JACOBS UNIVERSITY

DATA MINING

Project Report

Bremen Big Data Challenge - 2019

By
Sowrabha Ravishankar
Vijaya Nawale

May 17, 2020

Executive Summary

It is an approach to learn on the supervised classification task that is presented by the Bremen Big data Challenge. Data set consists of various sensors which are placed on the legs of a human. The purpose of getting to know the model is to identify accurately the 22 distinct labels for human activities. The task is then to classify 22 different moves and then to build a machine learning algorithm which gives better prediction. In this paper, we have implemented Feature Extraction as a preprocessing method in this project. Features like mean, standard deviation, variances were extracted, but we have used standard deviation features for the implementation. After applying a feature extraction approach to all 6401 subject files, matrix is obtained. After the extraction phase, a modeling procedure of random forest is applied to obtain prediction accuracy results of 95% . Results and other conclusions can be seen in our report along with our submission files.

Table of Contents

1. Introduction

"The Bremen Big Data Challenge" is a competition organized annually by the University of Bremen in Bremen, Germany. This is a project in which a data set is given by the Bremen Big data team. With technical advancement, developments, a number of multi-sensor networks of electronic and laptop devices have developed low-sized, low-cost, and unnecessary computing capacity. HAR does have a variety of applications, from unusual behavior identification in security surveillance and tracking in tele-rehabilitation to hand gesture recognition in augmented-reality and virtual-reality, and so on. One type of HAR is based on motion sensor records from which the HAR system tries to deduce patterns of effort for Classification and Prediction[10]. In this paper models were trained with the use of algorithms in machine learning to infer various actions performed. Various sensors, such as gyroscopes, accelerometers and so on, were mounted on the leg, and data were gathered for each individual when performing a variety of actions. A variety of basic and numerical plots have been used to identify the data using python visualization libraries. A number of features were later extracted from this sensor data, which was used later to train machine learning models. A clear explanation and results obtained from various models have been mentioned in the report with conclusive remarks[10].

2. Background of the data

The accumulated facts are based entirely on everyday athletic movements given by "The Bremen Big Data Challenge 2019" organisers[1] . The task is to classify the specific kinds of activities for every day and sports actions [1]. The data which are recorded from the wearable sensors and are positioned one above the knee and one under the knee (See Figure 2.1). The data set incorporates information recorded from 19 individuals. Out of 19, 15 of the complete subject's data is used for training motive and the remaining four subject's information is the testing set of data. The dataset is publicly on hand online on the social website.

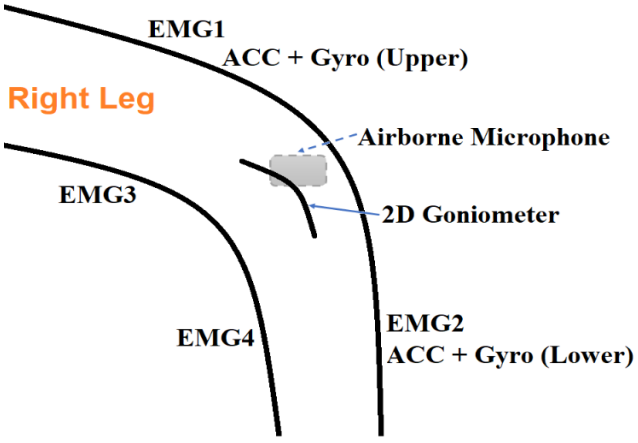

Figure 1: Placement of Sensors[1]

3. Data Format and Description

In the Data there are 22 movements:

- “Race ('run')
- Walking ('walk')
- Standing (standing)
- Sitting ('sit')
- Get up and sit down ('sit-to-stand', 'stand-to-sit')
- Up and down stairs ('stair-up', 'stair-down')
- Jump on one or both legs ('jump-one-leg', 'jump-two-leg')
- Run left or right ('curve-left-step', 'curve-right-step')
- Turn left or right on the spot, left or right foot Rfirst ('curve-left-spin-Lfirst', 'curve-left-spin-Rfirst’ Rfirst', 'curve-right-spin-Lfirst', 'curve-right- spin-Rfirst ')
- Lateral steps to the left or right ('lateral-shuffle-left', 'lateral-shuffle-right')
- Change of direction when running to the right or left, left or right foot first ('v-cut-left-left','v-cut-left-right', 'v-cut-right-left', 'v-cut' right-Rfirst ')[1]”

The records or the data which are available are in the form of CSV archives and they are divided/split into training and testing data sets. The training and testing data set (train.csv & Challenge.csv) has 3 columns specifically Subject, Data file, and Label.
Table 1 illustrates the data of the record in a dataset file.

