Human Activity Recognition on Smartphones

Mohammed Ashfaq Ali | Machine leaning prediction | Aug 2022.

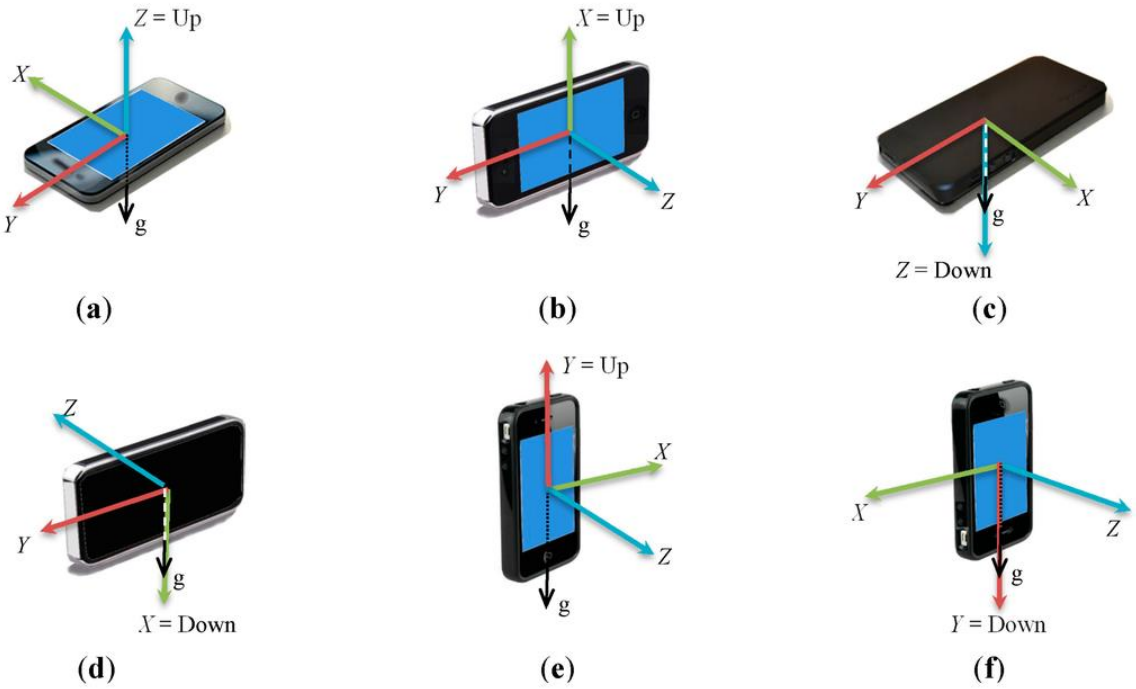
In the guidance of Mr. Saurabh Ranjan.

Abstract:

Activity-Based Computing aims to capture the state of the user and its environment by exploiting heterogeneous sensors in order to provide adaptation to exogenous computing resources. When these sensors are attached to the subject's body, they permit continuous monitoring of numerous physiological signals. This has appealing use in healthcare applications, e.g. the exploitation of Ambient Intelligence (AmI) in daily activity monitoring for elderly people

Introduction:

Consider the below figure where different axes that is x, y and z axes of accelerometer sensors in different directions are shown and these axes change their directions according to the rotation of phone. So if anyone having a smartphone performs some activities then direction of these 3-axes will change accordingly. Then by observing these different values of accelerometer axes we can predict different type of activities performed by that person.



Background:

The first HAR approach contains a large number of sensor type technologies that can be worn on-body known as wearable sensors, ambient sensors, and, together, both will make hybrid sensors that help in measuring quantities of human body motion. Various opportunities can be provided by these sensor technologies which can improve the robustness of the data through which human activities can be detected and also provide the services based on sensed information from real-time environments, such as cyber-physical-social systems there is also a type of magnetic sensors when embedded in smartphone can track the positioning without any extra cost. 2. Vision-based—RGB video and depth cameras being used to obtain human actions. 3. Multimodal—Sensors data and visual data are being used to detect human activities

Dataset: **Kaggle**

Data collection:

Steps to be performed:

- Collecting the dataset and importing in jupyter notebook.
- Cleaning the data by checking the null values and removing it.
- Performing "Exploratory data analysis (EDA)".
- Transforming Non numerical Labels into numerical labels.
- Performing standard scalar and splitting the data into training and test data.
- Applying machine learning algorithms.

