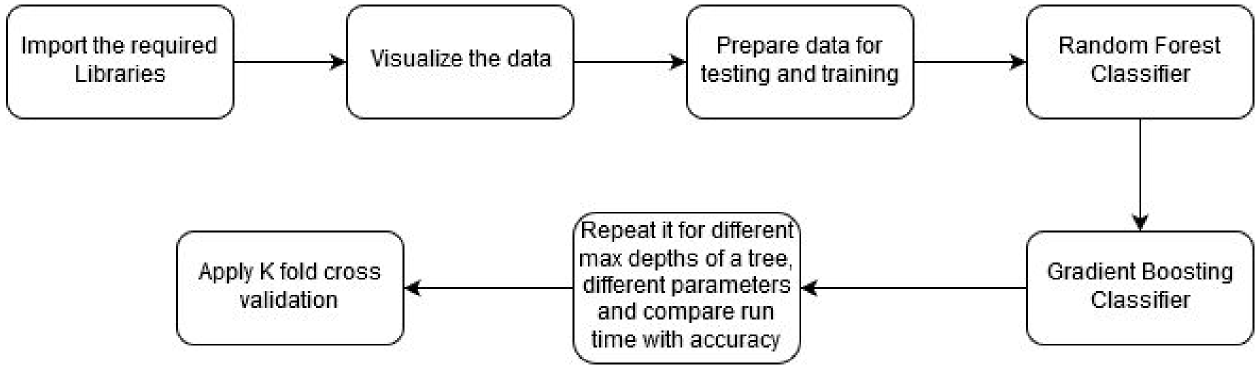
Human Activity Recognition

ABSTRACT

This project module classifies a dataset of 561 features into 6 human activities and compares accuracy of ensemble models with running time. 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. In UCI dataset, data is collected from systems for human physical activity recognition (HAR) using smartphone inertial sensors.

PROJECT FLOW DIAGRAM


Import the required Libraries
Visualize the data
Prepare data for testing and training
Random Forest Classifier
Gradient Boosting Classifier
Repeat it for different max depths of a tree, different parameters and compare run time with accuracy
Apply K fold cross validation

DATA DESCRIPTION

UCI HAR dataset was collected in an experiment that was performed on people of ages between 19 and 48. They were asked to wear smartphones on their waist and were asked to perform 6 activities - **WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, LAYING, STANDING, and SITTING**. Smartphone sensors on their waist collected tri-axial acceleration data of accelerometer and gyroscope. The estimated body acceleration and angular velocity were collected using the gyroscope. In total, for each person, it collected **561 features** including standard deviation, minimum, maximum of the data. This data is already normalized and bounded within [-1,1]. There are in total **10,299 instances** in the data.

FUNCTIONS AND LIBRARIES USED

The following functions were used in the project module:

1. **Timeit library**

This module times small bits of Python code. It avoids a number of common traps for measuring execution times.

2. **<u>Pandas</u>**

   Pandas library provides high-performance, easy-to-use data structures and data analysis tools for Python. Pandas is used for manipulation of data which is in CSV format.

3. **<u>Seaborn</u>**

   Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics**.**

4. **<u>Scikit-learn</u>**

   Scikit-learn is an open-source Python library that has powerful tools for data analysis and data mining. It is built on the following machine learning libraries:

   - **NumPy**, a library for manipulating multi-dimensional arrays and matrices. It also has an extensive compilation of mathematical functions for performing various calculations.
   - **SciPy**, an ecosystem consisting of various libraries for completing technical computing tasks.
   - **Matplotlib**, a library for plotting various charts and graphs.

   Scikit-learn offers an extensive range of built-in algorithms that make the most of data science projects.

   Here are the ways in which the Scikit-learn library is used in the project:

   - The **classification** tools identify the category associated with the provided data. Random forest and Gradient Boost classification algorithms are used in this project.
   - **Dimensionality reduction** lowers the number of random variables for analysis. For example, to increase the efficiency of visualizations, outlying data may not be considered. PCA is used in this project module.
   - **Model selection** algorithms offer tools to compare, validate, and select the best parameters and models to use in your data science projects. Cross-validation and metrics model selection modules are used that can deliver enhanced accuracy through parameter tuning.

FEATURES OF THE PROJECT

Dataset analysis

Dataset analysis is performed in order to find valuable insights from the given dataset - number of samples present for testing and validation, whether the data is imbalanced with respect to its target classes, whether there are any null values, etc. The following characteristics of the data are presented:

1. **Number of samples per activity in train and test**
2. **Activity count per with respect to subjects**
3. **Train validation split ratio = 75:25**
   Train samples = 5514
   Validation samples = 1838
   Test samples = 2947

Dimensionality reduction

Principal Component Analysis

Principal Component Analysis is a dimensionality-reduction method that is used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set. 561 features are reduced to 120 features in order to easily explore and visualize algorithms without extraneous variables to process.

