Physical Activity Prediction

A project report submitted to ICT Academy of Kerala

in partial fulfillment of the requirements

for the certification of

CERTIFIED SPECIALIST

IN

DATA SCIENCE & ANALYTICS

submitted by

Yedhukrishna P R

Jenis Mathew

Zainal Abdeen

Jinu K V

Neeraja T V

Arjun R

ICT ACADEMY OF KERALA

THIRUVANANTHAPURAM, KERALA, INDIA

July 2023

List of Figures

List of Abbreviations

1. TTM - Trans-Theoretical behavior Model
2. FBM - Fogg Behavior Model
3. SVM - Support Vector Machine
4. EDA - Exploratory Data Analysis
5. PCA - Principal Component Analysis
6. KNN - K Nearest Neighbor
7. SVM - Support vector machine

Table of Contents

Abstract

This represents an abstract on our dataset **"Physical Activity Prediction dataset"** which is based on exploring the Physical Activity Monitoring Dataset.This dataset consists of data collected from wearable devices worn by 8 subjects performing 18 different activities. Main aim of our project is to create insights between users and various physical activities. The findings from this analysis we tend to develop and understand user's behaviour, health, lifestyles etc. so that we can study deeply and recommend on applications and systems that can benefit individual user's needs and their ongoing activities**.**

First of all, we will examine and understand the dataset's format and structure including the number of columns, records and key variables. Will determine the data types, potential values and examine how we can get accurate quality insights which will be done by data cleaning, data preprocessing, feature selection and subsequent analysis.

There are following fields in our dataset which are Heart rate, Hand temperature, Hand acceleration, Chest temperature, Chest acceleration, Ankle temperature, Ankle acceleration, Peopleld etc. by which we can accurately predict the specific activity the user is engaged in.

Our Project will be done in following steps:

- **Explorartory Data Analysis:** To help identify various aspects of our domain and data ,understand the patterns within the data, detect outliers, develop/find relations among the variables which give key insights.
- **Data Preprocessing:** This will be performed on our dataset to improve the accuracy and quality of our dataset which will be suitable for our further analysis which can be done by handling the missing values, addressing the outliers, scaling of data if required making it more reliable where we will only select relevant features as needed for the model which will mainly include the data cleaning.
- **Predictive Modeling:** It will be done to create accurate models for predicting the physical activites of the user or predict the behaviours/patterns of the users from the input we have. Algorithms like regression,classification will be done so that it will create a better understanding of the user activities.
- **Fine Tuning:** Fine-tuning requires tuning hyperparameters like learning rate, batch size, and the number of epochs. Experiment with different values and monitor the validation performance to find the best combination of hyperparameters.
- **Model Deployment:** After developing and fine-tuning a model, the next step is to deploy it so that it can be integrated into systems or applications to make predictions.

1. Problem Definition

1.1 Overview

Our objective is to predict physical activity using the fitness data.

1.2 Problem Statement

The PAMAP2 Physical Activity Monitoring dataset contains data from 8 subjects performing 18 different physical activities. The goal is to develop an algorithm that can accurately recognize the different activities from the sensor data.

Challenges: The challenges in this problem include the following:

The sensor data is noisy and contains a lot of variation.

The activities are often performed in a similar way, making it difficult to distinguish them.

The dataset is relatively small, which can make it difficult to train an accurate algorithm.

Approaches: Some possible approaches to solving this problem include the following:

Feature selection: This involves selecting a subset of the sensor data that is most informative for activity recognition.

Dimensionality reduction: This involves reducing the number of dimensions in the sensor data, which can make it easier to train an accurate algorithm.

Machine learning: This involves using machine learning algorithms to learn the patterns in the sensor data and identify the different activities.

Potential benefits: The potential benefits of solving this problem include the following:

The development of a more accurate and reliable activity recognition algorithm.

The use of activity recognition to improve health monitoring and fitness tracking.

The development of new applications for wearable sensors.

