HUMAN ACTIVITY RECOGNITION USING SMARTPHONES

Project Report
Submitted in partial fulfilment of the
Requirements for the award of the degree
of
BACHELOR OF TECHNOLOGY (Hons.)
in
COMPUTER SCIENCE & ENGINEERING
by

SOURAV GUPTA	**(2015UGCS056)**
ADITYA NIHAL KUMAR SINGH	**(2015UGCS061)**
MD SALMAN KASHIF	**(2015UGCS066)**

Under the guidance of
Mr. SANJAY KUMAR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY, JAMSHEDPUR
831014, JHARKHAND, INDIA
MAY 2019

CANDIDATE’S DECLARATION

We hereby certify that the work which is being presented in the project work entitled “**HUMAN ACTIVITY RECOGNITION USING SMARTPHONES”** towards the partial fulfilment of the requirements for the award of the degree of Bachelor of Technology (Hons.) in Computer Science & Engineering and submitted in the Department of Computer Science & Engineering, National Institute of Technology, Jamshedpur is an authentic record of our own work carried out during the academic session 2018-19 under the supervision of Mr. Sanjay Kumar, Assistant Professor, Department of Computer Science & Engineering, National Institute of Technology, Jamshedpur.

The matter presented in this project work has not been submitted by us for the award of any other degree of this or any other institute.

SOURAV GUPTA (2015UGCS056)

ADITYA NIHAL KUMAR SINGH (2015UGCS061)

MD SALMAN KASHIF (2015UGCS066)

Date:
NIT JAMSHEDPUR

CERTIFICATE

This is to certify that the B.Tech (Hons.) project work entitled "**Human Activity Recognition Using Smartphones**" is an authentic record of the work done by Sourav Gupta (2015UGCS056), Aditya Nihal Kumar Singh (2015UGCS061) and Md Salman Kashif (2015UGCS066) in the partial fulfilment of the requirement for the award of degree Bachelor of Technology (Hons.) in Computer Science and Engineering in the Department of Computer Science and Engineering, National Institute of Technology, Jamshedpur during the session 2018-19.

The work contained in this project has not been submitted in any other University/Institute for the award of any degree/diploma.

(Sanjay Kumar)
Associate Professor
Department of Computer Science & Engineering,
NIT Jamshedpur

Viva Voce held on ________________

PANEL OF EXAMINERS

(Binod Kumar Singh)
Head of the Department
Department of Computer Science & Engineering

ACKNOWLEDGEMENT

We are extremely thankful and indebted to Mr Sanjay Kumar, for his guidance. Without his constant support and encouragement throughout our project, it would not have been possible.

Our sincere thanks to Dr B.K. Singh, Head of the Computer Science and Engineering Department, for his advice and providing us with facility for our work.

The direct and indirect assistance received from faculty members of the Computer Science & Engineering Department is also acknowledged.

Last but not least we would like to thank our friends who have enormous support during the whole tenure of our stay at NIT Jamshedpur.

ABSTRACT

Human activity recognition using either wearable devices or smartphones can benefit various applications including healthcare, fitness, smart home, etc. Instead of using wearable devices which are intrusive and require extra cost, we shall leverage on modern smartphones embedded with a variety of sensors. These sensor values can be used to recognize the activity being performed.

We are applying machine learning to a dataset of the above-mentioned values and training a classifier to recognize activities of daily living like walking, sitting, standing, laying, going upstairs, going downstairs etc. In this project, we propose a robust human activity recognition system in terms of orientation, placement, and subject variations based on various machine learning models like Logistic Regression, Support Vector Machines, Random Forest & K Nearest Neighbor. Finally, we analyze the performance of various models on the given dataset based on accuracy, precision, recall and F1 score.

CONTENTS

PARTICULARS **PAGE NO.**

CHAPTER 1

CHAPTER 2

CHAPTER 3

LIST OF FIGURES

LIST OF TABLES

1.0 INTRODUCTION

The demands for understanding human activities have grown in the health-care domain, especially in elder care support, rehabilitation assistance, diabetes, and cognitive disorders. A huge amount of resources can be saved if sensors can help caretakers record and monitor the patients all the time and report automatically when any abnormal behaviour is detected. Other applications could be in the surveillance camera. We can teach cameras to define a restricted area and mark the objects under focus. These objects could be human or any bag or so. If a strange bag appears and remains on position for a period then it would alarm the police or forces.

Many studies have successfully identified activities using wearable sensors with a very low error rate, but the majority of the previous works are done in the laboratories with very constrained settings. Readings from multiple body-attached sensors achieve low error-rate, but the complicated setting is not feasible in practice. This project uses low-cost and commercially available smartphones as sensors to identify human activities.

The growing popularity and computational power of smartphone make it an ideal candidate for non-intrusive body-attached sensors. Unlike many other works before, we relaxed the constraints of attaching sensors to fixed body position with fixed device orientation. In our design, the phone can be placed at any position around waists such as a jacket pocket and pants pocket, with arbitrary orientation. These are the most common positions where people carry mobile phones.

1.1 OBJECTIVE

We are applying machine learning to a dataset of the above-mentioned values and training a classifier to recognize activities of daily living like walking, sitting, standing, laying, going upstairs, going downstairs etc. In this project, we propose a robust human activity recognition system in terms of orientation, placement, and subject variations based on various machine learning models like Logistic Regression, Support Vector Machines, Random Forest & K Nearest Neighbor. Finally, we analyze the performance of various models on the given dataset based on accuracy, precision, recall and F1 score.

1.2 MOTIVATION

Understanding people's actions and their interaction with the environment is a key element for the development of the aforementioned intelligent systems. Human Activity Recognition is a field that specifically deals with this issue through the integration of sensing and reasoning, in order to deliver context-aware data that can be employed to provide personalized support in many applications.

As a simple example, imagine a smart home equipped with ambient sensors able to detect people's presence and the activation of household appliances. It is possible to infer the activities performed by its residents based on the sensors signals along with other relevant aspects such as time of the day and date (e.g. a person in the kitchen during morning time while a coffee machine is on suggests that person is making breakfast). Consequently, the collected HAR information can be exploited to anticipate future people requirements and become responsive to them (e.g. by automatically pre-heating the coffee machine, controlling room lighting and temperature, etc.).

In the HAR framework, there are still several issues that need to be addressed, some of which are: obtrusiveness of current wearable sensors; lack of fully pervasive systems able to reach users at any location any time; privacy concerns regarding invasive and continuous monitoring of activities (e.g. by using video cameras); the difficulty of performing HAR in real-time; battery limitations of wearable devices; and dealing with content extraction from sparse multi-sensor data.

1.3 ORGANISATION OF REPORT

The project is divided into four main sections, namely, Background, Literature Survey, Proposed Methodology, Implementation and Results. The Literature survey explains about the related work in the field of machine learning and many techniques which have been used in the area of activity recognition. The Methodology explains about the model architecture on which the project revolves around. It contains the details of the workflow and parameters used to analyze the model. The Implementation part explains about the environmental setup, the details about the datasets which are used in the training and validation of the model, the details regarding the training of the model. The fourth section is about implementing the model and running the model in order to predict the output. The Results & Validation compares the models we have run the datasets on.

2.0 LITERATURE REVIEW

2.1 RELATED WORK

Human activity recognition has been studied for years and researchers have proposed different solutions to attack the problem. Existing approaches typically use vision sensor, inertial sensor and the mixture of both [1]. Machine learning and threshold-based algorithms are often applied. Machine learning usually produces more accurate and reliable results, while threshold-based algorithms are faster and simpler. One or multiple cameras have been used to capture and identify body posture. Multiple accelerometers and gyroscopes attached to different body positions are the most common solutions [6]. Approaches that combine both vision and inertial sensors have also been proposed. Another essential part of all these algorithms is data processing. The quality of the input features has a great impact on performance. Some previous works are focused on generating the most useful features from the time series data set [5]. The common approach is to analyses the signal in both the time and frequency domain.

2.2 MACHINE LEARNING

Machine learning is the area of study concerned about the design, development and evaluation of systems capable to learn from data. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide [8]. In many common situations where we need, for instance, to complete a particular task, or perhaps to make some prediction regarding a given issue, it is possible to find solutions by the inspection and analysis of previous observations with similar characteristics to the addressed problem [25]. The primary aim is to allow the computers to learn automatically without human intervention or assistance and adjust actions accordingly. In other words, Machine Learning systems are capable of predicting future actions based on past experiences. Machine Learning algorithms have been categorized according to the type of input used for training and its expected outcome.

Supervised learning

Supervised learning where the algorithm generates a function that maps inputs to desired outputs. This algorithm can apply what has been learned in the past to new data using labeled examples to predict future events. One standard formulation of the supervised learning task is the classification problem: the learner is required to learn (to approximate the behaviour of) a function which maps a vector into one of several classes by looking at several input-output examples of the function [9].

2.3 LOGISTIC REGRESSION

Logistic Regression is the appropriate regression analysis to conduct when the dependent variable has discrete values. Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. It uses the sigmoid function as its activation function [21]. Logistic regression is used in various fields, including machine learning, most medical fields, and social sciences.

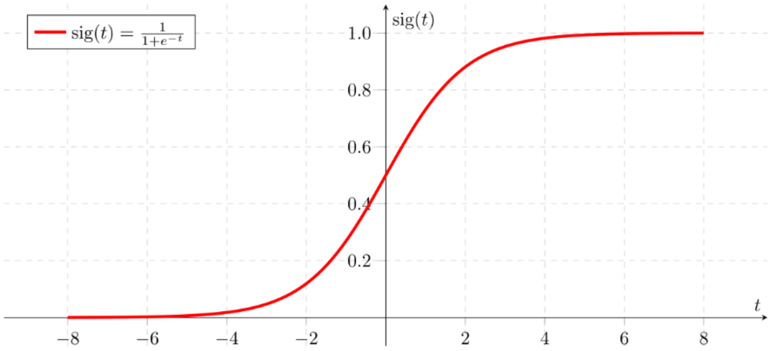


Fig 1: Sigmoid Function

Types of Logistic Regression

1. Binary Logistic Regression
The categorical response has only two 2 possible outcomes. Example: Predicting whether an email is spam or not.

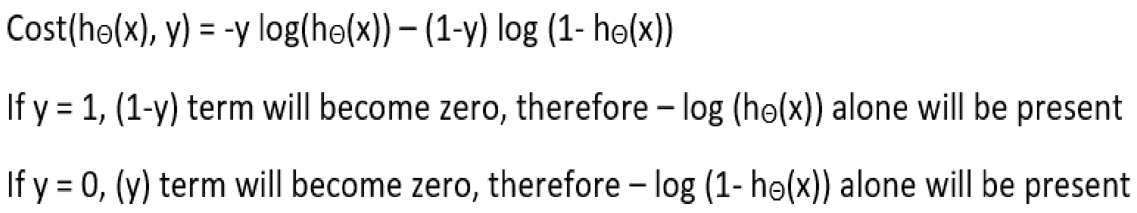
2. Multinomial Logistic Regression
Three or more categories without ordering. Example: Predicting which food is preferred more (Veg, Non-Veg, Vegan)

3. Ordinal Logistic Regression
Three or more categories with ordering. Example: Movie rating from 1 to 5

Decision Boundary

To predict which class a data belongs, a threshold can be set. Based on this threshold, the obtained estimated probability is classified into classes. Decision boundary can be linear or non-linear. Polynomial order can be increased to get a complex decision boundary [11].

Cost Function



$\mathrm{Cost}(h_\Theta(x), y) = -y \log(h_\Theta(x)) – (1-y) \log (1- h_\Theta(x))$

If y = 1, (1-y) term will become zero, therefore – log ($h_\Theta(x)$) alone will be present

If y = 0, (y) term will become zero, therefore – log (1- $h_\Theta(x)$) alone will be present

2.4 K-NEAREST NEIGHBORS

The k-nearest neighbours (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems. The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other. KNN captures the idea of similarity (sometimes called distance, proximity, or closeness) with some mathematics we might have learned in our childhood— calculating the distance between points on a graph [12].

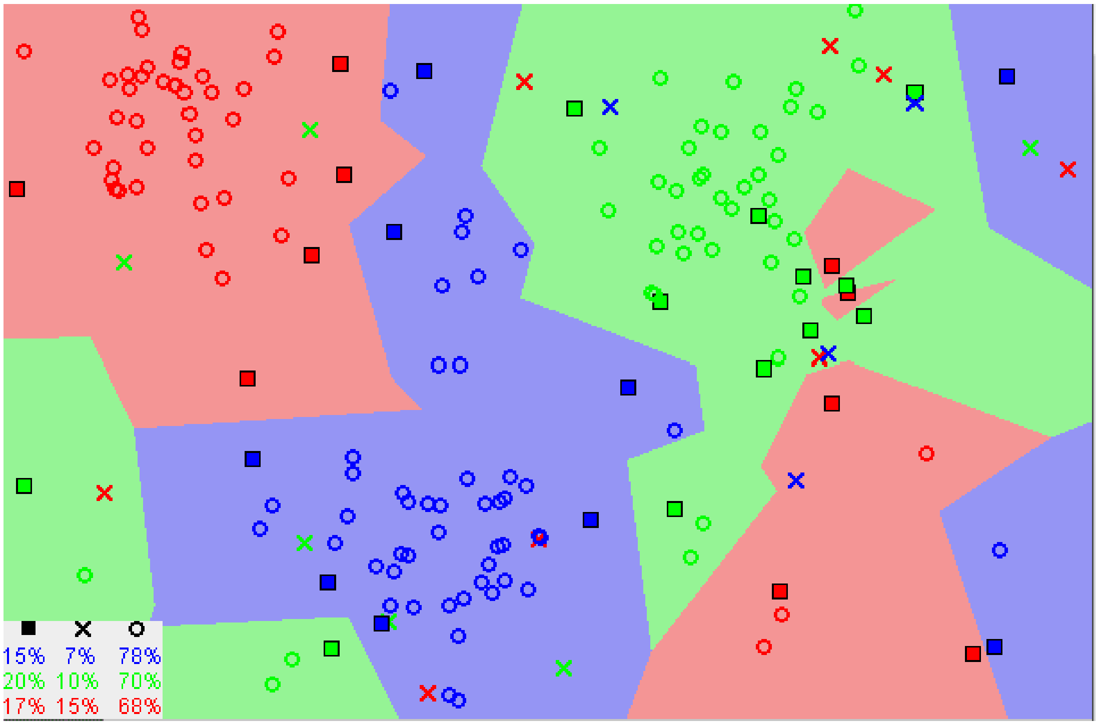


Fig 2: Showing how similar data points typically exist close to each other

The KNN Algorithm

1. Load the data
2. Initialize K to our chosen number of neighbours
3. For each example in the data
    3.1 Calculate the distance between the query example and the current example from the data.
    3.2 Add the distance and the index of the example to an ordered collection
4. Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances
5. Pick the first K entries from the sorted collection
6. Get the labels of the selected K entries
7. If regression, return the mean of the K labels
8. If classification, return the mode of the K labels

To select the K that’s right for our data, we run the KNN algorithm several times with different values of K and choose the K that reduces the number of errors we encounter while maintaining the algorithm’s ability to accurately make predictions when it’s given data it hasn’t seen before [22].

