Project-1
Report Questions

1. What are the classification results for each experiment? (Create a table of this – sample template available at end of document)

Classifier/ DataType	**Original Data**			**Translated**			**Rotated X | Y | Z**		
Metric	**Accuracy**	**Precision**	**Recall**	**Accuracy**	**Precision**	**Recall**	**Accuracy**	**Precision**	**Recall**
Random Forest	34.35	34.03	34.42	49.35	48.25	49.32	34.49	33.87	34.57
SVM	47.89	50.28	48.01	55.56	57.42	55.46	48.68	51.14	48.73
Decision Tree	28.27	27.79	28.33	36.98	36.69	37.03	27.63	27.94	27.31

2. Which of the classifiers worked the best for each data type (original, translated, rotated)? Why? If multiple had the same, then why did this happen? Note: this question is left intentionally vague. What does work best mean? Which metric is best? Think about what we've talked about in class in regards to expressions and emotion.

SVM demonstrated superior performance compared to the other two classifiers across all data types: original, translated, and rotated (Rotate X, Rotate Y, Rotate Z).
For SVM the accuracy score was highest for the translated data type, and second comes the Rotated (Rotated X,Y,Z) and then original .

SVM has the capability to effectively handle high-dimensional data and its robustness to noise and variations in the dataset, which might explain why it performed better than the other classifiers, especially with translated data. It's important to note that while accuracy is a common metric for evaluating classifiers, other metrics like precision, recall, and F1-score can provide additional insights into classifier performance, especially in emotion recognition tasks. It's important to choose the right metric based on what we're trying to do. So, when picking a way to measure how well a classifier works, we need to think about what we're using it for. Moreover, SVM's robustness to noise and its ability to handle non-linear relationships between features might have played a role in its success, particularly in scenarios where the data underwent translation or rotation.

3. For the top classifier, for each data type, describe the misclassification for each classifier. What was misclassified as what (e.g., sad looks like happy)? Based on your intuition about how expressions look, do the misclassifications make sense (i.e., do you think the expressions are similar – not based on any specific example)?

Top classifier is SVM:
Original Data: Running SVM Classifier on Original Data,

- A total of 3652 subjects related to "Angry" misclassified as "Sad" emotion.
- A total of 1272 subjects related to "Disgust" misclassified as "Fear" emotion.
- A total of 1782 subjects related to "Fear" were misclassified as "Happy" emotion.
- A total of 1401 subjects related to "Happy" were misclassified as "Fear" emotion.
- A total of 2242 subjects related to "Sad" misclassified as "Angry" emotion.

• A total of 1349 subjects related to “Surprise” were misclassified as “Fear” emotion.
Translated Data: Running SVM Classifier on Translated Data,
• A total of 2857 subjects related to “Angry” misclassified as “Sad” emotion.
• A total of 1672 subjects related to “Disgust” were misclassified as “Fear” emotion.
• A total of 1412 subjects related to “Fear” were misclassified as “Happy” emotion.
• A total of 1503 subjects related to “Happy” misclassified as “Fear” emotion.
• A total of 2051 subjects related to “Sad” misclassified as “Angry” emotion.
• A total of 893 subjects related to “Surprise” misclassified as “Fear” emotion.
Rotated X Data: Running SVM Classifier on Rotated X Data,
• A total of 3605 subjects related to “Angry” misclassified as “Sad” emotion.
• A total of 1194 subjects related to “Disgust” misclassified as “Fear” emotion.
• A total of 1774 subjects related to “Fear” were misclassified as “Happy” emotion.
• A total of 1432 subjects related to “Happy” misclassified as “Fear” emotion.
• A total of 2172 subjects related to “Sad” misclassified as “Angry” emotion.
• A total of 1315 subjects related to “Surprise” misclassified as “Fear” emotion.
Rotated Y Data: Running SVM Classifier on Rotated Y Data,
• A total of 3613 subjects related to “Angry” misclassified as “Sad” emotion.
• A total of 1161 subjects related to “Disgust” misclassified as “Fear” emotion.
• A total of 1751 subjects related to “Fear” misclassified as “Happy” emotion.
• A total of 1435 subjects related to “Happy” were misclassified as “Fear” emotion.
• A total of 2159 subjects related to “Sad” misclassified as “Angry” emotion.
• A total of 1278 subjects related to “Surprise” misclassified as “Fear” emotion.
Rotated Z Data: Running SVM Classifier on Rotated Z Data,
• A total of 3619 subjects related to “Angry” misclassified as “Sad” emotion.
• A total of 1129 subjects related to “Disgust” misclassified as “Fear” emotion.
• A total of 1729 subjects related to “Fear” misclassified as “Happy” emotion.
• A total of 1472 subjects related to “Happy” misclassified as “Fear” emotion.
• A total of 2179 subjects related to “Sad” were misclassified as “Angry” emotion.
• A total of 1279 subjects related to “Surprise” misclassified as “Fear” emotion.

