Design and Implementation of Analytics System

Human Activity Recognition

Akshay Jawale
Dominic Thomas
Parva Jain

School of Graduate Professional Studies
Data Analytics
DAAN 888 – Design and Implementation of Analytics System

Fall, 2022

TABLE OF CONTENTS

Table of Contents

Introduction

Our goal for this project is to classify Human Activities based on data collected from accelerometer and gyroscope sensors of a smartphone. The focus of this project will be on finding optimum number of features for classification tasks followed by comparison of traditional ML techniques.

Business Goals:

To recognize different human activities from aggregates of gyroscope and accelerometer data obtained from a smartphone.

Queries:

- Can human activities be recognized from accelerometer and gyroscope data?
- Can the number of features be reduced to a feasible number?
- Are traditional machine learning techniques accurate enough?
- Which traditional machine learning technique performs better?

Data Description

Data Source:

- Data source is available on the UCI repositories. Since the data is old it contains a few unconventional formats.
- UCI has posted a more usable version of their dataset on Kaggle
- Link: https://www.kaggle.com/datasets/uciml/human-activity-recognition-with-smartphones

Reasoning for selecting dataset:

- It contains a lot of features which might be useful in classification of human activities.
- Each datapoint(metric of human activity) has a label based on footage of the camera.

Description:

- The Human Activity Recognition database was built from the recordings of 30 study participants performing activities of daily living (ADL) while carrying a waist-mounted smartphone with embedded inertial sensors. *The objective is to classify activities into one of the six activities performed*.
- The data collectors have picked a time slice of the data where a particular activity happens and then calculated various aggregates on them from that time slice. These aggregates will act as our independent variables

These signals were used to estimate variables of the feature vector for each pattern:
'-XYZ' is used to denote 3-axial signals in the X, Y and Z directions.

tBodyAcc-XYZ, tGravityAcc-XYZ, tBodyAccJerk-XYZ, tBodyGyro-XYZ,
tBodyGyroJerk-XYZ, tBodyAccMag, tGravityAccMag, tBodyAccJerkMag,

tBodyGyroMag, tBodyGyroJerkMag, fBodyAcc-XYZ, fBodyAccJerk-XYZ, fBodyGyro-XYZ, fBodyAccMag, fBodyAccJerkMag, fBodyGyroMag, fBodyGyroJerkMag

The set of variables that were estimated from these signals are:

mean(): Mean value
std(): Standard deviation
mad(): Median absolute deviation
max(): Largest value in array
min(): Smallest value in array
sma(): Signal magnitude area
energy(): Energy measure. Sum of the squares divided by the number of values.
iqr(): Interquartile range
entropy(): Signal entropy
arCoeff(): Autoregression coefficients with Burg order equal to 4
correlation(): correlation coefficient between two signals
maxInds(): index of the frequency component with largest magnitude
meanFreq(): Weighted average of the frequency components to obtain a mean frequency
skewness(): skewness of the frequency domain signal
kurtosis(): kurtosis of the frequency domain signal
bandsEnergy(): Energy of a frequency interval within the 64 bins of the FFT of each window.
angle(): Angle between two vectors.

Dataset Dimensions: ***Train Set:*** 562(Columns/Features) and 7352(Datapoints)
Test Set: 562(Columns/Features) and 2948(Datapoints)

Category Distribution:

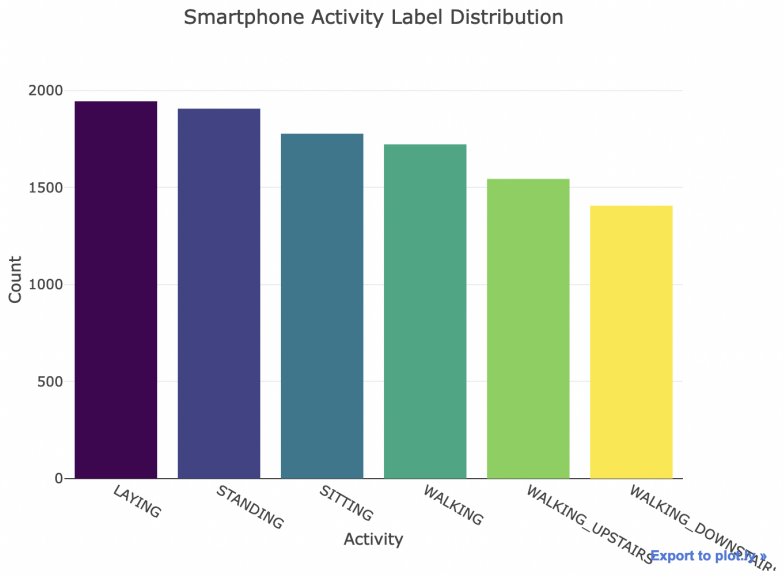

Figure 1: Class Distribution for Activities

Pre-Processing

Data Preparation:

Dataset has 571 features. This is because the data collectors have performed a lot of aggregations on a lot of different metrics out of which may not be useful or relevant for our classification task. Since the data has no inconsistencies or missing values no actual data cleaning is required. As a result we will concentrate most of our efforts on dimensionality reduction.

Dimension Reduction:

For dimension reduction we have tried two techniques, namely Factor Analysis and Principal Component Analysis (PCA). Out of which PCA performed significantly better than Factor Analysis. As a result, we decided to proceed with PCA.

Principal Component Analysis:

Principal Component Analysis aims at reducing the dimensionality of datasets while increasing interpretability and minimizing information loss. This is achieved by creating new variables that are uncorrelated and maximize variance as much as possible.

