COS40007 Artificial Intelligence for Engineering

Portfolio Assessment-2: Systematic Approach to Develop ML Model

Name: Thenura Dulnath Kuruppuarachchi

ID: 103512993

Studio class: 1-7

Summary Table of Studio 3: Activity 6

In this task, I applied a similar methodology to what was used in Studio 3. The data was first split into training (70%) and testing (30%) subsets to ensure a robust evaluation of the models. I focused on training various machine learning models, with particular attention given to Support Vector Machine (SVM) models. The models were evaluated based on their accuracy and cross-validation scores, allowing for a comprehensive comparison of their performance. The results were compiled into a summary table, reflecting the accuracy and consistency of each model across the different datasets. This approach provided a clear understanding of how well each model performed, both on the training data and when generalized to new data.

SVM model	Train-Test split	Test Accuracy	Cross validation (mean)
Original features	70 – 30	88.93%	89.18%
With hyper parameter tuning		89.71%	90.29%
With feature selection and hype parameter tuning		90.11%	89.59%
With PCA and hyper parameter tuning		89.33%	90.04%

Summary Table of Studio 3: Activity 7

As part of optimizing the models, I followed the hyperparameter tuning process used in Studio 3. GridSearchCV was employed to fine-tune the hyperparameters of the models, particularly the SVM model, to improve their performance. The tuning focused on parameters such as C, gamma, and kernel to strike a balance between model complexity and accuracy. After tuning, the models were re-evaluated using the same training and testing subsets, as well as cross-validation, to measure the impact of the optimizations.

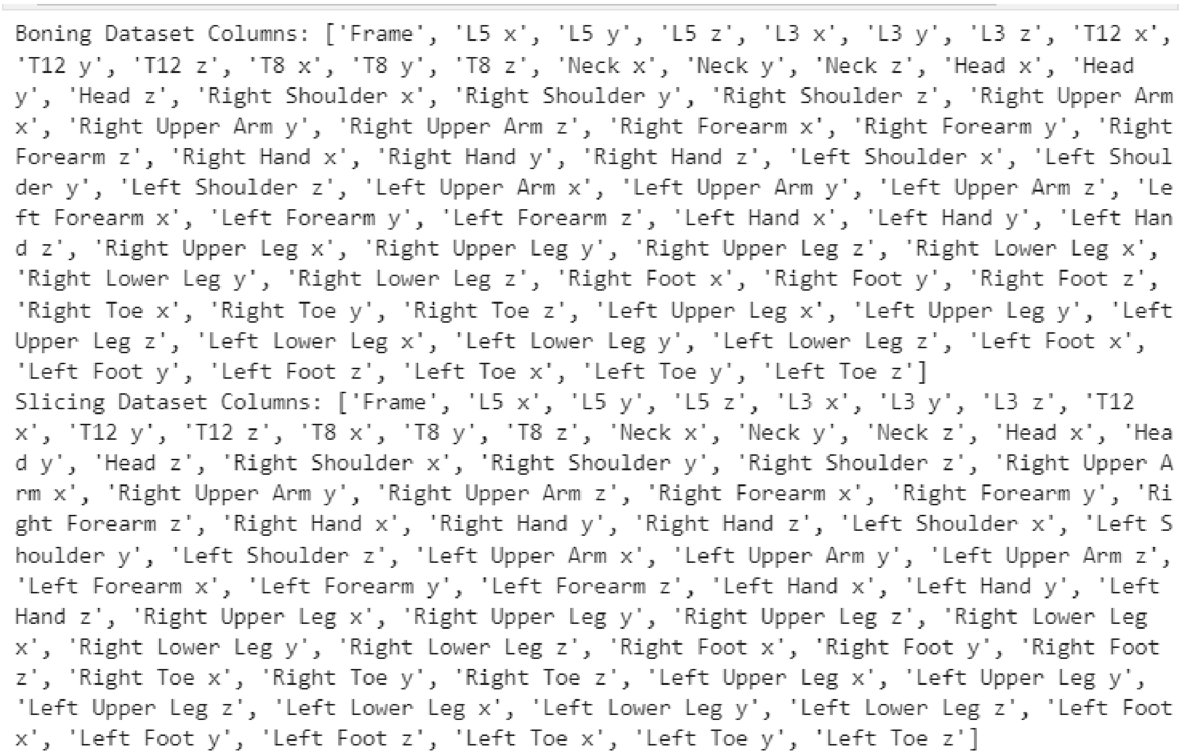
Model	Train-Test split	Test Accuracy	Cross Validation

