Classifying Physical Activities Using KNN and Logistic Regression on Wearable Sensor Data

Brian Johnson

1 Introduction

1.1 Problem Statement

The primary objective of this project is to accurately classify physical activities using data collected from wearable sensors. Wearable technology, such as smartphones and fitness trackers, is equipped with sensors like accelerometers and gyroscopes that capture detailed motion data. This data can be used to identify and distinguish between various physical activities, such as walking, running, sitting, standing, and more. By leveraging machine learning techniques, this project aims to develop a model that can automatically classify these activities based on the sensor readings. Accurate classification of physical activities has numerous applications, including health monitoring, which provides insights into a person's daily activity levels crucial for health assessments and fitness tracking. It also assists in rehabilitation by monitoring patients' progress in physical therapy, ensuring they perform exercises correctly. In sports analytics, it offers detailed analysis of athletes' movements to improve performance and reduce the risk of injury. Additionally, it enhances the functionality of smart environments by adapting to the occupants' activities in smart homes and workplaces. My code can be found here on GitHub, where the implementation of the data and ML algorithms can be seen: https://github.com/BrianJohnsonJr/MachineLearningFinalProject.git

1.2 Motivation and Challenges

Understanding and classifying physical activities is crucial for developing personalized fitness programs and enhancing health monitoring systems. Wearable technology provides a continuous stream of data, but the challenge lies in effectively processing and interpreting this data to provide meaningful insights. Accurate activity recognition can lead to improved health outcomes and more engaging fitness applications.

1.3 Concise Summary of My Approach

My approach involves using two machine learning algorithms: K-Nearest Neighbors (KNN) and Logistic Regression. KNN is chosen for its simplicity and effectiveness in capturing non-linear relationships, while Logistic Regression offers a probabilistic approach that is both efficient and interpretable. By comparing these models, the aim is to identify the most effective method for activity classification, providing a robust solution to the problem.

2 Data - Human Activity Recognition Using Smartphones

The Human Activity Recognition Using Smartphones dataset is a comprehensive collection of data designed to facilitate the classification of physical activities based on sensor readings. This dataset was collected from 30 participants who performed six different activities while wearing a smartphone on their waist. The smartphone's embedded accelerometer and gyroscope captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz.

2.1 Data - Key Features

Activities: The dataset includes six activities: walking, walking upstairs, walking downstairs, sitting, standing, and laying.

Participants: Data was collected from 30 individuals, providing a diverse range of movement patterns.

Features: The dataset contains 561 features, which are derived from time and frequency domain variables. These features include statistical measures such as means, standard deviations, and signal magnitudes.

Data Structure: The dataset contains a total of 10,299 samples.

