CIS 6180 Final Project Report

Eye Tracking Event Detection and Classification

By Erich MacLean, B. ENG

emacle05@uoguelph.ca

Introduction

Eye tracking is a technology that is used to measure the position and movement of the human eye. Commonly used in applications such as healthcare and research, eye tracking proves to be a versatile tool in determining where someone is looking. Eye tracking can also be applied in other applications too, such as in driving, or gaming technologies. The ultimate goal with eye tracking is to measure what is known as the *gaze point*. This point may be measured in 3D space, or 2D space depending on the type of eye tracking that is used. The gaze point is useful for many reasons, but mainly because it is the point of origin of visual information.

Eye tracking technologies come in many forms. Some are purely camera-based technologies, such as with a webcam. These methods tend to be less accurate due to the fact that only a single frame of video must be processed and analyzed to measure the eye movement. Camera technology however is highly accessible which adds benefit to that method. Sensor based eye tracking uses a technique called *pupil-corneal reflection.* Light, or near-infrared light, is directed at the outer layer of the eye, also referred to as the cornea. Sensors then detect that light and can measure the distance from the point of light to the center of the pupil. From this information, the gaze point can be calculated. Several sensors can be used to increase the accuracy of this technique. Due to the accuracy of the sensors, this tends to be a better method.

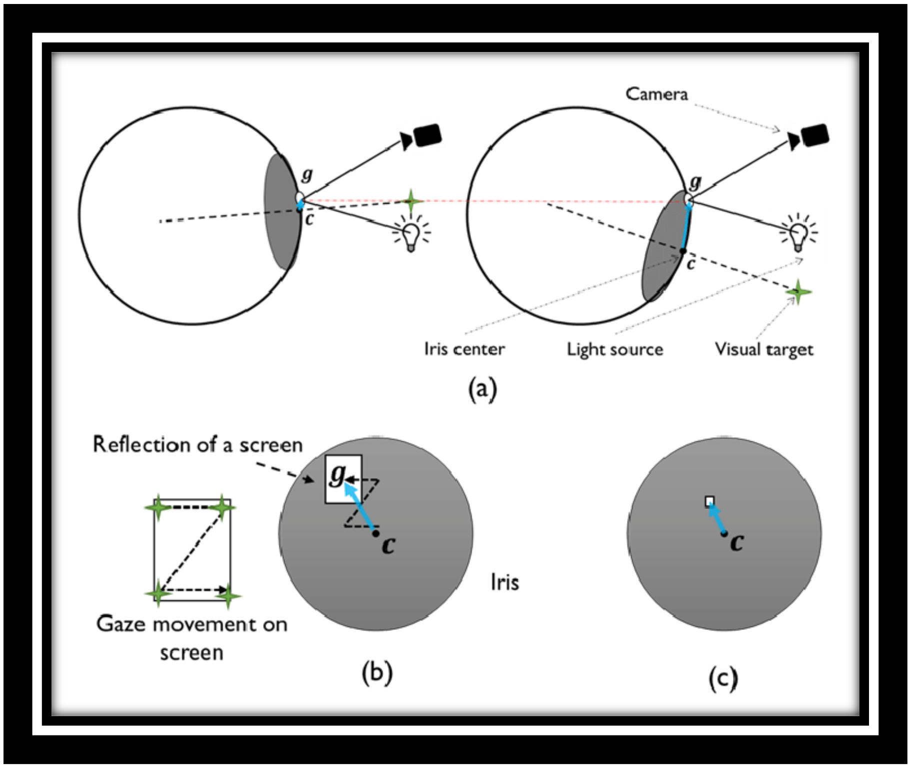


Figure 1: Diagram of pupil-corneal reflection.

From measuring the gaze point, eye tracking events are created. These events are simply different categories of eye movements. Fixations are when the eye remains focused on an object. This event is when the eye gathers most of the visual information. The duration of a fixation is between 80ms-600ms and can vary depending on the task. Both the speed and distance of the eye movements are low during a fixation event. Saccades are events where the eye moves between fixations. These events are much faster, and the eyes travel further than they do for a fixation. A saccade lasts for roughly 30ms-80ms. These two events are the most common in eye tracking literature.