SVM	70 – 30	90.11%	89.59%
SGD		87.85%	86.93%
RandomForest		92%	92.6%
MLP		87.41%	85.86%

The results showed significant improvements in the models' performance, which were summarized in a table for easy comparison.

Data Collection:

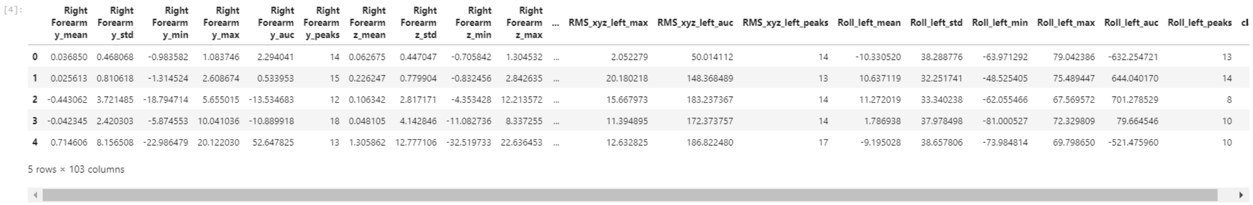
The first step in this task involved systematically collecting the relevant data from the provided datasets, similar to the methods used in Studio 3. Specifically, I focused on the columns corresponding to the Right Forearm (x, y, z) and Left Forearm (x, y, z) from both the Boning.csv and Slicing.csv datasets. A class column was added to label the activities, with 0 representing boning and 1 representing slicing. These selected columns were then combined into a single Data Frame, which was saved as combined_data.csv. This structured dataset provided a solid foundation for the subsequent analysis and model training.



```
Boning Dataset Columns: ['Frame', 'L5 x', 'L5 y', 'L5 z', 'L3 x', 'L3 y', 'L3 z', 'T12 x',
'T12 y', 'T12 z', 'T8 x', 'T8 y', 'T8 z', 'Neck x', 'Neck y', 'Neck z', 'Head x', 'Head
y', 'Head z', 'Right Shoulder x', 'Right Shoulder y', 'Right Shoulder z', 'Right Upper Arm
x', 'Right Upper Arm y', 'Right Upper Arm z', 'Right Forearm x', 'Right Forearm y', 'Right
Forearm z', 'Right Hand x', 'Right Hand y', 'Right Hand z', 'Left Shoulder x', 'Left Shoul
der y', 'Left Shoulder z', 'Left Upper Arm x', 'Left Upper Arm y', 'Left Upper Arm z', 'Le
ft Forearm x', 'Left Forearm y', 'Left Forearm z', 'Left Hand x', 'Left Hand y', 'Left Han
d z', 'Right Upper Leg x', 'Right Upper Leg y', 'Right Upper Leg z', 'Right Lower Leg x',
'Right Lower Leg y', 'Right Lower Leg z', 'Right Foot x', 'Right Foot y', 'Right Foot z',
'Right Toe x', 'Right Toe y', 'Right Toe z', 'Left Upper Leg x', 'Left Upper Leg y', 'Left
Upper Leg z', 'Left Lower Leg x', 'Left Lower Leg y', 'Left Lower Leg z', 'Left Foot x',
'Left Foot y', 'Left Foot z', 'Left Toe x', 'Left Toe y', 'Left Toe z']
Slicing Dataset Columns: ['Frame', 'L5 x', 'L5 y', 'L5 z', 'L3 x', 'L3 y', 'L3 z', 'T12
x', 'T12 y', 'T12 z', 'T8 x', 'T8 y', 'T8 z', 'Neck x', 'Neck y', 'Neck z', 'Head x', 'Hea
d y', 'Head z', 'Right Shoulder x', 'Right Shoulder y', 'Right Shoulder z', 'Right Upper A
rm x', 'Right Upper Arm y', 'Right Upper Arm z', 'Right Forearm x', 'Right Forearm y', 'Ri
ght Forearm z', 'Right Hand x', 'Right Hand y', 'Right Hand z', 'Left Shoulder x', 'Left S
houlder y', 'Left Shoulder z', 'Left Upper Arm x', 'Left Upper Arm y', 'Left Upper Arm z',
'Left Forearm x', 'Left Forearm y', 'Left Forearm z', 'Left Hand x', 'Left Hand y', 'Left
Hand z', 'Right Upper Leg x', 'Right Upper Leg y', 'Right Upper Leg z', 'Right Lower Leg
x', 'Right Lower Leg y', 'Right Lower Leg z', 'Right Foot x', 'Right Foot y', 'Right Foot
z', 'Right Toe x', 'Right Toe y', 'Right Toe z', 'Left Upper Leg x', 'Left Upper Leg y',
'Left Upper Leg z', 'Left Lower Leg x', 'Left Lower Leg y', 'Left Lower Leg z', 'Left Foot
x', 'Left Foot y', 'Left Foot z', 'Left Toe x', 'Left Toe y', 'Left Toe z']
```