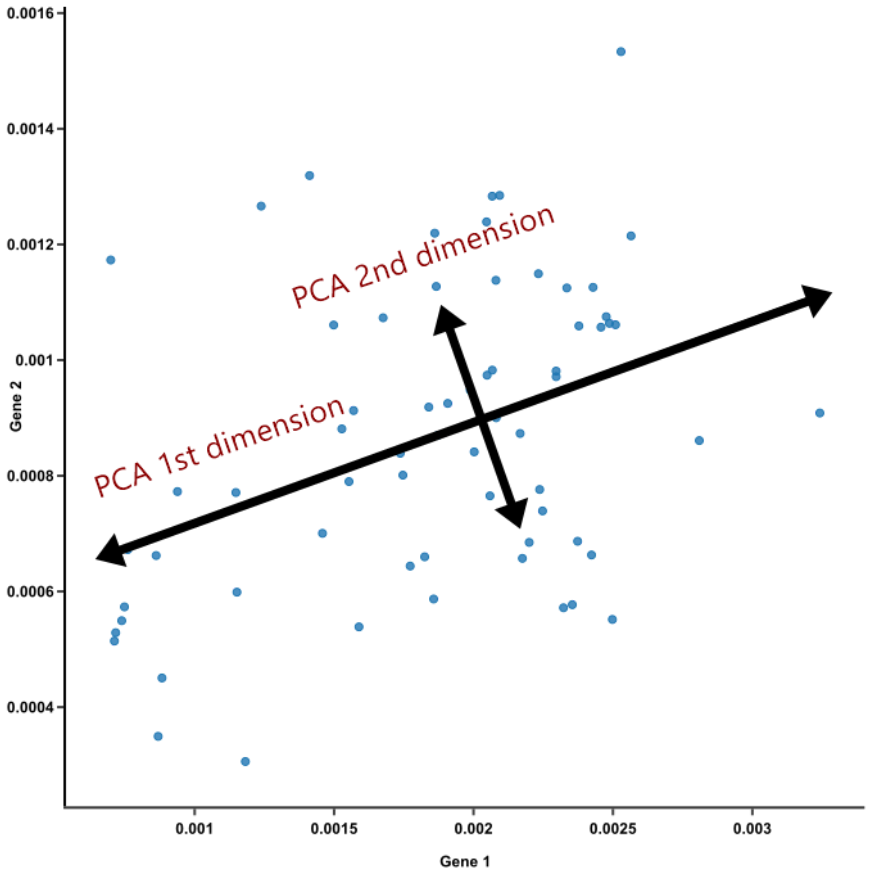

Figure 2: PCA Methodology

PCA recomputes a set of variables in terms of linear equations known as components that represent linear relationships in the data.

- The first component generates a single linear function which tries to capture as much variance as possible
- The second component is generated that tries to capture as much variance as possible from the remaining data.
- This keeps on going until we have same number of components as there are variables

Results of Principal Component Analysis:

To find the optimum number of components for classification, we ran all the base models (Logistic Regression, Support Vector Machine, Random Forest Classifier and XGBoost on base settings) for different number of Principal Components as shown in following figure.

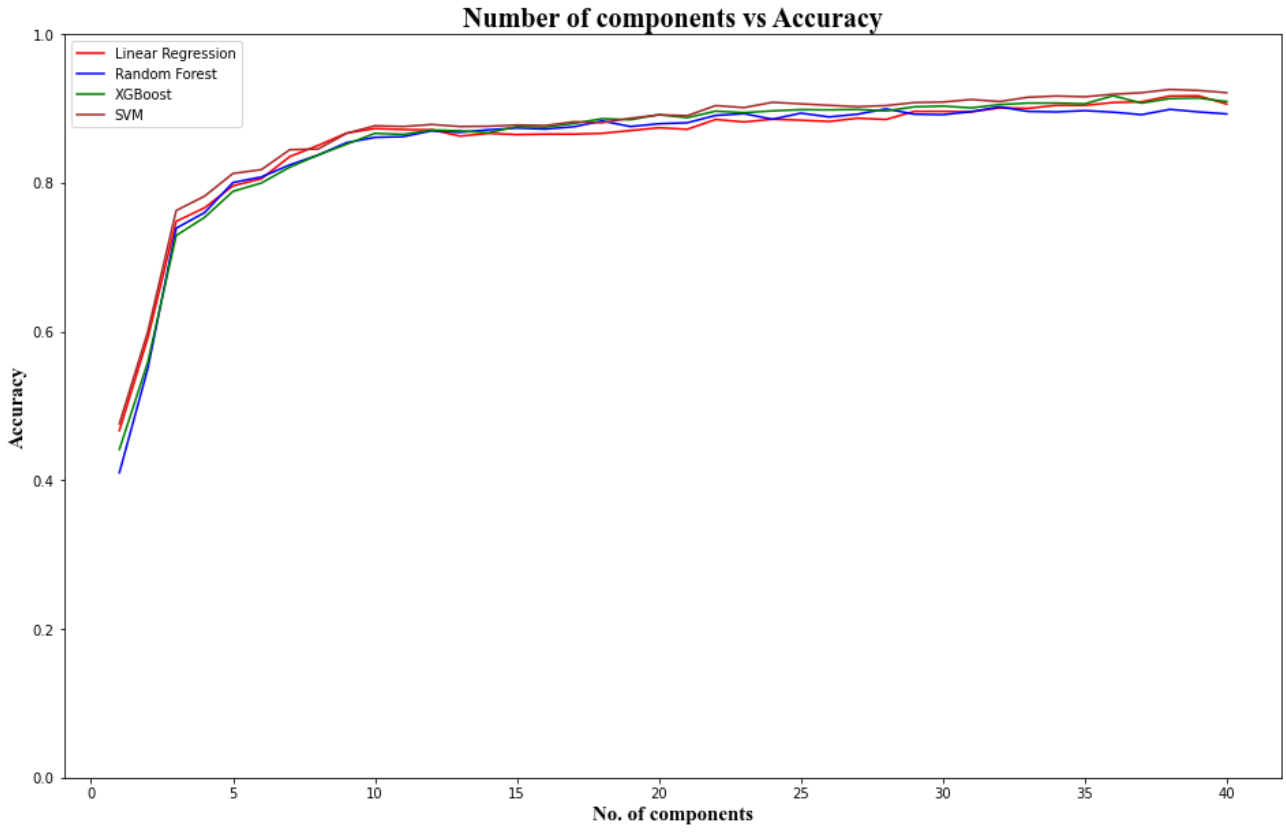

Figure 3: PCA Components vs Model Accuracy

From the above figure we can see that accuracy of models stabilizing between 6 and 12 factors:

