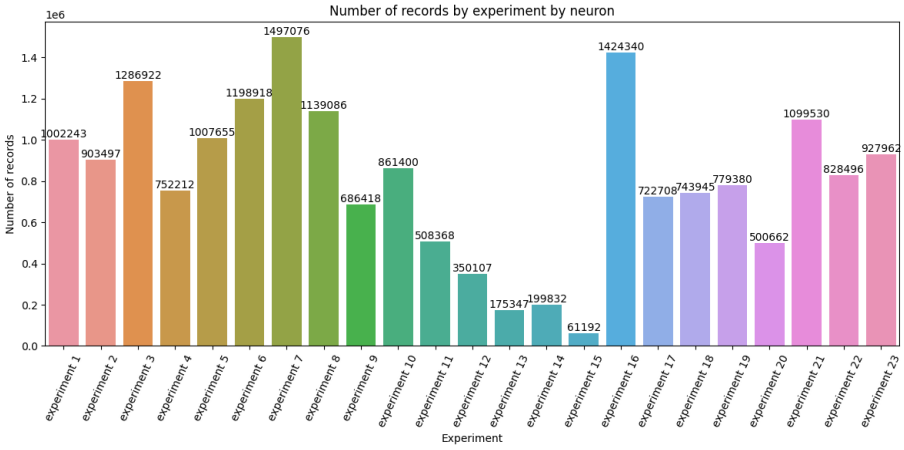

Figure 4.3: Number of records by experiment by neuron

4.2 Grating Intervals

As mentioned in the data collection section, visual stimuli are presented in two-second periods, and each interval has a different grid angle ranging from 0 to 330 in steps of 30°.

The general overview of a single experiment (experiment 1 SS038) is shown next. The figure 4.4 shows the dff values over the time. Each vertical line corresponds to an interval in which the visual stimulus is displayed. Figure 4.5 shows the intervals of the same experiment but only for the angles 0°, 90°, 180°, and 270°.

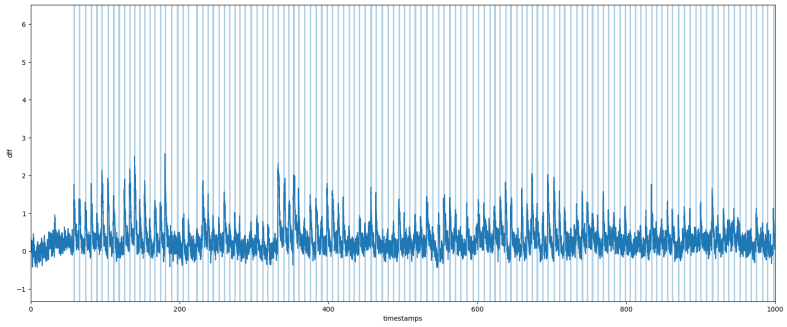

Figure 4.4: Grating intervals over the time for a single experiment

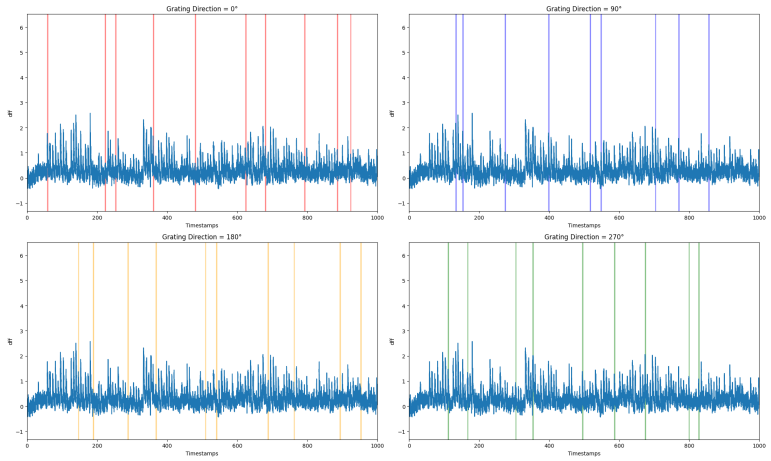

Figure 4.5: Grating intervals over the time for a single experiment, for angles 0°, 90°, 180 and 270°

Analyzing the distribution of the number of intervals for each angle in each experiment (Figure 4.6), we note that there is a symmetrical quantity in each experiment.

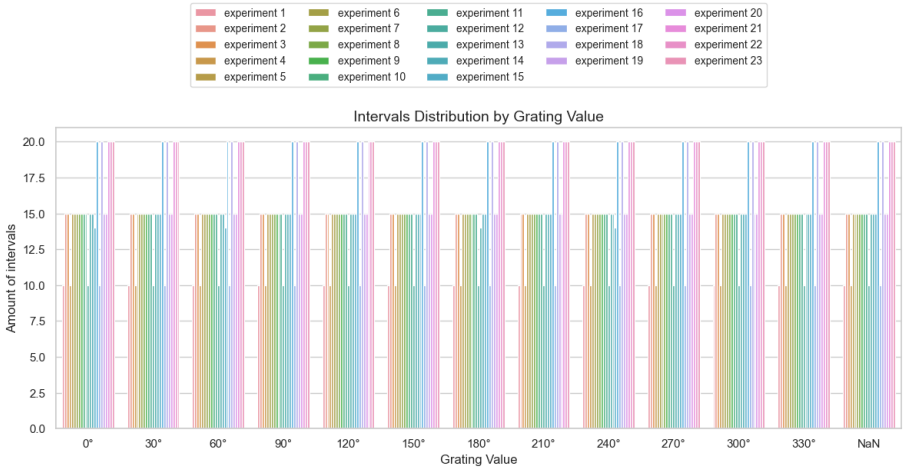

Figure 4.6: Intervals distribution by grating value

Then, within each 2-second interval, we have the dff measurements that we are interested in using for the machine learning model. If we perform a count of the number of measurements within intervals that exist for each angle, we obtain the following graph:

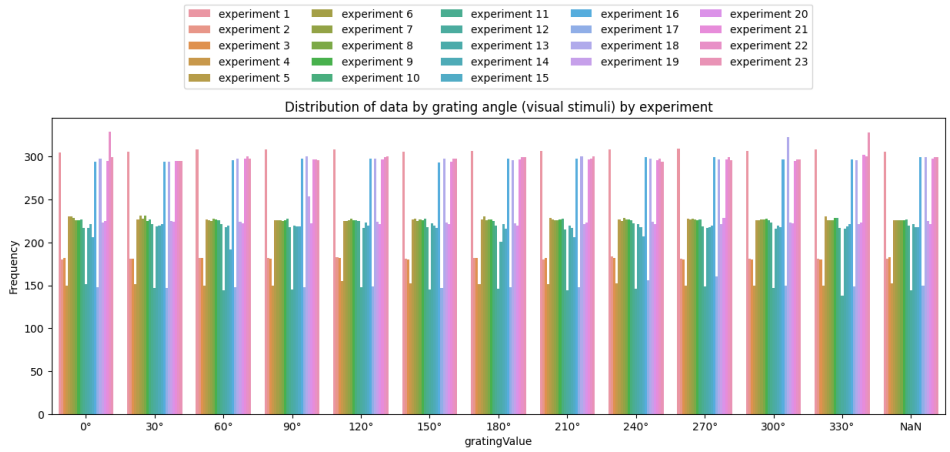

Figure 4.7: Distribution of data by grating angle by experiment

The graph indicates a symmetrical distribution of measurements across different angles within each experiment. This symmetry suggests that when constructing a comprehensive model using the entire dataset, class balancing techniques will not be required.

4.3 Data variability

In this section, we will explore the variability present in the dff values (relative change in calcium fluorescence) in response to different grating angles, as well as the variability between experiments. Analyzing this variability is crucial for understanding the robustness of the observed patterns and assessing the consistency of neuronal responses.

4.3.1 Inter-Experiment Variability and Considerations

Neuronal activity data from different experiments display significant variability, both between experiments and within individual ones, particularly in relation to the arousal state as measured by pupil size. It is observed that neuronal activity tends to be higher during states of large pupil size, suggesting a potential correlation with arousal levels. Furthermore, intra-experimental variability is substantial, as reflected in metrics such as the standard deviation and the interquartile range, indicating differences in the consistency of neuronal activity across varying states of arousal.