Machine Learning:

What is Machine learning?

Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.

Algorithms used:

1) Linear Regression
2) Logistic Regression
3) Support vector machine (SVM)
4) Decision tree
5) Random forest
6) Bagging classifier
7) XG Boosting
8) Ada Boosting
9) Gradient Boosting

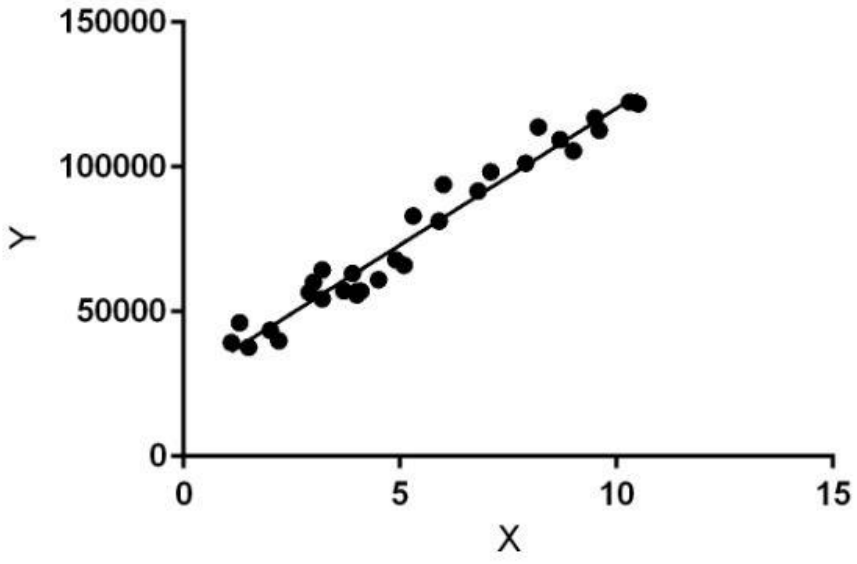
Linear Regression:

Linear regression analysis is used to predict the value of a variable based on the value of another variable. The variable you want to predict is called the dependent variable. The variable you are using to predict the other variable's value is called the independent variable.

Equation is Y= MX+C

---- X is independent variable

---- Y is dependent variable



Accuracy score for linear model is defined using

r_sq = model.score(X_test, y_test)

	Predicted Values	Actual Values
0	-0.266657	0
1	3.613355	4
2	4.276231	4
3	4.061033	4
4	3.167156	3
5	1.762185	2
6	-0.027905	0
7	3.024659	3
8	4.700019	5
9	4.735600	5

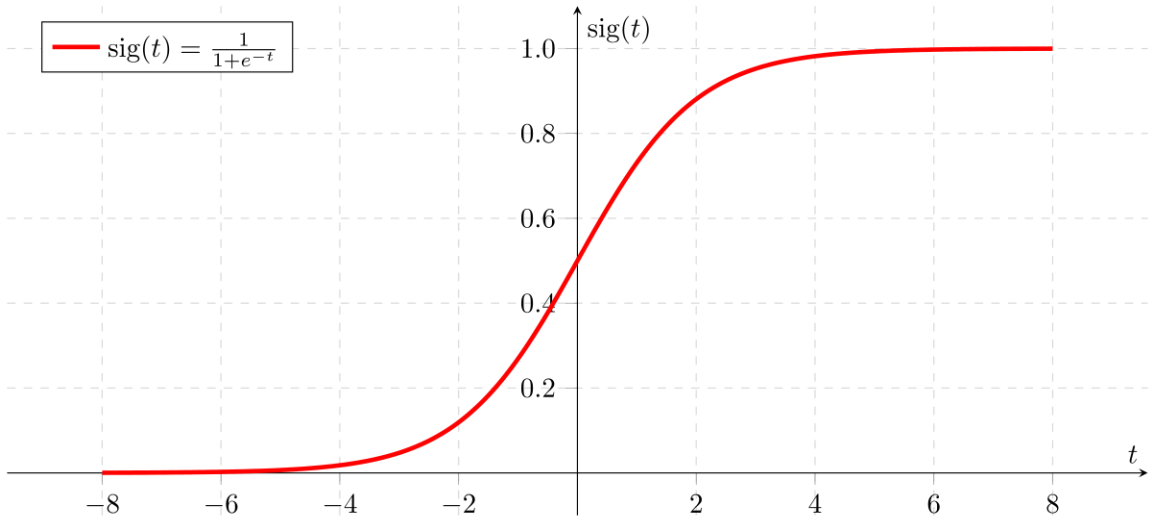
Logistic Regression:

Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1. In logistic regression, a logit transformation is applied on the odds—that is, the probability of success divided by the probability of failure. This is also commonly known as the log odds, or the natural logarithm of odds, and this logistic function is represented by the following formulas:

Logit(pi) = 1/(1+ exp(-pi))

ln(pi/(1-pi)) = Beta_0 + Beta_1*X_1 + … + B_k*K_k

Sigmoid Function:



Support Vector Machine:

Support vector machines (SVMs) are powerful yet flexible supervised machine learning algorithms which are used both for classification and regression. But generally, they are used in classification problems.

Working of SVM:

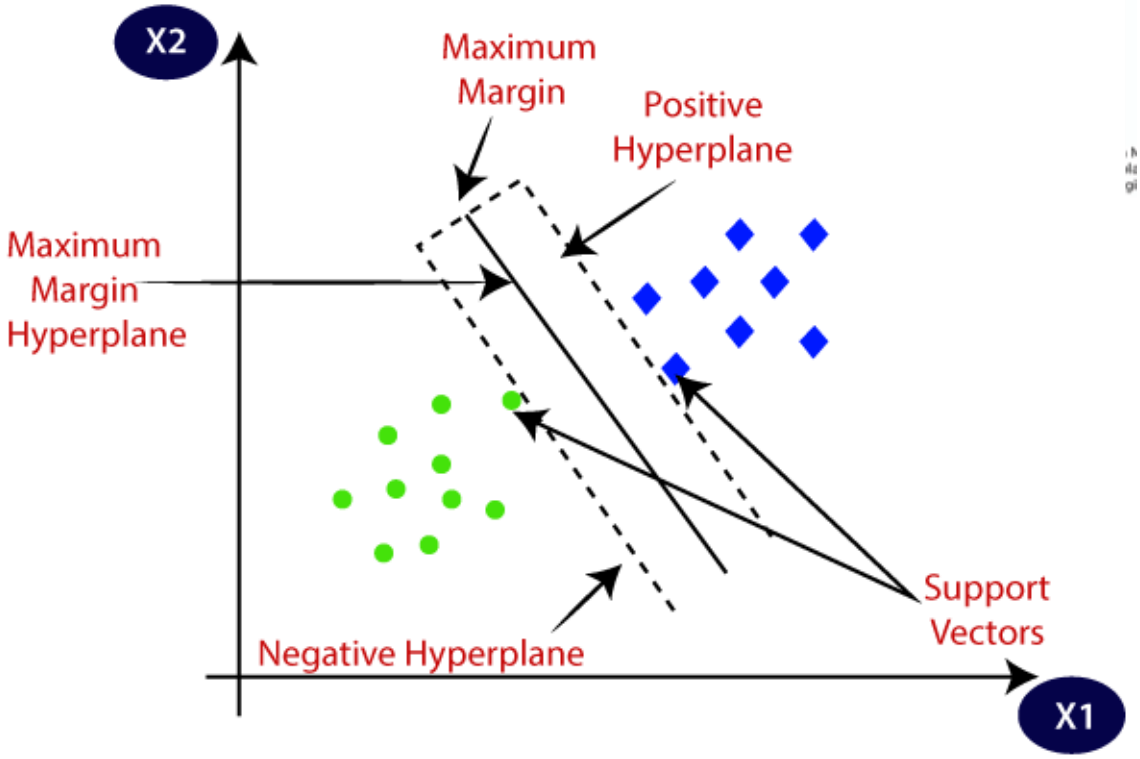
An SVM model is basically a representation of different classes in a hyperplane in multidimensional space. The hyperplane will be generated in an iterative manner by SVM so that the error can be minimized. The goal of SVM is to divide the datasets into classes to find a maximum marginal hyperplane (MMH).

