DATA MINING

Project Report
on
Bremen Big Data Challenge
– Edition 2019

By Rohit Anand, Nithya Swaminathan, Sanyukta Suman
May 18, 2020

Table of Contents

Abstract

The report proposes a machine learning solution for the supervised classification task provided by the University of Bremen as The Bremen Big Data Challenge. The task is to predict the appropriate classification model for a given data for human activity recognition with a high degree of accuracy. The available data set is made up of signals detected by motion sensors that can be placed on the human subjects' leg. The training data is organized in such a way that each of its lines is, in fact, a .csv subject file composed of 19 columns (number of sensors) and Nfr rows, the [19 × Nfr] matrix, for a particular class of movement. The optimization method is applied to the data set. Due to the high features of the data set, the feature extraction method - Principal Component Analysis is performed. After the extraction phase, different types of Machine Learning Algorithm are used. Three types of Machine Learning methods are used to obtain test results of 80%, 78% and 75% accuracy. Finally, conclusions are presented in our report trying to find out the reason for the low model performance, opening the space for various approaches in the future.

Introduction

The "Bremen Big Data Challenge 2019" is an annual competition organized by the University of Bremen in Bremen, Germany. This is a competition in which the data set is presented, and the classification task is specified. In this 2019 edition, teams had to come up with a solution for classifying data. According to the data collected by different sensors placed on the legs of human subjects, groups should classify various movements of the legs. The team with the best Accuracy is considered the winner of the competition. This report describes structure of the data used to solve the suggested task in "The Bremen Big Data Challenge". First, we start by defining the data structure. Followed by, preprocessing of dataset reducing dimension using Principal Component Analysis and then classifying data using different machine learning approaches. Finally, to conclude the report result and insight are presented.

Data description

Provided by" The Bremen Big Data Challenge 2019" Organizers, the collected data are based on daily athletic movements. Using wearable sensors above and below the knee of the individual (athletic), a dataset consists of 19 individuals, mainly identified as subjects, has been recorded. Here, 15 out of the total number of subjects are used as the training dataset and the rest of the part of the total number are used as testing dataset.

Description and Format

The data comprise the following 22 movements:

Race	Run
Go	Walk
Standing	Stand
Sitting	Sit
Get up	Sit-to-stand
Sit Down	Stand-to-sit
Going up	Stair-up
Downstairs	Stair-down
Jump on one leg	Jump-one-leg
Jump on both legs	Jump-two-leg
Run left	Curve-left-step
Run right	Curve-right-step
Turn in place left	Curve-left-spin-Lfirst
Turn in place right	Curve-left-spin-Rfirst
Turn in left foot first	Curve-right-spin-Lfirst
Turn in right foot first	Curve-right-spin-Rfirst
Sideways step to the left	Lateral-shuffle-left
Sideways step to the right	Lateral-shuffle-right
Change of direction while running to the left with left foot first	v-cut-left-Lfirst
Change of direction while running to right with left foot first	v-cut-right-Lfirst
Change of direction to the left with right foot first	v-cut-left-Rfirst
Change of direction to the right with right foot first	v-cut-right-Rfirst

Table 1: The dataset contains 22 movements.

All the data is available as CSV files, or Comma-Separated Values, and is divided into training and test data, represented by in "train.csv" and "challenge.csv" files respectively. Starting with the training dataset file (train.csv), whose first line identifies the feature names followed by the feature values. This file contains a total of 6402 rows, including both feature names and feature values. Feature Names are Subjects, Database, Label, and feature names map with each feature Name. For example, the initial file values are: Subject02, Subject02 / Subject_ Aufnahme002. csv, stand-to-sit. Table 2 shows a version of the data partition for the training data file.

Subjects	Datafile	Label
Subject02	Subject02/Subject02_Aufnahme000.csv	curve-left-step
Subject02	Subject02/Subject02_Aufnahme001.csv	curve-left-step
Subject02	Subject02/Subject02_Aufnahme002.csv	stand-to-sit
...	….	….
Subject19	Subject19/Subject19_Aufnahme438.csv	curve-right-step
Subject19	Subject19/Subject19_Aufnahme439.csv	curve-right-spin-Rfirst

Table 2: Tabular visualization of the “train.csv” dataset.

Similarly, the testing dataset file (challenge.csv) is formatted using the same structure except for the Label column, which is unknown and marked with an X. The datafile contains a total of 1739 lines counting both the feature names and feature values. Table 3 displays a lightweight version of the data partition of the testing dataset file.

