Classifying Neuronal Morphological Characteristics—Spiny vs Aspiny—Using Electrophysiological Features from the Allen Cell Types Database

Saud Zakariya Hussain | CS7180 Special Topics in AI | 08/16/2019

Introduction:

The biological sciences are a prime hotspot for retrieving exorbitant amounts of data—ranging from physiological characteristics of large mammals (e.g., Whales, humans, etc.), down to nanoscale phenomena that can be measured.

In this paper, I focus on bridging the gap between data acquired from neurobiological experimentation with modern, classification-techniques currently used in data science—a field involved in extracting insight from structured and unstructured data. More specifically, I seek to answer the following question: “Is it feasible to classify whether a neuron has Aspiny or spiny dendrites given information about its firing properties?”

To succeed in painting a clear picture of how I conducted my analysis to you, the reader, I will use the following outline:

- Define key terminology/background knowledge for understanding the biology.
- Provide an overview of the data.
- Provide an overview of the experiment used for data-acquisition
- Express the importance of why classifying morphological features of neurons using electrophysiological properties makes this dataset exhilarating to work with.
- Delve into my exploration of the dataset.
- Explain my approach to selecting classification models to fit this dataset.
- Discuss the results of my classification models
- Conclusion: summarize my data exploration and determine future directions.

Background & Key Terms to Understand the Biology:

Aspiny vs Spiny Neuron:

A neuron itself is a type of cell found within the nervous system of anything capable of locomotion. There are some x-billion of these cells within a human brain, and they form complex circuitries by communicating to each other across a synapse via an axon terminal to dendrite connection; where the axon terminal can be thought of as the tail of an upstream cell, and the dendrite belongs to the head of a downstream cell **(Figure 1)**.

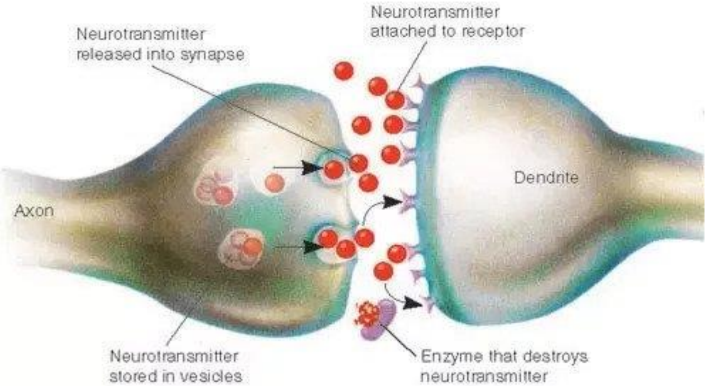


Figure 1: This image illustrates the connection between an upstream and downstream neuron across a synapse. The red circles show neurotransmitter release from axon terminal travelling across the synapse towards the dendrite of a downstream neuron. Source: https://qph.fs.quoracdn.net/main-qimg-295b6866e095f34a588a13c06f9a645d.webp

One important realization, is that there are different types of neurons with different morphologies (shapes), electrophysiological characteristics, etc.

For my data science exploration, I will be looking at two categories of neurons, spiny and Aspiny. This is a morphological characteristic expressed across most neurons. Specifically, this is a feature found on the dendrites of a neuron. A spiny neuron has projections seen along its dendrites, thus increasing the surface area available to maximize the number of connections possible with other neurons; whereas, an Aspiny does not have these projections **(Figure 2)**. A neuron can morph between these two categories.

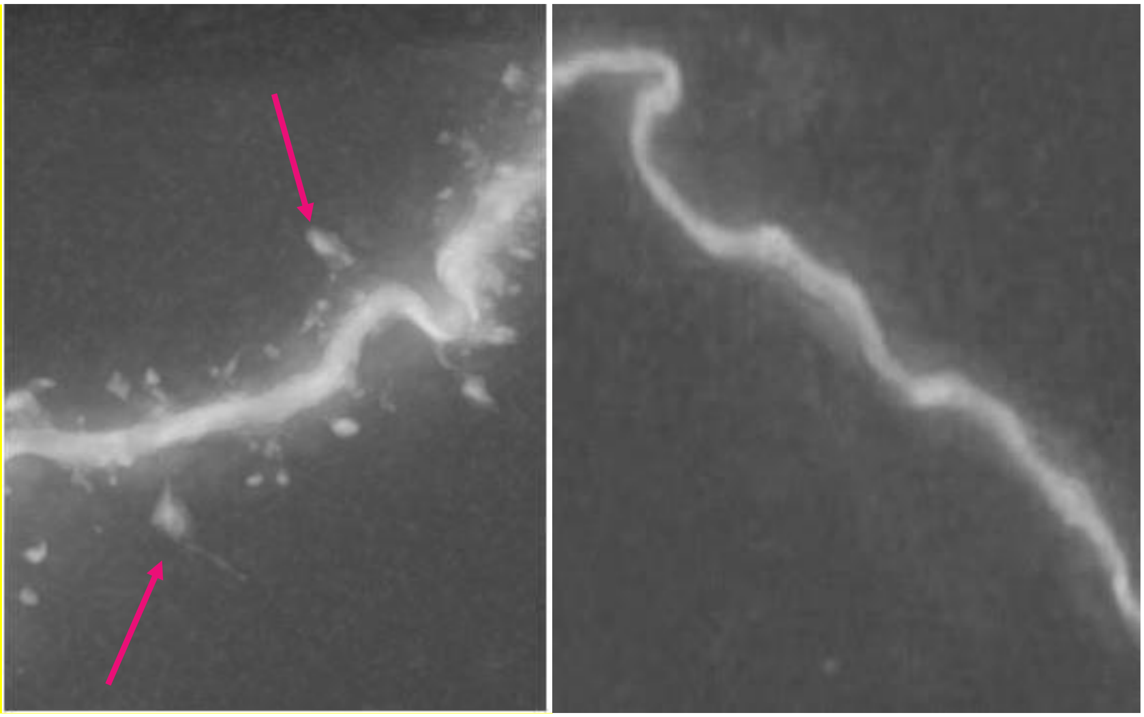


Figure 2: illustrates the difference between a spiny dendrite (left) vs an Aspiny dendrite (right). In the left image depicting a spiny dendrite, the red arrows point to some example of spines on the dendrite. We see in the image of the Aspiny dendrite it is smooth and has no projections. Source: https://www.mozak.science/guide/neuron-types

Electrophysiology:

In the context of this paper, electrophysiological studies involve measuring the response of neurons in a living organism to a given stimulus (e.g., current injections using a stimulation electrode, exposing the subject video recordings). The data collected from this type of experiment provides insight into both intra- and extracellular activity and generally comes in the form of a voltage recording—the result of changing membrane potential of the cell due to stimulus.

Action Potential: Single Spike vs Spike Train

As mentioned above, the result of changing the membrane potential of a neuron is observed through a voltage measurement between the inside and outside of a cell. A cell at rest has a resting potential that is generally near -70mV. When a stimulus causes the neuron to depolarize (i.e., become more positive), if the difference across the cell membrane reach a threshold value (generally above -45 mV) specific to the neuron, an action potential occurs.

This event causes the stimulated cell to release of neurotransmitters across the synaptic cleft towards the dendrite of a downstream neuron.