2. Introduction

Physical activity plays a pivotal role in maintaining overall health and well-being, impacting various aspects of an individual's life, from reducing the risk of chronic diseases to enhancing mental well-being. As the world becomes more sedentary due to technological advancements and lifestyle changes, predicting and promoting physical activity levels have gained increasing importance in public health and healthcare research.

Main aim of our project is to create insights between users and various physical activities. The findings from this analysis we tend to develop and understand user's behaviour, health, lifestyles etc. so that we can study deeply and recommend on applications and systems that can benefit individual user's needs and their ongoing activities**.**

In this report, we present a comprehensive analysis of the physical activity prediction dataset, consisting of data collected from different individuals. The dataset includes a wide range of features, such as Heart rate, Hand temperature, Hand acceleration, Chest temperature, Chest acceleration, Ankle temperature, Ankle acceleration, Peopleld etc. Our objective is to explore the relationships between these features and physical activity levels to build an accurate predictive model.

By successfully predicting physical activity levels, this project can contribute to public health initiatives by enabling personalized recommendations and interventions for individuals at risk of leading sedentary lifestyles. Moreover, the predictive model can be integrated into wearable devices, mobile applications, or health monitoring systems, providing real-time feedback and encouragement for users to engage in more physical activity.

3. Literature Survey

1.Physical Activity Prediction using Fitness Data: Challenges and Issues - Universiti Teknologi MARA, Malaysia.

They observed 10 models that are most appropriate for predicting physical activities using fitness data. They believe there is a need to develop an application for predicting suitable physical activities using fitness data and personal context data, and have taken a first step towards this goal by constructing the framework called fitness personalization, which consists of combination of TTM (trans-theoretical behavior model) and FBM (Fogg behavior model).

https://www.researchgate.net/publication/348938039_Physical_activity_prediction_using_fitness_data_Challenges_and_issues

2.Physical Activity Monitoring and Classification Using Machine Learning Techniques- Najran University Saudi Arabia, Edge Hill University & Northumbria University in UK.

The study utilized motion sensors' data of 30 participants, recorded while performing a variety of daily life activities. The findings suggest that the class imbalance plays a significant role in the performance of the system, and the underrepresentation of physical activity during the training stage significantly impacts the performance of machine learning classifiers. The SVM (support vector machine) proved itself to be the best performance among all classifiers.

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9332439/

4. Exploratory Data Analysis

For performing EDA the following steps are done.