2.5 SVM (SUPPORT VECTOR MACHINE)

A Support vector machine is one of the most commonly used supervised ML algorithms. Afterwards, this algorithm has been adapted for its application in multiclass classification and regression analysis. The SVM for classification is a deterministic approach that aims to find the hyperplanes that best separate the data into classes [24]. These subspaces are the ones that provide the largest margin separation from the classes of the training data with the intention of providing a model with low generalization error for its use with unseen data samples.

SVMs are the basis for the classification of activities in this work [3]. For this reason, we now introduce them, starting from the binary SVM model which is its simplest representation, to the extended case that allows the classification of more than two classes: the multiclass SVM [23].

Extension to the Multiclass SVM

It is possible to generalize binary Machine learning models to solve problems with more than two classes. This process is known as multiclass or multinomial classification. Figure in the next page shows a simple example of a set of elements from 3 different classes in a space of two dimensions, each one represented with a different colour (red, green and blue). Also, separating hyper planes are chosen as a possible solution to this problem. There are several methods that have been previously proposed for solving multiclass problems from binary formulations. But generally, the two most commonly used are OVA (one-vs-all) and one vs - one (OVO). Their difference relies on in the way they compare each class of interest against the remaining ones: either all together for the first case and one by one for the latter [14].

In this work, we use OVA and take advantage of it because its output directly represents how likely each class to match a new test sample against the rest is. The OVA approach consists of constructing a set of m binary SVMs, each one associated with each existing class c. They are built from positive training samples coming from the class of interest (labelled as +1) and negative samples which contain the remaining samples (labelled as −1). Once the SVMs are learned, it is possible to compare them to determine which class is the most likely to represent a test sample [13]. The output of the FFP for every class is either positive or negative and its sign represents if the new sample is either classified as a given class or not. Ideally, for a given sample in a multiclass problem, only one of the binary classifiers should be positive.

2.6 RANDOM FOREST

Random forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set [16].

One big advantage of random forest is, that it can be used for both classification and regression problems, which form the majority of current machine learning systems. Random Forest adds additional randomness to the model while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model [26].

Therefore, in Random Forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node [15]. We can even make trees more random, by additionally using random thresholds for each feature rather than searching for the best possible thresholds (like a normal decision tree does).

Feature Importance

Another great quality of the random forest algorithm is that it is very easy to measure the relative importance of each feature on the prediction. Sklearn provides a great tool for this, that measures the importance of a feature by looking at how much the tree nodes, which use that feature, reduce impurity across all trees in the forest. It computes this score automatically for each feature after training and scales the results so that the sum of all importance is equal to 1.

Through looking at the feature importance, we can decide which features we may want to drop because they don’t contribute enough or nothing to the prediction process [17]. This is important because a general rule in machine learning is that the more features we have, the more likely our model will suffer from overfitting and vice versa.

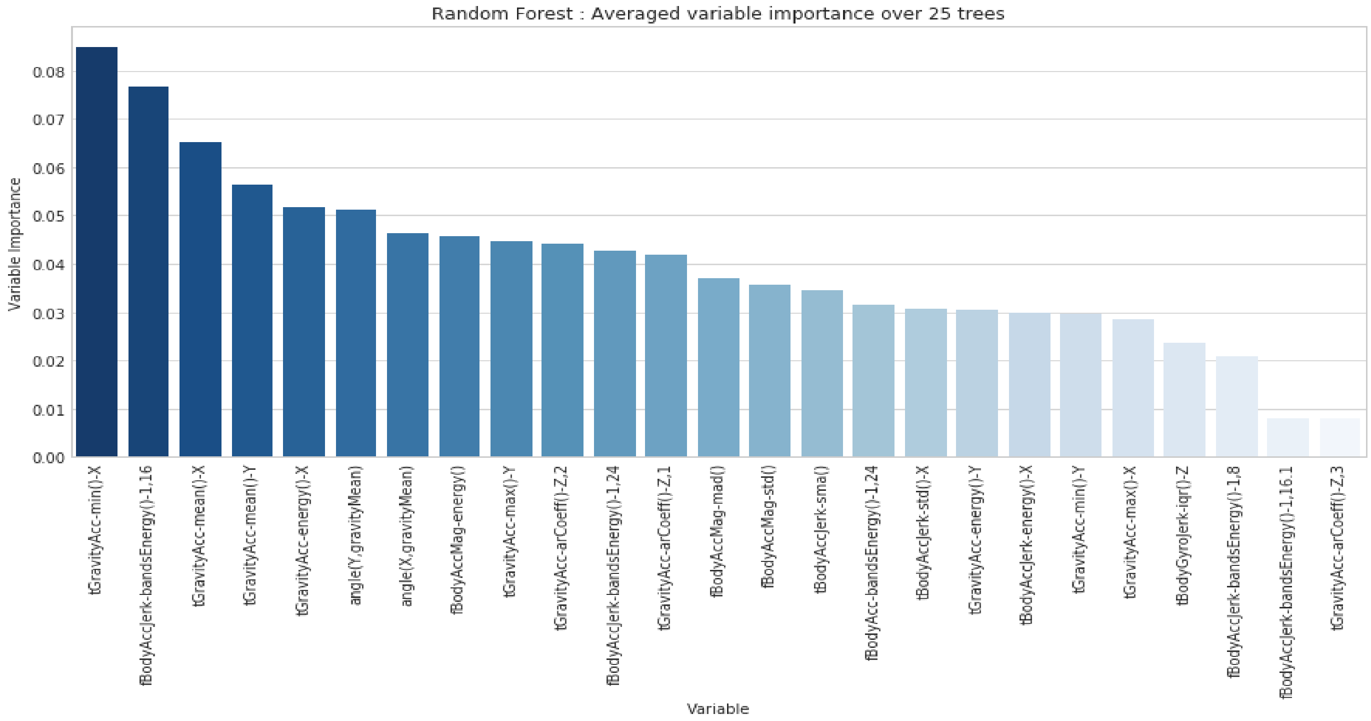
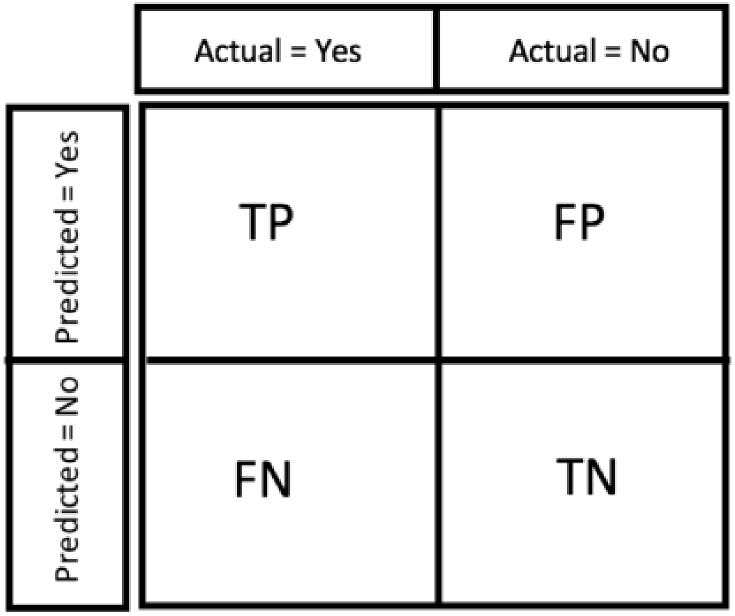


Fig 3: Feature Importance based on Random Forest

2.7 PERFORMANCE EVALUATION

The evaluation of Machine learning algorithms is predominantly made through the statistical analysis of the models using the available experimental data. The most common method is the confusion matrix which allows representing the algorithm performance by clearly identifying the types of errors (false positives and negatives) and correctly predicted samples over the test data. From it, various metrics can also be extracted such as model accuracy, sensitivity, specificity, precision and F1-Score [18]. In addition, other comparative qualitative indicators, such as the number of available activities, prediction speed and memory consumption can support the selection of human activity recognition algorithms [19]. A common method to visualize the performance of a Machine learning algorithm is through the confusion matrix C, also called a contingency table [20].

• True Positives (TP): actual samples of the class A correctly predicted as class A
• True Negatives (TN): actual samples of class B correctly predicted as class B
• False Positives (FP): actual samples of class B incorrectly predicted as class A
• False Negatives (FN): actual samples of class A incorrectly predicted as class B



	Actual = Yes	Actual = No
Predicted = Yes	TP	FP
Predicted = No	FN	TN

Fig 4: Confusion Matrix

3.0 PROPOSED METHODOLOGY

The raw data is taken from UCI Machine Learning repository and divided into training and test data the details of which is explained further. Now with the base model, feature extraction is done and the data is trained with various machine learning classifiers. The accuracy of the classifier is evaluated. The evaluation metrics and feature extraction are then done. If the accuracy is enough, the next classifier is chosen otherwise iterate the process again until all the classifiers are done right. Finally, the results are compared.

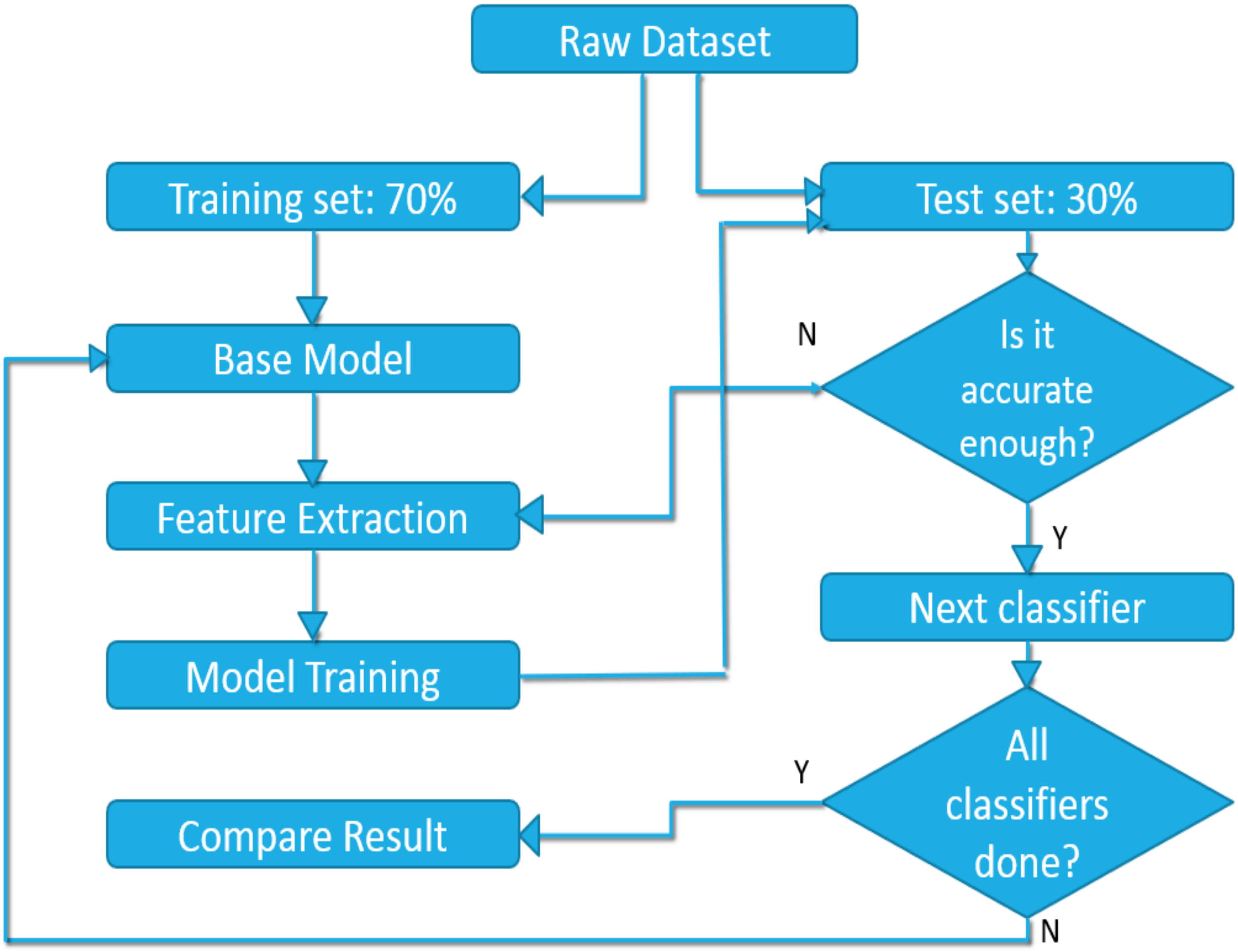


Fig 5: Proposed workflow

3.1 ABOUT THE DATASET

The Human Activity Recognition database was built from the recordings of 30 study participants performing activities of daily living (ADL) while carrying a waist-mounted smartphone with embedded inertial sensors. The objective is to classify activities into one of the six activities performed.

