

Human Behviour Predictor

[ML Model]



April 11, 2025

# Human Behaviour Prediction Model

Powered by R.F + KNN + D.T

This project focuses on Human Activity Recognition using motion sensor data from the WISDM dataset. The aim was to classify physical activities like walking, jogging, sitting, etc., based on accelerometer data. We applied and compared three machine learning models — K-Nearest Neighbors (KNN), Decision Tree, and Random Forest. After evaluating each model, Random Forest outperformed others with the highest accuracy, making it the most reliable choice for this classification task.

# Problem Statement:

Human Activity Recognition (HAR) using mobile sensor data is a growing area of interest in the field of machine learning. The goal of this project is to classify different physical activities such as walking, jogging, sitting, standing, etc., based on time-series accelerometer data. By applying multiple machine learning models including K-Nearest Neighbours (KNN), Decision Tree, and Random Forest, this project aims to identify the most effective algorithm for accurately predicting human activities from raw sensor readings.

# Dataset Description:

The dataset used in this project is the **WISDM (Wireless Sensor Data Mining) dataset**, sourced from Kaggle. It contains time-series accelerometer data collected from smartphones placed in the user's pocket while performing various physical activities.

* **Total Instances**: ~1 million rows
* **Features**:
  + timestamp: Time at which data was recorded
  + x-axis, y-axis, z-axis: Accelerometer readings on three axes
  + Derived features: rolling mean, standard deviation, exponential moving average, squared acceleration, etc.
* **Target Variable**: activity
  + **Classes**:
    - Walking
    - Jogging
    - Upstairs
    - Downstairs
    - Sitting
    - Standing

The dataset is preprocessed to handle missing values, remove duplicates, extract new features, and normalize the input data before applying the machine learning models.

# Data Preprocessing

To ensure optimal performance of machine learning models, several preprocessing steps were applied to the raw WISDM dataset:

1. **Missing Values Handling**
   * Filled missing sensor values using the **mean** of respective columns to avoid model distortion.
2. **Duplicate Removal**
   * Duplicate entries (based on timestamp) were removed to maintain data integrity.
3. **Sorting**
   * Data was sorted by timestamp to preserve the time-series nature of the dataset.
4. **Label Encoding**
   * The categorical activity labels were encoded into numerical form using **LabelEncoder** for model compatibility.
5. **Feature Engineering**  
   Additional features were created to capture the trends and patterns in the data:
   * **sq\_acc**: Squared acceleration = x^2 + y^2 + z^2
   * **Rolling Mean & Std**: For x, y, z axes to detect motion smoothness.
   * **Exponential Moving Averages (EMA)**: To give more weight to recent readings.
   * **Aggregated features** like sq\_mean, sq\_std for better sequence insights.
6. **Feature Scaling**
   * Features were normalized using **StandardScaler** to standardize input distribution.
7. **Train-Test Split**
   * The dataset was split into **training (70%)** and **testing (30%)** sets to evaluate model generalization.

# Model Building & Evaluation:

**1. Models Applied**

We applied three classic machine learning models to predict human activity based on sensor data:

* **K-Nearest Neighbours (KNN)**
* **Decision Tree**
* **Random Forest (RF)**

**2. Model Training**

The models were trained using the following setup:

* **Features:** timestamp, x-axis, y-axis, z-axis, and engineered features like sq\_acc, mean, std, and ema.
* **Target:** Encoded activity labels.
* **Cross-Validation:** 70% data for training, 30% for testing.
* **Evaluation Metric:** Accuracy, Confusion Matrix, Precision, Recall and F1-Score were used to evaluate the model performance.

**3. Performance Evaluation**

* **KNN:**  
  The KNN model showed **good performance** with a reasonable accuracy, suitable for non-linear data distributions. However, it performed slower with larger datasets.
* **Decision Tree:**  
  The Decision Tree model provided **relatively faster results** but showed very slight overfitting in some cases due to high variance in the data.
* **Random Forest (RF):**  
  The Random Forest model performed the **best** overall, with an accuracy of **94.65%**. This model, being an ensemble method, is more robust and reduces the risk of overfitting compared to Decision Tree.

**4. Model Evaluation Metrics**

For each model, we evaluated the following metrics:

* **Accuracy:** Measures the percentage of correct predictions.
* **Confusion Matrix:** Provided detailed insight into false positives, false negatives, true positives, and true negatives.