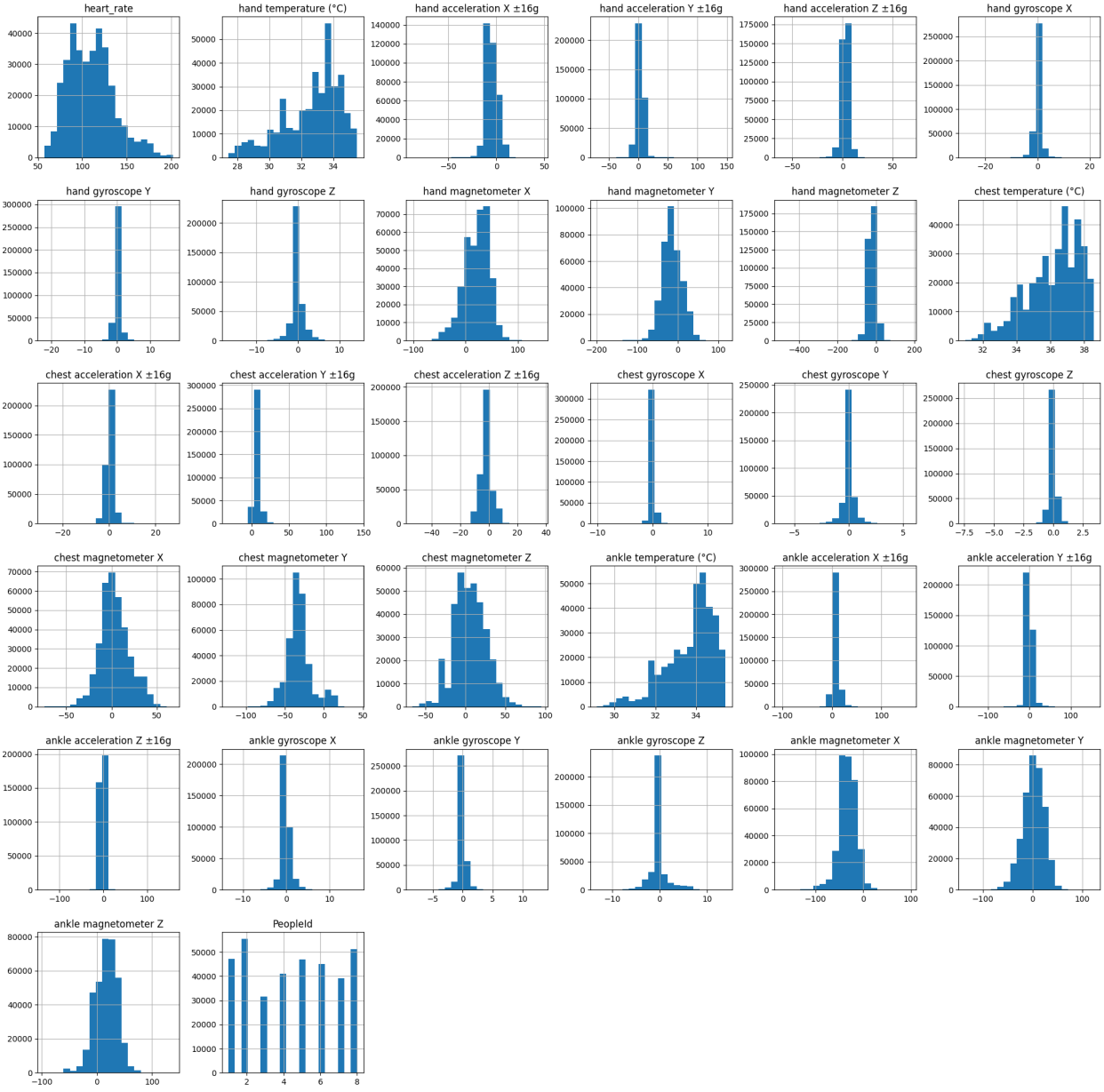
1. Importing the necessary libraries, plotting packages, ML packages etc.
2. Importing our CSV file dataset.
3. Checked the basic information of the dataset:
    a. Printed the first few rows and last few rows along with the columns.
    b. Dropped out the unnamed column.
    c. Checked for the shape of the dataset-(358007 rows & 33 columns), in that we got 32 numerical values and one categorical value.
    d. Checked for the missing values and we got 9 no. of null values of feature "Heart rate".
    e. Checked for the Statistics data of the dataset.
4. Data splitting done by separating the Numerical and categorical features by appending the data into the list form.