Description of the experiment

The experiments have been carried out with a group of 30 volunteers within an age bracket of 19-48 years. Each person performed six activities (WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, LAYING) wearing a smartphone (Samsung Galaxy S II) on the waist. Using its embedded accelerometer and gyroscope, we captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz. The experiments have been video-recorded to label the data manually. The obtained dataset has been randomly partitioned into two sets, where 70% of the volunteers were selected for generating the training data and 30% the test data.

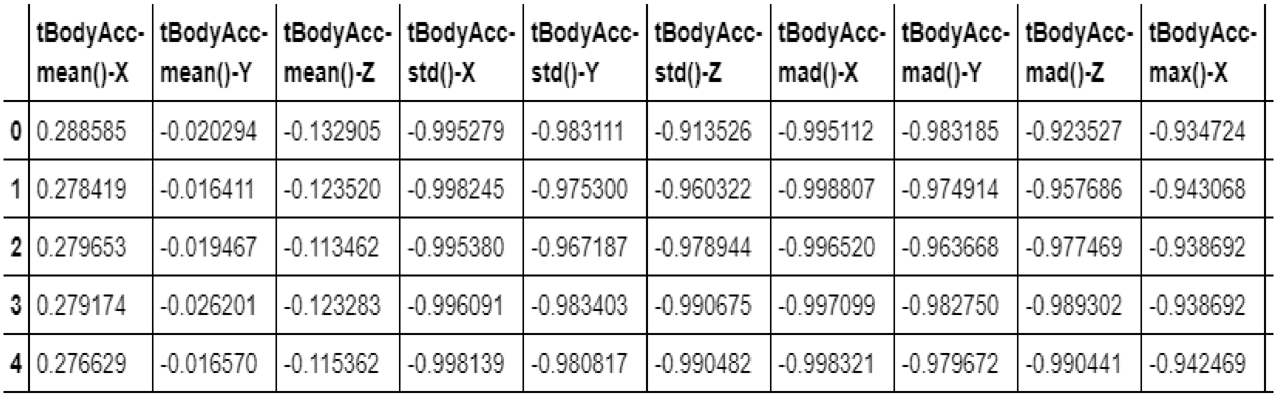
The sensor signals (accelerometer and gyroscope) were pre-processed by applying noise filters and then sampled in fixed-width sliding windows of 2.56 sec and 50% overlap (128 readings/window). The sensor acceleration signal, which has gravitational and body motion components, was separated using a Butterworth low-pass filter into body acceleration and gravity. The gravitational force is assumed to have only low-frequency components, therefore a filter with 0.3 Hz cutoff frequency was used. From each window, a vector of features was obtained by calculating variables from the time and frequency domain.

Attribute information

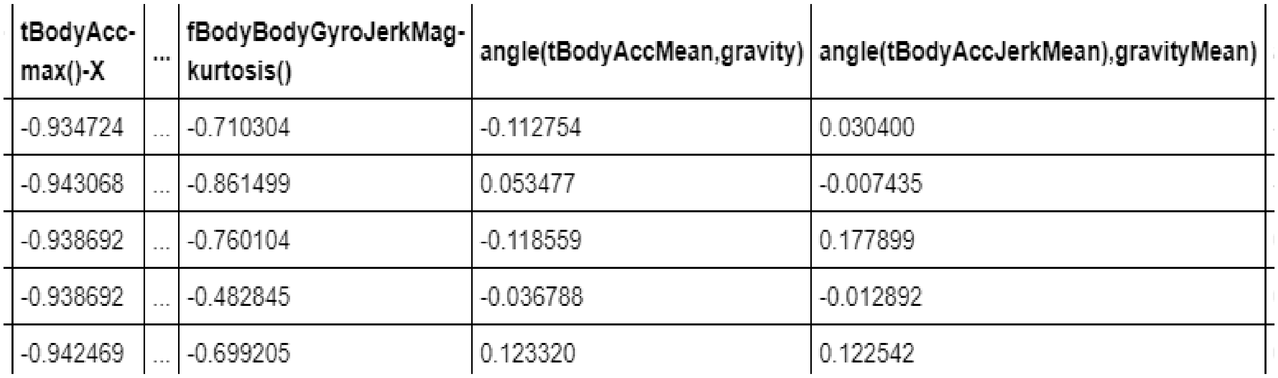
For each record in the dataset, the following is provided:

- Triaxial acceleration from the accelerometer (total acceleration) and the estimated body acceleration.
- Triaxial Angular velocity from the gyroscope.
- A 561-feature vector with time and frequency domain variables.
- Its activity labels.
- An identifier of the subject who carried out the experiment

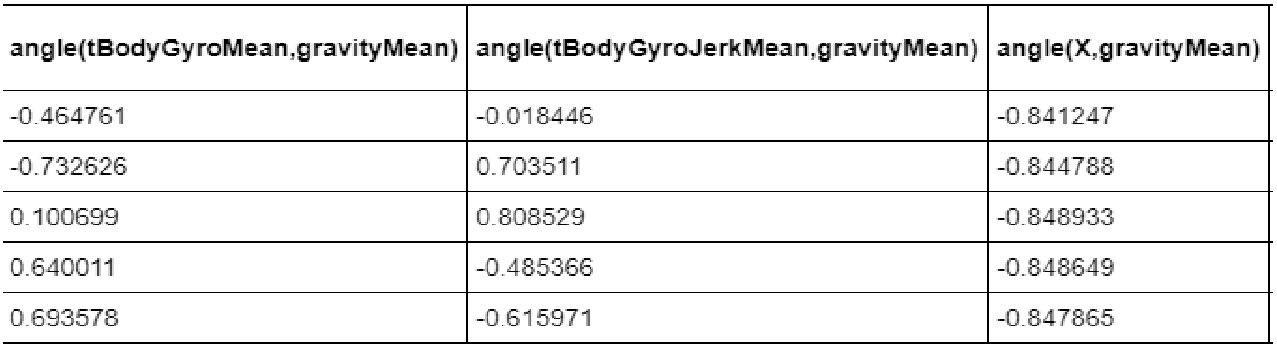
Fig 6: Screenshots of Dataset



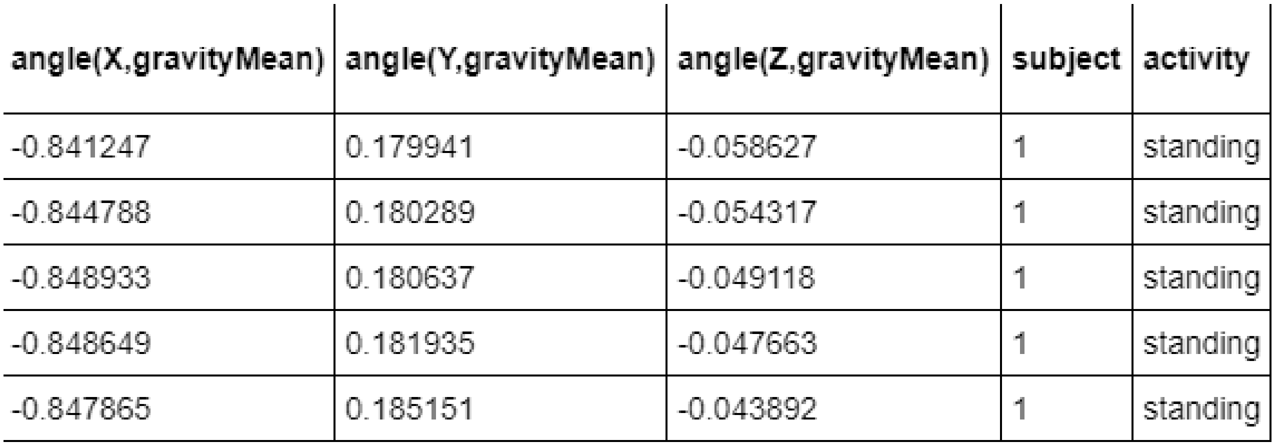
	tBodyAcc-mean()-X	tBodyAcc-mean()-Y	tBodyAcc-mean()-Z	tBodyAcc-std()-X	tBodyAcc-std()-Y	tBodyAcc-std()-Z	tBodyAcc-mad()-X	tBodyAcc-mad()-Y	tBodyAcc-mad()-Z	tBodyAcc-max()-X
0	0.288585	-0.020294	-0.132905	-0.995279	-0.983111	-0.913526	-0.995112	-0.983185	-0.923527	-0.934724
1	0.278419	-0.016411	-0.123520	-0.998245	-0.975300	-0.960322	-0.998807	-0.974914	-0.957686	-0.943068
2	0.279653	-0.019467	-0.113462	-0.995380	-0.967187	-0.978944	-0.996520	-0.963668	-0.977469	-0.938692
3	0.279174	-0.026201	-0.123283	-0.996091	-0.983403	-0.990675	-0.997099	-0.982750	-0.989302	-0.938692
4	0.276629	-0.016570	-0.115362	-0.998139	-0.980817	-0.990482	-0.998321	-0.979672	-0.990441	-0.942469



tBodyAcc-max()-X	...	fBodyBodyGyroJerkMag-kurtosis()	angle(tBodyAccMean,gravity)	angle(tBodyAccJerkMean),gravityMean)
-0.934724	...	-0.710304	-0.112754	0.030400
-0.943068	...	-0.861499	0.053477	-0.007435
-0.938692	...	-0.760104	-0.118559	0.177899
-0.938692	...	-0.482845	-0.036788	-0.012892
-0.942469	...	-0.699205	0.123320	0.122542



angle(tBodyGyroMean,gravityMean)	angle(tBodyGyroJerkMean,gravityMean)	angle(X,gravityMean)
-0.464761	-0.018446	-0.841247
-0.732626	0.703511	-0.844788
0.100699	0.808529	-0.848933
0.640011	-0.485366	-0.848649
0.693578	-0.615971	-0.847865



angle(X,gravityMean)	angle(Y,gravityMean)	angle(Z,gravityMean)	subject	activity
-0.841247	0.179941	-0.058627	1	standing
-0.844788	0.180289	-0.054317	1	standing
-0.848933	0.180637	-0.049118	1	standing
-0.848649	0.181935	-0.047663	1	standing
-0.847865	0.185151	-0.043892	1	standing

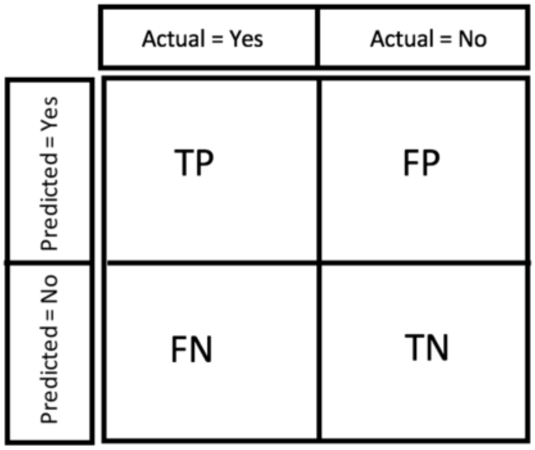
3.2 SPLITTING DATA SET

Training Dataset: The sample of data used to fit the model. This is the data for which our algorithm knows the activity labels (sitting, standing, running etc.) and which we will feed it to the training process to build our model. We have trained our model using 70% of the overall dataset.

Test Dataset: The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset. This is a portion of the data that we keep hidden from our algorithm and only use it after the training takes place to compute some metrics (confusion matrix, accuracy, precision, recall etc.) that can give us a hint on how our algorithm behaves. We have tested our trained model using 30% of the overall dataset.

4.3 EVALUATION METRICS

Confusion Matrix: A confusion matrix is a table that is used to describe the performance of a classification model on a set of test data for which the true values are known. It can be shown as below.



	Actual = Yes	Actual = No
Predicted = Yes	TP	FP
Predicted = No	FN	TN

Fig 7: Confusion Matrix

Accuracy: It is defined as percentage of total items classified correctly.
It is given as: (TP+TN)/(N+P)

Precision: It is the number of items correctly identified as positive out of total items identified as positive. It is given as: TP/(TP+FP)

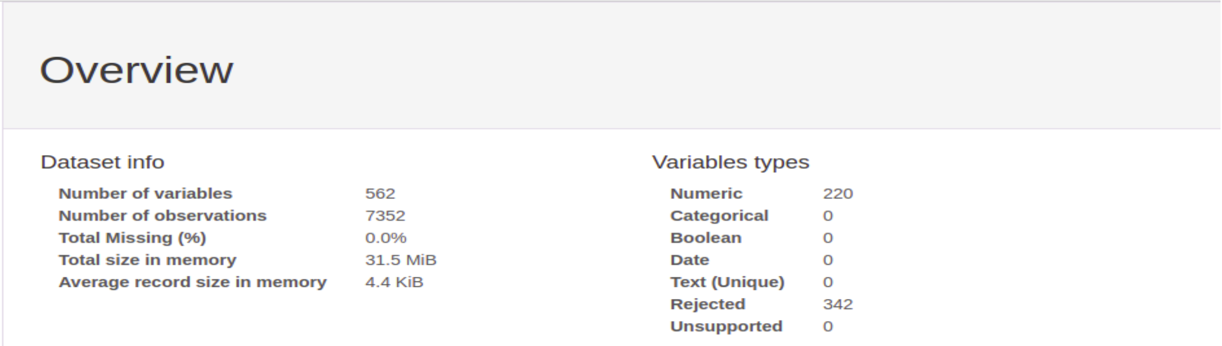
Recall or Sensitivity or TPR (True Positive Rate): Number of items correctly identified as positive out of total true positives. It is given as: TP/(TP+FN)

F1 Score: It is the harmonic mean of precision and recall and is given by.
F1 = (2*Precision*Recall) / (Precision + Recall)

3.4 FEATURE EXTRACTION

Pandas Profiling

Pandas profiling is used for simple and fast exploratory data analysis of a Pandas Dataframe. It is all about summarizing the dataset through descriptive statistics. Data types, missing values, mean, median and standard deviation are few elements needed for profiling a dataset. The goal of data profiling is to have a solid understanding of the data so that it can be queried and visualized in various ways. The following screenshots below shows the significance of pandas profiling.



