Data Mining Coursework 2 Report

Author: Chibuike Lawrence Orji-Oko

Student ID removed for public version

MSc Data Science & Artificial Intelligence

Table of Contents

Table of figures:

Introduction:

The following document is a report based on the WISDM (Wireless Sensor Data Mining) dataset, the exploration of various data visualisation methods, and their implementation using Python and its libraries for the WISDM dataset on human activity recognition. This report focuses on the exploration of the WISDM, which contains sensor data collected from on-body sensing devices to record physical activity patterns. The WISDM dataset can provide valuable insights into daily living activity patterns, enabling healthcare professionals to deliver personalised healthcare solutions. This project aims to demonstrate how visualisations can be used to extract meaningful insights from complex datasets and support data mining solutions. Data visualisation is an indispensable tool that enables analysts and researchers to explore, understand, and communicate complex data effectively (Franconeri et al., 2021). This involves transforming raw data into visual representations such as graphs, charts, and maps to identify trends, patterns, and relationships inherent in the data. This report will highlight the steps that were taken to implement the various data visualisation methods, discuss the reasoning why certain methods were chosen, interpretation of the visuals, and the data mining solution.

Data Visualisation and Analysis Tools:

There are different tools available for creating visualisations, each with unique strengths and limitations for specific purposes, some are more suited for business applications than others. These libraries below were the most appropriate choice for this data set due to its complexity and volume:

Numpy (Numerical Python): NumPy is a powerful library for scientific computing. Robust for large, multi-dimensional arrays and matrices, along with a broad collection of high-level mathematical functions to handle and manipulate these arrays (Gupta and Bagchi, 2024).

Pandas: Pandas is a powerful Python library for data manipulation and analysis (Gupta and Bagchi, 2024). It provides powerful data structures (Series and DataFrame) and data analysis tools that make manipulation and preprocessing of structured (tabular, multidimensional, heterogeneous) large datasets easier (Gupta and Bagchi, 2024). Pandas seamlessly integrate with NumPy, allowing for efficient numerical operations on data.

Matplotlib: A comprehensive and flexible Python library for producing static and interactive visualisations. It provides an extensive variety of plot types, including line charts, scatter plots, bar charts, and histograms. Its flexibility allows users to fine-tune every aspect of their visualisations (Lavanya et al., 2023).

Seaborn: This is a data visualization Python library built on top of Matplotlib. It provides a higher-level interface for producing attractive and informative statistical visuals (Lavanya et al., 2023). Seaborn strength is in making visualisation more accessible and attractive, with better built-in styles and colour palettes (Lavanya et al., 2023).

Plotly: A powerful Python library for producing interactive and web-based visualizations. It supports a wide range of chart types, including bar charts, scatter plots, histograms, and line charts. Plotly's strength lies in its ability to produce highly interactive and dynamic visualisations that can be easily shared online (Lavanya et al., 2023).

Pydot: Pydot is a Python interface to the Graphviz graph visualization software. A powerful tool for creating, manipulating, and visualising graph data structures (nodes and edges) using the capabilities of Graphviz. Useful for visualising decision trees, neural networks, and other graph-based structures (Kulkarni et al., 2023).

Regardless of the visualisation tools used, below are some of the common graphs and visuals that are produced across all software platforms:

Data Visualisation Methods:

1. Line Charts: Represent data as a series of points connected by straight line segments, fitting for visualising trends over time or ordered categories (Lo et al., 2022).
2. Bar Charts: Use rectangular bars to display categorical data, suitable for easy comparison between different categories or groups.
3. Scatter Plots: Plot data points on a Cartesian coordinate system, powerful for visualising the relationship between two or more variables.
4. Histograms: Display the distribution of a single continuous variable by dividing the data into bins or ranges and showing the count or density of data points in each bin.
5. Pie Charts: Circular charts are divided into slices to display the proportions or percentages of different categories within a whole.
6. Heatmaps: Two-dimensional representation of data, where individual values are represented as colors, powerful for visualising patterns and trends in large datasets (Qu et al., 2019).
7. Box Plots: Display the distribution of a continuous variable by showing the median, quartiles, and outliers.
8. Geographic Maps: Represent spatial data on a geographic map, such as choropleth maps for visualising regional data or scatter maps for displaying point-based data (Lo et al., 2022).

The below table summarizes the pros and cons of the various data visualisation methods:

Table 1: Pros and Cons of Visualisation Methods

Method	Pros	Cons
Line Charts	Effective for displaying trends over time. Easy to interpret patterns and relationships	Possibility of becoming cluttered with too many lines. Not suitable large multi-dimensional datasets
Bar Charts	Suitable for comparing categories. Can display multiple variables side-by-side. Easy to understand and interpret.	Only suitable categorical or discrete data. Possibility of becoming cluttered with too many bars
Scatter Plots	Suitable for displaying relationships between variables. Reveals patterns, clusters, and outliers. Add additional dimensions with colour or size.	Overlapping points can obscure data. Possibility of becoming cluttered with large datasets.
Histograms	Suitable for visualising the distribution of a continuous variable. Displays skewness, modality, and outliers. Simple and easy to understand and interpret.	Can obscure details as it requires data binning. Difficult to compare multiple distributions
Pie Charts	Simple and intuitive for displaying proportions. Suitable for a small number of categories. Easy to interpret.	Comparing values accurately is difficult. Restricted to a single dimension. Not suitable for many categories
Heatmaps	Effective and suitable for visualizing large datasets. Displays patterns and correlations between variables. Multiple dimensions are displayed using colour.	Interpretation can be difficult for new users. Sometimes colour scales can be misleading. Little quantitative details for individual data points.
Box Plots	Distribution display of a continuous variable. Displays median, quartiles, and outliers. Suitable for multiple distributions comparison.	Restricted to a single dimension. Misleading when incorrectly interpreted.
Geographic Maps	Suitable for displaying spatial or geographic data. Displays patterns and trends across regions. Visualisation of location-based data	Limited to spatial or geographic data. Complex to create and interpret

The choice of visualisation method depends on the data type, the research questions or problem, and the audience. A combination of different visualization techniques is necessary to effectively communicate insights from the data.

