**Classification Based Human Activity Recognition Model**

**Project overview:** Human activity recognition finds many applications in areas such as surveillance, and sports. Such a system classifies a spatio-temporal feature descriptor of a human figure in a video, based on training examples. However many classifiers face the constraints of the long training time, and the large size of the feature vector. Our method, due to the use of an Support Vector Machine (SVM) classifier, on an existing spatio-temporal feature descriptor resolves these problems in human activity recognition. Comparison of our system with existing classifiers using two standard datasets shows that our system is much superior in terms of the computational time, and either it surpasses or is on par with the existing recognition rates. It performs on par or marginally inferior to existing systems, when the number of training examples are a few due to the imbalance, although consistently better in terms of computation time.

**Dataset overview:** 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.

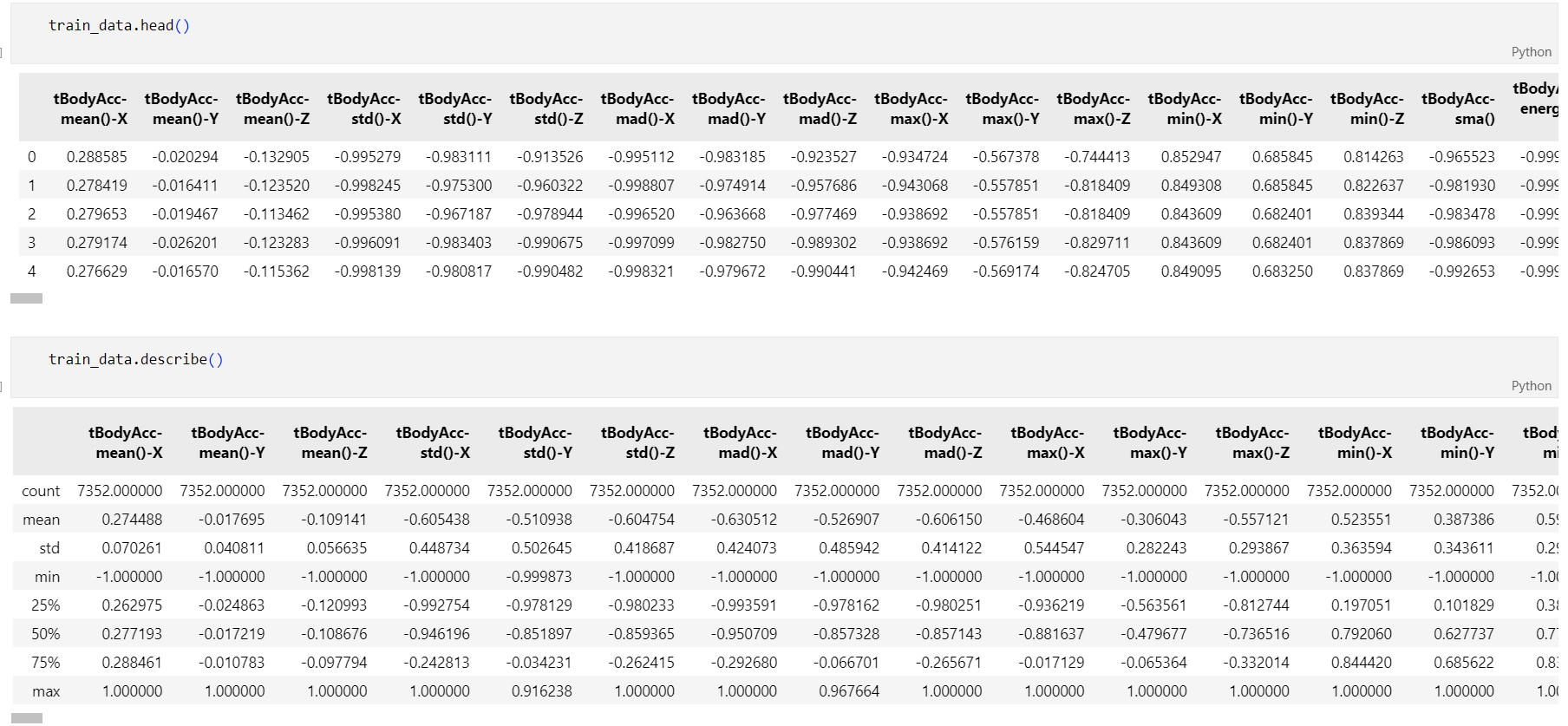
Data source: <https://www.kaggle.com/code/saikiranpolu/human-activity-recognition-iml/data>