Overview

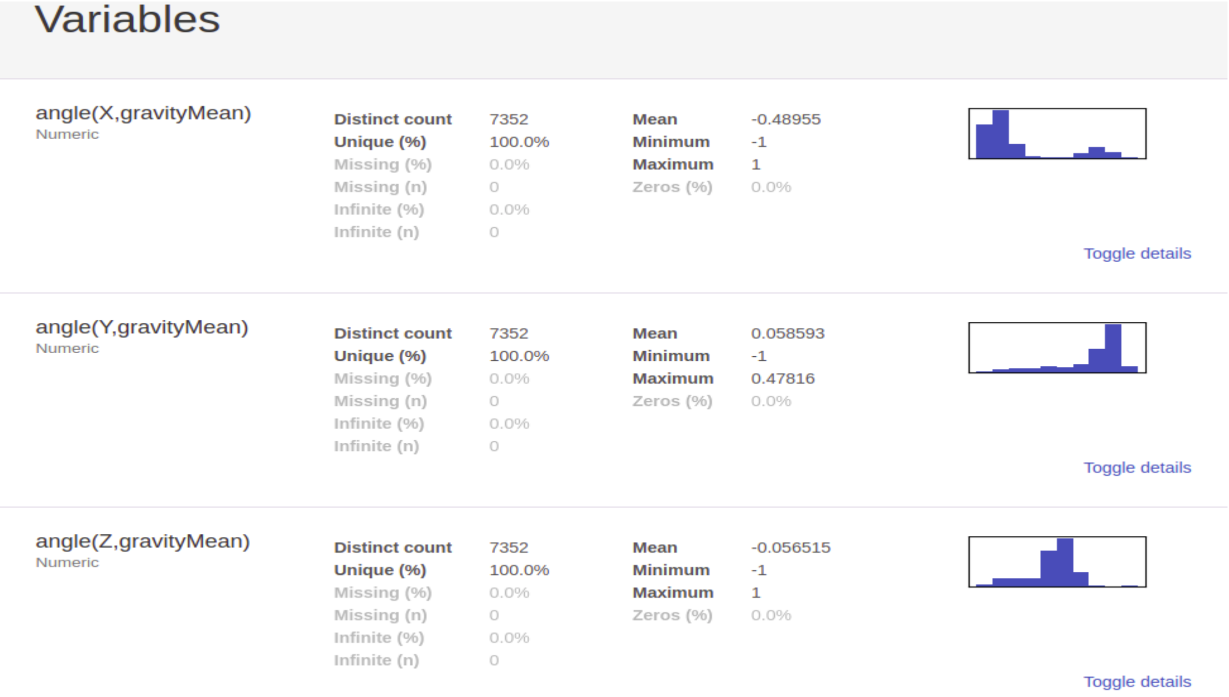
Dataset info

Number of variables	562
Number of observations	7352
Total Missing (%)	0.0%
Total size in memory	31.5 MiB
Average record size in memory	4.4 KiB

Variables types

Numeric	220
Categorical	0
Boolean	0
Date	0
Text (Unique)	0
Rejected	342
Unsupported	0

Fig 8: Overview of the dataset



Variables

angle(X,gravityMean)
Numeric

Distinct count	7352	Mean	-0.48955
Unique (%)	100.0%	Minimum	-1
Missing (%)	0.0%	Maximum	1
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

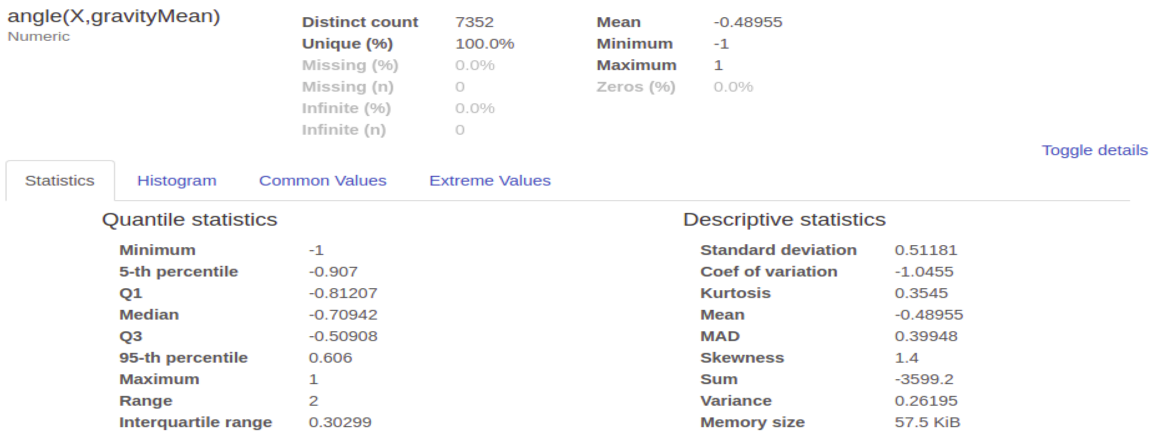
angle(Y,gravityMean)
Numeric

Distinct count	7352	Mean	0.058593
Unique (%)	100.0%	Minimum	-1
Missing (%)	0.0%	Maximum	0.47816
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

angle(Z,gravityMean)
Numeric

Distinct count	7352	Mean	-0.056515
Unique (%)	100.0%	Minimum	-1
Missing (%)	0.0%	Maximum	1
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Fig 9: Details about the features in the dataset



angle(X,gravityMean)
Numeric

Distinct count	7352	Mean	-0.48955
Unique (%)	100.0%	Minimum	-1
Missing (%)	0.0%	Maximum	1
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Toggle details

Statistics | Histogram | Common Values | Extreme Values

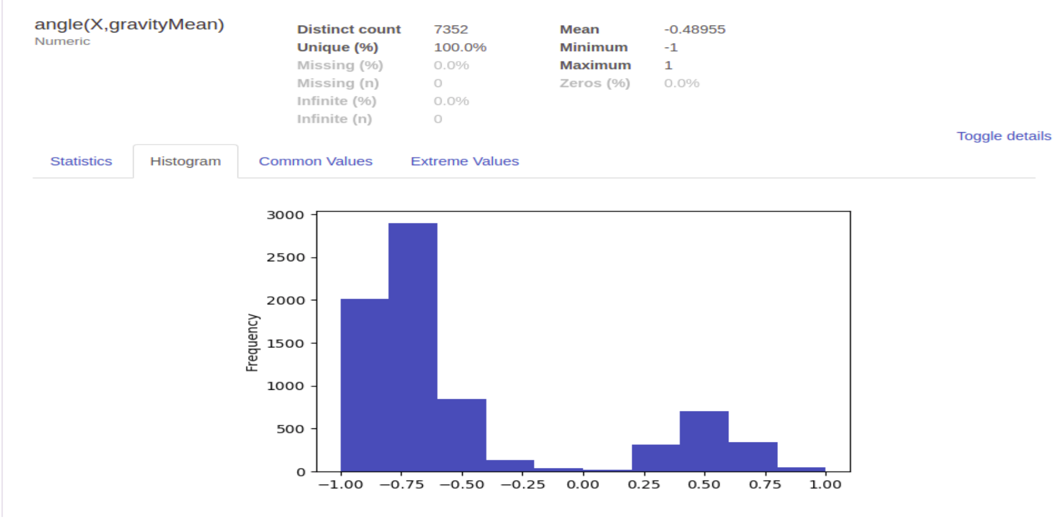
Quantile statistics

Minimum	-1
5-th percentile	-0.907
Q1	-0.81207
Median	-0.70942
Q3	-0.50908
95-th percentile	0.606
Maximum	1
Range	2
Interquartile range	0.30299

Descriptive statistics

Standard deviation	0.51181
Coef of variation	-1.0455
Kurtosis	0.3545
Mean	-0.48955
MAD	0.39948
Skewness	1.4
Sum	-3599.2
Variance	0.26195
Memory size	57.5 KiB

Fig 10: Details about each individual feature



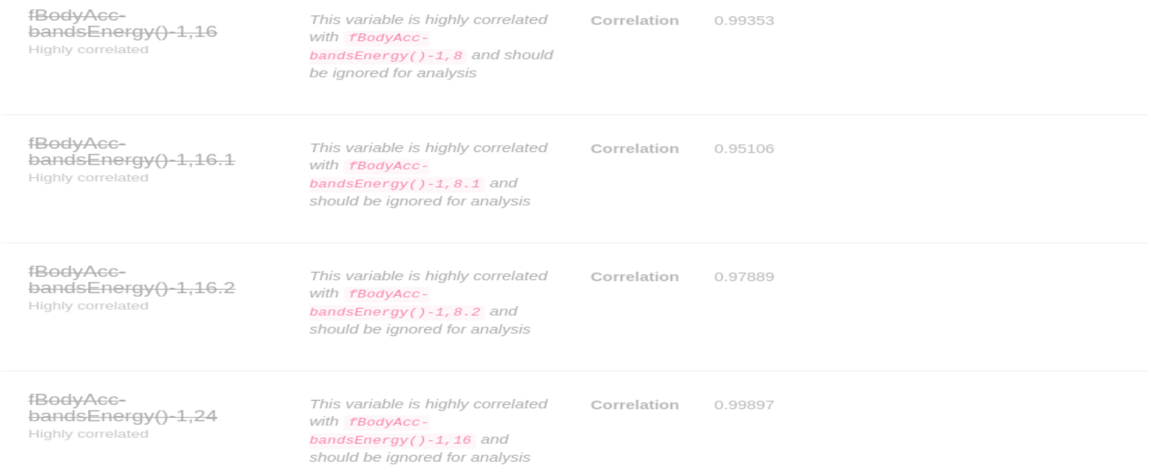
angle(X,gravityMean)
Numeric

Distinct count	7352	Mean	-0.48955
Unique (%)	100.0%	Minimum	-1
Missing (%)	0.0%	Maximum	1
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Toggle details

Statistics | Histogram | Common Values | Extreme Values

Fig 11: Distribution of the values of the features



~~fBodyAcc-bandsEnergy()-1,16~~ Highly correlated	*This variable is highly correlated with* `fBodyAcc-bandsEnergy()-1,8` *and should be ignored for analysis*	Correlation	0.99353
~~fBodyAcc-bandsEnergy()-1,16.1~~ Highly correlated	*This variable is highly correlated with* `fBodyAcc-bandsEnergy()-1,8.1` *and should be ignored for analysis*	Correlation	0.95106
~~fBodyAcc-bandsEnergy()-1,16.2~~ Highly correlated	*This variable is highly correlated with* `fBodyAcc-bandsEnergy()-1,8.2` *and should be ignored for analysis*	Correlation	0.97889
~~fBodyAcc-bandsEnergy()-1,24~~ Highly correlated	*This variable is highly correlated with* `fBodyAcc-bandsEnergy()-1,16` *and should be ignored for analysis*	Correlation	0.99897

Fig 12: Pandas profiling gives information about highly correlated features which can be ignored

4.0 IMPLEMENTATION

4.1 SYSTEM REQUIREMENTS

This section specifies hardware and software requirements for implementing ad training our model.

Hardware Requirements

- Core i5 Processor
- 8 GB RAM
- 2 GB GPU

Software Requirements

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system conda. The Anaconda distribution is used by over 13 million users and includes more than 1400 popular data-science packages suitable for Windows, Linux, and MacOS.

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

We used Jupyter Notebook for our work. Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating Jupyter notebook documents. The "notebook" term can colloquially make reference to many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format depending on context. A Jupyter Notebook document is a JSON document, following a versioned schema, and containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media, usually ending with the ".ipynb" extension.