SL. No	Subject	Datafile	Label
0	Subject04	Subject04/Subject04_Aufnahme000.csv	curve-left-step
1	Subject04	Subject04/Subject04_Aufnahme001.csv	curve-left-step
2	Subject04	Subject04/Subject04_Aufnahme002.csv	stand-to-sit
3	Subject04	Subject04/Subject04_Aufnahme003.csv	curve-right-spin-Rfirst

Table 1: Train.csv dataset

In the same way, the testing data set (file name - Challenge.csv) includes three columns and has the same meaning which is described in the training data set. The label column incorporates the letter X which shows that the labels are no longer classified. The value of X ought to be replaced

with the estimated labels amongst the above 22 specific movements. The datafile consists of a complete of 1739 lines counting both the function names and feature values.

As mentioned above, both train and test files consist of Subject, Datafile and Label. Subject denotes the ID of the subject, Datafile has all the sensor recordings (human movement) and Label denotes the recorded motion.

There are 19 columns in the file in which every column represents the values of the sensor data which are measured at one point in time (scanned at 1000 Hz). The columns represent each and every sensor and its positions as shown in Figure 1.

The datafiles are in CSV files. Basically, the file has a set of numbered-values (five digit) that appears like the following ones: 32688, 32224, ……..34736…..and so on[1].

3.1 Cross Validation

Cross-validation is one of the most commonly used methods of resampling statistics to estimate the right prediction error and to modify parameters of the model. Cross-validation is a method of re-sampling records to test predictive generalization ability and to forestall overfitting[1, 2].Consider a data set D consisting of n instances (or instances) marked out, xi, i= 1 ... n. Every case is defined by a set of attributes (or functionalities). We assume that each xi case belongs exactly to one yi-class. Cross-validation is similar to the repeated random subsampling process, except the sampling is performed in such a manner that does not overlap any two test sets. The available learning of sets in k-fold cross-validation is split into k disjoint subsets of about equal size. "Fold" refers to the wide variety of subsets which follow.[11].

For example, the cross-validated precision is the popular of all ten accuracies performed on the validation sets. Generally, let's −k denote dummy until trained on all but the learning set's kth subset. The < yi = f−k(xi) value is the estimated value for the actual type label, yi, of case xi, which is an element of the kth subset[11].

4. Data Preprocessing

4.1 Feature Extraction

Some of the features can be inherently used. However, there is a need of derived features many times for solving any problem with accurate results. Considering our data set - that every activity is associated with a file containing numerous sensor recording parameters and it would be inconvenient for the identical signals to be definitely indistinguishable even for a subject which

is equal. This is a good justification to use the extraction feature on this collection of data. This is an important explanation why feature extraction is performed on our data set. Our aim here is to combine raw records with a single sequence of functions that corresponds to each undertaking using the relevant extraction methods for features. We have extracted features like mean, standard deviation and variance for all the 19 csv files and investigated on ML algorithms like SVM and random forest[12].

Following are steps carried out in our project.

1. Considering all .csv files (each file contains a class movement) from subject folders and extracting mean/standard deviation/variance from it. In our project, we have extracted standard deviation from all the files and have generated one single file for feature extraction.
2. After extracting standard deviation features for each subject in the training and testing sample, we combine all these features into one data-frame.

5. Data Exploration

Data exploration is usually performed with the combination of automated and guide activities. Automated tasks may involve profiling documents or analysis of information or tabular evaluations to give the analyst a detailed view of the information and an overview of key features. Through this step we are aiming at making visualization a core section of data exploration and capture [1].