N	Log Regression	SVM	Random Forest	XGBoost
6	0.805564981	0.817780794	0.807940278	0.799796403
7	0.835425857	0.844587716	0.824228029	0.821174075
8	0.850356295	0.845605701	0.837461826	0.837461826
9	0.866983373	0.866644045	0.854088904	0.852052935
10	0.873091279	0.876823889	0.861214795	0.866644045
11	0.871733967	0.875805904	0.862232779	0.865286732
12	0.871394639	0.878520529	0.870037326	0.87105531

Table 1: Base accuracies for different number of principal components

Based on above results we see the model accuracies improving less than 1% from 9 components onwards thus we decided to choose 9 as optimum number for our classification task. These 9 components explain 79.6458% of variation of data which is significant improvement in terms of number of features to model accuracy.

Modeling

For Modeling we have chosen to use two traditional and two ensemble machine learning techniques namely Multinomial Logistic Regression, Support Vector Machine(SVM), Random Forest Classifier and XGBoost Classifier.

Logistic Regression is known to be very fast at classifying unknown records and since a lot of use cases for our model would require real time classification, logistic regression would be a good model to test. Another problem we anticipated was not being able to reduce the features significantly, so we also included SVM since it performs very effectively on high dimensional data. In terms of the two ensemble techniques, they are overall good at being stable as well as preventing overfitting as compared to LR and SVM. We also decided to include both random forest and gradient boosting approaches to compare how each would perform on this data.

While finding the optimum number of Principal Components for PCA we have established the base-line accuracies of models under consideration which are as follows:

- Logistic Regression: 86.69%
- Support Vector Machine: 86.66%
- Random Forest Classifier: 85.40%
- XGBoost Classifier: 85.20%

Further we tried to find optimum settings for Hyper-Parameters and ensure the models performance using K-Fold Cross Validation.

Cross Validation

While tuning hyper-parameters we can directly tune the model to improve the performance on the test set but this can result in **test set** information being leaked into the model and evaluation metrics and the model would not be generalized anymore.

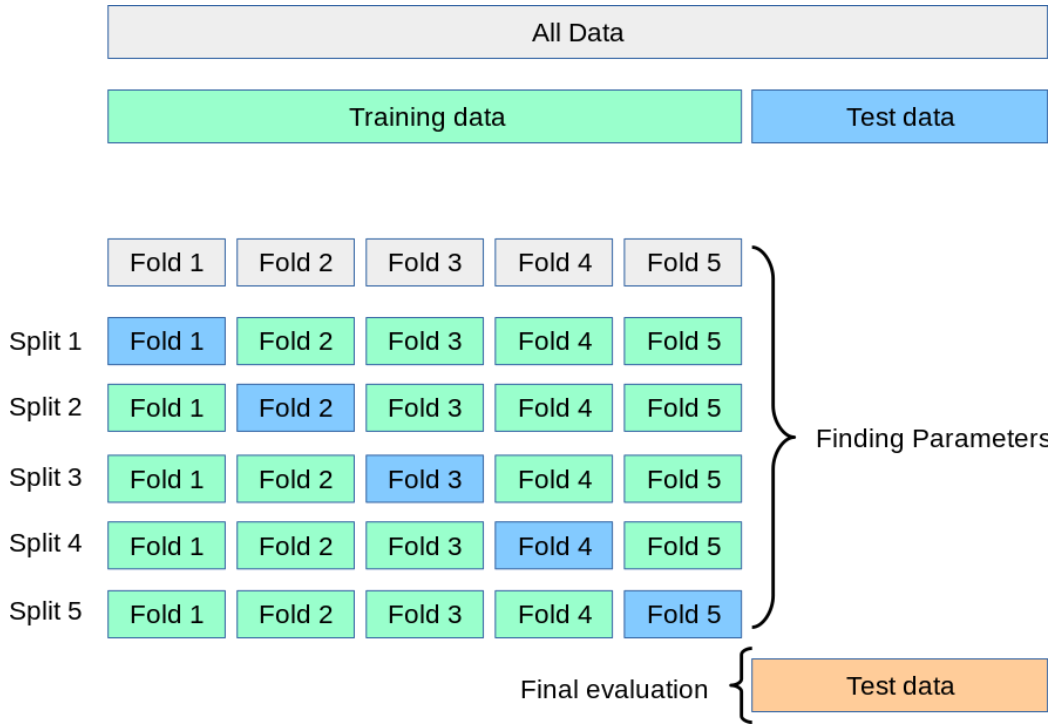

Figure 4: Cross Validation Methodology

This situation can be avoided by performing Cross Validation in which: Dataset is first shuffled and divided into k equal sets and the model is trained on k-1 sets and the remaining set is used for its performance evaluation. This process is repeated k times using each set as a test set and finally the mean of accuracies is calculated as models training accuracy. Thus Cross validation helps ensure that the model is well generalized and not overfitting.

