Acts, New Delhi

Human Activity Recognition Using Smartphones

Submitted in partial fulfillment of the requirements of

PG-DIPLOMA IN BIG DATA ANALYTICS

By

Teena Thakre	230310125020
Ankush Kuchankar	230310125002
Viraj Khamankar	230310125021

Project Guide

Mr. Pappu D. Kapgate

(SDE, C-DAC, New Delhi)

CENTER FOR DEVELOPMENT OF ADVANCED COMPUTING

New Delhi.

March 2023 – September 2023

TABLE OF CONTENTS

Acts, New Delhi

CERTIFICATE

This is to certify that the project entitled “HUMAN ACTIVITY RECOGNITION USING SMARTPHONES” is a teamwork work **of “Teena Thakre (230310125020), Ankush Kuchankar (230310125002), Viraj Khamankar (230310125021).”** Submitted to **C-DAC ACTS, New Delhi** in partial fulfillment of the requirement for the PG- Diploma in Big Data Analytics.

Mr. Pappu Kapgate
(Faculty Supervisor/Guide)

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Teena Thakre)

(Ankush Kuchankar)

(Viraj Khamankar)

ABSTRACT

This project aims to create an accurate machine learning system for recognizing human activities using smartphone sensor data. By analyzing recordings of daily activities performed by individuals wearing smartphones with inertial sensors, the project seeks to develop a real-time model that identifies activities like walking, sitting, and standing. Through meticulous data pre-processing, advanced feature extraction, and model training, the system aims to enhance context-aware applications and health monitoring. The project also focuses on integrating the model into a smartphone app for practical use. The outcomes and methodologies are doc

umented comprehensively for the benefit of researchers and practitioners in activity recognition, machine learning, and human-computer interaction. This project contributes to the evolving landscape of user-centric technologies and context-aware computing.

1. INTRODUCTION

- In this project, we aim to develop a system that can automatically recognize and classify human activities based on sensor data from smartphones.
- The project leverages the Human Activity Recognition (HAR) database, which records activities of daily living (ADL) using a waist-mounted smartphone with embedded inertial sensors.
- The goal of this project is to make a model on Smartphone that can easily recognize the human activity. Moreover, active learning models are developed in order to reduce labeling time and burden. Through testing and comparing with different learning algorithms, we find one best-fit model for our system.

2 . DESCRIPTION

The project uses a dataset of accelerometer and gyroscope data collected from 30 subjects performing six different activities:

- Walking
- Walking upstairs
- Walking downstairs
- Sitting
- Standing
- Lying down

The first step in the project is to clean the data and remove any rows that contain missing values. The next step is to perform feature selection to identify the most important features for activity recognition. This is done using the SelectKBest algorithm, which selects the top k features based on their statistical significance.

The final step is to train a machine learning model to classify the activities. The code trains four different models: logistic regression, k-nearest neighbors, decision tree, and random forest. The accuracy of each model is evaluated on a test set.

The project description is as follows:

This project uses a dataset of accelerometer and gyroscope data collected from 30 subjects performing six different activities. The data is cleaned and preprocessed, and then feature selection is used to identify the most important features for activity recognition. Four machine learning models are trained to classify the activities, and their accuracy is evaluated on a test set.

The results show that the random forest model achieves the highest accuracy of 96.7%. This suggests that the random forest model is the best model for this task.

3. SCOPE

The scope of this project covers the following aspects:

- Use of the Human Activity Recognition database for data collection.
- Recognition of six activities: **WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, And LAYING**.
- Data preprocessing, including noise filtering, feature extraction, and dimensionality reduction.
- Utilization of machine learning models to classify activities.
- Model evaluation and testing.