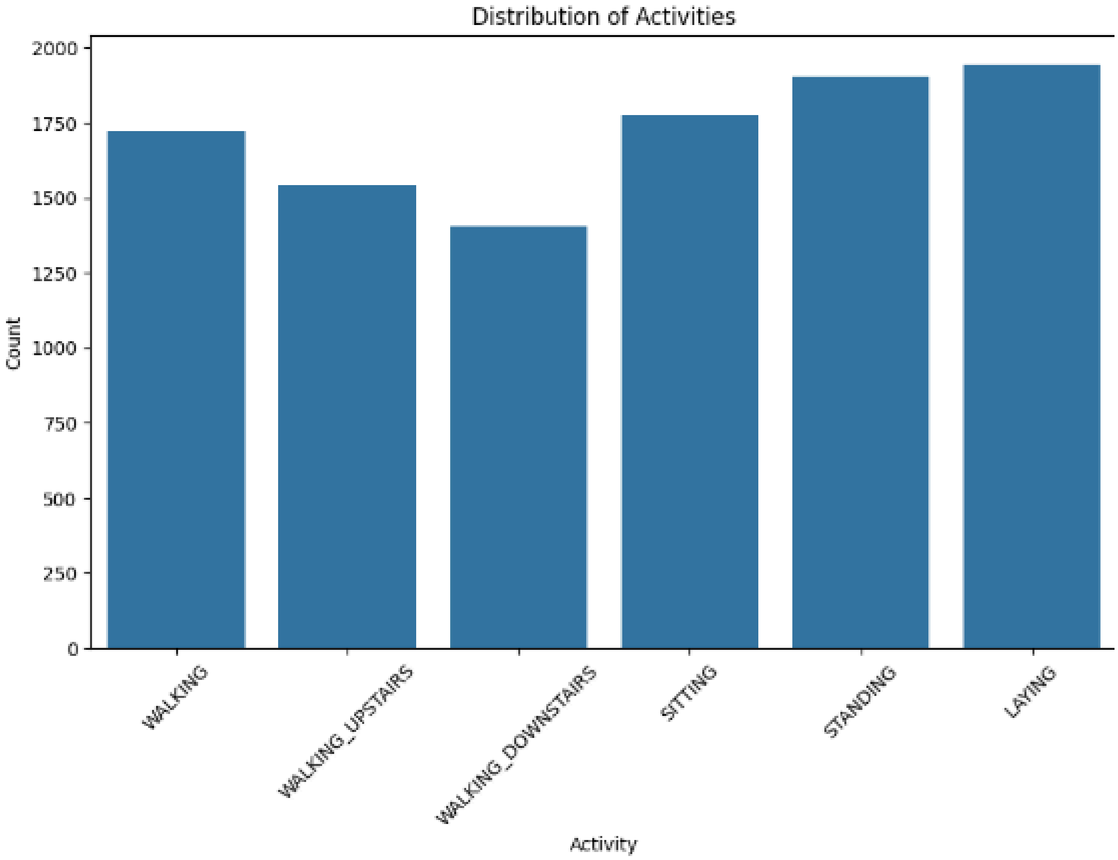
2.2 Data - Purpose and Use

The primary goal of this dataset is to enable the development and evaluation of machine learning models for activity recognition. By leveraging the rich set of features, researchers can explore various algorithms to accurately classify physical activities, contributing to advancements in wearable technology and personalized health monitoring.

This dataset provides a robust foundation for analyzing the effectiveness of different machine learning techniques, such as K-Nearest Neighbors and Logistic Regression, in the context of human activity recognition.

2.3 Data - Visualization

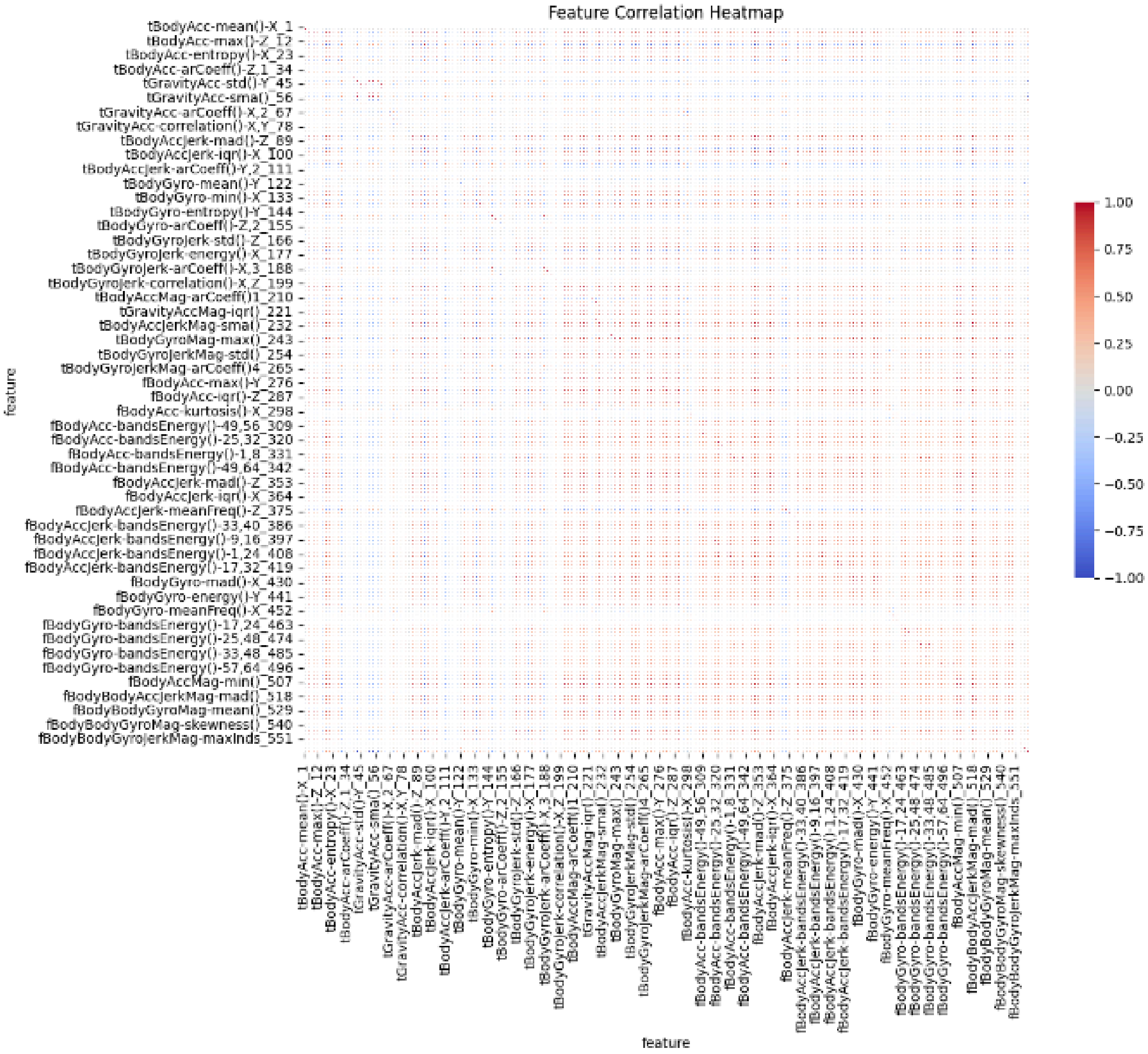
Distribution of Activities



Observation: The bar plot shows the number of samples for each activity: Walking, Walking Upstairs, Walking Downstairs, Sitting, Standing, and Laying. The counts are relatively balanced, with "Standing" and "Laying" having slightly more samples.