Across different data types and rotations, the SVM classifier consistently showed certain misclassification patterns. In the original dataset, a notable trend emerged with 3652 instances labeled as "Angry" being misclassified as "Sad," while 2242 instances labeled as "Sad" were mistakenly classified as "Angry." Furthermore, 1272 instances labeled as "Disgust" were inaccurately categorized as "Fear," and 1401 instances labeled as "Happy" were erroneously classified as "Fear." Similarly, 1349 instances labeled as "Surprise" were misclassified as "Fear."

Transitioning to the translated dataset, similar misclassification trends persisted. For instance, 2857 instances labeled as "Angry" were misclassified as "Sad," and 2051 instances labeled as "Sad" were incorrectly categorized as "Angry." Additionally, 1672 instances labeled as "Disgust" were mistakenly classified as "Fear," and 1412 instances labelled as "Fear" were inaccurately categorized as "Happy." Moreover, 1503 instances labeled as "Happy" were misclassified as "Fear," while 893 instances labeled as "Surprise" were incorrectly classified as "Fear."

Rotated data, whether along the X, Y, or Z axis, showcased consistent misclassification patterns. For example, in Rotated X data, 3605 instances labeled as "Angry" were misclassified as "Sad," and 2172 instances labeled as "Sad" were mistakenly classified as "Angry." Additionally, 1194 instances labeled as "Disgust" were inaccurately categorized as "Fear," and 1432 instances labeled as "Happy" were misclassified as "Fear." Similarly, 1315 instances labeled as "Surprise" were misclassified as "Fear." These trends persisted across Rotated Y and Rotated Z datasets as well. enduring difficulty in accurately recognizing these emotions solely from the 3D facial landmark data, regardless of the orientation of the face.

4. Why do you think you got the results that you got for each of the different data types/classifiers (i.e., why are they different, or why are they the same)? For example, if SVM and RF have different results, why are they different? If they are the same – why are they the same?

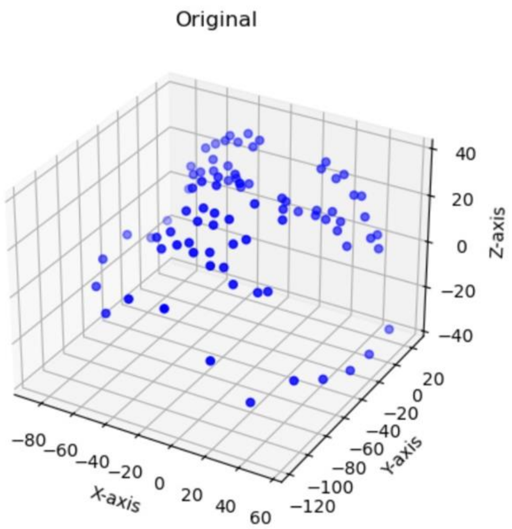
In terms of classifier comparison, it has been observed that SVMs perform more effectively in high-dimensional spaces. This is particularly beneficial when working with datasets that possess a large number of features, such as 3D facial landmarks, which have 249 features in this case. On the other hand, Random Forest's ensemble nature may not have produced the desired results as it relies on multiple decision trees. It is worth noting that decision trees may not be able to capture complex relationships in the data as effectively as SVM's optimization for finding the optimal hyperplane.