4. SYSTEM REQUIREMENTS

4.1 Hardware Requirements:

- Processor: Intel Core i3 and above
- Disk Space 5 GB Minimum
- RAM: 2 GB or more

4.2 Software Requirements:

- Internet: Web Browser, Operating System (Windows 7 or above).
- Internet Connection: (any).
- Transformation– Jupyter Notebook

4.3 Technologies Used:

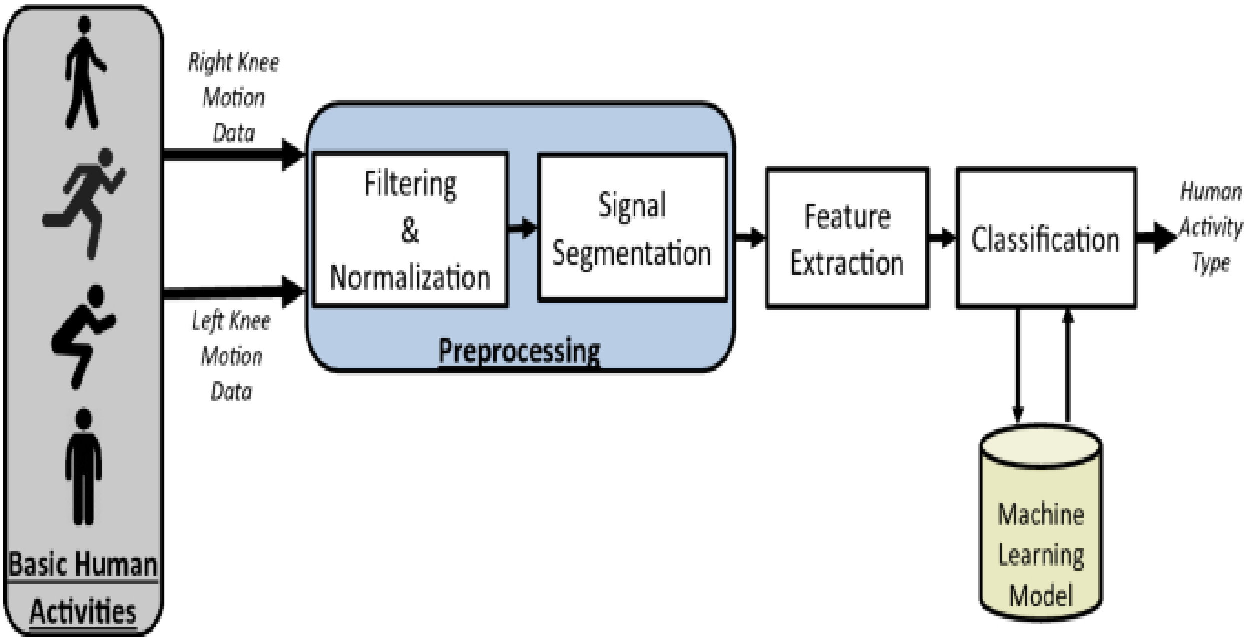
- Web Scraping– Python
- Data Cleaning – Python & Jupyter Notebook
- Visualization - Seaborn, Matplotlib

5. Data Flow Diagram

The data loading step loads the dataset of accelerometer and gyroscope data from a CSV file. The data is then cleaned to remove any rows that contain missing values. The feature selection step identifies the most important features for activity recognition.

This is done using the SelectKBest algorithm, which selects the top k features based on their statistical significance. The data visualization step visualizes the data to get a better understanding of the distribution of the data and the relationships between the features.

The model training step trains four different machine learning models: logistic regression, k-nearest neighbors, decision tree, and random forest. The accuracy of each model is evaluated on a test set. The model evaluation step evaluates the accuracy of the trained models on a test set. The results show that the random forest model achieves the highest accuracy of 96.7%.



6. Activity Flow Diagram

Data Loading: The dataset of accelerometer and gyroscope data is loaded from a CSV file using the pandas library.

Data Cleaning: The data is cleaned to remove any rows that contain missing values. This is done by filling in the missing values with the mean of the corresponding column.

Feature Selection: The feature selection step identifies the most important features for activity recognition. This is done using the SelectKBest algorithm, which selects the top k features based on their statistical significance. The value of k is typically chosen by cross-validation.

