# Modeling the Human Haptic Code: A Deep Learning Approach to Neuro-Prosthetic Development

Cleah Winston

# December 11

Cleah Winston

# Experiment Recap

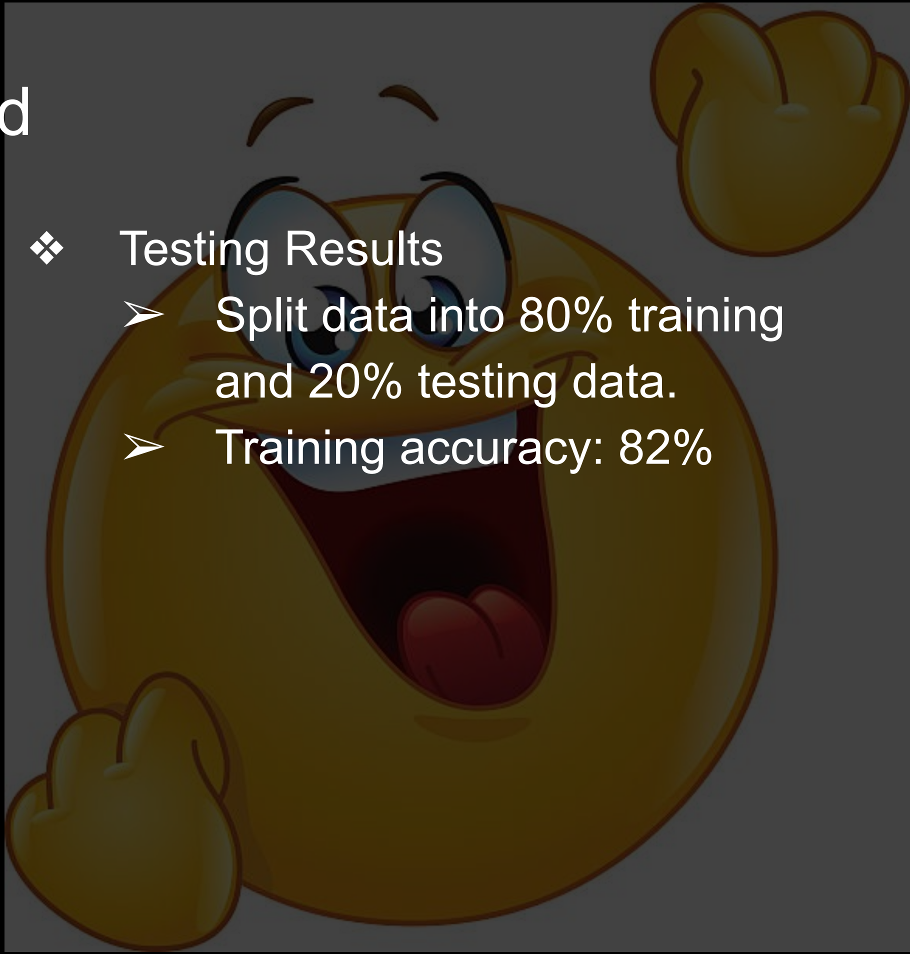

- Participant Number = 2
- Collected data for touch, vision, and touch-vision on 4 pairs of objects
- The above diagram shows the setup for an experiment for a single pair of objects
- Starting with touch hot/cold for model

# Model Pipeline

1. **Compile:** Concatenated the hot/cold data from participants.
2. **Label:** Added a binary column which contained -1 if the participant was not touching any object, 0 if they were touching hot object, and 1 if they were touching cold object.
3. **Remove unnecessary data:** Deleted empty rows and unnecessary columns.
4. **Normalize Data:** Normalized features (Alpha, Beta, Delta, Gamma, Theta on AF_7, AF_8, TP_9, TP_10)
5. **Train Model:** Trained logistic regression model on these features.

# Logistic Regression on Hot/Cold

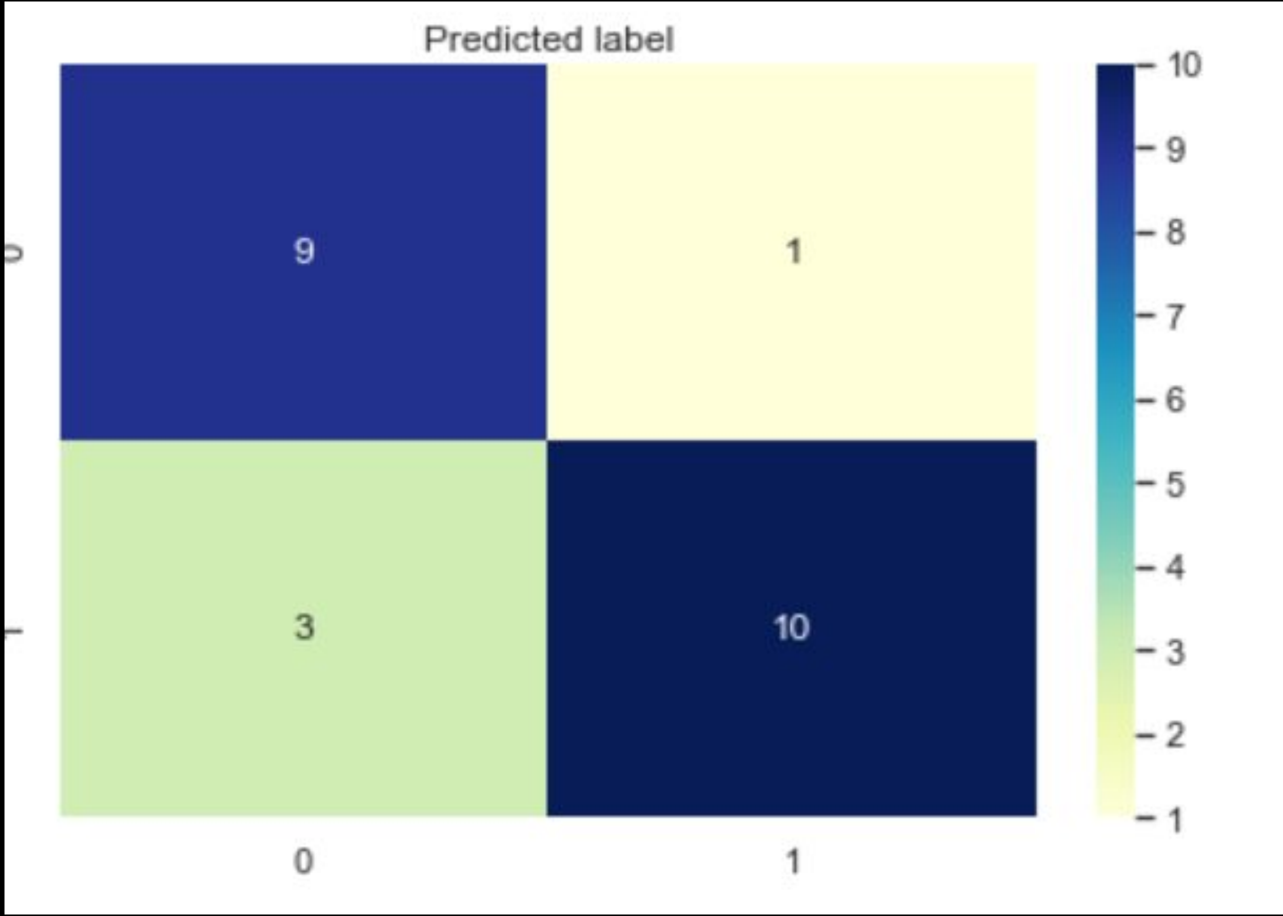
- Preliminary Results
  - This model was trained on all the data (no splitting.)
  - Training accuracy: 91%



- Testing Results
  - Split data into 80% training and 20% testing data.
  - Training accuracy: 82%

# Confusion Matrix for Logistic Regression

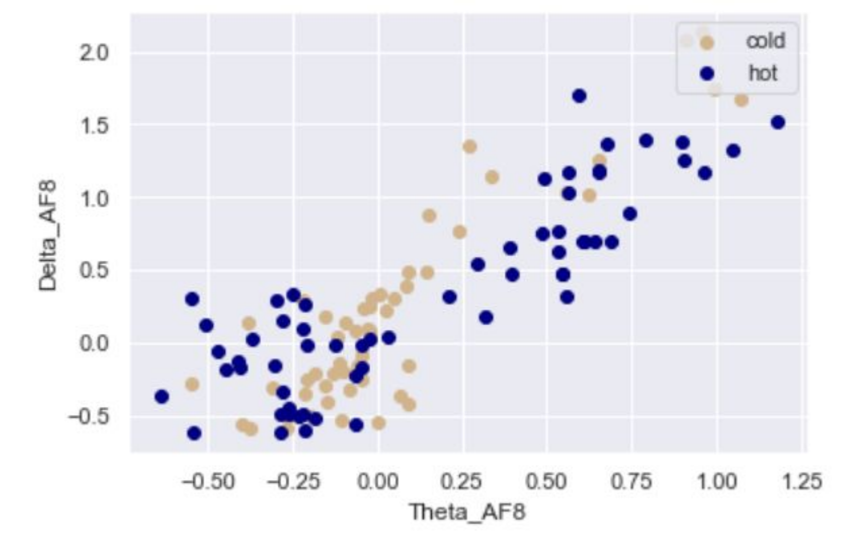
- I created a confusion matrix using results from the model trained on the testing data.
- I think this confusion matrix is good because the model predicted the correct binary number most of time!



Predicted label
9
1
3
10
0
1
0
1
10
9
8
7
6
5
4
3
2
1

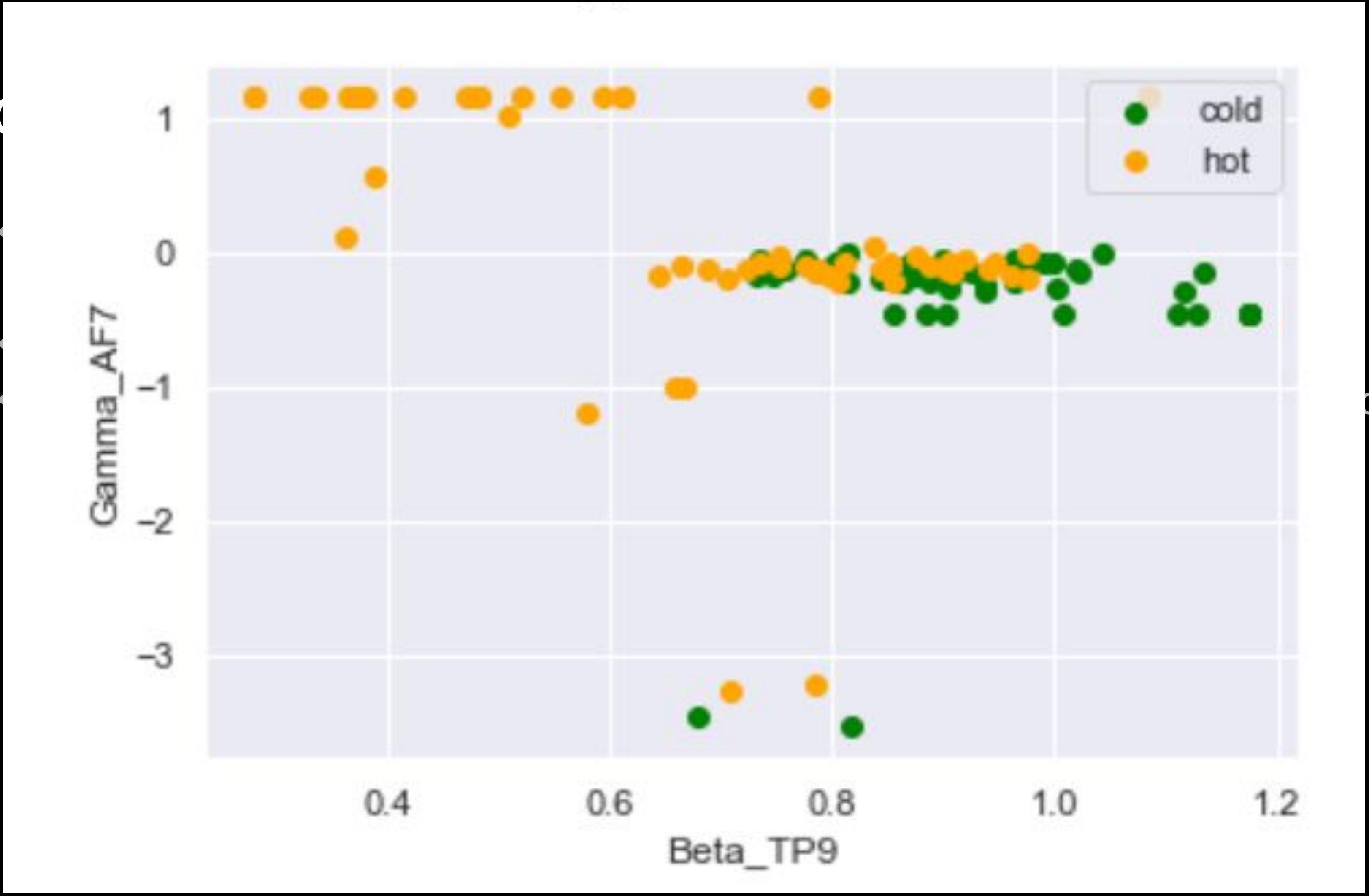
# Theta_AF8 vs. Gamma_AF8 Plot

- ❖ I created a plot using two random frequency bands.
- ❖ Here is the plot:
- ❖ I created this plot to visualize how the activity during hot and cold differed.



# Gamma_AF7 vs. Beta_TP9 Plot

- I used recursive feature elimination and found the two most significant features.
- Here is the plot:
- I created this plot to visualize how the activity during hot and cold differed.


cold
hot
Gamma_AF7
1
0
−1
−2
−3
0.4
0.6
0.8
1.0
1.2
Beta_TP9

# Potential Next Steps

- Collect more data
- Do principal component analysis (I am currently doing that!)
- Use other objects
- Use different models
- Try vision or vision-touch
- Analyze different people and objects
- Model power bands given object

# Questions

- ❖ Currently, my model is predicting a binary number based on the EEG data. How could I make the model predict the EEG data based on they binary numbers?
- ❖ Are there other visualizations or analysis I should do on the data?
- ❖ Are there any other models that I should make?
- ❖ What should I do next?

# January 4

Cleah Winston

# Purpose of Project

- Generating EEG signals that would produce touch sensation based on the response to vision (just vision to touch vision) and/or based on the tactile labels(eg., hot or cold) -----> This could be then used to build touch neuroprosthetics for individuals who cannot sense touch but can see!
- Predict what someone ??? using imaginary touch (imagining what you want to touch) ----> This could be used to build prosthetics/devices for individuals who can touch but can not speak/move

# Purpose of Project

- Generating EEG signals that would produce touch sensation based on the response to vision (just vision to touch vision) and/or based on the tactile labels(eg., hot or cold) -----> This could be then used to build touch neuroprosthetics for individuals who cannot sense touch but can see!
- Predict what someone ??? using imaginary touch (imagining what you want to touch) ----> This could be used to build prosthetics/devices for individuals who can touch but can not speak/move

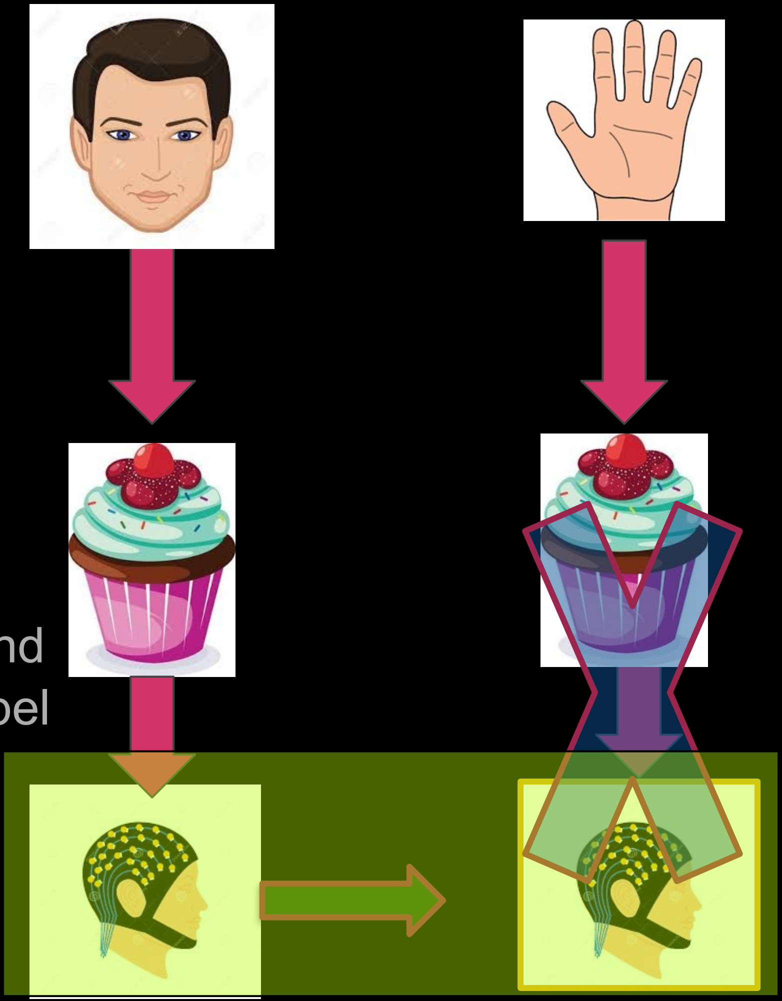
# My Progress

- Collected data for 4 participants for imaginary touch, touch vision, and touch (eyes close) of objects with various tactile features
- Created binary classifiers
  - for opposing touch types (hot vs. cold) when sensed by touch, vision, and touch+vision - Tested multiple machine learning models, reasonably high accuracies
  - Visualized data with PCA and tSNE
  - imaginary touch vs. touch vision had high accuracy meaning much separation

# Touch Vision Mapping

# Touch Vision Mapping

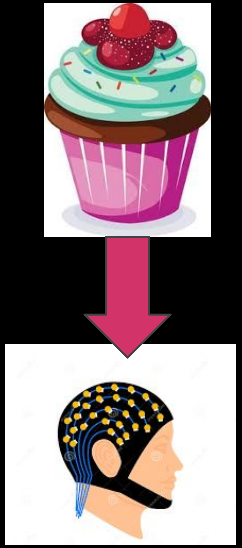
- Green Arrow is what I am doing
- Neural networks (multilayer and recurrent) to the rescue!
- Freq. bands for vision data to freq. bands of the touch vision data
- Correlatory analysis to also understand relationship -- do I need the tactile label as well? (hopefully not!)
- This would then be used in a prosth.



