Human Activity Recognition using Smartphones

Name:	**Gourav Sen**
Registration No./Roll No.:	20120
Institute/University Name:	IISER Bhopal
Program/Stream:	EECS
Problem Release date:	January 12, 2023
Date of Submission:	April 16, 2023

1 Introduction

1.1 Objective:

Human Activity Recognition is a Supervised learning problem in which our task is to predict Human activity from the six Activities : three static activities (standing, laying, and sitting) and three dynamic activities (walking, walking upstairs, and walking downstairs). We will proceed first with data cleaning and pre-processing. Then we will perform Feature Selection to select prominent features from the feature space. After that, we will employ several classification models and select the best among them to predict class levels of the given test data.

1.2 About the dataset:

There are 563 columns and 8239 data points in the training dataset. Two of these columns contain activity labels and the subject who performed the experiment. The dataset contains 3-axial linear acceleration data from the accelerometer, axial angular velocity data from the Gyroscope and estimated values of mean, std, max, etc derived from the time series data. We are also provided with train_labels data which contains two columns one for activity level and other containing target variable.

2 Methods

2.1 Data cleaning and pre-processing:

Started by merging train_data and train_labels for data cleaning and pre-processing. We found that there were no null values and duplicate values in the dataset. Also, all values lie in between -1 and 1 so there were no outliers in the dataset. Next, we looked whether the dataset is balanced or not using count plot in figure1. By looking at the below figure1 it follows that the dataset is well balanced

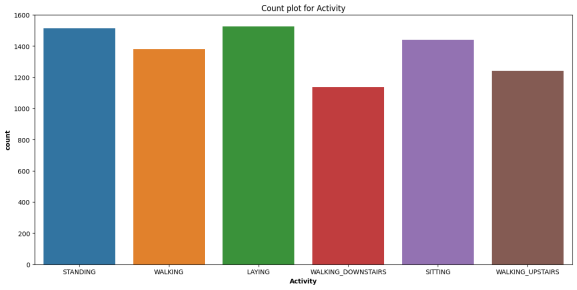


Figure 1: Count PLot for class levels

i.e,there is not much difference among the number of instances belonging to each class level.

2.2 Exploratory data analysis and Data Visualisation:

To get insights into the data and its features, we use EDA to the combined dataset. We begin by creating scatter plots and density plots for various features, and we notice that while some of them, such as tBodyAccJerkmeanX, were unable to distinguish between classes, other features, like tBodyAccJerkentropyX and tBodyAccMagmean,were able to tell the difference between static and dynamic activities, and angleXgravityMean, which could tell the difference between laying and all other classes. Box plots are another tool we use to distinguish between classes and determine the range of classes for a certain attribute. Finally, we mapped the 563 dimension feature vector to a 2D space using the tSNE and LDA plot displayed below, and we noticed standing and sitting class are most hard to classify.