Neurons, however, are not limited to eliciting a mere single, action protentional. Rather, depending on the duration/type of the stimuli, the neuron may respond with only a single spike **(Figure 3)**, or a series of spikes—referred to as a spike train **(Figure 4)**.

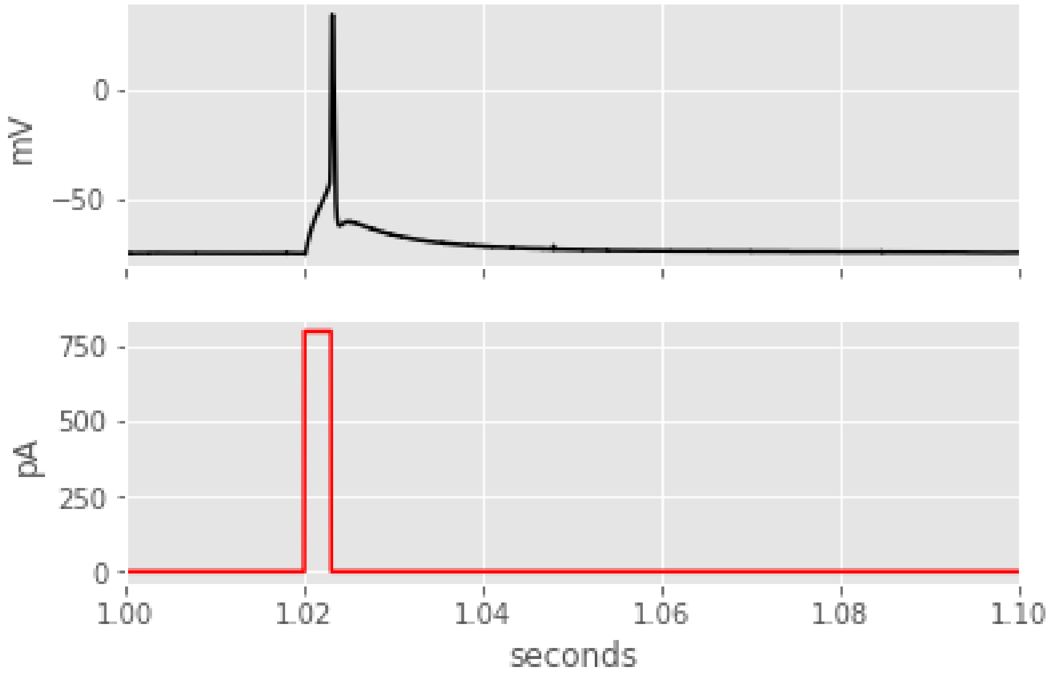


Figure 3: Depicts a single spike (action potential) response (top) elicited by a short-square current injection (bottom). We see from the top recording that pre-stimulation, the neuron has a resting potential below -60mV. Shortly after the short-square stimuli is applied, the neuron is brought to its threshold, just above -50mV, and the action potential occurs.
Data Source: Allen Cell types DB, plotted using matplotlib in Jupyter notebook

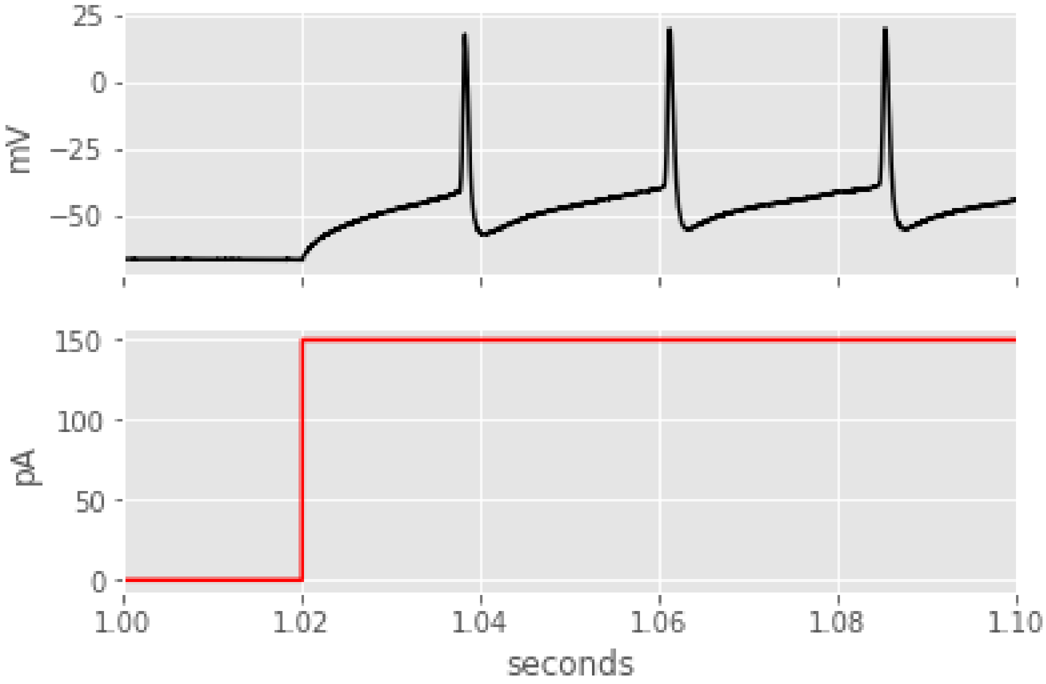


Figure 4: Depicts a spike train (i.e., a continuous series of action potential) response (top) elicited by a long-square current injection (bottom) on the same cell used in **Figure 3** above. We see from the top recording that pre-stimulation, the neuron has a resting potential below -60mV. Shortly after the long-square stimuli is applied, the neuron is brought to its threshold, just above -50mV, there is a continuous burst of action potential per the duration of the stimulus.
Data Source: Allen Cell types DB, plotted using matplotlib in Jupyter notebook

Overview of Data:

The data used in my analysis was gathered by the Allen Institute for Brain Science. It is stored in the Allen Cell Types Database, which contains electrophysiological, morphological, and transcriptomic data.

From within this database, I used their software development kit (sdk) and application programming interface (api) to extract the data containing electrophysiological responses from both human and mouse brains. The cells selected came from select brain areas, such the visual subregions in the mouse brain and different layers of the cortex in the human brain.

Experiment Used in Data Acquisition:

Electrophysiological data from single cells was acquired using whole cell patch clamping—a popular procedure used to study ionic currents in an individual cell (**Figure 5**). A micropipette is used to contact the membrane of an isolated cell. A small suction is applied creating a seal between the pipette and membrane. Therefore, when an ion channel opens, as a result of depolarizing the cell, ions flow into the pipette and an electronic amplifier is used to record the current flowing through an ion channel.