Ensemble models

An ensemble is just a collection of predictors which come together (e.g. mean of all predictions) to give a final prediction.

1. **Random forest classifier**

Random forest (RF) consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction. The basic principle of RF is that **when a large number of relatively uncorrelated models (trees) operate as a committee, they will outperform any of the individual constituent models**. Random forest is used for classification as the trees protect each other from their individual errors.

In scikit-learn, random forest classifier sub-samples the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement if bootstrap=True which is its default value. This is known as **"bagging."** Decision trees are very sensitive to the data they are trained on — small changes to the training set can result in significantly different tree

structures. Random forest takes advantage of this by allowing each individual tree to randomly sample from the dataset with replacement, resulting in different trees.
Random forest is faster to train than decision trees because we are working only on a subset of features in this model, so we can easily work with hundreds of features. Prediction speed is significantly faster than training speed because we can save generated forests for future uses.

2. **Gradient Boost Classifier**

Boosting is an ensemble technique in which the predictors are not made independently (unlike bagging), but sequentially. This technique employs the logic in which t**he subsequent predictors learn from the mistakes of the previous predictors**. Therefore, the observations have an unequal probability of appearing in subsequent models and ones with the highest error appear most. (So the observations are not chosen based on the bootstrap process, but based on the error).

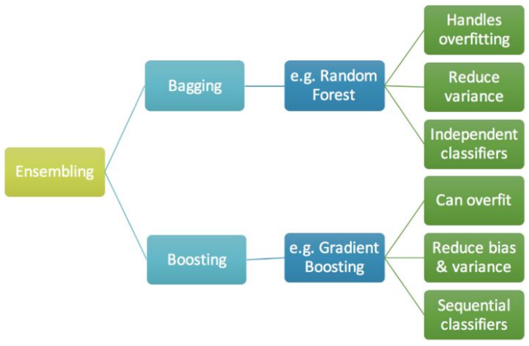


Fig 1. Ensembling

Comparison of different parameters of RFC and GBC with running time and accuracy

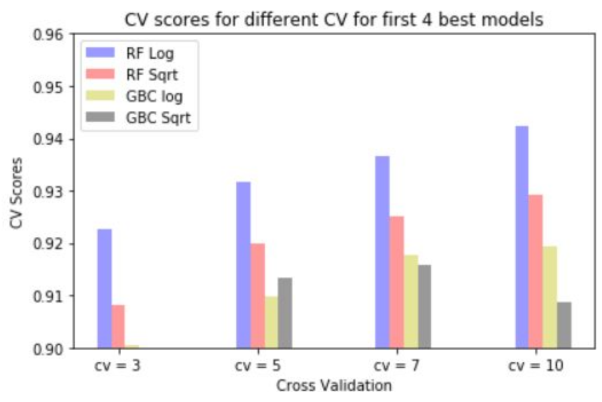
1. **Number of estimators** (Number of trees in the forest)
2. **Maximum depth** (Maximum depth of the tree)
3. **Max features** (Number of features to consider when looking for the best split)

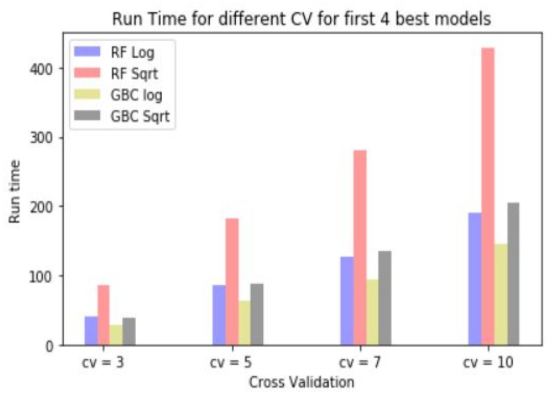
Accuracy measures for different cross validation folds on best 4 ensemble models

- Analysis of accuracy with cross-validation folds = [3,5,7,10]
- Run time of the models
- Mean accuracy of each model on 4 CV scores
- Comparison between accuracy achieved using dimension reduction and without using it.

RESULTS

The maximum accuracy of 0.976775956284153 was achieved by **random forest classifier with log2 feature split on CV score of 10** with run-time 18.949600689 s.





SCOPE/LIMITATIONS

Classification models such as Support vector machines (SVMs), Nearest neighbors, Artificial neural networks, etc. can be experimented and compared with ensemble models in order to achieve better accuracy. To compare between models, instead of statically instantiating parameters, grid search can be used to fine tune parameters dynamically.

CONCLUSION

The model employed for the classification of smartphone inertial data showed a recognition performance similar to previous work that has used special purpose sensors, therefore strengthening the application of these devices for HAR purposes.
The classification performance for each class is also shown in terms of recall and precision measures, with the sitting activity having the lowest recall equal to 90.5%. In particular, there is a noticeable misclassification overlap between this activity and standing attributed to the physical location of the device and its difficulty to categorize them.