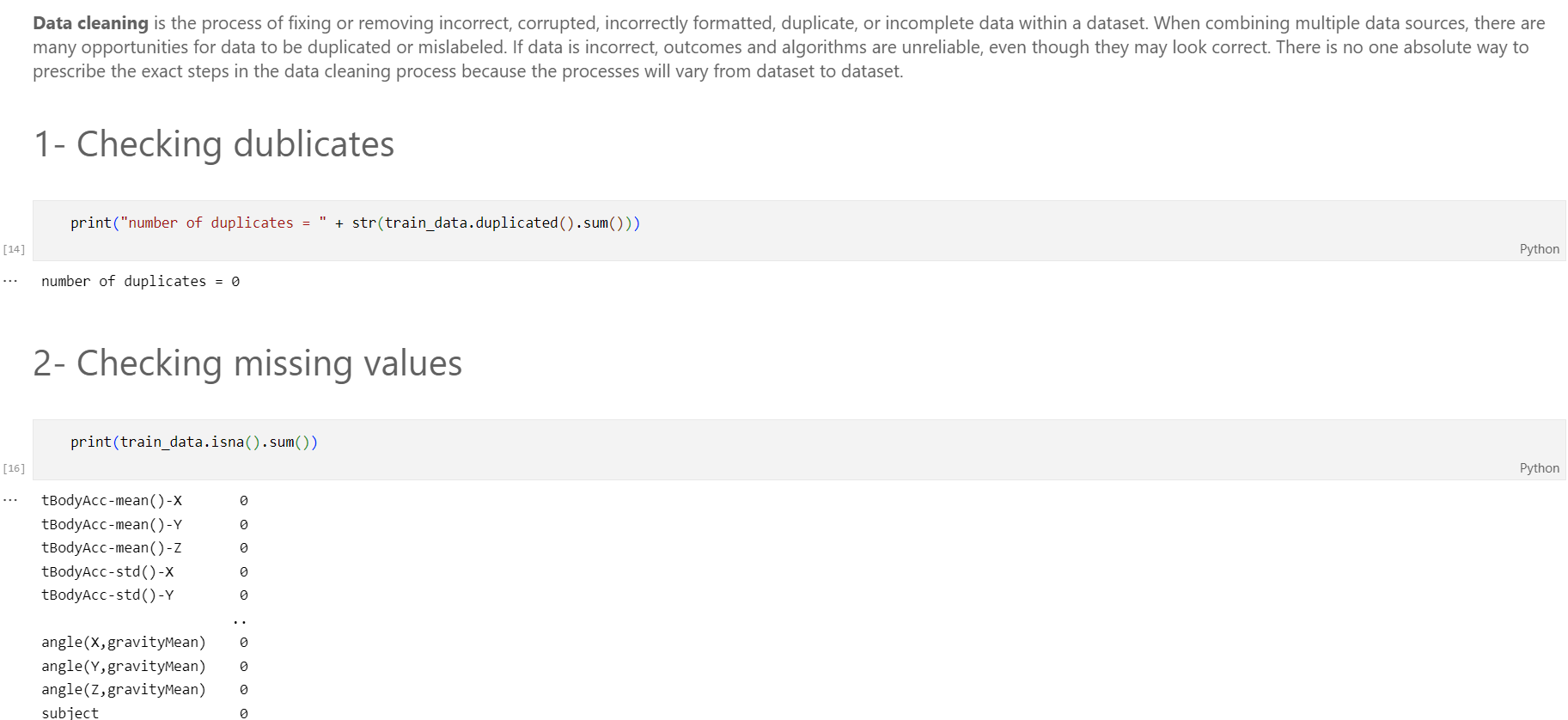
**Dataset description:** The experiments have been carried out with a group of 30 volunteers within an age bracket of 19-48 years. Each person performed six activities (WALKING, WALKINGUPSTAIRS, WALKINGDOWNSTAIRS, SITTING, STANDING, LAYING) wearing a smartphone on the waist. Using its embedded accelerometer and gyroscope, we captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz. The experiments have been video-recorded to label the data manually. The obtained dataset has been randomly partitioned into two sets, where 70% of the volunteers was selected for generating the training data and 30% the test data.