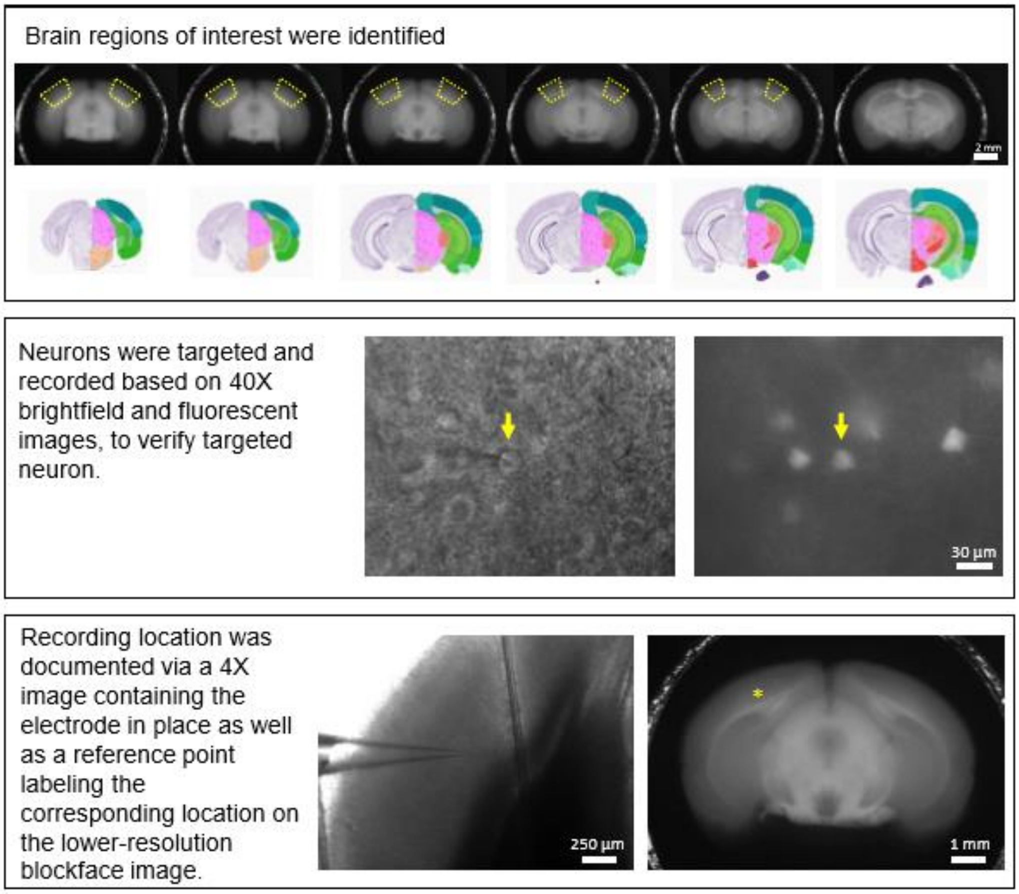

Brain regions of interest were identified
2 mm
Neurons were targeted and recorded based on 40X brightfield and fluorescent images, to verify targeted neuron.
30 µm
Recording location was documented via a 4X image containing the electrode in place as well as a reference point labeling the corresponding location on the lower-resolution blockface image.
250 µm
1 mm

Figure 5: This diagram depicts the process of identifying and targeting neurons in the mouse primary visual cortex. The Top panel indicates the region of interest identified in a within a slice of the mouse cortex bounded within the yellow boxes. The middle panel shows isolated cells within the bounded region. The bottom panel depicts the use of a micropipette and an electrode at the location indicated by the yellow asterisk. Source: This diagram comes directly from the Allen Cell Types Database Electrophysiology Technical Whitepaper.

Once an isolated cell within a select area is clamped, a range of stimulus protocols are used to characterize the neuron's electrophysiological properties. Each cell undergoes a single experiment composed of a series of different sweeps, where each sweep is used gather the cells response to a specified stimulus—types of stimulus are depicted in **Figure 7**. For quality data assurance, a set of test sweeps is used during each experiment. This ensures the data across cell experiments are consistent. For example, making contact to a cell membrane and applying suction could damage the membrane, thus causing a shift in the current measured through the micropipette. The chronology of the test sweeps in the experiment is shown in **Figure 6**.

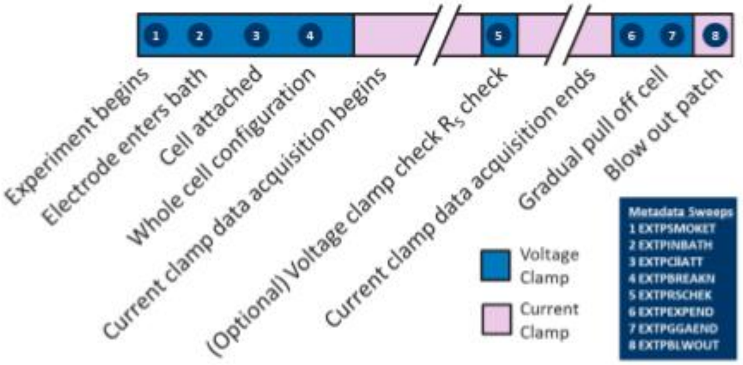
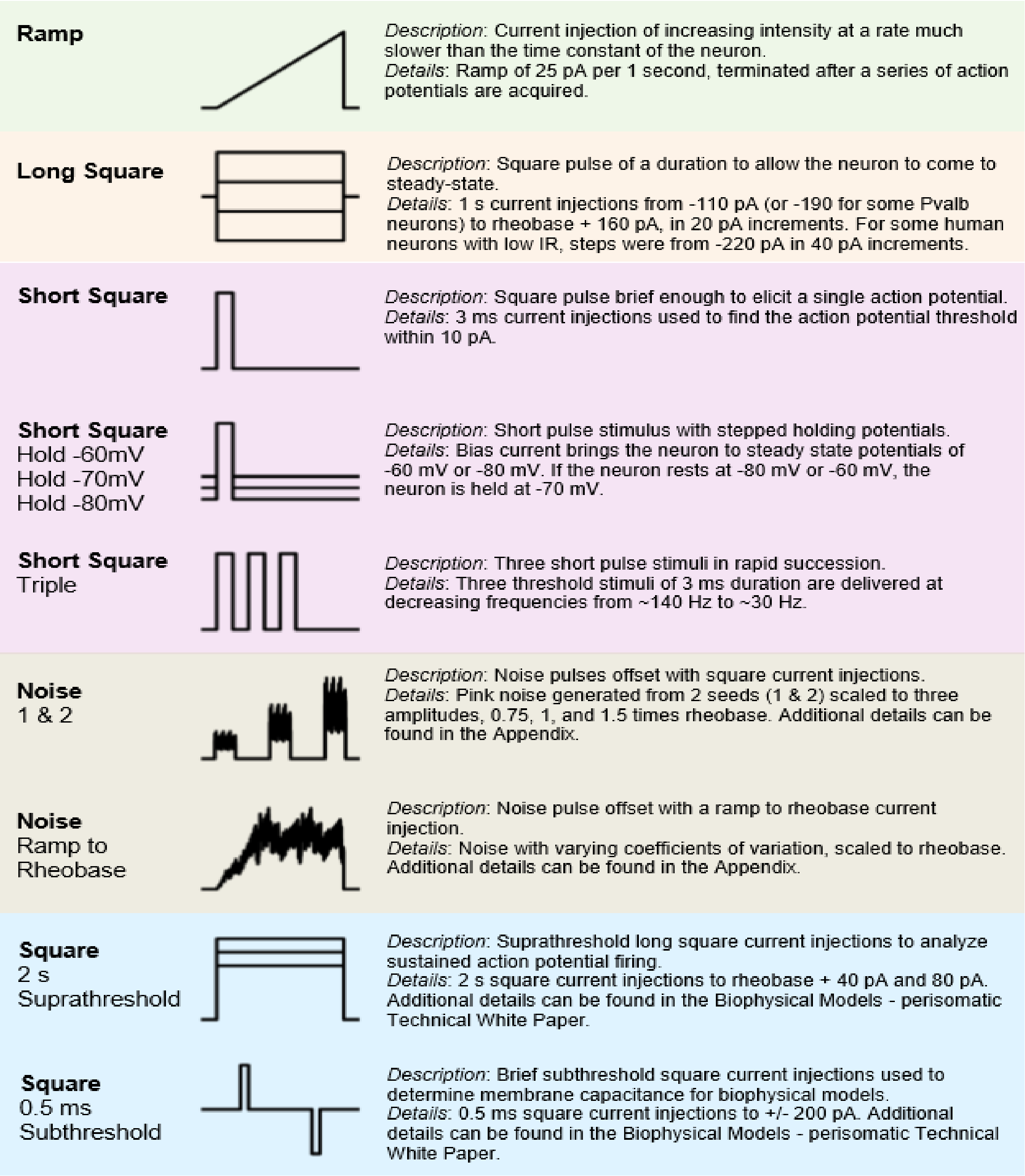