# Generating EEG signals from the tactile labels

# Variational Autoencoder

- Generates EEG signals or specific features of EEG signals (i.e., frequency bands) given tactile labels
- I have to research and then implement this



Can we use the same prosthetic for everyone?

# Transfer Learning

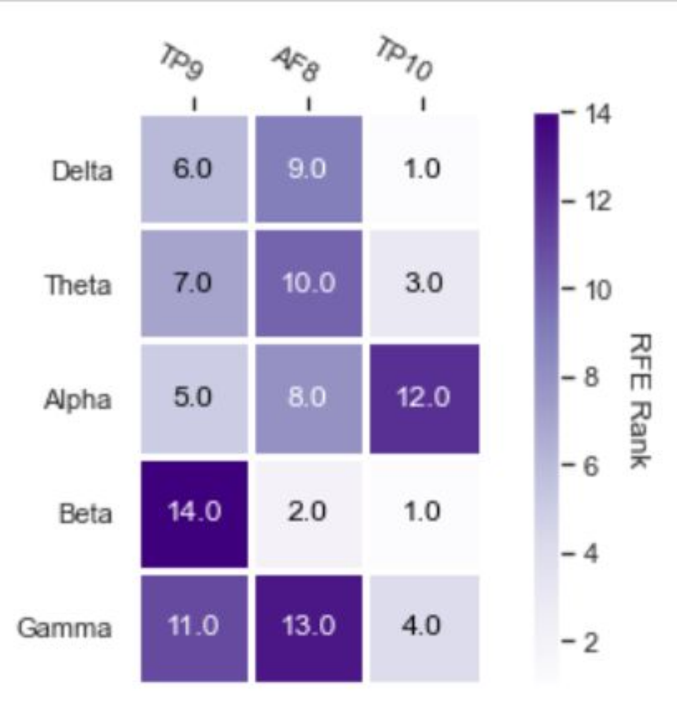
- Train the model on 1 person and test it on another person
- See if training the model partially on 1 person and then continuing to train on a different person speeds up training
- This would elucidate whether models learned on individuals with touch or just other individuals could be useful for creating neuroprosthetics for other individuals without touch sensation

# February 2

Cleah Winston

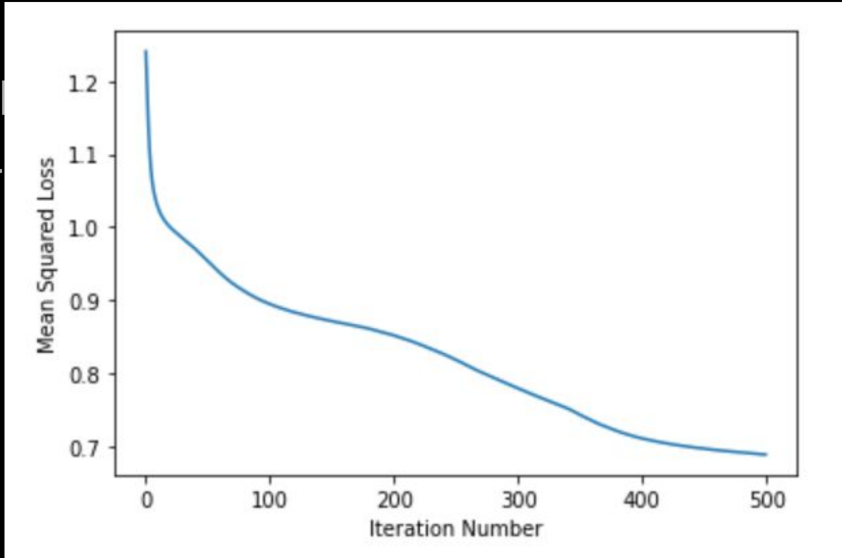
# Recursive Feature Elimination Heatmap

- I performed recursive feature elimination on my features for touch hot vs. cold.
- I then got a ranking for them where 1 means most important.
- From this, I think TP10 is the most important.
- This supports the involvement of the parietal lobe i representing touch.
- TP10 is between the temporal and parietal lobe.



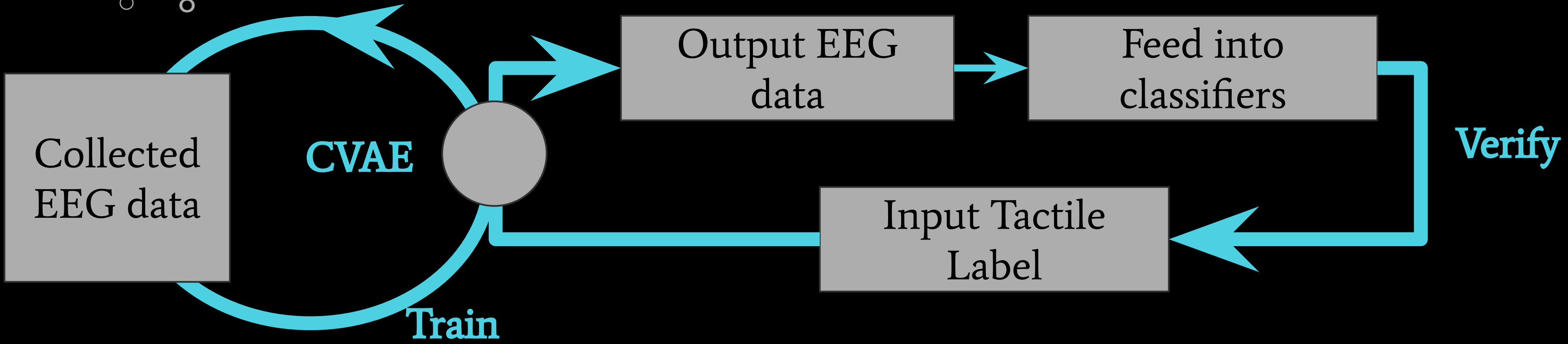
# Touch Vision Mapping

- I created a multilayer perceptron which predicts the EEG response to touch-vision data given the response to vision.
- I used all the data for all objects and participants.
- Layers: (10, 5, 10)
- Learning Rate = 0.1
- Solver = Stochastic Gradient Descent (SG
- Mean Squared Error = 0.73343066806644

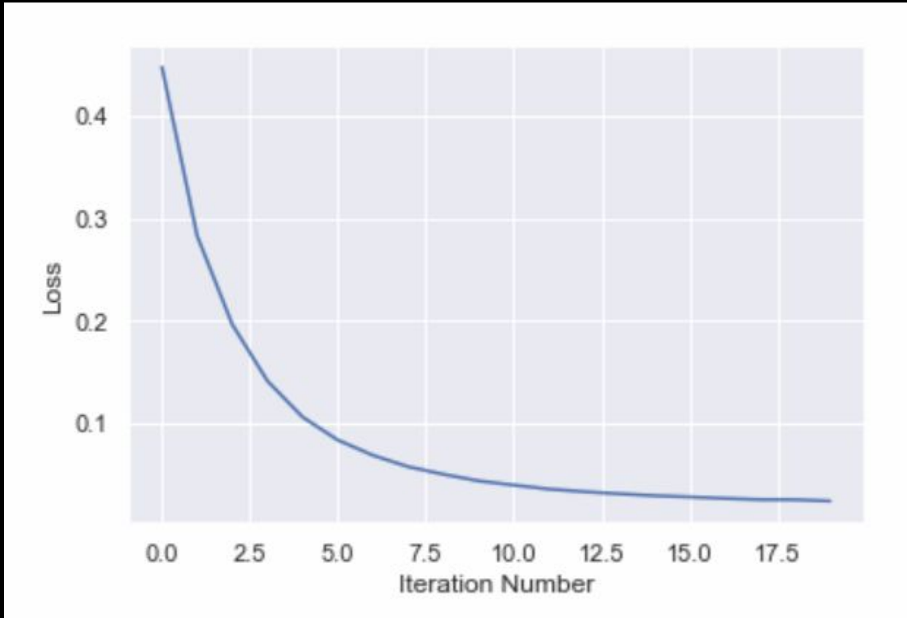


# Conditional Variational AutoEncoder

- I created a Conditional Variational AutoEncoder(CVAE) that generates a sample of EEG data for touch on a given label.
- Layer-Size
  - Encoder: [15, 10, 5]
  - Decoder: [5, 10, 15]
- Latent Space
  - 8

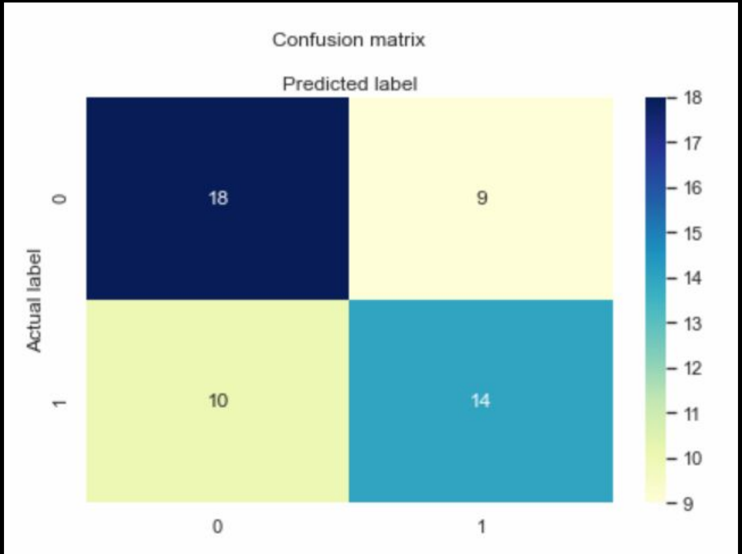


# Conditional Variational AutoEncoder Results

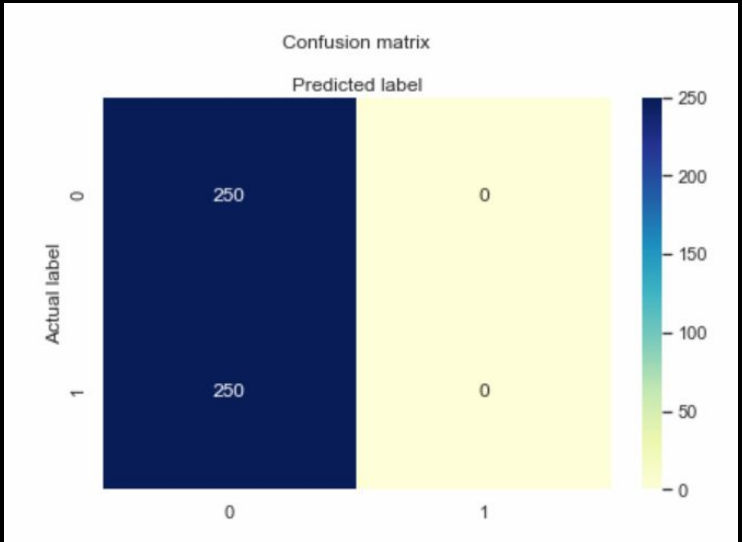
- Loss Curve
  - Trained on collected data on 20 epochs and found that the loss decreased! :)
  - I used mean squared error and Kullback–Leibler divergence
- 

# Conditional Variational AutoEncoder Results (cont.)

- This is the confusion matrix for a multilayer perceptron for binary classification(hot vs. cold) without using CVAE.
  - Trained on collected EEG data
  - Tested on collected EEG Data

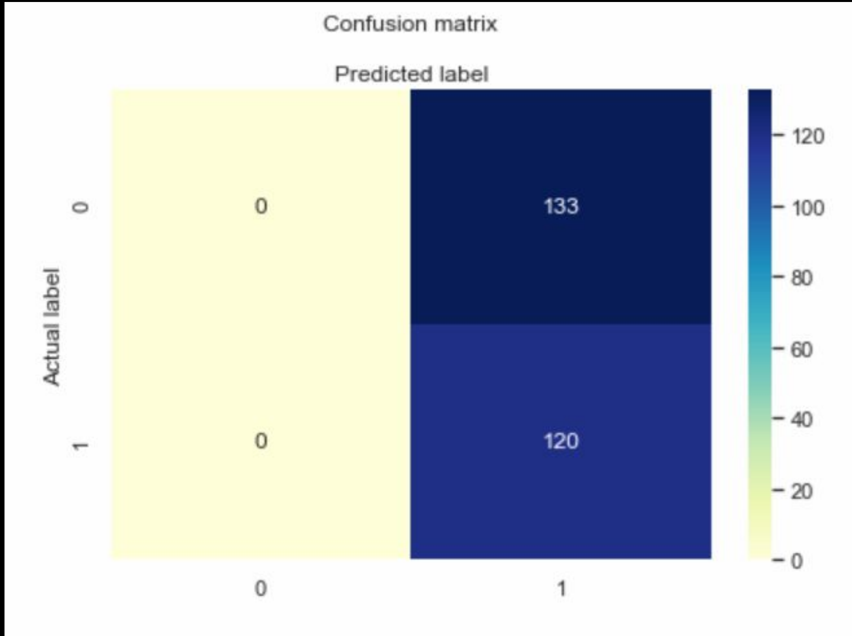



- This is the confusion matrix for a multilayer perceptron for binary classification(hot vs. cold).
  - Trained on collected EEG data
  - Tested on collected CVAE output data



# Conditional Variational AutoEncoder Results (cont.)

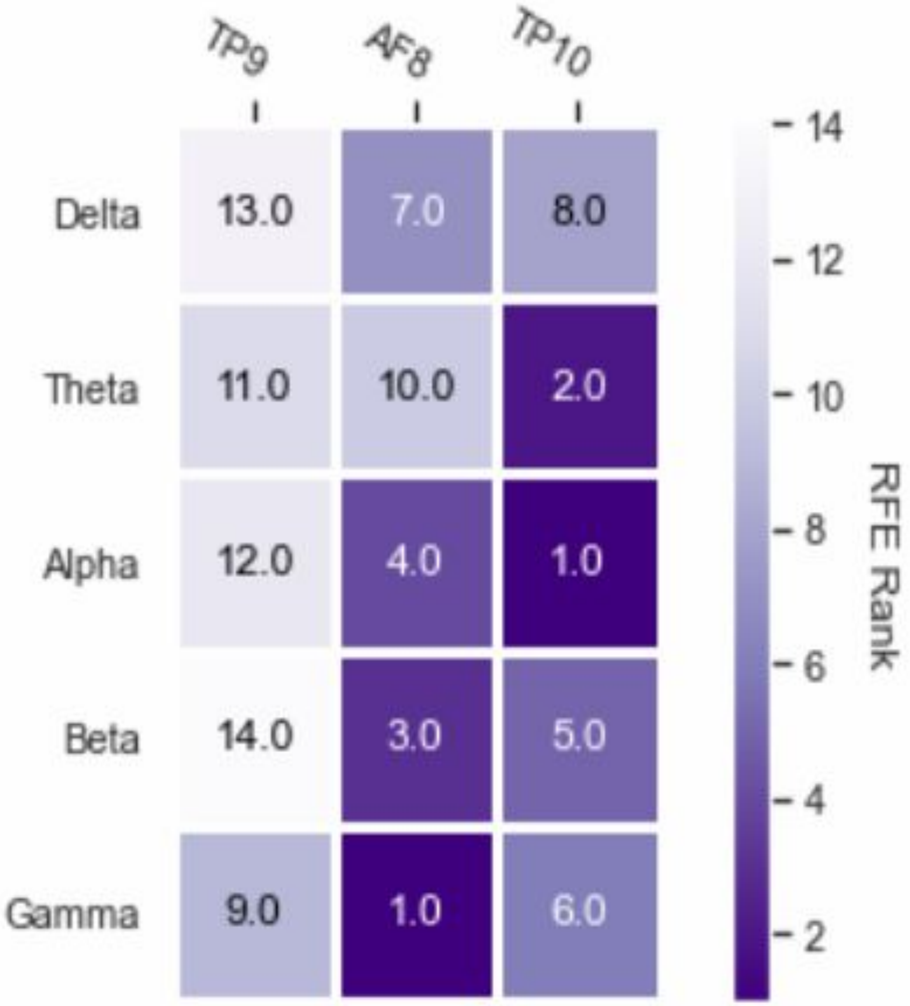
- The CVAE is always generating data that is similar to hot(0) data.
- I am not sure why this is happening.
- I also created confusion matrices for when:
    - Trained on CVAE output
    - Tested on Real Data



