

Applied machine learning

Abstract

We are a three-member team implementing 3 algorithms in Python using open source datasets from UCI & from Human Activity Recognition. We will evaluate our implementation of the algorithm and also compare those with the output of Weka as well. We draw conclusions from these comparison, provide pitfalls of our implementation, performance of the algorithm, and ways to improve our implementation and also the algorithm’s output.

# Group Members

Balaji Rajaram

Dipankar Biswas

Malabika Biswas

**Table of Contents**

# Dataset 1

***Dataset Name***: Human Activity Recognition

***Dataset Description***: Human Activity Recognition (HAR) - has emerged as a key research area in the last years and is gaining increasing attention by the pervasive computing research community, especially for the development of context-aware systems. There are many potential applications for HAR, like: elderly monitoring, life log systems for monitoring energy expenditure and for supporting weight-loss programs, and digital assistants for weight lifting exercises. This dataset has 5 classes (sitting-down, standing-up, standing, walking, and sitting) collected on 8 hours of activities of 4 healthy subjects [1].

Naïve Bayes Algorithm:

1) We first present our implementation of Naïve Bayes in Python.

**Pre-processing**:

The dataset initially had 159 columns of which the following columns were not useful for prediction (user\_name, raw\_timestamp\_part\_1, raw\_timestamp\_part\_2, cvtd\_timestamp, new\_window, num\_window) and hence we dropped those columns. Next, we identified columns that were having less than 5 percent of the actual record count. We dropped these columns as well as those columns will not help much in predicting the classes. Finally, we arrived at 53 columns including the target class.

**Learning & Understanding**:

The Naïve Bayes algorithm did not do well because of the correlation between features in this dataset. The algorithm works based on naive assumptions that all the variables are uncorrelated to each other, which is not true in this case.

**Python Implementation Results**:

Figure 1 shows the output of our python implementation.