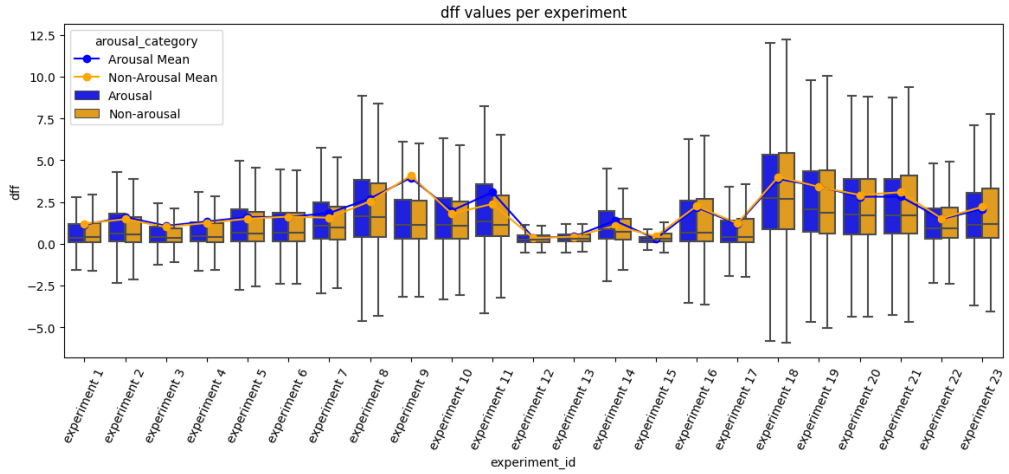

Figure 4.8: Distribution of dff values by experiment and arousal level

4.3.2 Variability in Neuronal Response to Different Angles

Boxplots were used to visualize the distribution of dff values corresponding to each grating angle (Figure 4.9). This graphical representation revealed variability in neuronal responses to different visual stimuli. Peaks and differences in the amplitude of distributions were observed based on specific grating angles, suggesting a neuronal selectivity toward specific visual stimuli. These findings are consistent with existing literature on neuronal selectivity in visual processing in the mouse brain.

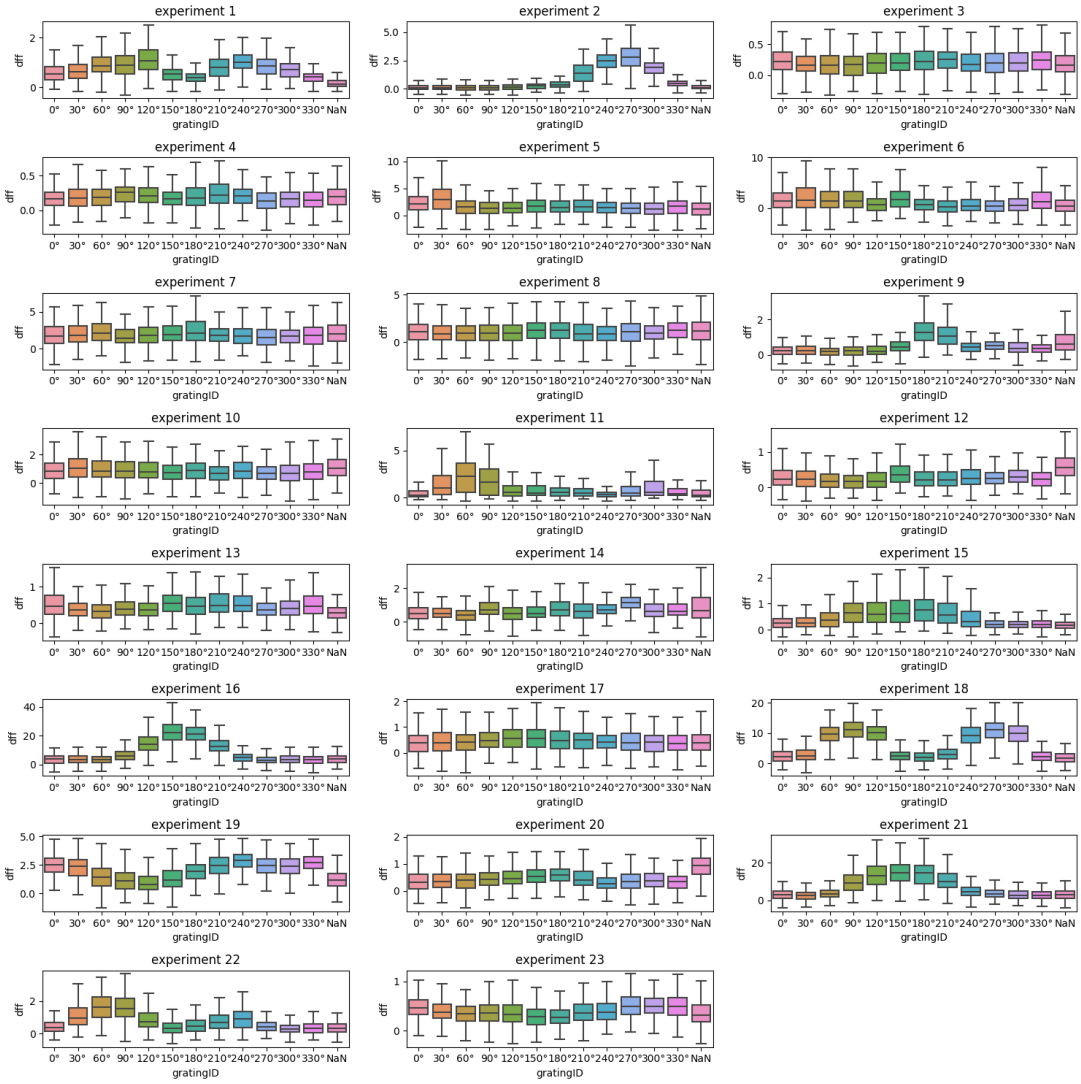

Figure 4.9: Distribution of dff values by grating values by experiment

4.4 Neural behavior

Each neuron exhibits a unique response to visual stimuli, which is reflected in the variability of neuronal activity spikes. It is notable that certain neurons show more intense responses when presented with stimuli at specific angles, compared to other angles. This phenomenon may be linked with neuronal selectivity and information encoding in the brain [17].

To analyze this in more detail, we have generated the "directional tuning curve" for each neuron, which is obtained from the Mean dff per grid at 2-second intervals (Figure 4.10 A). With this, we obtain a general directional tuning curve (Figure 4.10 B.1) and categorize it by pupil size (Figure 4.10 B.2) and speed (Figure 4.10 B.2). Additionally, another effective way to visualize these data is through polar plots (Figure 4.10 C).

Although at first glance each neuron appears to have unique behavior, a recurring pattern can be identified in relation to the angles that generate greater neuronal activity. We observe that there is a higher number of activity spikes at these specific angles across various experiments.

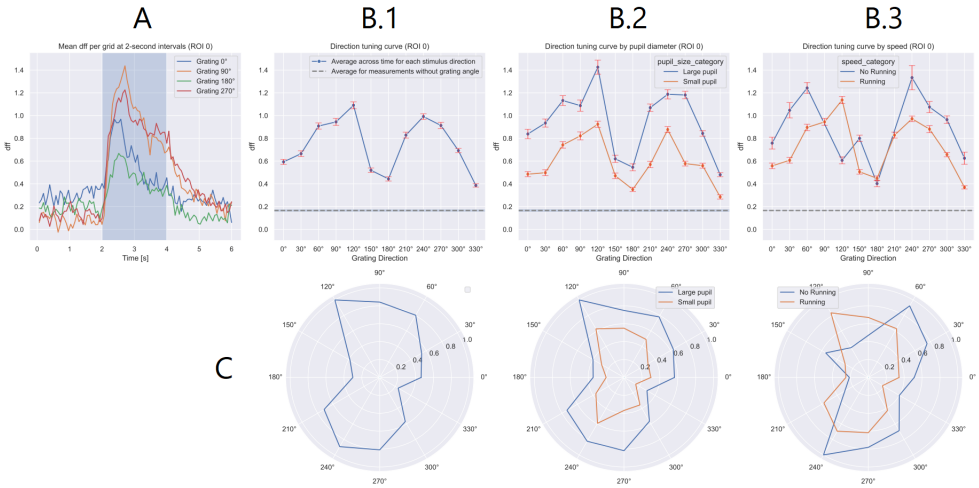

Figure 4.10: Direction tuning curve for a single neuron

Chapter 5

Machine Learning Model

This chapter delves into the application of Support Vector Machines (SVM) as a robust method for handling high-dimensional data in neuronal decoding tasks. Specifically, we will focus on using SVMs to classify different orientations of visual stimuli based on neural recordings measured as differences in firing frequency (DFF) in neurons. In addition, the different stages of the model are detailed, such as: standardization of functions, adjustment of hyperparameters, and unbalanced classes, offering a complete guide to implementing and interpreting the SVM model.