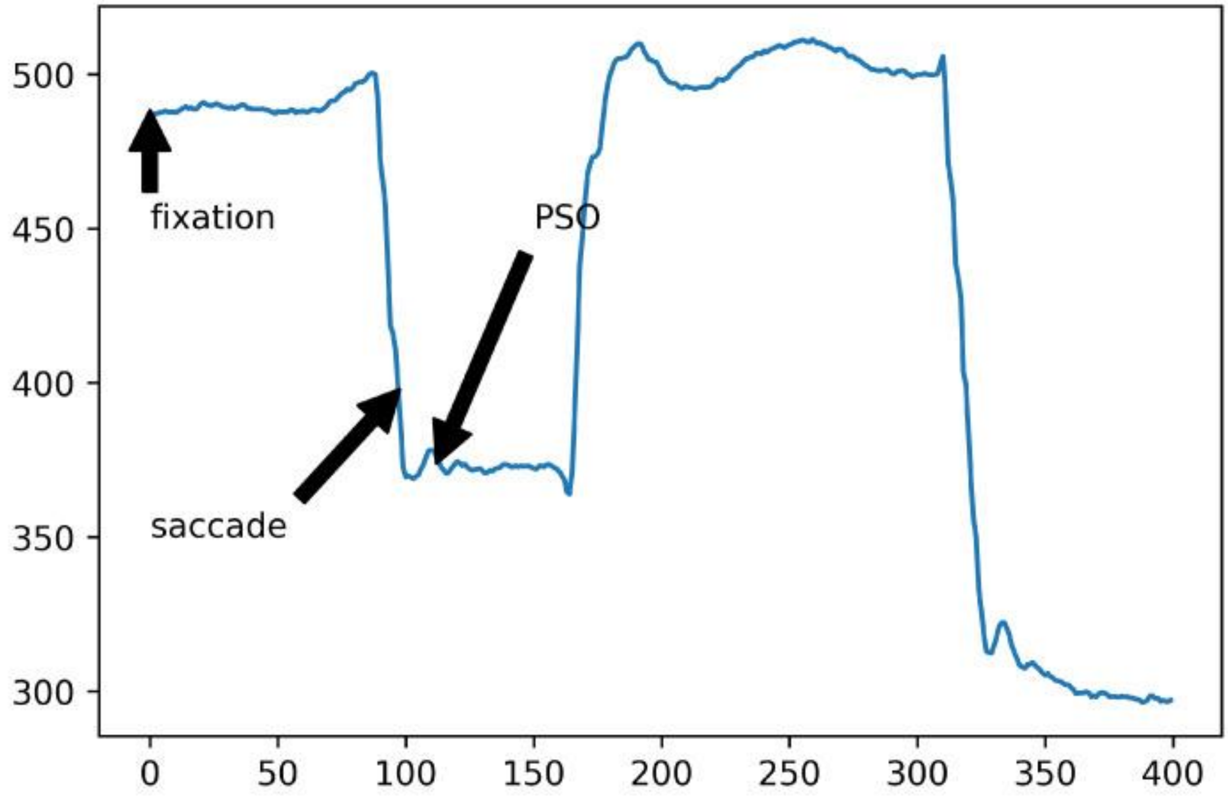


Figure 2: Eye movement event types.

Other eye tracking events that are not covered in this paper include post-saccadic oscillations and glissades, where the eye either overshoots or undershoots the target of fixation, respectively. Smooth pursuit is when the eyes fixate on a target that is moving slowly. Vergence movements are when the eyes do not converge on a target, but rather diverge. Vestibular-ocular reflex is when the eyes move opposite to the head. Nystagmus is a phenomenon where the eyes move randomly. Lastly, pupillary response is another measure that can be obtained using eye tracking and can provide more information about the visual stimuli.

Problem Statement

The main issue with classifying these kinds of events is there is no good way to obtain the ground truth for these events. Often times, human coders will manually classify the eye tracking events based the descriptions above. Another common technique is to use thresholds. I-VT is a speed based threshold, that classifies an eye movement as a fixation is the speed between two gaze points is less than 0.5px/ms, otherwise it is a saccade. I-DT is a dispersion based threshold which calculates the dispersion between points in a given window. If the dispersion is less than 1 degree, it is classified as a fixation. If the dispersion is greater than 1 degree, it is classified as a saccade. These thresholds are generally good in practice, but they can only classify fixations and saccades – not other events.

Therefore, the purpose of this project is to develop an unsupervised machine learning pipeline that will take raw gaze data and attempt to classify eye tracking events. Rather than using supervised models, which would rely on having the ground truth of the event classification, the unsupervised models will try to learn from the provided features if they are fixations or saccades. The success of the models will be determined by comparison to the threshold classifications, and by use of the silhouette score metric for unsupervised learning algorithms.

Method

Dataset and Preprocessing

Dataset

The dataset used for this project comes from a paper written by B. Caren and E. Ziraldo at the University of Guelph DRiVE lab, on the comparison of visual fixations in various driving hazards[1]. The study performed consisted of 72 participants. Each participant completed a 15 minute drive wearing Tobii Pro 3 eye tracking glasses. The glasses recorded data at 60Hz, so per participant, there are 20000 records in each eye tracking dataset. In total, there is roughly 1.5 million records.

The dataset also contains 21 features. The features include a timestamp, the gaze position in 2D and 3D, the gaze origin and gaze direction for both the left and right eye in 3D space, as well as the pupil diameter for both eyes.

Preprocessing

This dataset is available in the form of Excel files. Each excel file has three sheets. One contains device information, the second one contains the eye tracking measures above, and the third contains IMU data for device orientation and positioning. Using the pandas library in Python, these files are read into panda dataframes.

Initially, the data is split into training, testing and validation, using a 70-15-15 split. Presplitting the dataset ensures reproducibility in the final models and ensures no leakage between participant data or among the other categories.

Before implementing PySpark, the pandas interpolate function is used to fill in missing values in the data. Linear interpolation is the most common technique used for filling in missing values. Next the data needs to be organized so that each record in the final dataset contains two gaze points per record. This way, each record can have an associated speed and dispersion value, as well as a class, among others. To do this, two copies of the dataframe are created. The last record is removed from the first dataframe and the first record is removed from the second dataframe. The two are then aligned side-by-side so that the dataset has twice as many features.

[1] B. Caren, E. Ziraldo.“Comparing Visual Fixations between Initially Stopped and In-motion Turn Across Path Hazards”, Society of Automotive Engineers. 04/11/2023.

Once merged, the dataframe is converted to PySpark, with the pandas API. Then the EDA can be performed using PySpark but using panda functions.

Modelling

Because the main objective is to compare the performance of the unsupervised models not to themselves, but to the threshold techniques, only two different machine learning models were selected. K-Means and Agglomerative Clustering were selected based on their performance and simplicity.

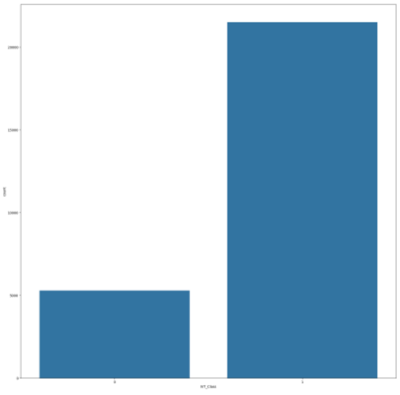
Dimensionality reduction is performed before the clustering as well to reduce the number of features down to 2. PCA and t-SNE were both used to reduce the number of features, while maintaining as much information as possible. The K-Means and Agglomerative clustering was then trained on both of these reduced datasets.