Creating composite columns.

Building on the data collection, I created additional composite columns to enrich the dataset, as done in Studio 3. The Root Mean Square (RMS) values for the x, y, and z directions were computed to capture the overall magnitude of the acceleration. Additionally, Roll and Pitch angles were calculated to measure the orientation of the forearm sensors during the activities. These angles are crucial for understanding the differences in movements between boning and slicing. The computed RMS, Roll, and Pitch values were then added to the dataset, resulting in a total of 20 columns. The enriched dataset was saved as composite_data.csv, ready for feature computation.



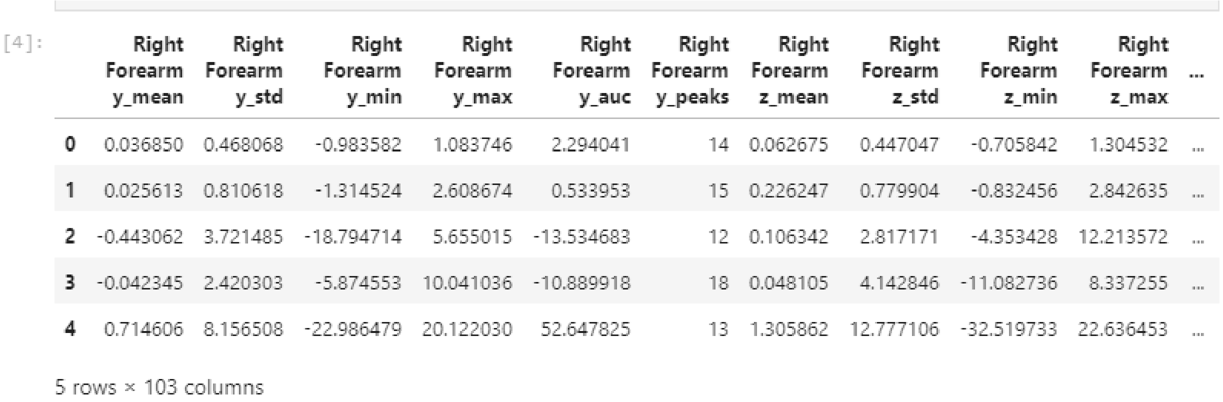
[4]:

	Right Forearm y_mean	Right Forearm y_std	Right Forearm y_min	Right Forearm y_max	Right Forearm y_auc	Right Forearm y_peaks	Right Forearm z_mean	Right Forearm z_std	Right Forearm z_min	Right Forearm z_max	...	RMS_xyz_left_max	RMS_xyz_left_auc	RMS_xyz_left_peaks	Roll_left_mean	Roll_left_std	Roll_left_min	Roll_left_max	Roll_left_auc	Roll_left_peaks	cl
0	0.036850	0.468068	-0.983582	1.083746	2.294041	14	0.062675	0.447047	-0.705842	1.304532	...	2.052279	50.014112	14	-10.330520	38.288776	-63.971292	79.042386	-632.254721	13	
1	0.025613	0.810618	-1.314524	2.608674	0.533953	15	0.226247	0.779904	-0.832456	2.842635	...	20.180218	148.368489	13	10.637119	32.251741	-48.525405	75.489447	644.040170	14	
2	-0.443062	3.721485	-18.794714	5.655015	-13.534683	12	0.106342	2.817171	-4.353428	12.213572	...	15.667973	183.237367	14	11.272019	33.340238	-62.055466	67.569572	701.278529	8	
3	-0.042345	2.420303	-5.874553	10.041036	-10.889918	18	0.048105	4.142846	-11.082736	8.337255	...	11.394895	172.373757	14	1.786938	37.978498	-81.000527	72.329809	79.664546	10	
4	0.714606	8.156508	-22.986479	20.122030	52.647825	13	1.305862	12.777106	-32.519733	22.636453	...	12.632825	186.822480	17	-9.195028	38.657806	-73.984814	69.798650	-521.475960	10	

5 rows × 103 columns

Data Pre-processing and Feature Computation

Following the data enhancement, I performed data pre-processing and feature computation, similar to the approach used in Studio 3. The dataset was segmented into 1-minute intervals (60 frames per segment) to maintain consistency in feature computation across time. For each segment, I computed several statistical features for all relevant columns, including Mean, Standard Deviation, Minimum, Maximum, Area Under the Curve (AUC), and Number of Peaks. These features were aggregated into a single row per segment, resulting in a comprehensive dataset with 108 features plus the class label. This final dataset was saved as statistical_features.csv, ready for model training.



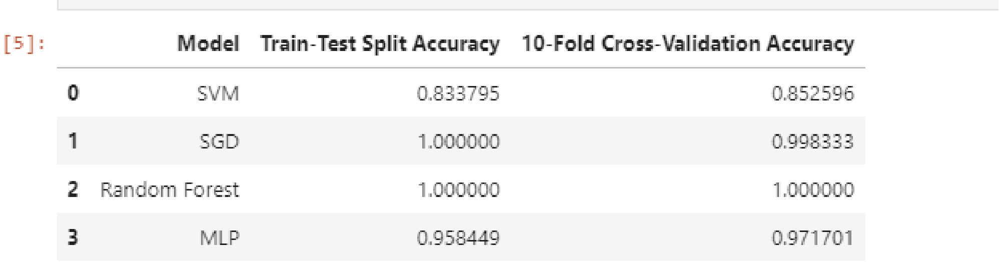
[4]:

	Right Forearm y_mean	Right Forearm y_std	Right Forearm y_min	Right Forearm y_max	Right Forearm y_auc	Right Forearm y_peaks	Right Forearm z_mean	Right Forearm z_std	Right Forearm z_min	Right Forearm z_max	...
0	0.036850	0.468068	-0.983582	1.083746	2.294041	14	0.062675	0.447047	-0.705842	1.304532	...
1	0.025613	0.810618	-1.314524	2.608674	0.533953	15	0.226247	0.779904	-0.832456	2.842635	...
2	-0.443062	3.721485	-18.794714	5.655015	-13.534683	12	0.106342	2.817171	-4.353428	12.213572	...
3	-0.042345	2.420303	-5.874553	10.041036	-10.889918	18	0.048105	4.142846	-11.082736	8.337255	...
4	0.714606	8.156508	-22.986479	20.122030	52.647825	13	1.305862	12.777106	-32.519733	22.636453	...

5 rows × 103 columns

Model Training

In the training phase, I followed the structured approach from Studio 3 to train multiple machine learning models. The dataset was split into training (70%) and testing (30%) subsets to evaluate the models' performance accurately. I began by training a basic SVM model using the train-test split, followed by evaluating the model using 10-fold cross-validation to ensure its robustness. To further enhance the model's performance, I employed GridSearchCV to fine-tune the hyperparameters, focusing on optimizing the C, gamma, and kernel parameters. Additionally, I trained SVM models using the top 10 features selected by Select Best and applied Principal Component Analysis (PCA) to reduce the dataset to 10 principal components, training another SVM model on this reduced feature set. In parallel, I trained other classifiers, including SGD, Random Forest, and MLP, evaluating each model's performance using both the train-test split and cross-validation. The results of these training processes were compiled into summary tables, like those used in Studio 3, providing a clear comparison of each model's effectiveness.