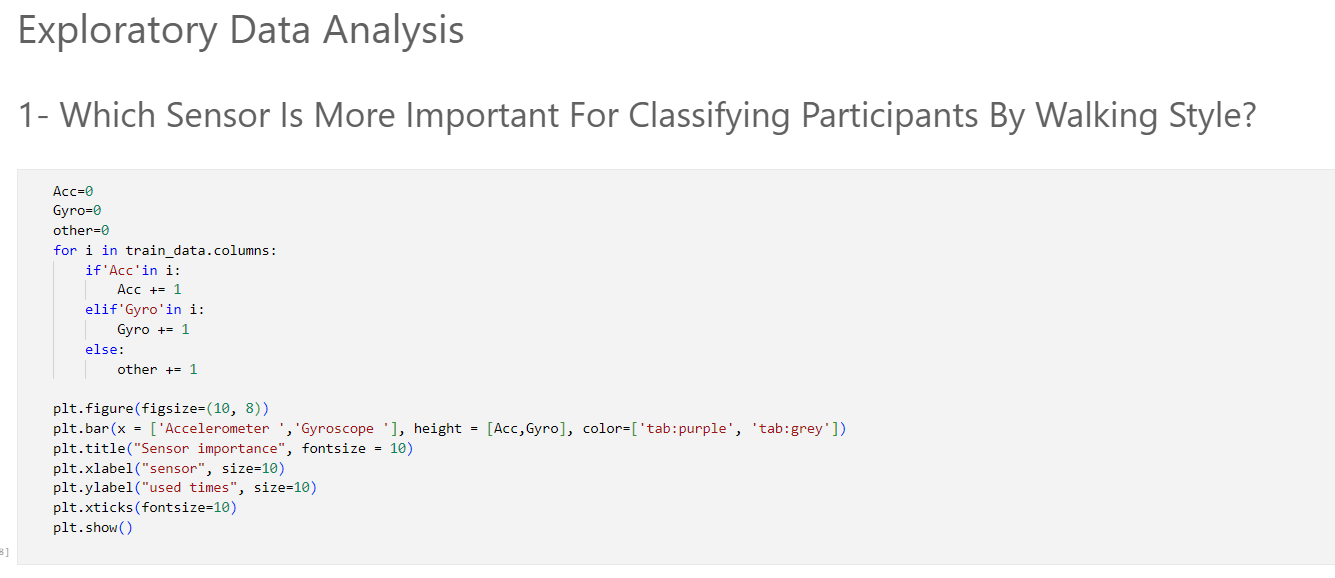
The sensor signals (accelerometer and gyroscope) were pre-processed by applying noise filters and then sampled in fixed-width sliding windows of 2.56 sec and 50% overlap (128 readings/window). The sensor acceleration signal, which has gravitational and body motion components, was separated using a Butterworth low-pass filter into body acceleration and gravity. The gravitational force is assumed to have only low frequency components, therefore a filter with 0.3 Hz cutoff frequency was used. From each window, a vector of features was obtained by calculating variables from the time and frequency domain.

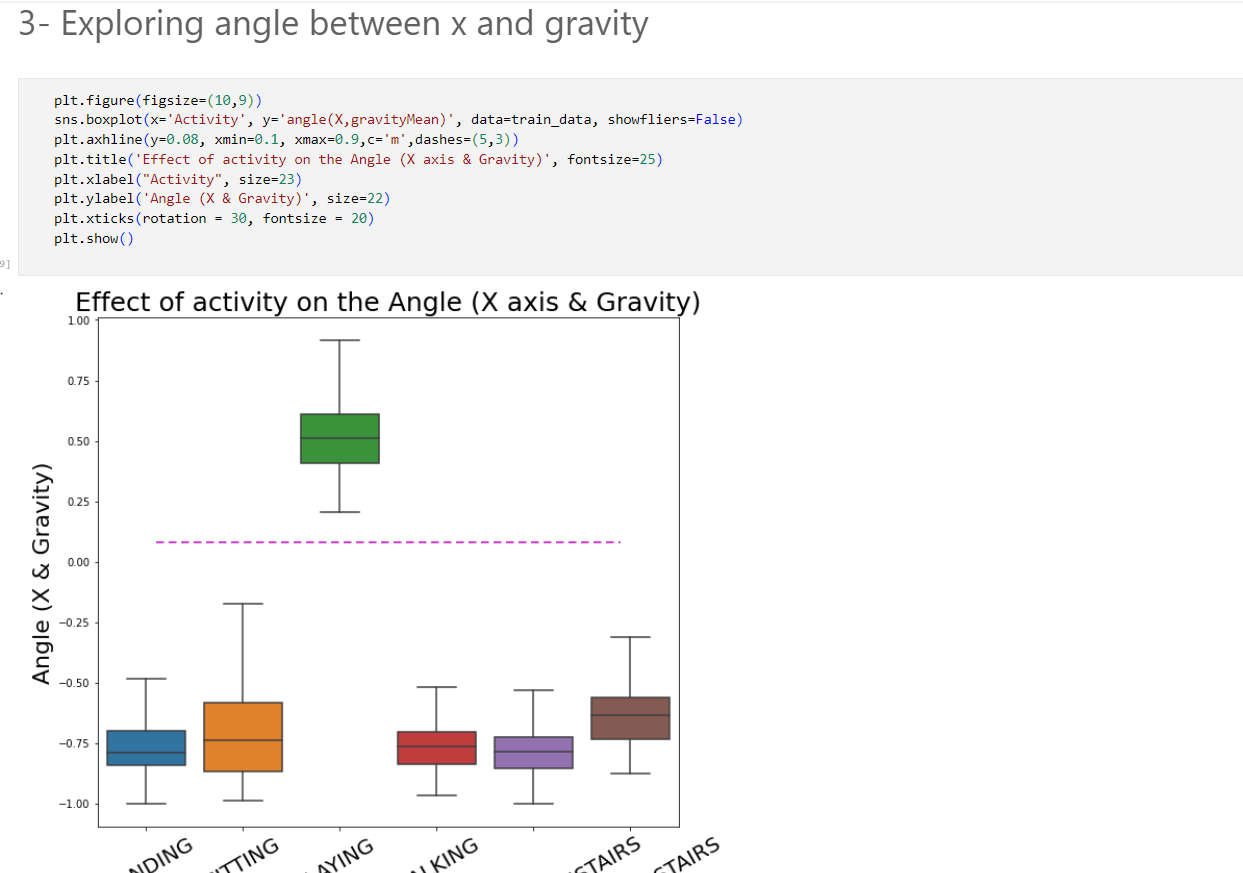
**Data Preprocessing and Exploratory data analysis:**