Adopted Data Visualisation Methods:

In this project, the following data visualisation methods were adopted: line charts, bar charts, histograms, and scatter plots to visualize various aspects of the human activity recognition dataset.

Utilised Features of Each Data Visualisation Method:

1. Line Charts: Used to visualize the acceleration and angular velocity time histories for different activities, allowing observation of patterns and changes over time.
2. Bar Charts: Used to visualize the distribution of activities and participants within the dataset, enabling easy comparison and identification of any potential imbalances.
3. Histograms: Utilized to explore the distribution of individual features, such as acceleration and angular velocity components, as well as derived features like correlation and cosine distances. By plotting accelerometer features such as XPEAK, YPEAK, and ZPEAK, the distribution pattern and any outliers are easily observed.
4. Heatmaps: Heatmaps were used to visualise the correlation matrix of accelerometer features. By visualising the correlation matrix of accelerometer features as a heatmap, highly correlated or anticorrelated features were identified. This is useful in feature selection and dimensionality reduction steps.
5. Scatter Plots: Implemented to investigate potential relationships or clusters between different features or variables within the dataset. scatter plots were utilised to explore the relationship between accelerometer features such as XAVG, YAVG, and ZAVG.
6. 3D Scatter Plots: 3D scatter plots extends scatter plots to three dimensions. Effective for simultaneous visualisation of relationships between three variables. The relationship between three accelerometer features, such as XAVG, YAVG, and ZAVG was visualised using the 3D scatter plots. The plotting of accelerometer features such as XAVG, YAVG, and ZAVG in a 3D scatter plot gave insight into the spatial distribution of the data.

Implementation:

The implementation of data visualization and analysis for the human activity recognition dataset was carried out using Python and various libraries, including pandas, numpy, matplotlib, and seaborn. The code provided in the document covers the following steps:

Data Loading and Preprocessing:

- The dataset (both phone and watch accelerometer datasets) was loaded from text files into pandas DataFrames, with appropriate column names and data types assigned.
- Necessary data cleaning and formatting operations were performed, such as handling missing values and converting data types.

Data Visualisation:

- Line charts were created using the plot function from pandas and matplotlib to visualize the acceleration and angular velocity time histories for different activities. These plots were generated by selecting the relevant features ('x', 'y', 'z') and plotting them against the 'duration' column, which represented the elapsed time. Line charts are useful for identifying patterns and trends over time, enabling the analysis of signal characteristics for different activities.
- Bar charts were generated using pandas and matplotlib to display the count of activities and participants within the dataset. The value_counts function was used to obtain the counts, which were then plotted using bar charts. These visualizations allowed for easy comparison of activity and participant distributions, revealing potential imbalances or biases in the data.
- Histograms were plotted using pandas and matplotlib to explore the distribution of individual features, such as acceleration components and derived features. The hist function was used to generate histograms, which divided the data into bins and displayed the count or density of data points in each bin. Histograms are useful for understanding the shape and spread of feature distributions, identifying potential outliers or skewness.
- Scatter plots were implemented using pandas and seaborn to investigate potential relationships or clusters between different features. The scatterplot function from seaborn and px.scatter_3d function from plotly were used to create these visualizations, allowing for the exploration of correlations or patterns among the selected features.
- Additional visualizations, such as box plots and distribution plots, were created using seaborn's displot and boxplot functions. These plots provided insights into the distribution of features, revealing outliers, and facilitating comparisons across different variables or subgroups.

Data Preparation for Modeling:

- The dataset was split into training and testing sets using the train_test_split function from scikit-learn.
- Appropriate data transformations and feature engineering techniques were applied to prepare the data for modeling.

Modeling and Evaluation

- Various machine learning models were employed for the task of human activity recognition, including K-Nearest Neighbors (KNN), Random Forest, Logistic Regression, and Decision Tree.
- These models were trained on the prepared dataset using scikit-learn's implementation.
- Evaluation metrics such as accuracy, precision, recall, and F1-score were calculated to assess the performance of the models.

Interpretation of Data Visualisation

Through the implemented data visualizations, we can derive several insights and interpretations from the human activity recognition dataset:

1. Time History Plots:

The acceleration and angular velocity time histories exhibit distinct patterns and characteristics for different activities, suggesting that these features can be useful for activity recognition.

Activities involving more movement and dynamic motion, such as walking, jogging, and playing sports, tend to have higher amplitudes and more variability in the acceleration and angular velocity signals. The observation in the figures below is similar to the one observed for watch data.