Figure 6: Shows the chronological ordering in which test sweeps where perform during experimentation. After the first four sweeps, clamp data acquisition begins and the stimuli, such as the short-square ramp depicted in **Figure 7**, is used on a single cell. At the very end of the experiment test sweeps are performed to measure the resistance of the patch and determine drift from experimentation.
Source: This diagram comes directly from the Allen Cell Types Database Electrophysiology Technical Whitepaper.



Ramp	*Description*: Current injection of increasing intensity at a rate much slower than the time constant of the neuron. *Details*: Ramp of 25 pA per 1 second, terminated after a series of action potentials are acquired.
Long Square	*Description*: Square pulse of a duration to allow the neuron to come to steady-state. *Details*: 1 s current injections from -110 pA (or -190 for some Pvalb neurons) to rheobase + 160 pA, in 20 pA increments. For some human neurons with low IR, steps were from -220 pA in 40 pA increments.
Short Square	*Description*: Square pulse brief enough to elicit a single action potential. *Details*: 3 ms current injections used to find the action potential threshold within 10 pA.
Short Square Hold -60mV Hold -70mV Hold -80mV	*Description*: Short pulse stimulus with stepped holding potentials. *Details*: Bias current brings the neuron to steady state potentials of -60 mV or -80 mV. If the neuron rests at -80 mV or -60 mV, the neuron is held at -70 mV.
Short Square Triple	*Description*: Three short pulse stimuli in rapid succession. *Details*: Three threshold stimuli of 3 ms duration are delivered at decreasing frequencies from ~140 Hz to ~30 Hz.
Noise 1 & 2	*Description*: Noise pulses offset with square current injections. *Details*: Pink noise generated from 2 seeds (1 & 2) scaled to three amplitudes, 0.75, 1, and 1.5 times rheobase. Additional details can be found in the Appendix.
Noise Ramp to Rheobase	*Description*: Noise pulse offset with a ramp to rheobase current injection. *Details*: Noise with varying coefficients of variation, scaled to rheobase. Additional details can be found in the Appendix.
Square 2 s Suprathreshold	*Description*: Suprathreshold long square current injections to analyze sustained action potential firing. *Details*: 2 s square current injections to rheobase + 40 pA and 80 pA. Additional details can be found in the Biophysical Models - perisomatic Technical White Paper.
Square 0.5 ms Subthreshold	*Description*: Brief subthreshold square current injections used to determine membrane capacitance for biophysical models. *Details*: 0.5 ms square current injections to +/- 200 pA. Additional details can be found in the Biophysical Models - perisomatic Technical White Paper.

Figure 7: Depicts the type of stimuli used on each cell to invoke responses, as well as provides a short description of the stimuli's properties. Source: This diagram comes directly from the Allen Cell Types Database Electrophysiology Technical Whitepaper.

Importance of Creating a Classification Model on This Data Set:

As mentioned in the introduction, I am curious to know the feasibility of classifying a neuron based on whether it is Aspiny or spiny dendrites given its electrophysiological properties. But is this a valuable question to explore?

The simple answer is absolutely. Based on a discussion held with a fellow peer, who actually took part in the data acquisition process of this dataset, being able to determine the morphological features of a neuron based on its electrical response could greatly reduce the size of the data and the amount of computation required to analyze 2d neuronal images. Not to mention, you could determine the physical attributes without having to image and trace the cell (Neuron tracing is a time-consuming reconstruction technique for mapping the axons and dendrites, as well as other physical features of a neuron).

The dataset itself is very robust. For each cell experiment there are around 100 sweeps. Also, for each sweep there are numerous characteristics that can be observed within either the single action potential response **(Figure 8)**, or spike train response **(Figure 9**), depending on stimulus.

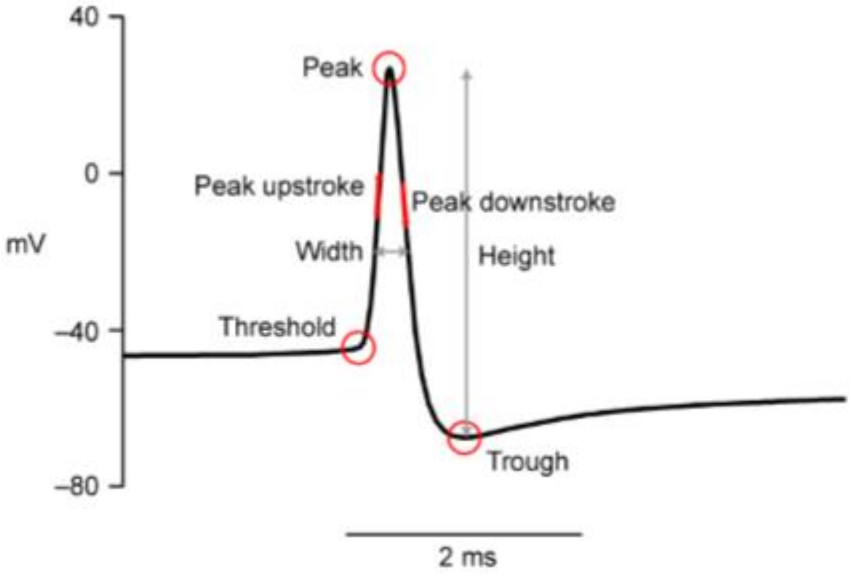


Figure 8: shows the features of single action potentional response. The peak is the max voltage from the which the neuron membrane potential begins to repolarize back to it’s resting state. The other features such as the peak downstroke and upstroke are later described. For further information about these features, visit the Source: This diagram comes directly from the Allen Cell Types Database Electrophysiology Technical Whitepaper.