The followings are important concepts in SVM −

Support Vectors − Data points that are closest to the hyperplane is called support vectors. Separating line will be defined with the help of these data points.

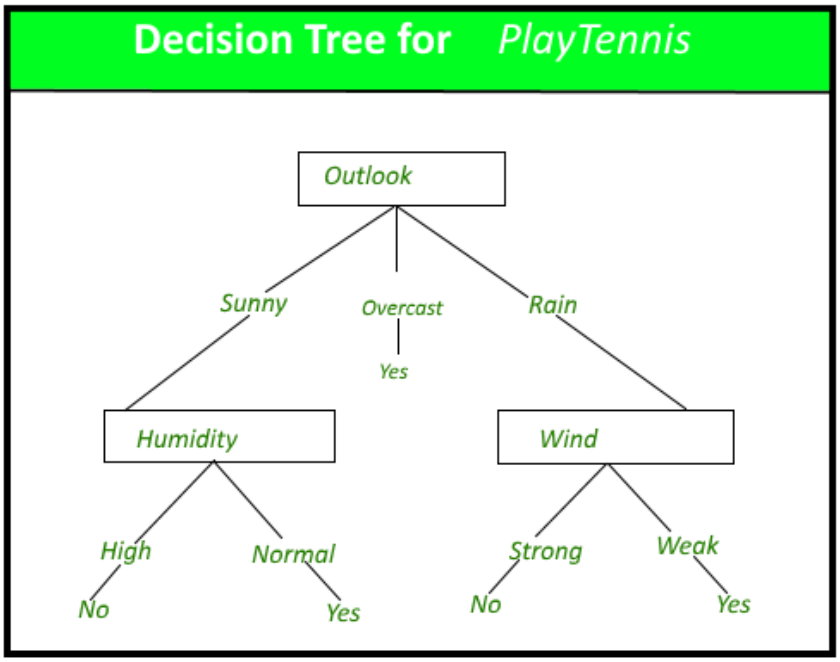
Hyperplane − As we can see in the above diagram, it is a decision plane or space which is divided between a set of objects having different classes.

Margin − It may be defined as the gap between two lines on the closet data points of different classes. It can be calculated as the perpendicular distance from the line to the support vectors. Large margin is considered as a good margin and small margin is considered as a bad margin.



Decision Tree:

It is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.



Random Forest:

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

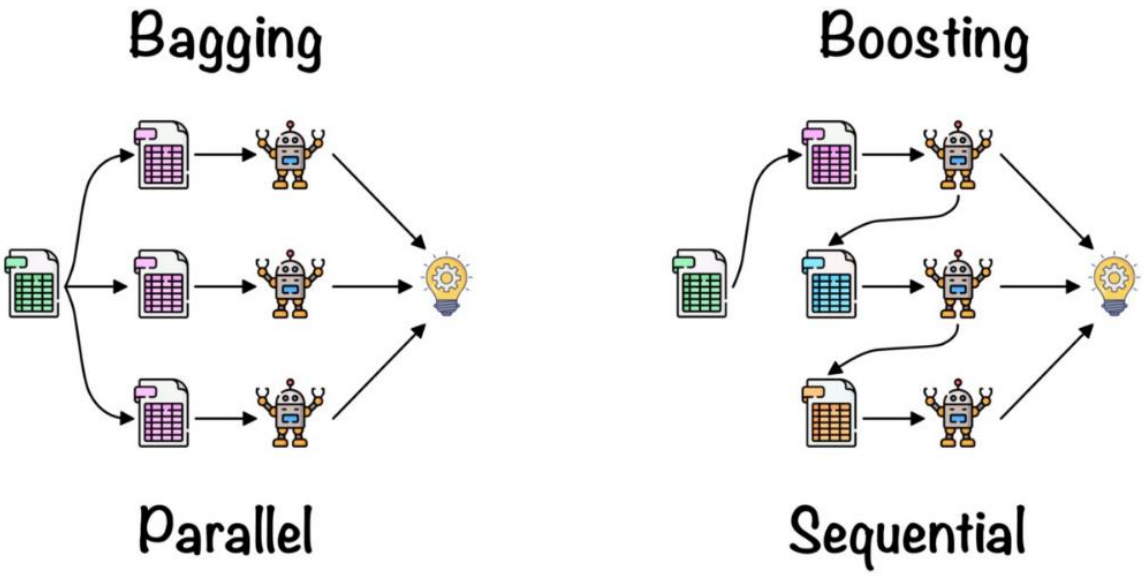
One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.