We have used Matplot and seaborn libraries to visualize our dataset in this project. Matplotlib is a Python programming language plotting library, and its numerical arithmetic extension NumPy. There is additionally a procedural "pylab" interface primarily based on a country desktop, designed to carefully resemble that of MATLAB, even though its use is discouraged.[3]

Seaborn is a library in Python, based on matplotlib closely integrated with data structures for pandas. Seaborn ambitions to turn visualization into a central phase of data exploration and grasping[6].

Below table shows list of unique labels along the number of occurrences.

```
curve-left-spin-Rfirst     320
curve-right-spin-Lfirst    301
curve-right-spin-Rfirst    300
run                        300
walk                       300
v-cut-left-Lfirst          300
v-cut-right-Lfirst         300
curve-left-step            299
curve-right-step           295
stand-to-sit               289
sit                        289
sit-to-stand               289
stand                      289
lateral-shuffle-left       282
v-cut-left-Rfirst          280
curve-left-spin-Lfirst     280
jump-two-leg               280
v-cut-right-Rfirst         280
jump-one-leg               279
stair-down                 278
stair-up                   278
lateral-shuffle-right      277
lay                         16
Name: Label, dtype: int64
```

Table 2. List of unique labels used.

It depicts the movements along with the count of each movement in the train.csv file. This table shows that movements have been almost equally distributed except for the lay movement. Below Bar Chart shows graphical representation.

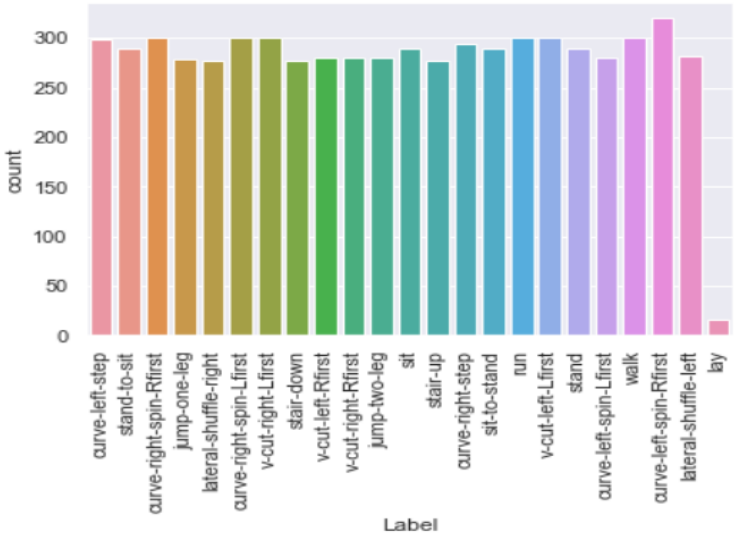

Figure 2. Bar chart shows graphical representation of unique labels used.

A correlation matrix shows correlation coefficients between variables. Each cell in the below image indicates the correlation between two sensor variables which summarizes the data, as an input into an extra superior analysis.