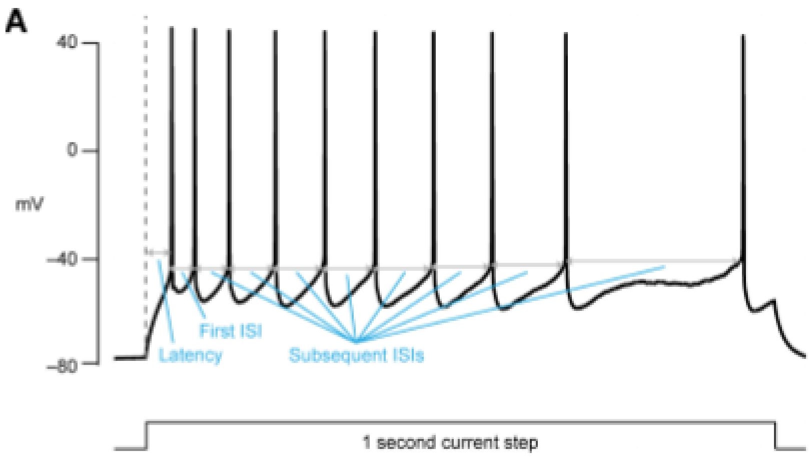


Figure 8: A sweep illustrating features found within a spike train: Interspike Interval (First ISI), the Latency, etc. Source: This diagram comes directly from the Allen Cell Types Database Electrophysiology Technical Whitepaper.

On top of having single cell data, there are over 50 features available across the ephysiological recordings for each cell experiment, that come from averages across the sweeps. Therefore, in terms of complexity, this dataset provides a large playground to experiment with many principles of data science, such as preprocessing and feature extraction which will ultimately lead to modeling and evaluating classifiers.

Exploring the Dataset:

Diving into the Allen Cell Types Database via the sdk and api is a relatively straightforward. The code is documented using sphinx—a python documentation tool; therefore, if you can figure out what you are looking for, you will find it.

From the beginning, I decided to reduce any variability in the data by solely focusing on recordings from Mus musculus—the common house mouse.

In the dataset, there are a total of 1920 instances of mouse cells found. With each cell containing metadata describing features such as the exact 3-dimensional coordinates of the specimen, the transgenic mouse line, dendrite type, cell id, etc.

In my first attempt at parsing the data, I used a single specimen's id value, and collected the ephysiological data from the experiment involving the specified cell. This came in the form of a NwbDataSet object, which is a class object that contains methods for retrieving stimulus and trace responses across all sweeps. I then observed the cell responses to the different stimuli, as shown in **Figure 3** and **Figure 4**.

For a single sweep, the sdk/api has an EphysSweepFeatureExtractor class that provides access to the feature values computed from the cell's response. These values include the trough, peak voltage, threshold voltage, and around 33 other features.

Conveniently, the sdk provides a function to get the recordings for all cells, where the feature values are computed as the average across all sweeps within a cell experiment. I constructed a dataframe from this, and connected a column containing whether the specified cell was spiny or Aspiny. I realized there was also a third category in the data—sparsely spiny, which as the name suggest, is a neuron that has dendrite with projects and without. For the early stages in data exploration, I did not remove the cells that were Aspiny.

With a dataframe at hand, I originally thought to try the clustering approach used with the Iris dataset earlier in the semester. Clustering would enable me to distinguish a set of features that could separate out the three classes: spiny, Aspiny, and sparsely spiny. However, because of the number of features, creating a cluster matrix would generate near 2500 different plots. This was not feasible. Instead, I randomly selected a few features and generated multiple cluster matrices as shown below in **Figure 9**.

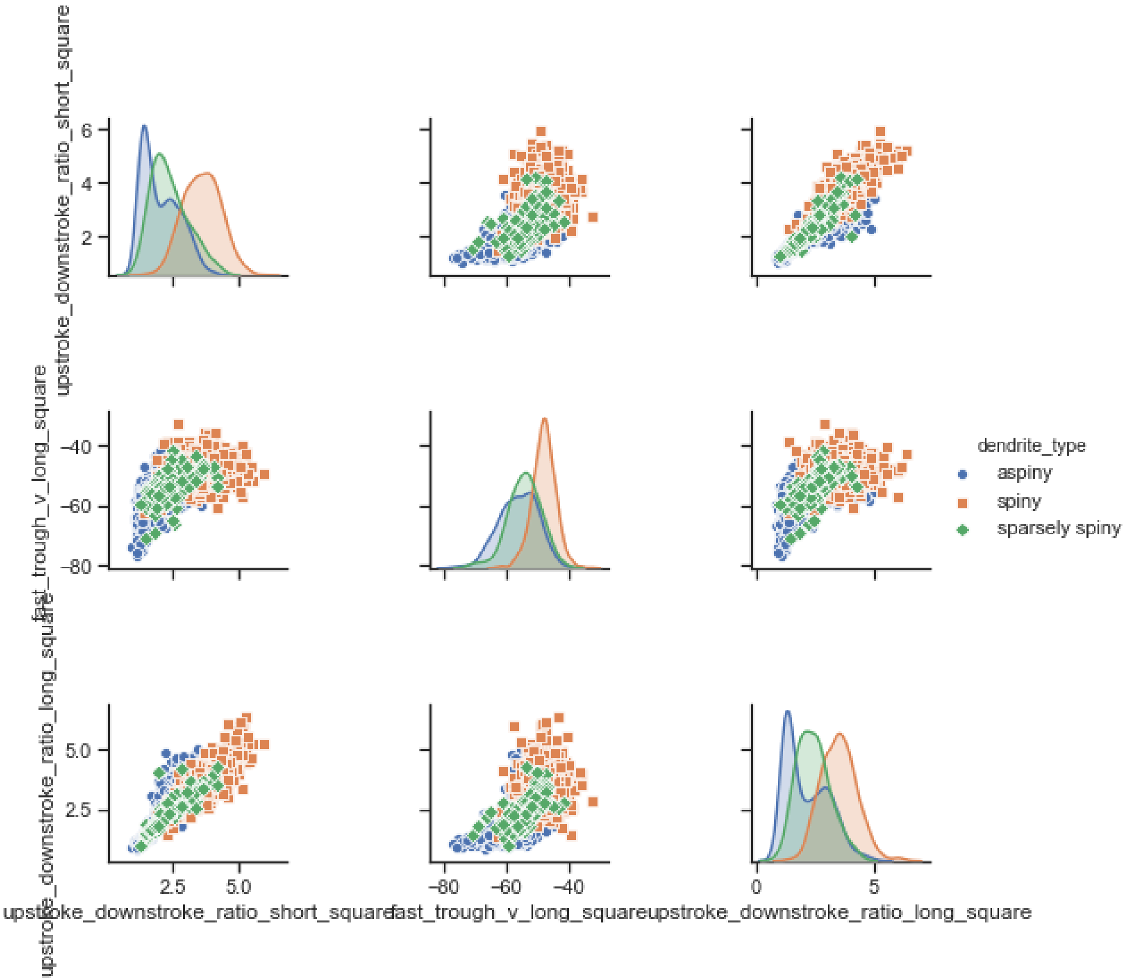


Figure 9: A cluster matrix generated from the following features: upstroke-downstroke ratio short-square, fast-trough (v) long-square, and upstroke downstroke ratio long square. It aims to see if there is a mapping that best separates the data into three distinct clusters.

Following that, I noticed that the fast trough depth and upstroke-downstroke ratio provided the best distinction between the spiny and Aspiny categories, with the sparsely spiny neuron being scattered between the two **(Figure 10)**. This feature distinction matched with an example Jupyter notebook I came across in the Allen Cell Types sdk. With a solid footing on how to work with the dataset, I then turned my focus on how I would train classifiers to predict whether a dendrite type was spiny or Aspiny.