In our case we have chosen the value of k to be 10 which is the default value. The main reason behind this is as the value of k increases, the difference between training (k-1 sets chosen now) set and resampling (k-1 other sets chosen the next time) set becomes smaller, which ultimately reduces bias of the training model. Similarly the accuracies for our model were nearly identical when the value of k was set to 5. Thus we choose k=10 to check if a certain model performs better than the other. We could also afford a 90:10 split since we had a fairly large train set (7352 samples).

Multinomial Logistic Regression

Logistic Regression is a classification method which functions similar to linear regression. The base method of Y = WX + B is performed on the input variables and then is run through a

sigmoid function:

$$S(x) = \frac{1}{1+e^{-x}}$$

The sigmoid function has the curve shown in *Figure X* that helps it decide the class of the particular input.

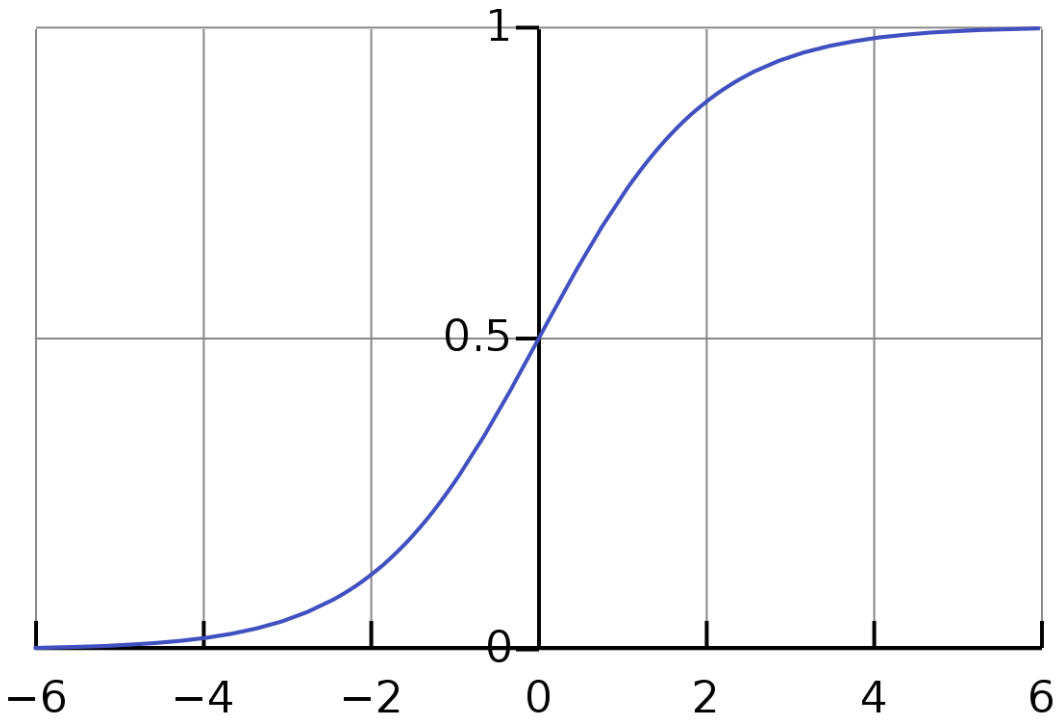

Figure 5: Curve for Sigmoid Function

The loss function for Logistic regression is the maximum likelihood estimation.

Since logistic regression is a binary classifier i.e., it can only distinguish between 2 classes, multinomial logistic regression is used to classify multiple classes. Multinomial Logistic regression computes Logistic regression for each class in a one-versus-all situation and then compares the probabilities for each class to predict the most likely class.

Support Vector Machine:

Support Vector Machine or SVM classifies based on the intuition that for n-features if we plot an n-dimensional graph, we can draw an N-1 dimensional hyperplane that separates the data points.

Since there can be multiple hyperplanes satisfying this condition, the best hyperplane is selected based on the maximum distances from the classes as shown in Figure 5.