Analysis: A balanced dataset is crucial for training machine learning models, as it prevents bias towards any particular class. This balance ensures that the model can learn equally well across all activities, leading to more reliable predictions.

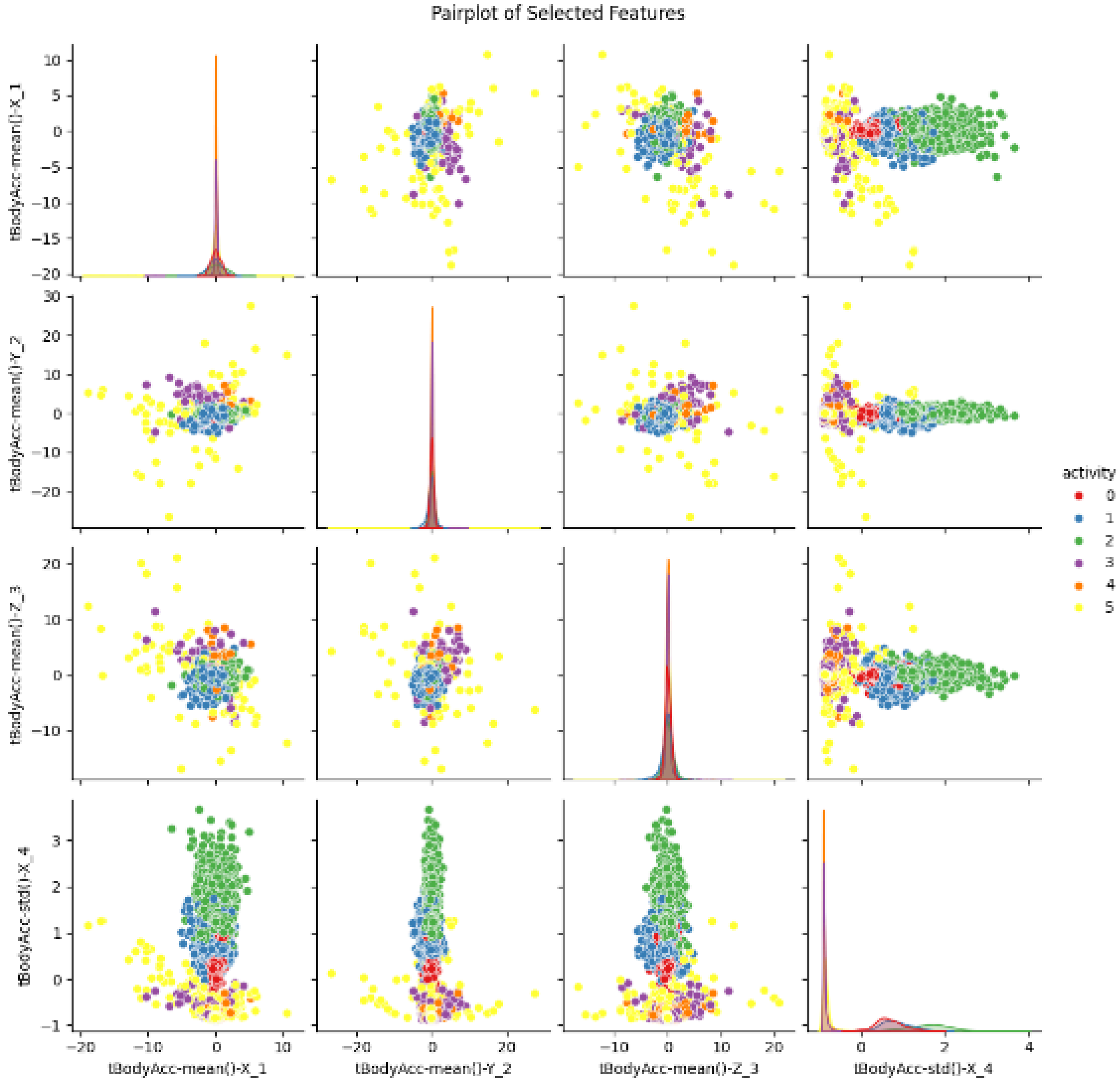
Feature Correlation Heatmap



Observation: The heatmap displays the correlation coefficients between features, ranging from -1 to 1. Most features have low correlation, but some show moderate to high correlation, indicating potential redundancy.