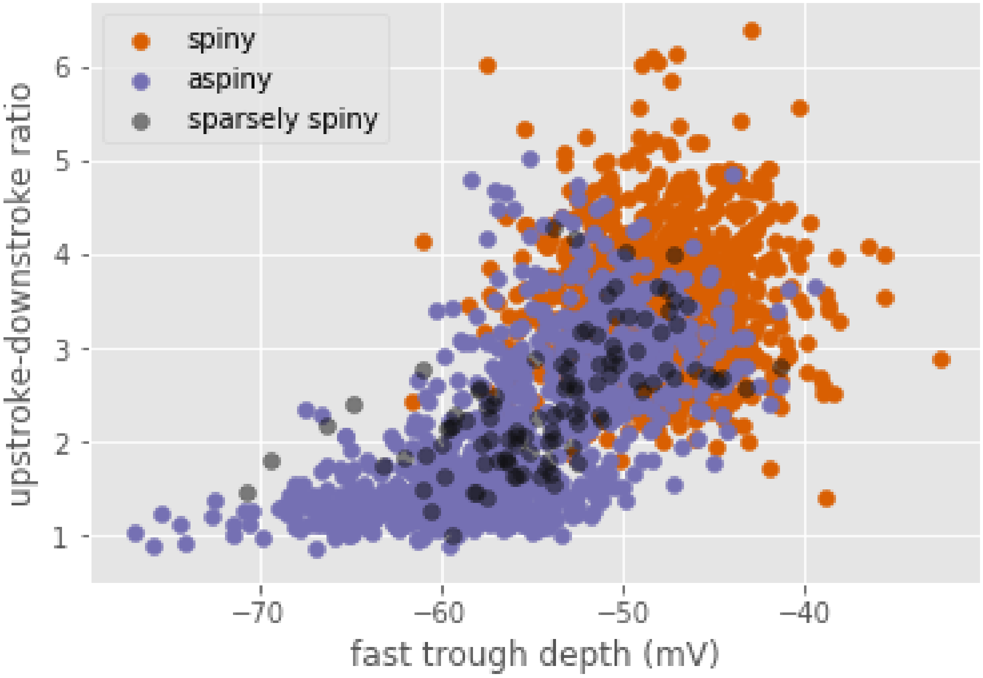


Figure 10: a scatter plot generated between the fast trough depth vs upstroke-downstroke ratio. The orange, purple, and grey points indicate spiny, Aspiny, and sparely spiny neurons, respectively.

My Approach to Selecting Classification Models to Fit This Dataset:

Classification algorithms are a form of supervised learning. The main idea is to train a classification model (i.e. generate a target function), that maps each instance of a feature set X to a class label y.

In my CS7180 course (Special topics in AI) we discussed two types of classification models—descriptive and predictive modeling. Descriptive modeling serves as a tool to distinguish between objects of different classes by describing the features that make up an instance of a class; whereas, Predictive modeling uses an instance of the features as input to predict the class label.

For the scope of this project, I decided to focus on predictive modeling. The largest factor being that, this project must be delivered as a web-service application. Therefore, if a user has a set of feature values from an electrophysiological recording, they would be able to plug in those values, and have my models predict the likelihood of the neuron being spiny or Aspiny.

In choosing models for predictive modeling, I trained a probabilistic predictive classification model (i.e. Logistic Regression), a decision tree, and an ensemble model known as a Random Forest classifier.

Training & Evaluating Classification Models:

Preprocessing:

Before training my classifiers I focused on reducing the feature set from 50 down to 24. I realized there were columns not indicative of the cell's electrophysiology traits, for example, the peak time ramp. Column based on the time a stimulus was elicited on a cell. A cells response is not affected by the time a stimulus is used. A full list of the features dropped Is in the finalized Jupyter notebook.

Also, columns missing more than 250 values were removed. Rows with features missing were dropped afterwards as to minimize the total number of instances lost. Lastly, sparsely spiny cells were dropped. As a result, there were a total of 1566 instances left in my dataset.

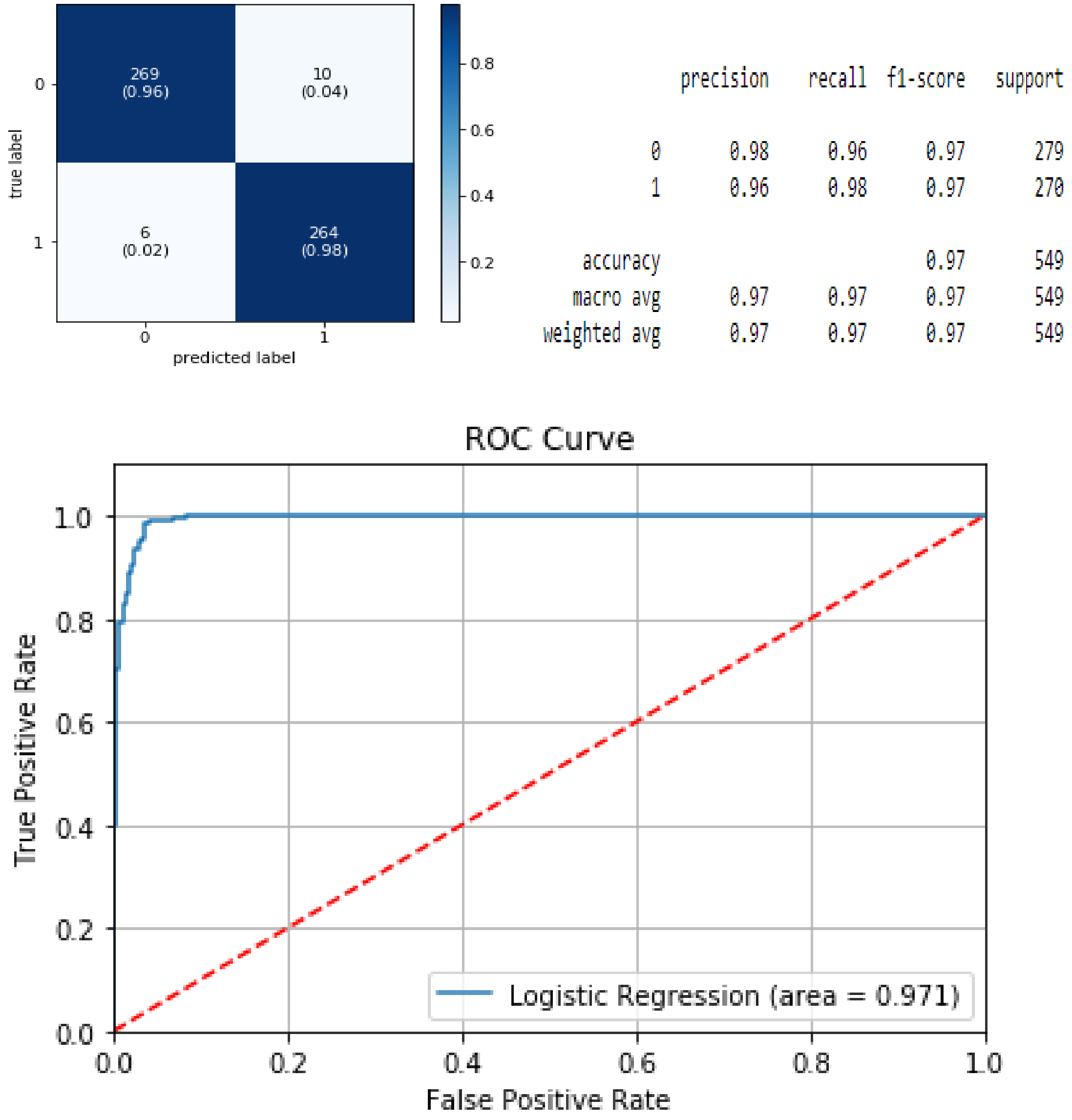
Training and Evaluating:

For Training I split the data 65% training and 35% testing. Also, for each model I used the following evaluation metrics:

- Accuracy score – a ratio of the number of correct predictions.
- Confusion matrix – a table which describes a classifier's performance based on true values.
- Precision – a ratio equal to the # true positives / (# true positives + # false positives).
- Recall – a ratio equal to # true positives / (# true positives + # false negatives).
- F1-score – The harmonic mean of precision and recall values.
- ROC curve – (receiver operating characteristic) plot of true positive rate against the false positive rate.
- AUC – Area under the ROC curve.