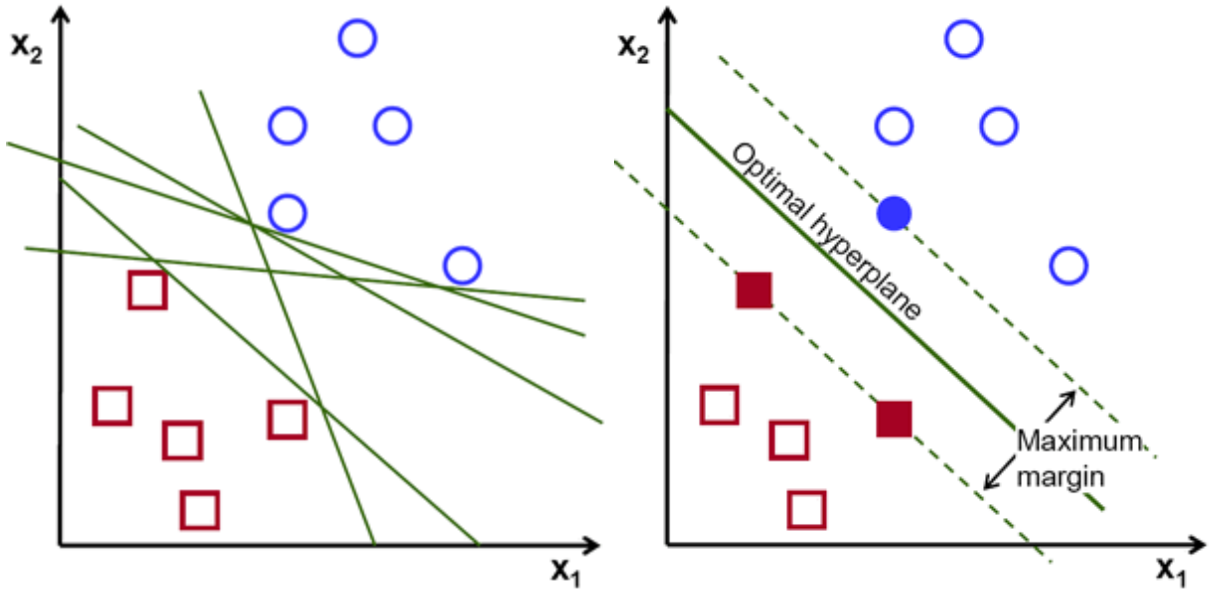

Figure 6 Visualizing optimal hyperplane for SVM

Considering the equation of the hyperplane is

$$\mathbf{w}^{\mathsf{T}}\mathbf{x} - b = 0,$$

For determining the optimal hyperplane is the following equation must be minimized:

$$\lambda\|\mathbf{w}\|^2 + \left[\frac{1}{n}\sum_{i=1}^{n}\max\left(0, 1 - y_i(\mathbf{w}^{\mathsf{T}}\mathbf{x}_i - b)\right)\right],$$

The above example is for a linear hyperplane, but an SVM plane can be of many different shapes. They can be polynomials of varying degrees, radial basis functions (high dimensional equivalent of circles in 2D), logarithm, etc. Support Vector Machines produce significant accuracy with less computation power.

Random Forest Classifier

Decision trees are a category of supervised machine learning algorithms that can be used both for classification and regression tasks.

Random forests are constructed using multiple Decision Trees (ensemble). An element of randomization is introduced in the construction of tree by Random Forest

Below is an example using random forest in a classification problem.

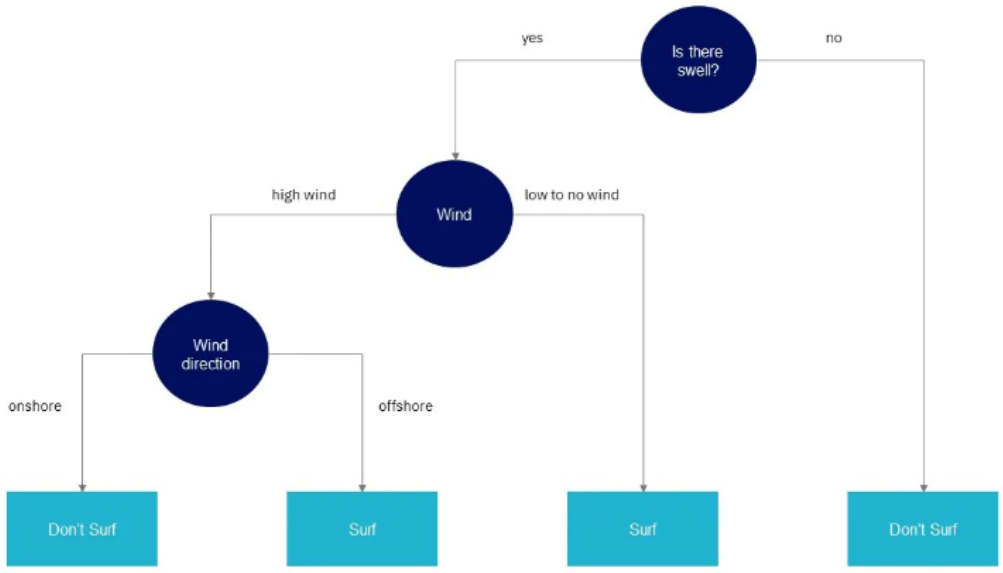

Figure 7: Example of a Decision Tree

XGBoost Classifier

XGBoost is another classifier that uses decision trees as its basis. The n-estimators or number of trees parameter builds n number of trees giving them minor inconsistencies based on the selected regularization penalties. The difference in Gradient Boosting techniques such as XGBoost is that the outputs from all the trees are used as residuals in the next tree's calculations. The resulting equation looks as follows:

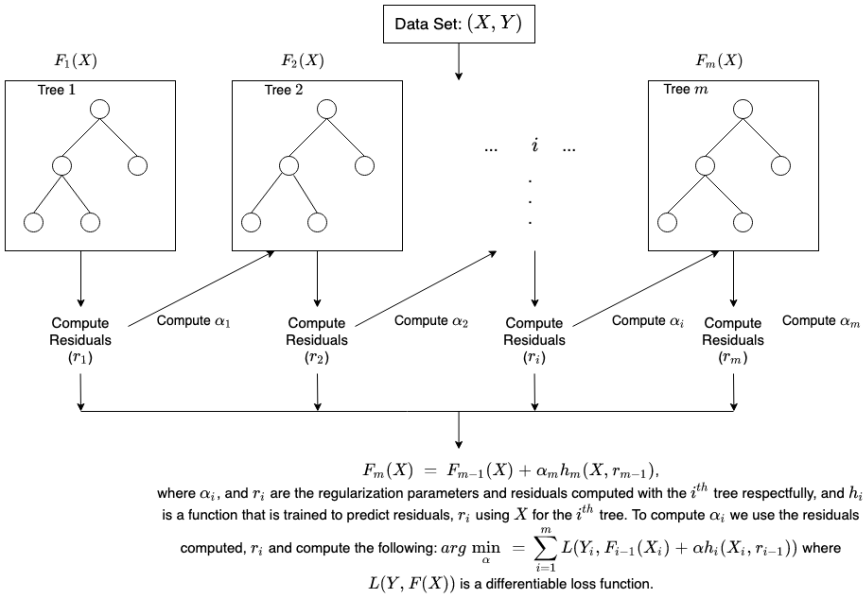

Figure 8: Gradient Boosting

Hence the trees form an equation that can be minimized and hence the word ‘gradient’ is used. XGBoost is just an extremely efficient open-source implementation of this gradient boosting method.