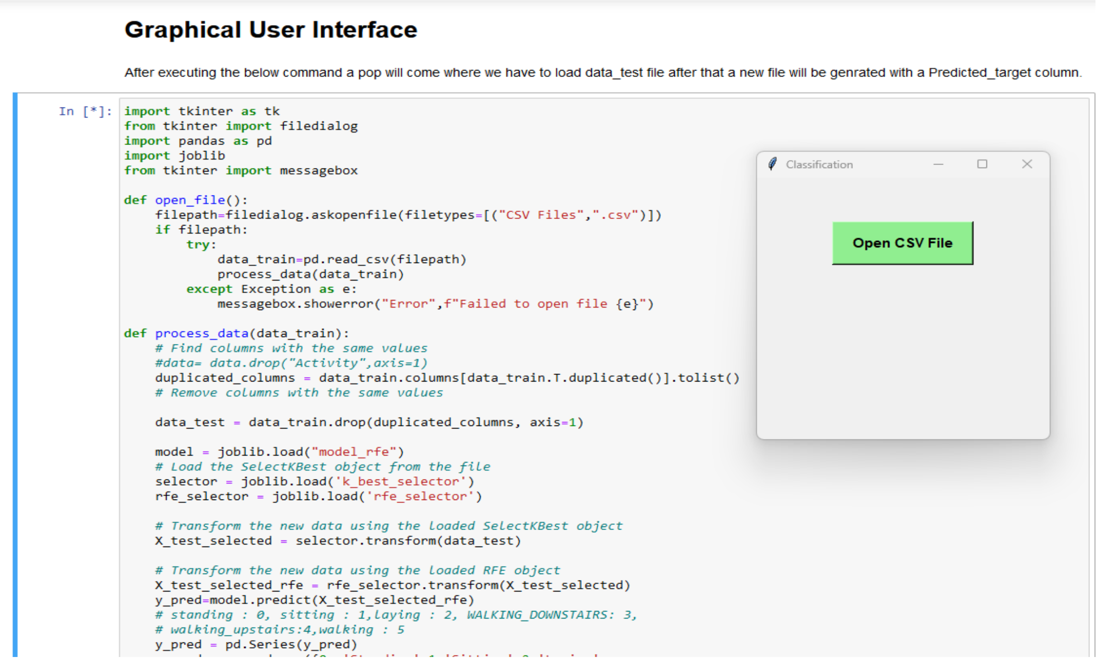
Data Visualization: The data visualization step visualizes the data to get a better understanding of the distribution of the data and the relationships between the features. This is done using the matplotlib library and seaborn library.

Model Training: The model training step trains four different machine learning models: logistic regression, k-nearest neighbors, decision tree, and random forest. The models are trained on the cleaned and preprocessed data.

Model Evaluation: The model evaluation step evaluates the accuracy of the trained models on a test set. The test set is a set of data that was not used to train the models. The accuracy of the models is measured using the accuracy score.