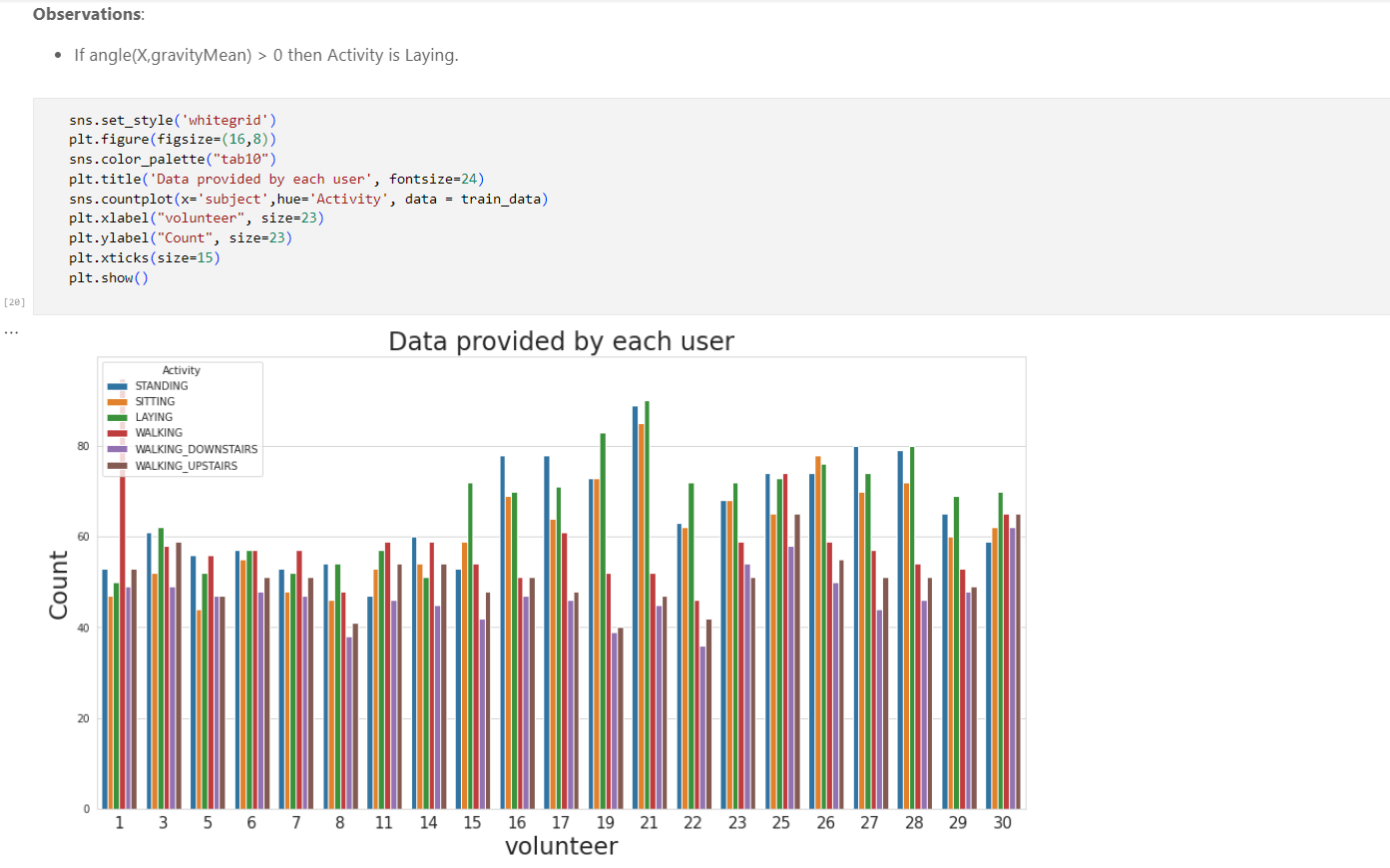




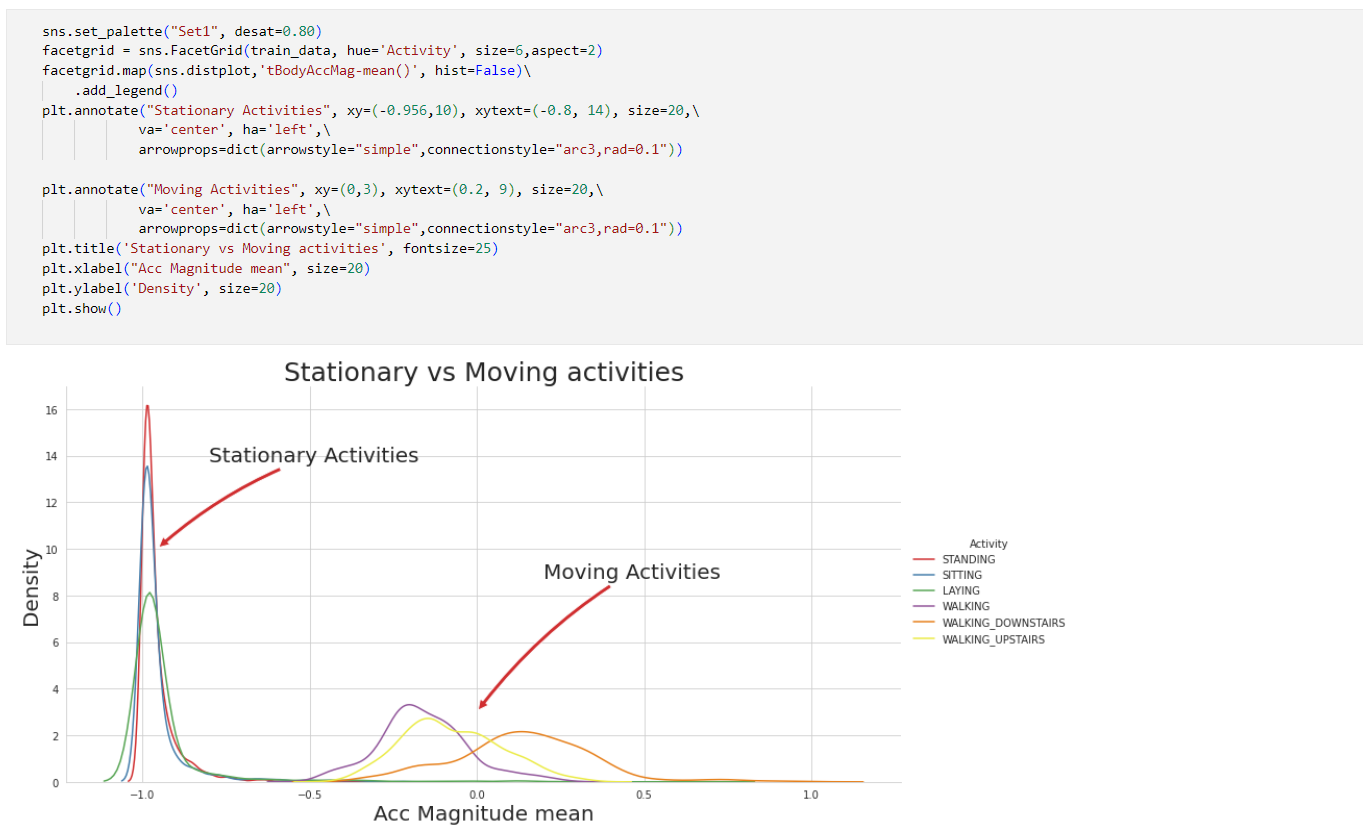






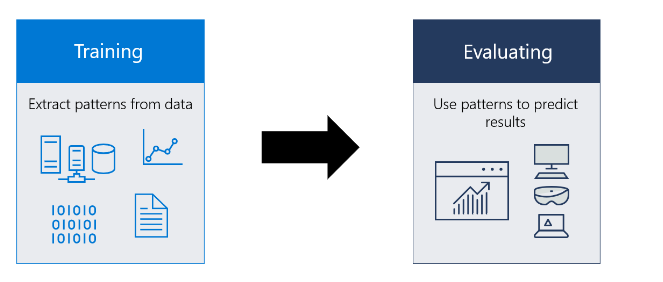


**Visualizing distribution of Stationary and Moving activities:**



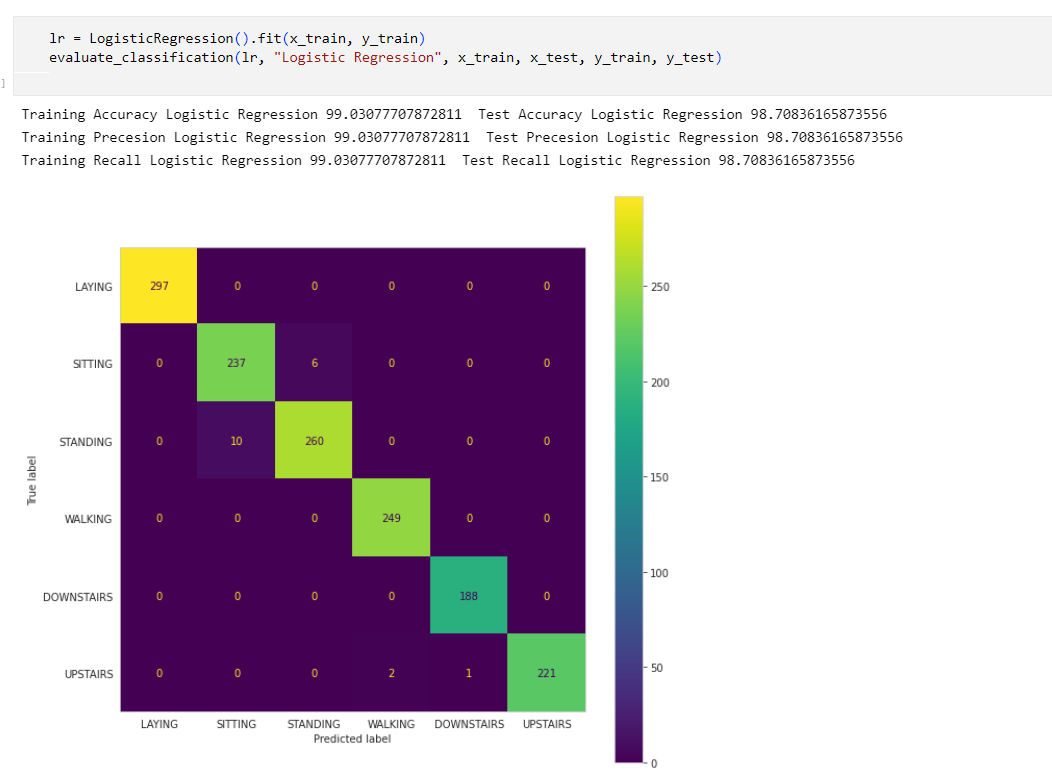
**Model Development:**

The process of modeling means training a machine learning algorithm to predict the labels from the features, tuning it for the business need, and validating it on holdout data. The output from modeling is a trained model that can be used for inference, making predictions on new data points.



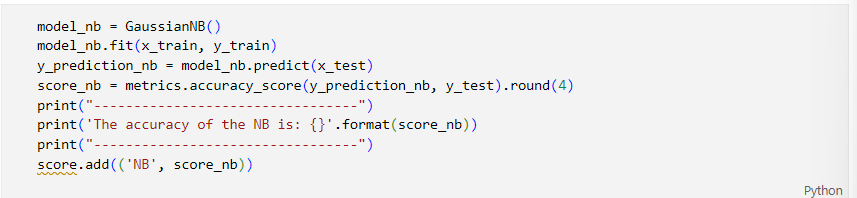
A machine learning model itself is a file that has been trained to recognize certain types of patterns. You train a model over a set of data, providing it an algorithm that it can use to reason over and learn from those data. Once you have trained the model, you can use it to reason over data that it hasn't seen before, and make predictions about those data. For example, let's say you want to build an application that can recognize a user's emotions based on their facial expressions. You can train a model by providing it with images of faces that are each tagged with a certain emotioo. and then you can use that model in an application that can recognize any user's emotion.