Figure 1: Acceleration Line Chart of Walking Activity

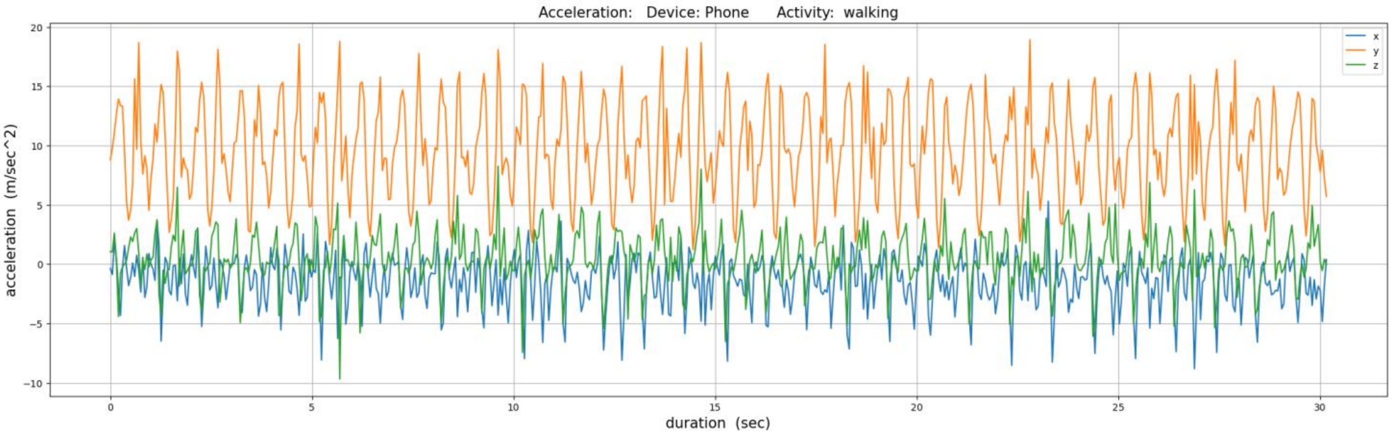
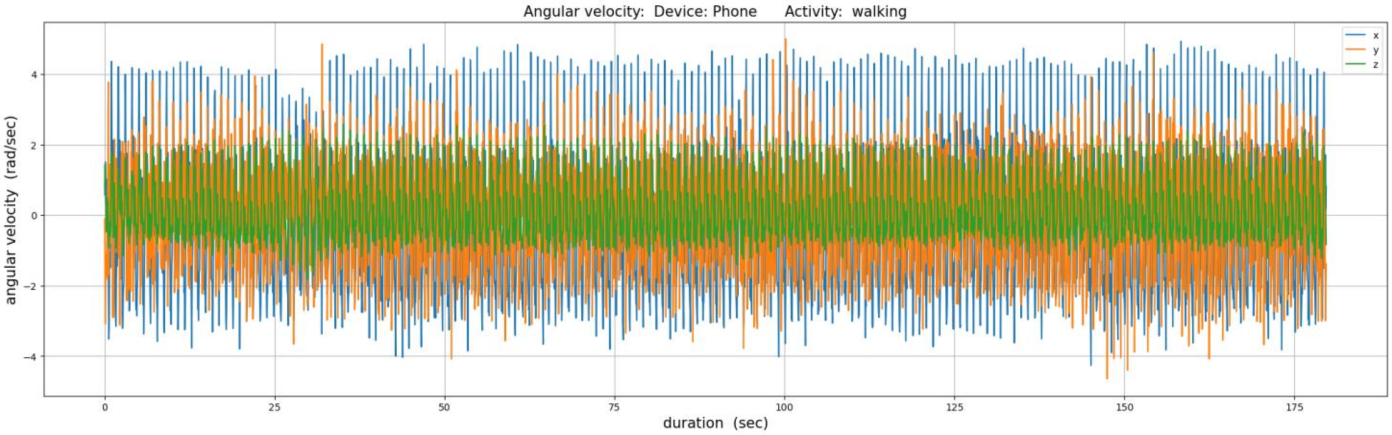


Figure 2: Angular Velocity Line Chart of Walking Activity



Stationary or less dynamic activities, like sitting or standing, generally have lower amplitudes and less variability in the signals. The observation in the figures below is similar to the one observed for watch data.

Figure 3: Acceleration Line Chart of Standing Activity

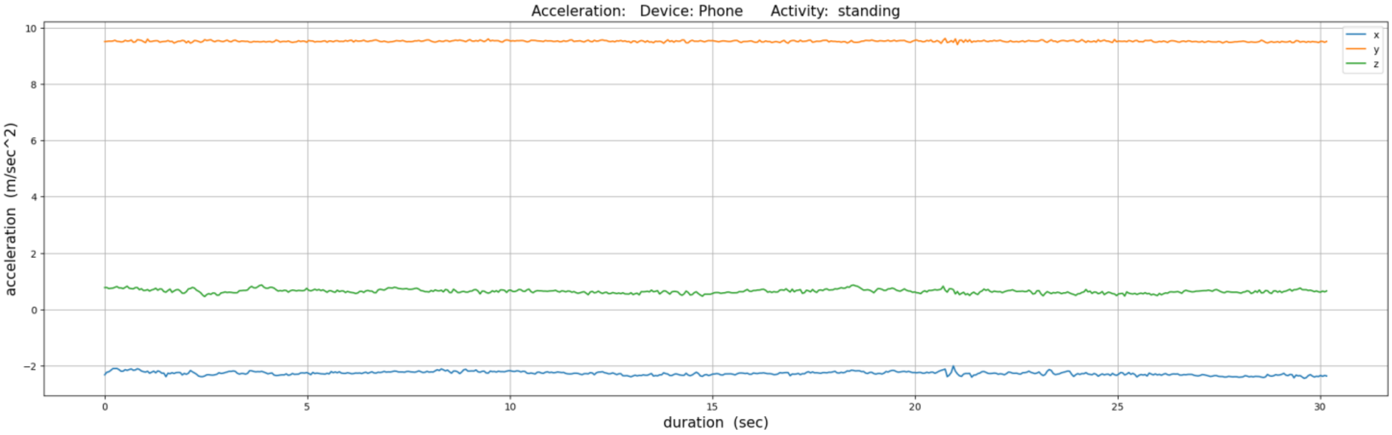
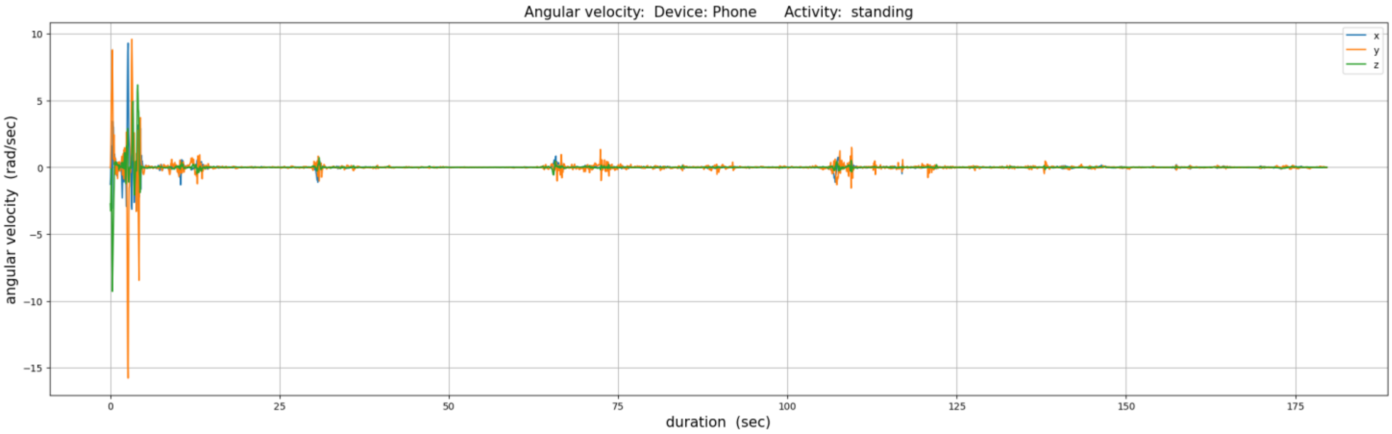