7. SCREENSHOTS OF RESULT

1. After executing the GUI code a pop will get generated:

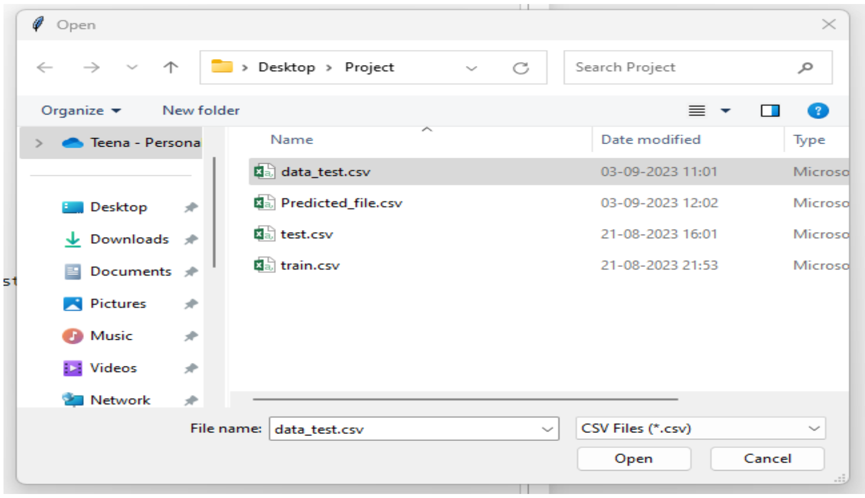


Graphical User Interface

After executing the below command a pop will come where we have to load data_test file after that a new file will be genrated with a Predicted_target column.

```
In [*]: import tkinter as tk
        from tkinter import filedialog
        import pandas as pd
        import joblib
        from tkinter import messagebox

        def open_file():
            filepath=filedialog.askopenfile(filetypes=[("CSV Files",".csv")])
            if filepath:
                try:
                    data_train=pd.read_csv(filepath)
                    process_data(data_train)
                except Exception as e:
                    messagebox.showerror("Error",f"Failed to open file {e}")

        def process_data(data_train):
            # Find columns with the same values
            #data= data.drop("Activity",axis=1)
            duplicated_columns = data_train.columns[data_train.T.duplicated()].tolist()
            # Remove columns with the same values

            data_test = data_train.drop(duplicated_columns, axis=1)

            model = joblib.load("model_rfe")
            # Load the SelectKBest object from the file
            selector = joblib.load('k_best_selector')
            rfe_selector = joblib.load('rfe_selector')

            # Transform the new data using the loaded SelectKBest object
            X_test_selected = selector.transform(data_test)

            # Transform the new data using the loaded RFE object
            X_test_selected_rfe = rfe_selector.transform(X_test_selected)
            y_pred=model.predict(X_test_selected_rfe)
            # standing : 0, sitting : 1,laying : 2, WALKING_DOWNSTAIRS: 3,
            # walking_upstairs:4,walking : 5
            y_pred = pd.Series(y_pred)
```

2. In that Open CSV file we have to load data_test.csv file for Predictions:

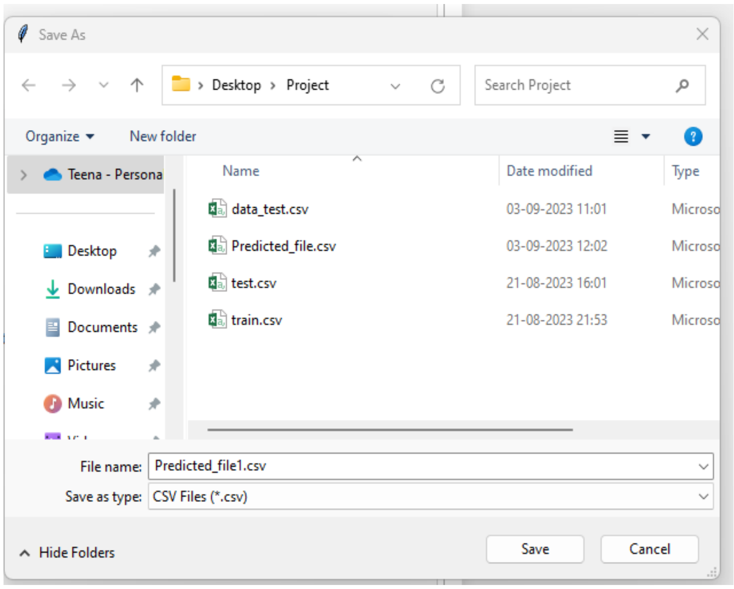


Name	Date modified	Type
data_test.csv	03-09-2023 11:01	Microso
Predicted_file.csv	03-09-2023 12:02	Microso
test.csv	21-08-2023 16:01	Microso
train.csv	21-08-2023 21:53	Microso

File name: data_test.csv

CSV Files (*.csv)

3. After opening the data_test.csv file we have to save a new file to get the Predicted_target column.

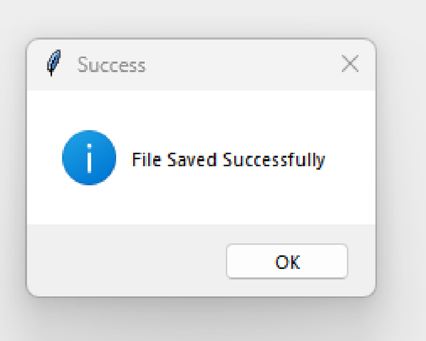


Name	Date modified	Type
data_test.csv	03-09-2023 11:01	Microso
Predicted_file.csv	03-09-2023 12:02	Microso
test.csv	21-08-2023 16:01	Microso
train.csv	21-08-2023 21:53	Microso