5.1 Features and Classes

5.1.1 Features

In this particular context, each neuron's activity—measured as the difference in firing frequency (DFF)—serves as a distinct feature for our machine learning model. For example, if we are conducting an experiment involving 250 neurons, our feature set would comprise 250 dimensions, transforming the problem into a high-dimensional space. This high dimensionality is a challenge but also an opportunity to capture complex patterns that a less rich dataset might miss.

5.1.2 Classes

The classes for our classification task correspond to the various grating values, which represent different orientations of visual stimuli. Specifically, grating values range from 0° to 330°, moving in steps of 30°.

5.2 Support Vector Machine

Support Vector Machines (SVM) are a set of supervised machine learning techniques primarily used for classification and regression problems. Their popularity and effectiveness lie in their ability to transform original datasets into higher dimensions and find an optimal hyperplane that maximizes the margin between classes or allows for a more accurate fit to the data [18].

The fundamental principle behind SVM is to find the hyperplane that best separates the input data classes. In a two-dimensional space, this hyperplane is simply a line; in three dimensions, it is a plane; and in higher dimensions, it is called a hyperplane. The "support vectors" are those data points that are closest to the hyperplane and, therefore, are the most critical in determining the position and orientation of the hyperplane. The "margin" is the distance between the hyperplane and the nearest support vector. SVM aims to maximize this margin to ensure robust separation and minimize classification error [19].

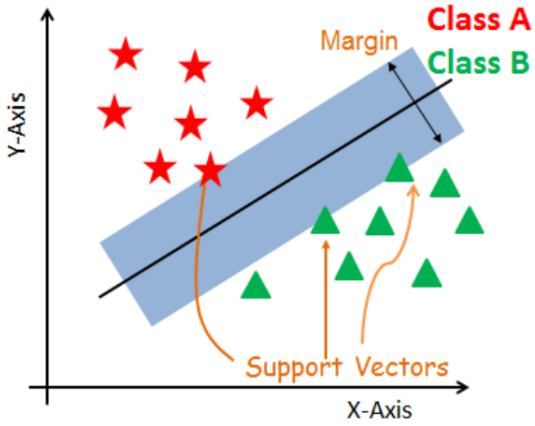

Figure 5.1: Example of support vector machine

One of the highlights of Support Vector Machines (SVM) is the kernel trick, allowing data that are not linearly separable in their original space to be transformed into a higher-dimensional space where they can be [20]. There are various types of kernels, including:

- **Linear:** This kernel is simply the dot product of two vectors and is used when the data are linearly separable or nearly so.
- **Polynomial:** Allows for polynomial classifications and can adapt to more complex shapes.
- **Radial Basis Function (RBF) or Gaussian:** One of the most commonly used, this kernel can handle cases where the separation between classes is highly complex.
- **Sigmoidal:** Shares similar characteristics with the sigmoid function in neural networks.

In the past decade, Support Vector Machines (SVM) have gained traction in neuroscience, especially in tasks related to neuronal decoding. Studies have shown that SVMs can effectively

predict patterns of neuronal activity associated with different stimuli or actions, even in scenarios where other models fall short. For instance, research has employed SVMs to decode visual images, movement patterns, and other stimuli from neuronal recordings [21].

The advantages of using SVM in neuronal analysis include [22]:

- **Handling High-Dimensional Data:** Since neuronal recordings can involve the activity of numerous neurons, the resulting data can be high-dimensional, as each neuron corresponds to a feature. SVM is well-equipped to handle such data.
- **Flexibility:** Thanks to the kernel trick, SVMs can adapt to various patterns of neuronal activity, from linear to complex non-linear relationships.
- **Robustness Against Overfitting:** Given SVM's margin-maximizing nature, they tend to be robust against overfitting, especially when dealing with a limited amount of data, a common situation in neuroscience studies.

5.3 Feature Standardization

Before training the SVM model, it is crucial to ensure that all input features are on a comparable scale. For this reason, feature standardization is employed. This process involves subtracting the mean and dividing by the standard deviation for each feature, resulting in a distribution with a zero mean and a standard deviation of one. We utilized the fit_transform method of the StandardScaler class from the scikit-learn library to implement this standardization step.

SVMs are notably sensitive to the scale of input data. The algorithm aims to find an optimal hyperplane in an n-dimensional space to best separate the classes. The orientation and position of this hyperplane are determined based on the margins between the different classes and their support vectors. If the features are not scaled, those with higher magnitudes will disproportionately influence the orientation of the hyperplane, leading to a suboptimal classification.

In the context of neuroscience, particularly when applying SVMs to predict visual stimuli based on differences in firing frequency (DFF) in neurons, the amplitude of these signals and their temporal patterns can vary significantly across neurons and stimuli. Feature standardization ensures that all neurons and stimuli contribute equally to the model. This uniform contribution prevents any individual neurons or stimuli from dominating the model simply due to their inherent magnitudes.

5.4 Hyper-parameter Tuning

To fine-tune the model's hyperparameters, we employ the grid search technique using the GridSearchCV class from the scikit-learn library [23]. This approach involves specifying a range of values for each hyperparameter and systematically evaluating the model's performance for each combination of these values.

The hyperparameters that can be modified in this code to obtain a better result are [24]:

- **Kernel:** Refers to the function used to transform the data into a higher dimensional space. The most common kernels are linear, polynomial and radial basis kernels (RBF). [25]
- **C:** It controls the trade-off between maximising the margin of the decision function and minimising the number of misclassified points in the training set. [25]
- **Gamma:** It is used in the RBF kernel to define the size of the area of influence of a training point. Large values of gamma can cause overfitting of the model, while small values can cause underfitting. [25]

In order to find the best combination of values, we create the following matrix and train the model with each possible combination:

```
param_grid = {
    'kernel': ['linear', 'rbf', 'poly'],
    'C': [0.1, 1, 10, 100],
    'gamma': ['auto', 'scale', 0.1, 1, 10]
}
```

Figure 5.2: Hyperparameter combination

The following chart displays the accuracy scores for the optimal combination of hyperparameters, obtained using the grid search method, for each experiment.

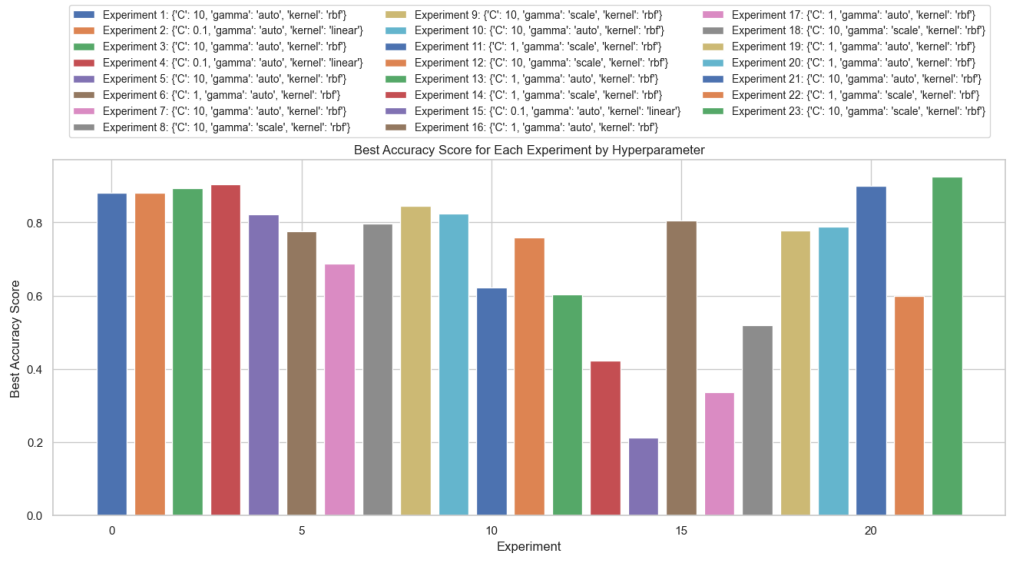

Figure 5.3: Best accuracy for each experiment by hyper-parameter

5.5 Handling Imbalance

When dealing with imbalanced datasets, such as our data contrasting large-pupil and small-pupil states, there are two general approaches to address the imbalance:

- **Undersampling:** This method involves randomly removing samples from the majority class until the category distribution becomes balanced. However, this approach can be problematic as important information might be lost due to data removal [26]. In our case, reducing the amount of small-pupil data could result in a model that is neither robust nor representative of the actual state of affairs.
- **Oversampling:** Instead of eliminating data, oversampling involves adding more samples to the minority class. In our study, this would mean adding more instances with the large-pupil filter. While one could argue that adding repeated data (simple oversampling) could lead to overfitting [27], more advanced techniques like SMOTE offer a solution to this issue.

5.5.1 Class Imbalance

When we partition the data into a subset corresponding to arousal states (large pupil) and a subset without arousal (small pupil), the classes (grating values) become imbalanced, as illustrated in Figure 5.4 B.