Results and Evaluation

After all the models were run on their tuned settings we evaluate the models based on precision, recall, F1 score and accuracy. Along with that we also evaluate its confusion matrix to determine which classes are having more trouble classifying and discuss it in the Discussion section.

Precision: Proportion of Correct Positive Classifications

$$Precision = \frac{TP}{TP+FP}$$

Recall: Proportion of Actual Positive Correctly Classified

$$Recall = \frac{TP}{TP+FN}$$

F1 Score: F1 is a measure that finds balance between Precision and Recall also helping in evaluation if the dataset is unbalanced.

$$F1\ Score = \frac{2PR}{P+R}$$

Accuracy: Instances Correctly Classified

Confusion Matrix: Matrix to display number of True Positives, True Negatives, False Positives and False Negatives for each Class helping us check proportion of correct classification and class imbalance if present.

Results of Logistic Regression are as follows:

	Precision	Recall	F1 Score	Support
LAYING	1.00	0.96	0.98	537
SITTING	0.79	0.73	0.76	491
STANDING	0.78	0.86	0.82	532
WALKING	0.88	0.97	0.92	496
WALKING_DOWNSTAIRS	0.88	0.77	0.82	420
WALKING_UPSTAIRS	0.87	0.88	0.87	471
Accuracy			0.87	2947
Macro Average	0.87	0.86	0.86	2947
Weighted Average	0.87	0.87	0.87	2947

Table 2: Multinomial Logistic Regression Classification Report

From the classification report above we conclude that "LAYING" class is always classified correctly while "SITTING" and "STANDING" classes have the highest rates of miss classification

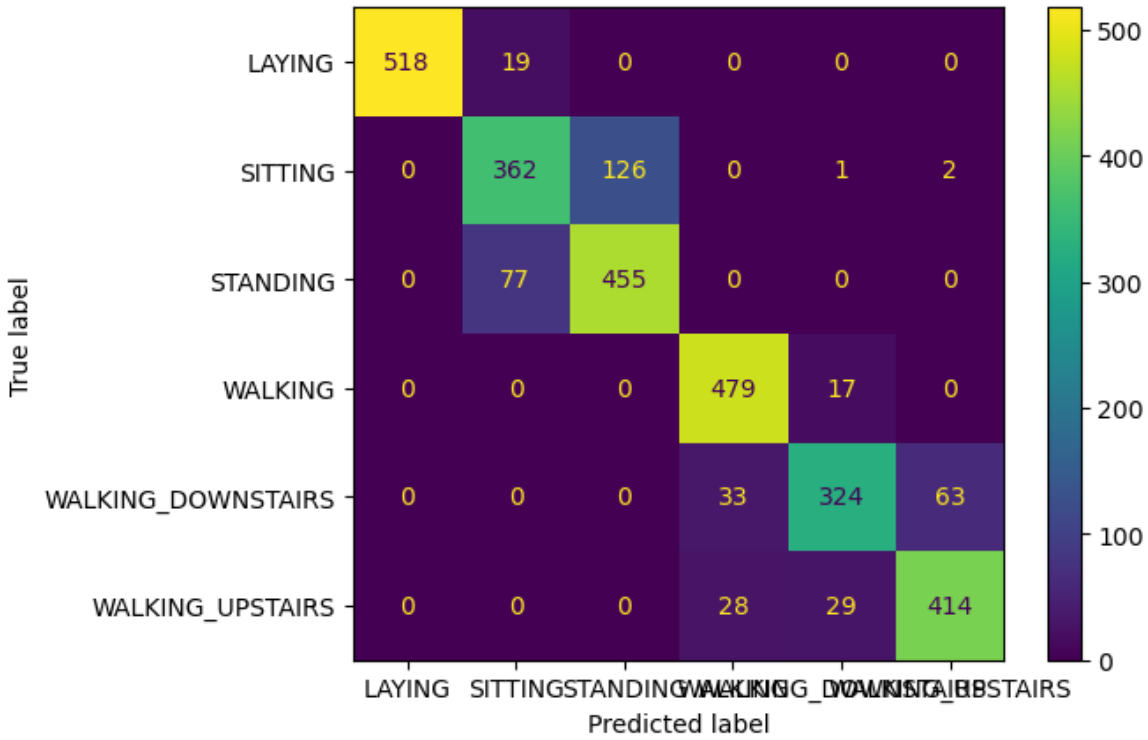

Figure 9: Confusion matrix for Logistic Regression

From the confusion matrix we understand that "Lying" class has the highest instances of correct classification. 126 instances of class "Sitting" has been misclassified as Standing which is the

highest amount of misclassification. The classes ``WALKING_DOWNSTAIRS", "WALKING_UPSTAIRS" are also getting misclassified amongst each other.

Results of Support Vector Machine (SVM) are as follows:

	Precision	Recall	F1 Score	Support
LAYING	1.00	0.97	0.98	537
SITTING	0.79	0.74	0.77	491
STANDING	0.78	0.85	0.82	532
WALKING	0.89	0.97	0.93	496
WALKING_DOWNSTAIRS	0.88	0.79	0.83	420
WALKING_UPSTAIRS	0.88	0.88	0.88	471
accuracy			0.87	2947
macro avg	0.87	0.87	0.87	2947
weighted avg	0.87	0.87	0.87	2947

Table 3: Support Vector Machine(SVM) Classification Report

From the classification report above we conclude that the "LAYING" class is always classified correctly while "SITTING" and "STANDING" classes have the highest rates of miss classification.