4.2 IMPLEMENTING RANDOM FOREST

A. **I**mported packages and data.

B. Built the random forest model containing all 561 variables

C. Plotting variable importance graph with respect to each variable.

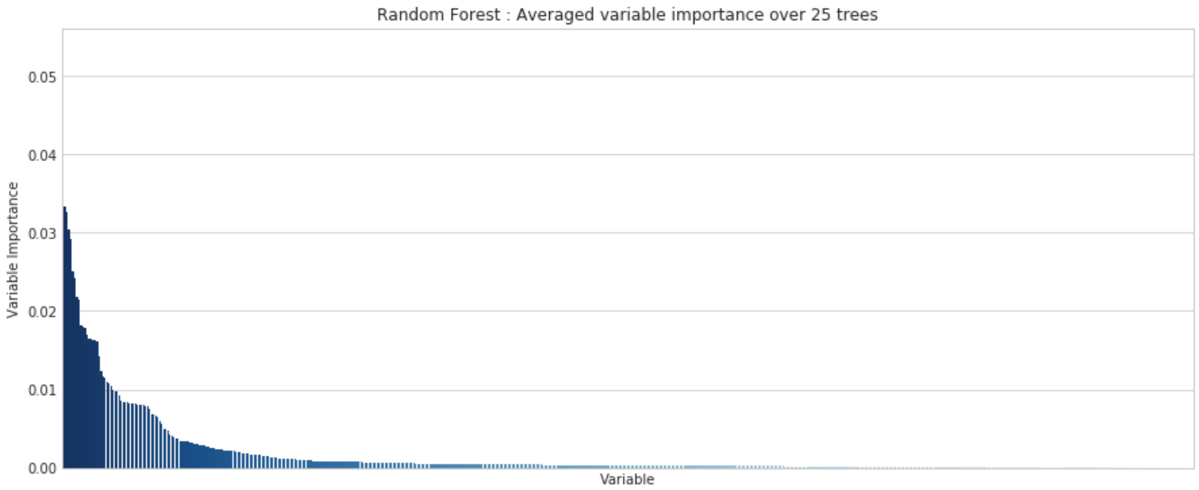


Fig 13: Feature importance of different variables in the dataset

D. Variable importance of top 25 variable is plotted which is obtained by taking average result over 25 trees.

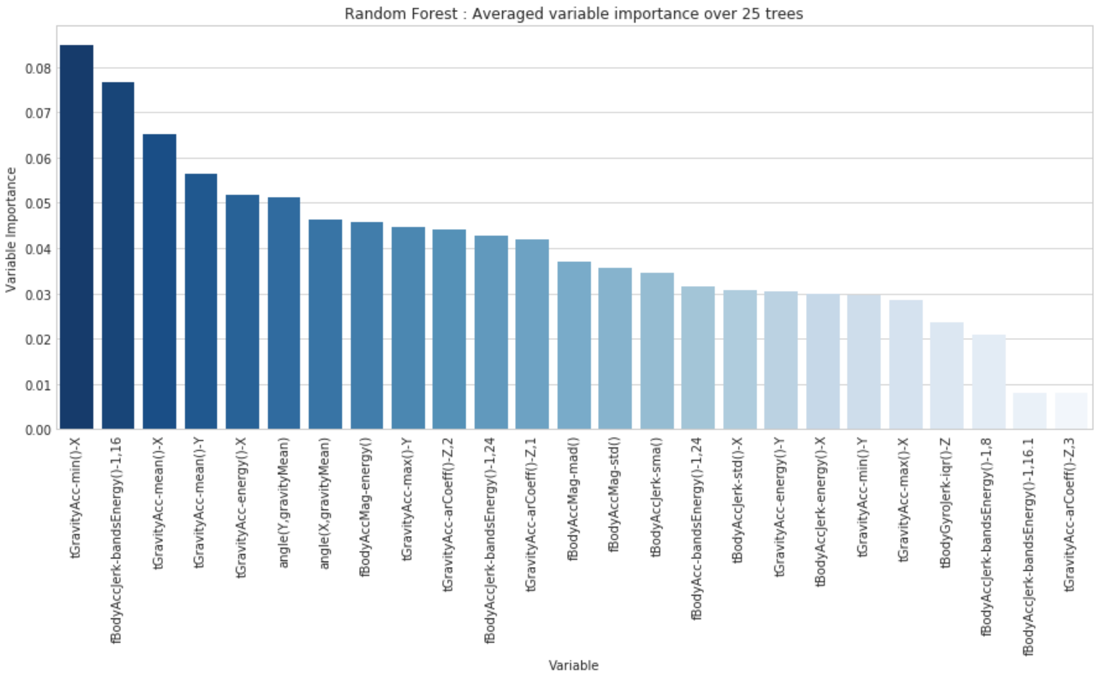


Fig 14: Random Forest: Average variable importance over 25 trees

E. We choose top 25 variables according to their feature importance. Then we finally select first four features and the seventh feature for our final evaluation using random forests. We don't add features beyond seventh one because their addition doesn’t cause significant increase in OOB accuracy.

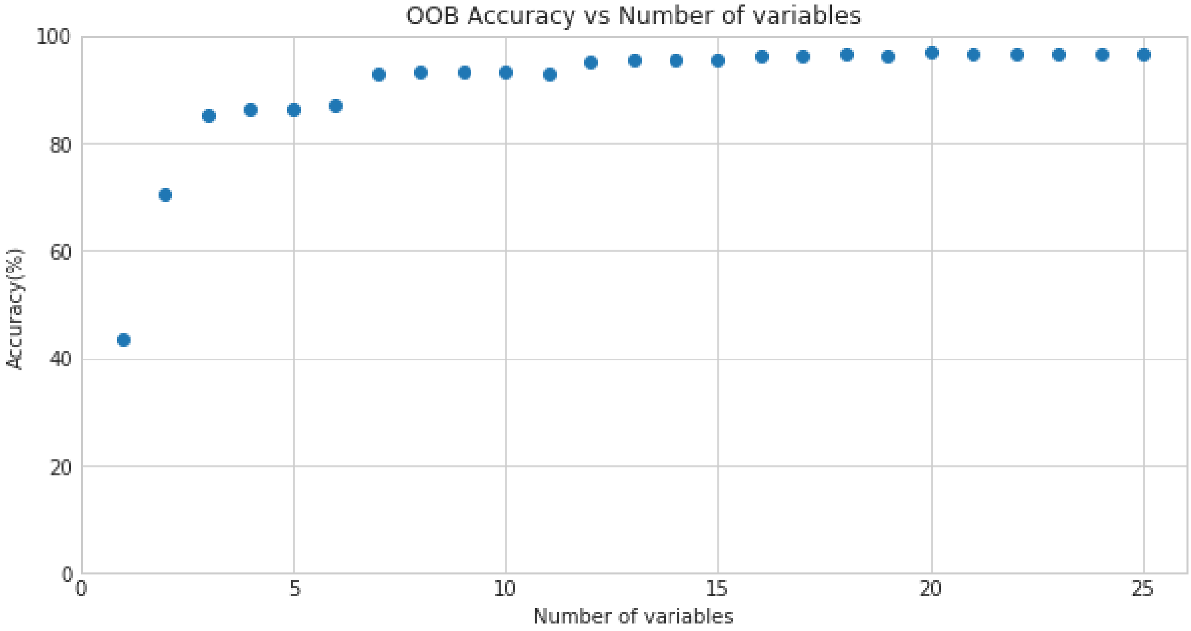


Fig 15: OOB accuracy vs number of variables

F. Deciding the number of trees to be used in training the classifier. From the graph obtained above, we have chosen 50 trees to train our model.

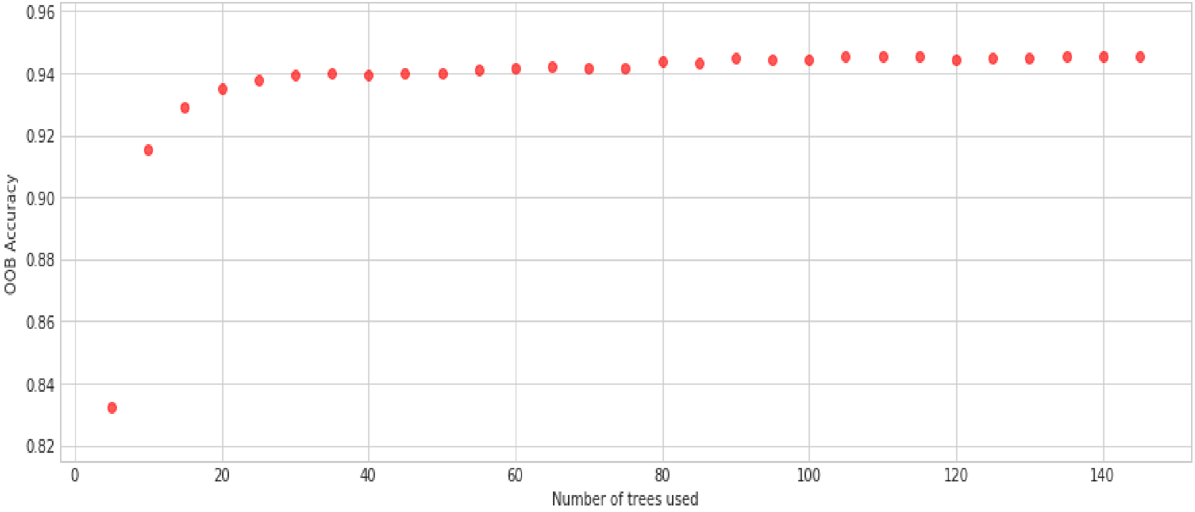


Fig 16: OOB Accuracy vs Number of trees used

G. We found out the confusion matrix for our training and test data.

H. From the above methods we have selected 5 best variables for training our classifier. Next step is to plot the value of each variable with respect to all the defined activity labels (sitting, standing, laying, walking etc.) to see their distribution.

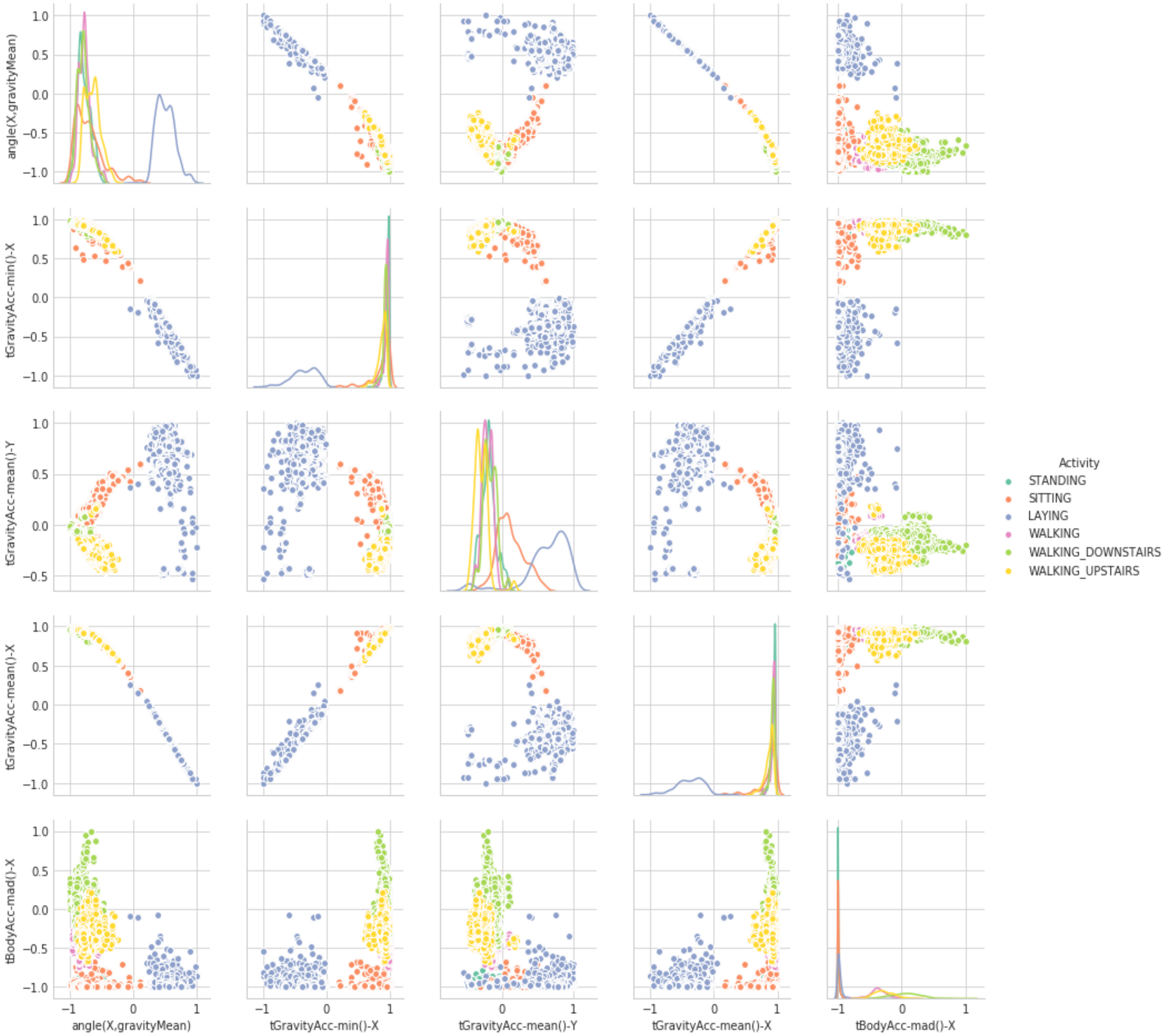


Fig 17: Pair plot of final 5 features

4.3 IMPLEMENTING K NEAREST NEIGHBORS

A. Imported the required packages

B. Imported the training and test data

C. Training KNN classifier using all features

D. Selecting the number of neighbors required for training the classifier.

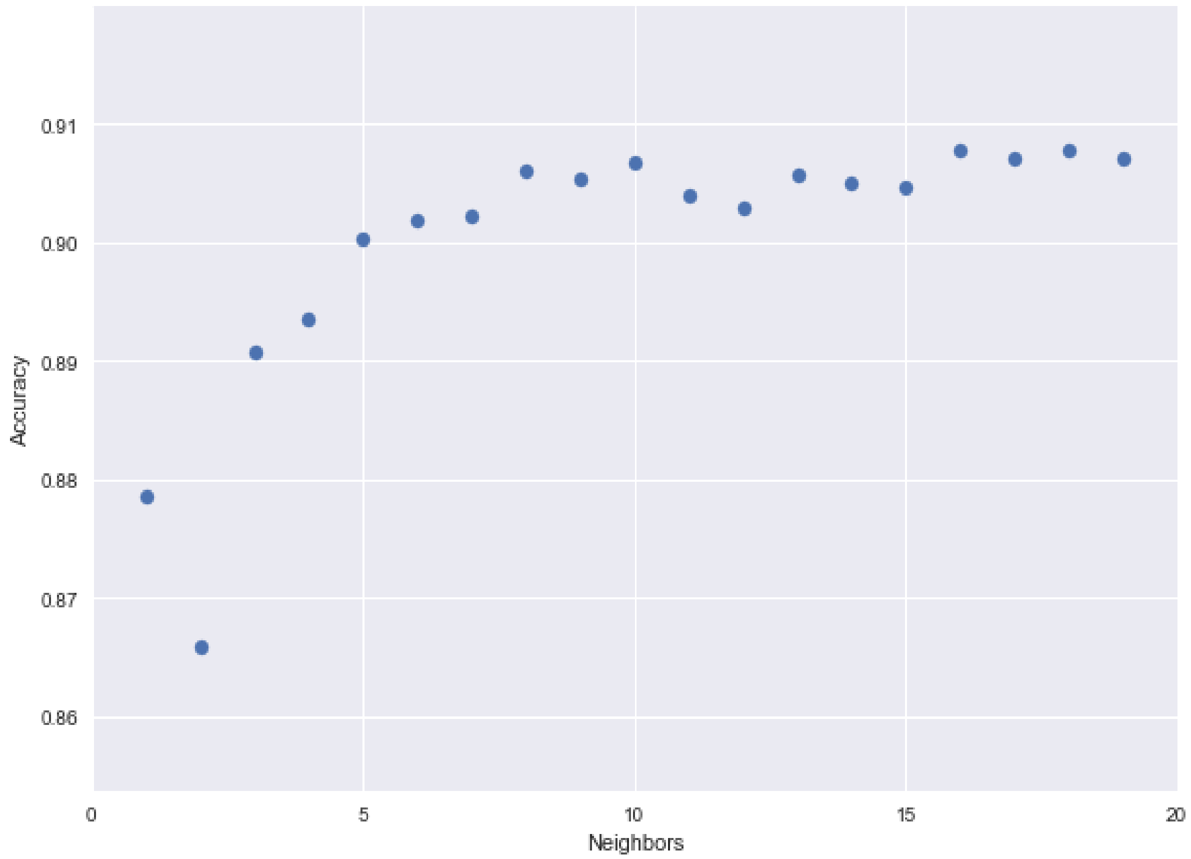


Fig 18: Accuracy vs number of neighbors in KNN

E. Using 10 neighbours to train the classifier based on the deduction from the previous step.

F. Using Principal Component Analysis for dimensionality reduction using 10 components. Principal Component Analysis (PCA) is a statistical procedure that uses an orthogonal transformation which converts a set of correlated variables to a set of uncorrelated variables.