Oversampling and the SMOTE technique

To address the issue of class imbalance within each subset, we employ the SMOTE (Synthetic Minority Over-sampling Technique) technique. SMOTE is an oversampling method that generates synthetic samples for the minority class rather than merely duplicating existing instances [28]. This is achieved by identifying two or more similar samples and generating a new sample that is a linear combination of these neighboring samples. In this manner, SMOTE creates additional data that, while not identical to the original samples, are consistent with the underlying data space.

We chose SMOTE over simple oversampling to mitigate the class imbalance problem because it helps us avoid the risk of overfitting that can arise from simply replicating the same samples. Additionally, SMOTE is preferable to undersampling in this context, as we do not wish to lose the valuable information contained within the small-pupil samples.

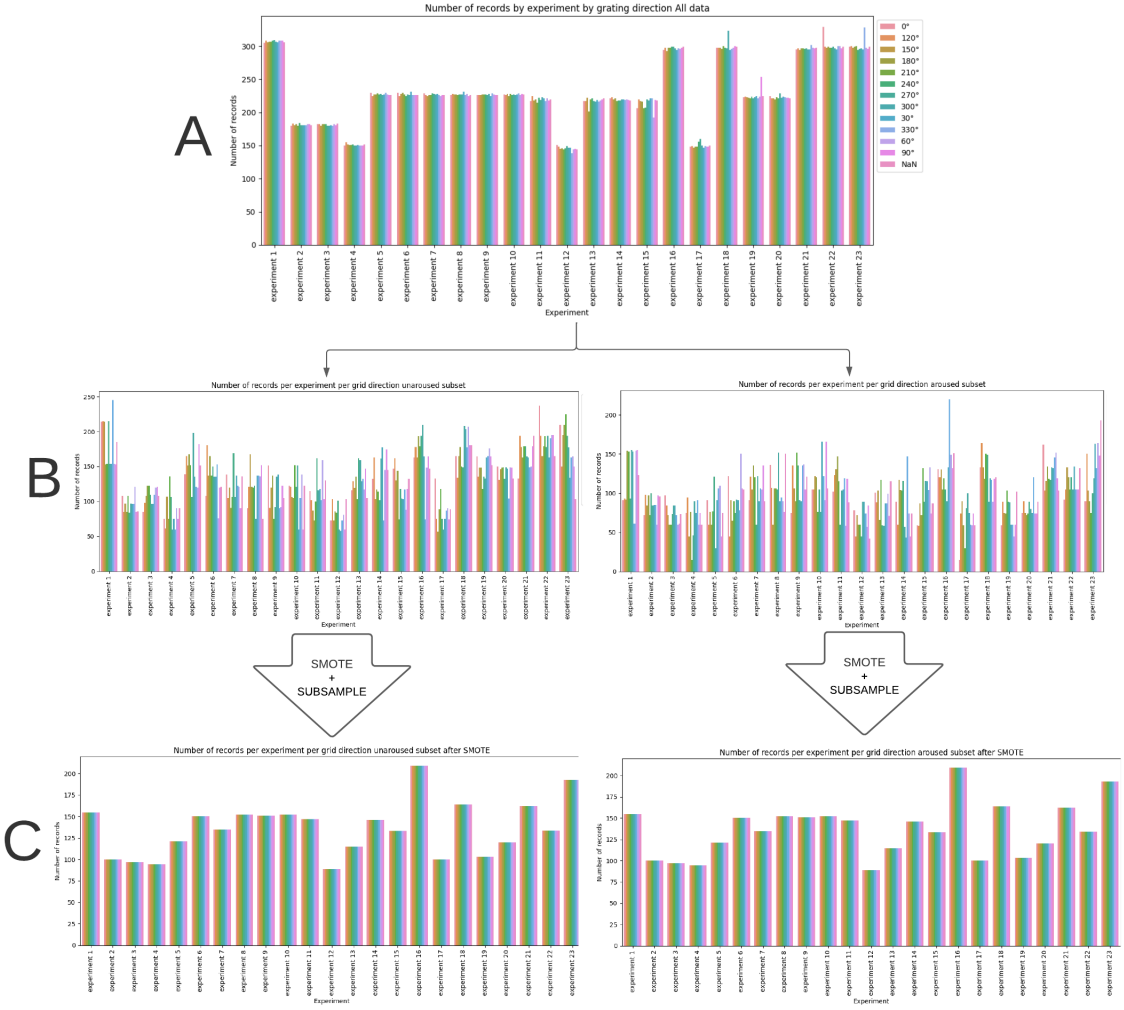

Figure 5.4: Class and sample balancing process.

5.5.2 Sample Imbalance

Subsampling

After performing oversampling using the SMOTE technique, we still face a second challenge: the number of samples in the non-arousal or small pupil subset is greater than the number of samples in the arousal or large pupil subset, as illustrated in Figure 5.5. In order to make an unbiased comparison, it is imperative to balance the two categories before proceeding with the analysis of results.

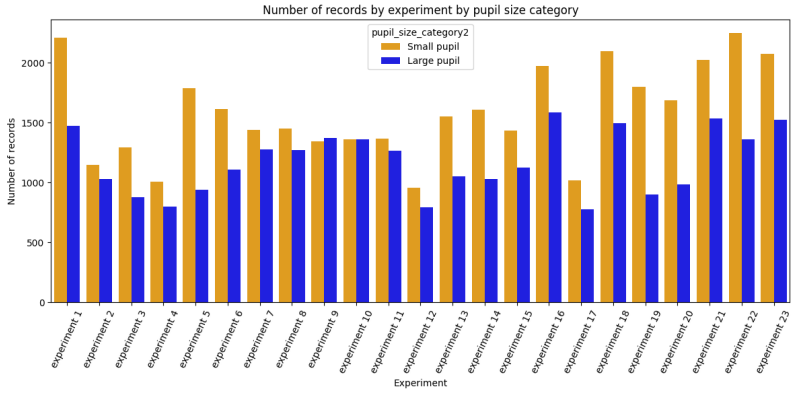

Figure 5.5: Number of records by experiment by pupil size category

In summary, by employing oversampling with the SMOTE technique and then simple subsampling, we are able to balance our dataset without losing crucial information and, at the same time, without running the risk of overfitting. This allows us to train two separate models, one for data with large pupils and another for small pupils, which are more accurate and representative of their respective categories. The final result of applying oversampling and subsampling to the dataset is shown in Figure 5.4 C.

5.6 Evaluation and Metrics

To evaluate the data we will use the metrics Accuracy, Precision, Recall and F1-score. In addition we will use the **k-fold cross-validation** technique to avoid relying on partitioning the data into a single training and test set, which can lead to biased or unreliable results [29].

Accuracy: Accuracy measures the proportion of correctly predicted instances over the total number of instances. It provides an overall assessment of the model's correctness.

Precision: Precision measures the proportion of true positive predictions (correctly predicted positive instances) over the total number of positive predictions (true positives + false positives). It indicates how well the model predicts positive instances accurately.

Recall: Recall (also known as sensitivity or true positive rate) measures the proportion of true positive predictions over the total number of actual positive instances (true positives + false negatives). It indicates how well the model captures positive instances.

F1-score: The F1-score is the harmonic mean of precision and recall. It provides a balanced measure of the model's accuracy and completeness.

K-fold cross validation

The K-fold cross-validation technique allows for the evaluation of a model's performance without being limited to a single training and test set, thus avoiding the bias that would

come from training the model on a reduced portion of the data. This is especially important with neural data, where measurements are highly variable. Cross-validation can help ensure that the model is not overfitting to noise or specific variability from a particular subset of the data.

For the model evaluation in each experiment, we perform a K-fold cross-validation with 10 iterations or 10 folds. In each iteration, the test fold changes, and the remaining 9 folds form the training set. At the end of the iterations, an average is obtained for the 10 values in each metric (Accuracy, Precision, Recall, and F1-score), as shown in Figure 5.6.

Figure 5.6: Example of K-fold cross validation

5.7 Cross-Generalization Analysis

Understanding the relationship between neuronal activity and responses to specific stimuli is crucial for deciphering the enigmas of brain processing. Particularly, visual stimuli, such as grating directions. However, there are different contexts or states under which this activity can be recorded, such as "arousal" and "non-arousal" states, which may influence how the brain responds to such stimuli. Therefore, a fundamental question arises: How do these models generalize when faced with data different from those on which they were trained? That is, if a model is trained on data from a specific state (for example, "arousal"), can it effectively predict responses in the other state ("non-arousal") or in a combination of both?