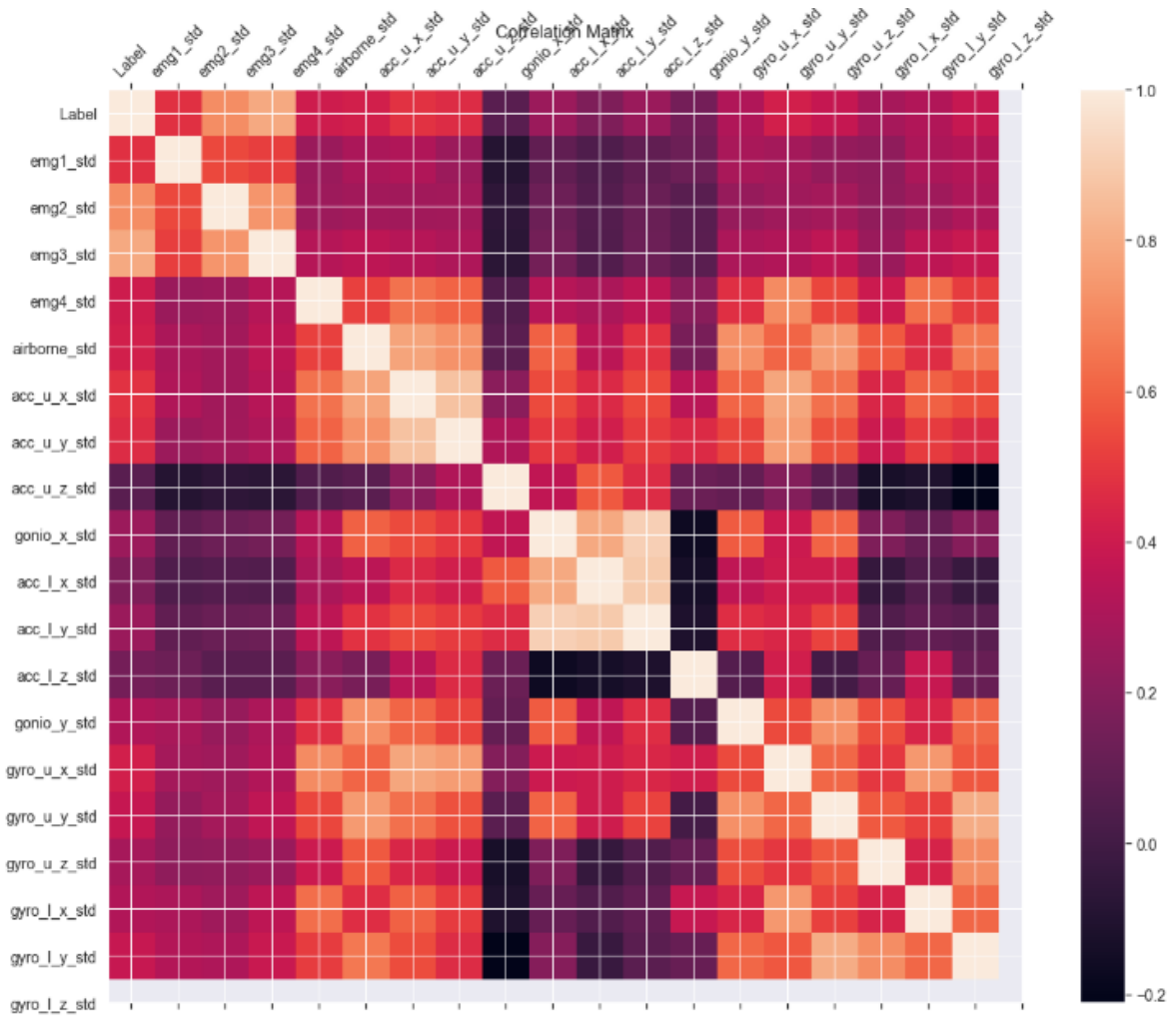

Figure 3. Correlation between two sensors

A histogram is the most regularly used diagram to show distributions. In this sensor data set, histogram will graphically summarize the sensor information and the frequency of using each

sensor.

