Machine Learning Classification Libraries for Postural Activity

Authors:

Albert Patrick Sankoh, sankoh.a@northeastern.edu
Shreyas Terdalkar, terdalkar.s@northeastern.edu
Sri Lakshmi Tirupathamma Manduri, manduri.s@northeastern.edu
Venkata Krishna Rao Chelamkuri, chelamkuri.v@northeastern.edu

Github repository:

https://github.com/kraoNEU/Machine-Learning-Classification-Libraries-for-Postural-Activity

Summary:

In today's modern lifestyle, it is paramount to have a good posture as it is an important factor to have healthy lumbar support and a good muscular and skeletal orientation. Similarly, older people require constant care and monitoring. In critical or unusual situations, they may require urgent medical attention. The following paper presents a multi-agent system for the care of elderly people living at home on their own, to prolong their independence. Our project is based on the dataset generated through this system, composed of seven sensors, tested on different people and in different sets of environments. We aim to design various ways to implement machine learning classification on the given dataset.

Paper:
http://ai.ijs.si/MitjaL/documents/Kaluza-An_agent-based_approach_to_care_in_independent_living-AmI-10.pdf

Dataset:
https://archive.ics.uci.edu/ml/datasets/Localization+Data+for+Person+Activity
In this project, we will be implementing 3 machine learning classification algorithms on a fixed data set. Our project, based on the data, will classify the dataset into different target variables such as different postures: sitting, standing, lying, walking, etc. It will also recommend which algorithm works best for a given postural data-set. Along with this, each algorithm will calculate the accuracy of its model.

Proposed Design:

Python Module: PosturalActivityClassification

Common Functions:

function_for_reading_dataset: This function is for reading the CSV file for the dataset.
function_for_eda_anlaysis: This Function is for getting the EDA analysis of the dataset.

External Libraries Used: numpy , sklearn , matplotlib, Seaborn and MatPlotLib and SciKit Learn (for Splitting the Train and Test Dataset)

1. Logistic Regression Classification
Class: logisticRegression
Methods: __init__() , fit() , predict() , sigmoidFunction() , accuracy() , plot()
Implementation: Logistic regression classification will be used to predict if the patient is sleeping, walking, sitting, running, etc. It will make use of logistic function (sigmoid) over the data to provide binary classification for each postural activity. The data will be fitted into the model using fit() method. This includes data processing

and feature segregation. The data will then be trained over to evaluate the model using sigmoidFunction() method. Once the model has been trained, the accuracy can be found using the accuracy() method.

2. Decision Tree Classification
Class: decisionTreeClassification
Implementation: The attributes of interest are mainly Tag,x,y,z. Here x,y,z are numerical attributes. But Tag being a string type is a categorical variable. Unfortunately, Sklearn Decision Trees do not handle categorical variables[challenge faced in this assignment]. But we can convert these features to numerical values and use pr-processing section to handle them. We will be using a train/test split on our decision tree. We will import train_test_split from sklearn.cross_validation. Now train_test_split will return 4 different parameters.

3. Bernoulli Naive Bayes Classification

Class: BernoulliNaivesBayesClassifier
Methods and Functions:
calculating_the_likelihood(): This is for calculating the likelihood of the class function,
calculating_the_posterior(): This is for calculating the posterior of the bayes classifier.
calculating_naives_bayesian_classifier(): This is for calculating the bayesian classifier from the posterior and the likelihood. This culminates in the entirety of the classification module.
Implementation: The Implementation will include the building of Naives Bayes Model using standard libraries (only NumPy and Pandas) and all the mathematical Models will be implemented and the classifier itself will be custom-built to suit the dataset which we are trying to employ our classifier-library for postural activity. All the Posterior Predictions, Bayes Estimation, Probability Hypothesis and the actual classification will be included in our implementation.

Challenges:
Naives Bayes Model seems to work well on the Discrete variables rather than continuous variables, therefore, our first job is to check the extent of the continuous variables in our dataset. Based on that we check whether to convert the continuous variables to discrete variables to better suit the algorithm which we are building. Secondly, we need to check whether multiple classes (for eg: our dataset contains 6-7 target classes) therefore, we need to check whether these would give any problem while we try to implement as we need to consider the bin sizes for multiple classes as well.