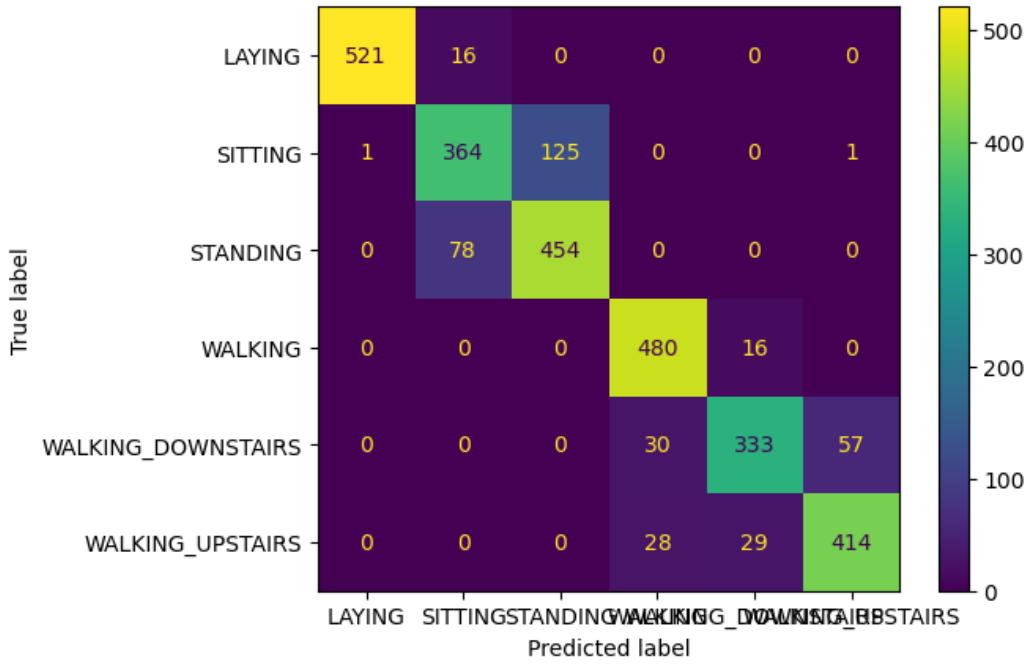

Figure 10: Confusion matrix for Support Vector Machine

From the configuration matrix we understand that “Lying” class has the highest instances of correct classification . 125 instances of class “Sitting” has been misclassified as Standing which is the highest amount of misclassification. The Classes ``WALKING_DOWNSTAIRS”, “WALKING_UPSTAIRS” are also getting misclassified amongst each other.

Result of Random Forest Classifier are as follows:

	Precision	Recall	F1 Score	Support
LAYING	1.00	0.97	0.98	537
SITTING	0.78	0.69	0.74	491
STANDING	0.76	0.85	0.80	532
WALKING	0.83	0.96	0.89	496
WALKING_DOWNSTAIRS	0.84	0.77	0.80	420
WALKING_UPSTAIRS	0.90	0.83	0.86	471
accuracy			0.85	2947
macro avg	0.85	0.85	0.85	2947
weighted avg	0.85	0.85	0.85	2947

Table 4: Random Forest Classifier Classification Report

From the classification report above we conclude that the “LAYING” class is always classified correctly while “SITTING” and “STANDING” classes have the highest rates of miss classification.

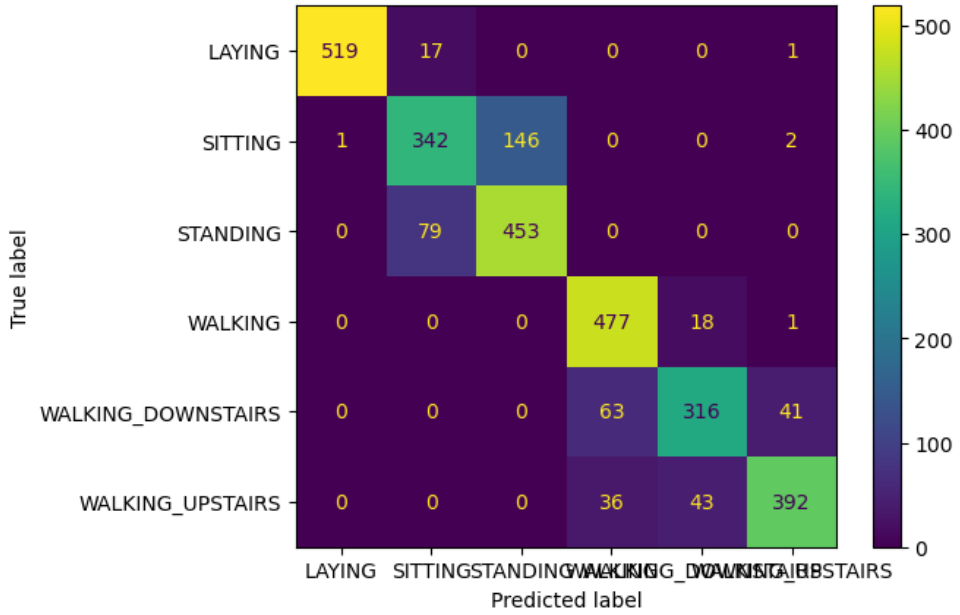

Figure 11: Confusion matrix for Random Forest Classifier

From the configuration matrix we understand that “Lying” class has the highest instances of correct classification . 146 instances of class “Sitting” has been misclassified as Standing which is the highest amount of misclassification. Classes ``WALKING_DOWNSTAIRS”, “WALKING_UPSTAIRS” are also getting misclassified amongst each other.