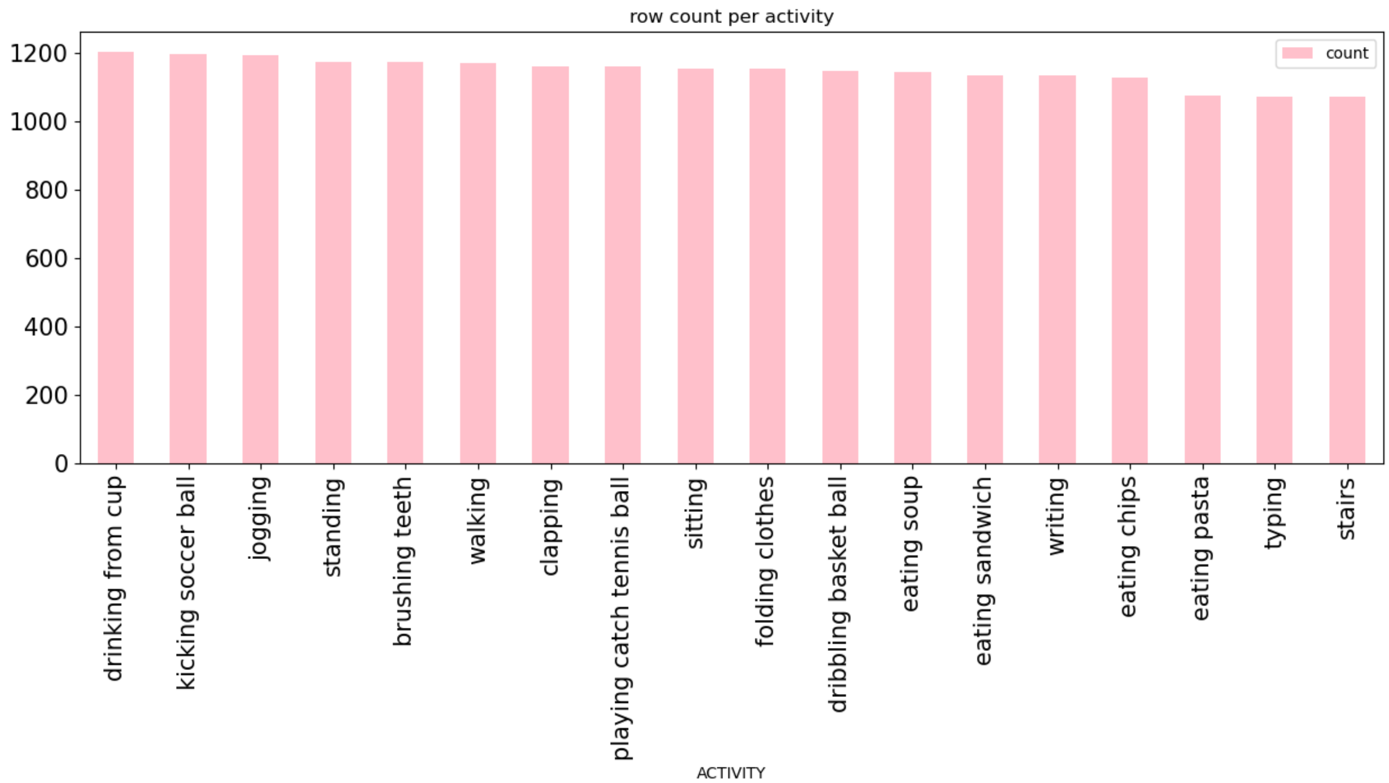
Figure 4: Angular Velocity Line Chart of Standing Activity