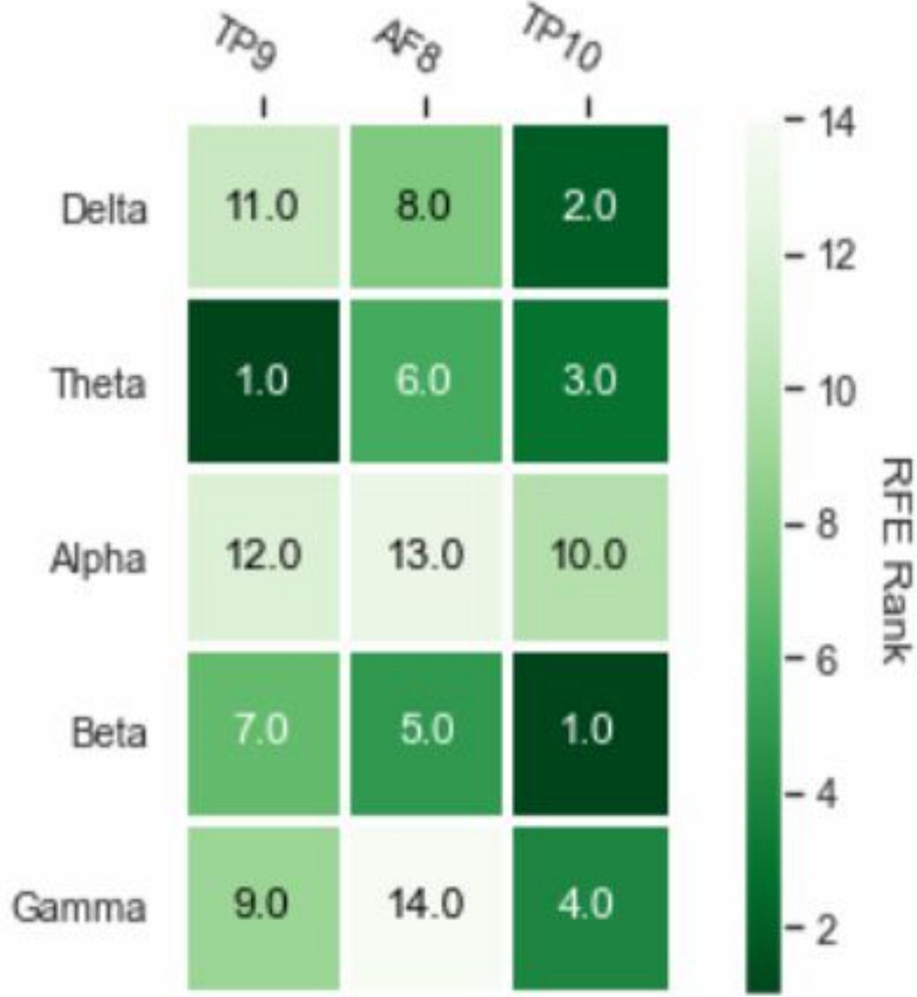
# February 10

Cleah Winston

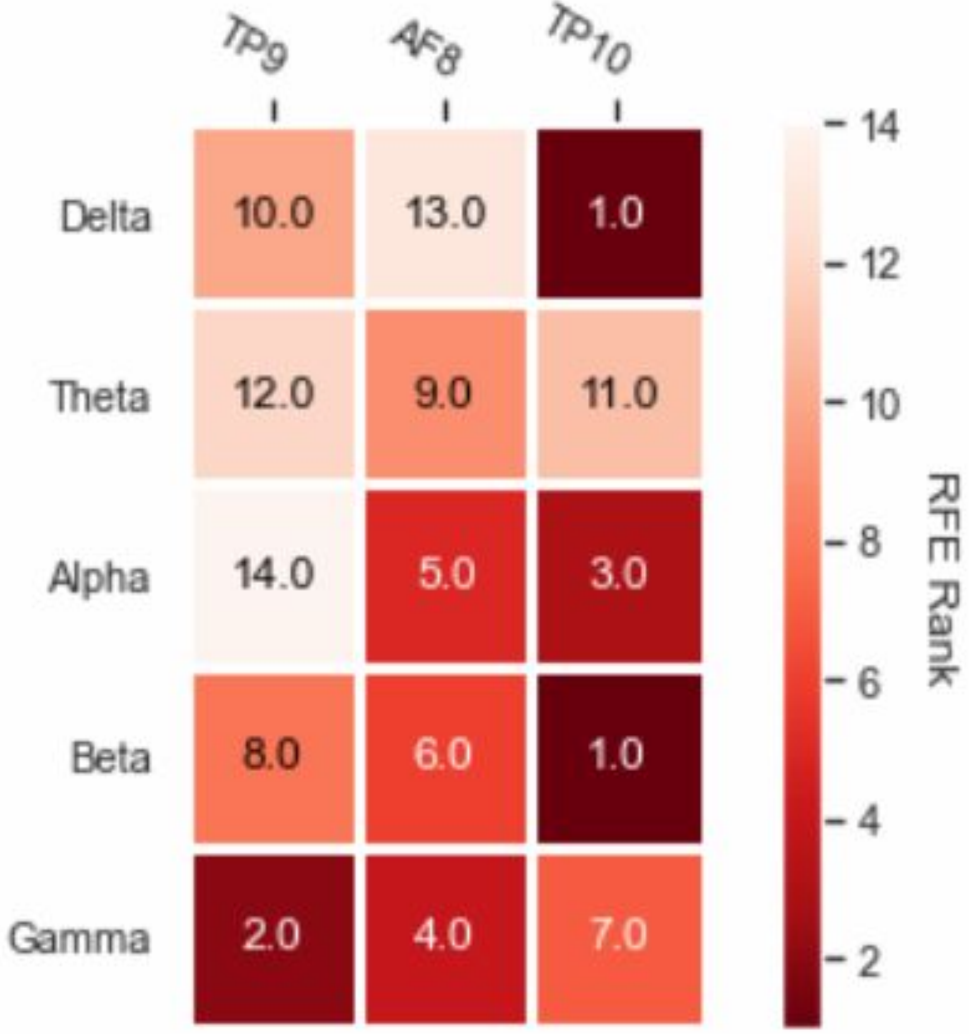
# Hot: Touch vs. Vision



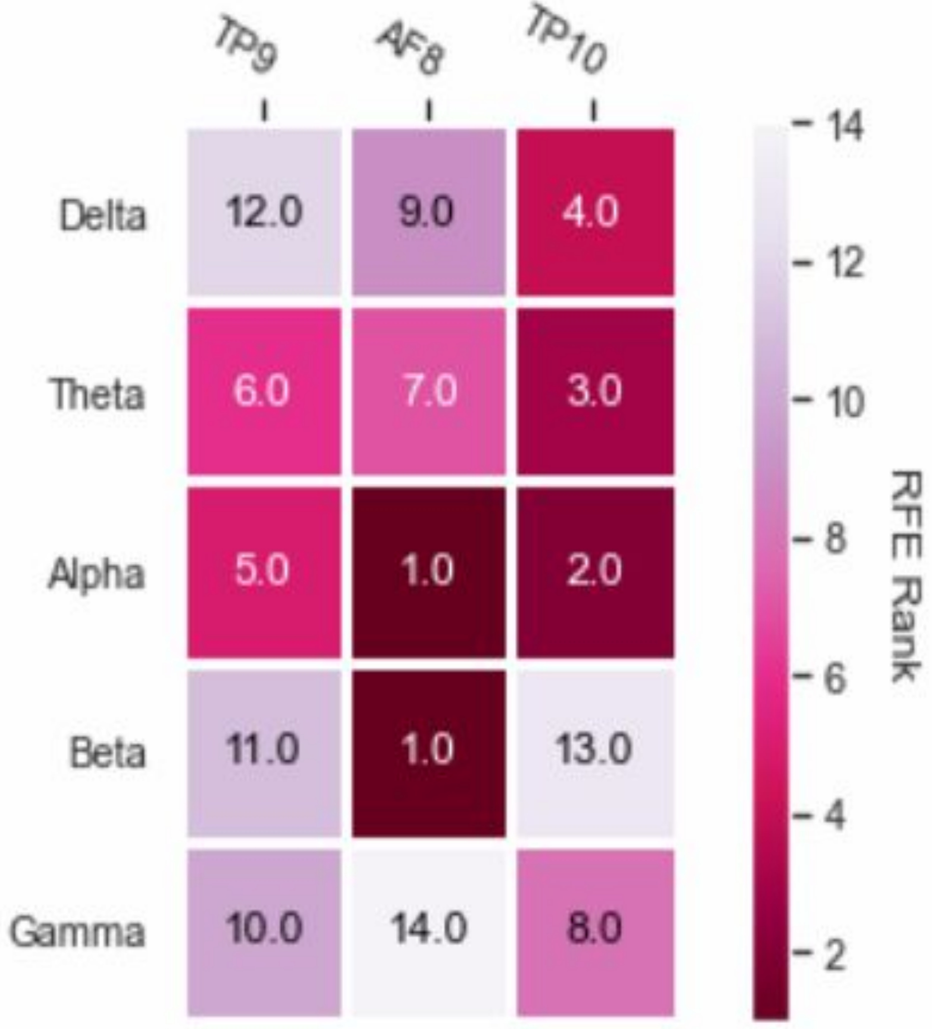
# Hot: Vision vs. Touch-Vision



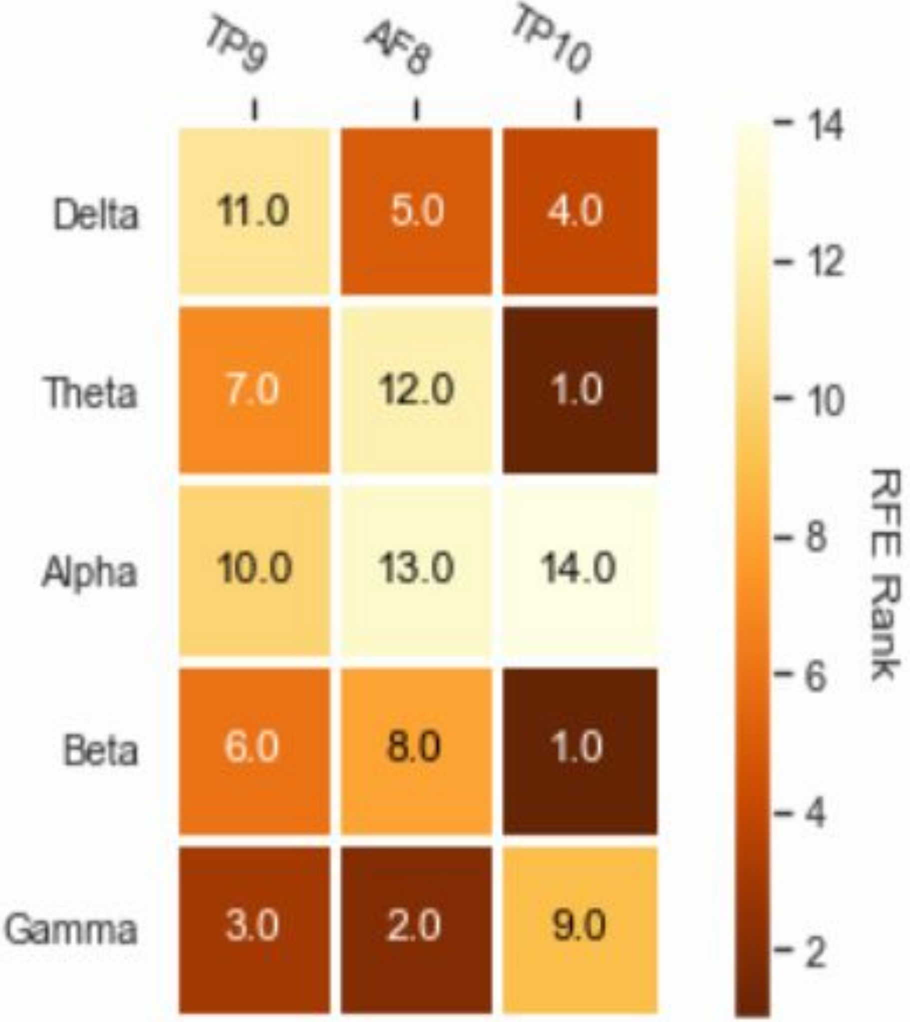
# Hot: Touch vs. Touch-Vision



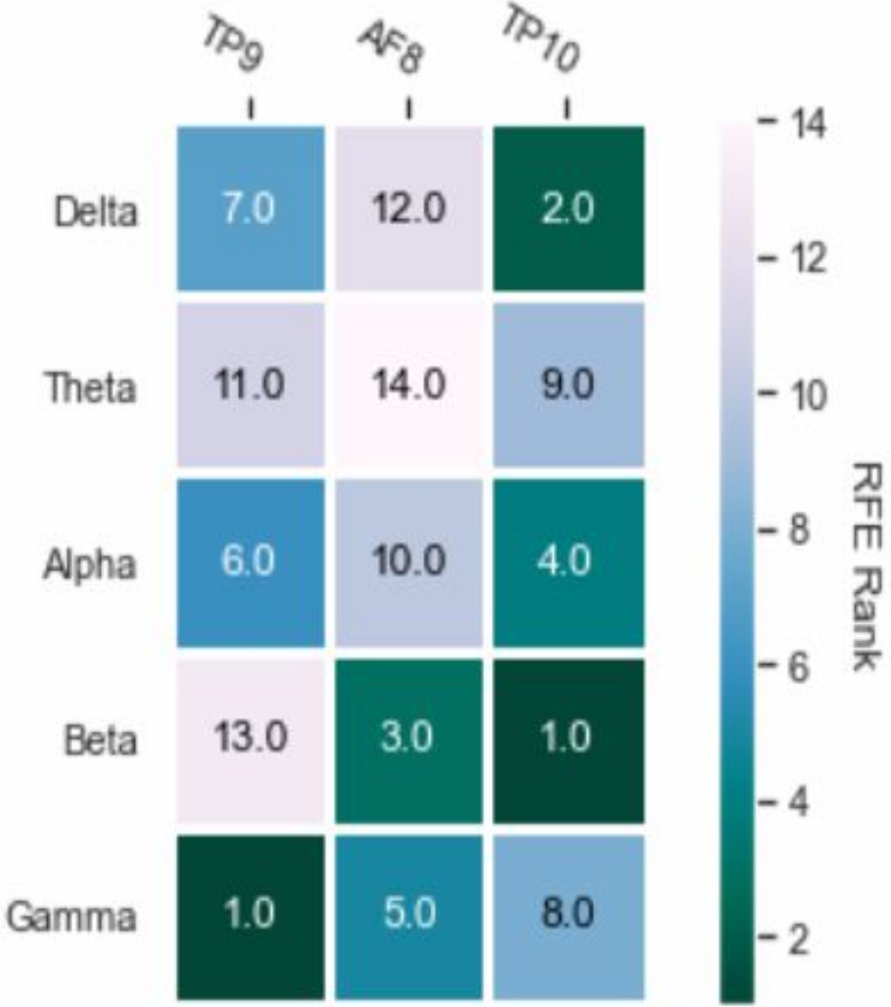
# Cold: Touch vs. Vision



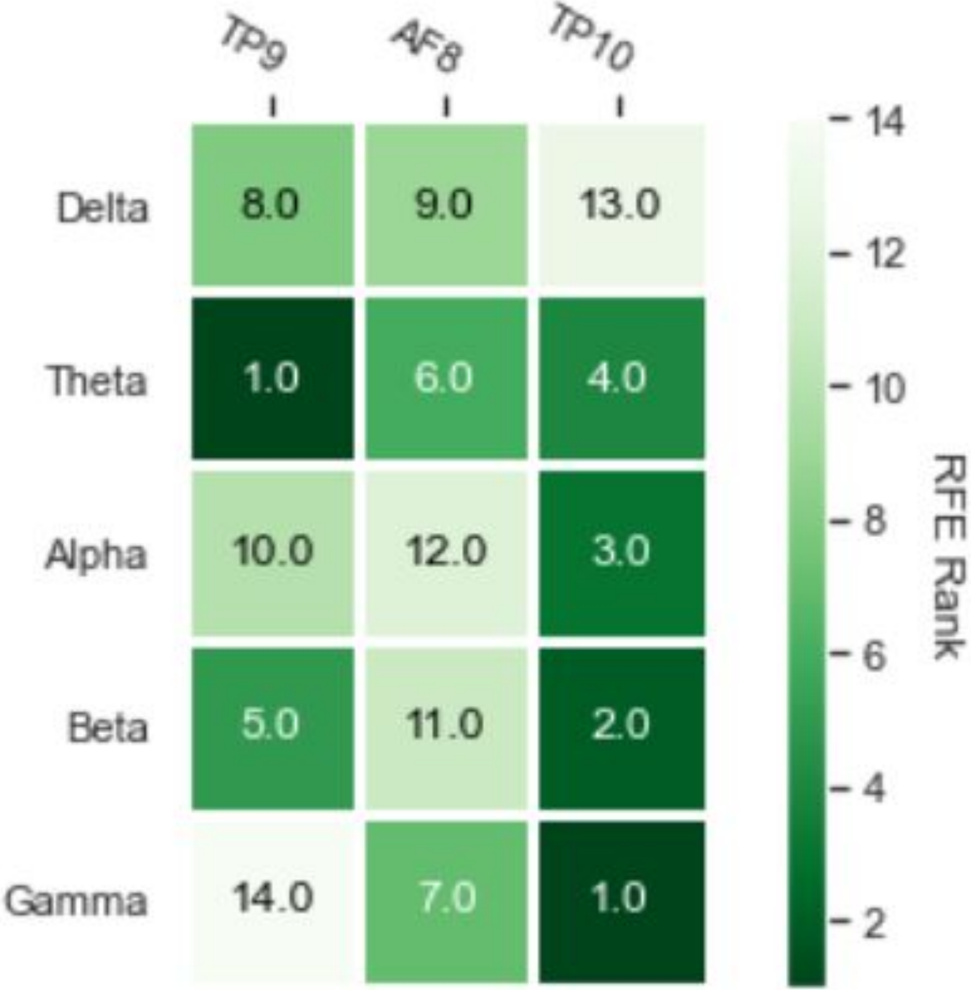
# Cold: Vision vs. Touch-Vision



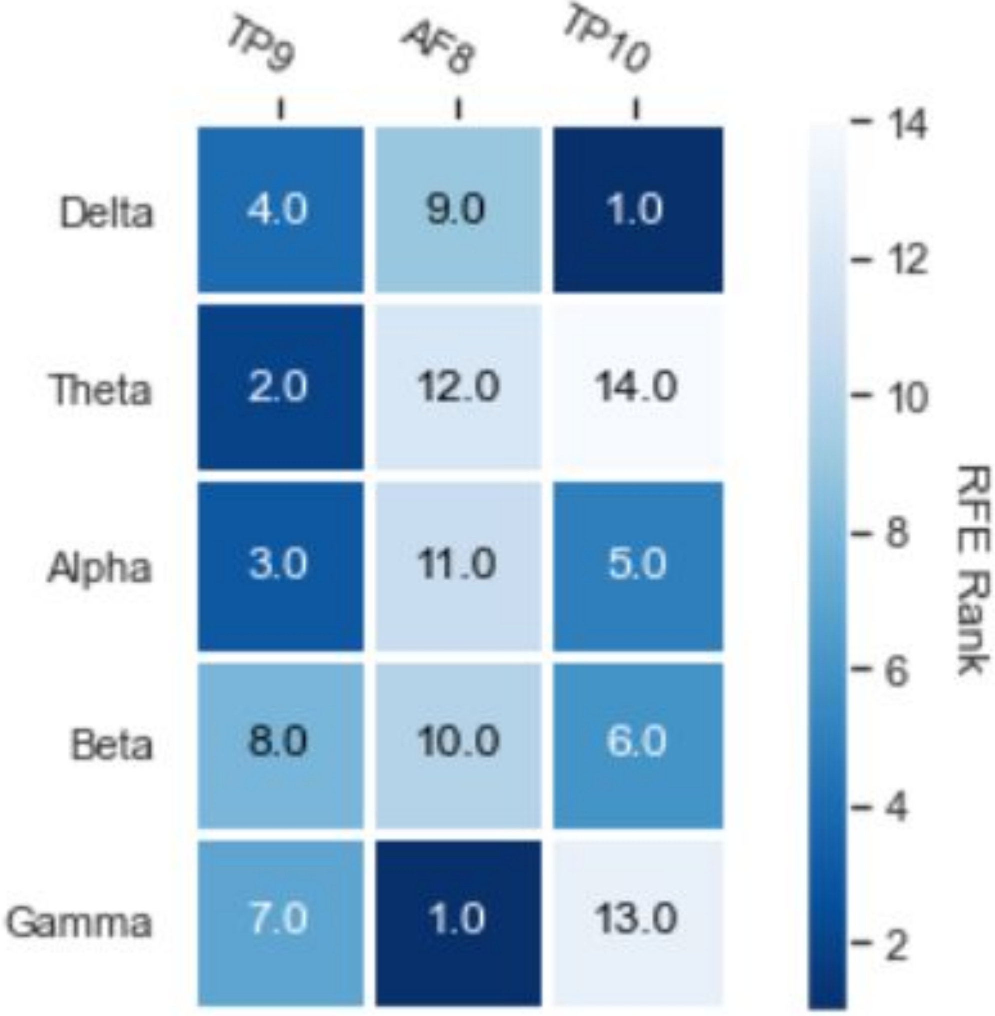
# Cold: Touch vs. Touch-Vision



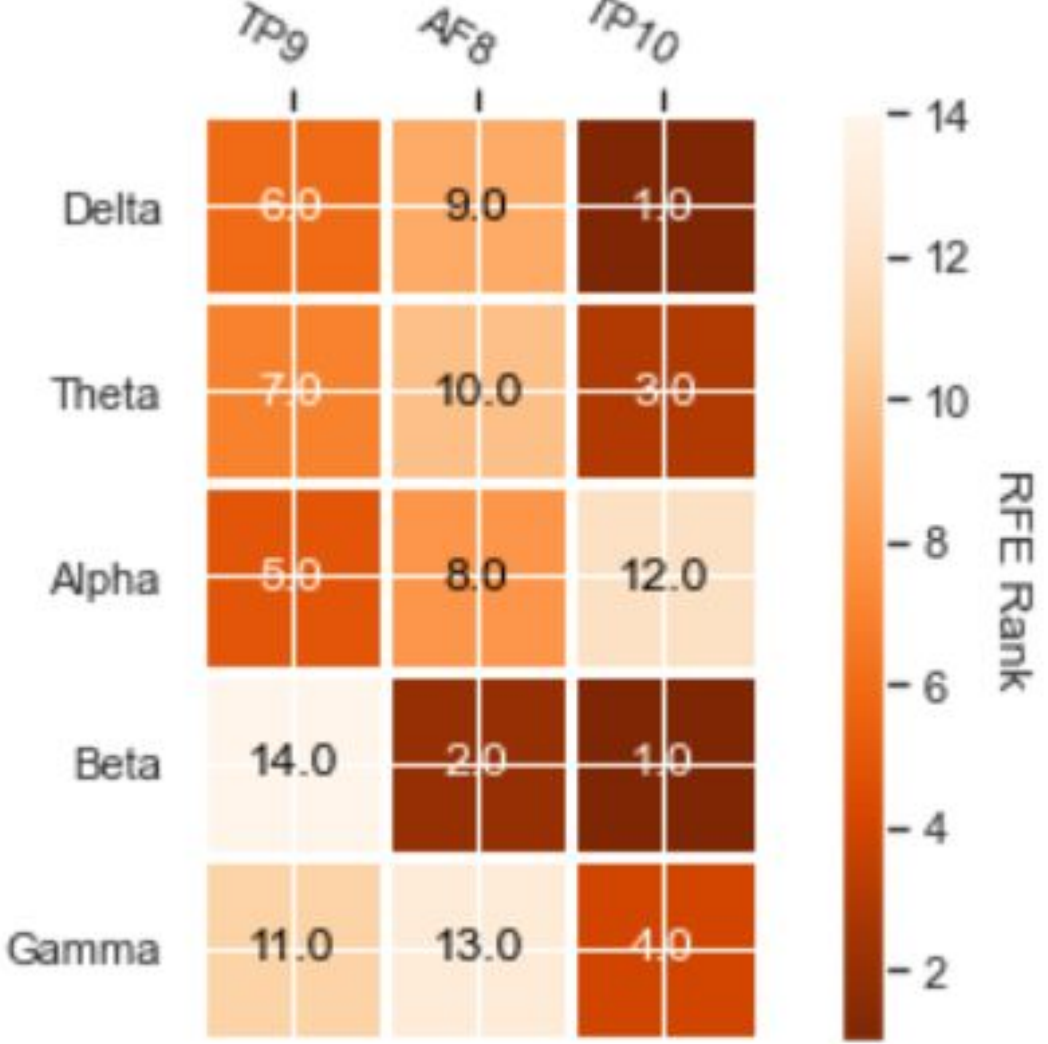