Descriptive Analysis

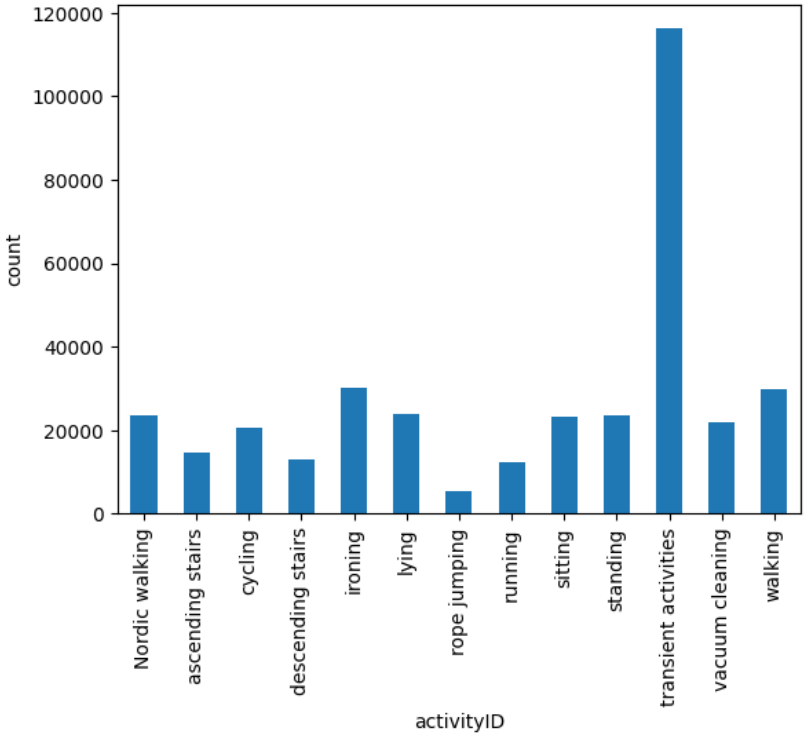
1. In descriptive Analysis, we analyze each variable separately to get inferences about the features like mean, and mode. From this we observed that:
    a. Features such as ankle acceleration X, and chest gyroscope X have high positive skew ranges.
    b. And ankle gyroscope Y has a high negative skew.
    c. Rest of the features show a moderate range of positive and negative skew.
2. We plot Histograms to find the distributions in each feature.
3. We do some preprocessing to find missing values in the dataset.
4. From the above steps we observed that:
    a. hand temperature (°C), chest temperature (°C), ankle temperature (°C) contains mostly non-unique values but the rest of the features show more variations.
    b. heart_rate contains missing values so we need to use any statistical imputations.
    c. Except 'PeopleId' all other features show gaussian-like distribution.

5. We do data visualization to get some more insights.
6. We created a bar plot to find the distribution of target variable (activityIDs).
7. We do correlation analysis to get the correlation matrix of the whole dataset.
8. Also did some other visualizations of features to get more insights Density plots of numerical features.