Analysis: Highly correlated features may not provide additional information and can be candidates for removal or dimensionality reduction. Focusing on uncorrelated features can improve model efficiency and performance by reducing noise and redundancy.

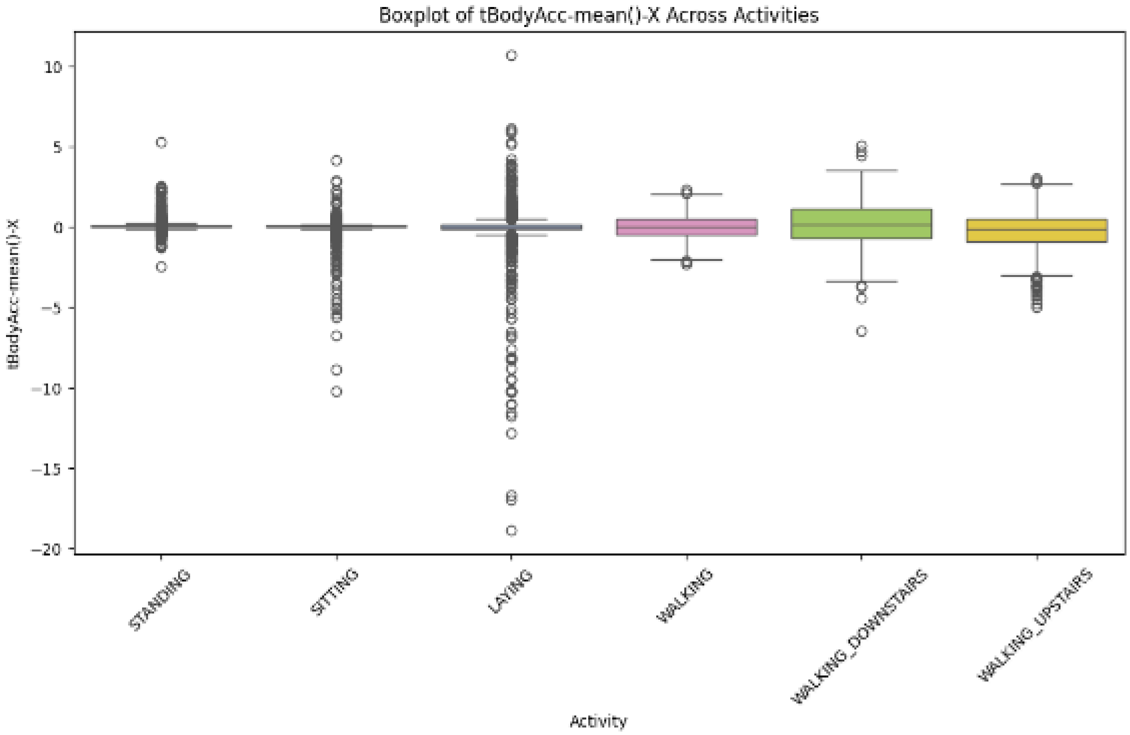
Pairplot of Selected Features



Observation: The pairplot illustrates relationships between selected features (tBodyAcc-mean()-X, tBodyAcc-mean()-Y, tBodyAcc-mean()-Z, tBodyAcc-std()-X) and activities. Distinct clusters are visible for some activities, while others overlap.

Analysis: Features like tBodyAcc-mean()-X show clear separation for activities such as "Walking" and "Laying," indicating their discriminative power. Overlapping clusters, particularly for "Walking Upstairs" and "Walking Downstairs," suggest these activities are more challenging to distinguish and may require additional features or complex models.

Boxplot of a Specific Feature



Observation: The boxplot for tBodyAcc-mean()-X across activities shows distinct medians and interquartile ranges. "Walking" and "Laying" have non-overlapping ranges, while others like "Sitting" and "Standing" overlap.

Analysis: The clear separation for some activities suggests that tBodyAcc-mean()-X is a strong feature for distinguishing them. However, overlapping ranges for other activities indicate that this feature alone may not suffice, necessitating the use of additional features or advanced classification techniques to improve accuracy.

2.4 Overall Insights

Balanced Data: The data supports robust model training and evaluation.

Feature Selection: There needs to be focus on uncorrelated and discriminative features to enhance model performance.

Model Challenges: Addressing overlapping feature distributions with additional data or more sophisticated models.

3 Method (KNN and Logistic Regression)

3.1 KNN

K-Nearest Neighbors is a simple, non-parametric classification algorithm. It classifies a data point based on the majority class of its k nearest neighbors in the feature space. The distance between data points is typically measured using Euclidean distance, although other distance metrics can be used.

Application in the Project

Initialization: The KNN model is initialized with n_neighbors=5, meaning it considers the 5 closest data points to determine the class of a new sample.

