**PUNJAB UNIVERSITY COLLEGE OF INFORMATION TECHNOLOGY**

MACHINE LEARNING

**PROJECT TITLE :** ACTIVITY MONOTORING

**SUBMITTED TO :** PROF. ADEEL NISAR

**DATED :** 06/01/2024

**SEMESTER :** 5

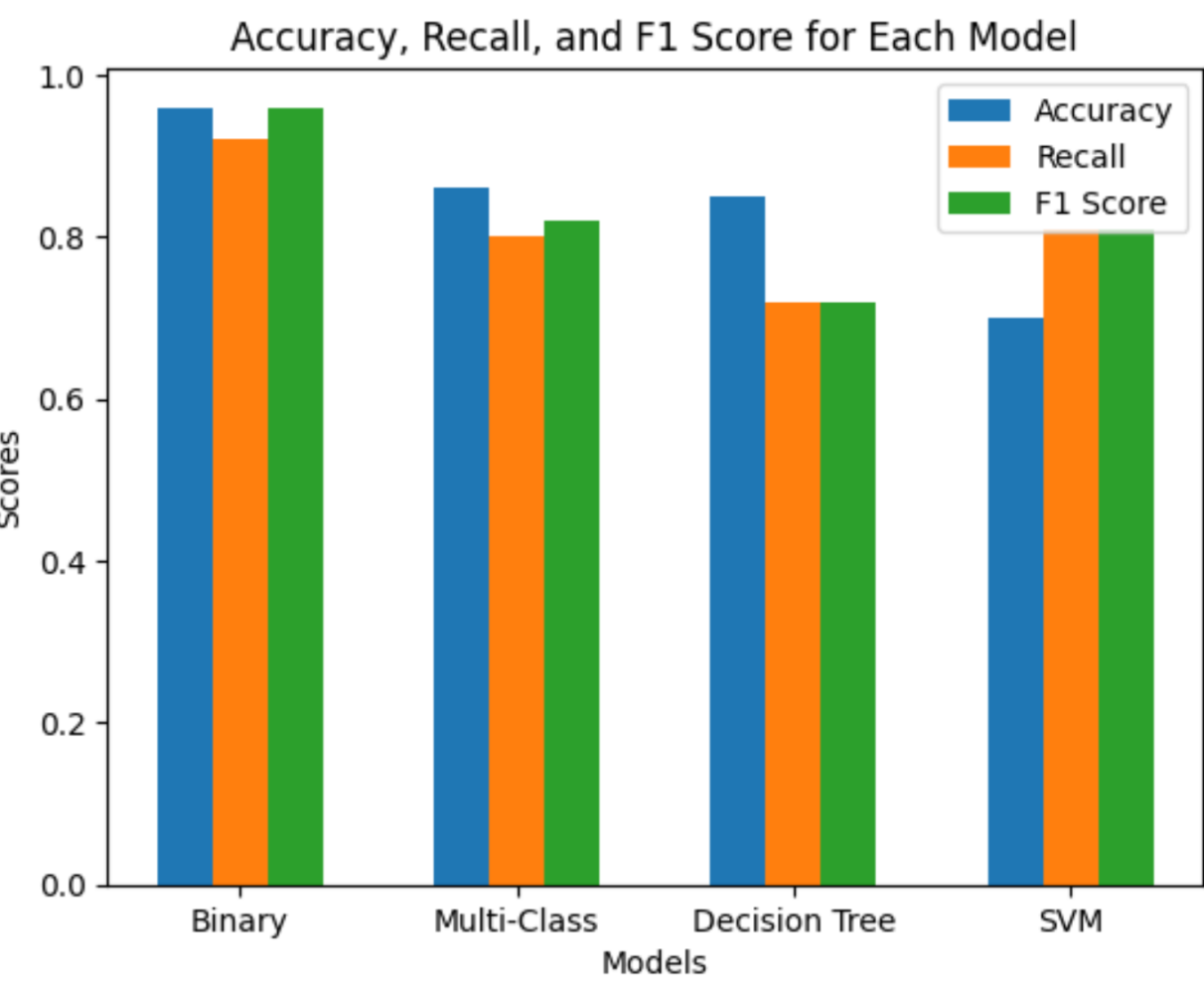
**SUBMITTED BY :**

|  |  |
| --- | --- |
| BSEF21M013 | Muhammad Amir |
| BSEF21M035 | Maryam Iqbal |
| BSEF21M037 | Saadullah |