2. Activity and Participant Distribution:

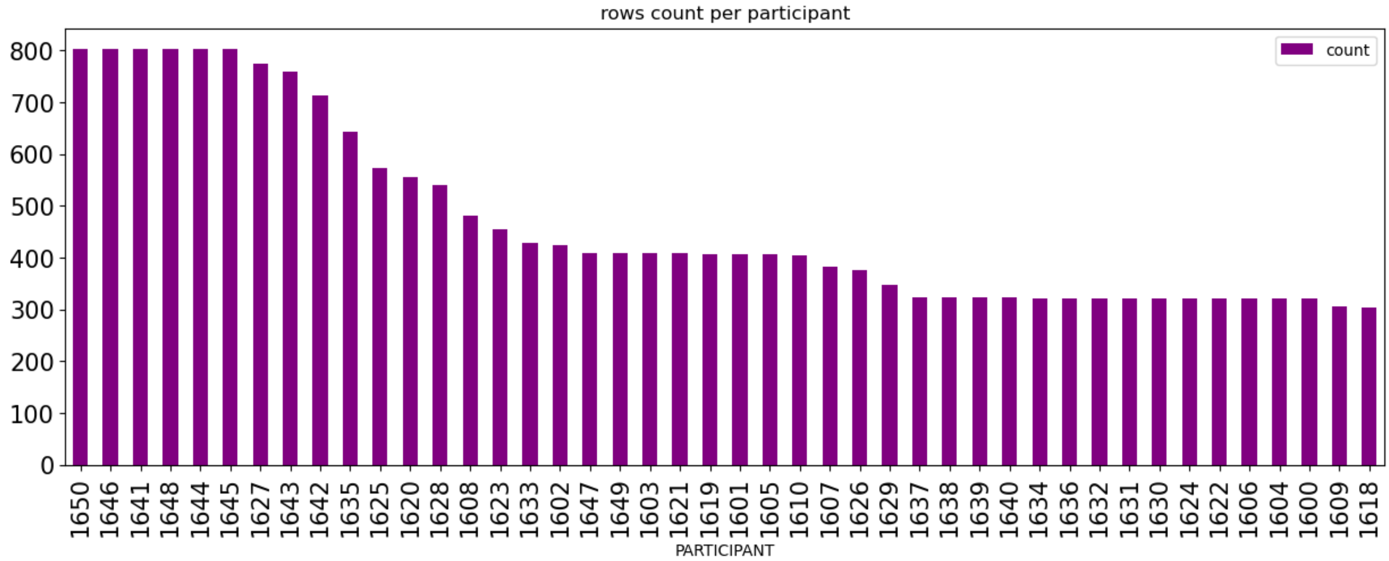
The bar charts revealed an imbalance in the number of instances for different activities, with some activities being more represented than others. For example, activities like "drinking from cup" and "kicking soccer ball" have a substantially higher number of instances compared to activities like "typing" and "stairs”. The imbalance in the activity distribution could introduce biases in the dataset, potentially impacting the performance of machine learning models trained on this data. Activities with fewer instances might be underrepresented, leading to poorer performance in recognizing those activities accurately.

Figure 5: Bar Chart Showing Count Distribution Various Activities



This pattern below indicates that the data collection or participation was skewed, with a small number of participants contributing a significantly larger portion of the data compared to the majority of participants. It suggest varying levels of engagement or activity among the participant pool.

Figure 6: Bar Chart Showing Count Distribution of Participants



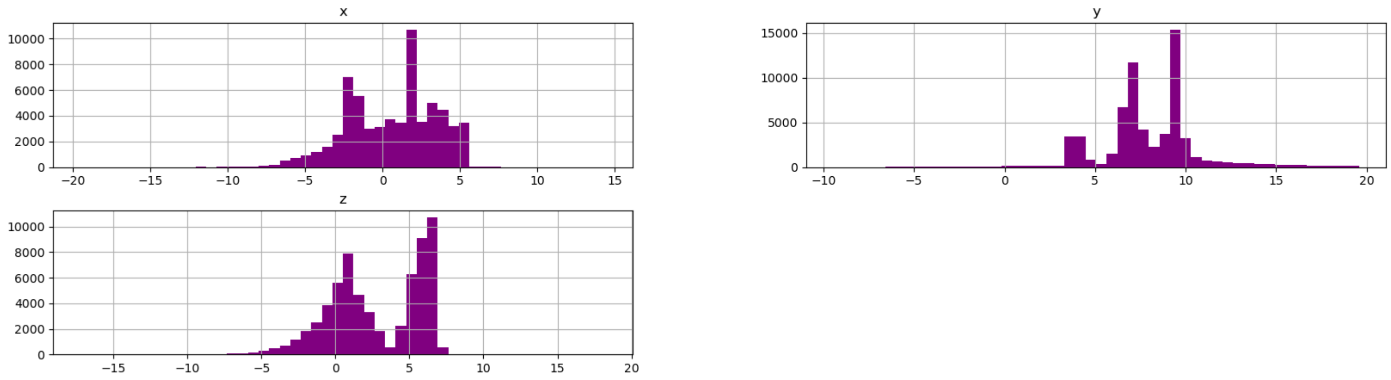
3. Feature Distributions:

The histograms of individual features showed varying distributions, some exhibiting normal or Gaussian-like shapes, while others displayed skewed or multi-modal distributions.

Features like correlation and cosine distances between axes exhibited distinct distributions, potentially indicating their usefulness in activity recognition.

Outliers may indicate anomalies in the data that require further investigation. The histogram of a skewed distribution of X, Y, and Z values suggests a bias or systematic error in the data collection process.