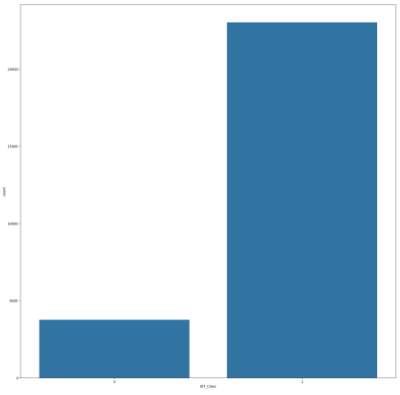
Results and Discussion

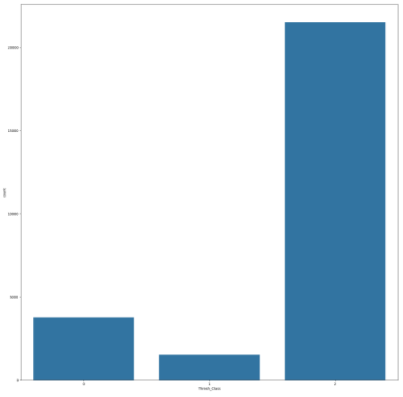
EDA Results

For the EDA, the first stage consisted of finding the distribution of classes based on the threshold techniques. Initially, the fixations are identified by a label of 0, and the saccades are identified by a label of 1. Below are three figures. The first one is the class distribution based on the I-VT threshold. The second one is the class distribution based on the I-DT threshold. The final one is the agreement between the models. From this, we then have three labels in total. 0 for when both I-VT and I-DT classify the record as a fixation. 2 identifies when both I-VT and I-DT classify the record as a saccade. 1 identifies when the labels are inconsistent between the two models. The machine learning models are trained on this multi-class label.

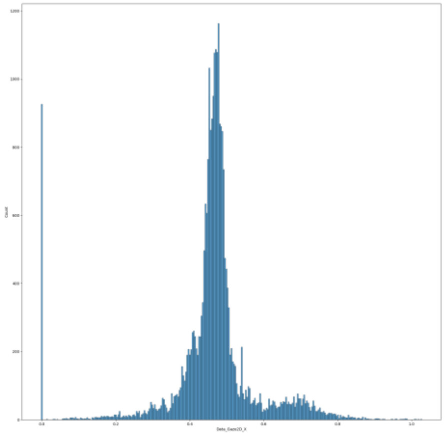
From these initial graphs, it is found that most of the records are saccades, while about a quarter of the samples are fixations. Since there is so much data of both classes, and the fixations are not a huge minority, the data is left as is and is not balanced. However, in future models, the data may need to be balanced to produce better results.

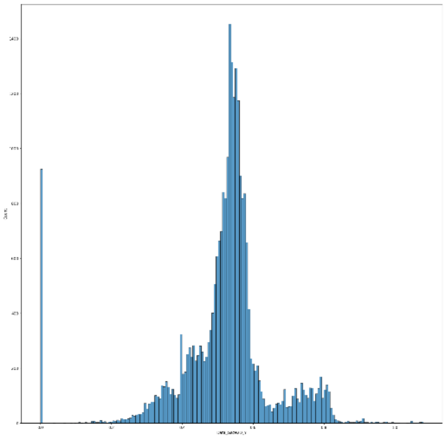


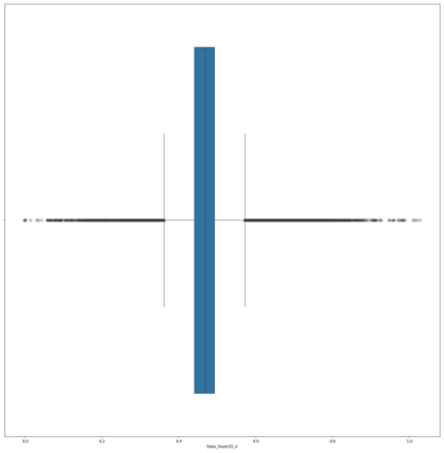


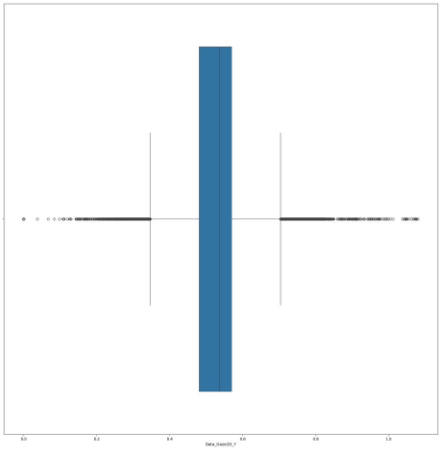


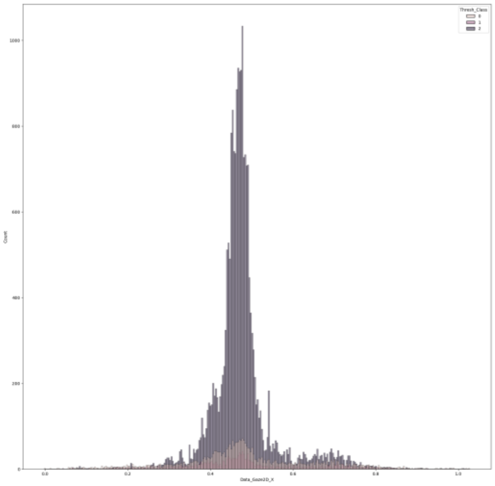
Next, the distribution of the 2D gae data is plotted. From these visualizations below, there is a spike at the point (0,0), which is most likely a result of the software using (0,0) as a null or default point. This point is removed from the data since it is a strong outlier. The box plots for these features are also plotted. The outliers are retained as they provide valuable information for the data. There is a large amount of data outside the fences of the plots that would be lost if removed. Further on in the EDA analysis, the class distribution is also plotted over the gaze. It is also interesting to note that in the y data, there is the peak in the center, and a smaller peak to the right that represents when the driver is looking at the dashboard of the vehicle. This is similar to the left of the peak, representing when the driver looks at the rear view mirror. The x data is wider on the right hand side of the peak, which suggests that the driver must have a further distance to look even with a head movement to view the right hand side mirror.