# Hot vs. Cold: Touch-Vision



# Hot vs. Cold: Vision



# Hot vs. Cold: Touch

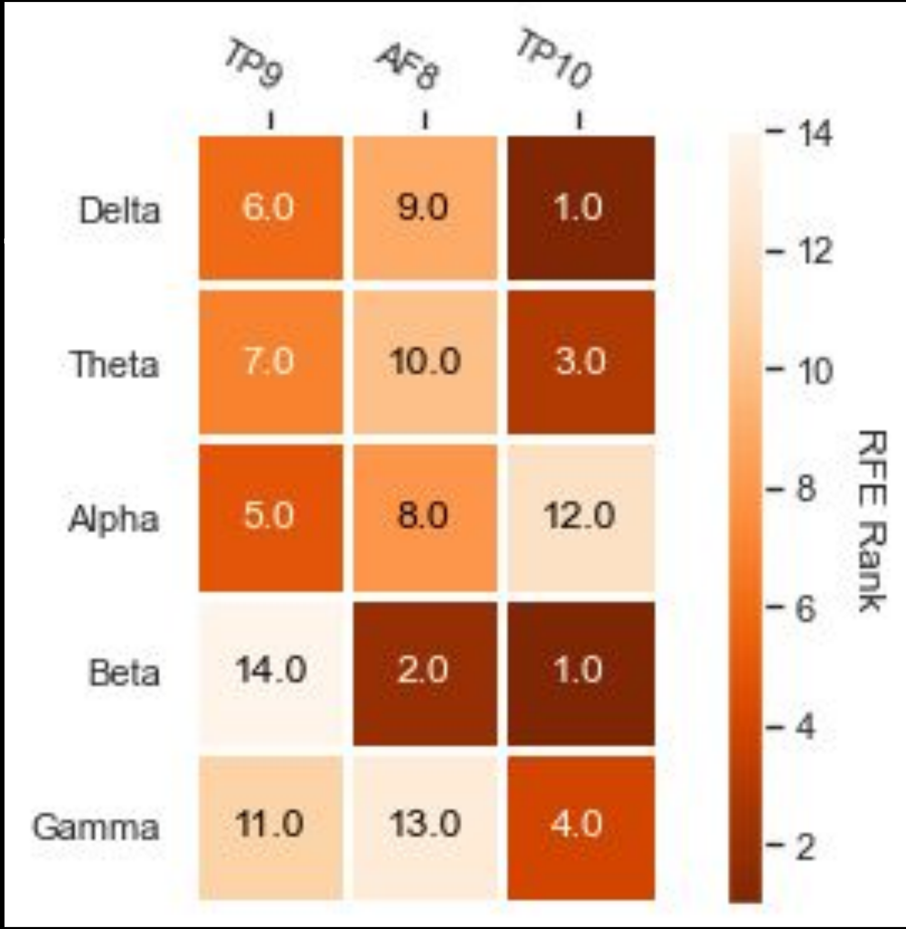


# Conclusions

- It seems as though TP10 is the most important feature for all of the touch types for hot and cold.
- I feel that from order of most important to least important, TP10>AF8>TP9.

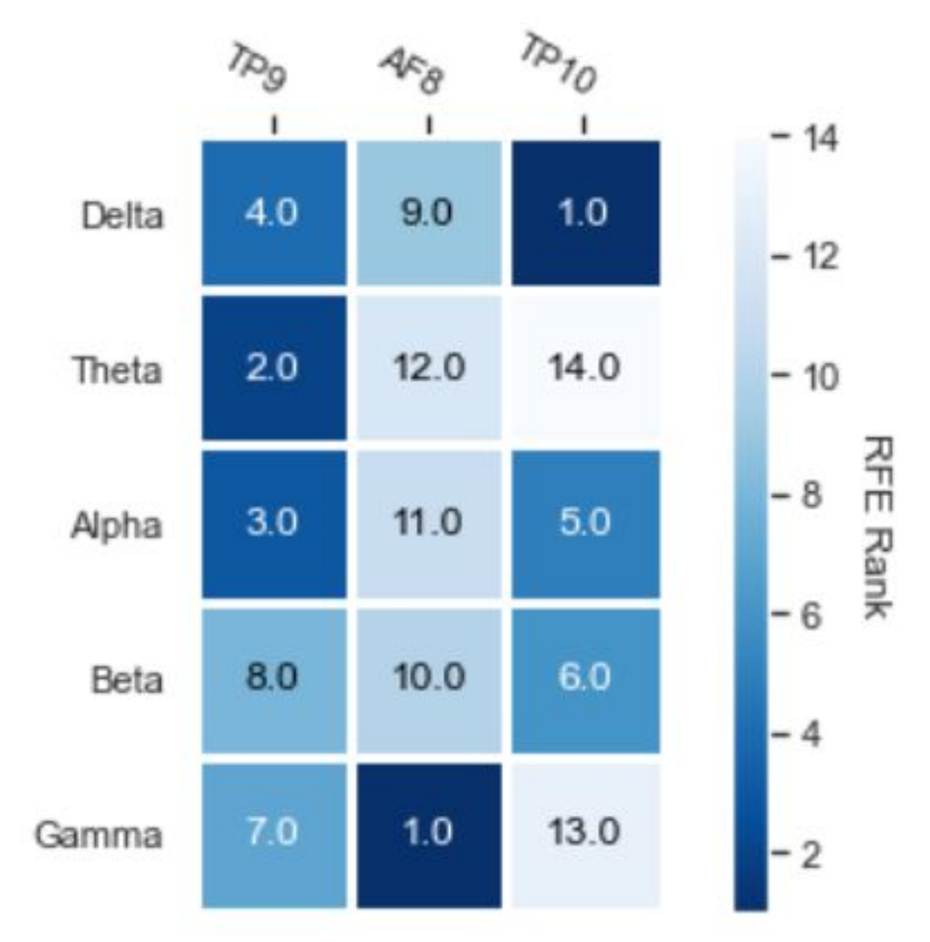
# Logistic Regression: Touch

- Logistic Regression for hot vs. cold.
- Darker the color means more important.
- TP9 seems to be the most important.



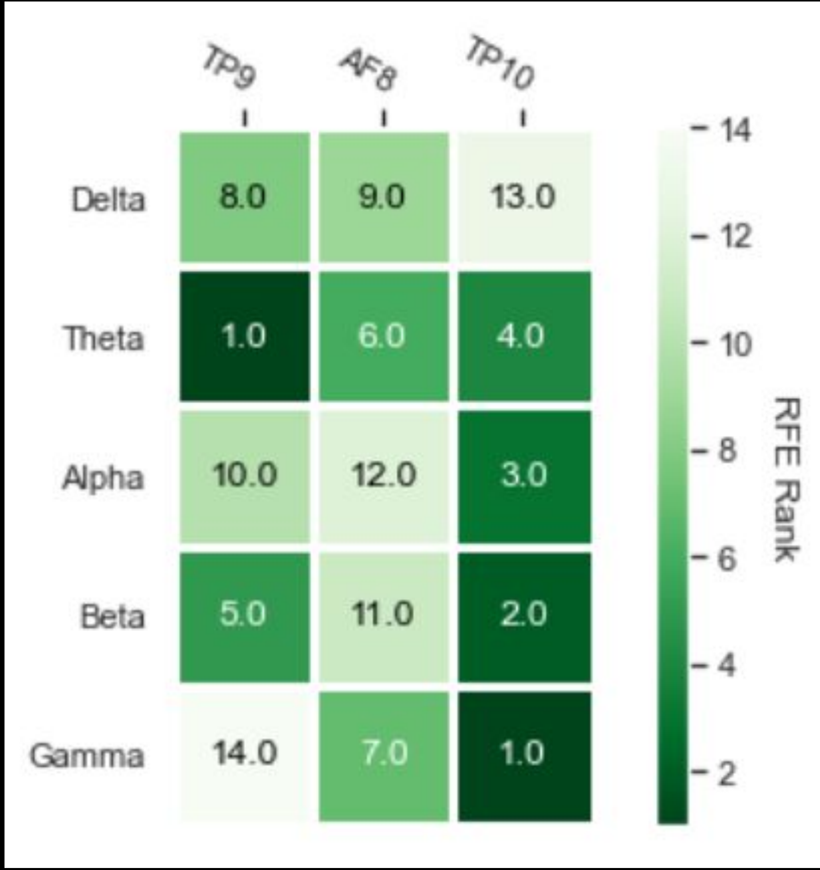
# Logistic Regression: Vision

- Hot vs Cold
- Darker means more important.
- This says TP10 is the most important.