**Logistic Regression:**



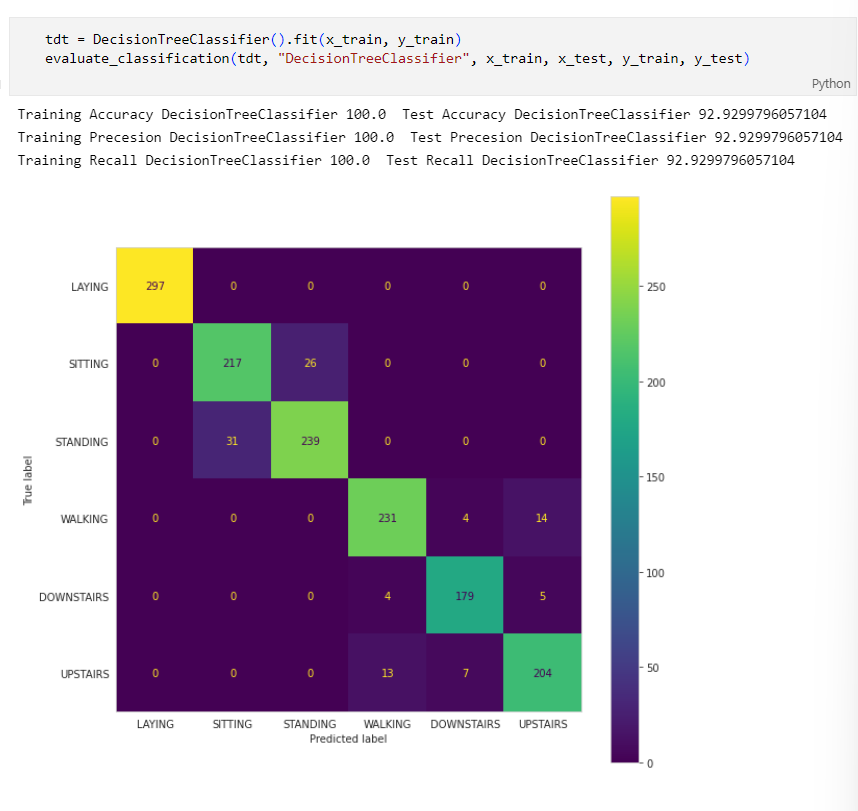
**Naïve Bayes:**

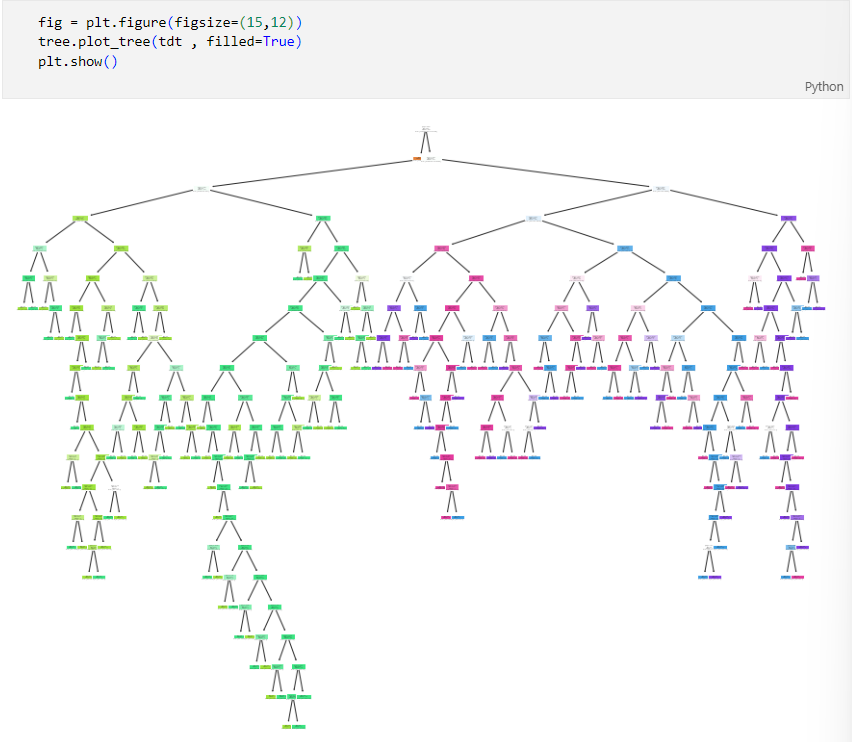
Naive Bayes models are a group of extremely fast and simple classification algorithms that are often suitable for very high-dimensional datasets. Because they are so fast and have so few tunable  
parameters, they end up being very useful as a quick-and-dirty baseline for a classification problem.



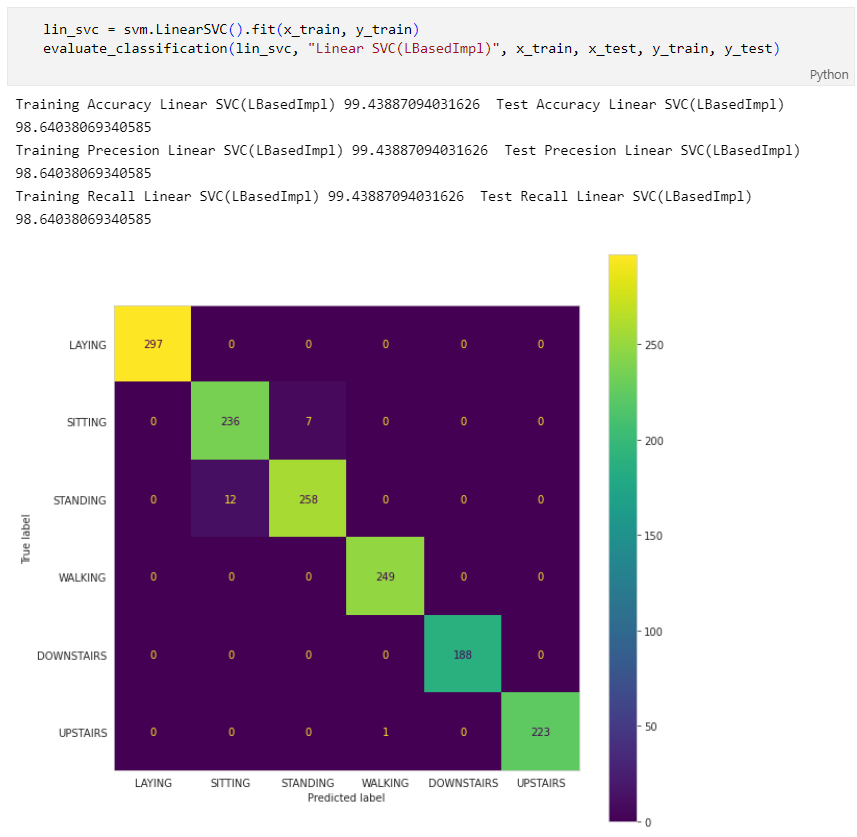
**Decision Tree:**

A decision tree is a flowchart-like tree structure where an internal node represents feature(or attribute), the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node. It learns to partition on the basis of the attribute value.