Training: KNN is a lazy learner, meaning it doesn't explicitly train a model. Instead, it stores the training data and makes predictions based on the stored data.

Prediction: For each validation sample, KNN calculates the distance to all training samples, identifies the 5 nearest neighbors, and assigns the most common class among these neighbors.

Evaluation: The model's performance is evaluated using accuracy and a classification report, which includes precision, recall, and F1-score for each activity class.

3.2 Logistic Regression

Logistic Regression is a linear model used for binary and multiclass classification. It models the probability that a given input belongs to a particular class using the logistic function. The model finds the best-fitting hyperplane that separates the classes in the feature space.

Application in the Project

Initialization: The Logistic Regression model is initialized with max_iter=1000 to ensure convergence during training.

Training: The model learns the weights for each feature by maximizing the likelihood of the observed data. It uses these weights to create a decision boundary that separates the classes.

Prediction: For each validation sample, the model calculates the probability of belonging to each class and assigns the class with the highest probability.

Evaluation: Similar to KNN, the model's performance is assessed using accuracy and a classification report, providing insights into how well the model distinguishes between different activities.

3.3 Summary

KNN is effective for capturing non-linear relationships and is straightforward to implement. It works well with smaller datasets but can be computationally expensive with larger ones.

Logistic Regression offers a probabilistic approach that is efficient and interpretable. It assumes a linear relationship between features and the log-odds of the classes.

3.4 Overall Contribution

Model Comparison: By using both KNN and Logistic Regression, their performance can be compared and determine which is more suitable for my specific dataset and problem.

Robust Classification: Together, these models help ensure robust classification of physical activities, contributing to the development of accurate and reliable fitness applications.

Insightful Analysis: The combination of models provides a comprehensive analysis of the dataset, highlighting strengths and areas for improvement.

4 Results

4.1 Experimental Setup

Data Preparation:

The dataset was split into training and validation sets using an 80-20 split. Features were normalized using StandardScaler to ensure consistent scaling across all features.

Model Training:

K-Nearest Neighbors (KNN): Initialized with 5 neighbors. This choice balances complexity and performance, allowing the model to capture local patterns in the data.

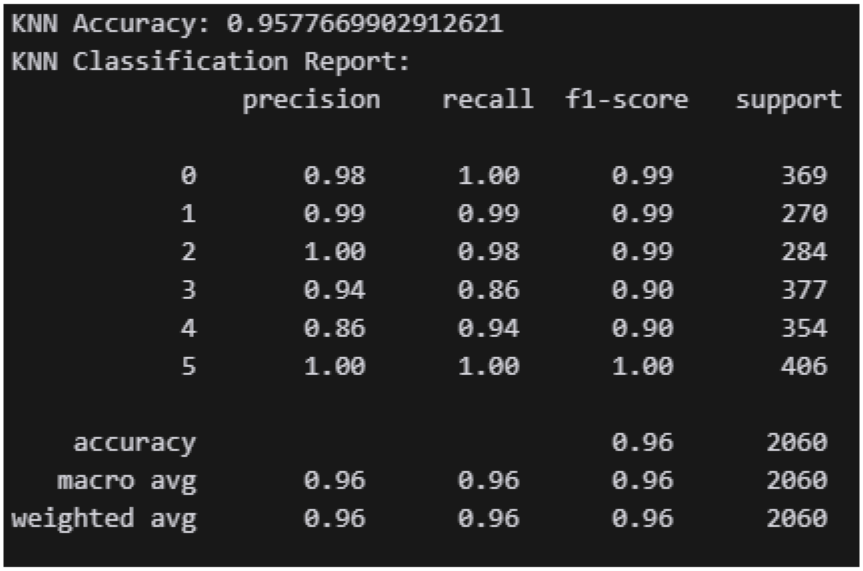
Logistic Regression: Configured with a maximum of 1000 iterations to ensure convergence. This model provides a linear decision boundary for classification.

Evaluation Metrics:

Both models were evaluated using accuracy, precision, recall, and F1-score. These metrics provide a comprehensive view of model performance across different activity classes.

4.2 Test Results and Observations

KNN Model Evaluation:



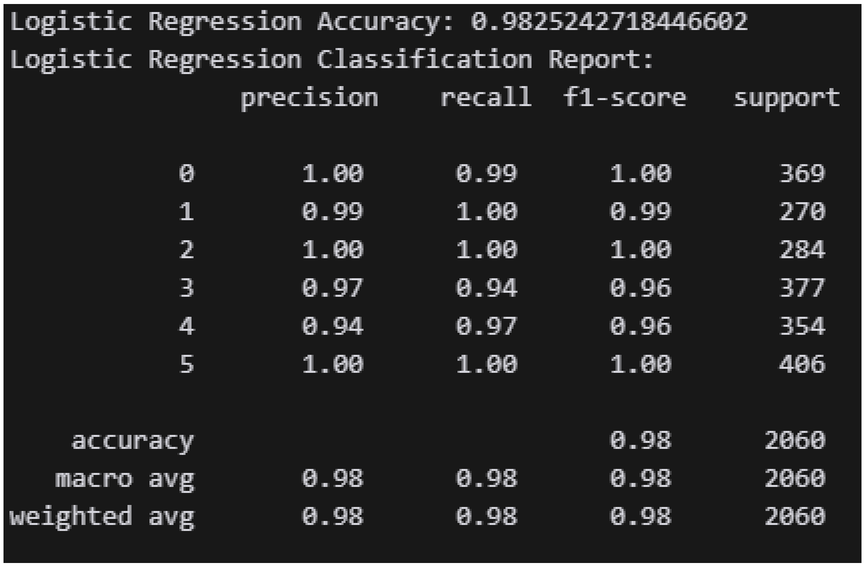
```
KNN Accuracy: 0.9577669902912621
KNN Classification Report:
              precision    recall  f1-score   support

           0       0.98      1.00      0.99       369
           1       0.99      0.99      0.99       270
           2       1.00      0.98      0.99       284
           3       0.94      0.86      0.90       377
           4       0.86      0.94      0.90       354
           5       1.00      1.00      1.00       406

    accuracy                           0.96      2060
   macro avg       0.96      0.96      0.96      2060
weighted avg       0.96      0.96      0.96      2060
```

Accuracy: Achieved an accuracy of 95.8%.

Precision and Recall: High precision and recall for most classes, indicating effective classification. Class 4 ("Sitting") had slightly lower precision and recall, suggesting some misclassification.

F1-Score: Consistently high across classes, reflecting a good balance between precision and recall.

Logistic Regression Model



```
Logistic Regression Accuracy: 0.9825242718446602
Logistic Regression Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.99      1.00       369
           1       0.99      1.00      0.99       270
           2       1.00      1.00      1.00       284
           3       0.97      0.94      0.96       377
           4       0.94      0.97      0.96       354
           5       1.00      1.00      1.00       406

    accuracy                           0.98      2060
   macro avg       0.98      0.98      0.98      2060
weighted avg       0.98      0.98      0.98      2060
```

Accuracy: Achieved an accuracy of 98.3%.

Precision and Recall: Excellent precision and recall across all classes, indicating strong performance. The model effectively distinguishes between activities, even those with overlapping features.

F1-Score: High F1-scores across the board, demonstrating the model's robustness in handling class imbalances.

4.3 Analysis and Insights

1. KNN Performance:

KNN's high accuracy and balanced precision-recall indicate its strength in capturing non-linear relationships. The slight drop in performance for class 4 suggests that additional features or a different distance metric might improve results.

2. Logistic Regression Performance:

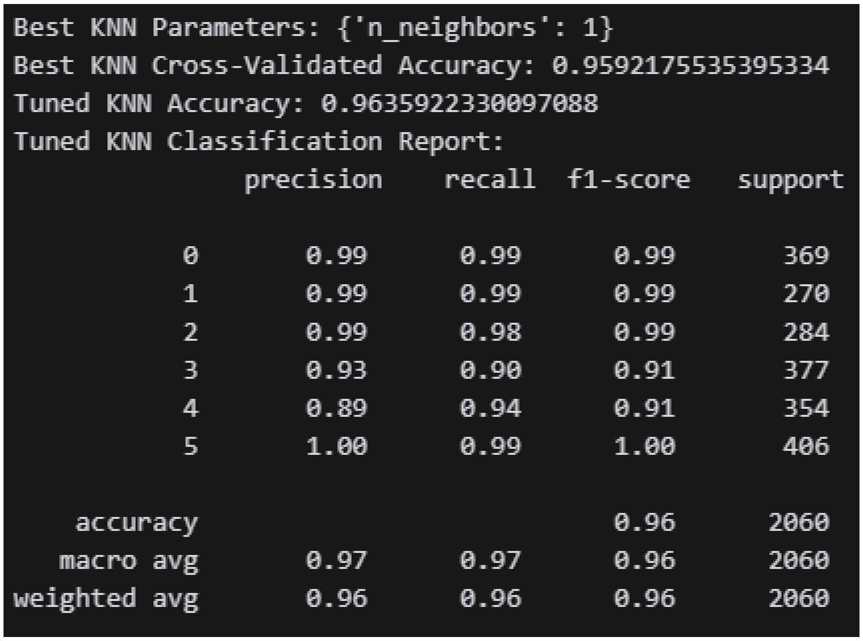
Logistic Regression outperformed KNN, achieving higher accuracy and F1-scores. Its ability to handle linear separability effectively distinguishes between activities, making it a strong candidate for this dataset.

3. Comparison:

While KNN is effective for non-linear patterns, Logistic Regression's superior performance suggests it is better suited for this dataset. Its efficiency and interpretability make it ideal for applications requiring real-time predictions.