To address this issue, we introduce "Cross-Generalization Analysis." This approach examines a model's ability to generalize its learning from one subset of data to another. By training and evaluating models on different combinations of these data subsets (Figure 5.7), we aim to gain a deeper understanding of the similarities and differences underlying neuronal activity associated with visual stimuli between the "arousal" and "non-arousal" states.

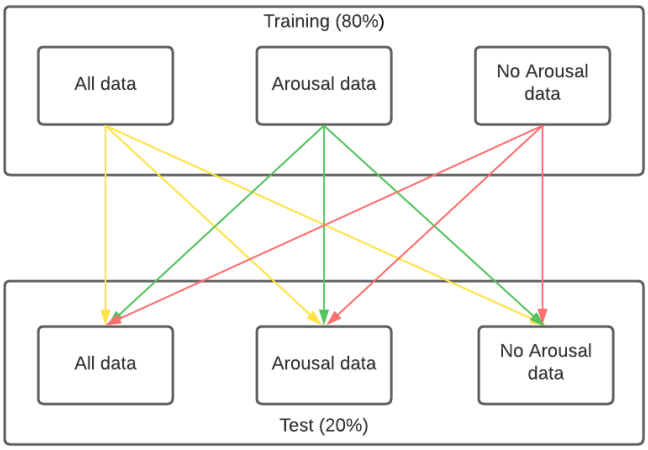

Figure 5.7: Cross-Generalization Analysis diagram

5.7.1 Model Trained on All Data:

Tested with aroused data: This metric tells us how the model, trained on both subsets, specifically generalizes on aroused data. Tested with unaroused data: Similar to the above, but for the unaroused subset.

5.7.2 Model Trained on Aroused Data:

Tested with all data: This gives us a perspective on how a model trained exclusively on aroused data generalizes to the entire dataset. Tested with unaroused data: This metric is especially interesting. If the model performs poorly here, it could indicate that the neural features for visual stimuli differ significantly between the aroused and unaroused states.

5.7.3 Model Trained on Unaroused Data:

Similar to the previous scenario but reversed. It tells us how the model trained exclusively on unaroused data behaves on different subsets.

Why Perform This Analysis?

- **Generalization:** This approach allows us to understand how models trained on one data subset can generalize to other subsets or to the entire dataset.

- **Specificity:** If we find that a model trained on a specific subset (e.g., aroused) performs significantly better on that same subset compared to the other, it could indicate that there are unique and specific features in that subset that the model is learning.
- **Transfer Learning:** If a model trained on one subset performs well on the other subset, it could indicate that there is an overlap in neural features between the two states and that the model is capable of transferring what it has learned from one subset to the other.

In summary, this analysis will allow us not only to understand the performance of the model, but also the underlying characteristics of its data. This can be crucial for interpreting the results and identifying key features.

5.8 Hypothesis

The hypothesis being proposed is that the accuracy in the subset of data exhibiting arousal is superior to the accuracy in the subset of data that do not exhibit arousal. To test or refute this hypothesis, we perform a paired T-test.

The following are established:

- **Null Hypothesis (H_0):** Posits that there is no difference in the means of the two paired samples. In this context, the null hypothesis would be that there is no difference in accuracy between the 'Arousal' and 'Non-arousal' subsets for each experiment.

  $\mu_{Arousal} = \mu_{NonArousal}$

- **Alternative Hypothesis (H_a):** This is the hypothesis you want to test. In this case, you are interested in seeing if accuracy is higher in the subsets with 'Arousal.' Therefore, your alternative hypothesis would be:

  $\mu_{Arousal} > \mu_{NoArousal}$

  Where:

  $\mu_{Arousal}$ is the mean accuracy for the 'Arousal' subset.

  $\mu_{NonArousal}$ is the mean accuracy for the 'No Arousal' subset.

  Upon conducting the T-test, if the p-value obtained is less than a significance threshold (usually 0.05), the null hypothesis is rejected in favor of the alternative hypothesis. In other words, if you obtain a p-value less than 0.05, it is concluded that there is a significant difference in accuracy between the subsets, and that accuracy is, on average, higher in the 'Arousal' subsets than in the 'No Arousal' subsets.

Chapter 6

Results and discussion

This section addresses the results obtained after implementing various modeling strategies and their subsequent evaluation. First, the implementation of individualized models for each of the 23 experiments is discussed, focusing on their respective performances and the variables that showed the highest correlation with accuracy. Following this, an additional dimension to the analysis is introduced: Arousal as a variable of interest and how it may impact the efficacy of the model.

Subsequently, the results of the hypothesis test are presented to evaluate differences in the model's accuracy between 'Arousal' and 'Non-Arousal' states. This test provides a statistical perspective on the influence of emotional states on the prediction of neuronal activity.

Finally, the results of the cross-generalization capability analysis of the model are described. This last analysis seeks to understand how models trained under certain conditions (e.g., data representing 'Arousal' or 'Non-Arousal' states) perform when subjected to conditions different from those under which they were trained.

6.1 Individualized Modeling

Each experiment, out of a total of 23, was treated as a unique entity, developing a specific model for each one. This strategy was based on the premise that different experiments may have subtle variations that could affect the model's effectiveness.

Following the first signs of overfitting in an 80/20 split scheme without cross-validation, we opted for the cross-validation technique with 10 folds. This decision was supported by existing literature that highlights the effectiveness of cross-validation in providing a more robust and generalizable assessment of the model's performance [30].

6.1.1 Model Performance

Out of the 23 experiments, 10 yielded an overall accuracy higher than 80%. 5 experiments showed an accuracy between 60% and 80%. Only 2 experiments resulted in an accuracy between 40% and 60%, and another 2 experiments had accuracy lower than 40%.

The range of accuracy of the models varied, indicating that some experiments offer clearer or more consistent data for prediction. For a detailed breakdown of the results by class in the confusion matrix for each experiment, see the appendix A.1. These findings could open the door to additional investigations on the specific characteristics of the experiments and their influence on predictability.

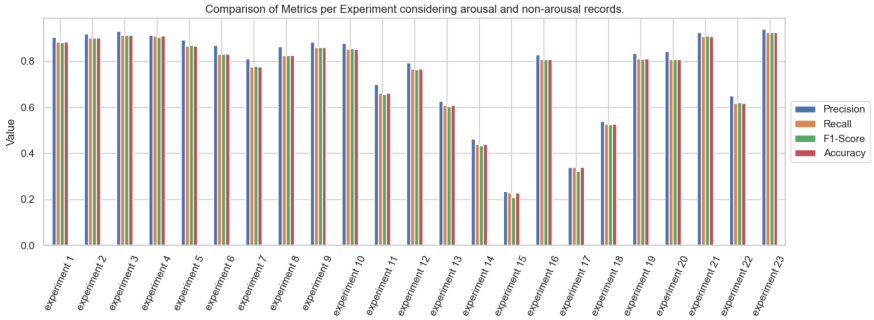

Figure 6.1: Comparison of metrics by experiment considering all data in the model.

6.1.2 Correlations

We plotted the correlations between evaluation metrics and various factors such as the number of neurons, the number of data points, the product of the number of neurons and data points (information density), and the standard deviation of the direction tuning curves.

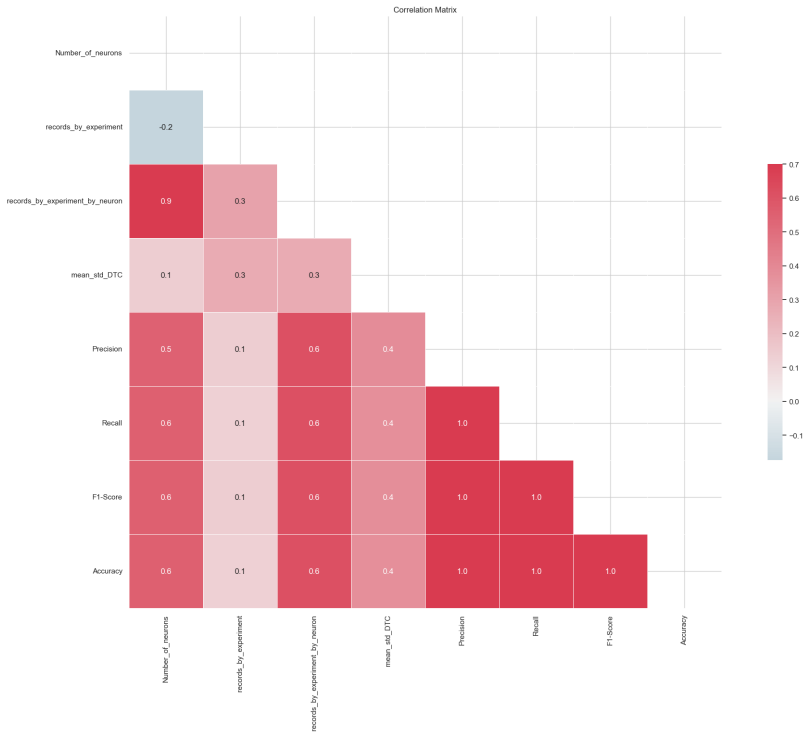

Figure 6.2: Correlations between metrics and different parameters

We found that accuracy is most strongly correlated with the product of the number of data points and the number of neurons per experiment, exhibiting a Pearson correlation of 0.6. Interestingly, neither the number of data points alone nor the number of neurons alone showed a significant correlation. This suggests that information density, defined as the product of data points and neurons, could be a valuable metric for future studies. The scatter plot illustrating the relationship between accuracy and the information density (data points multiplied by neurons) is presented below

Additionally, we identified a correlation of 0.4 between accuracy and the average standard deviation of the direction tuning curves for all neurons in each experiment. While this correlation is not particularly strong, it does suggest that there is some relationship between the accuracy in predicting visual stimuli and the variability in neuronal responses to stimuli from different directions.

