EE552 Project 2

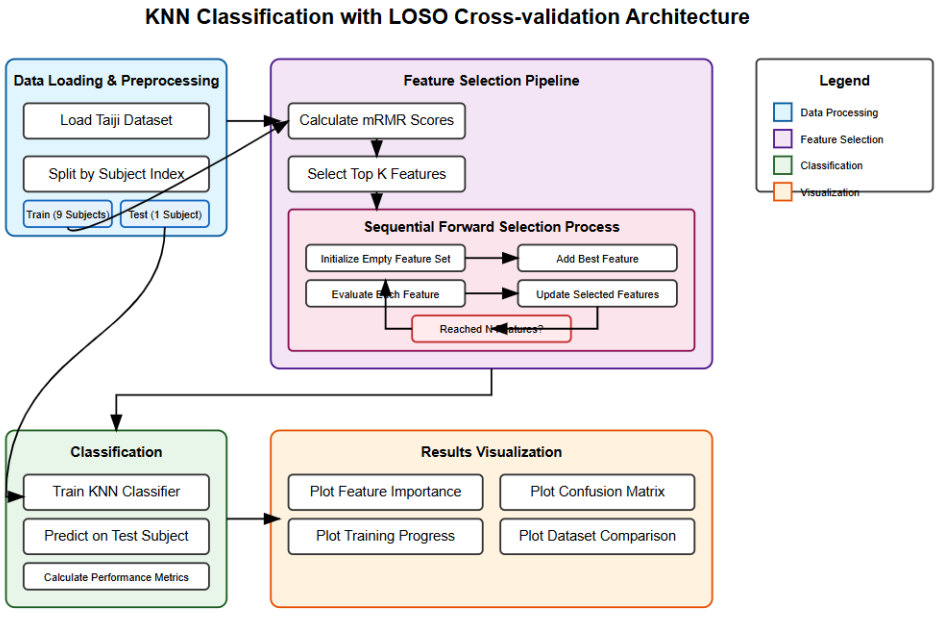
Azubuine Samuel Tochukwu

First Submission

Classification with Classical/ Traditional Classifiers

For this approach or first submission I initially choose the SVM classifier (partly because Non-linear classifier , selecting the RBF- radial basis kernel functions and probability parameter which no-doubt are computationally expensive) , also its Margin optimization for better generalization and handling of multi-class classification through one-vs-one approach) .However it was computationally expensive and mostly time consuming (It timed-out a couple of times, Hence I had to switch to another classifier.

I switched and adapted KNN base classifier in my **TraditionalClassifier** class(as the parent class) and it seemed a bit faster for my machine to train on/with. For this, I choose K = 5 and ran the classifier on the training set for N= 100, 200 and 300, iterating for the different subject indexes(indices), from i=1 to 9. Below is a simplified architecture of my implementation (See zipped code file too).



Implementation and structure:

I have an extra file I called utils for the case of separation of concern and modularity, that contains most of my helper/utility functions for plotting and computing minor operations/functions.

I essentially used KNN classification using Leave One Subject Out (LOSO) cross-validation and I plotted the feature importance for each iteration to synthesize the important features for that particular iteration, as well as the accompanying confusion matrix for that particular subject, (See for example below figure for N =1 and the first subject , and confusion matrix plot provided).

Description of each of major component of the architecture:

Data Loading & Preprocessing (Blue): Loads the Taiji dataset, implements LOSO by splitting data into 9 subjects for training and 1 for testing, and Handles data preprocessing and normalization

Feature Selection Pipeline (Purple): The primary feature selection using minimum Redundancy and Maximum Relevance- mRMR algorithm, th Sequential Forward Selection (SFS) process which Iteratively builds feature set, evaluates feature combinations and selects optimal features.

Classification Component (Green): Houses the TraditionalClassifier based on KNN classifier training on selected features, makes prediction on held-out test subject. Also handles performance metric calculation and results storage for analysis

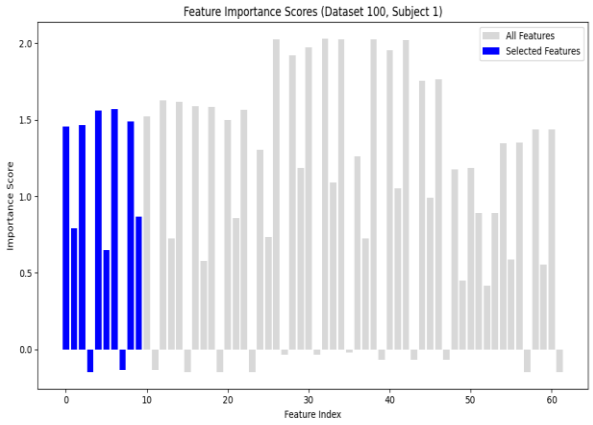
Visualization Component (Orange): Houses the comprehensive visualization suite implementations both for confusion matrix plotting, training progress tracking, feature importance heatmap for each N and subject as well as dataset comparison across subjects.