**Project Report**

We’ve experimented with four ML models for classification. First of all we used logistic regression classifier for 2 classes. Then, we used the neural networks with 5 number of classes for multi-class classification. In the end, we used random forest and SVM algorithm for better performance.

Here is the graph providing a quick overview of different models :



**Key differences of models**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithm** | **Z-norm** | **# Classes** | **PCA** | **Sensors** |
| Binary | No | 2 | No | 2 |
| Multi-Class | Yes | 5 | No | 2 |
| RF | Yes | 55 | Yes | All |
| SVM | Yes | 55 | Yes | All |

**Activity Recognition Model V 1.0**

1. **Reading of training and test data from numpy files**

This step was easy. We just used the np library to read the files. However, the number of labels used was different for each model.

1. **Storing this data into appropriate data structures**

**Binary & Multi-Class Classifier:** Stored the data directly in numpy arrays.

**Random Forest and SVM:** Since we had to read the data of all the files , we stored multiple numpy arrays as values of dict with keys as file names.

1. **Applying pre-processing techniques[Optional]**

|  |  |
| --- | --- |
| **Algorithm** | **Z-norm** |
| Binary | No |
| Multi-Class | Yes |
| RF | Yes |
| SVM | Yes |

1. **Extracting/learning meaningful features**

In this activity recognition data set each sensor has 3D data.

* Axis-0 ---> no of examples
* Axis-1 ---> features
* Axis-3 ---> human activities are in three dimensional space. Axis-3 has size of 3 columns each belong to one dimension in space.

**Problem faced:**

Since the frequency of devices was different , so there was heterogeneous number of rows in data files.

**Solution:**

We can extract exactly 18 mathematical features and feed it to the models for better predictions and consistency.These features are following :

|  |  |  |
| --- | --- | --- |
| Maximum | Percentile 50 | First-order mean |
| Minimum | Percentile 80 | Norm of the first-order mean |
| Average | Interquartile | Second-order mean |
| Standard-deviation | Skewness | Norm of the second-order mean |
| Zero-crossing | Kurtosis | Spectral energy |
| Percentile 20 | Auto-correlation | Spectral entropy |

1. **Applying binary classification method**

* Logistic Regression(2-activities)

**Applying Multi-class classification classification method(5-activities)**

In Neural Network Model with 5 number of classes:

**Changes :**

* Increased the number of classes from **2 to 5**
* Used data from 2 sensors channels
* Implemented the Neural Network from tensorflow.keras

**Experimentation:**

* No of layers increase or decrease
* No of units in each layer
* Regularization
* Activation functions
* Batch size in model.fit
* # No of epochs

**Activity Recognition Model V 2.0**

**Changes :**

* Increased the number of classes from **5 to 55**
* Used data from all sensor channels
* Used the random forest algorithm with **100 estimators** and then **Support Vector Machine** algorithm
* Used the **PCA** algorithm for automatic feature selection

**Problems Faced and their Solution:**

* Since we were using the data from all sensors, the number of input features increased from 108 to 486

**Solution:** Used the PCA algorithm to reduce the dimensionality of input data

* The data was having some **NAN** values after feature extraction, so we weren’t able to use the PCA algorithm on this data