Working of Random Forest Algorithm:

Before understanding the working of the random forest we must look into the ensemble technique. Ensemble simply means combining multiple models. Thus a collection of models is used to make predictions rather than an individual model.

1. **Bagging**– It creates a different training subset from sample training data with replacement & the final output is based on majority voting. For example, Random Forest.

2. **Boosting**– It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy. For example, ADA BOOST, XG BOOST



Bagging classifier:

A Bagging classifier is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction.

XG Boosting:

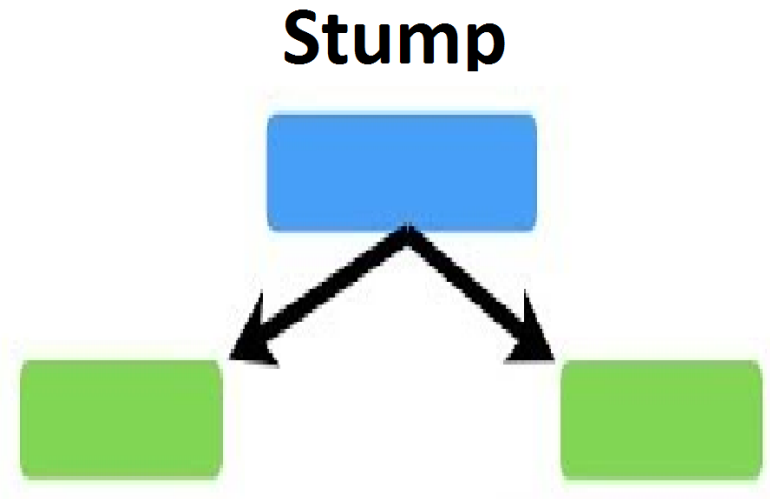
XGBoost, which stands for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems.

It's vital to an understanding of XGBoost to first grasp the machine learning concepts and algorithms that XGBoost builds upon: supervised machine learning, decision trees, ensemble learning, and gradient boosting.

Supervised machine learning uses algorithms to train a model to find patterns in a dataset with labels and features and then uses the trained model to predict the labels on a new dataset's features.

Ada Boosting:

AdaBoost also called Adaptive Boosting is a technique in Machine Learning used as an Ensemble Method. The most common algorithm used with AdaBoost is decision trees with one level that means with Decision trees with only 1 split. These trees are also called Decision Stumps.



Source: Google

Gradient Boosting:

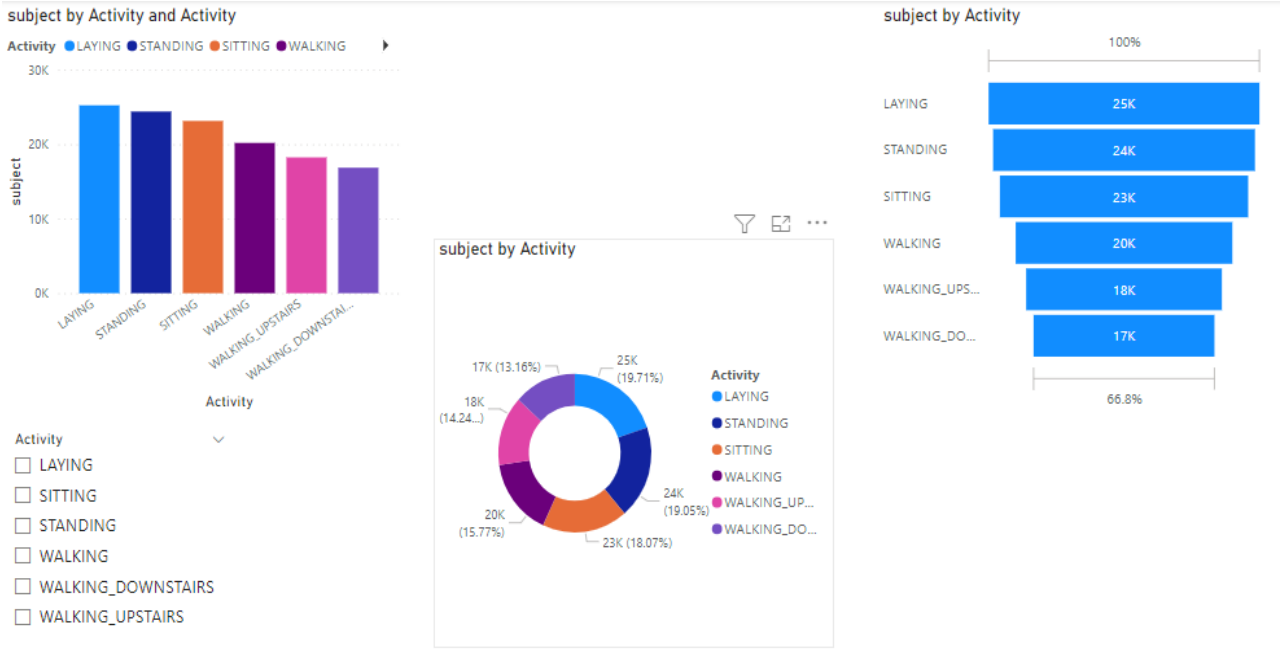
Gradient Boosting is a popular boosting algorithm. In gradient boosting, each predictor corrects its predecessor's error. In contrast to Adaboost, the weights of the training instances are not tweaked, instead, each predictor is trained using the residual errors of predecessor as labels.