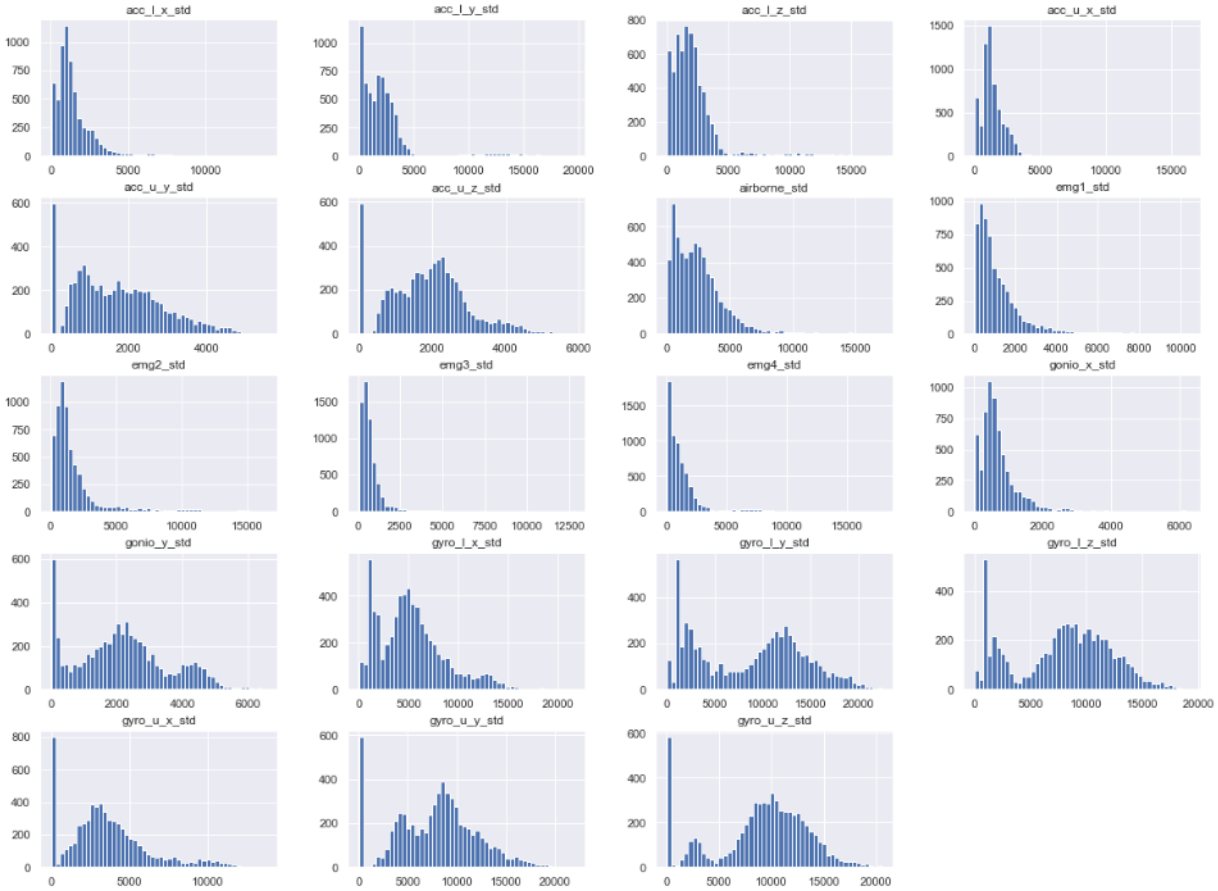

Figure 4. Histogram representation - frequency of using each sensor

6. Data Analysis

High-dimensional datasets can be incredibly difficult to imagine. To resource visualization of a dataset's shape, the dimension has to be diminished in some way. The easiest way to achieve this Dimensionality Reduction is by taking a random data projection. In a random projection, the extra interesting structure within the data is likely to get lost [7].

6.1 Support Vector Machine used for classification

sklearn.svm – library used for Support Vector Machine (SVM) support both type of inputs dense and sparse. We have performed classification with the *SVC* and *LinearSVC.* Classifiers are developed, and each trains two class records. Decision function of SVM rely on train dataset's subsets that is nothing but the support vectors [7].
After applying this once model fits then it can be used for predictions that is on test dataset.

6.1.1 Advantages

Effective in over-dimensional spaces. Still enormous in cases the number of measurements in place is greater than the sample range. Uses a subset of decision-function training variables (called help vectors), and it's also successful reminiscence [7].

6.1.2 Disadvantages

If the variety of features is greater than the number of samples, it is important to stay away from over-fitting when deciding on the features of the kernel and the time of regularization. SVMs do not have probability estimates of chance, these are measured using a five-fold cross-validation.

In Platt scaling, the worried cross-validation is a high-priced process for giant datasets so that predictions may also be incompatible with the scores. For classification Random Forest offers the chance to belong to the class and SVM gives the distance to the boundary. Random Forest works well with a combination of numerical and basic characteristics. It's perfect when elements are on a variety of scales, too. Roughly, statistics can be used with Random Forest as they are [7].

6.2 Random Forest

Random forests are a kind of ensemble approach that makes predictions by using the predictions of various unbiased base models on average. Implementation:

6.2.1 Preprocessing

Exchange the raw function vectors into a representation more appropriate for downstream estimators.Dataset standardisation is a common prerequisite for many system mastering estimators.Standard Scaler measure a training set's *mean and standard deviation* in order to be able to reapply the equivalent transformation on the test set later. The scaler instance can then be used on new information to seriously change the training set [7].