Subjects	Datafile	Label
Subject01	Subject01/Subject01_Aufnahme000.csv	X
Subject01	Subject01/Subject01_Aufnahme001.csv	X
Subject01	Subject01/Subject01_Aufnahme002.csv	X
….	….	…..
Subject15	Subject15/Subject15_Aufnahme438.csv	X
Subject15	Subject15/Subject15_Aufnahme439.csv	X

Table 3: Tabular visualization of the ”challenge.csv” dataset.

The columns have the following meanings:

1. Subject: The ID of the subject.
2. Datafile: Path of the file containing the sensor data for this recording. For each subject, there is a folder in which individual data files contain the sensor data for individual motion recordings.
3. Label: The movement that was recorded

Each data file contains 19 attributes and each column represents individual sensor data. A list of these attributes are as follows:

EMG1	Airborne	Goniometer X	Goniometer Y	Gyro lower X
EMG2	ACC upper X	ACC lower X	Gyro upper X	Gyro lower Y
EMG3	ACC upper Y	ACC lower Y	Gyro upper Y	Gyro lower Z
EMG4	ACC upper Z	ACC lower Z	Gyro upper Z	

Data Pre-processing

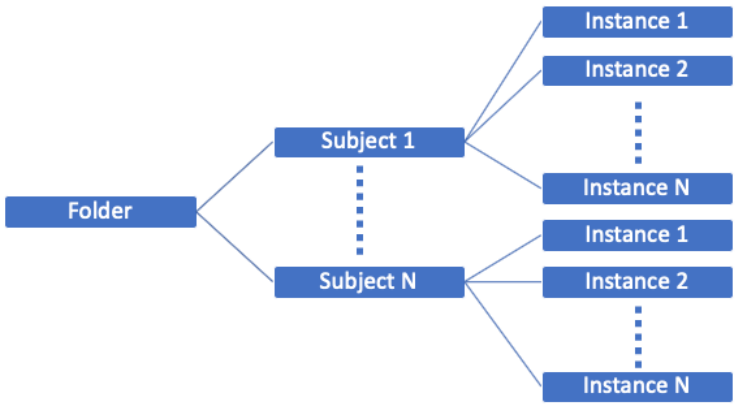

Figure 1: Directory structure of the Dataset

The dataset provided for the analysis and machine learning has the directory structure as presented above in *Figure 1* For each subject provided in the train set has N instances associated at different time stamps with each instance having a different predicted variable(label).
Source of the dataset - https://bbdc.csl.uni-bremen.de/index.php/2019h/25-bbdc-2019

The preprocessing step of the data involves –

1. Mapping the subjects from training set to each subject directory.
2. Iterating through all the instances for each subject.
3. For each instance per subject from the training set, map to the respective labels (Ground Truth)
4. Create a new label with the ground truth and concatenate it with individual data frame.
5. Append a list of data frames and in the final iteration, concatenate all the data frames into a main data frame.
6. Added an extra dimension named Subject for exploratory data analysis purpose which will be dropped before training the model.
7. Added column names (taken from the dataset info file) to the main data frame.

The shape of the main data frame –

- Number of records/rows: 16367293
- Dimension: 2

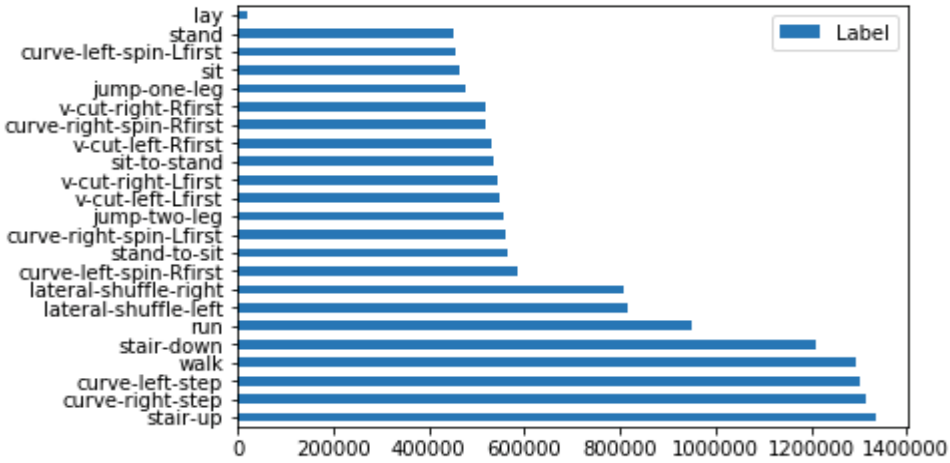

Figure 2: An overview of the labels/ground truth for respective data records

Histogram in *Figure 2* represents an overview of the labels/ground truth for respective data records. The most common label which was recorded was found to be **Stair-up.**