There is a technique called the Gradient Boosted Trees whose base learner is CART (Classification and Regression Trees).

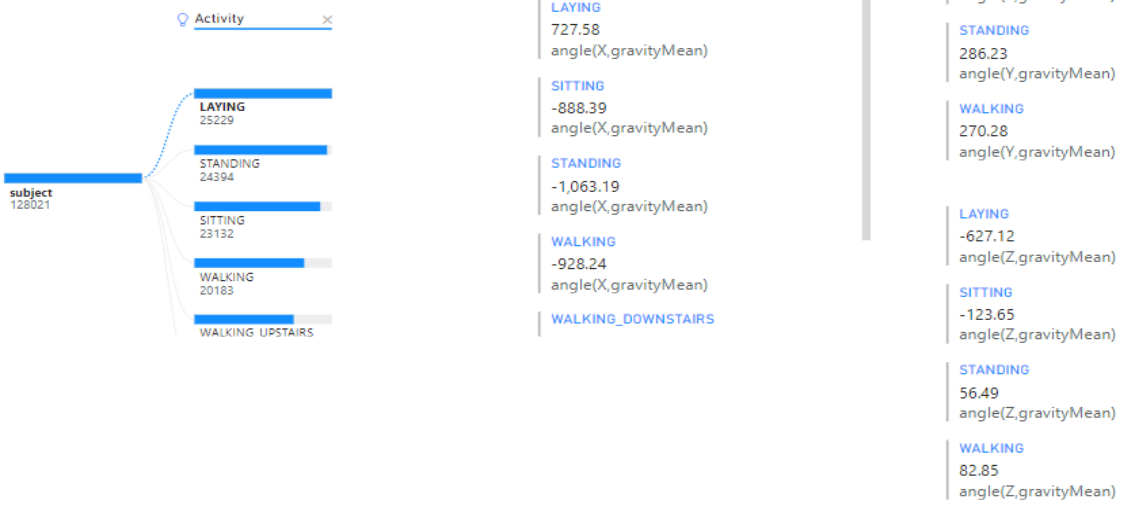
When the target column is continuous, we use Gradient Boosting Regressor whereas when it is a classification problem, we use Gradient Boosting Classifier. The only difference between the two is the "Loss function". The objective here is to minimize this loss function by adding weak learners using gradient descent. Since it is based on loss

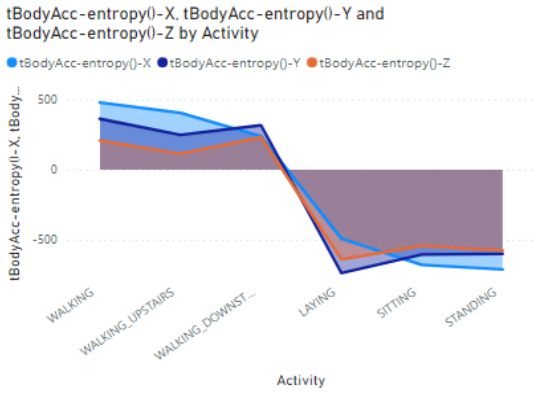
function hence for regression problems, we'll have different loss functions like Mean squared error (MSE) and for classification, we will have different for e.g log-likelihood.