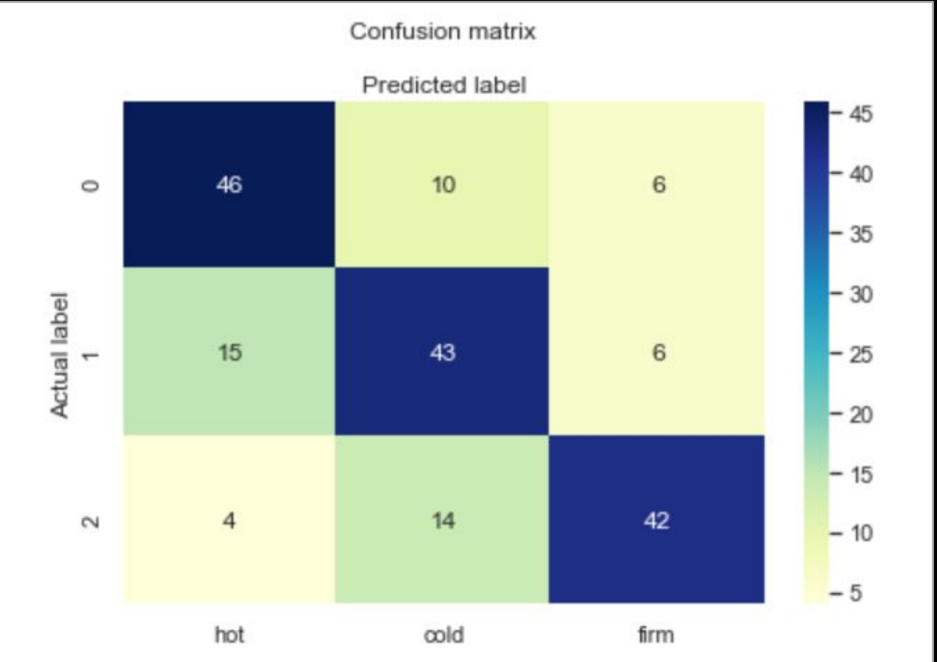
# Logistic Regression: Touch-Vision

- Darker colors mean more important.
- This says that TP9 is the most important feature.

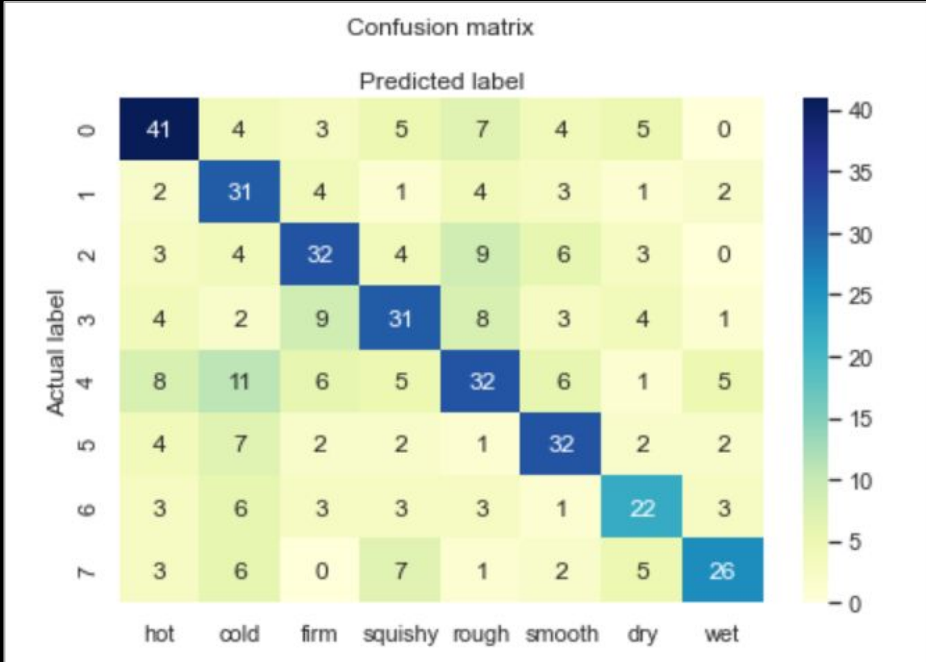


# Multiple Classifier: k_Nearest-Neighbours

- Neighbours = 1
- Accuracy = 0.7043010752688172

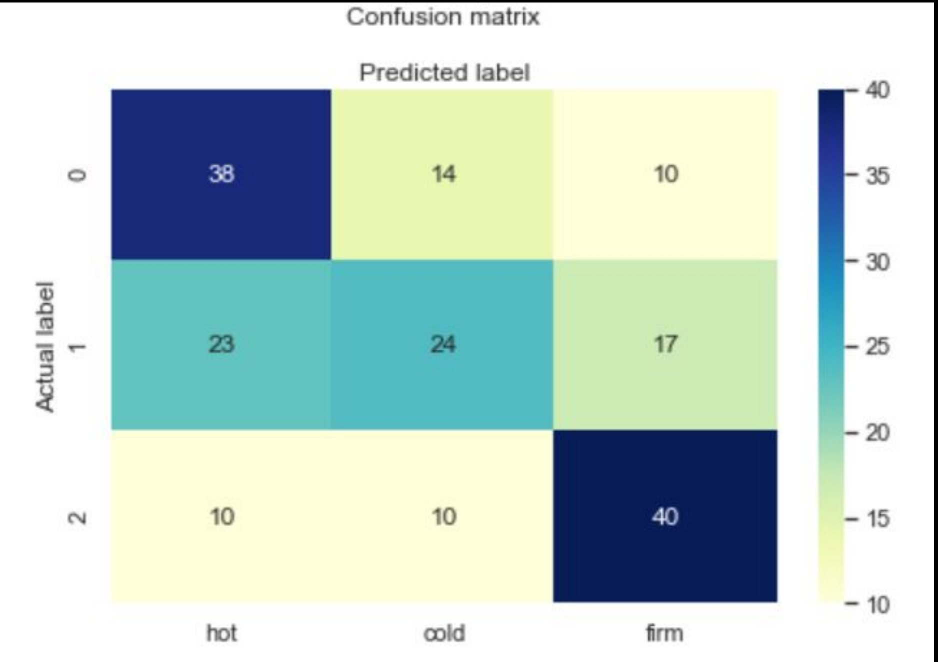



- Neighbours = 1
- Accuracy = 0.5369565217391304

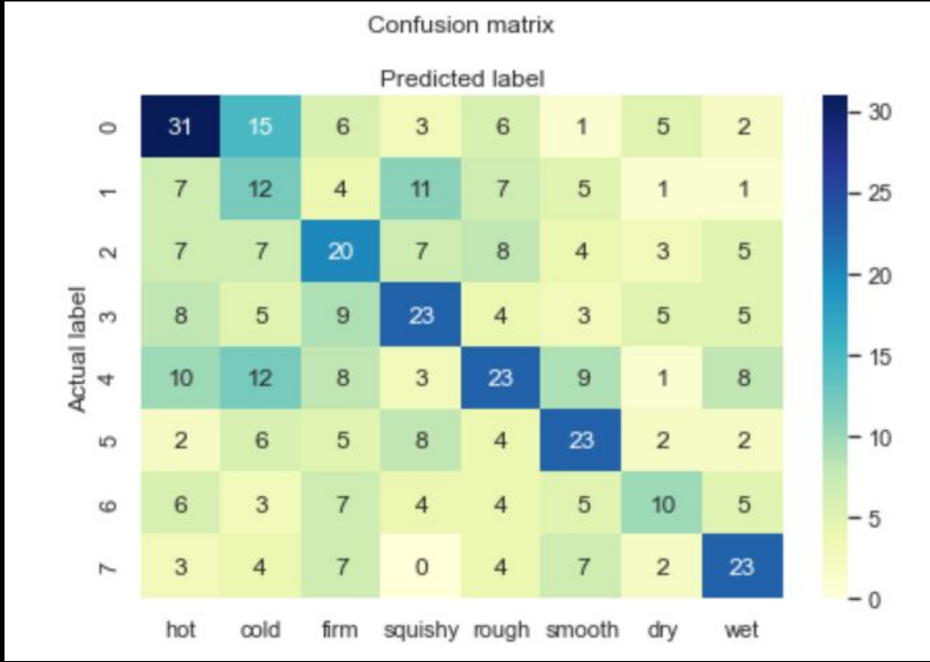


# Multiple Classifier: Multi-Layer Perceptron

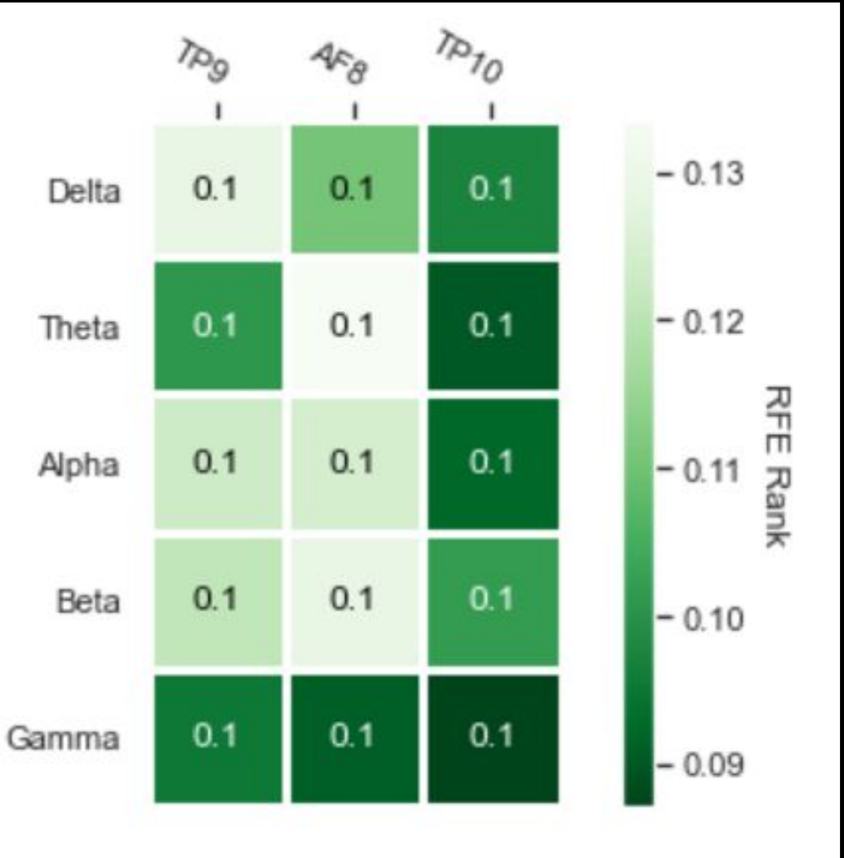
- Layers = 100, 100
- Accuracy = 0.5483870967741935




- Layers = 100, 100
- Accuracy = 0.358695652173913



# Multiple Classifier: k_Nearest Neighbours




- I did RFE for the KNN with all features.

# Important Dates

- Abstract
  - **March 5, 2021**
  - 250 word length
- Virtual Project Board ← Poster? Examples?
  - **March 7, 2021**
  - PDF showing most important data of project
- Video
  - Video discussing project
  - **March 7, 2021**
  - This is for special award judging.

# February 16

Cleah Winston

# Justification: TP10 is the most important

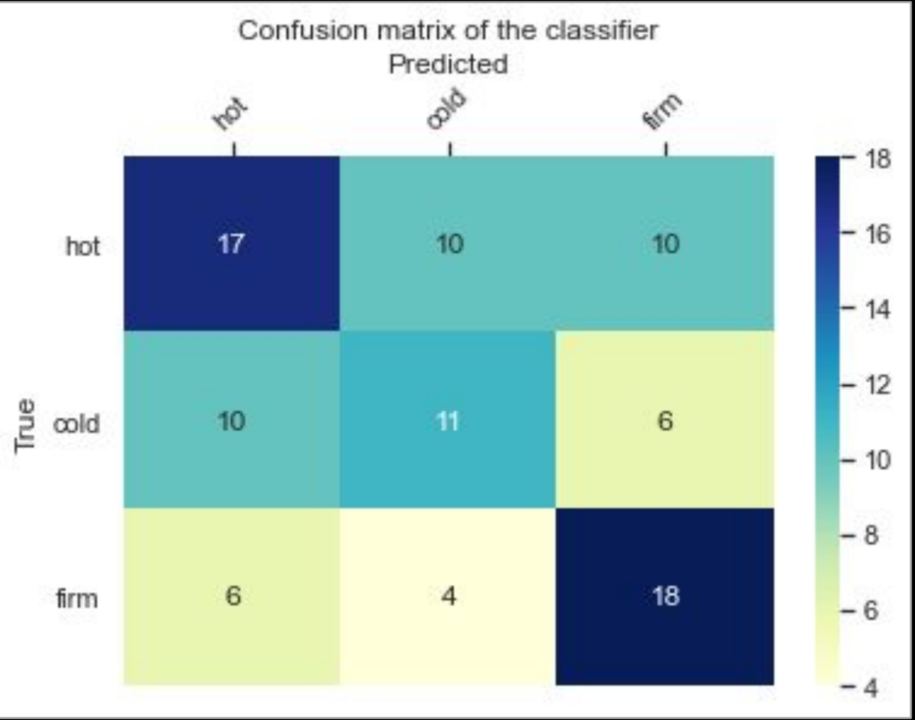
| HOT | TP9 | AF8 | TP10 |
| --- | --- | --- | --- |
| Touch vs. Vision | 59.0 | 25.0 | 22.0 |
| Touch vs. Touch-Vision | 46.0 | 37.0 | 23.0 |
| Vision vs. Touch-Vision | 40.0 | 46.0 | 20.0 |

| COLD | TP9 | AF8 | TP10 |
| --- | --- | --- | --- |
| Touch vs. Vision | 44.0 | 32.0 | 30.0 |
| Touch vs. Touch-Vision | 38.0 | 44.0 | 24.0 |
| Vision vs. Touch-Vision | 37.0 | 40.0 | 29.0 |

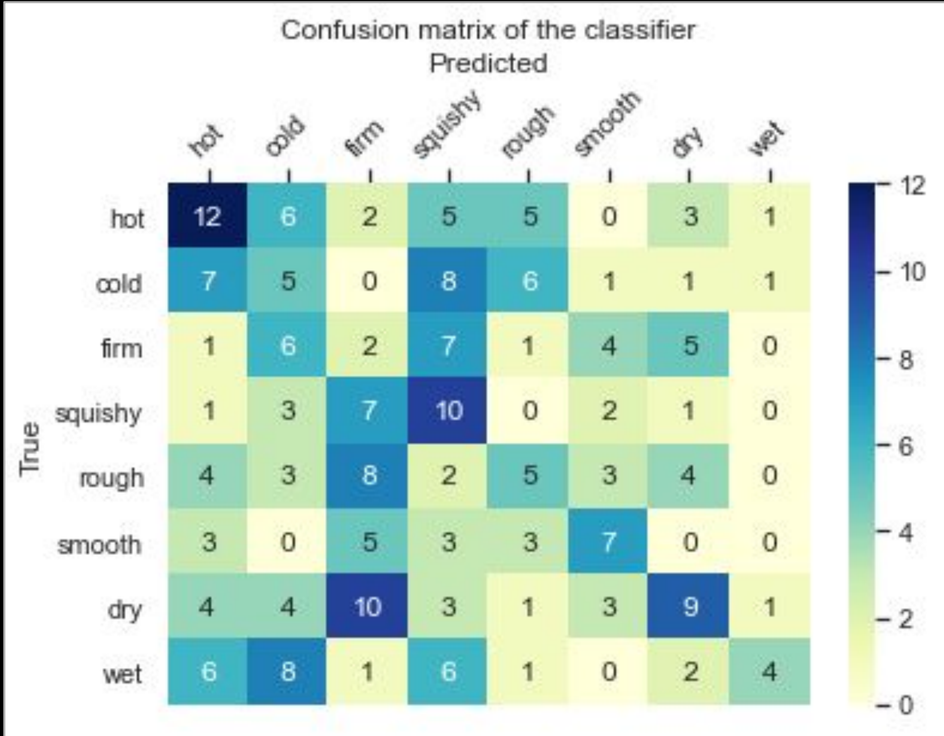
# Multiple Classification of Touch types in the Touch state

# Multiple Classifier: Logistic Regression

- Accuracy = 0.5

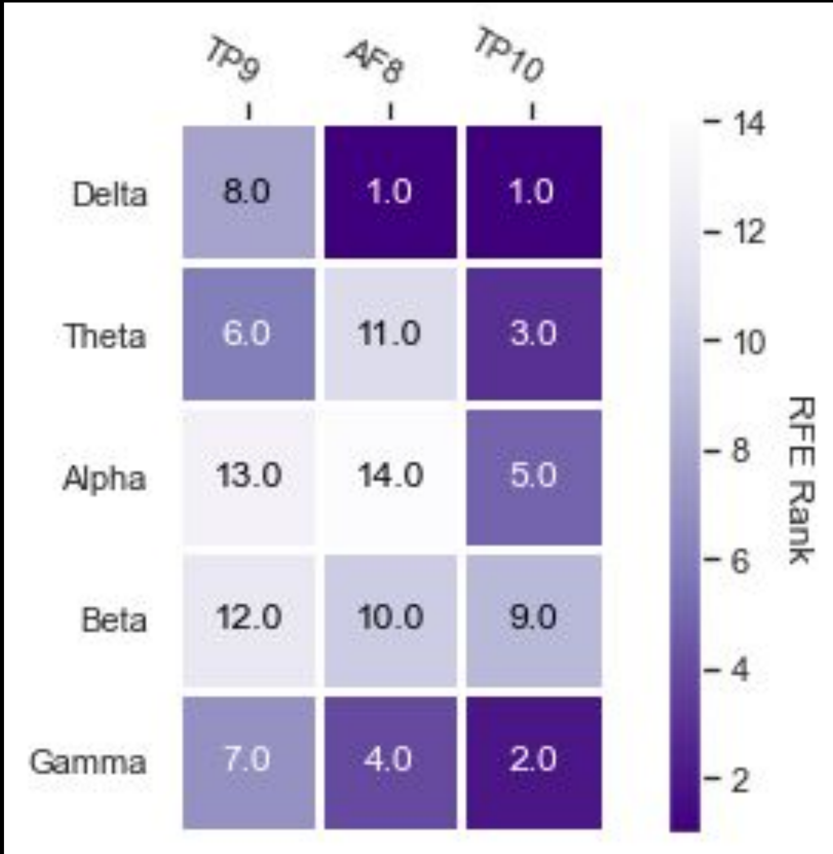



- Accuracy = 0.23893805309734514



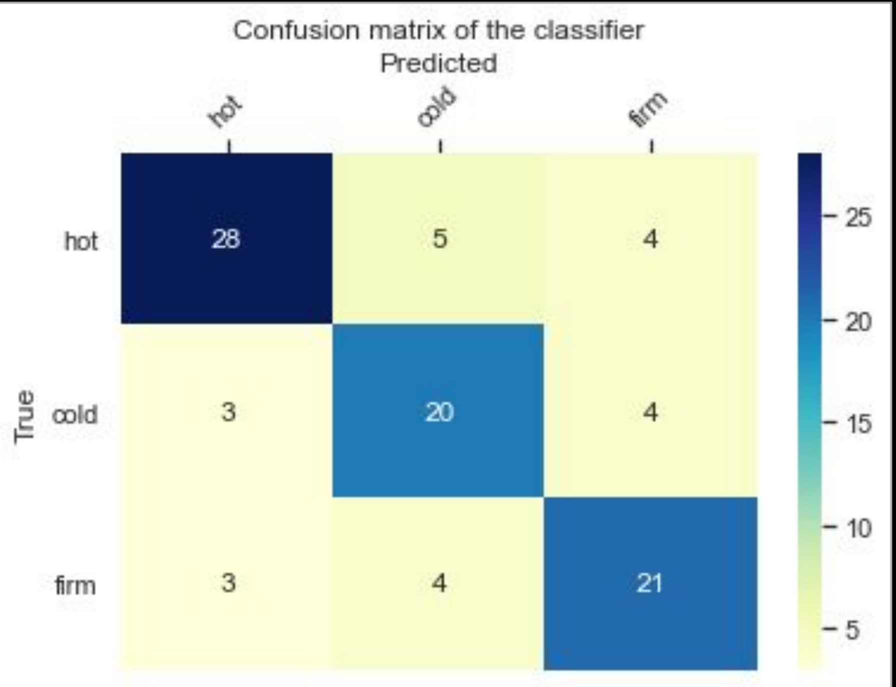
# Multiple Classifier: Logistic Regression

- I did RFE for Logistic Regression for all features.