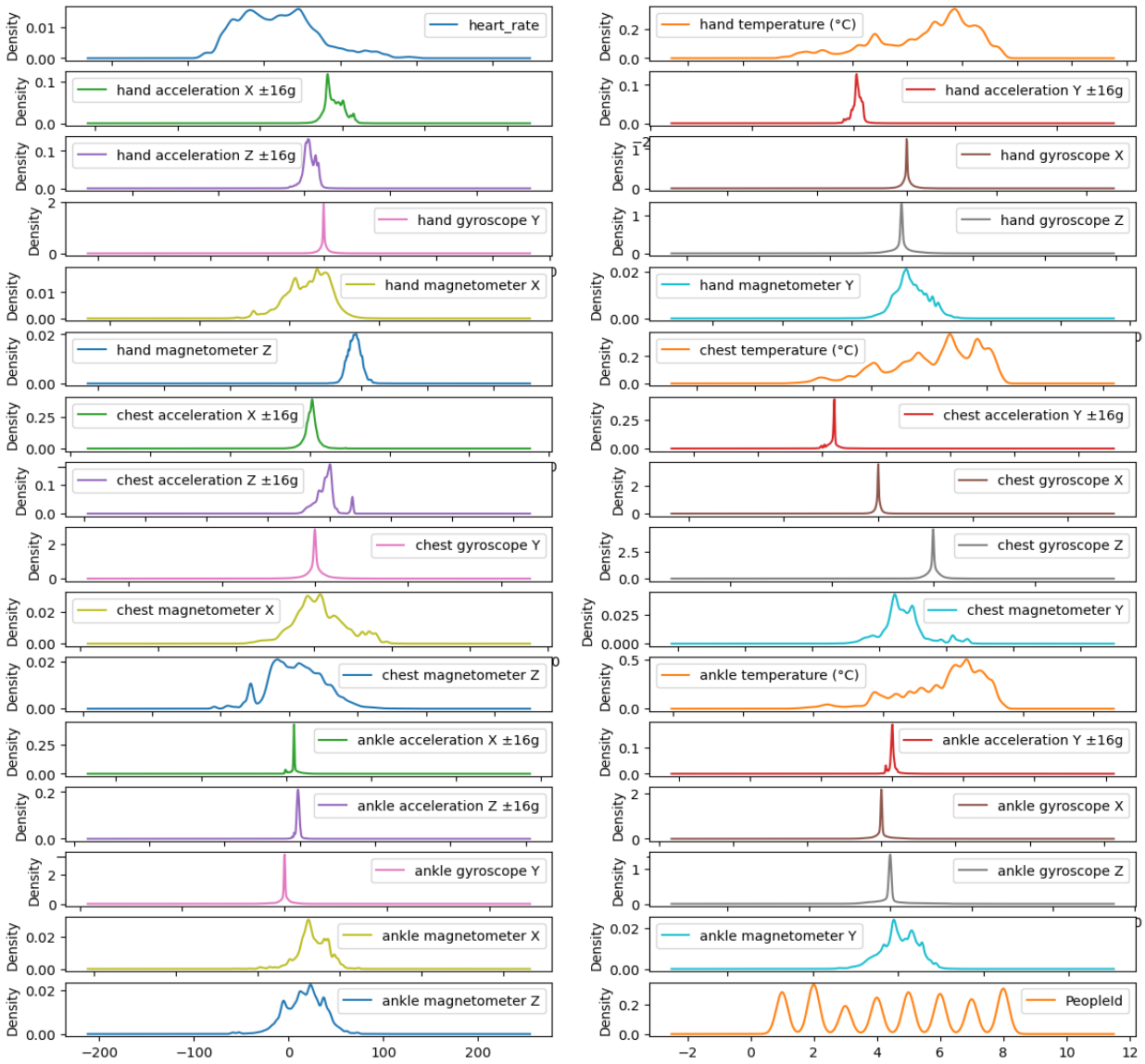
4.1 Histogram plots for each variable



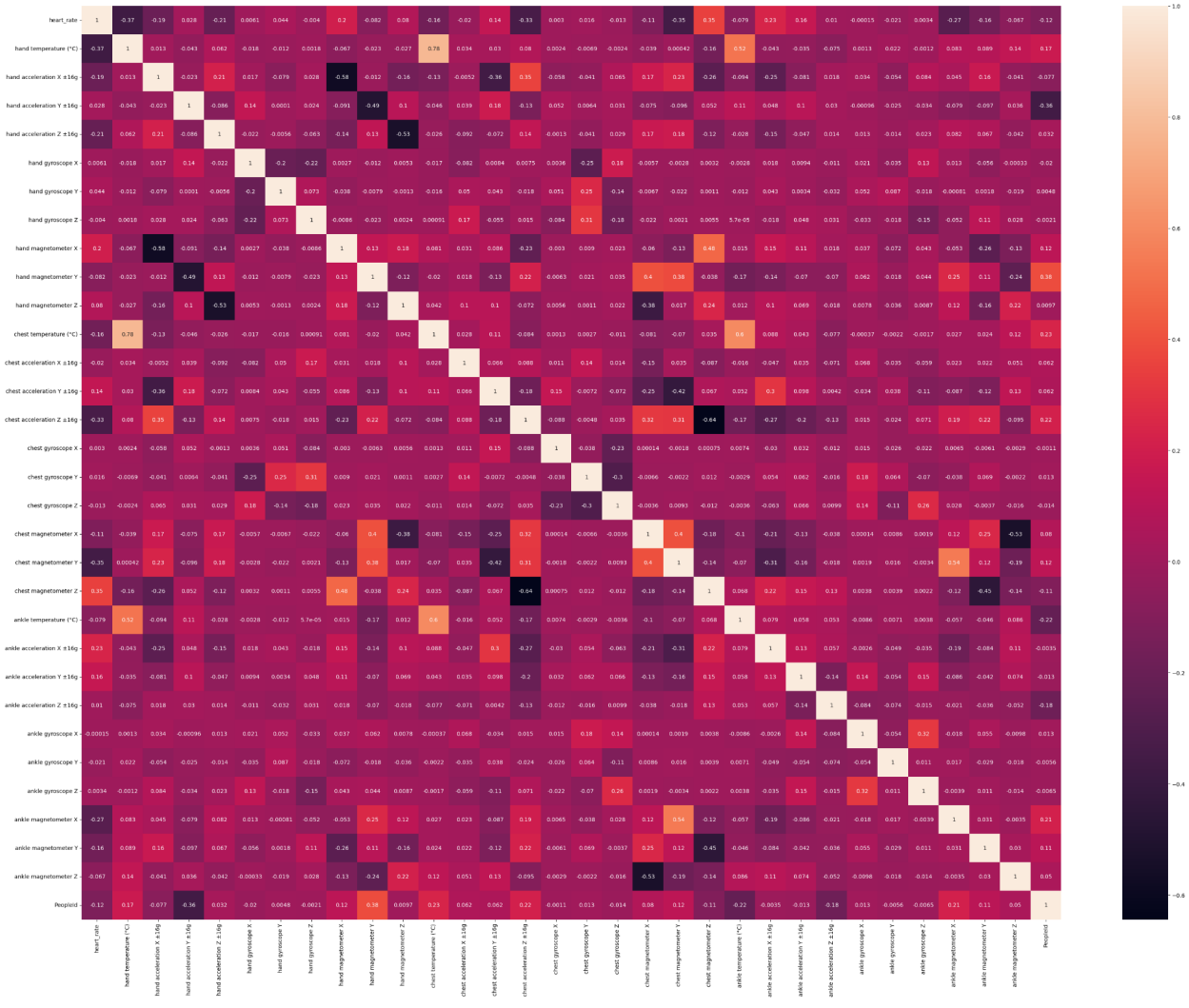
4.2 Barplot of unique value counts in every categorical features



4.3 Density plot of numerical features



4.4 Correlation heatmap of all features



5. Data Preprocessing

Part I

Preprocessing involves a series of operations that aim to enhance the quality of data, eliminate inconsistencies, and ensure it is ready for the specific analysis task.

In this section, we did some data cleaning which involves identifying and handling missing values and also outlier detection and removal.

Handling Missing Values

- We identified 9 missing values in the column called heart_rate.
- By using the SimpleImputer object with the 'mean' strategy, we replaced missing values with the mean of the non-missing values in the "heart_rate" column.

Steps followed in the above process:

1. Initialized the SimpleImputer with 'mean' strategy.
2. Fit and transform the 'heart_rate' column to fill missing values.
3. Checked the number of missing values after imputation.

Managing Outliers

We detected outliers using the Z-score method.

Steps followed are:

1. Defined a function to detect outliers using Z-score.
2. Found outliers in all columns (except the 'activityID' and 'PeopleId' columns if they exist) and the index of potential outliers.
3. Removed the potential outliers from the original DataFrame.
4. After removing outliers and storing the cleaned DataFrame in 'df_cleaned‘, reset the index of the Dataframe.

Part II

In this section we did Encoding of categorical data, splitting the dataset, Feature scaling and Principal Component Analysis.

Encoding of categorical data

- Encoding of categorical values done by using LabelEncoder class from the sci-kit library.
- We converted all categorical values in the 'activityID' column to numerical values.

Splitting the dataset

- We split the data into independent and dependent variables, x and y along with splitting the dataset to train and test sets.
- For splitting the dataset, we import the train_test_split model from the sci-kit library.

And the code includes four variables:

- x_train –independent features for the training data
- x_test – independent features for the test data
- y_train – dependent variables for training data
- y_test – dependent variables for testing data

Feature Scaling

- We did Min Max Scaling for the independent variables in the dataset.
- For scaling we import the MinMaxScaler class from the sci-kit library.
- We created the MinMaxScaler object and then fit and transform the data to perform scaling.