G. Selecting number of components out of 561 to get the most prominent 170 components using the result shown in the graph in the next page.

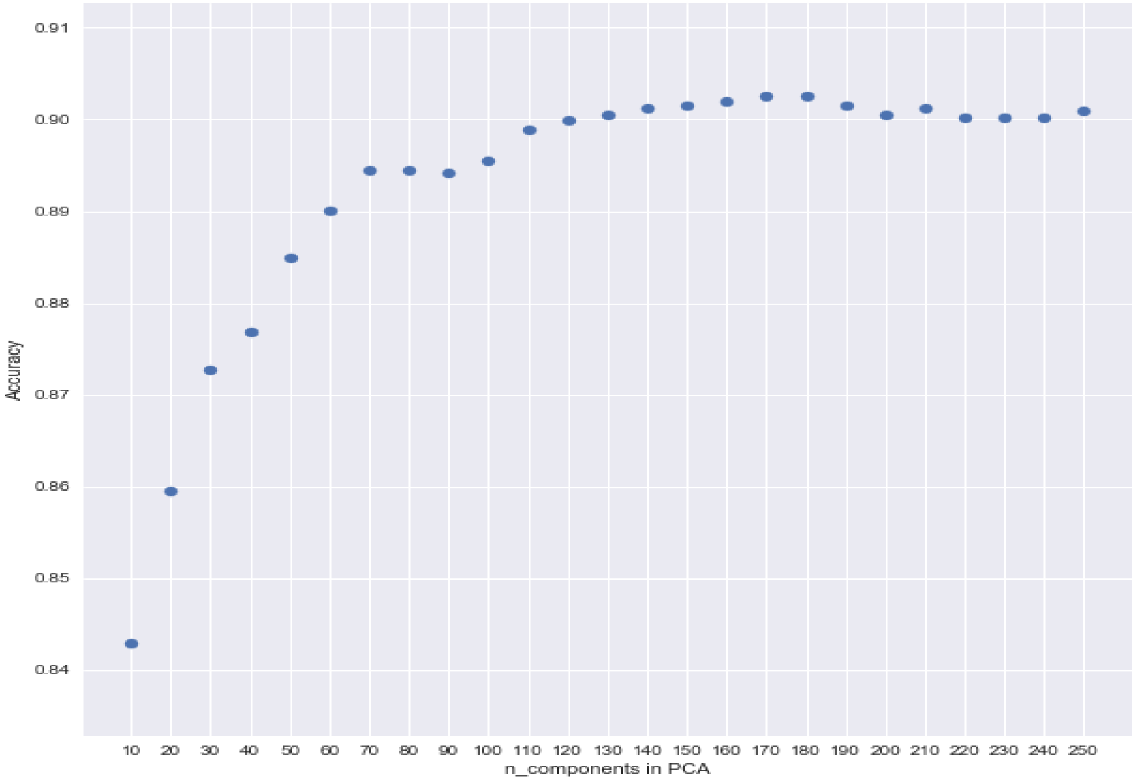


Fig 19: Number of Components against accuracy

H. Selecting number of neighbors after PCA reduction

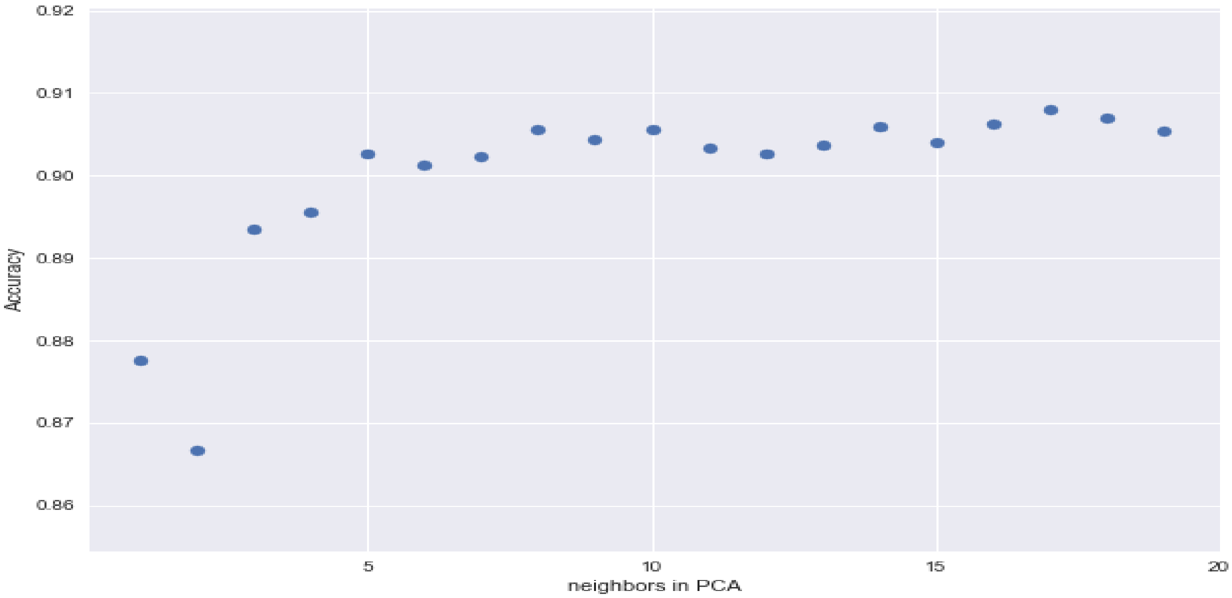


Fig 20: Number of neighbors versus accuracy

4.4 IMPLEMENTING LOGISTIC REGRESSION

A. Importing packages and data

B. Training the model using all features of the dataset

C. Choosing the value of regularisation parameter

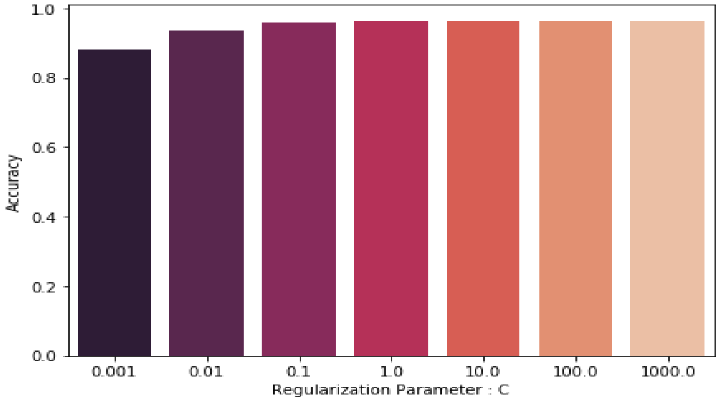


Fig 21: Regularisation Parameter vs Accuracy

D. Training model with the value of C as 10

E. Using Principal Component Analysis for dimensionality reduction using 10 components.

F. Selecting the number of components based on the graph.

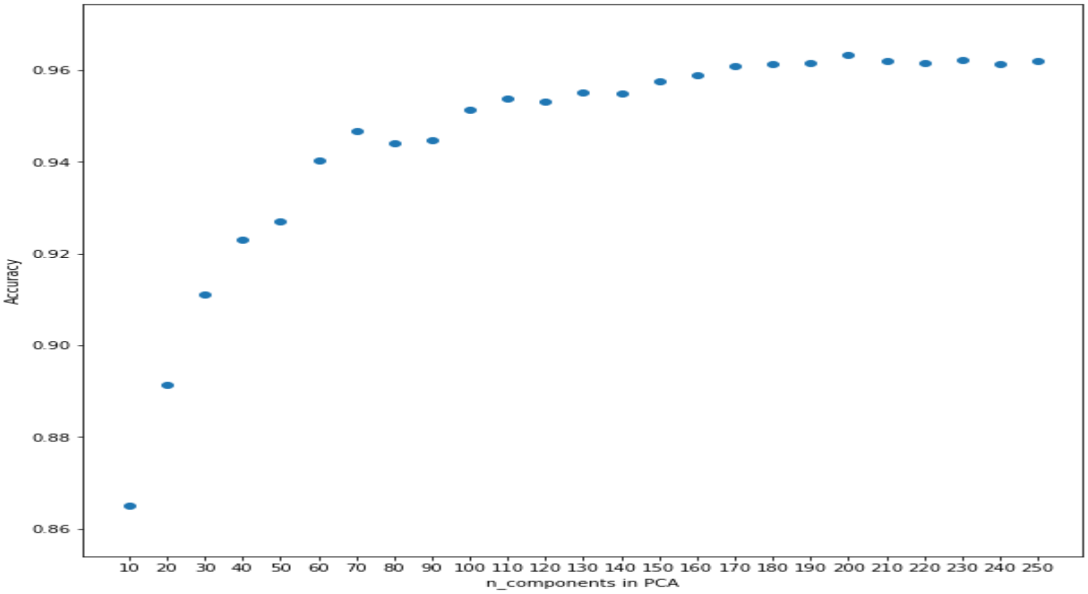


Fig 22: Number of components of PCA vs Accuracy

G. Using the above model to select the value of regularization parameter

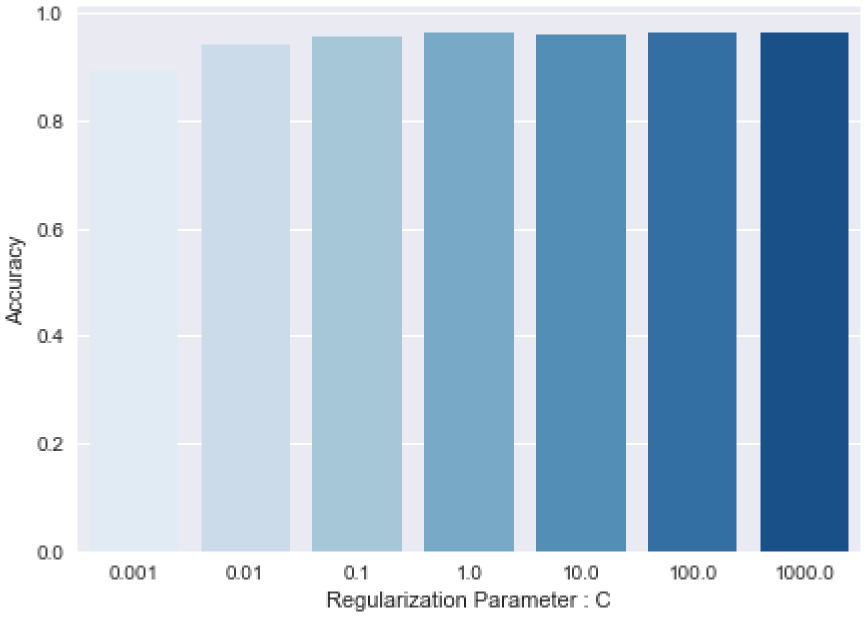


Fig 23: C vs Accuracy

H. Training the model with C=100

4.5 IMPLEMETATION OF SVM

A. Importing packages and data

B. Training model using all features

C. Choosing the kernel among rbf, linear, poly and sigmoid

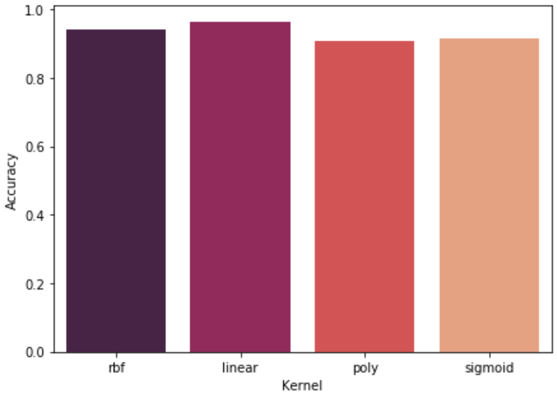


Fig 24: Accuracy for each of the kernels

D. Choosing the value of regularisation parameter using linear kernel

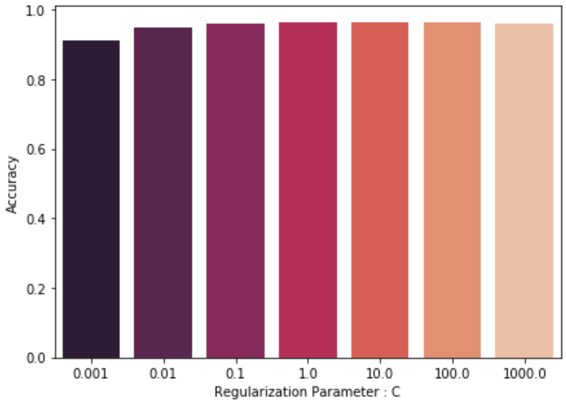


Fig 25: C vs Accuracy

E. Using Principal Component Analysis for dimensionality reduction using 10 components.

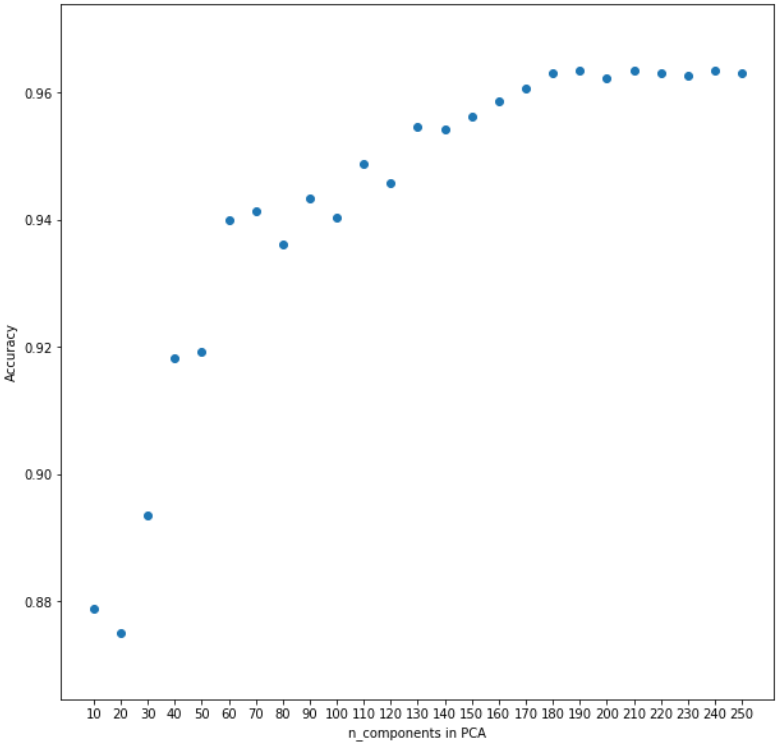


Fig 26: Number of PCA components vs Accuracy

F. Training models with 190 components

5.0 RESULTS

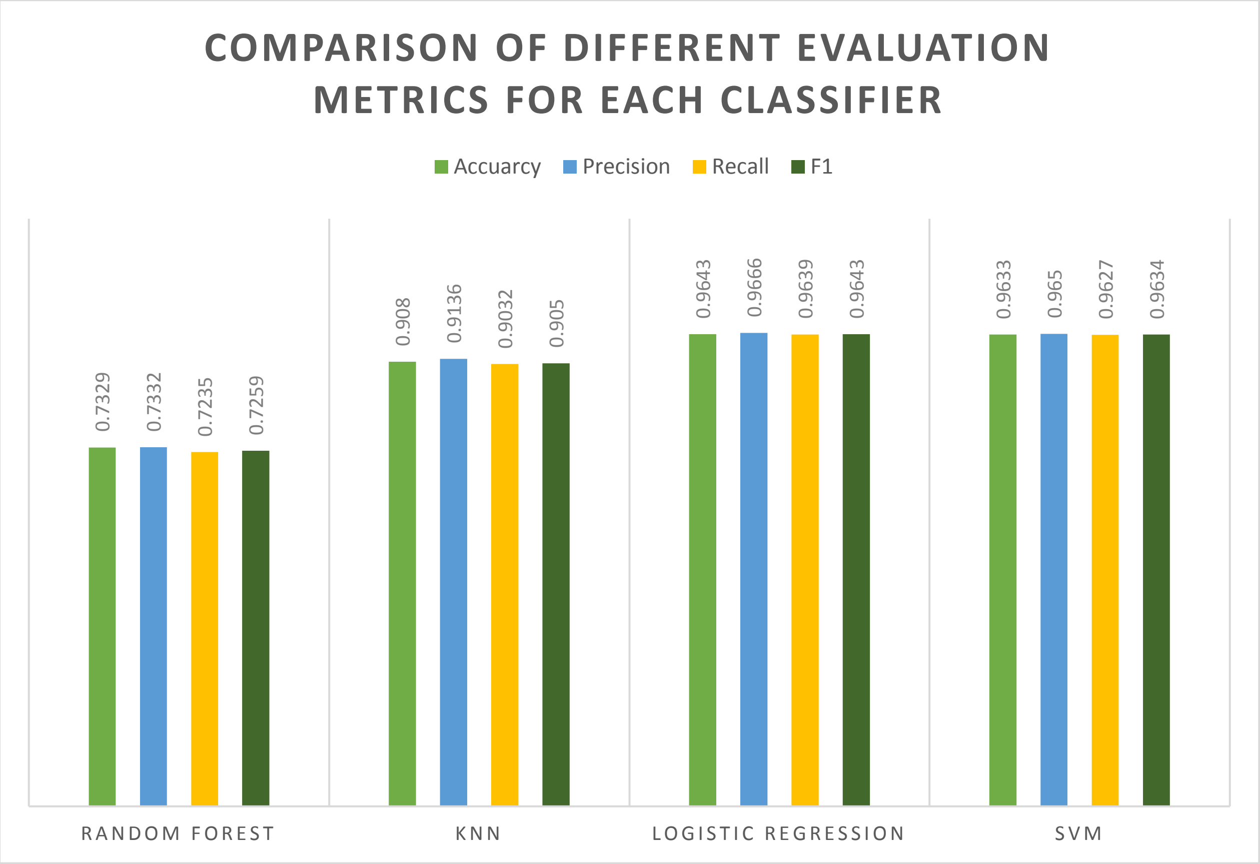


Fig 27: Visualisation of various evaluation metrics for each classifier