**Solution**: We employed data imputation techniques with ‘mean’ strategy to handle the missing values.

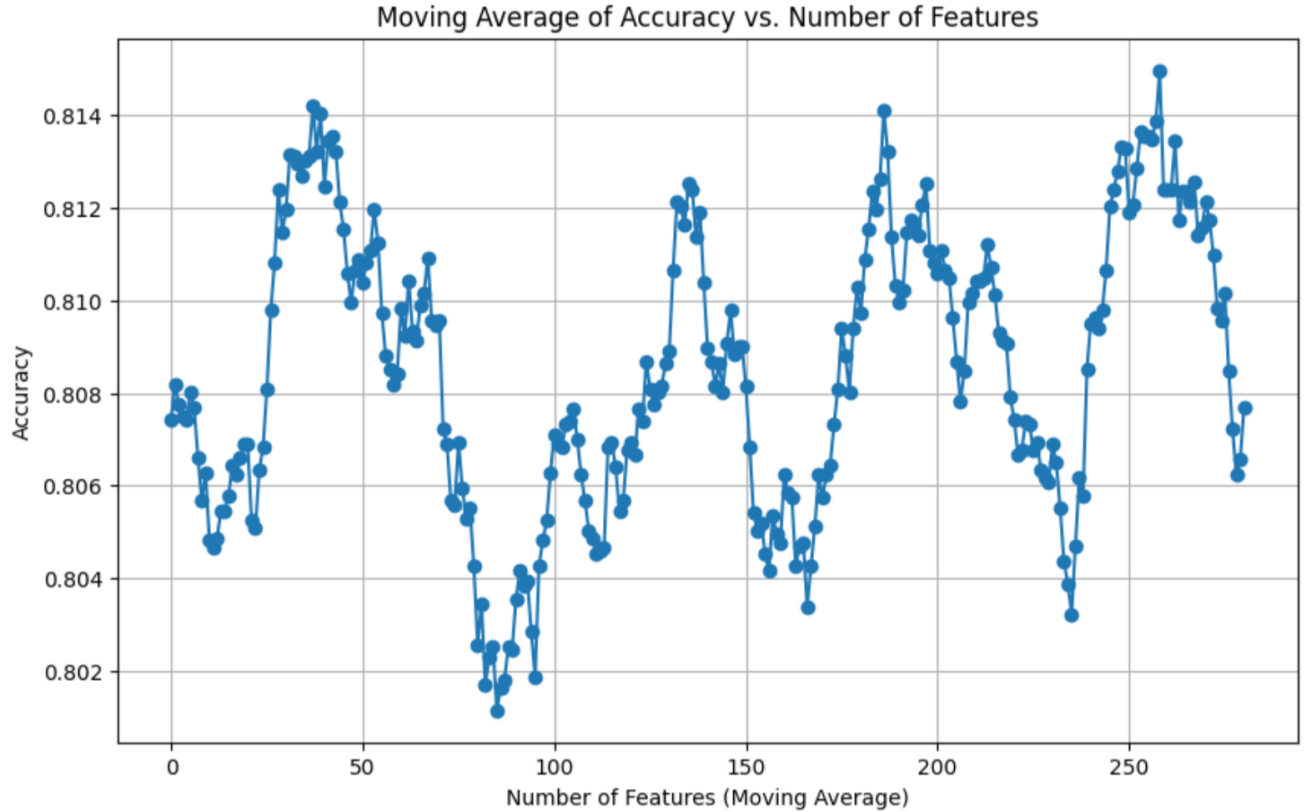
* Range of some features was dominating which led to poor performance of model.