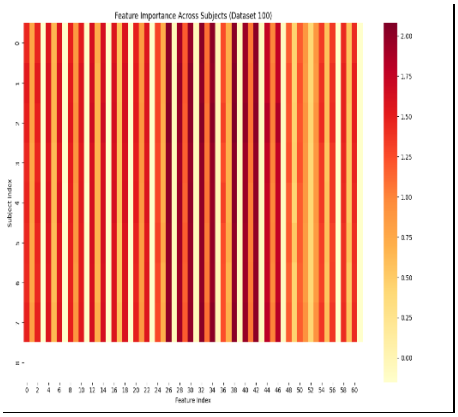
Training and Testing Analysis with LDA (Fisher Projection) For N = 100, 200, 300

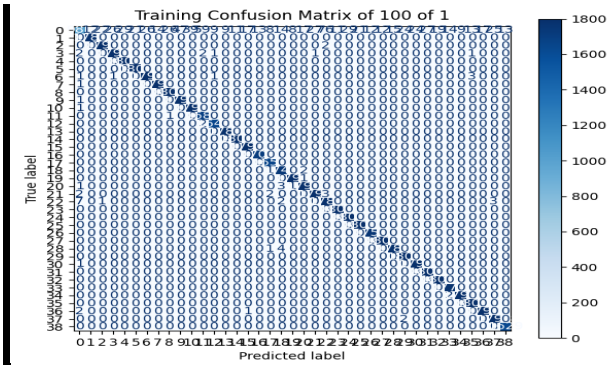
For N=100; SET 1(Key Components): First I trained the model and then use the left out data sample to test and for each of the iteration I ran , I did a proper feature selection(filtering) and sequential forward selection and plotted the graph of the feature importance and the graph of the training progress as we move through the data. I also plotted the graph of the heatmap of the important features and the accuracy of the predictions on the test set and as can be seen, there is a dwindling progression across as we alternate across the training and test samples of the dataset (here for N = 100, SET 1)

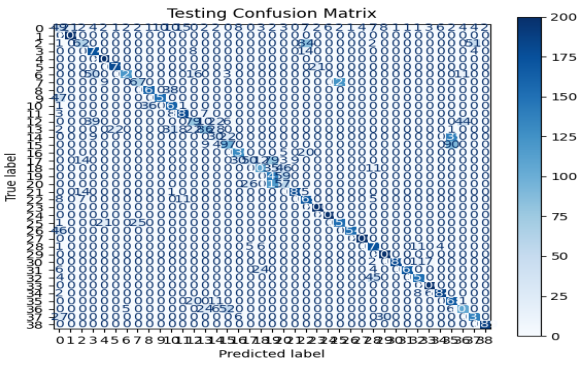
1. **Feature Selection Pipeline and Graph:**

- In the implementation I used a combination of feature selection methods, primarily mRMR (Minimum Redundancy Maximum Relevance) and Sequential Forward Selection (SFS)
- From the plot, we can see(more plots are added in the project files for the diff SETs and iterations):
    - Gray bars represent all features (68 in total)
    - Blue bars highlight the selected features (appears to be selecting features with higher importance scores)
    - Features 0-10 show particular importance for Subject 1 in Dataset 100









Looking at the Feature Importance Plot:

- The x-axis shows feature indices (0-60)
- The y-axis shows importance scores (0 -2.0)
- Blue bars indicate the selected features that were deemed most important
- For Subject 1, features in the 0-10 range were particularly informative
- The varying heights of gray bars show the relative importance of all features