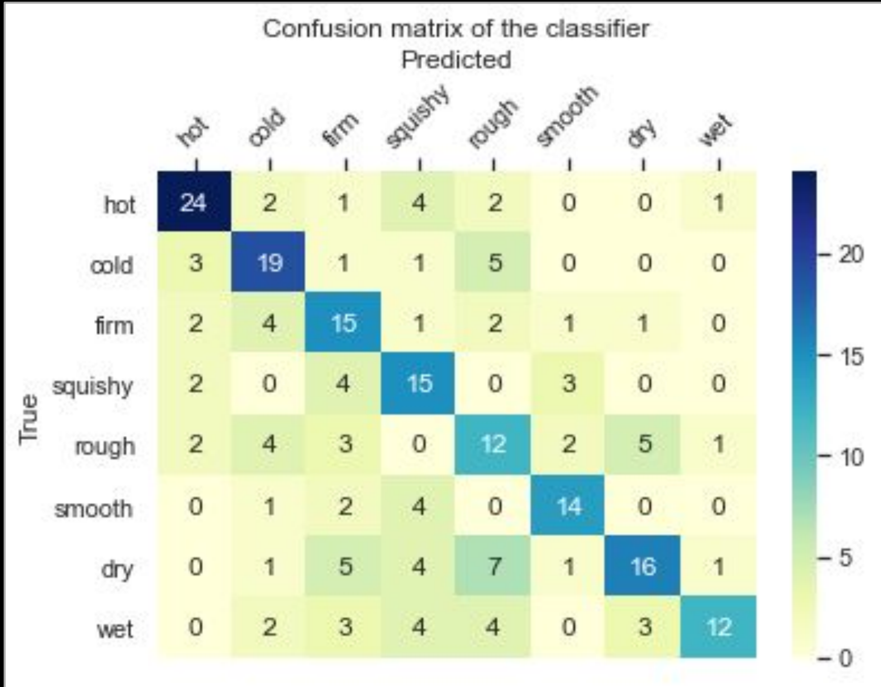


# Multiple Classifier: k_Nearest-Neighbours

- Neighbours = 1
- Accuracy = 0.75

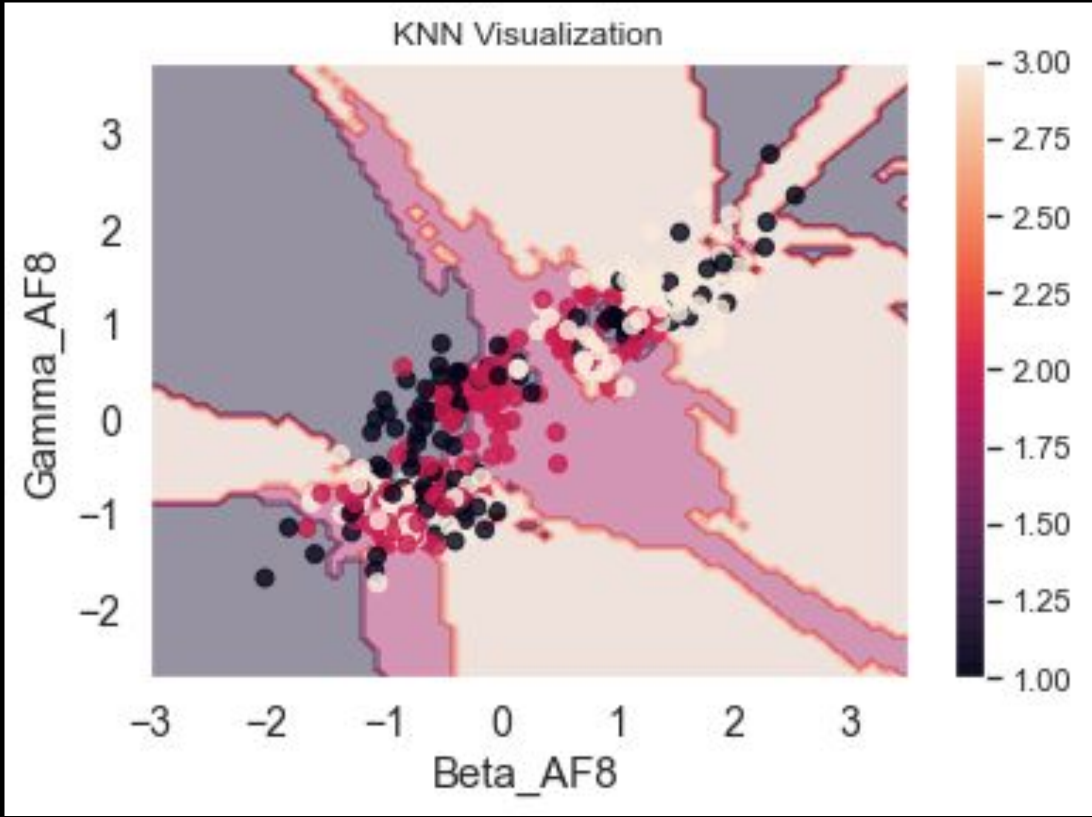



- Neighbours = 1
- Accuracy = 0.5619469026548672



# k-Nearest-Neighbours: Visualization

- This shows the KNN separation.
- The point represent the training points and the fills represent regions in the testing mesh.
- These are the most important features according to RFE.
- Gray → Hot
- Pink → Cold
- White → Firm

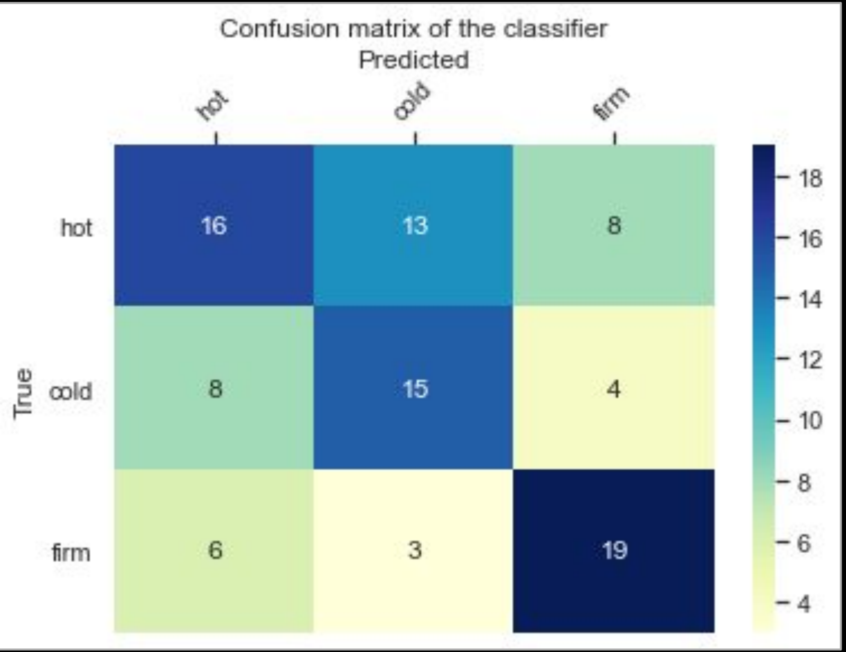


# Multilayer Perceptron Architecture

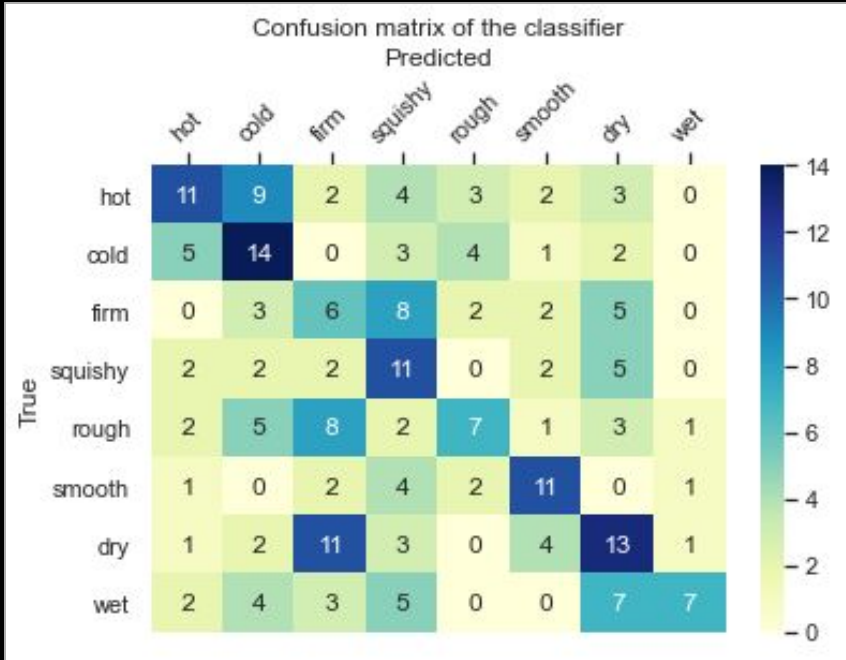
- 2 layers
- 100 neurons per layer
- (Multilayer Perceptron on next slide)

# Multiple Classifier: Multi-Layer Perceptron

- Layers = 100, 100
- Accuracy = 0.5434782608695652



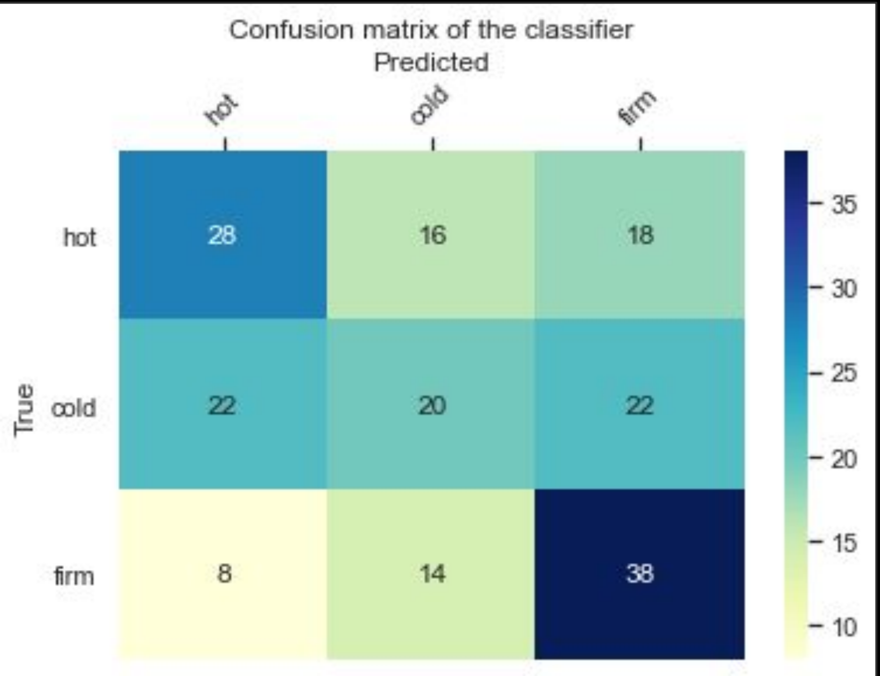

- Layers = 100, 100
- Accuracy = 0.35398230088495575

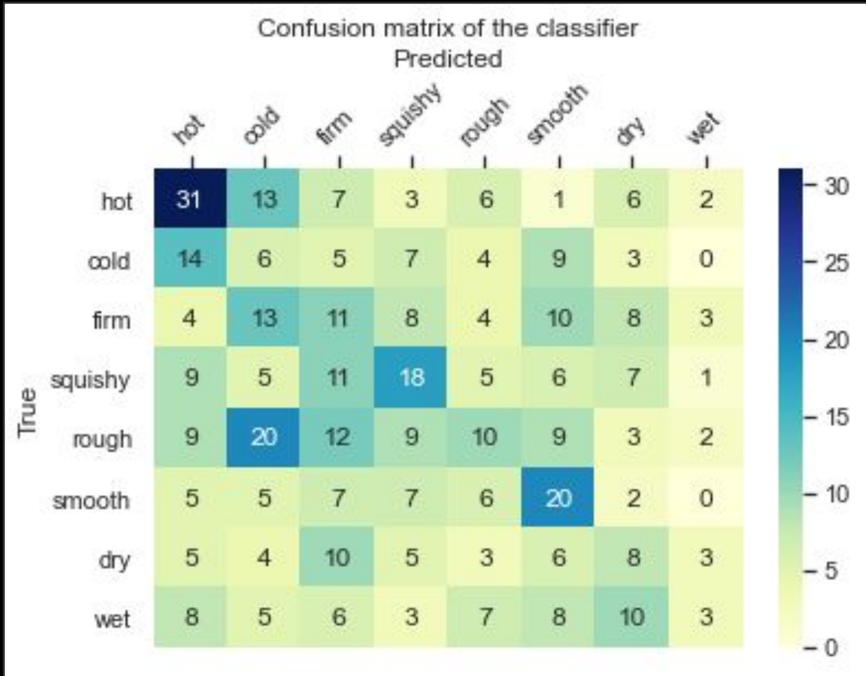


# Multiple Classification for Touch types in the Touch-Vision and Touch State

# Multiple Classifier: Logistic Regression

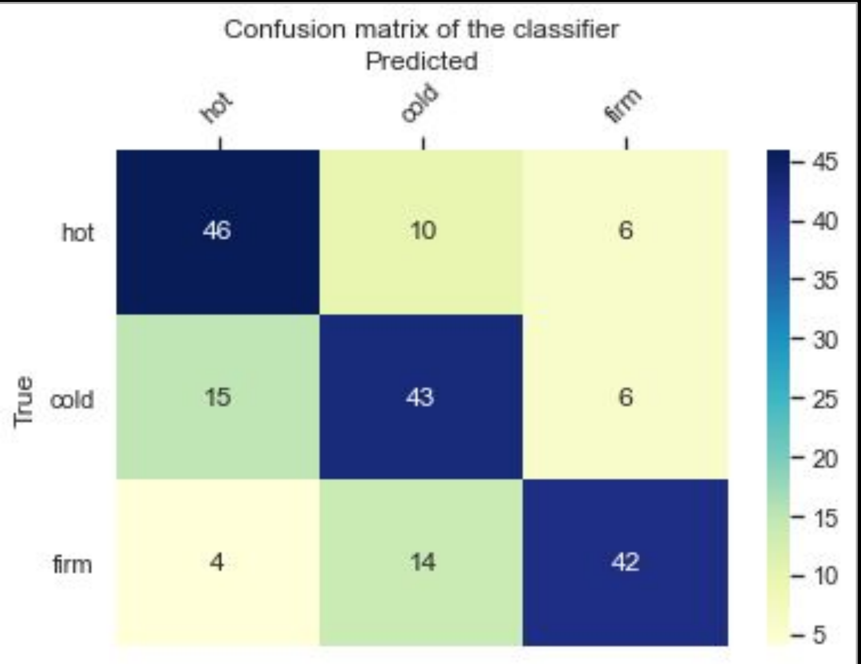
- Accuracy = 0.46236559139784944
- Accuracy = 0.2326086956521739