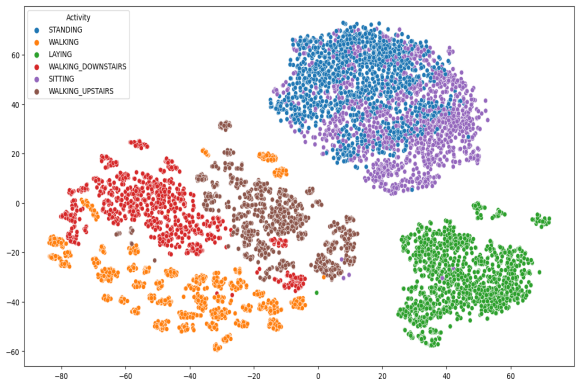


Figure 2: tSNE Plot

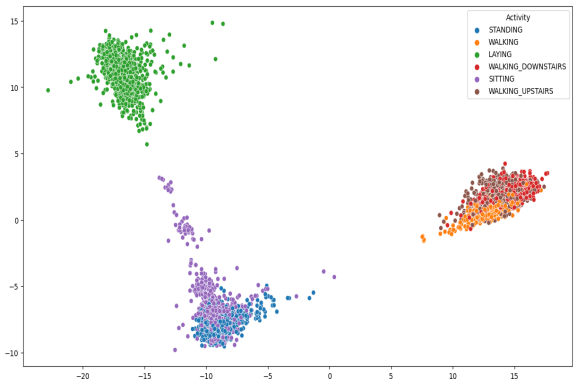


Figure 3: LDA plot

2.3 Feature Selection:

For feature selection we start by first label encoding the target variable. We are using label encoding because our feature vector are all numerical and target variable is only categorial. One hot encoding is used in case when feature vectors are categorial. the encoded labels are as follows 2- STANDING, 3- WALKING, 0- LAYING, 4- WALKING DOWNSTAIRS, 1- SITTING, 5-WALKING UPSTAIRS. I shall use filter methods for feature selection because it is computationally cheaper for high dimensional data.

- F1 test. -F1-test or Anova test is a statistical test used to assess whether there is a significant difference between the means of three or more samples of data. It is based on variation, both within and across groups.
- Mutual Information Gain- It is based on the concept of entropy from information theory. It is frequently used to choose the features that are the most informative.
- Extra tree classifier.- Extra Trees Classifier is a faster and more randomized version of Random Forest. This classifier's feature_importance attribute provides feature importance values for each feature. Then I employed the SelectFromModel() function, which essentially returns only those features which has the highest importance values.
- Correlation Coefficient- It is a statistical method used to compute the correlation between each feature and the target variable and select the features with the highest absolute correlation coefficients.

Next, we fit the selected features given by each feature selection techniques with several classifiers to check which feature selection method gave highest performance. We found that ExtraTreeClassifier gave the highest performance with accuracy : 0.978613909 and F1-score : 0.97925933.

2.4 Building the models:

The Github link for the project.

We will use the following classification models for finding best classification framework for predicting class levels of test data. 1.KNN 2. Logistic Regression 3.Decision tree 4.Random Forest 5.Linear SVM The pseudocode for the classification framework to run all models is shown below:

def classification_result(model, Xtrain, ytrain, Xtest,grid_params):

create a dictionary to store the classification results
run GridSearchCV on the passed model with the passed hyperparameters for that model
fit the model on Xtrain,ytrain
predict y_train_labels and store in results
predict y_test_labels and store in results
print and store accuracy in results
print and store classification report in results
print and store F1-score in results
print and store best estimator in results
print and store best hyperparameters for that model in results
print and store no of cross validation set in results
print and store average cross validation score in results
print and store model in results
return result

Pseudocode for plotting confusion matrix:

create the confusion matrix
cm = confusion_matrix(ytrain actual, ytrain predicted)
cmn= normalised confusion matrix by dividing each element of cm by number of samples
plot the normalised confusion matrix as heatmap
sns.heatmap(cmn, annot=True, fmt='.2f', xticklabels=class names, yticklabels=class names)

The feature selection process followed the same classification framework, but with fewer hyperparameters as we need to select best features only. We got best features using ExtraTreeClassifier as it gave best performance. We then first use KNN with n neighbours (value of k), leaf size and distance metrics as the hyperparameters. Then I passed logistic regression model with hyperparameters: Regularization parameter C and penalty(L1 or L2). The lower value of C can prevent overfitting and higher value of C can help to fit training data more closely. Then for DecisionTreeClassifier() We use the hyperparameter max depth to indicate the deepest level to which the tree can grow.Then, we employ a random forest classifier and input the hyperparameters n estimators (the number of trees in the forest), max depth (the maximum depth of the tree), and criterion (entropy or gini as the splitting criteria).Finally, we use LinearSVC with C (Regularization parameter) as the hyperparameter. Then we passed the models along with their hyperparameters to our classification_result function. We measured accuracy and macro-average F1-score for all these classification techniques and choose the classifier which has the highest macro-average F1-score and accuracy to predict the class levels of test data.