Logistic Regression:

Logistic Regression is a common method in data science for handling binary classification problems. In the case of my data set, either a cell can be spiny or Aspiny. It will use a feature set to produce a mapping to one of the two classes. Since it is probabilistic, it computes a probability that one of the two classes are likely. Once the logistic regression classifier was fit to the test set, Using the metrics above its evaluation is shown in **Figure 11** below.



```
              precision    recall  f1-score   support

           0       0.98      0.96      0.97       279
           1       0.96      0.98      0.97       270

    accuracy                           0.97       549
   macro avg       0.97      0.97      0.97       549
weighted avg       0.97      0.97      0.97       549
```

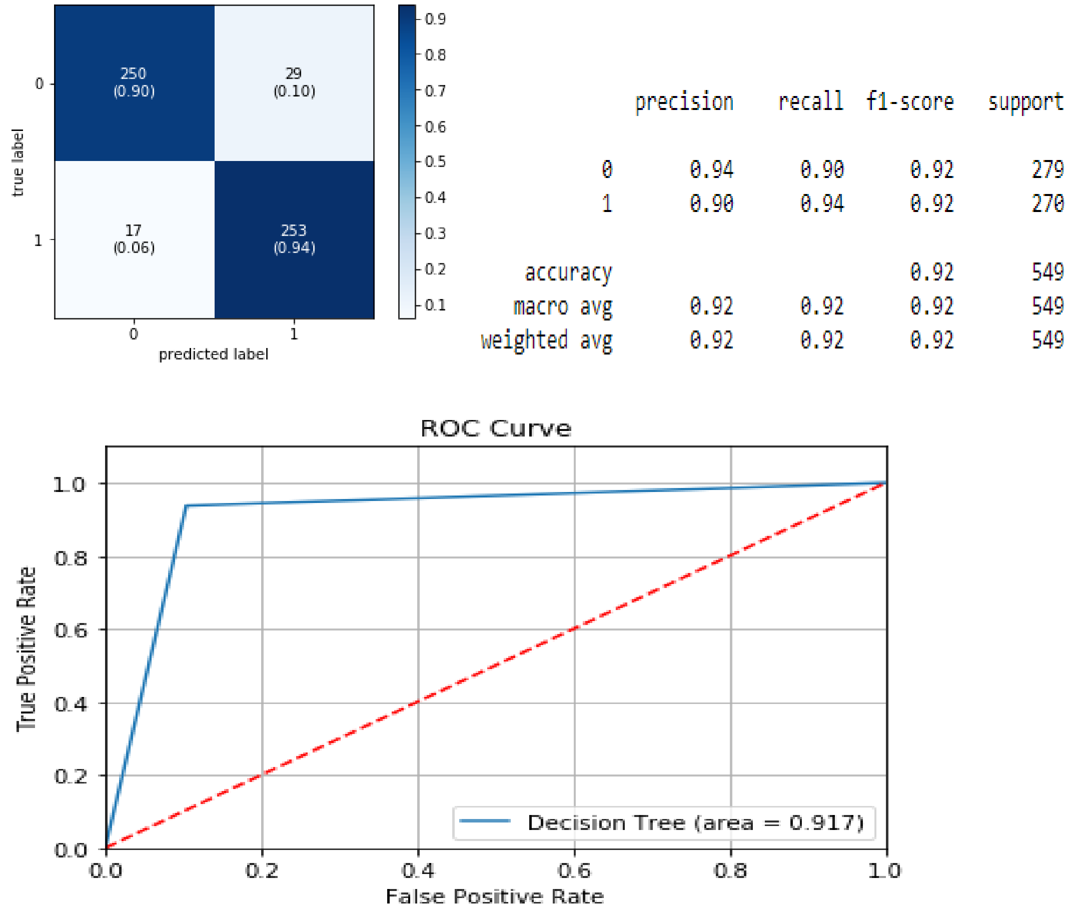
Figure 11: Depicts the evaluation metrics of the Logistic Regression classifier. top left shows the confusion matrix. Top left shows the precision, recall, f1-score, and support. The bottom panel shows the ROC and AUC.

We see that in total 533 (296 are spiny and 264 are Aspiny) instances are correctly mapped to their correct label (shown in the top left confusion matrix). From the classification report (top right) 97% of the cells are precisely mapped, and from the ROC and AUC we have a curve close to the top left of the plot, which means the classifier is not purely random.

Decision Tree:

A decision tree is another method used in classification modeling. Unlike the Logistic Regression, it is not probabilistic. Rather it is composed of branches that represent a decision rule and a leaf node with represent a potential outcome. It randomly breaks the dataset into subsets until it has exhausted the instances.

From fitting the decision tree classifier to my dataset, the model performed 92% accurately on the test set. The resulting evaluation metrics are seen in **Figure 12.**



```
              precision    recall  f1-score   support

           0       0.94      0.90      0.92       279
           1       0.90      0.94      0.92       270

    accuracy                           0.92       549
   macro avg       0.92      0.92      0.92       549
weighted avg       0.92      0.92      0.92       549
```

Figure 12: Depicts the evaluation metrics of the decision tree classifier. top left shows the confusion matrix. Top left shows the precision, recall, f1-score, and support. The bottom panel shows the ROC and AUC.

We see that in total 503 (250 are spiny and 253 are Aspiny) instances are correctly mapped to their correct label (shown in the top left confusion matrix). From the classification report (top right) 92% of the cells are precisely mapped, and from the ROC and AUC we have a curve close to the top left of the plot, which means the classifier is not purely random

Random Forest:

Random Forest is a type of Ensemble learning method used for predictive classification. Given a dataset, is selects n random samples, constructs n decision tree for each sample, then it does a vote on a prediction.

The random Forest classifier received an accuracy score of 96% on the test set. However, prior to performing an evaluation, I was able to compute the feature importance on this model **(Figure 13)**. This allowed me to reduce the number of features down to the four that had the largest impact on the model's prediction performance. I kept the features that scored above a 0.08.