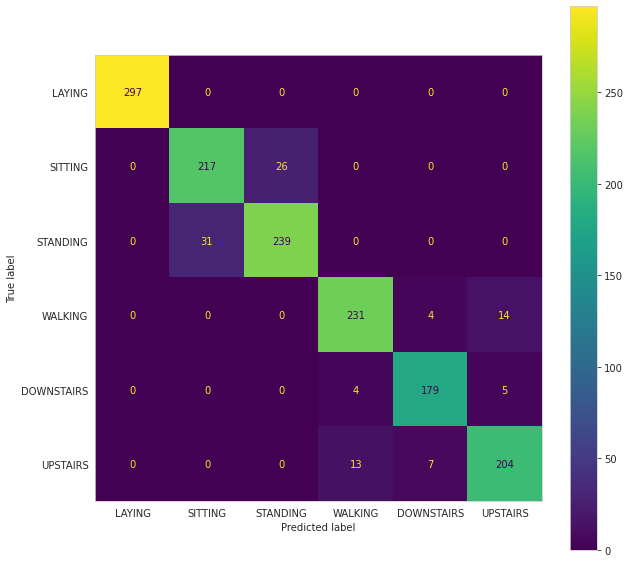
**SVM:**



**Discussion and conclusion:**

In Decision tree model --

1.Training Accuracy DecisionTreeCIassifier 100. e Test Accuracy DecisionTreeCIassifier 92.9299796057104. 2.Training Precesion DecisionTreeC1assifier 100.0 Test Precesion DecisionTreeC1assifier 92.9299796057104. 3.Training Recall DecisionTreeCIassifier 100.0 Test Recall DecisionTreeCIassifier 92.9299796057104

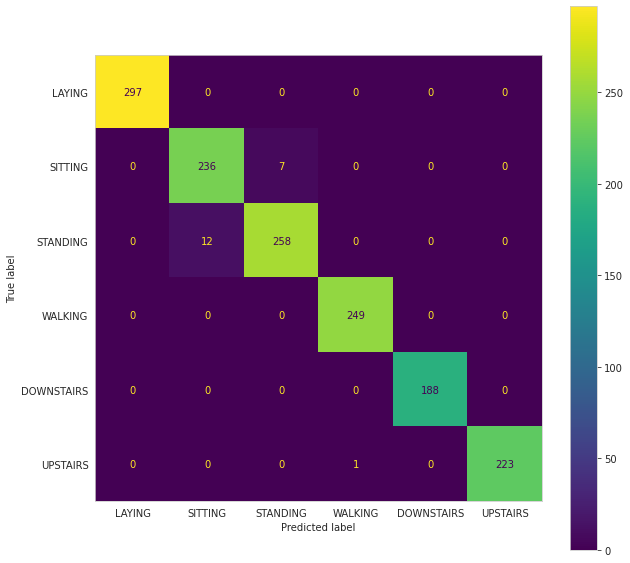


In SVM model

1.Training Accuracy Linear SVC(LBasedImpI) 99.43887094031626 Test Accuracy Linear SVC(LBasedImpI)98.64038069340585

2.Training Precesion Linear SVC( LBasedImpI) 99.43887094031626 Test Precesion Linear SVC(LBasedImpI) 98.64038069340585

3.Training Recall Linear SVC(LBasedImp1) 99.43887094031626 Test Recall Linear SVC(LBasedImp1) 98.64038069340585



In logistic regression model

1.Training Accuracy Logistic Regression 99.03077707872811 Test Accuracy Logistic Regression

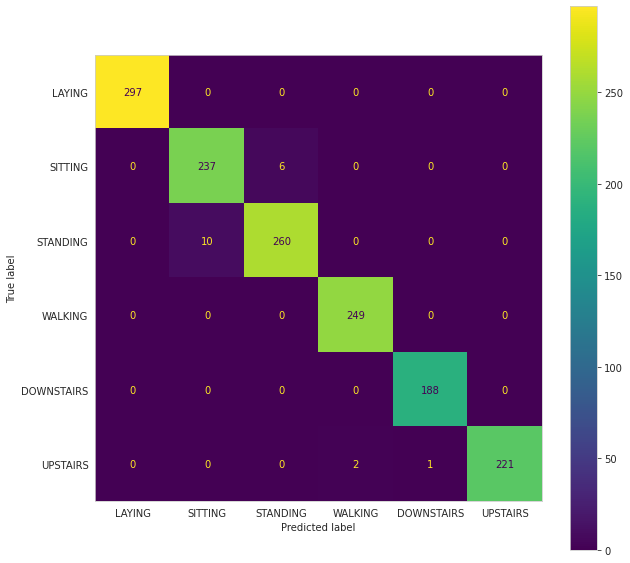
98.70836165873556

2.Training Precesion Logistic Regression 99.03077707872811 Test Precesion Logistic Regression

98.70836165873556

3.Training Recall Logistic Regression 99.03077707872811 Test Recall Logistic Regression

98.70836165873556



We find the best Test Accuracy in logistic regression and SVM model. The Test Accuracy all most 99%. Comparison of our system with existing classifiers using two standard datasets shows that our system is much superior in terms of the computational time, and either it surpasses or is on par with the existing recognition rates. It performs on par or marginally inferior to existing systems, when the number of training examples are a few due to the imbalance, although consistently better in terms of computation time.