Figure 7: Distribution of the X, Y, & Z Accelerometer Feature

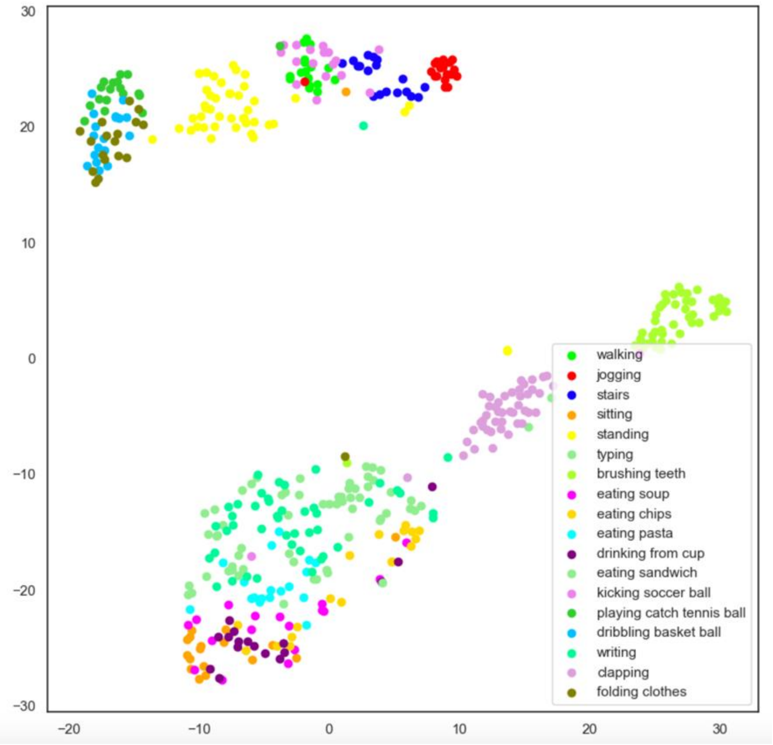


This histogram visualises the distribution of the x, y, and z accelerometer feature. The x-axis represents the values and the y-axis represents the frequency of occurrence.

4. Feature Relationships:

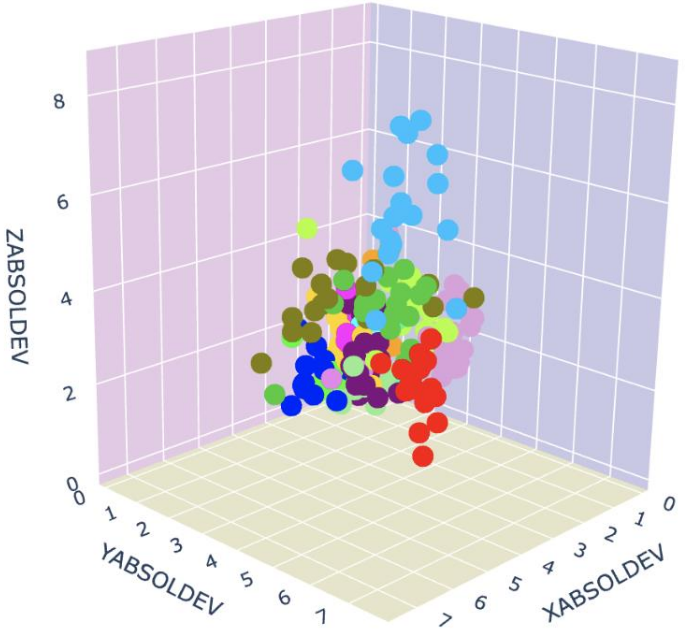
Scatter plots revealed potential relationships or clusters between different features, suggesting that certain combinations of features could be informative for distinguishing between activities. In the scatter plot of XAVG vs. YAVG, distinct clusters suggest different types of physical activities associated with specific ranges of accelerometer readings.

Figure 8: Relationship between XAVG and YAVG Accelerometer Features



This scatter plot visualises the relationship between XAVG and YAVG accelerometer features. Each point represents a data sample, and the colour represents the activity label. This allows the visualisation of the underlying structure of the data points.