6.2.2 Model_selection

Learning the parameters of a prediction function and testing it on the same data is a methodological error: a model that will simply replicate the sample labels that it has already considered would have a perfect rating but would not predict anything useful on the data that has yet to be seen. That state is known as overfitting. In scikit-learn, the train test split helper function can easily compute a random break up into training and take a look at units.

There is still a chance of overfitting on the check set when evaluating extraordinary settings for estimators, such as the C setting that should be set manually for an SVM, knowledge about the check set may "leak" on the performance of generalization into the model and contrast metrics no longer document. To explain this issue, validation dataset can be considered: after training_set, evaluation is carried out on the validation_set, and when the experiment appears to be successful, the closing evaluation can be completed on test_set. By dividing the reachable data into sets can be used and the results may depend on an accurate random preference for the pair of sets.

When doing **cross-validation** (CV, in short), there is no more requirement of validation set. The training set is divided into **k = 5** smaller sets, called k-fold CV. For each of the k "folds," the procedure is as followed:

The use of ***k – 1*** folds as training data is trained by a model. The resulting model is validated on the ultimate part of the statistics. To find the parameters that function well for the optimization of challenges, we perform *Grid Search* by combining the above-mentioned alternatives and tuning them in the same variety as in our first study. In the Grid search technique Hyper-parameters are parameters not known in estimators immediately and determine the most reliable values, they are passed on to the estimator object constructor as arguments. The complete model's overall performance is based on the specified hyper parameter values.

cross_val_score - Estimate the accuracy of the data set by dividing the data, becoming a model and calculating the score 5 instances in a row [7].

6.2.3 Random Forest Classifier

A random forest is a meta-estimator that matches a number of choice tree classifiers on different dataset sub-samples and uses averaging to improve predictive accuracy and over-fitting power. The sub-sample dimension is constantly the same as the calculation of the authentic input sequence, but the samples are drawn with substitution *if bootstrap = True* [7].

The unpruned trees which on some data sets will probably be quite large. The elements at each split are always permuted at random. The consistency found break up can therefore also differ, even with the same training data, if the criterion improvement is the same for countless splits, random_state has to be set to achieve a deterministic behavior at some point of fitting.

feature_importance
The higher the function the more significant. The significance of a function is measured as the entire discount of the criterion that is applied via that feature. This is Gini importance. The envisaged form of enter pattern is a ballot through the trees in the forest, weighted by their estimates of probability. That is, the category predicted is the one with the best possible estimate of mean chance across all trees [7].

7. Results and Conclusion

$Train_{\ Score}$	$Test_{\ Score}$	$Challenge_{\ Score}$
97%	95%	65%

Table 3: Results

7.1 Confusion matrix

We can infer from Figure 5 below that most misclassification occurs with the classes stand and sit, which are intuitively similar. The pronounced averages include macro average, weighted average and average sample. Calculated classification accuracy with confusion_matrix with each row corresponding to the class proper. The function Classification report builds a text report displaying the most important classification metrics.