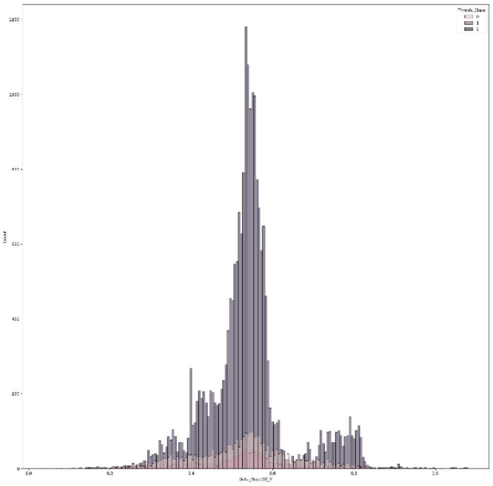




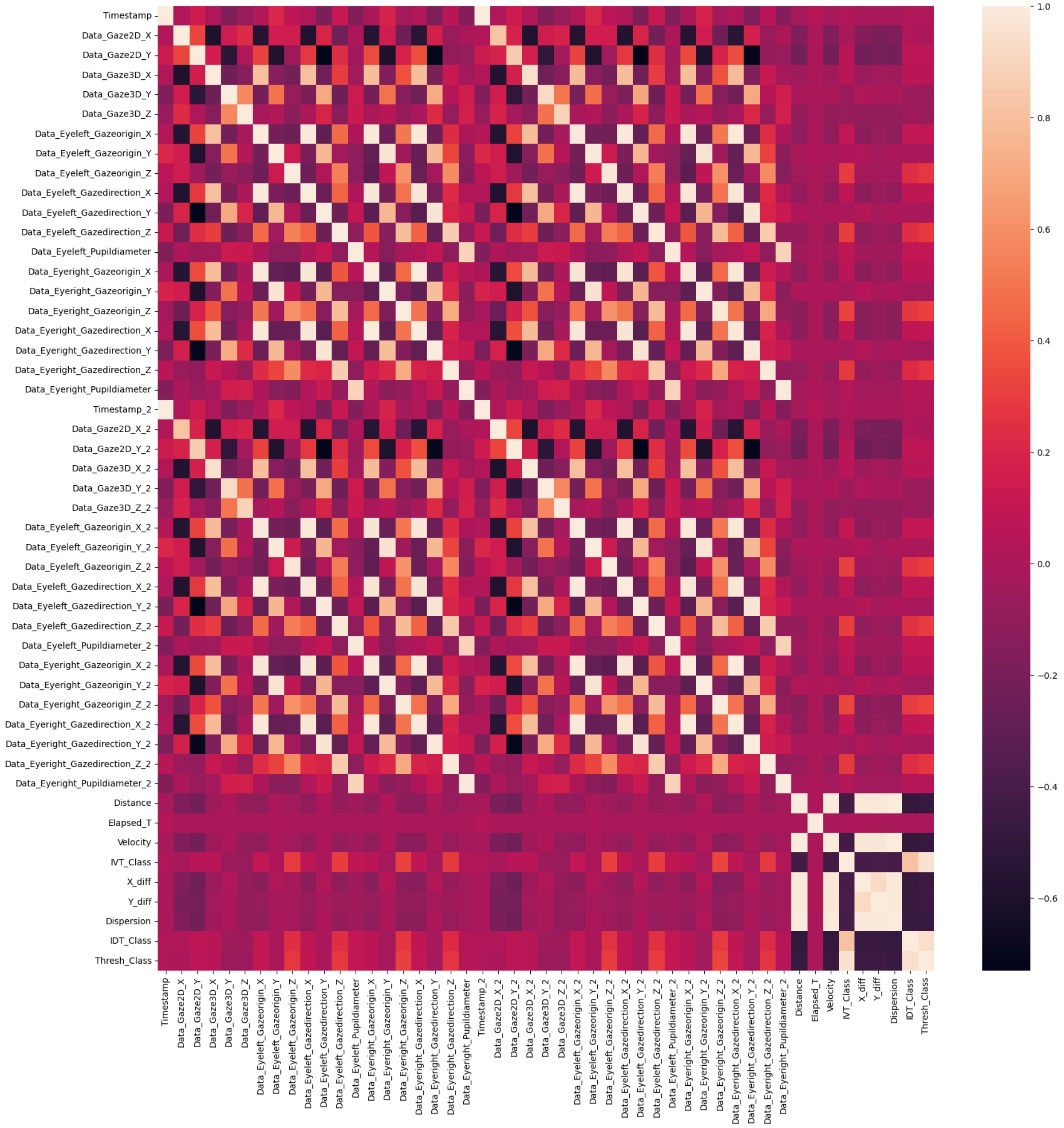




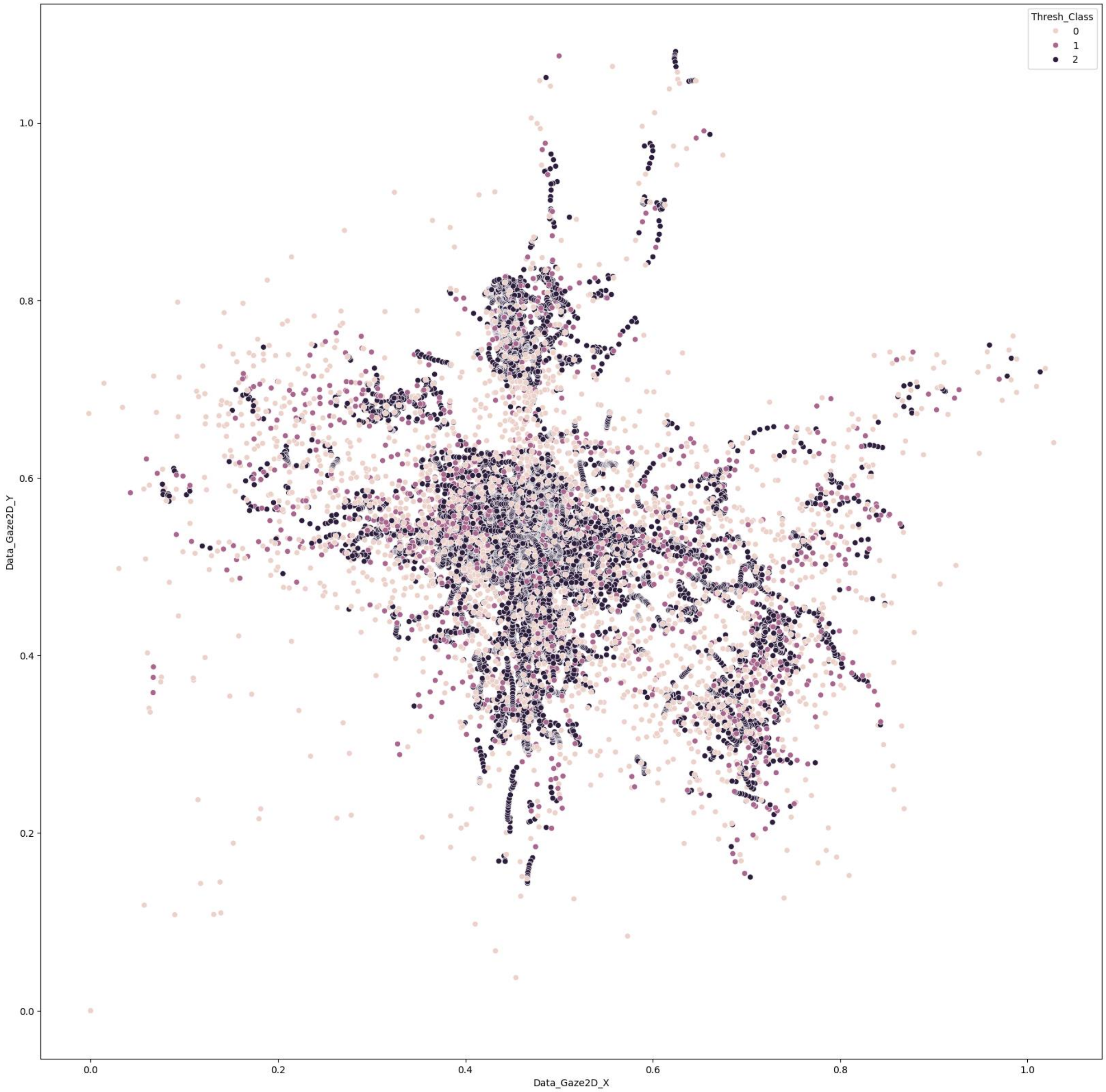


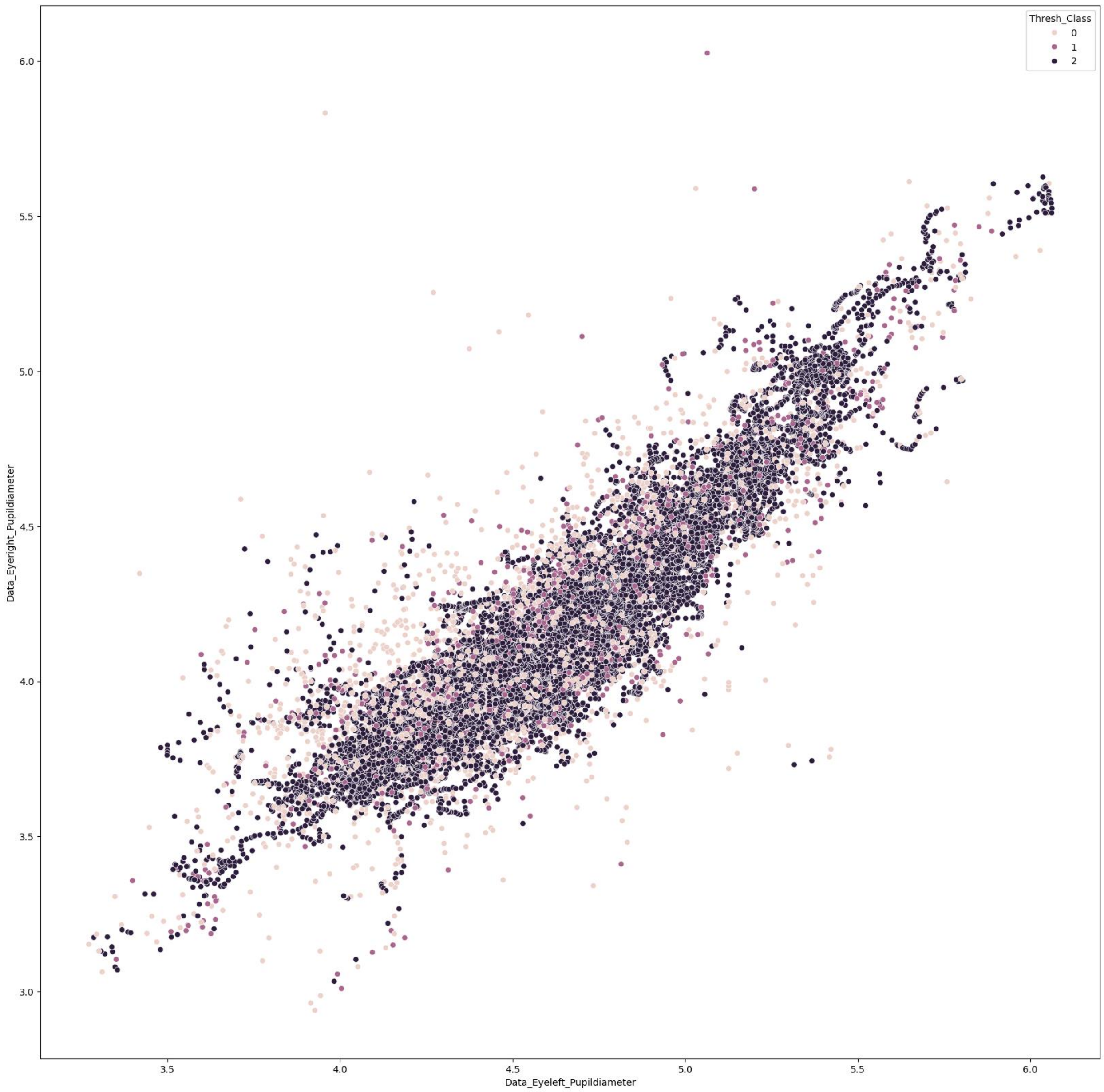


Next, a correlation heatmap is generated between all the features. From this plot, there are a few things that are noticed. First, is that there is a correlation between the coordinate of the first gaze point and the second gaze point in the records. This is most likely due to the fact that the points are copied over in the initial preprocessing, rather than an actual correlation between the two gaze points. One other interesting discovery is that the left and right pupil diameter does seem to be correlated between each other. This makes sense because the pupils should respond similarly to the same visual stimuli. The labels are also highly correlated with the engineered features, which makes sense because the class labels are derived from the engineered features.