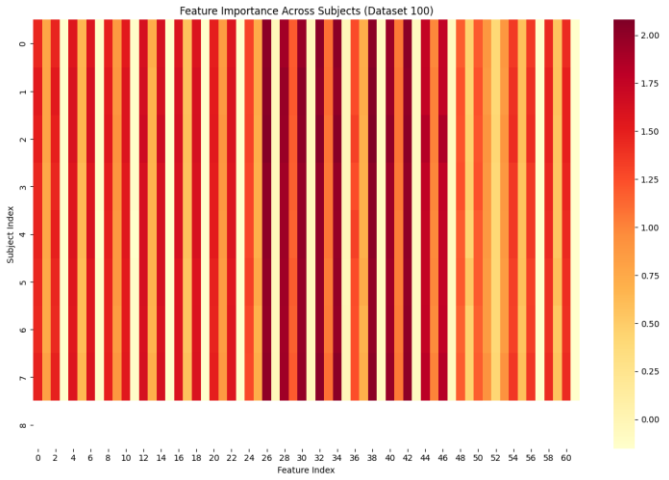
This implementation effectively:

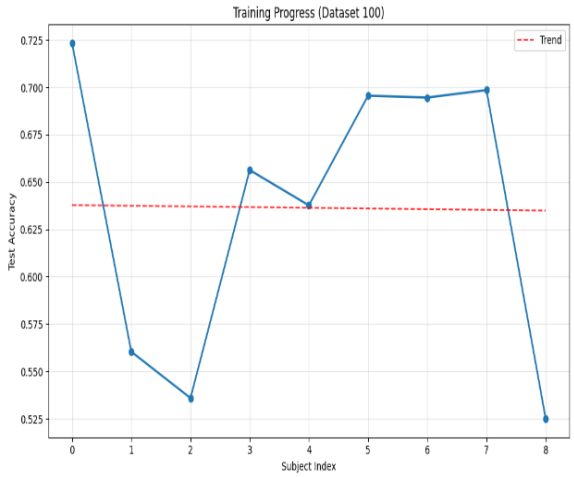
1. Isolates test subjects to prevent data leakage
2. Selects robust features that work across subjects
3. Uses KNN's non-parametric nature to capture complex patterns in the movement data
4. Validates performance on truly unseen data through LOSO

The result is a classifier that can generalize to new subjects while maintaining interpretability through feature importance visualization.

For N = 100

Mean Test Accuracy: 0.636 ± 0.072





Feature Importance and Training Progress For N = 100 SET1 (with LDA)

Insights from the Heatmap

- **Color Intensity**:
  - Darker red tones indicate higher feature importance.
  - Lighter shades represent lower importance.
- **Feature Variability**:
  - Some features (columns) have **consistent importance** across subjects.
  - Others exhibit **subject-specific variation**.
- **Key Features**:
  - Features around **25-35 and 40-50** appear to have **high importance**.

Below is a tabulated data for the train and test samples along with their respective accuracies for dataset N= 100, 200 and 300(Comparisons with the no-LDA case, Conclusion and insights would follow next for intermediate submission) :

For N = 100(Mean Test Accuracy: 0.636 ± 0.072)

Subject	Training Samples	Testing Samples	Train Accuracy	Test Accuracy
1	69,347	7,740	0.987	0.723
2	69,314	7,773	0.987	0.560
3	69,381	7,706	0.987	0.536
4	69,440	7,647	0.986	0.656
5	69,312	7,775	0.987	0.638
6	69,357	7,730	0.987	0.696
7	69,287	7,800	0.987	0.695
8	69,490	7,597	0.987	.

For N = 200 (Mean Test Accuracy: 0.516 ± 0.056)

Subject	Classes	Features	Train Samples	Test Samples	Train Accuracy	Test Accuracy
1	39	65	115797	13778	0.997	0.532
2	39	65	117088	12487	0.997	0.550
3	39	65	117783	11792	0.996	0.497
4	39	65	116815	12760	0.996	0.580
5	39	65	115718	13857	0.997	0.546
6	39	65	115812	13763	0.997	0.538
7	39	65	115853	13722	0.998	0.395
8	39	65	117979	11596	-	-

For N=300 (Mean Test Accuracy: 0.483 ± 0.058)

Subject	No. of Classes	No. of Features	Training Samples	Testing Samples	Train Accuracy	Test Accuracy
1	39	65	131,441	16,648	0.999	0.498
2	39	65	134,318	13,771	0.999	0.466
3	39	65	135,745	12,344	0.999	0.443
4	39	65	134,144	13,945	0.999	0.543
5	39	65	131,018	17,071	0.999	0.484
6	39	65	131,968	16,121	0.999	0.505
7	39	65	132,018	16,071	0.999	0.346
8	39	65	Pending	Pending	Pending	Pending
9	39	65	Pending	Pending	Pending	Pending

Training with No LDA

(Still running and training my model for this as of 12:00am on the day of submission ; 13th , cause its time and resource consuming and had to restart over a couple of times,)

To be continued…