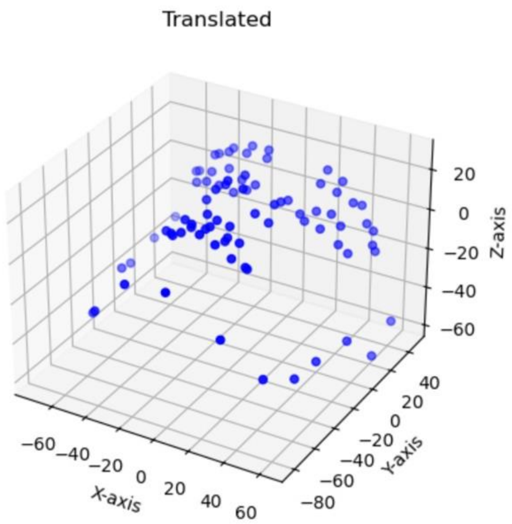
Furthermore, SVMs with non-linear kernel provide the advantage of being able to capture complex patterns in the data by mapping the input features into a higher-dimensional space. This allows SVMs to learn intricate decision boundaries that may not be achievable with simple decision trees. SVMs are also less prone to overfitting compared to decision trees, particularly when the number of features is large, as decision trees tend to over fit the training data.

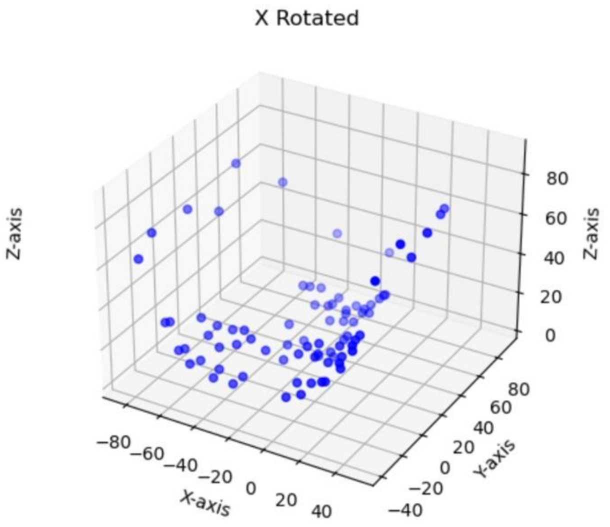
In terms of data type comparison, it has been observed that the accuracy for translated data is higher than that of both rotated and original data. Translating the data to the origin (0, 0, 0) effectively normalizes positional variations across different samples. This normalization process can enhance the separability of different classes and improve classification accuracy.

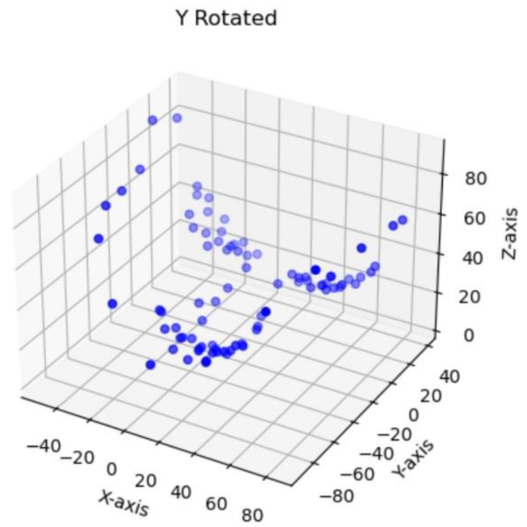
It is also noted that rotating the data or using the original coordinates may introduce additional complexity or non-linearity that could negatively impact classifier performance.

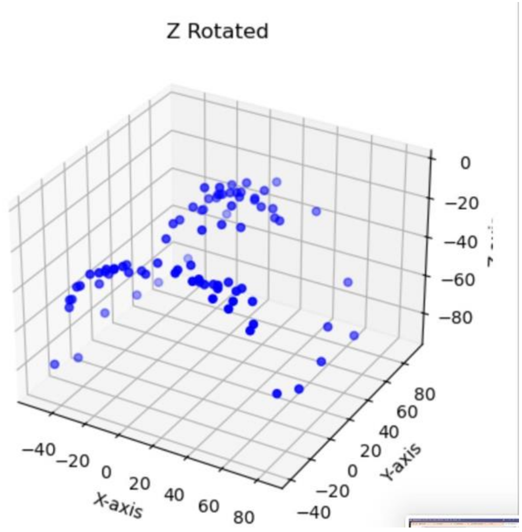
5. You must plot one sample of each data type – original, translated, rotated (1 each for x, y, and z). In total, you will have 5 figures/plots. Note: if you are unsure how to do this, look up 3D scatter plot in python.











6. (CAP 5627 only) Describe, in your own words, what Random Forest, SVM, and Decision Tree classifiers are/how they work. You may need to look this information up. Your description should be approximately 100-150 words in length for each classifier.