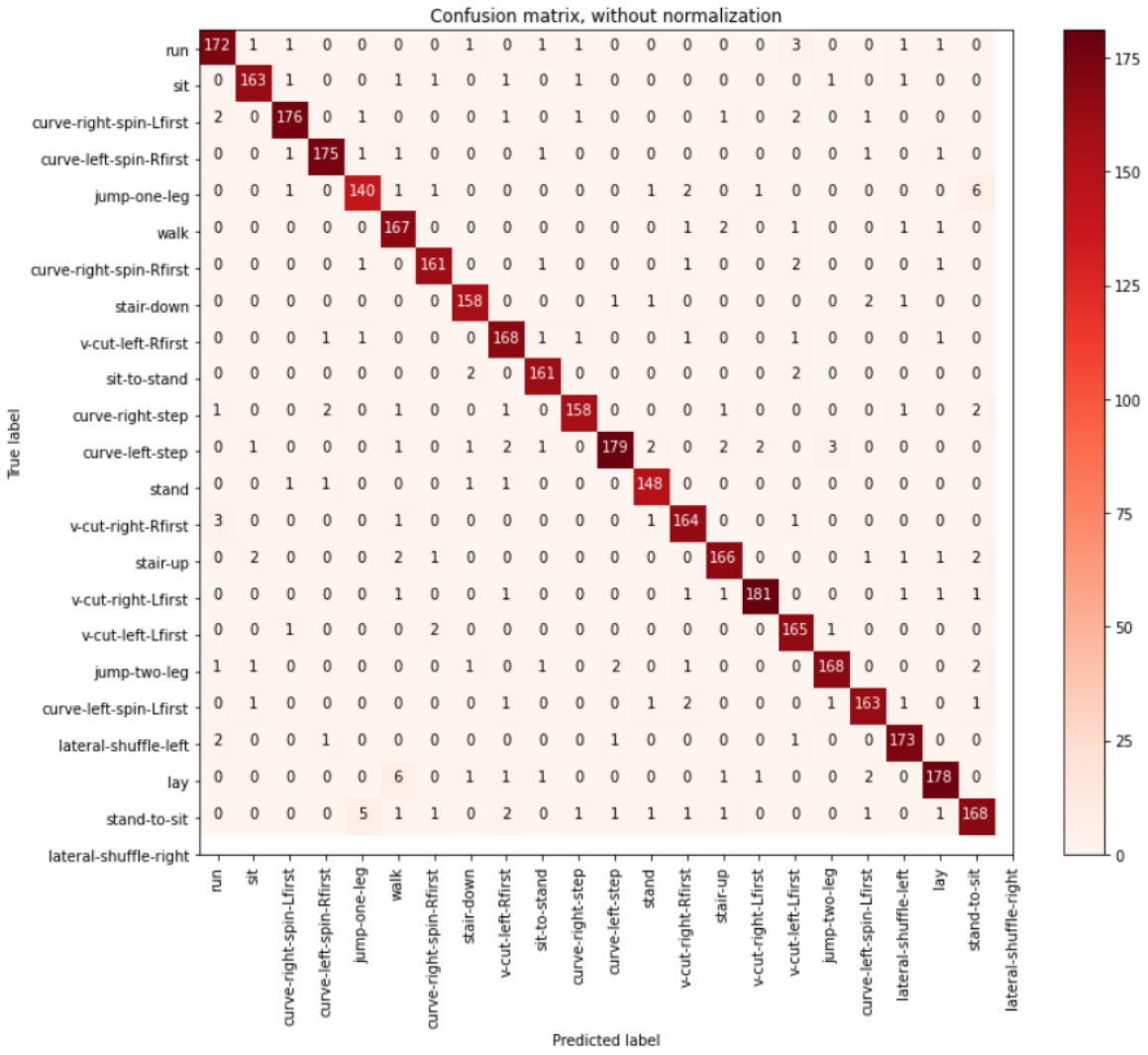

Figure.5 Confusion matrix on test data

CV_{train}	CV_{test}	$Challenge_{Score}$
	77%	64%

Table 3: Results with cross_validation

Appendix A Code Repository

All the code implemented in this report and also the output files are available at the Dropbox - https://www.dropbox.com/s/h5zyagtjz6lzbyv/DataMining_Spring2020_ProjectReport_Vijaya%26Sowrabha.zip?dl=0

References

[1] https://bbdc.csl.uni-bremen.de/index.php/2019h/28-aufgabenstellung-2019

[2] http://minds.jacobs-university.de/uploads/teaching/lectureNotes/LN ML4IMS.pdf

[3]https://www.semanticscholar.org/paper/Feature-Analysis-to-Human-Activity-Recognition-Suto-Oniga/bccfa23a493285026d4bd82aac18c8b62109e11c

[4] https://deepai.org/machine-learning-glossary-and-terms/feature-extraction

[5] Wikipedia, Matplotlib, 18 March 2020, URL: https://en.wikipedia.org/wiki/Matplotlib

[6] https://seaborn.pydata.org/introduction.html

[7] The Scikit-Learn community. May. 2019. Url: https://scikit-learn.org/stable/modules/classes.html#

[8] https://matplotlib.org/

[9] https://seaborn.pydata.org/

[10] Simon Fong et al. “Training classifiers with shadow features for sensor-based human activity recognition”.

[11] https://www.researchgate.net/publication/324701535_Cross-Validation

[12] O. D. Lara and M. A. Labrador. “A Survey on Human Activity Recognition using Wearable Sensors”. In: IEEE Communications Surveys Tutorials