Principal Component Analysis

- We did PCA for independent variables and found that out of 32 columns only 27 are found to have high correlations between data points.
- Columns such as 'ankle gyroscope Z', 'ankle magnetometer X', 'ankle magnetometer Y', 'ankle magnetometer Z ' and 'PeopleId' are dropped since these variables do not really affect the target variable 'activityID'.

6. Model Evaluation

A machine learning model determines the output you get after running a machine learning algorithm on the collected data. It is important to choose a model which is relevant to the task at hand.

Steps done for evaluation of different models:

- Training the model
- Make predictions on the test set
- Evaluating the model by finding the Accuracy score and classification report.

The following models were trained for evaluation and made predictions:

- Logistic Regression model
- Decision Tree Classifier Model
- Random Forest Classifier Model
- KNN Model
- SVM Model

7. Model Selection

The below are 5 trained models and accuracy estimations for each. We need to compare the models to each other and select the most accurate.

7.1 Logistic Regression model

Accuracy: 0.7215490700898403

	precision	recall	f1-score	support
0	0.65	0.32	0.43	3158
1	0.64	0.34	0.45	2063
2	0.87	0.91	0.89	3697
3	0.54	0.12	0.20	1486
4	0.81	0.86	0.84	5529
5	0.99	0.95	0.97	4017
6	0.80	0.45	0.57	314
7	0.89	0.54	0.67	305
8	0.83	0.86	0.84	4592
9	0.74	0.74	0.74	4363
10	0.62	0.78	0.69	18553
11	0.76	0.65	0.70	3482
12	0.65	0.49	0.56	3984
accuracy			0.72	55543
macro avg	0.75	0.62	0.66	55543
weighted avg	0.72	0.72	0.71	55543

7.2 Decision Tree Classifier Model

Accuracy: 0.875339826800857

	precision	recall	f1-score	support
0	0.84	0.82	0.83	3158
1	0.63	0.62	0.63	2063
2	0.93	0.93	0.93	3697
3	0.54	0.52	0.53	1486
4	0.93	0.94	0.93	5529
5	0.99	0.99	0.99	4017
6	0.82	0.74	0.78	314
7	0.73	0.75	0.74	305
8	0.98	0.97	0.98	4592
9	0.95	0.96	0.95	4363
10	0.87	0.88	0.87	18553
11	0.81	0.79	0.80	3482
12	0.82	0.81	0.82	3984
accuracy			0.88	55543
macro avg	0.83	0.82	0.83	55543
weighted avg	0.87	0.88	0.87	55543

7.3 Random Forest Classifier Model

Accuracy: 0.9566101939038223

	precision	recall	f1-score	support
0	0.99	0.96	0.97	3158
1	0.95	0.78	0.86	2063
2	0.99	0.98	0.99	3697
3	0.97	0.59	0.73	1486
4	0.98	0.99	0.98	5529
5	1.00	0.99	1.00	4017
6	0.98	0.83	0.90	314
7	0.99	0.87	0.92	305
8	1.00	0.99	0.99	4592
9	0.99	0.99	0.99	4363
10	0.91	0.98	0.94	18553
11	0.98	0.92	0.95	3482
12	0.97	0.94	0.95	3984
accuracy			0.96	55543
macro avg	0.98	0.91	0.94	55543
weighted avg	0.96	0.96	0.96	55543

7.4 KNN Model

Accuracy: 0.970077237455665

	precision	recall	f1-score	support
0	0.97	0.98	0.98	3158
1	0.91	0.90	0.91	2063
2	0.99	1.00	0.99	3697
3	0.90	0.84	0.87	1486
4	0.98	0.99	0.99	5529
5	1.00	0.99	0.99	4017
6	0.94	0.92	0.93	314
7	0.98	0.93	0.95	305
8	1.00	0.99	1.00	4592
9	0.99	0.99	0.99	4363
10	0.97	0.96	0.96	18553
11	0.98	0.97	0.97	3482
12	0.92	0.98	0.95	3984
accuracy			0.97	55543
macro avg	0.96	0.96	0.96	55543
weighted avg	0.97	0.97	0.97	55543