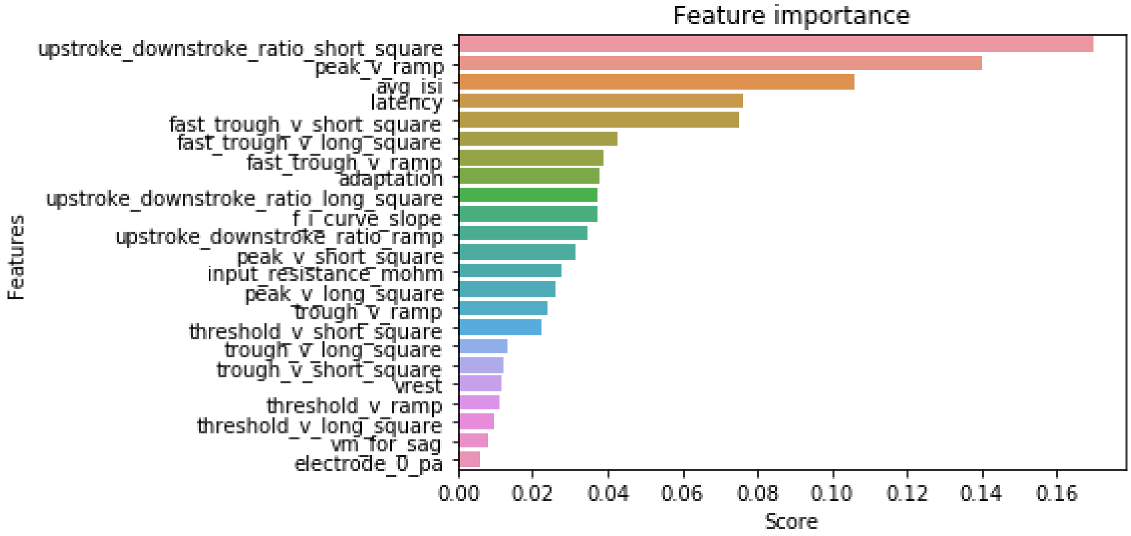
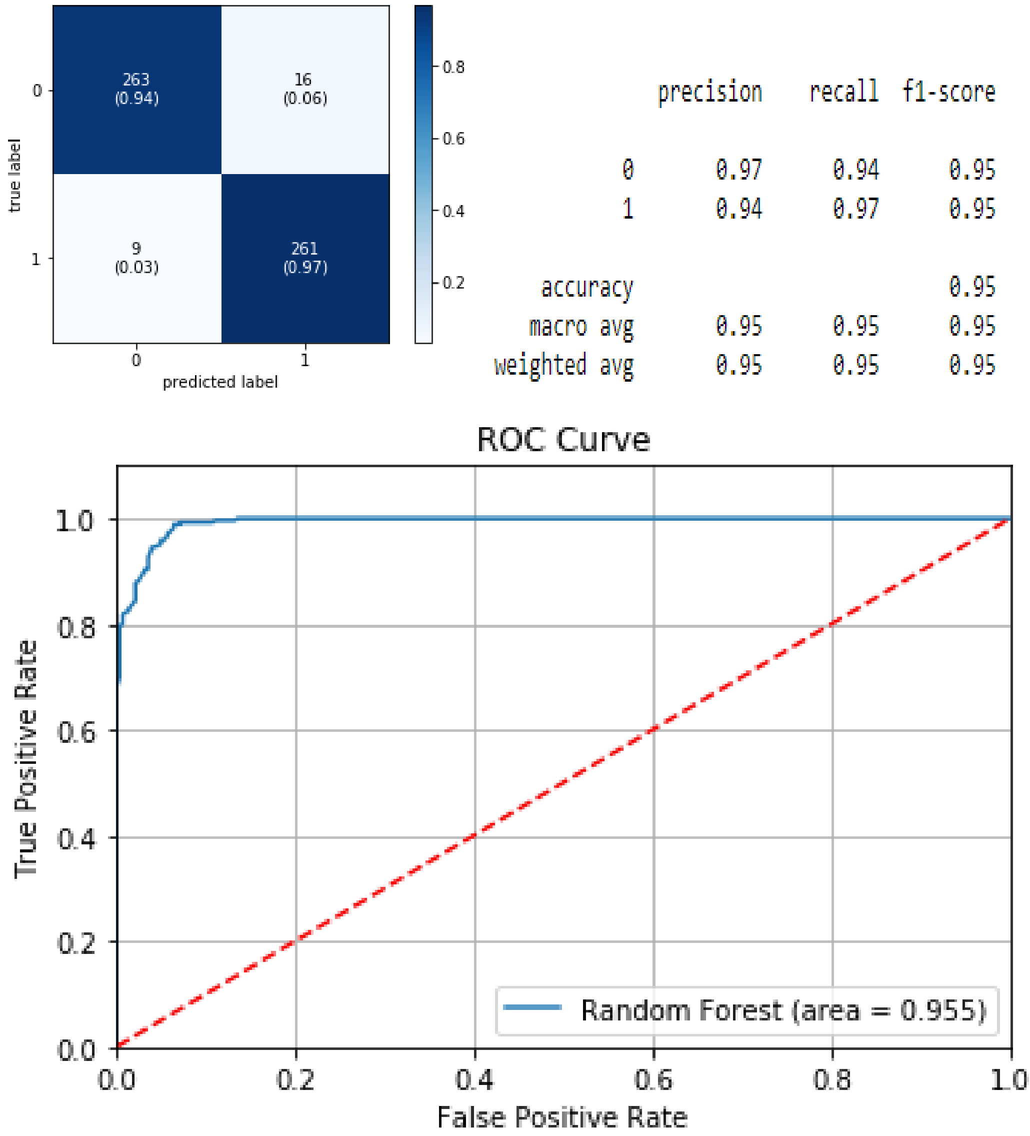


Figure 13: depicts the importance of features in ascending order.

The updated model had an accuracy of 95%. Since the performance did not drastically decrease and the number of features was decreased down to four (reducing complexity), I decided to stick with the updated model. The evaluation metrics are depicted in **Figure 14**.



```
              precision    recall  f1-score   support

           0       0.97      0.94      0.95       279
           1       0.94      0.97      0.95       270

    accuracy                           0.95       549
   macro avg       0.95      0.95      0.95       549
weighted avg       0.95      0.95      0.95       549
```

Figure 14: Depicts the evaluation metrics of the random forest classifier. top left shows the confusion matrix. Top left shows the precision, recall, f1-score, and support. The bottom panel shows the ROC and AUC

We see that in total 524 (263 are spiny and 261 are Aspiny) instances are correctly mapped to their correct label (shown in the top left confusion matrix). From the classification report (top right) 95% of the cells are precisely mapped, and from the ROC and AUC we have a curve close to the top left of the plot, which means the classifier is not purely random

Comparison:

All three models performed accurately and showed positive results on the evaluation metrics. Each model had a high AUC value, and the fact that each ROC curve was close to the top left is a strong indicator the classifiers did not make purely random predictions. The Logistic regression had the best accuracy, predicting 97% of instances correctly; whereas, the decision tress classifier made the least accurate predictions, 92% accuracy. However, the Random Forest classifier held a 95% accuracy score despite being trained on 4 features instead of 23.

Conclusion:

To conclude, building predictive classification models for biological datasets could have the potential to revolutionize the rate of scientific discoveries. The field of neuroscience is a prime target for data science, since it contains large volumes of data that contain noise and unnecessary variables. This paper illustrates that given electrophysiological data, it may be highly possible to correctly predict whether a neuron has Aspiny or spiny dendrites. To further the work done for this project, I hope to delve further into how some of the neuronal electrophysiological features in my model relate; and answer the question of which features are truly dependent vs independent of each other from a biological perspective.