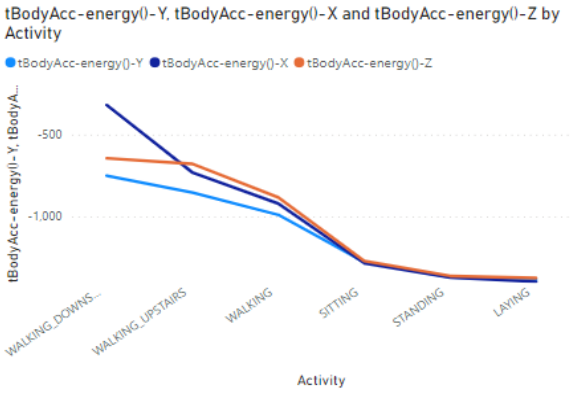
Data Visualization using Power BI:



The Dashboard is created using 'Power BI'. It clearly shows the plotting between subject and activity.

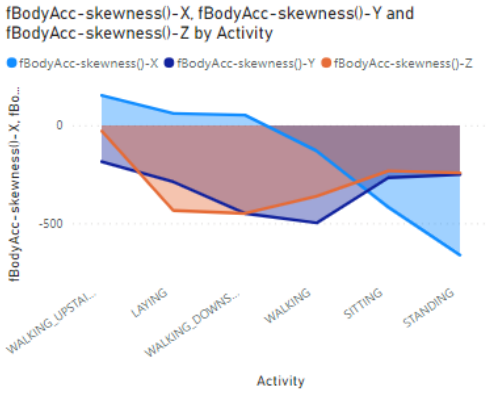


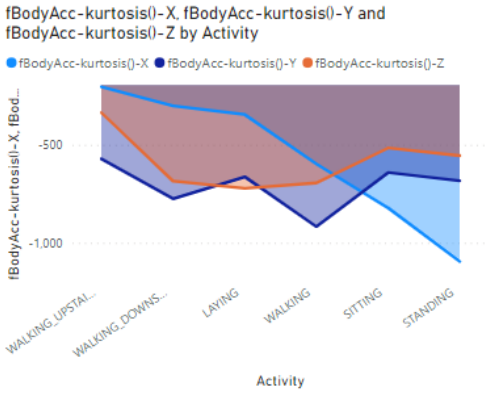




Here graph 1 shows the plotting between bodyAcc entropy and activity.

Here graph 2 shows the plotting between bodyAcc energy and activity.





Here graph 1 shows the plotting between bodyAcc skewness and activity.

Here graph 2 shows the plotting between bodyAcc kurtosis and activity.

Comparison of bagging and boosting models:

It is done by importing the library called “PrettyTable”.

Class inside the prettytable library is used to create relational tables in Python

```
from prettytable import PrettyTable
x.field_names = ["Model", "Accuracy"]
x.add_row(["Linear Regression Model", round(r_sq,2)])
x.add_row(["Logistic Regression model", round(Lr,2)])
x.add_row(["Support Vector Machine", round(sv,2)])
x.add_row(["Decision Tree Model", round(dt,2)])
```

x.add_row(["Random Forest Classifier Model",round(rn,2)])

x.add_row(["Bagging Classifier Model", round(bc,2)])

x.add_row(["XGB Classifierr Model", round(xg,2)])

x.add_row(["AdaBoost Classifier Model", round(ad,2)])

x.add_row(["Gradient Boosting Classifier Model", round(gb,2)])

Model	Accuracy
Linear Regression Model	0.98
Logistic Regression model	0.99
Support Vector Machine	0.98
Decision Tree Model	0.94
Random Forest Classifier Model	0.94
Bagging Classifier Model	0.96
XGB Classifier Model	0.99
AdaBoost Classifier Model	0.54
Gradient Boosting Classifier Model	0.98

XG Boosting and Logistic regression have given highest accuracy of 99% and while Ada Boosting had given the low accuracy of 54%