Project 1 Report

Following are the results for subject independent 10-fold cross validation (rounded off to 3 decimal places)

Data_type	Original Data			Translated Data		
Classifier	Accuracy	Precision	Recall	Accuracy	Precision	Recall
RF	**0.399**	**0.402**	**0.400**	**0.413**	**0.418**	**0.413**
SVM	0.398	0.401	0.398	0.403	0.403	0.405
TREE	0.293	0.293	0.299	0.300	0.300	0.300

Rotated	X			Y			Z		
Classifier	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall
RF	**0.393**	**0.398**	**0.393**	**0.405**	**0.408**	**0.406**	**0.391**	**0.392**	**0.392**
SVM	0.388	0.389	0.392	0.393	0.393	0.394	0.383	0.383	0.384
TREE	0.310	0.305	0.311	0.301	0.296	0.302	0.305	0.297	0.305

2. The following classifiers have worked best for different data types
Original data - Random Forest (Precision = 0.402)
Translated data - Random Forest (Precision = 0.418)
Rotated data - Random Forest (Precision = 0.398 | 0.408 | 0.392)
Random Forest worked best as it is an ensemble technique and voting in RF can beat overfitting better than other models. SVM has slightly lesser values, maybe because the data is not linear. Decision Tree is the least accurate of all, as it is a simpler model and could not capture the patterns well.
Works best means the model evaluation has given better metric values/ results and less error as compared to other classifiers. These metrics tell us how well the model is able to make predictions. Coming to the metrics, precision is the best suitable metric for evaluation for expression recognition, than accuracy. While accuracy gives an overview of the entire data, precision focuses on positive predictions only. Misclassification of a negative emotion (like Angry, Sad, Pain, Depressed) as a positive emotion (Happy, Neutral, Joy) can cost a lot in case expression recognition (specifically when used for medical purposes). Hence it is important to check precision scores.

3. Random Forest is the top classifier for all data types.
Original Data - Fear is mostly misclassified as surprise and vice versa (1302 and 1941 samples) It could be because we usually open our eyes and mouth wide open in both cases. Fear is highly misclassified as other emotions as the true positives are very low. Surprise has the most correctly classified samples (6452).

Actual → Predicted	Angry	Disgust	Fear	Happy	Sad	Surprise
Angry	4537	1754	1577	1082	2784	736
Disgust	1979	4089	1929	1289	1192	530
Fear	625	1482	1941	1258	1556	1302
Happy	549	785	956	4213	579	471
Sad	795	568	793	341	2397	567
Surprise	1639	1493	2848	1790	1960	6452

Translated Data - The model seems to perform best for the class Surprise with a high number of correctly classified instances (6837) and relatively low misclassifications (less than 10% of data samples). Sad also has a decent number of correct predictions (2751) but seems to be confused with other emotions like Disgust and Angry more often, could be because of the eyebrows being close to each other for all 3 emotions. There are many misclassified instances for Fear and Surprise, and the highest value in their rows doesn't correspond to the actual class Fear.

Actual → Predicted	Angry	Disgust	Fear	Happy	Sad	Surprise
Angry	4383	1451	1194	851	2518	852
Disgust	1775	3724	1799	1120	1267	925
Fear	817	1123	2369	1340	1127	756
Happy	646	869	1218	4665	452	295
Sad	976	885	813	195	2751	286
Surprise	1420	2013	2520	1688	1923	6837

Rotated Data - X-axis - The model's accuracy in predicting Happy has decreased (4144 correctly predicted instances) compared to the previous matrix (4665). Compared to the previous matrix, there's a noticeable improvement in correctly predicting 'Angry' and 'Disgust' emotions (4501 and 3926 correctly predicted instances)

Actual → Predicted	Angry	Disgust	Fear	Happy	Sad	Surprise
Angry	4501	1830	1498	966	2825	630
Disgust	1963	3926	1898	1366	1241	667
Fear	683	1341	2099	1530	1498	1114
Happy	536	924	1058	4144	471	540
Sad	783	479	795	333	2358	478
Surprise	1551	2586	2586	1520	1645	6522

Rotated Data - Y-axis -
Surprise continues to be the most correctly classified expression.
Y-axis rotation captures the nuances of disgust more effectively than the other rotations.