Classifiers	Accuracy	Precision	Recall	F1
Random Forest	0.7329	0.7332	0.7235	0.7259
KNN	0.908	0.9136	0.9032	0.905
Logistic Regression	0.9643	0.9666	0.9639	0.9643
SVM	0.9633	0.965	0.9627	0.9634

Table 1: Tabular representation of evaluation metrics for each classifier

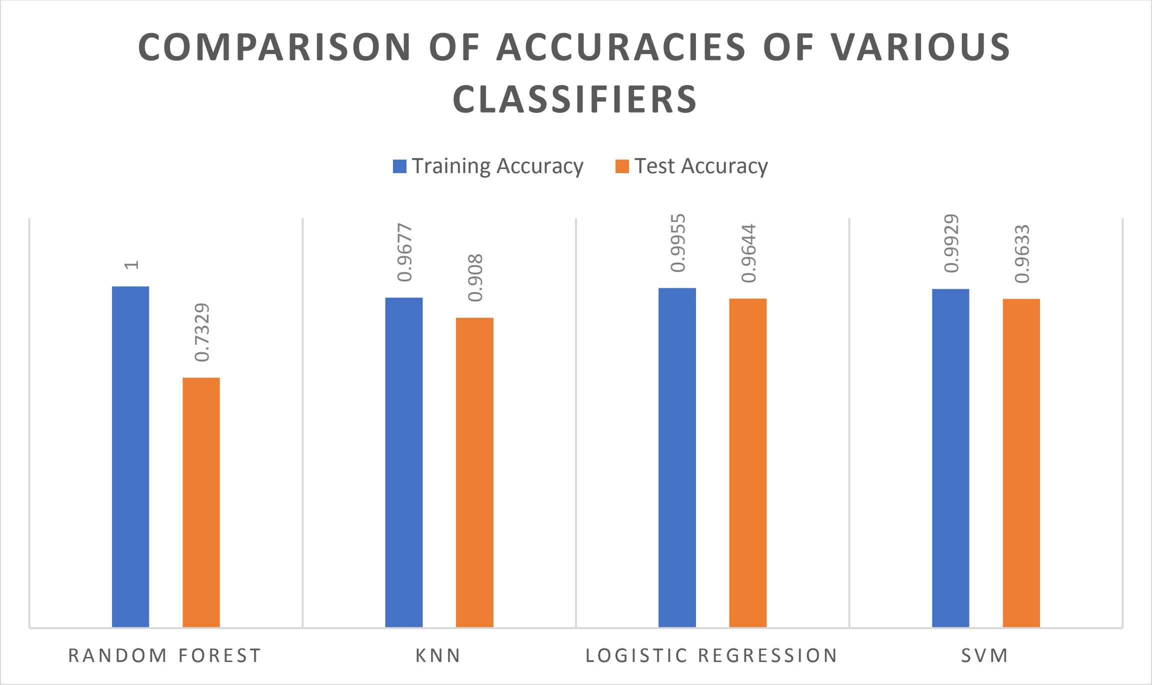


Fig 28: Comparison Chart

Classifiers	Training Accuracy	Test Accuracy
Random Forest	1.0000	0.7329
KNN	0.9677	0.9080
Logistic Regression	0.9955	0.9644
SVM	0.9929	0.9633

Table 2: Tabular representation of obtained accuracy

Activity	Precision	Recall	F1-score	Support
Laying	1.00	1.00	1.00	537
Sitting	0.75	0.73	0.74	491
Standing	0.76	0.78	0.77	532
Walking	0.55	0.67	0.60	496
Walking Downstairs	0.72	0.57	0.64	420
Walking Upstairs	0.62	0.59	0.60	471
Avg / Total	0.74	0.73	0.73	2947

Table 3: Classification Report for Random Forest

Activity	**Precision**	**Recall**	**F1-score**	**Support**
Laying	1.00	1.00	1.00	537
Sitting	0.93	0.80	0.86	491
Standing	0.84	0.94	0.89	532
Walking	0.85	0.98	0.91	496
Walking Downstairs	0.96	0.79	0.87	420
Walking Upstairs	0.91	0.91	0.91	471
Avg / Total	0.91	0.91	0.91	2947

Table 4: Classification Report for KNN

Activity	**Precision**	**Recall**	**F1-score**	**Support**
Laying	0.99	1.00	1.00	537
Sitting	0.97	0.86	0.91	491
Standing	0.91	0.98	0.94	532
Walking	0.96	1.00	0.98	496
Walking Downstairs	1.00	0.98	0.99	420
Walking Upstairs	0.97	0.96	0.97	471
Avg / Total	0.97	0.96	0.96	2947

Table 5: Classification Report for Logistic Regression

Activity	**Precision**	**Recall**	**F1-score**	**Support**
Laying	1.00	1.00	1.00	537
Sitting	0.96	0.89	0.92	491
Standing	0.91	0.97	0.94	532
Walking	0.96	0.99	0.98	496
Walking Downstairs	0.99	0.97	0.98	420
Walking Upstairs	0.98	0.96	0.97	471
Avg / Total	0.96	0.96	0.96	2947