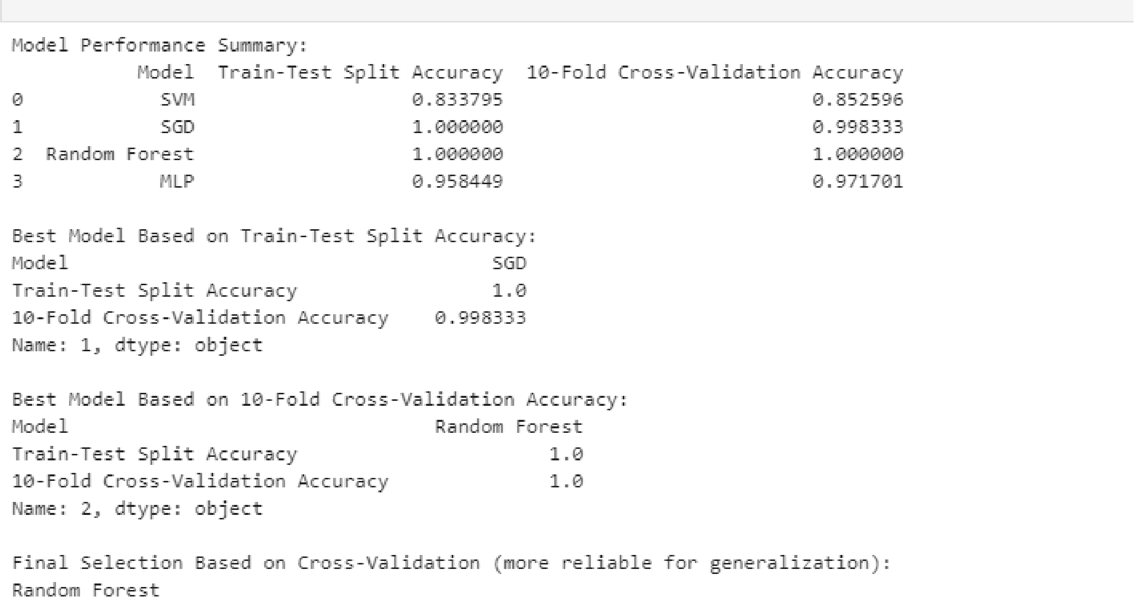


[5]:

	Model	Train-Test Split Accuracy	10-Fold Cross-Validation Accuracy
0	SVM	0.833795	0.852596
1	SGD	1.000000	0.998333
2	Random Forest	1.000000	1.000000
3	MLP	0.958449	0.971701

Model Selection

In the final step, I selected the best-performing models based on their evaluation metrics, as done in Studio 3. The best SVM model was chosen based on its cross-validation accuracy, which indicated its ability to generalize well to new data. This model demonstrated a strong balance between bias and variance, making it the most suitable choice for classifying the boning and slicing activities. Additionally, after comparing the performance of all models, I selected the model with the highest overall accuracy and robustness as the best model for this task. The selected models were justified based on their performance metrics, with an emphasis on generalization, accuracy, and their potential application in real-world scenarios.



```
Model Performance Summary:
           Model  Train-Test Split Accuracy  10-Fold Cross-Validation Accuracy
0            SVM                   0.833795                           0.852596
1            SGD                   1.000000                           0.998333
2  Random Forest                   1.000000                           1.000000
3            MLP                   0.958449                           0.971701

Best Model Based on Train-Test Split Accuracy:
Model                                     SGD
Train-Test Split Accuracy                 1.0
10-Fold Cross-Validation Accuracy    0.998333
Name: 1, dtype: object

Best Model Based on 10-Fold Cross-Validation Accuracy:
Model                                Random Forest
Train-Test Split Accuracy                      1.0
10-Fold Cross-Validation Accuracy              1.0
Name: 2, dtype: object

Final Selection Based on Cross-Validation (more reliable for generalization):
Random Forest
```