Results of XGBoost Classifier are as follows:

	Precision	Recall	F1 Score	Support
LAYING	1.00	0.96	0.98	537
SITTING	0.77	0.69	0.73	491
STANDING	0.75	0.86	0.80	532
WALKING	0.83	0.96	0.89	496
WALKING_DOWNSTAIRS	0.87	0.75	0.80	420
WALKING_UPSTAIRS	0.89	0.85	0.87	471
accuracy			0.85	2947
macro avg	0.85	0.84	0.85	2947
weighted avg	0.85	0.85	0.85	2947

Table 5: XGBoost Classifier Classification Report

From the classification report above we conclude that the “LAYING” class is always classified correctly while “SITTING” and “STANDING” classes have the highest rates of miss classification.

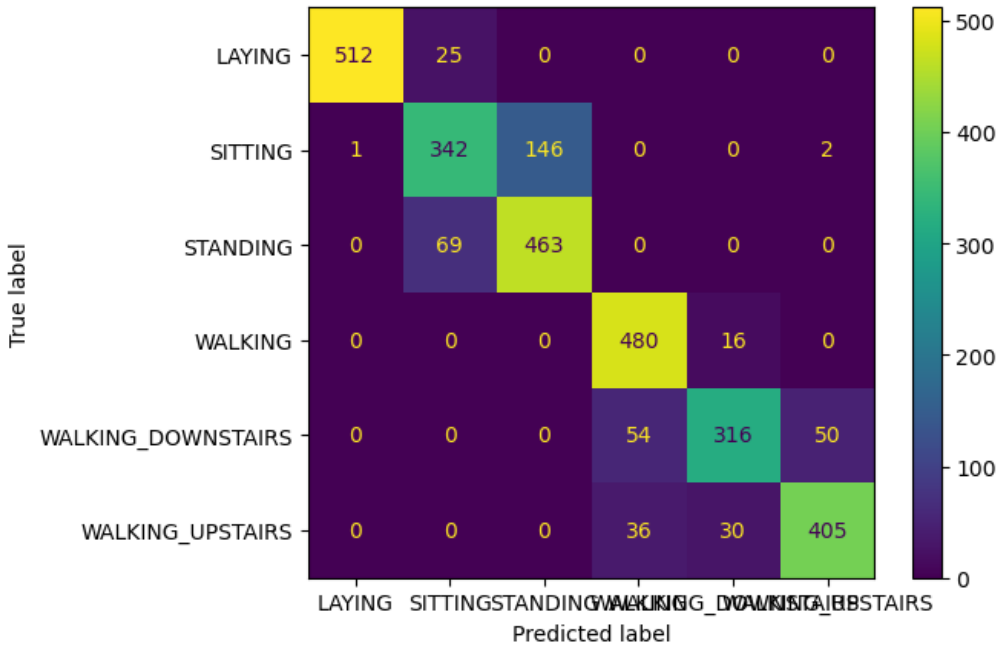

Figure 12: Confusion matrix for XGBoost Classifier

From the configuration matrix we understand that the “Lying” class has the highest instances of correct classification . 146 instances of class “Sitting” have been misclassified as Standing which is the highest amount of misclassification. Classes ``WALKING_DOWNSTAIRS”, “WALKING_UPSTAIRS” are also getting misclassified amongst each other.

Discussion and future work

To address our business questions, our results verify that gyroscope and accelerometer data can be used to classify Human Activities(Certain activities up to certain extent) using machine learning techniques. Using PCA, we were able to reduce the number of features from 571 to 9 Principal components which could explain 79.6458% of variation of data. Results from the 4 models average around 86% accuracy suggesting that models are good enough for classification purposes. Finally, we can see that the Multinomial Logistic Regression and Support Vector Machine is performing better than Random Forest Classifier and XGBoost Classifier.

From the results we understand that Logistic Regression and SVM are slightly outperforming the two ensemble methods. This difference is not significant enough to warrant choosing one model over the others. One of the possible reasons for this outcome could be that both Random Forest Classifier and XGBoost use Decision Trees which are more favorable for categorical data as compared to our continuous data. Another reason could be that our data is sparse and easy to classify which helps SVM generate a better hyperplane.

We can see from the classification report and the confusion matrix that the majority of misclassification in most models occurs between ‘SITTING’ and ‘STANDING’. Both have precision, recall and f1 scores closer to 0.8. This might be because they are both stationary activities with the upper body staying vertical since the smartphone is strapped to the waist during data collection.

Similar conclusions can be drawn for ‘WALKING’, ‘WALKING_DOWNSTAIRS’ and “WALKING_UPSTAIRS’ classes as they have very similar upper body movements.

In the future a lot of these deficiencies would be solved by just recording accelerometer and gyroscope data from different areas of the body. For example, one set from the upper body and one from the lower body. This would be extremely beneficial in improving classification of activities that have similar upper body movements.

Another approach that could improve detection of the ambiguous classes would be to use Neural Networks as those are better at recognizing more minute patterns in the data.

References

- https://en.wikipedia.org/wiki/Sigmoid_function
- https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47
- https://machinelearningmastery.com/multinomial-logistic-regression-with-python/
- https://en.wikipedia.org/wiki/Support_vector_machine
- https://scikit-learn.org/stable/modules/cross_validation.html
- https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9
- https://blog.bioturing.com/2018/06/14/principal-component-analysis-explained-simply/
- https://docs.aws.amazon.com/sagemaker/latest/dg/xgboost-HowItWorks.html
- https://www.ibm.com/cloud/learn/random-forest
- Dusan Ramljak, 2022, Data Driven Decision Making, Lesson 05
- Jolliffe Ian T. and Cadima Jorge, 2016, Principal component analysis: a review and recent developments, Phil. Trans. R. Soc. A.3742015020220150202 http://doi.org/10.1098/rsta.2015.0202
- Mastering Predictive Analytics with - R Rui Miguel Forte
- Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd Edition - Aurélien Géron