Table 6: Classification Report for SVM

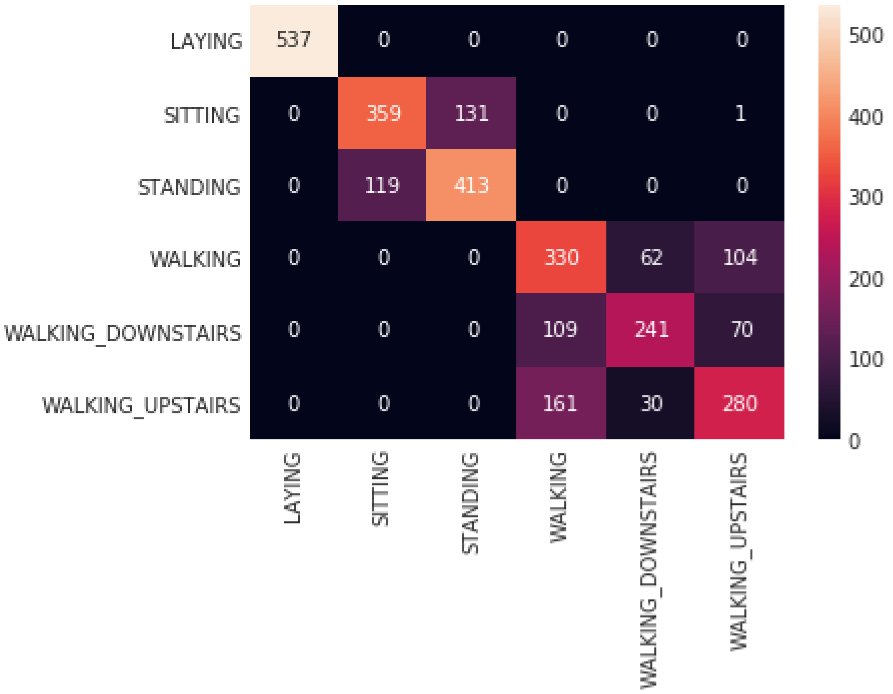


Fig 29: Confusion Matrix for Random Forest

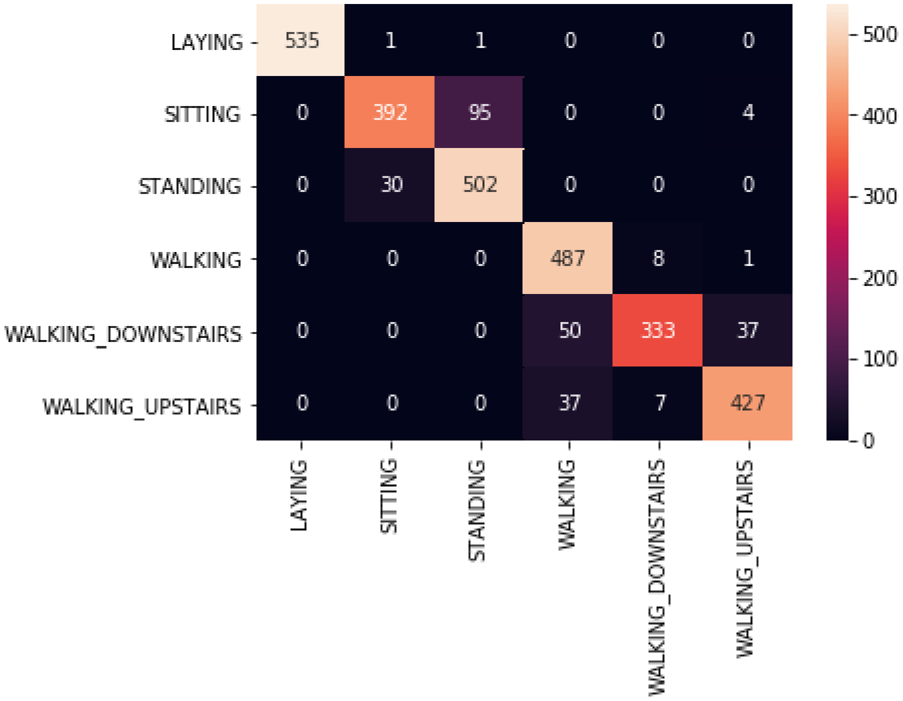


Fig 30: Confusion Matrix for KNN

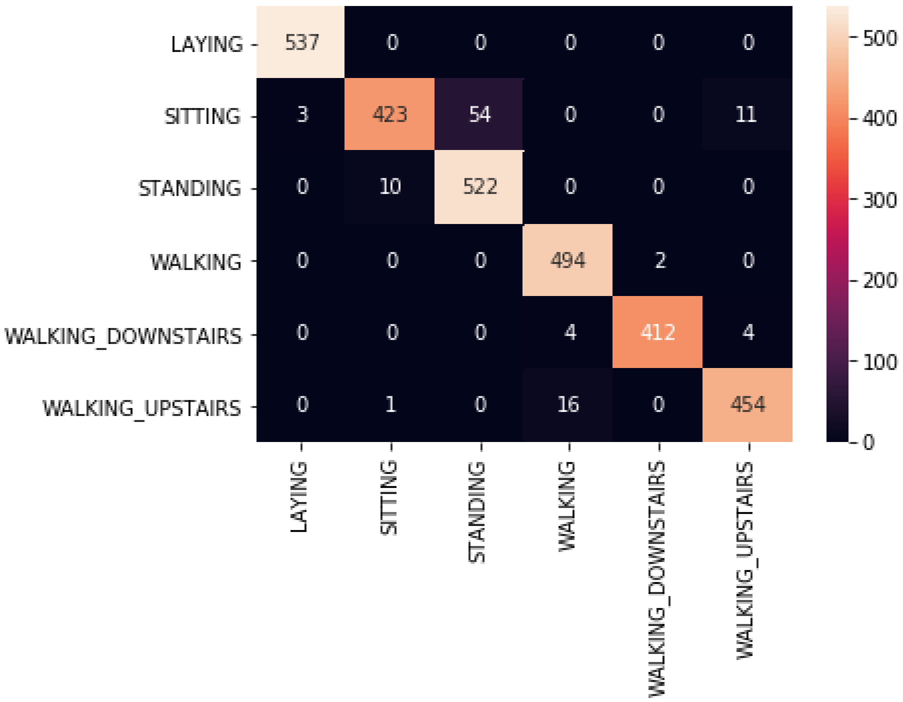


Fig 31: Confusion Matrix for Logistic Regression

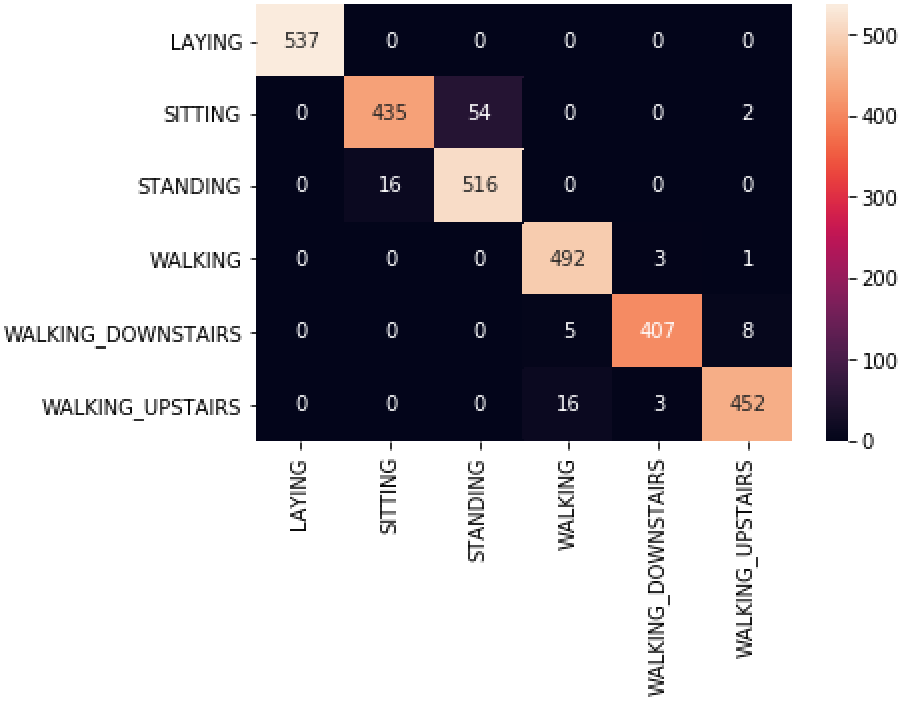


Fig 32: Confusion Matrix for SVM

The best classification rate in our experiment were 96.33% and 96.44%, which were achieved by SVM and Logistic Regression respectively. Classification performance is robust to the orientation and the position of smartphones. Among the four classifiers, KNN and SVM improve most after applying feature extraction and dimensionality reduction using PCA. Conclusively, Logistic Regression and SVM are the optimal choice for our problem. Therefore, we will compare the SVM model with all other models which have given the best accuracies in the recent times.

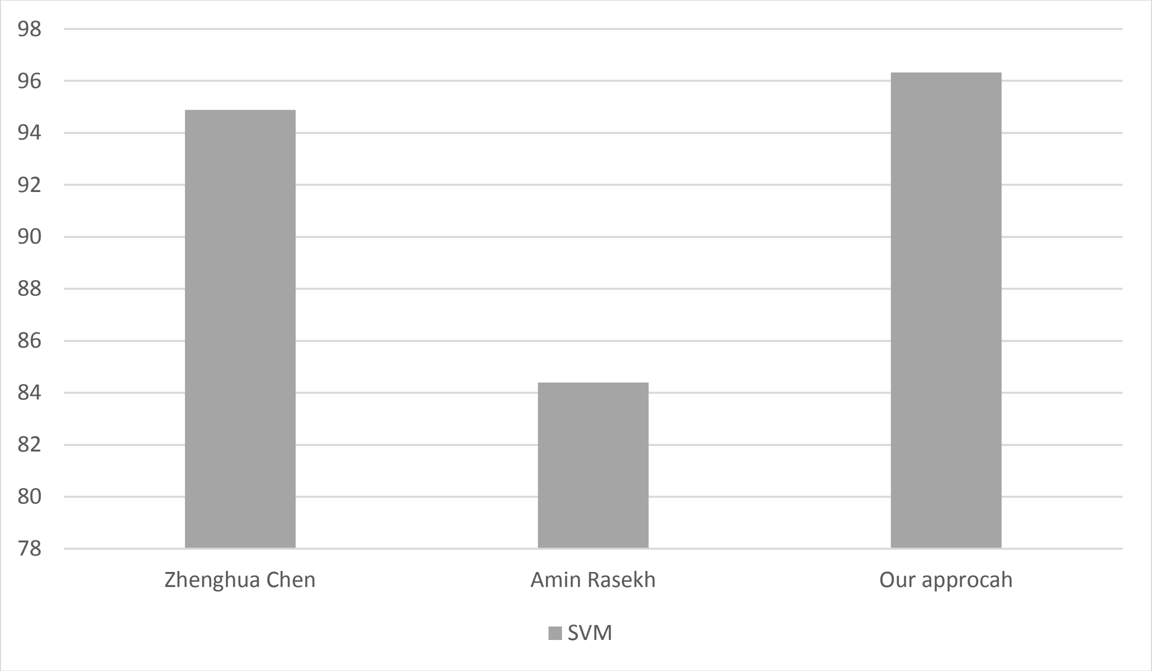


Fig 33: Comparison of Accuracies for SVM with other related works

6.0 Conclusion and Future Work

In this project, we designed a smartphone-based recognition system that recognizes six human activities: walking, sitting, standing, laying, going upstairs and going downstairs. We used several classifiers like KNN, SVM, Logistic Regression, etc. for predicting the outputs to the corresponding training and test datasets. So, finally we compared the accuracy and other metrics for the classifiers analyzed in our work.

The best classification rate in our experiment were 96.33% and 96.44%, which were achieved by SVM and Logistic Regression respectively. Classification performance is robust to the orientation and the position of smartphones. Among the four classifiers, KNN and SVM improve most after applying feature extraction and dimensionality reduction using PCA. Conclusively, Logistic Regression and SVM are the optimal choice for our problem.

From this project, we noticed the following facts which will guide us for the future work:

- Adding more data is still improving our test error on the margin. We would like to try to add more data to see how much this will improve the model.

- Having a better definition of activities would improve this model. We would like to allow the possibility of tagging multiple activities at the same data point since some of activities can be done simultaneously.

- Sensor data collection from different types of phones can be significantly different. The reason for this could be the sensors calibration and precision are different for each phone. We end up using the data from the same phone.

- There exists an activity pattern for each user. Using one user's training data to predict another user's behaviour is performing worse than predicted this user's behaviour. Personalized model for each user might improve the prediction accuracy. We would like to solve these problems in the future.

REFERENCES

[1] **Recognizing human activities from multi-modal sensors.** Shu Chen and Yan Huang. *Intelligence and Security Informatics, 2009. ISI '09. IEEE International Conference on, Dallas, TX, 2009, pp. 220-222*

[2] **Human Activity Recognition using Smartphone.** Amin Rasekh, Chien-An, Chen Yan Lu. *Texas A&M University*. 2011 Fall CSCE666

[3] **Human Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine.** Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz. *International Workshop of Ambient Assisted Living (IWAAL 2012)*. Vitoria-Gasteiz, Spain. Dec 2012

[4] **Human activity recognition using smartphone's sensors and machine learning**; Enrique Alejandro Garcia Ceja. *Instituto Tecnológico y de Estudios Superiores de Monterrey;* 2012

[5] **A Public Domain Dataset for Human Activity Recognition Using Smartphones.** Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge, L. Reyes-Ortiz. *21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN* 2013. Bruges, Belgium 24-26 April 2013.

[6] **Human Activity Recognition via Cellphone Sensor Data.** Wei Ji, Heguang Liu, Jonathan Fisher. *Stanford University*. 2016 Fall CS229 Project Report

[7] **Robust Human Activity Recognition Using Smartphone Sensors via CT-PCA and Online SVM.** Zhenghua Chen, Qingchang Zhu, Yeng Chai Soh, Le Zhang. *IEEE Transactions on Industrial Informatics* PP (99):1-1 · June 2017

[8] **A Brief Introduction into Machine Learning.** Gunnar Ratsch. Friedrich Miescher Laboratory of the Max Planck Society, Spemannstrabe 37, 72076 Tubingen, Germany. 25 September 2012.

[9] **Supervised Machine Learning: A Review of Classification Techniques.** S. B. Kotsiantis.Department of Computer Science and Technology University of Peloponnese, Greece. 17 August 2014.

[10] **Unsupervised Learning.** Peter Dayan. MIT. 21 September 2015.

[11] **An Introduction to Logistic Regression Analysis and Reporting.** CHAO-YING JOANNE PENG, KUK LIDA LEE, GARY M. INGERSOLL. Indiana University-Bloomington. 27 October 2013.

[12] **k-Nearest neighbour classifiers.** Padraig Cunningham, Sarah Jane Delany. Georgia Tech University. 9 January 2015.

[13] **Data Classification Using Support Vector Machine.** DURGESH K. SRIVASTAVA - Ass. Prof., Department of CSE/IT, BRCM CET, Bahal, Bhiwani, Haryana, India-127028, LEKHA BHAMBHU - Ass. Prof, Department of CSE/IT, BRCM CET, Bahal, Bhiwani, Haryana, India-127028. 14 February 2012.

[14] **Multiclass Classification and Support Vector Machine.** Yashima Ahuja & Sumit Kumar Yadav**.** Lovely Professional University, Jalandhar (Punjab) India. 21 December 2016.

[15] **Analysis of a Random Forests Model.** Gerard Biau. Universite Pierre et Marie Curie. 12 March 2012.

[16] Random **Forest.** Leo Breiman**.** Statistics Department University of California Berkeley, CA 94720. January 2001.

[17] **Narrowing the Gap: Random Forests in Theory and In Practice.** Misha Denil, David Matheson, Nando de Freitas.University of Oxford, United Kingdom, University of British Columbia, Canada. 2 April 2014.

[18] **Performance Evaluation in Machine Learning: The Good, The Bad, The Ugly and The Way Forward.** Peter Flach. Intelligent Systems Laboratory, University of Bristol, UK The Alan Turing Institute, London, UK. 17 September 2011.

[19] **Performance Evaluation of Machine Learning Algorithms.** Mohamad Hazim, Adewole K. S., Nor Badrul Anuar, Amirrudin Kamsin. February 2018.

[20] **Evaluation and Performance Analysis of Machine Learning Algorithms.** International Journal of Engineering Sciences & Research Technology, 2014.

[21] **Human Activity Recognition Using Multinomial Logistic Regression.** Ramin Madarshahian, Juan M. Caicedo. University of South Carolina. Jan 2014.

[22] **Human Activity Recognition using Android Smartphone (using KNN algorithm).** Usharani J, Dr. Usha Sakthivel.PG Scholar, Professor and HOD, Department of Computer Science. 7 May 2017.

[23] **The research of the fast SVM classifier method.** Yujun Yang, Jianping Li, Yimei Yang. 13 June 2017.

[24] **Human Activities Recognition in Android Smartphone Using Support Vector Machine.** Duc Ngoc Tran, Duy Dinh Phan. July 2017.

[25] **Introduction to Machine Learning.** Alex Smola and S.V.N. Vishwanathan. Departments of Statistics and Computer Science Purdue University –and– College of Engineering and Computer Science Australian National University. 21 August 2017.