Figure 9: Relationship between XAVG, YAVG, and ZAVG Accelerometer Features

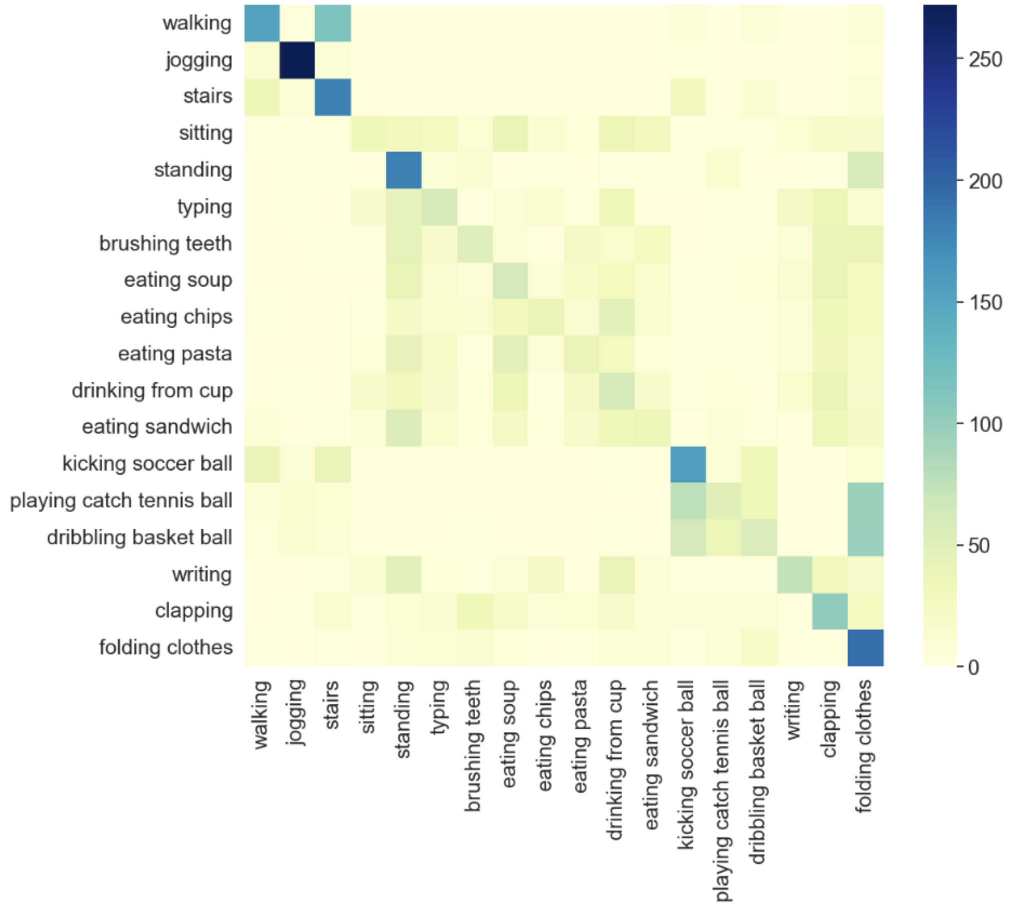


The above 3D scatter plot visualises the relationship between XAVG, YAVG, and ZAVG accelerometer features. Each point represents a data sample, and the colour represents the activity label. This visualisation helps in understanding how the accelerometer data varies across different activities and can be useful for further analysis or classification tasks,

In heatmaps, high correlation coefficients between features represent a strong linear relationship, low correlation coefficients represent weak relationships, and negative correlation coefficients represent inverse relationships. Highly correlated features which indicate redundancy or multicollinearity in the dataset were identified by analysing the heatmap. Feature selection techniques handle these issues and improve model performance.

The below heatmap visualises the correlation matrix of accelerometer features. The colour represents the strength of the correlation between different features, with darker colours indicating higher correlation.

Figure 10: Correlation Matrix of Accelerometer Features



Data Mining Solution

Based on the insights gained from the data visualization and the patterns observed in the dataset, a suitable data mining solution for human activity recognition could involve the following steps:

Feature Selection and Preprocessing:

Identify and select the most informative features for activity recognition based on the observed patterns and distributions. Features like acceleration, angular velocity, correlation, and cosine distances between axes could be particularly useful predictors. Feature selection and preprocessing enhance model performance and reduce computational complexity. Relevant features for model training are identified using feature selection techniques such as principal component analysis (PCA) and correlation analysis. Normalisation or scaling preprocessing steps ensure that all features contribute equally to the model.

Imbalanced Data Handling:

Techniques like oversampling, undersampling, or generating synthetic data using methods like SMOTE (Synthetic Minority Over-sampling Technique) will be useful to address the imbalance in the number of instances for different activities.

Model Building and Training:

Explore machine learning algorithms suitable for multiclass classification problems, such as Support Vector Machines (SVM), Gradient Boosting, or Neural Networks. Train the selected models on the prepared dataset, using appropriate evaluation metrics like accuracy, precision, recall, and F1-score. Logistic regression, decision trees, k-nearest neighbours (KNN), and random forests models were explored to classify human activities based on accelerometer data with low accuracy results. Every model has its strengths and weaknesses. Using algorithms suitable for multiclass classification or reducing the number of activities used to train the models two or three could give a better performance.

Model Evaluation and Selection:

Evaluate the performance of different models using cross-validation techniques or a hold-out test set. K-fold cross-validation or stratified shuffle split can be employed to assess model generalisation performance and detect overfitting. Select the model with the best performance based on the evaluation metrics and the trade-offs between model complexity and accuracy.

Model Deployment and Monitoring:

Deploy the selected model for real-time or batch prediction of human activities based on the input sensor data. Implement monitoring and feedback mechanisms to detect any potential model drift or performance degradation over time, and retrain the model as necessary.

Summary

This report explores data visualization techniques and their application to a human activity recognition dataset using Python. Various visualization methods are discussed, including line charts, bar charts, histograms, scatter plots, and others. The advantages and use cases of each method are highlighted.

For the given dataset, line charts were used to visualize time-series signals like acceleration and angular velocity across different activities. Bar charts helped analyze the distribution of activity types and participants. Histograms provided insights into the spread and shape of individual feature distributions. Scatter plots enabled the investigation of relationships and potential clusters among features.

The data visualization process involved loading and preprocessing the dataset, creating visualizations with libraries like Matplotlib and Seaborn, and then interpreting the patterns and trends observed. Time history plots revealed distinct characteristics for different activities, while distribution plots highlighted imbalances and potential outliers.

Based on these insights, the report outlines a data mining solution for activity recognition. This includes feature selection, handling data imbalances, building and evaluating machine learning models, and deploying the best-performing model with monitoring mechanisms. By following this data mining approach, informed by the insights gained from data visualization, an effective and robust solution for human activity recognition can be developed, enabling a wide range of applications in areas such as healthcare, fitness tracking, and smart environments (Vadim, 2018).

References

FRANCONERI, S.L., PADILLA, L.M., SHAH, P., ZACKS, J.M. AND HULLMAN, J., 2021. The Science of Visual Data Communication: What Works. Psychological Science in the Public Interest [online]. 22(3). pp.110-161. Available from: https://doi.org/10.1177/15291006211051956 [Accessed 20 April 2024].

GUPTA, P. AND BAGCHI, A., 2024. Data Manipulation with Pandas. In Essentials of Python for Artificial Intelligence and Machine Learning [online]. pp. 197-235. Cham: Springer Nature Switzerland. Available from: https://link.springer.com/chapter/10.1007/978-3-031-43725-0_6 [Accessed 20 April 2024].

KULKARNI, A., SHAH, H., D'MELLO, L. AND SHAH, K., 2023. Flowchart Generation and Mind Map Creation using Extracted Summarized Text. In 2023 International Conference on Recent Advances in Science and Engineering Technology (ICRASET) (pp. 1-6). IEEE [online]. Available from: https://ieeexplore.ieee.org/abstract/document/10420315 [Accessed 22 April 2024]

LAVANYA, A., GAURAV, L., SINDHUJA, S., SEAM, H., JOYDEEP, M., UPPALAPATI, V., ALI, W. AND SD, V.S., 2023. Assessing the Performance of Python Data Visualization Libraries: A Review. International Journal of Computer Engineering in Research Trends [online]. 10(1). pp.29-39. Available from: https://doi.org/10.22362/ijcert/2023/v10/i01/v10i0102 [Accessed 21 April 2024].