In addition to the distributions from previously, the gaze data was mapped using a seaborn scatterplot. The class labels were overlayed using the hue of the plot, to identify any other patterns in the gaze data. The figure shows a slight boundary between encasing the saccades within the points classified as fixations. However, this boundary is not a hard boundary.

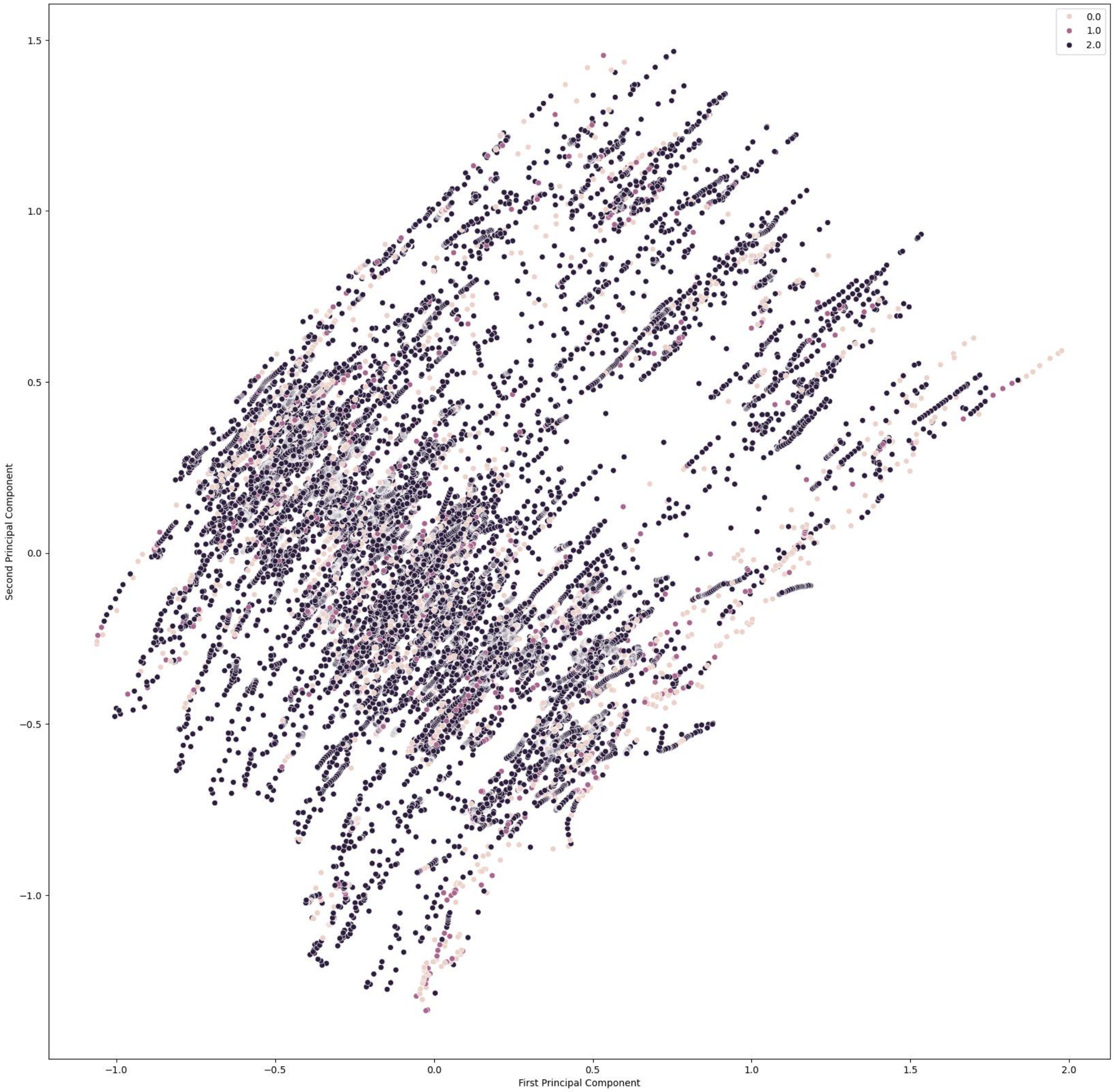




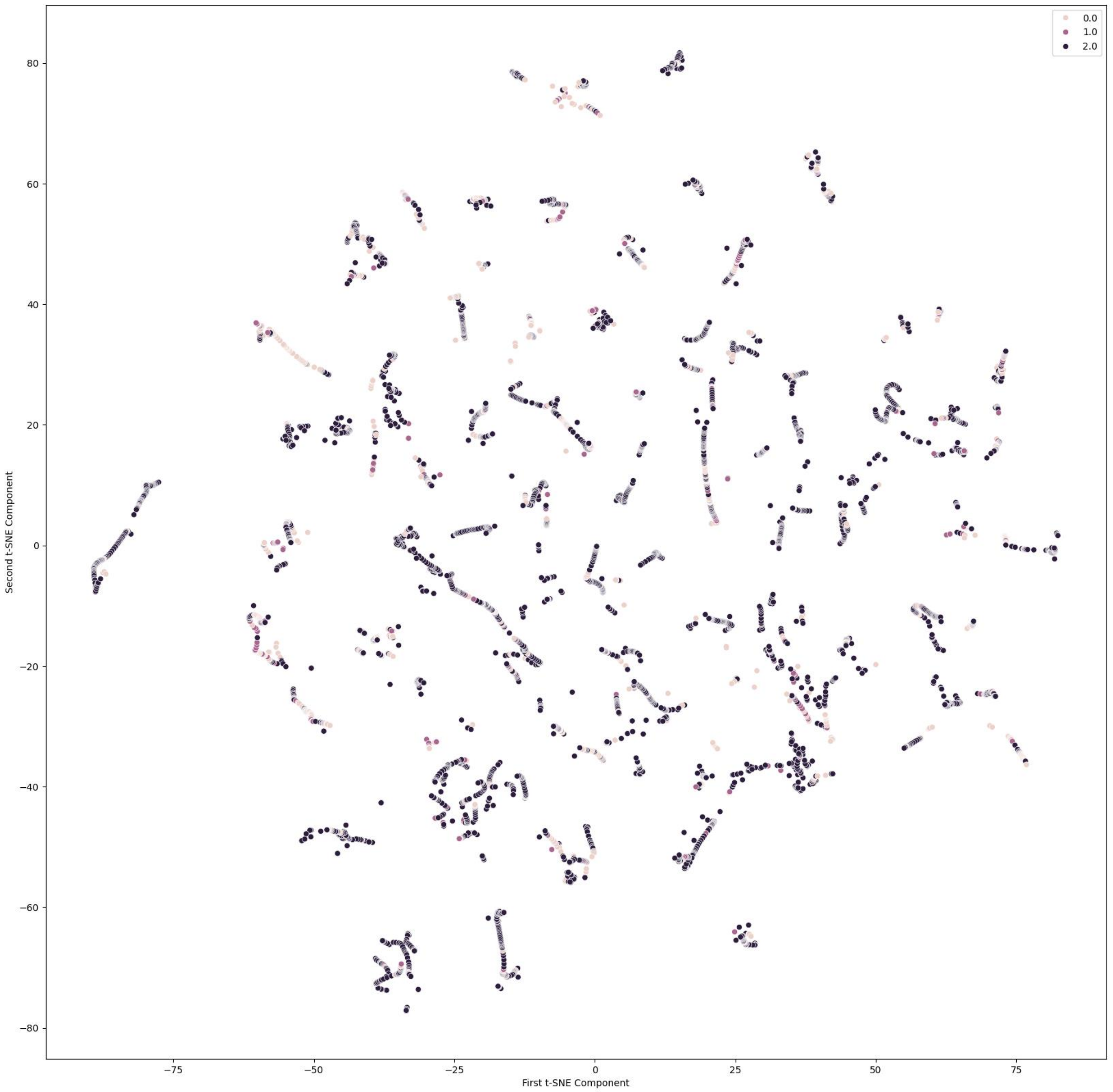
The pupil response is also mapped using a scatter plot. The hue is overlayed again to determine any patterns that may correlate to the label. For this, the pupil response does not appear to show any valuable patterns.

Machine Learning Results

First, the dimensionality reduction results are plotted using scatter plots to view the success of the reduction. For these, it is difficult to analyze because the components are