File name: Predicted_file1.csv

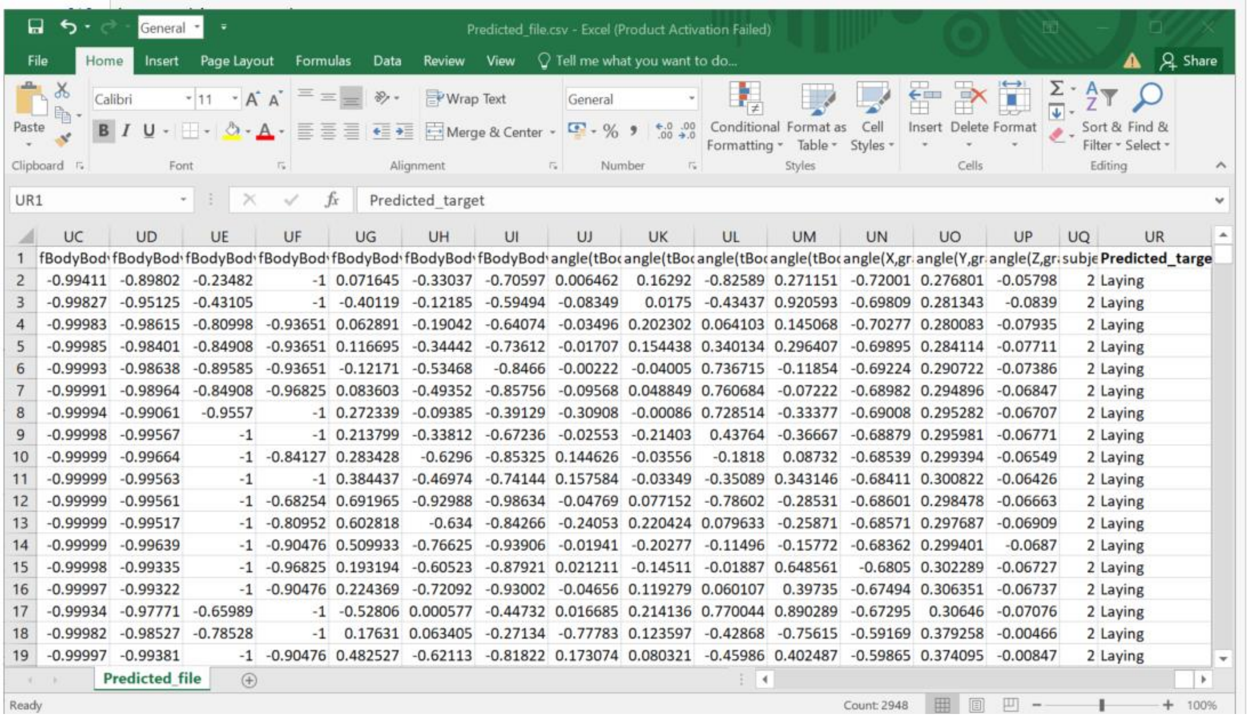
Save as type: CSV Files (*.csv)

4.The new file will get saved sucessfully:



File Saved Successfully

5.The final Predicted_file is:



	UC	UD	UE	UF	UG	UH	UI	UJ	UK	UL	UM	UN	UO	UP	UQ	UR
1	fBodyBod	fBodyBod	fBodyBod	fBodyBod	fBodyBod	fBodyBod	fBodyBod	angle(tBo	angle(tBo	angle(tBo	angle(tBo	angle(X,gr	angle(Y,gr	angle(Z,gr	subje	**Predicted_targe**
2	-0.99411	-0.89802	-0.23482	-1	0.071645	-0.33037	-0.70597	0.006462	0.16292	-0.82589	0.271151	-0.72001	0.276801	-0.05798	2	Laying
3	-0.99827	-0.95125	-0.43105	-1	-0.40119	-0.12185	-0.59494	-0.08349	0.0175	-0.43437	0.920593	-0.69809	0.281343	-0.0839	2	Laying
4	-0.99983	-0.98615	-0.80998	-0.93651	0.062891	-0.19042	-0.64074	-0.03496	0.202302	0.064103	0.145068	-0.70277	0.280083	-0.07935	2	Laying
5	-0.99985	-0.98401	-0.84908	-0.93651	0.116695	-0.34442	-0.73612	-0.01707	0.154438	0.340134	0.296407	-0.69895	0.284114	-0.07711	2	Laying
6	-0.99993	-0.98638	-0.89585	-0.93651	-0.12171	-0.53468	-0.8466	-0.00222	-0.04005	0.736715	-0.11854	-0.69224	0.290722	-0.07386	2	Laying
7	-0.99991	-0.98964	-0.84908	-0.96825	0.083603	-0.49352	-0.85756	-0.09568	0.048849	0.760684	-0.07222	-0.68982	0.294896	-0.06847	2	Laying
8	-0.99994	-0.99061	-0.9557	-1	0.272339	-0.09385	-0.39129	-0.30908	-0.00086	0.728514	-0.33377	-0.69008	0.295282	-0.06707	2	Laying
9	-0.99998	-0.99567	-1	-1	0.213799	-0.33812	-0.67236	-0.02553	-0.21403	0.43764	-0.36667	-0.68879	0.295981	-0.06771	2	Laying
10	-0.99999	-0.99664	-1	-0.84127	0.283428	-0.6296	-0.85325	0.144626	-0.03556	-0.1818	0.08732	-0.68539	0.299394	-0.06549	2	Laying
11	-0.99999	-0.99563	-1	-1	0.384437	-0.46974	-0.74144	0.157584	-0.03349	-0.35089	0.343146	-0.68411	0.300822	-0.06426	2	Laying
12	-0.99999	-0.99561	-1	-0.68254	0.691965	-0.92988	-0.98634	-0.04769	0.077152	-0.78602	-0.28531	-0.68601	0.298478	-0.06663	2	Laying
13	-0.99999	-0.99517	-1	-0.80952	0.602818	-0.634	-0.84266	-0.24053	0.220424	0.079633	-0.25871	-0.68571	0.297687	-0.06909	2	Laying
14	-0.99999	-0.99639	-1	-0.90476	0.509933	-0.76625	-0.93906	-0.01941	-0.20277	-0.11496	-0.15772	-0.68362	0.299401	-0.0687	2	Laying
15	-0.99998	-0.99335	-1	-0.96825	0.193194	-0.60523	-0.87921	0.021211	-0.14511	-0.01887	0.648561	-0.6805	0.302289	-0.06727	2	Laying
16	-0.99997	-0.99322	-1	-0.90476	0.224369	-0.72092	-0.93002	-0.04656	0.119279	0.060107	0.39735	-0.67494	0.306351	-0.06737	2	Laying
17	-0.99934	-0.97771	-0.65989	-1	-0.52806	0.000577	-0.44732	0.016685	0.214136	0.770044	0.890289	-0.67295	0.30646	-0.07076	2	Laying
18	-0.99982	-0.98527	-0.78528	-1	0.17631	0.063405	-0.27134	-0.77783	0.123597	-0.42868	-0.75615	-0.59169	0.379258	-0.00466	2	Laying
19	-0.99997	-0.99381	-1	-0.90476	0.482527	-0.62113	-0.81822	0.173074	0.080321	-0.45986	0.402487	-0.59865	0.374095	-0.00847	2	Laying

8. CONCLUSION

- Human activity recognition (HAR) using smartphones is a versatile technology with widespread applications.
- Smartphones' portability, sensors, and algorithms allow us to track fitness, monitor health, analyze sports, enable gesture control, adapt to contexts, enhance security, and manage diseases.
- Despite challenges like data quality and privacy, smartphone-based HAR continues to advance, promising a future where technology enhances our understanding of human behavior and enriches various aspects of our lives.

9. References

- Kaggle - https://www.kaggle.com/datasets/uciml/human-activity-recognition-with-smartphones
- Model Study - https://www.javatpoint.com/machine-learning
- https://towardsdatascience.com/activity-recognition-using-smartphones-machine-learning-application-a10e7b5578f9
- https://ieeexplore.ieee.org/document/8567275
- https://www.niser.ac.in/~smishra/teach/cs460/2021/project/21cs660 _group22/