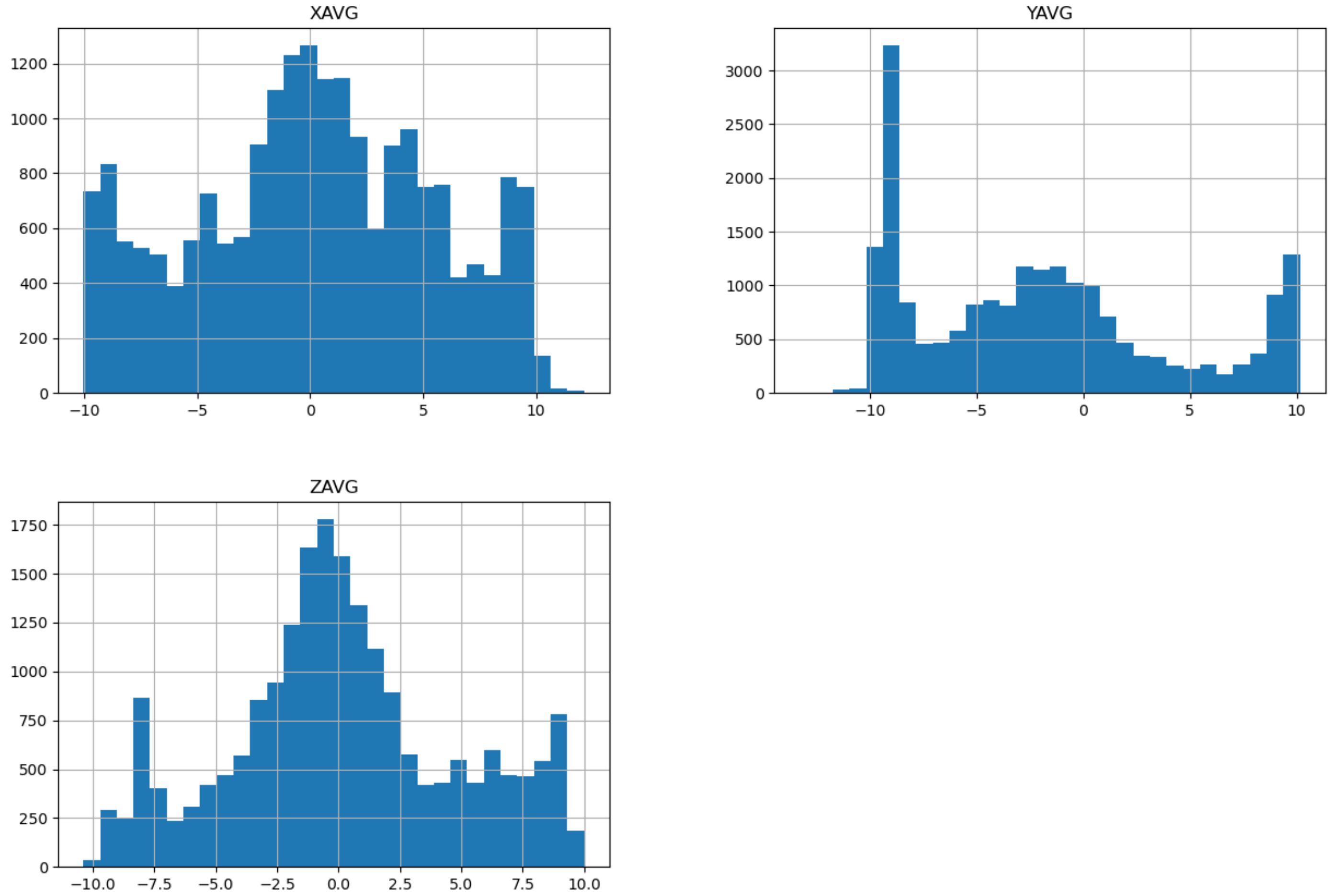
LO, L.Y.H., GUPTA, A., SHIGYO, K., WU, A., BERTINI, E. AND QU, H., 2022. Misinformed By Visualization: What Do We Learn from Misinformative Visualizations? In Computer Graphics Forum [online]. 41(3). pp. 515-525. Available from: https://doi.org/10.1111/cgf.14559 [Accessed 18 April 2024].

QU, Z., LAU, C.W., NGUYEN, Q.V., ZHOU, Y. AND CATCHPOOLE, D.R., 2019. Visual Analytics of Genomic and Cancer Data: A Systematic Review. Cancer Informatics [online]. 18. p.1176935119835546. Available from: https://doi.org/10.1177/1176935119835546 [Accessed 25 April 2024].

VADIM, K., 2018. Overview of Different Approaches to Solving Problems of Data Mining. Procedia Computer Science [online]. 123. pp.234-239. Available from: https://doi.org/10.1016/j.procs.2018.01.036 [Accessed 28 April 2024]

Appendices:

Figure 11: Histogram of Average X, Y, & Z Values of Accelerometer Reading



The histograms in Figure 11 above provide insights into the distribution of average values for the accelerometer readings along the x, y, and z coordinates.

Figure 2: Model Evaluation Metrics Scores

Model	Accuracy	Precision	Recall	F-1 Score
Decision Tree	0.35	0.34	0.35	0.34
Random Forest	0.46	0.48	0.46	0.44
Logistic Regression	0.36	0.34	0.36	0.35
KNN	0.36	0.34	0.36	0.35

The models performed poorly. Various factors could have contributed to this such as how well the models used suit the data (multiclass) and how well the datasets were preprocessed. Using algorithms suitable for multiclass classification or reducing the number of activities used to train the models to two or three could give a better performance.