4. Further Experiments:

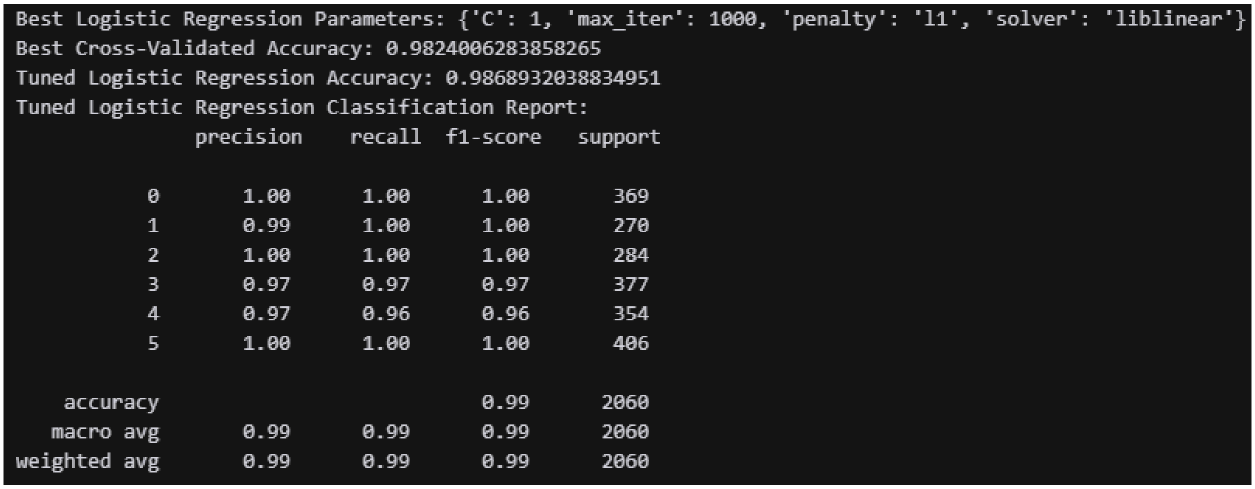
Hyperparameter Tuning - KNN



```
Best KNN Parameters: {'n_neighbors': 1}
Best KNN Cross-Validated Accuracy: 0.9592175535395334
Tuned KNN Accuracy: 0.9635922330097088
Tuned KNN Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99       369
           1       0.99      0.99      0.99       270
           2       0.99      0.98      0.99       284
           3       0.93      0.90      0.91       377
           4       0.89      0.94      0.91       354
           5       1.00      0.99      1.00       406

    accuracy                           0.96      2060
   macro avg       0.97      0.97      0.96      2060
weighted avg       0.96      0.96      0.96      2060
```

A range of different neighbor values were tested, leading to a slight improvement to 96% accuracy for KNN. The best parameter was identified as 1.

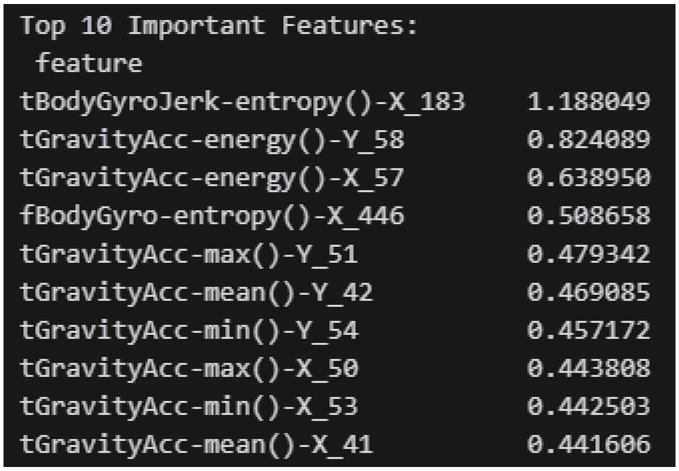
Hyperparameter Tuning - Logistic Regression



```
Best Logistic Regression Parameters: {'C': 1, 'max_iter': 1000, 'penalty': 'l1', 'solver': 'liblinear'}
Best Cross-Validated Accuracy: 0.9824006283858265
Tuned Logistic Regression Accuracy: 0.9868932038834951
Tuned Logistic Regression Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       369
           1       0.99      1.00      1.00       270
           2       1.00      1.00      1.00       284
           3       0.97      0.97      0.97       377
           4       0.97      0.96      0.96       354
           5       1.00      1.00      1.00       406

    accuracy                           0.99      2060
   macro avg       0.99      0.99      0.99      2060
weighted avg       0.99      0.99      0.99      2060
```

Multiple hyperparameters were tested for Logistic Regression, leading to a display of the best values for each. On the tuned accuracy, the model performed slightly better than it did originally.