abstractions of the original data, so they do not directly represent one specific feature in the dataset. The first figure is the result from the PCA. From analyzing the PCA, the data does not cluster particularly well. It does appear as though the fixations fall to the bottom of the graph, but several of the fixations are still spread out throughout the dataset.

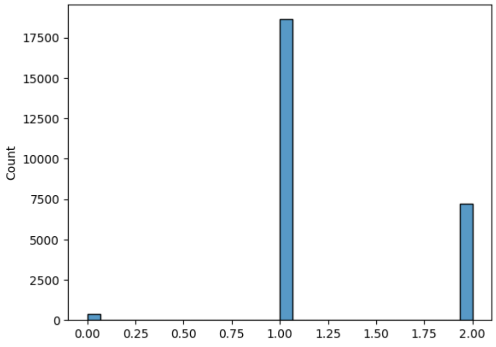
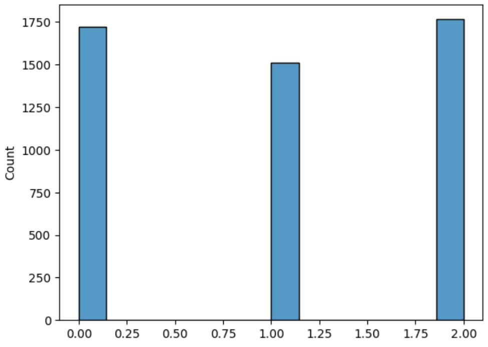
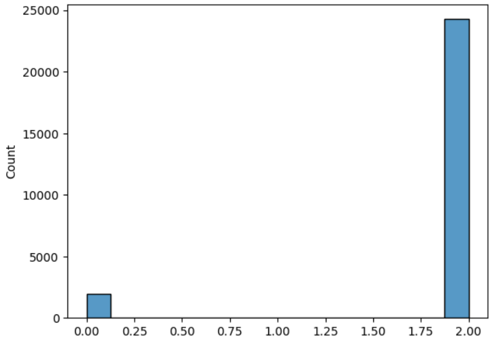
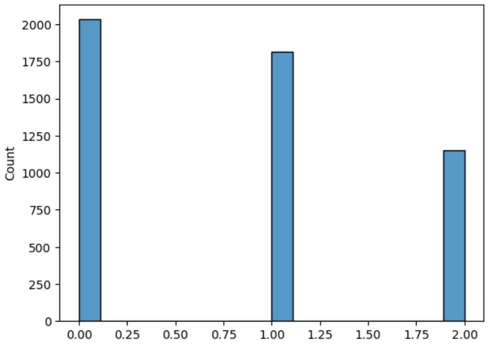


The other dimensionality reduction technique used was t-SNE. T-SNE was not performing as well as expected in terms of its processing performance, so only a subset of the data was used, as more data would be producing similar results. The resulting figure shows more linearly separable clusters that would appear better for the classification.