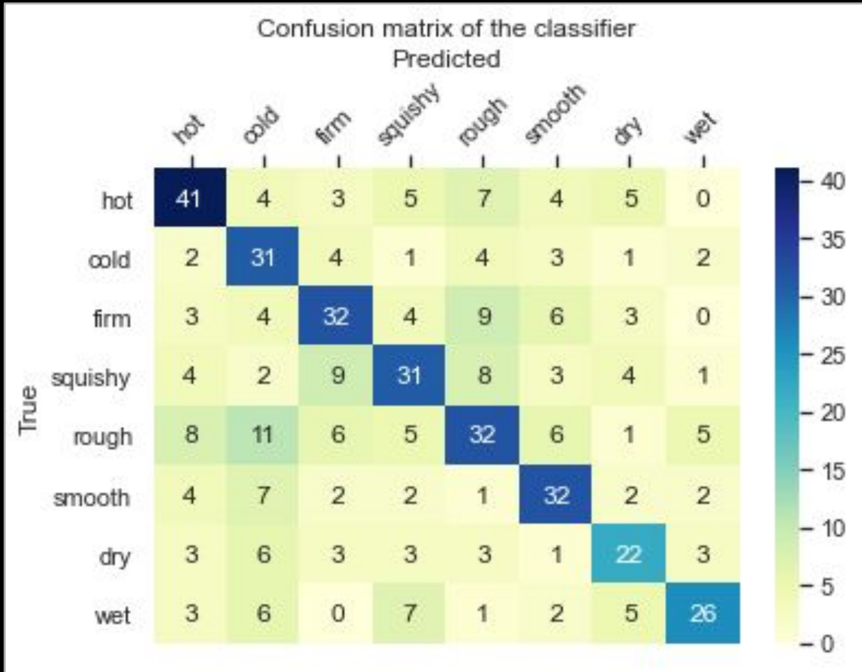



# Multiple Classifier: k_Nearest-Neighbours

- Neighbours = 1
- Accuracy = 0.7043010752688172

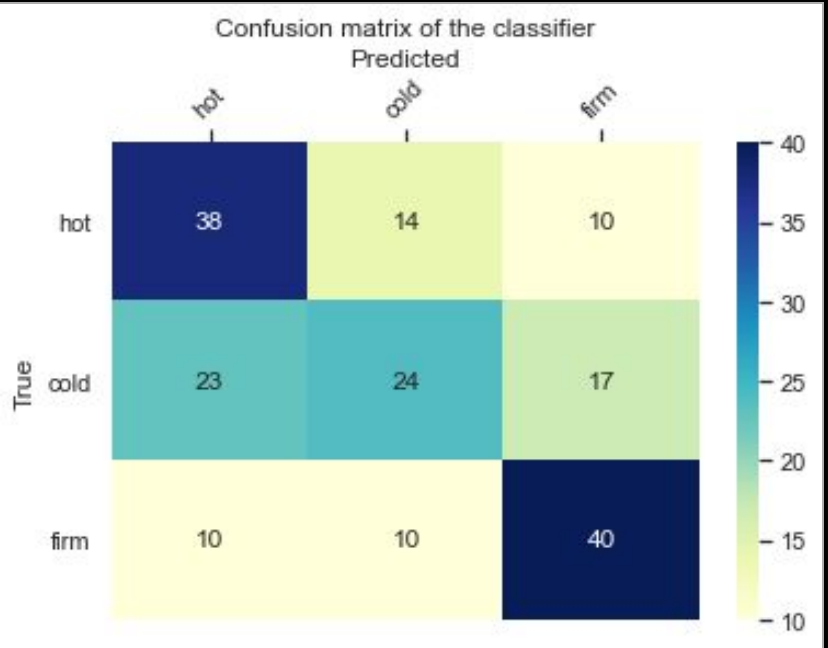



- Neighbours = 1
- Accuracy = 0.5369565217391304

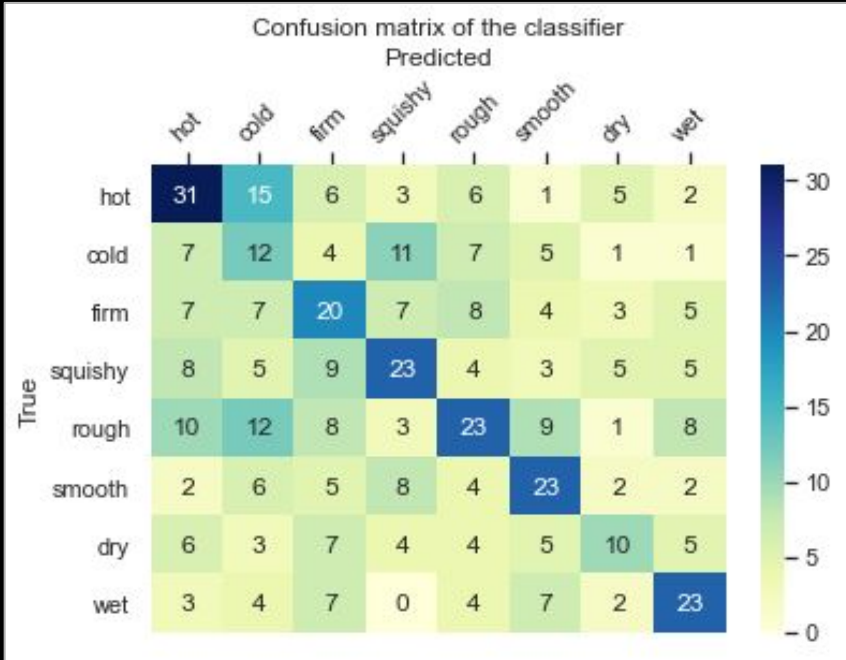


# Multiple Classifier: Multi-Layer Perceptron

- Layers = 100, 100
- Accuracy = 0.5483870967741935

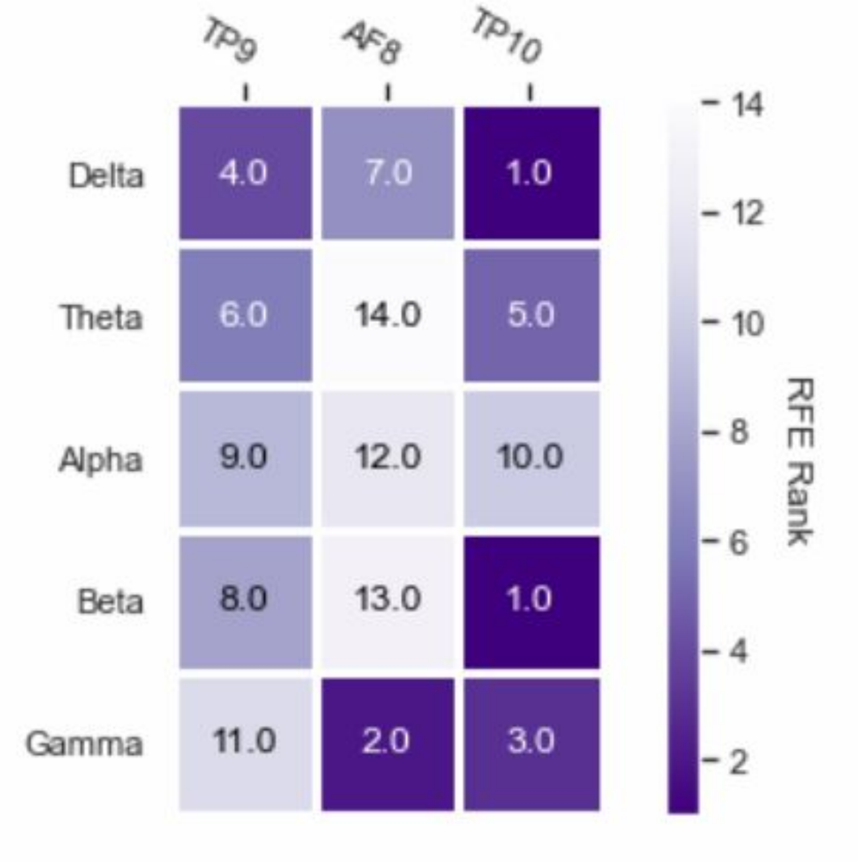



- Layers = 100, 100
- Accuracy = 0.358695652173913



# Multiple Classifier: Logistic Regression

- I did RFE for Logistic Regression for all features.



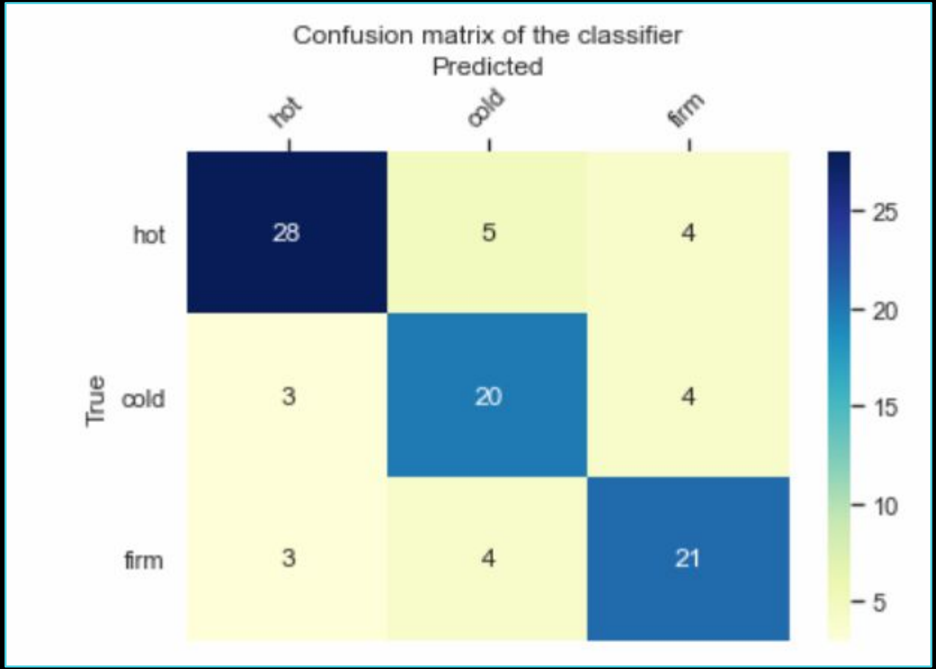
# February 27

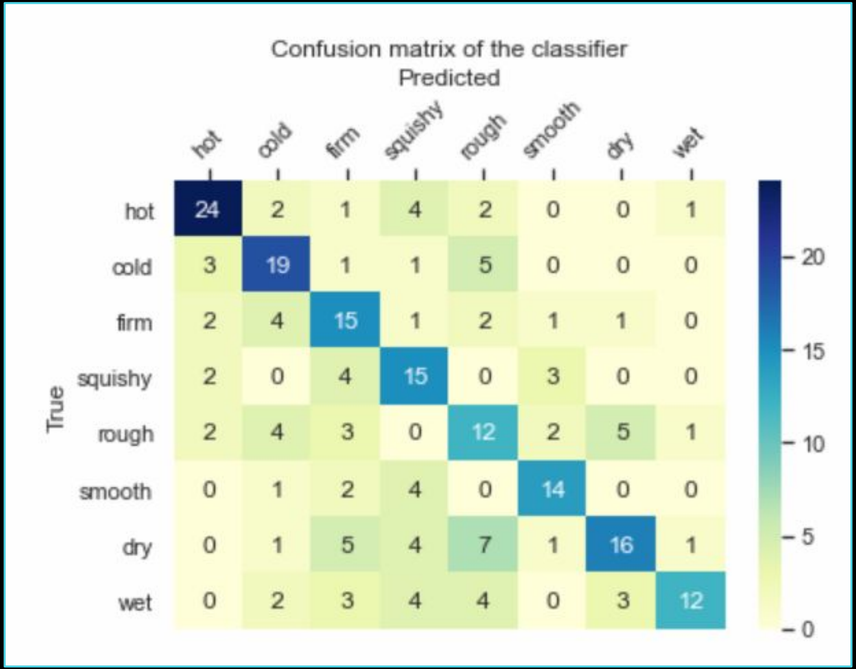
Cleah Winston

# How does Multiple Classification for Logistic Regression work?

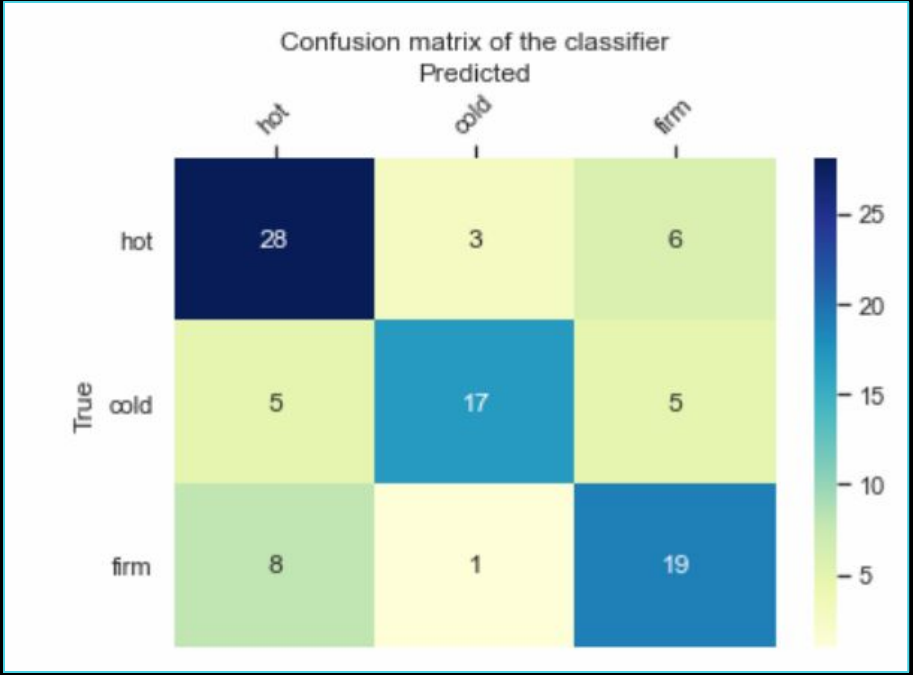
- https://towardsdatascience.com/multiclass-classification-algorithm-from-scratch-with-a-project-in-python-step-by-step-guide-485a83c79992
- Multiple Classification uses the *one vs. all* method to classify.
- Basically, it considers one class vs. all other classes.
- Logistic Regression uses a sigmoid function.
  - This function returns a value between 0 and 1.
  - If the value is between 0 and 0.5, the predicted value would be 0.
  - If the value is between 0.5 and 1, the predicted value would be 1.

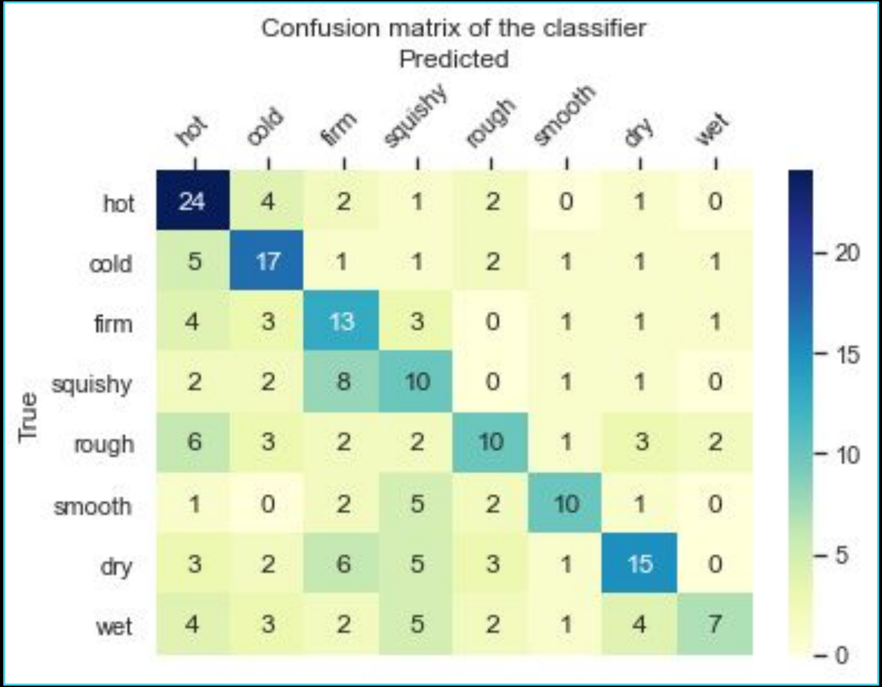
# k-Nearest-Neighbours Touch: Neighbours = 1