3 Experimental Results

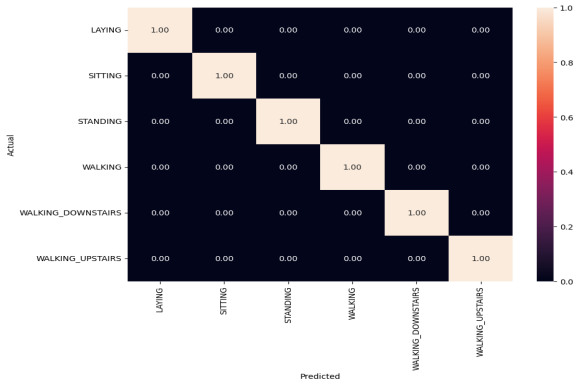
3.1 Feature Selection results:

All the feature Selection results are listed in the following table: From the below table the ExtraTreeClassifier gave maximum mean accuracy:0.978613909 and F1-score:0.97925933. Hence we choose it for feature selection and it reduced feature space from 561 to 159.

		KNN	Logistic regression	Random Forest	Decision Tree	Linear SVM	Mean Scores
	Accuracy	0.92147105	0.89282	0.9360359	0.920135939	0.892948173	**0.912683578**
Anova	F1-Score	0.924110532	0.894615055	0.936787039	0.920351892	0.894470534	**0.91406701**
	Cross Validation	0.882023877	0.890641266	0.895739017	0.869886776	0.890399063	**0.885738**
Information Gain	Accuracy	0.98106566	0.96407	0.99016871	0.974147348	0.974268722	**0.976744751**
	F1-score	0.98196190	0.965623553	0.990406623	0.973506339	0.975856338	**0.977470952**
	Cross Validation	0.981961906	0.965623553	0.990406623	0.973506339	0.975856338	**0.959145106**
ExtraTree Classifier	Accuracy	0.981065663	0.96407331	0.989076344	0.974268722	0.974268722	**0.978613909**
	F1-score	0.976257467	0.97983928 2	0.984735782	0.972021511	0.983442607	**0.97925933**
	Cross Validation	0.953635692	0.973297287	0.964315894	0.93445784	0.975724618	**0.960286266**
Corr. Coefficient	Accuracy	0.981065663	0.96407331	0.989076344	0.974268722	0.974268722	**0.976550552**
	F1-score	0.981961906	0.965623553	0.989235755	0.973619091	0.97593853	**0.977275767**
	Cross Validation	0.968078251	0.962495063	0.965408053	0.928874285	0.969291843	**0.958829499**

3.2 Classification Results:

After feature selection, we run our models and got the following results: For KNN the F1 score is- 0.99009014, the best parameters are- 'leaf size': 10, 'metric': 'manhattan', 'n neighbours': 3, average cross-validated score- 0.9734193090. For logistic regression the F1 score is- 0.97981402, the best parameters are- 'C': 10, 'penalty': 'l2',average cross-validated score is 0.97378272. For decision trees the F1 score is- 0.971096124, best parameters are- 'max depth': 9, the average cross-validated score is -0.933242848. For random forest the F1 score is- 1.0, the best parameters are- 'criterion': 'entropy', 'max depth': 13, 'n estimators': 190, average cross-validation score is-0.97560362. For linear SVM the F1 score is- 0.98240658, best parameters are-'C': 2, average cross-validated score is-0.97669578. From the results we can see that For RandomForestClassifier we got highest macro-average F1-score = 1.0, hence we choose RandomForestClassifier for predicting the class levels of test data.



4 Discussion

Human Activity Recognition (HAR) using Smartphones involves classifying human activities based on sensor data collected from smartphones. In comparison to other classifiers, the RandomTreeClassifier accurately and with the greatest F1 score predicted the actions carried out by humans. The proposed method is practical and widely available because of the utilisation of smartphones as a data source. However, The framework is constrained by the smartphone sensors, which might not be able to reliably record all elements of human behaviour. Additionally, the system depends on the user constantly having their phone on them. In the future, the suggested framework can be expanded to include other sensor data, such as GPS or audio. This project can be used to track and categorise human behaviour in various fields, like healthcare or sports, and develop custom activity monitoring and feedback systems using the framework. In conclusion, the suggested method for HAR using cell phones has proven to have good classification accuracy and has the potential for further development and use.