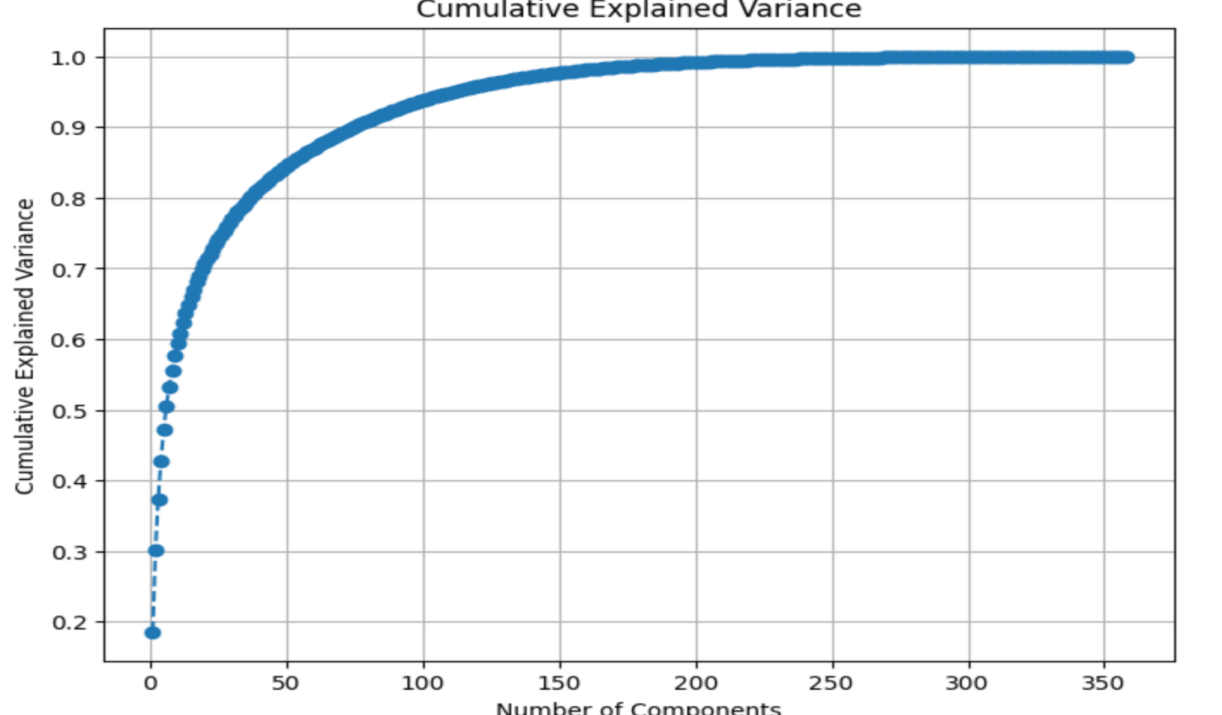
**Solution**: Used z-normalization to make all values fall within a specific range

* Perfect number of components to use in PCA was a bit challenging.

**Solution**: Performed the complete search,tried all possible values in search space and picked best one.

**Here are the results of accuracy w/ SVM model plus the variance explained by different number of components.**





* The number of training examples was low, model wasn’t able to generalize well.

**Solution**: Reserved 20% of examples for testing and used the rest for training.

**Overview of major components of code ( Structure of Code):**

1. **Imports and Setup**

The script begins with importing necessary libraries, including numpy, matplotlib, scipy, tensorflow, and sklearn. This section also sets random seeds for reproducibility.

1. **Utility Functions**

Utility functions are defined for file processing and data cleaning:

**clean\_filename**: Removes specific prefixes and extensions from filenames.

**process\_directory**: Processes all .npy files in a directory and loads them into a dictionary.

1. **Data Loading and Preprocessing**

This section includes functions for loading and preprocessing data, handling missing values, and normalizing the data.

1. **Feature Extraction and Dimensionality Reduction**

Feature extraction from sensor data and applying PCA for dimensionality reduction are performed in this section.

1. **Model Training and Evaluation**

The Random Forest classifier and SVM is trained on the processed data, and its performance is evaluated using accuracy, classification report, and confusion matrix.

**Summary**:

Support Vector Machine turned out to be the best model as it performed really well on 55 classes. The accuracy of binary classifier was highest but it only handled 2 classes.

**Scientific Reason:**   
 SVM performed better than Random Forest and Neural Network in the classification task due to its ability to handle high-dimensional data efficiently and utilize kernel functions to model complex relationships.In contrast, Random Forest and Neural Networks can suffer from overfitting , high computational demand, and extensive data requirements.