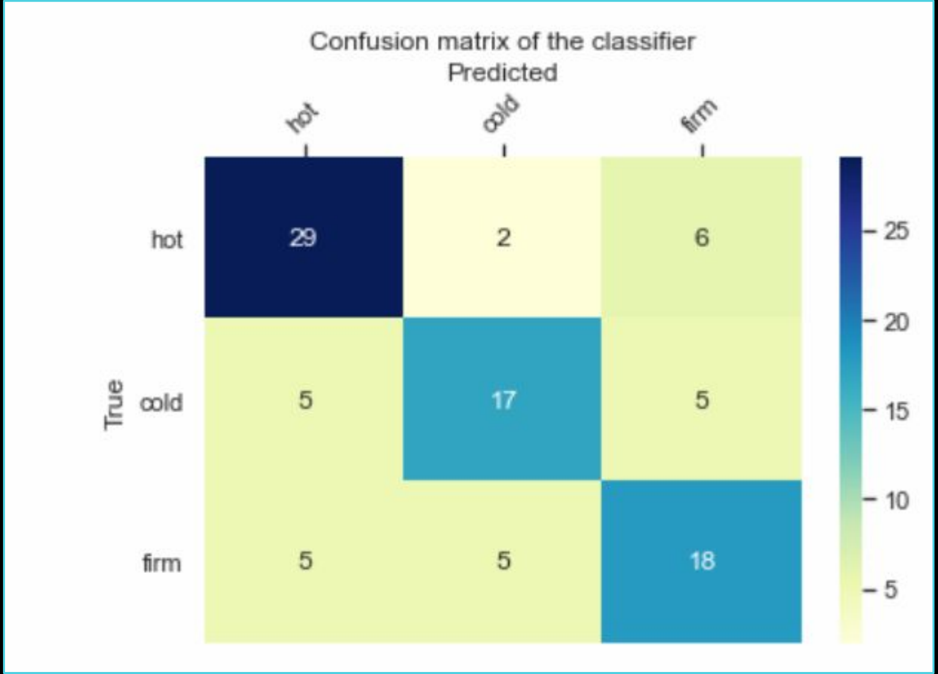



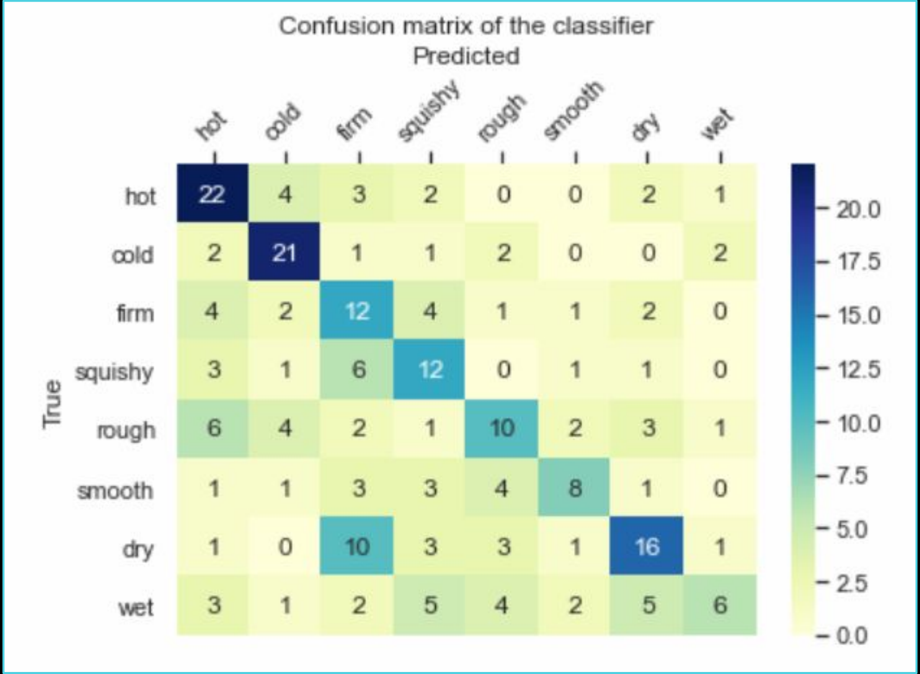
# k-Nearest-Neighbours Touch: Neighbours = 3



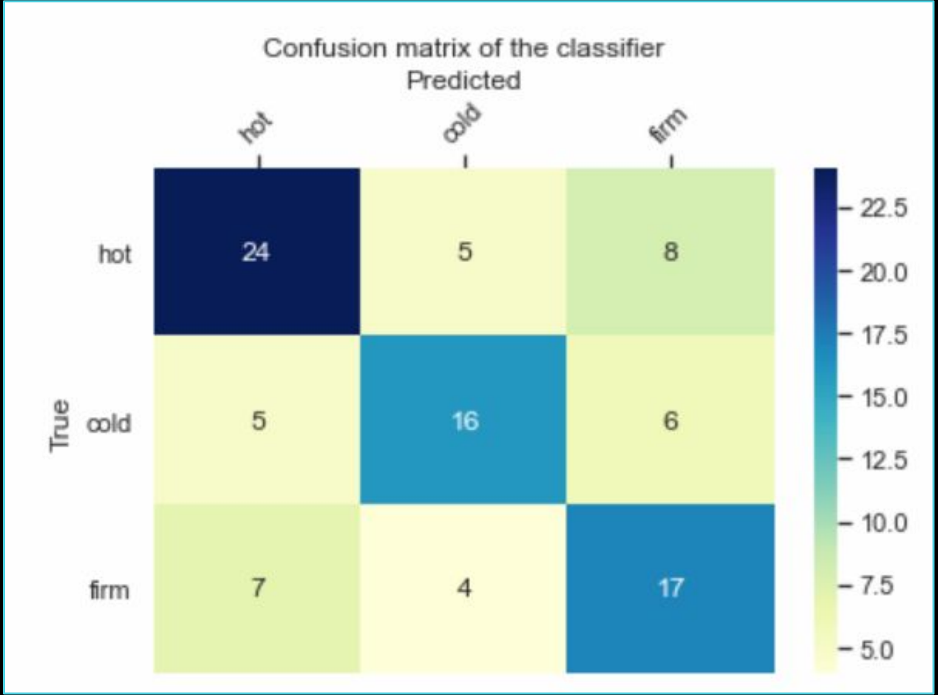



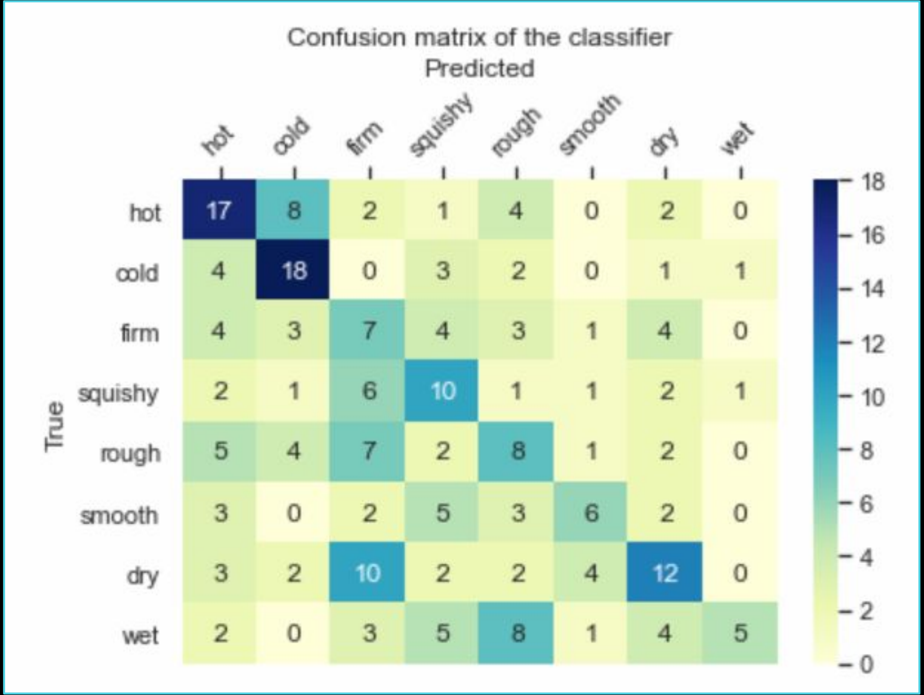
# k-Nearest-Neighbours Touch: Neighbours = 5



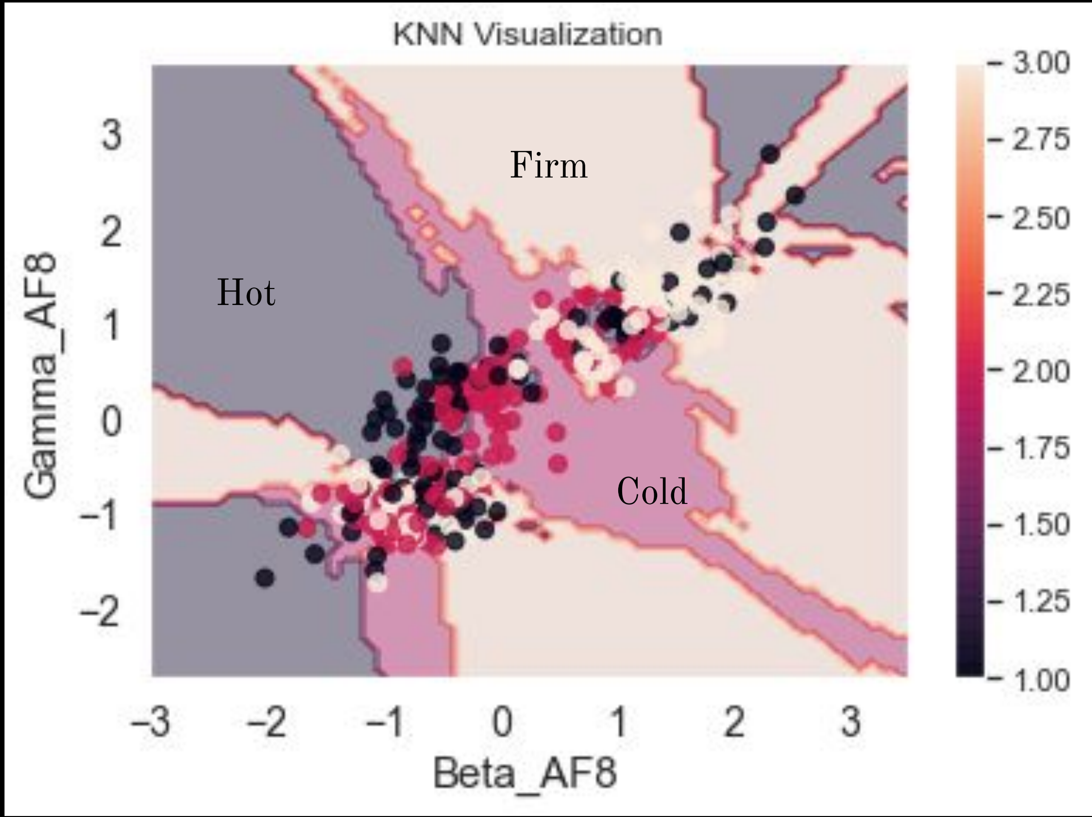
# k-Nearest-Neighbours Touch: Neighbours = 11



# Clarifying the KNN Visualization

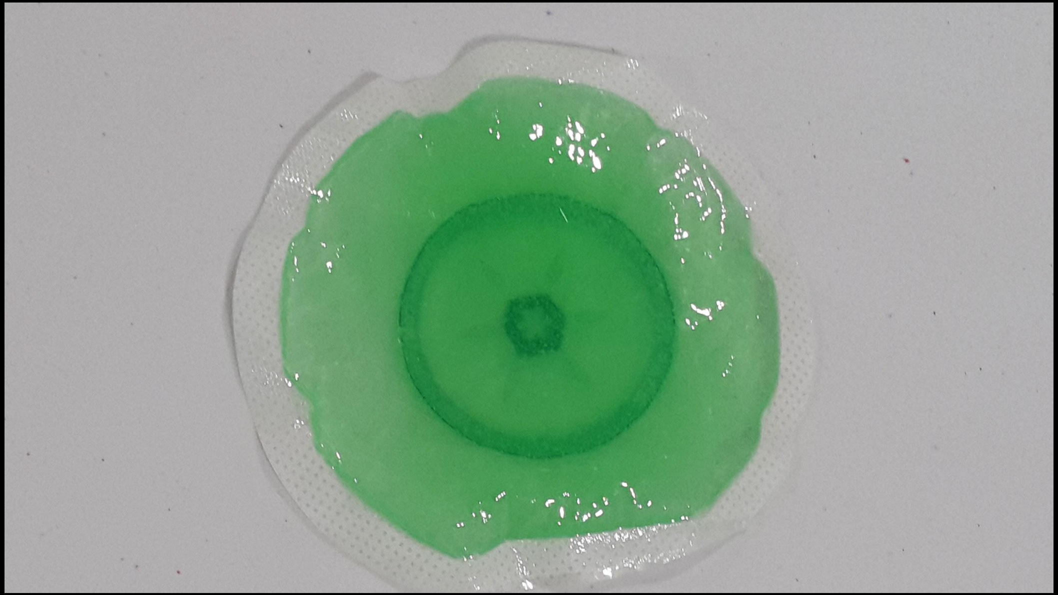
- The x-axis and y-axis represent the how much frequency there is in an actual
- I normalized the frequency bands, so they are z-scores.
- The legend represents the dots: black is hot, pink is cold, and peach is firm.
- The intermediate values are not



Screen to prevent other distractions in the room

Table where participants outstretched their arms and objects were placed on their left palm or on the table

Hot

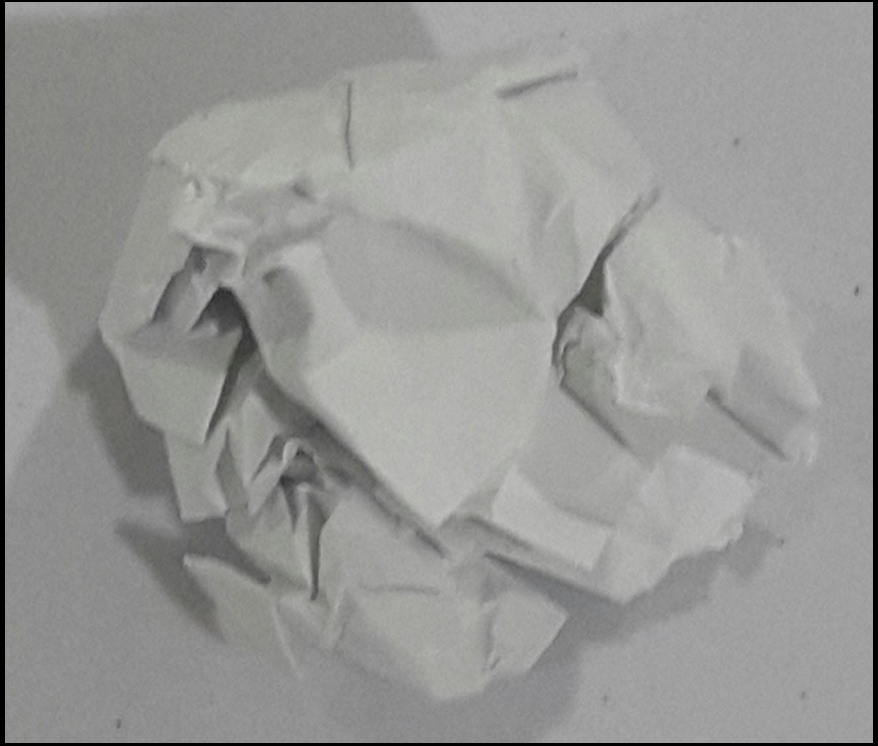


Cold

Firm

Squishy

Rough



Smooth

Dry

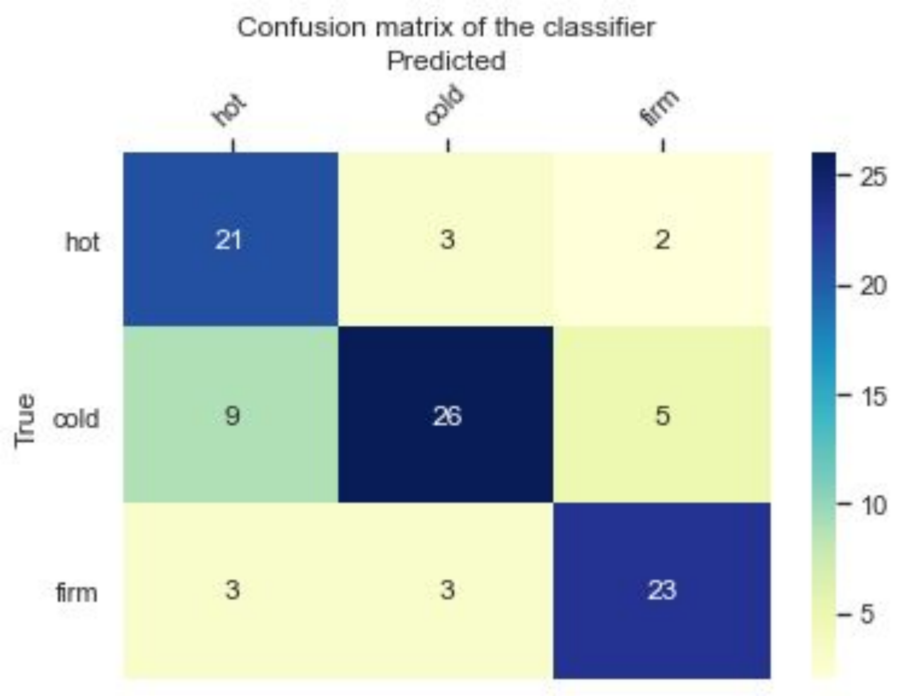
Wet

# March 1

Cleah Winston

# k-Nearest-Neighbours Vision: Neighbours = 3

- Accuracy = 0.7368421052631579




- Accuracy = 0.4765957446808511