Random Forest:
Random Forest, an ensemble learning technique, is versatile for both classification and regression tasks. During training, it constructs numerous decision trees and aggregates their predictions to enhance accuracy and mitigate overfitting. Initially, it randomly selects subsets of data and features, building decision trees based on different aspects of the data. Each tree independently evaluates the target variable, and during prediction, the final outcome is determined through majority voting or averaging.
Example: In classifying fruit ripeness, Random Forest selects random subsets of both training data and features, and each decision tree focuses on various fruit attributes like colour, size, and sweetness. By combining the predictions of all trees, typically through majority voting, the classifier accurately

predicts fruit ripeness, showcasing the collective wisdom of diverse decision trees and ensuring robustness in prediction outcomes.

SVM:

SVM is a robust supervised learning method for classification and regression tasks. It finds the best hyperplane to separate data points in a high-dimensional space, aiming to maximize the margin between classes and minimize errors. Support vectors, the points closest to the decision boundary, play a crucial role in SVM's operation. During training, it learns hyperplane parameters using optimization methods like gradient descent. For classification, SVM maps new data points into the same space and determines their class based on their position relative to the hyperplane.

Example: let's say we have a dataset with apples and oranges, each characterized by color and size. SVM aims to draw a line (or hyperplane) that best separates apples from oranges in this feature space. This line is found by identifying support vectors, the closest fruits to the boundary. So, if we get a new fruit, SVM predicts whether it's an apple or an orange based on which side of the line it falls. This demonstrates how SVM works.

Decision Tree:

Decision trees are a popular machine learning algorithm used for both classification and regression tasks. They work by recursively splitting the dataset into subsets based on the most significant attribute, creating a tree-like structure of decisions. Each internal node represents a feature, each branch represents a decision based on that feature, and each leaf node represents the final decision or prediction. Decision trees are easy to interpret and visualize, making them useful for understanding relationships between variables in the data. However, they can be prone to overfitting with complex datasets.

Example: In classifying emails as spam or not spam, a decision tree might split based on the presence of certain words in the subject line or sender's email address. The tree continues to split the dataset into subsets until it reaches a point where further splits don't improve classification. New data is then classified by following the path of decisions in the tree, making decision trees easy to interpret and useful for understanding data patterns.

<u>Confusion Matrix for the Top Classifier(SVM):</u>

SVM for Original:

Emotions	Angry	Disgust	Fear	Happy	Sad	Surprise
Angry	5765	2369	1367	1021	3652	1210
Disgust	633	4057	1272	796	173	842
Fear	732	1469	3419	1782	1085	1621
Happy	201	552	1401	5474	89	110
Sad	2242	804	1226	522	4870	749
Surprise	529	912	1349	343	251	5394

SVM for Translated:

Emotions	Angry	Disgust	Fear	Happy	Sad	Surprise
Angry	6481	2087	1157	1212	2857	997
Disgust	582	4980	1672	715	161	625
Fear	681	1358	3818	1412	685	1280
Happy	98	701	1503	6043	45	259
Sad	2051	574	984	382	6339	831

Surprise	217	451	893	183	42	5941

SVM for Rotate X:

Emotions	**Angry**	**Disgust**	**Fear**	**Happy**	**Sad**	**Surprise**
Angry	5947	2320	1332	1004	3605	1210
Disgust	652	3952	1194	772	87	801
Fear	772	1519	3472	1774	1094	1688
Happy	113	615	1432	5515	56	87
Sad	2172	819	1272	539	5039	756
Surprise	449	927	1315	342	247	5383

SVM for Rotate Y:

Emotions	**Angry**	**Disgust**	**Fear**	**Happy**	**Sad**	**Surprise**
Angry	5975	2352	1342	1004	3613	1217
Disgust	661	3910	1161	732	89	789
Fear	789	1545	3542	1751	1083	1710
Happy	87	611	1435	5601	43	82
Sad	2159	839	1249	539	5059	749
Surprise	423	852	1278	334	242	5372

SVM for Rotate Z:

Emotions	**Angry**	**Disgust**	**Fear**	**Happy**	**Sad**	**Surprise**
Angry	5979	2343	1342	1002	3619	1215
Disgust	661	3911	1129	727	99	781
Fear	779	1578	3569	1729	1071	1712
Happy	79	621	1472	5616	39	76
Sad	2179	851	1226	534	5054	752
Surprise	415	832	1279	329	224	5359