Actual → Predicted	Angry	Disgust	Fear	Happy	Sad	Surprise
Angry	4793	1700	1530	1106	2910	603
Disgust	1628	4011	1913	1282	1281	729
Fear	742	1475	2070	1230	1398	970
Happy	502	815	998	4474	470	501
Sad	783	464	756	466	2261	463
Surprise	1569	1600	2667	1301	1718	6685

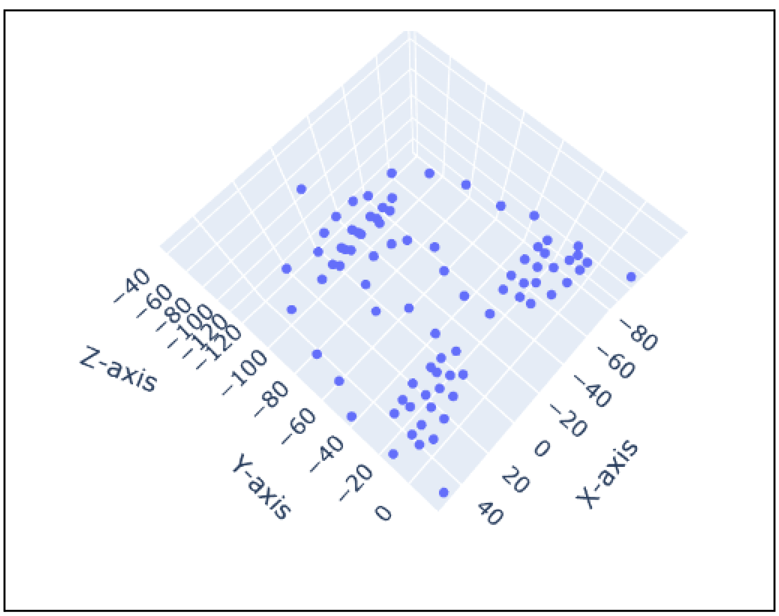
Rotated Data - Z-axis - Fear is the most misclassified compared to other rotations. Correctly classified samples have also decreased in Surprise class.
For all the rotations, we can notice that the model is able to distinguish between Happy and Sad properly.

Actual → Predicted	Angry	Disgust	Fear	Happy	Sad	Surprise
Angry	4603	1911	1558	1262	3035	782
Disgust	1597	3769	1747	1110	999	661
Fear	710	1475	1915	1313	1513	1008
Happy	405	919	1078	4456	557	291
Sad	1059	466	954	389	2234	531
Surprise	1643	1525	2682	1329	1700	6478

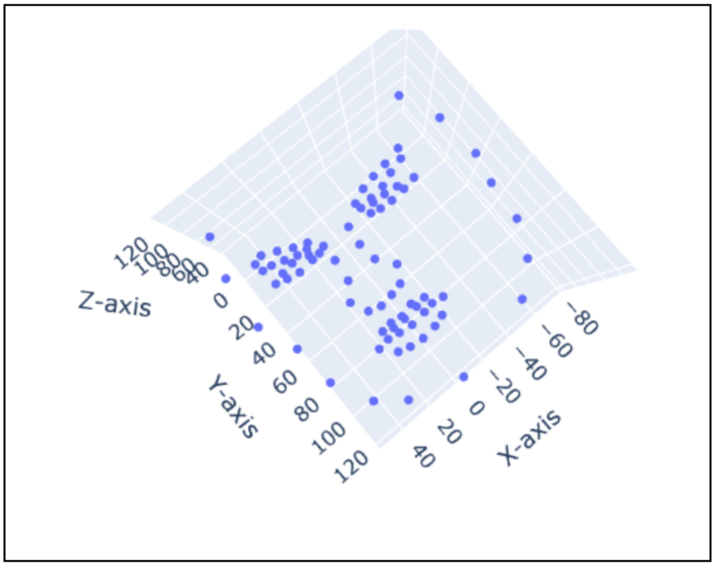
Overall, we can say the model is working well in detecting positive emotions - Happy, Surprise for all the data types. But the model has difficulty distinguishing between "Angry," "Disgust," and "Fear", evident from high misclassification values in these categories. We can also see a bias towards surprise, as most samples are mispredicted as Surprise.

4. Random Forest, being an ensemble method, can capture more complex relationships in the data and is more robust to noise and outliers compared to a single Decision Tree. We can see a slight difference in RF and SVM results. RF is slightly more robust to changes in data orientation than SVM, possibly due to its ensemble nature and ability to capture more complex patterns through multiple trees. RF is generally more versatile in handling both numerical and categorical data without the need for extensive preprocessing required by SVM, such as kernel transformations to handle non-linear data. The data has complex or non-linear relationships between features and emotions, and RF has the better ability to capture these complexities. Decision Trees are more prone to overfitting and it is single learner, hence the less accuracy and precision.