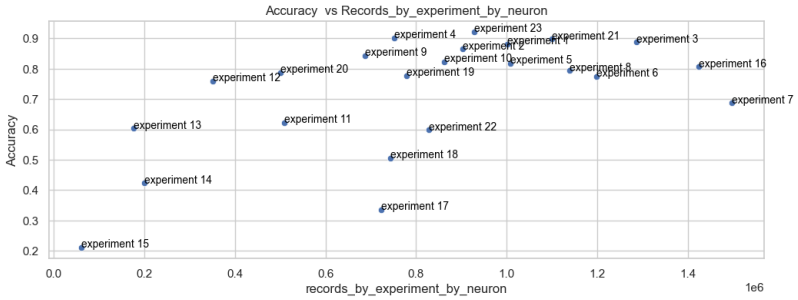

Figure 6.3: Accuracy vs Records by experiment by neuron

6.2 Modeling Based on Arousal Level

We developed two distinct models based on the arousal level (arousal/non-arousal): one for neuronal measurements that aligned with the arousal state, and another for those that aligned with the non-arousal state. After segmenting the data into arousal and non-arousal categories, we observed an imbalance in class distribution, with a significant asymmetry in the number of records. Specifically, the majority of experiments (21 out of 23) had more records associated with the small pupil (non-arousal) category.

Due to this imbalance, we employed the SMOTE technique for oversampling to balance the classes. Subsequently, we performed subsampling of the records to equalize the number of samples in both subsets before model training.

Overall, the model exhibits high performance, with metrics like Precision, Recall, F1-Score, and Accuracy exceeding 90% in most experiments. However, there is variability in these metrics across different experiments, as seen in Experiment 15 under 'Arousal' conditions, which demonstrates significantly lower performance compared to the others.

When comparing model performance under 'Arousal' and 'Non-Arousal' conditions (Figure 6.4), a trend towards better performance in the 'Arousal' state is observed. This suggests that the model may encode visual stimuli more effectively when the subject is in an alert or excited state. However, this improvement is not uniformly observed across all experiments, indicating variability that may warrant further investigation.

The implications of these findings are manifold. First, they confirm the model's general effectiveness in the prediction task. Second, they indicate that the 'Arousal' factor could play a significant role in model efficacy and may serve as a focal point for future research. Lastly, the experiments with lower performance could be examined more closely to improve the model.

For a detailed breakdown of the results by class in the confusion matrix for each experiment, see Appendix A.2 for arousal and A.3 for non-arousal.

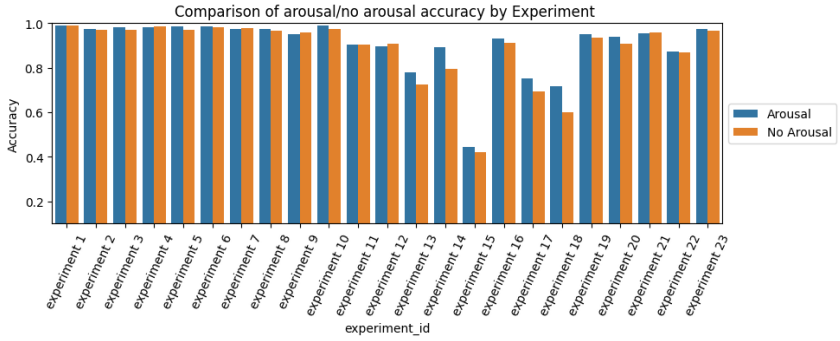

Figure 6.4: Comparison of arousal/non-arousal accuracy by experiment

6.3 Hypothesis Test Results

The t-test results indicate a statistically significant difference between the two groups ('Arousal' and 'Non-Arousal') concerning the 'Accuracy' metric.

1. **T-statistic Value**: 3.0063. A high t-value suggests that the means of the two groups differ significantly.
2. **P-value**: 0.0065. A p-value less than 0.05 (the commonly accepted threshold for statistical significance) suggests that the observed difference in means is statistically significant.

Given that the p-value is below 0.05, we reject the null hypothesis (H_0) stating that the 'Accuracy' means are equal across the 'Arousal' and 'Non-Arousal' groups. This supports the alternative hypothesis (H_a) that there is a difference in 'Accuracy' between the two groups.

Additionally, since the t-value is positive, we can infer that the 'Arousal' group has a higher mean 'Accuracy' compared to the 'Non-Arousal' group, which is in line with the original hypothesis.

In summary, the results indicate that there is a significant difference in 'Accuracy' between the two groups, and this difference favors the 'Arousal' group.

6.4 Cross-Generalization Analysis

6.4.1 Overall Performance

The model trained on the full dataset (and evaluated on an unseen 20% of the data) achieves an accuracy of approximately 82%. This suggests that the model demonstrates solid performance when generalizing to new data, provided that those new data represent a similar mix of "Arousal" and "Non-Arousal" states.

6.4.2 Generalization from "All Data"

The model trained on the entire dataset generalizes exceptionally well to both "Arousal" and "Non-Arousal" states, with accuracies of approximately 91% and 92%, respectively. This could indicate that the model is capable of capturing features that are generally useful for classifying both states.

6.4.3 Training on "Arousal"

When the model is trained exclusively on "Arousal" data, its accuracy significantly declines when generalizing to the entire dataset (68%) or solely to the "Non-Arousal" set (59%). However, it performs decently when evaluated on the same "Arousal" subset (82%). This suggests that features in "Arousal" data may not be fully generalizable.

6.4.4 Training on "Non-Arousal"

Similarly, the model trained on "Non-Arousal" data shows significantly lower accuracy when generalizing to the "Arousal" set (52%). However, much like its counterpart trained on "Arousal", it exhibits decent performance when evaluated on the same "Non-Arousal" subset (82%).

6.4.5 Implications

The results support the notion that there are significant differences between "Arousal" and "Non-Arousal" data. Models specifically trained on one of these sets struggle to generalize to the other.

Interestingly, a model trained on a mixture of both types of data appears to capture features that are useful for classifying both "Arousal" and "Non-Arousal" states. This suggests that there is an overlap in relevant features between the two states, which is only captured when they are trained together.

The distribution of the accuracy values of each of the analysis combinations can be seen in Figure 6.5.

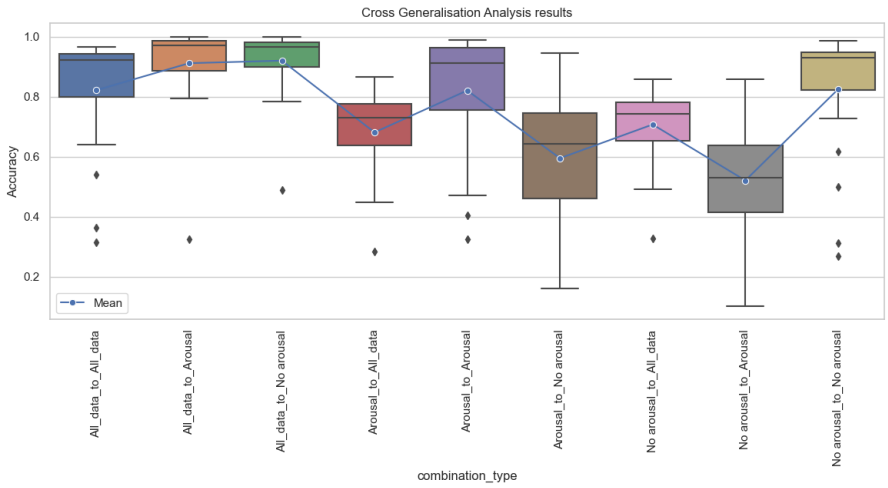

Figure 6.5: Cross generalisation analysis results.