7.5 SVM Model

```
Accuracy: 0.7295608807590516
              precision    recall  f1-score   support

           0       0.75      0.15      0.25      3158
           1       0.57      0.07      0.12      2063
           2       0.89      0.93      0.91      3697
           3       0.65      0.02      0.04      1486
           4       0.84      0.89      0.87      5529
           5       0.98      0.95      0.97      4017
           6       0.97      0.37      0.54       314
           7       0.94      0.52      0.67       305
           8       0.87      0.88      0.88      4592
           9       0.79      0.81      0.80      4363
          10       0.59      0.85      0.70     18553
          11       0.85      0.62      0.72      3482
          12       0.75      0.48      0.58      3984

    accuracy                           0.73     55543
   macro avg       0.80      0.58      0.62     55543
weighted avg       0.75      0.73      0.70     55543
```

We got a higher Accuracy score in the K-Nearest Neighbors (KNN) model and selected this model for prediction.

8. Fine Tuning

Fine-tuning is used to systematically search through a specified parameter space to find the best combination of hyperparameters for your model.

In this case we used Randomized search CV to do hyperparameter tuning for our KNN model.

Steps followed to perform Randomized search CV :

1. We import Randomized search CV from the sci-kit library.
2. Define a dictionary of hyperparameters.
3. Fit Randomized search CV to KNN-model.
4. Make predictions by using the best estimator.
5. Evaluate the model.

9. Flask application

We Created a Flask application for a machine learning model prediction using Visual Studio Code.

Steps involved are:

- ❖ After installing python and Visual Studio Code, create a Project Folder to store Flask application code and other related files.
- ❖ Create the necessary files and folders for the Flask application.
    - app.py to contain the Flask application code.
    - templates folder to store the HTML templates.
    - static folder to store static files like CSS and images.
- ❖ Develop the Flask Application.
- ❖ Run the Flask Application, this will start the development server, and will be able to access the application in the web browser.
- ❖ Open a web browser and navigate to the web page to access the Flask application.
- ❖ Test the application by entering input features and submitting the form.

10. Result

We created a Flask application in **pythonanywhere.com** that takes feature inputs from the user in a web page and predicts the physical activity in the result page.

Hosted webpage link: http://dsagroup8.pythonanywhere.com

11. Conclusion

The purpose of this study is to create a project using machine learning techniques in predicting physical activities from fitness data.

We observed 5 models that are most appropriate for predicting physical activities using fitness data. We also identified a few parameters and features that are most appropriate to be used in predicting the suitable physical activity based on personal context. We got a higher Accuracy score in the K-Nearest Neighbors (KNN) model and we selected this model for prediction.

We created a Flask application for a physical activity prediction model using Visual Studio Code. And successfully deployed and ran the Flask application on PythonAnywhere.com.

References

1. Kaggle.com/datasets/diegosilvadefrana/fisical-activity-dataset. https://www.kaggle.com/datasets/diegosilvadefrana/fisical-activity-dataset
2. Medium.com-Data Pre-Processing for Deep Learning (Classification or Regression). https://medium.com/@denzilsequeira/data-pre-processing-for-deep-learning-for-classification-or-regression-2bddb0b9183b
3. Physical activity prediction using fitness data: Challenges and issues. https://beei.org/index.php/EEI/article/view/2474
4. Github.com/stergiosbamp/deep-physical-activity-prediction. https://github.com/stergiosbamp/deep-physical-activity-prediction
5. Physical Activity Monitoring and Classification Using Machine Learning Techniques. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9332439/
6. Physical Activity Prediction - Data Every Day #213. https://youtu.be/w7Q7phWnOIY