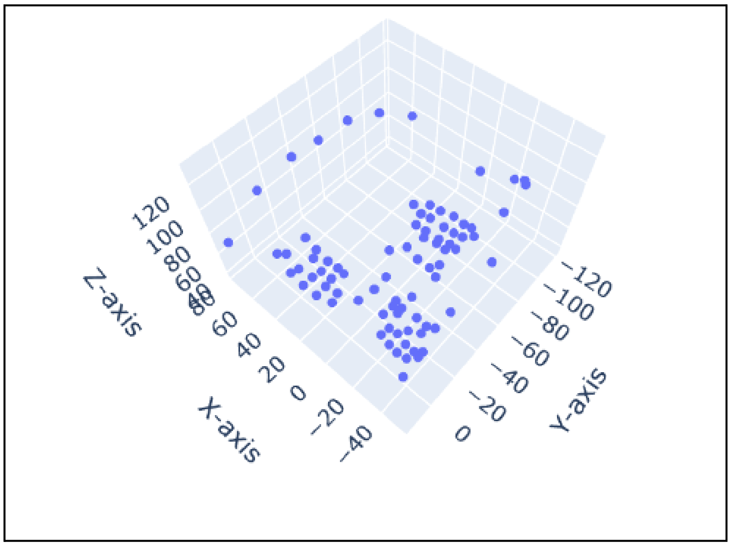
5. 3D Scatter Plot of a data sample for each data type



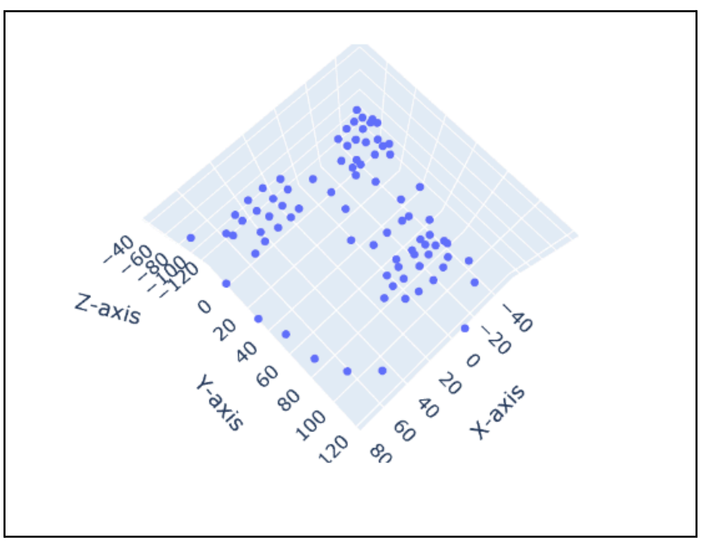
Original



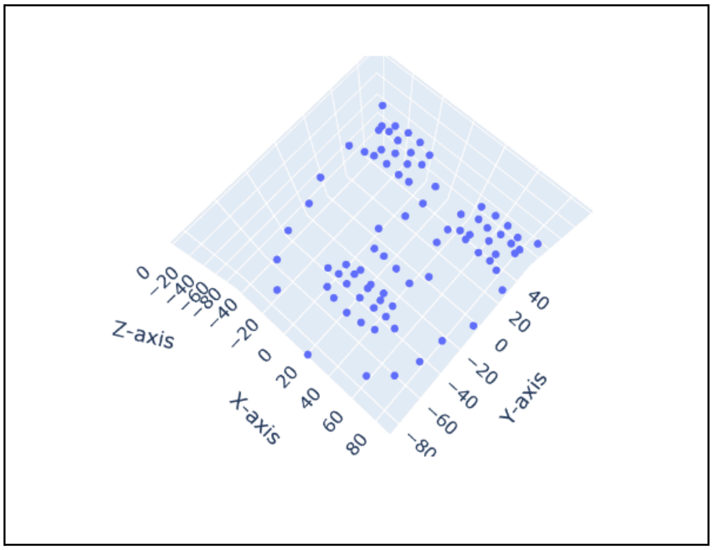
Rotate on x-axis



Rotate on Y-axis



Rotate on Z-axis



Translated data

6. Random Forest - It is an ensemble learning method that builds multiple decision trees during training. Each tree in the forest is trained on a random subset of the data and features. By combining the predictions of multiple trees through voting (for classification problems), Random Each tree in the forest has high variance due to the randomness injected during training. However, by averaging or voting, the final prediction tends to have lower variance, leading to more consistent performance across different datasets. Forest tends to reduce overfitting and improve generalization performance compared to a single decision tree.

SVM - In Support Vector Machine, We try to find the optimal hyperplane that separates the data into different classes with the largest possible margin. The hyperplane is a decision boundary that maximizes the margin, which is the distance between the hyperplane and the nearest data points from each class, known as support vectors. SVMs aim to maximize the margin not only improves the generalization performance of the model but also enhances its robustness to noise and outliers in the data. The advantage of SVM is, it can work well on smaller datasets as well. But, Training SVMs can become computationally expensive and time-consuming for very large datasets.

Decision Tree - This algorithm can be used for both classification and regression tasks. A decision tree consists of nodes and branches. The root node represents the entire dataset. Internal nodes contain conditions (questions) based on features. Leaf nodes (terminal nodes) represent the final predicted outcome (class in classification or value in regression). Branches connect nodes and represent the outcome (Yes/No or specific value) of the question at the parent node. We build a tree-like model by recursively splitting the data based on features (attributes) that best differentiate between classes. This algorithm is prone to overfitting and high variance.