**5. Performance Comparison**

|  |  |
| --- | --- |
| **Model** | **Accuracy** |
| **KNN** | 89.89% |
| **Decision Tree** | 91.82% |
| **Random Forest** | **94.65%** |

**6. Evaluation Graphs**

We also visualized model performance using:

* **Confusion Matrix Plots**  
  To identify model errors.
* **ROC Curve and AUC**  
  To evaluate the classification performance.

## Project Details

* Type: Classification
* Deployment: N/A
* Dataset: Kaggle (~1 M records, 5 columns)

## Technologies Used

* Python, Pandas, NumPy, Scikit-learn
* Random Forest, KNN, Decision tree, Matplotlib, Seaborn
* VS Code, Google Cloud

## Dataset Features

* activity (Target)
* user
* timestamp
* x-axis
* y-axis
* z-axis

## Steps Involved

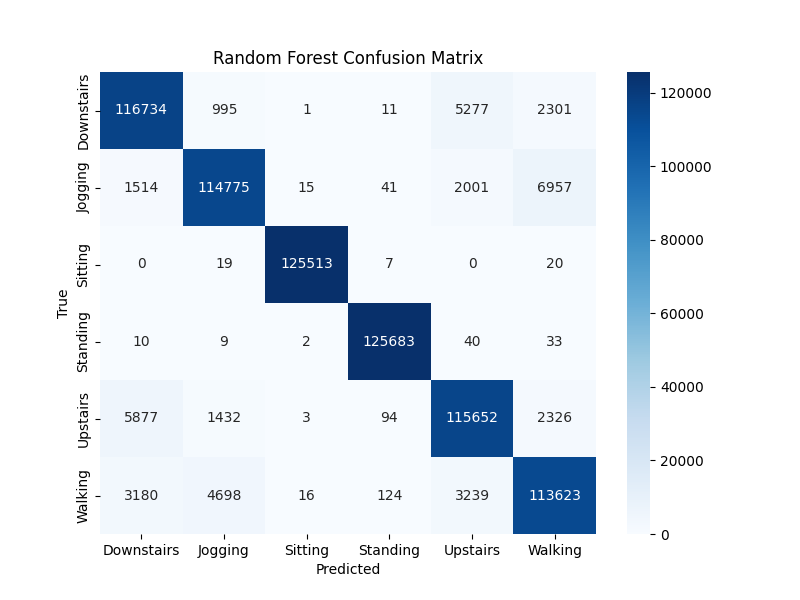
* Data cleaning & preprocessing
* Feature engineering
* Splitting data
* Model training
* Evaluation
* Visualization using graphs
* Saving trained models and encoder

## Bonus Optimizations

* Hyperparameter Tuning
* Balancing Train Data Classes

## Best Model Performance (RF)

* Accuracy: 94.65%
* Training Score: 95%
* Testing Score: 94.65%
* Highest Precision : 100%
* Highest Recall : 100%
* Highest F1-Score : 100%
* Cross Validation Score: 94.2%



--> Prediction error is ~ <6% —Excellent Model.

|  |  |
| --- | --- |
| **Accuracy** | **Remarks Considered** |
| < 70% | Poor |
| 70-80% | Average |
| 80-90% | Good |
| 90-95% | Very Good / Excellent |
| >95% | Exceptional / Almost Perfect |

## File Structure

project\_root/  
├── dataset/  
│ └── WISDM Lab - Dataset -Kaggle.zip  
├── encoder/  
│ └── encoder.pkl  
├── images/  
│ ├── ....

├──models/  
│ ├── RF\_Model.pkl

├── KNN\_Model.pkl

├── Tree\_Model.pkl

├── notebooks/  
│ ├── HBP\_Final Model.ipynb

├── Human\_Behaviour\_Prediction\_Model\_Training.ipynb  
├── project report/  
│ └── Project Report.doc  
├── source code/  
│ └── Combined\_Models.py  
├── requirements.txt  
└── Documentation.pdf

## Final Verdict

The implementation of KNN, Decision Tree, and Random Forest models on the WISDM dataset effectively demonstrated the potential of traditional machine learning algorithms in human activity recognition tasks. Among all, the **Random Forest model outperformed others with an excellent accuracy of 94.65%**, proving its robustness and efficiency in handling time-series sensor data. This project highlights how combining proper preprocessing, feature selection, and model evaluation can lead to highly reliable activity prediction systems.