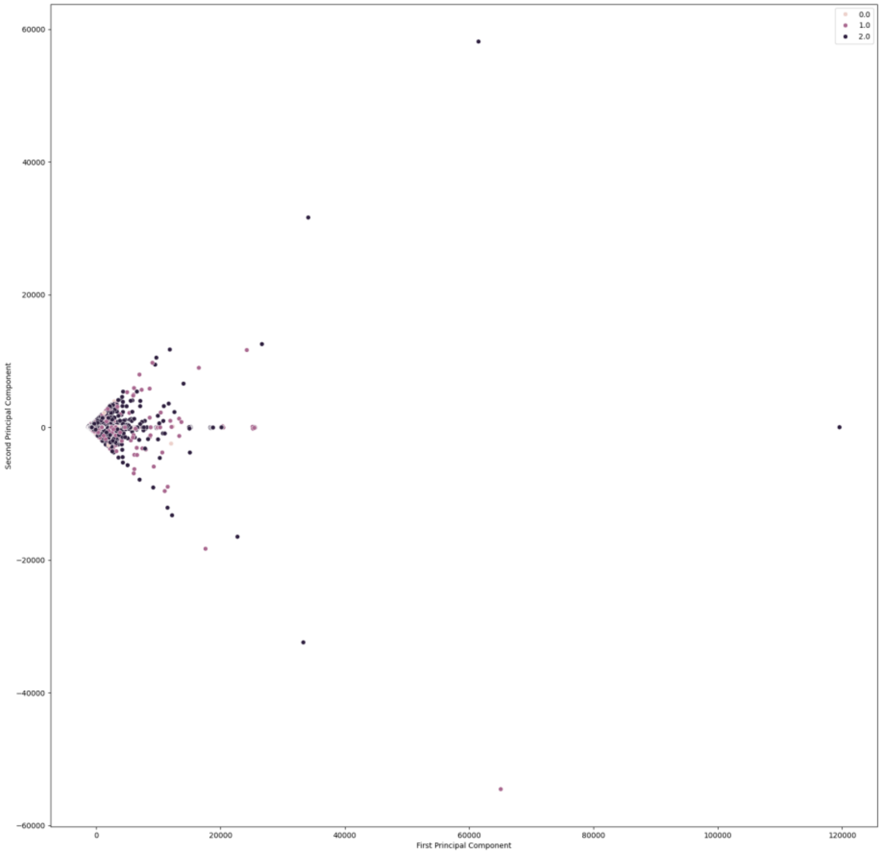


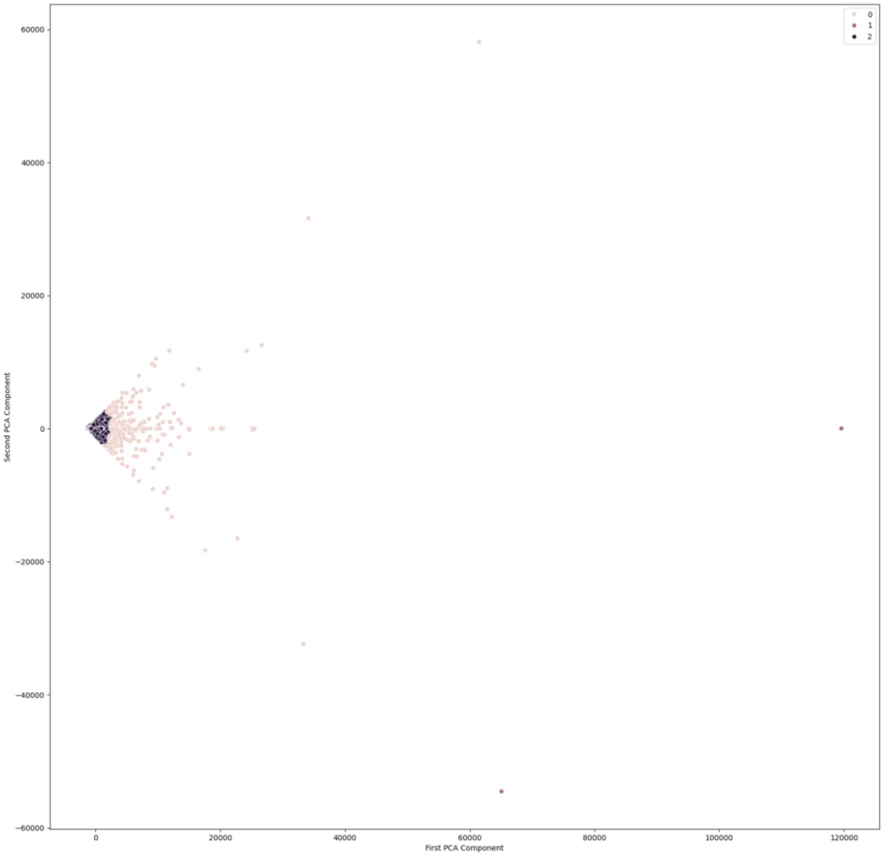
Once the dimensionality reduction was performed, the models were trained and tested on the PCA and the t-SNE features. Silhouette scores were calculated for the models and resulted in the following table. From the table, the PCA is identified as the stronger method of reduction compared to t-SNE. The agglomerative clustering was identified to be better than the K-Means implementation. After obtaining these scores, the results of the clusters was plotted on a scatter plot to compare back to the original PCA plot.

	PCA	t-SNE

K-Means	 SS=0.600	 SS=0.373
Agglomerative Clustering	 SS=0.801	 SS=0.325

Presented below is a scatter plot of the test PCA data and the predicted labels based on the clusters selected. The first figure is the PCA of the test data. The second figure is the predicted data from the clustering techniques. From this visualization, it appears as though the test data is not clustered correctly.


0.0
1.0
2.0
60000
40000
20000
0
−20000
−40000
−60000
0
20000
40000
60000
80000
100000
120000
First Principal Component
Second Principal Component


0
1
2
60000
40000
20000
0
−20000
−40000
−60000
0
20000
40000
60000
80000
100000
120000
First PCA Component
Second PCA Component

Conclusion

It is evident that while the evaluation of the model metrics demonstrated feasibility of the models, a review of the test data compared to the results of the clustering proved ineffective. This could be due to the PCA of the test data being dramatically different than the training data and the clusters were not reviewed thoroughly. So while the models are plausible, more work needs to be done for improvement.

For future work, more EDA would be required to train the model better. Another challenge was implementing PySpark. If this project was to be redone, pandas would be used solely. If the large dataset was an issue, a proper distributed computing network such as Sharcnet would be used. In terms of next steps, deep learning may be an interesting model to use in order to produce better results.