Chapter 7

Conclusion

The study embarked with clear and specific objectives focused on understanding how different behavioral states, specifically arousal and non-arousal, influence the neural encoding of visual stimuli in a large neural population. Additionally, it aimed to develop a machine learning model capable of predicting these visual stimuli based on neural activity and to evaluate its accuracy across different behavioral states.

In terms of meeting these objectives, the study was successful on multiple fronts. First, it managed to develop a Support Vector Machine model that showed solid performance in the prediction task, with an overall accuracy exceeding 80% in 10 out of the 23 experiments conducted.

Second, the study incorporated a meticulous evaluation of the model's accuracy across different behavioral states using a 10-fold cross-validation technique. The results showed that the model's accuracy was variable when trained and evaluated on different behavioral states. This variability supports the need to consider behavioral state in neural encoding and meets the objective of evaluating the model's accuracy across different states.

Third, a detailed comparison of models trained on data collected during arousal and non-arousal states was conducted. The results showed that these models struggled to generalize across the two states, suggesting that there are significant differences in neural encoding between these behavioral states.

Additionally, the study also yielded several key findings. A significant correlation was found between accuracy and information density, defined as the product of the number of records and the number of neurons per experiment. This finding could serve as a useful metric in future studies.

The t-test results were particularly revealing, indicating a statistically significant difference in accuracy between the 'Arousal' and 'Non-Arousal' groups. This finding supports the hypothesis that behavioral state has a significant impact on the efficacy of the machine learning model in predicting visual stimuli.

In summary, the study successfully met its general and specific objectives, providing valuable insights into the influence of behavioral states on the neural encoding of visual stimuli.

The results have both theoretical and practical implications and pave the way for future research that could focus on better understanding these dynamics and improving the efficacy of machine learning models in this context.

7.1 Future work

One area that requires further investigation is the identification of parameters that have a high correlation with accuracy and can explain neural behavior in a more comprehensive manner. For example, a metric that quantifies how selective each neuron is could offer more insights into the variability in model accuracy observed across different behavioral states.

Another interesting development for gaining better insights into neural behavior in relation to model accuracy is the use of a visual stimulus prediction model to determine which neurons are most critical for enhancing accuracy and which are not.

Reverse engineering could be applied to the visual stimulus prediction model in an initial state, and then the activity of certain neurons could be selectively altered to assess their impact on model accuracy. Specifically, we could "deactivate" or minimize the influence of particular neurons and observe how this affects the model's overall performance. This method may shed light on which neurons are essential for maintaining high accuracy.

7.2 References

[**1**] Christensen, A. J., Pillow, J. W. (2022). Reduced neural activity but improved coding in rodent higher-order visual cortex during locomotion. Nature Communications, 13(1), 1676.

[**2**] Stringer, C., Pachitariu, M., Steinmetz, N., Reddy, C. B., Carandini, M., Harris, K. D. (2019). Spontaneous behaviors drive multidimensional, brain-wide population activity. Neuron, 103(6), 1056-1069.

[**3**] Schröder S, Steinmetz NA, Krumin M, et al. Arousal Modulates Retinal Output. (2020) Neuron. 107(3):487-495.e9. doi:10.1016/j.neuron.2020.04.026

[**4**] Stringer, C., Pachitariu, M., Steinmetz, N., Reddy, C. B., Carandini, M., Harris, K. D. (2019). Spontaneous behaviors drive multidimensional, brainwide activity. Science, 364(6437), eaav7893.

[**5**] Lotfi, N., VanRullen, R., Schyns, P. G. (2019). Deep learning convolutional networks for predicting brain responses to visual stimuli. Scientific Reports, 9(1), 1-15.

[**6**] Yamins, D. L. K., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., DiCarlo, J. J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. Proceedings of the National Academy of Sciences, 111(23), 8619-8624.

[**7**] Güçlü, U., van Gerven, M. A. (2015). Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. Journal of Neuroscience, 35(27), 10005-10014.

[**8**] Rajalingham, R., Issa, E. B., Bashivan, P., Kar, K., Schmidt, K., DiCarlo, J. J. (2018). Large-scale, high-resolution comparison of the core visual object recognition behavior of humans, monkeys, and state-of-the-art deep artificial neural networks. Journal of Neuroscience, 38(33), 7255-7269.

[**9**] Databricks (2021). 'What is Parquet?'. Available at: https://www.databricks.com/glossary/what-is-parquet (Accessed: 1 September 2023).

[**10**]Bradley, M. M., Miccoli, L., Escrig, M. A., Lang, P. J. (2008). The pupil as a measure of emotional arousal and autonomic activation. Psychophysiology, 45(4), 602-607.

[**11**] Laeng, B., Sirois, S., Gredeback, G. (2012). Pupillometry: A window to the preconscious?. Perspectives on Psychological Science, 7(1), 18-27.

[**12**] Hess, E. H., Polt, J. M. (1960). Pupil size as related to interest value of visual stimuli. Science, 132(3423), 349-350.

[**13**] Oracle (2021). 'What is a Relational Database?'. Available at: https://www.oracle.com/cl/database/what-is-a-relational-database/ (Accessed: 1 September 2023).

[**14**] Heydt, M. (2020). 'Learning SQL: Master SQL Fundamentals'. O'Reilly Media.

[**15**] Seaborn developers (2021). 'Seaborn: Statistical Data Visualization'. Available at: https://seaborn.pydata.org/ (Accessed: 1 September 2023).

[**16**] Scikit-learn developers (2021). 'Scikit-learn: Machine Learning in Python'. Available at: https://scikit-learn.org/stable/documentation.html (Accessed: 1 September 2023).

[**17**] Gilbert, C. D., Wiesel, T. N. (1990). The influence of contextual stimuli on the orientation selectivity of cells in primary visual cortex of the cat. Vision Research, 30(11), 1689-1701.

[**18**] Scikit-learn developers (2021). 'Support Vector Machines'. Available at: https://scikit-learn.org/stable/modules/svm.html (Accessed: 1 September 2023).

[**19**] The principle behind SVMs: hyperplanes, margins and support vectors Cortes, C., Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297.

[**20**] Kernel types and their applicability to different problems Schölkopf, B., Smola, A. J., Müller, K. R. (1999). Kernel principal component analysis. In Advances in kernel methods (pp. 327-352). MIT Press.

[**21**] Previous studies that have used SVM to decode and predict neural activity Pereira, F., Mitchell, T., Botvinick, M. (2009). Machine learning classifiers and fMRI: a tutorial overview. NeuroImage, 45(1), S199-S209.

[**22**] Advantages of using SVM in the context of neural analysis Haynes, J. D., Rees, G. (2006). Decoding mental states from brain activity in humans. Nature Reviews Neuroscience, 7(7), 523-534.

[**23**] Scikit-learn developers (2021). 'GridSearchCV: Exhaustive search over specified parameter values for an estimator', scikit-learn: Machine Learning in Python. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html (Accessed: 1 September 2023).

[**24**] Scikit-learn.org. (2019). 'RBF SVM parameters — scikit-learn 0.21.3 documentation'. [online] Available at: https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html (Accessed: 1 September 2023).

[**25**] Bhatt, B. (n.d.). svm-c-gamma-hyperparameter. [online] Deepnote. Available at:

https://deepnote.com/@bhaveshbhatt/svm-c-gamma-hyperparameter-ec7cdd4f-b499-4b4d-a320-f483e8099691 .

[**26**] He, H., Ma, Y. (2013). Imbalanced learning: foundations, algorithms, and applications. John Wiley Sons.

[**27**] Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence, 5(4), 221-232.

[**28**] Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.

[**29**] Kohavi, R. (1995). "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection". Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2, 1137–1143.