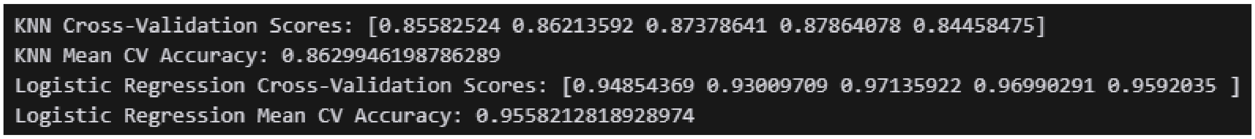
Feature Importance



```
Top 10 Important Features:
 feature
tBodyGyroJerk-entropy()-X_183    1.188049
tGravityAcc-energy()-Y_58        0.824089
tGravityAcc-energy()-X_57        0.638950
fBodyGyro-entropy()-X_446        0.508658
tGravityAcc-max()-Y_51           0.479342
tGravityAcc-mean()-Y_42          0.469085
tGravityAcc-min()-Y_54           0.457172
tGravityAcc-max()-X_50           0.443808
tGravityAcc-min()-X_53           0.442503
tGravityAcc-mean()-X_41          0.441606
```

Top features were identified from gyroscope and gravity acceleration measurements. Key sensor data was revealed, ultimately contributing to classification.

Cross-Validation



```
KNN Cross-Validation Scores: [0.85582524 0.86213592 0.87378641 0.87864078 0.84458475]
KNN Mean CV Accuracy: 0.8629946198786289
Logistic Regression Cross-Validation Scores: [0.94854369 0.93009709 0.97135922 0.96990291 0.9592035 ]
Logistic Regression Mean CV Accuracy: 0.9558212818928974
```

KNN Cross-Validation: The mean cross-validation accuracy for KNN was approximately 86.3%. This indicates variability in performance across different data splits, suggesting that KNN may be sensitive to the specific training data used.

Logistic Regression Cross-Validation: The mean cross-validation accuracy for Logistic Regression was approximately 95.6%. This consistency across folds demonstrates the model's robustness and reliability in handling the dataset.

5 Conclusion

5.1 Concluding Remarks

The project successfully demonstrated the application of machine learning algorithms to classify physical activities using sensor data from smartphones. By leveraging both K-Nearest Neighbors (KNN) and Logistic Regression, high accuracy was achieved in distinguishing between activities such as walking, sitting, and laying. The results highlight the potential of wearable technology in providing valuable insights for personalized fitness and health monitoring.

5.2 Thoughts About the Project

There are a plethora of things that I learned and improved upon during this project such as:

Data Preprocessing: The importance of data normalization and feature selection in improving model performance. Proper preprocessing ensures that models can learn effectively from the data.

Model Selection: Understanding the strengths and limitations of different algorithms was key in making a decision on what would be used. KNN is effective for capturing non-linear patterns, while Logistic Regression excels in linear separability and interpretability.

Evaluation Metrics: The use of precision, recall, and F1-score was essential to gain a comprehensive understanding of model performance beyond just accuracy.

5.3 Challenges

Feature Redundancy: Initially faced challenges with duplicate feature names, which were resolved by ensuring unique feature identifiers. This step was crucial for accurate data loading and model training.

Model Tuning: Finding the optimal parameters for KNN and Logistic Regression required experimentation. Using techniques like GridSearchCV helped identify the best hyperparameters, improving model accuracy.

Class Overlap: Some activities had overlapping feature distributions, making classification challenging. This was addressed by exploring additional features and considering ensemble methods to enhance model robustness.

Overall, this project provided valuable insights into the process of building and evaluating machine learning models for activity recognition. The experience gained will serve as a foundation for future projects involving wearable technology and data-driven health solutions.

6 Acknowledgement

Throughout this project, AI and online resources played a crucial role in guiding the research and development process. Initially, AI was instrumental in brainstorming ideas for potential topics, helping to identify a few very intriguing points of interest. This exploration along with constant thought about engaging subjects led to my eventual selection of a topic that I knew would precisely provide both practical application and technical challenge.

AI also provided valuable insights into potential algorithms to consider for the task. By analyzing the dataset characteristics, project requirements, and going off of what’s been learned in our class thus far, that is where I made the decision to go with the use of K-Nearest Neighbors (KNN) and Logistic Regression, due to research on their capabilities. These algorithms were chosen for their complementary strengths: KNN's ability to capture non-linear relationships and Logistic Regression's efficiency and interpretability.

References & Citation

"Human Activity Recognition Using Smartphones Dataset." *UCI Machine Learning Repository*, University of California, Irvine. Available at: https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones.

Brownlee, Jason. "A Gentle Introduction to Machine Learning." *Machine Learning Mastery*, 2019. Available at: https://machinelearningmastery.com/start-here/.

OpenAI. "ChatGPT." *OpenAI*, 2023, www.openai.com/chatgpt