[**30**]Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In IJCAI (Vol. 14, No. 2, pp. 1137-1145)

Appendix A

Confusion matrix

A.1 All data model confusion matrix

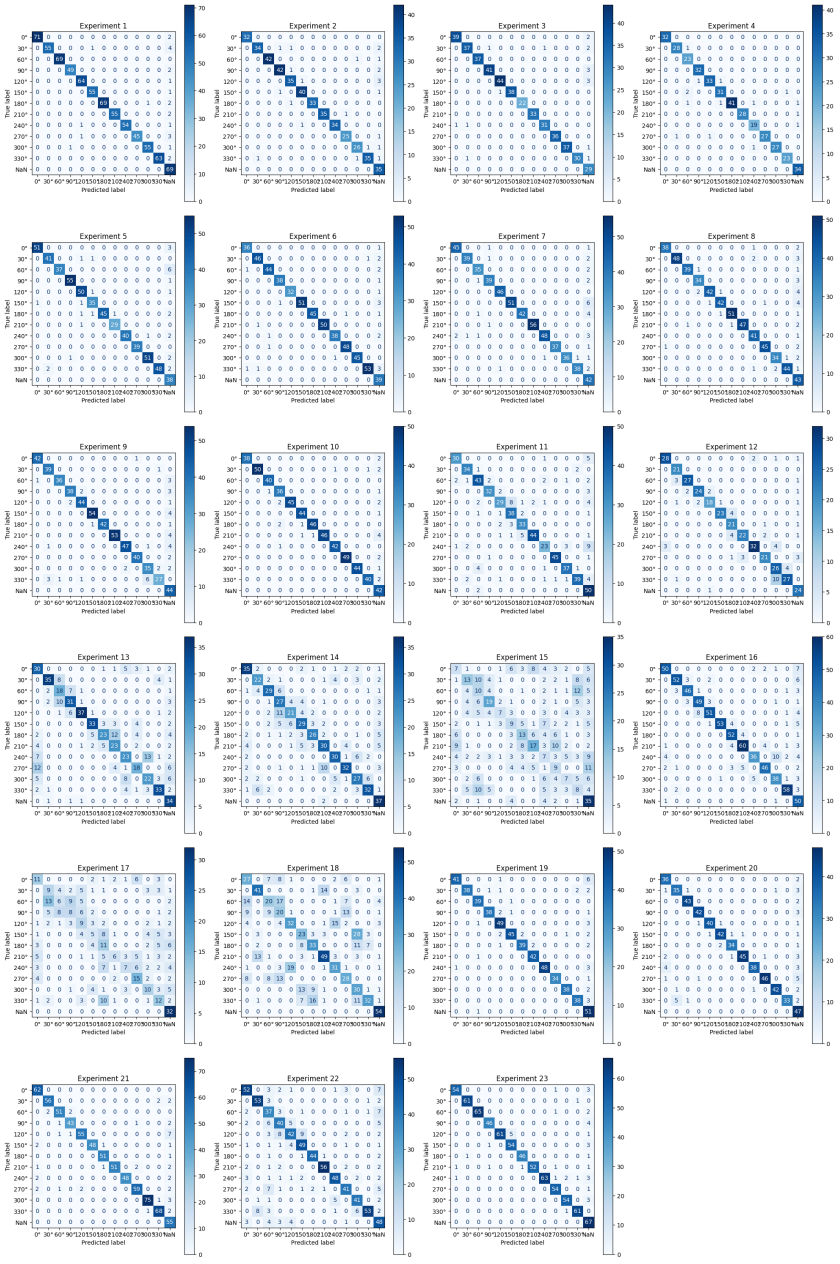

Figure A.1: All data model confusion matrix

A.2 Arousal model confusion matrix

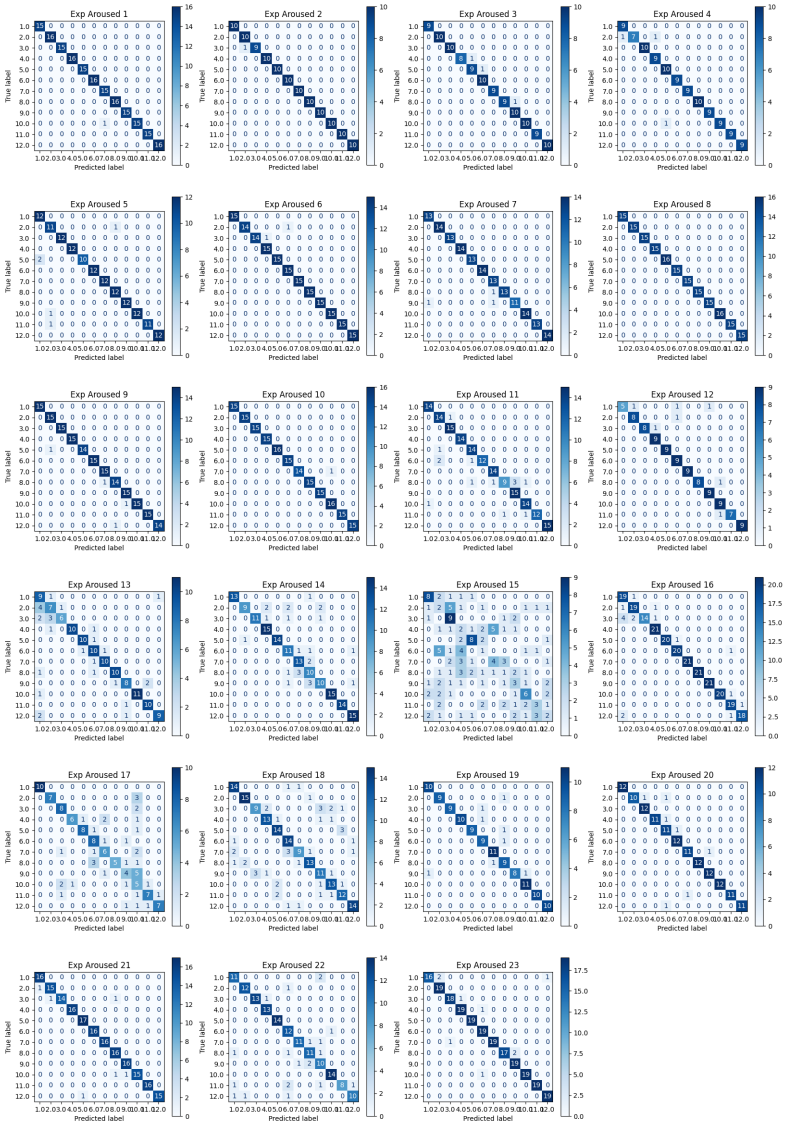

Figure A.2: Arousal model confusion matrix

A.3 Non-arousal Model Confusion Matrix

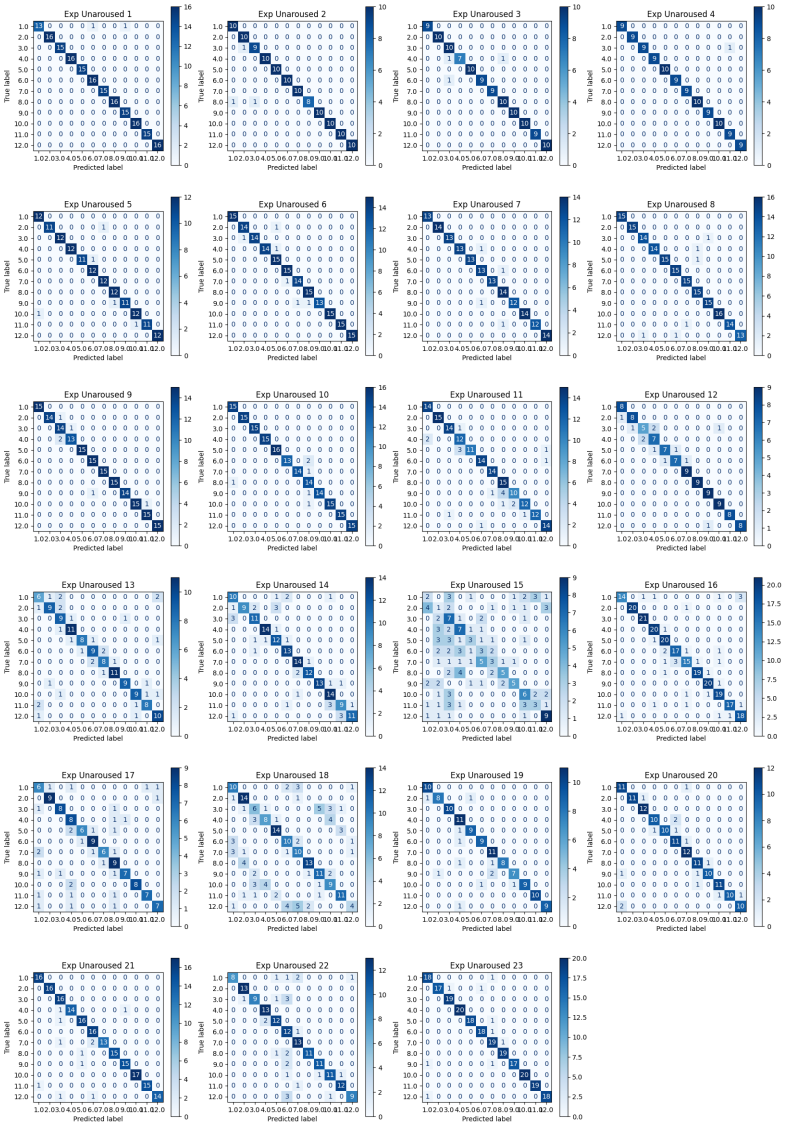

Figure A.3: Non-arousal Model Confusion Matrix