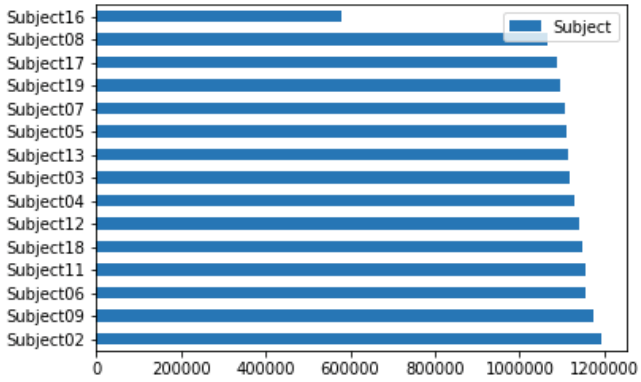

Figure 3: The number of records grouped by subjects in the training set.

The number of records grouped by subjects in the training set has been plotted in *Figure 3* *Figure 4* shows the datatype of each predictor variable.

```
Data columns (total 21 columns):
EMG1            int64
EMG2            int64
EMG3            int64
EMG4            int64
Airborne        int64
ACC upper X     int64
ACC upper Y     int64
ACC upper Z     int64
Goniometer X    int64
ACC lower X     int64
ACC lower Y     int64
ACC loewr Z     int64
Goniometer Y    int64
Gyro upper X    int64
Gyro upper Y    int64
Gyro upper Z    int64
Gyro lower X    int64
Gyro lower Y    int64
Gyro lower Z    int64
Label           object
Subject         object
```

Figure 4: The datatype of each predictor variable.

Kernel density estimation for one the variables EMG1 shows that the dominance of outliers in the dataset. *Figure 5* represents this distribution. The same pattern can be observed for other predictor variables as well. The red line depicts the interquartile range.

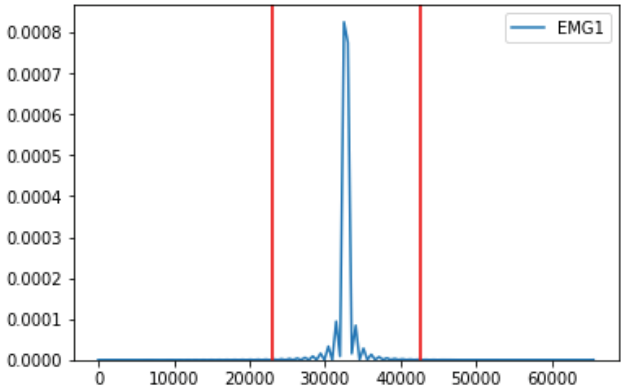

Figure 5: The dominance of outliers in the dataset.

Since the predicted variable was found to be a categorical value, it was converted to numerical value hence numbering the labels sequentially. This step was required at this stage because removing outliers(on categorical variables) wouldn't have meant appropriate for further data analysis.

Data Value Space for the predicted variable, Label = {1,2,3,4,5,6,7……,20,21,22}

Next step involves filtering data 1 Standard Deviation away from the mean so that we can consider more meaningful data. After this our sample space included –

- Number of records/rows: 1921598
- Dimensions: 20

Hence, we can notice a reduction of approximately 88% of the less meaningful records which could have been a result of wrong reading by the sensors.

At this point, we can notice most of the outliers being removed from the dataset in *Figure 6* and it mostly follows a Normal Distribution. This was noticed for all other predictor variables as well.

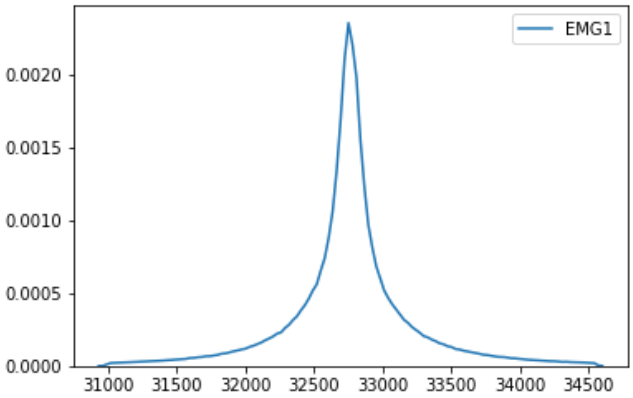

Figure 6: Normal Distribution of a dataset

Now, the Label, i.e., the ground truth variable was Encoded in such a way that one label has one column created and making sure to **avoid dummy variable trap** by removing one extra column.

As the dataset has continuous random variables with each having a varying range, it was considered necessary to scale them as it would reduce the running time of the Maximum Likelihood Estimation of Machine Learning algorithm by finding the optimal coefficients.

Using PCA to find the variables explaning the most variance caused the number of meaningful features to be reduced to 8 as seen in *Figure 7*

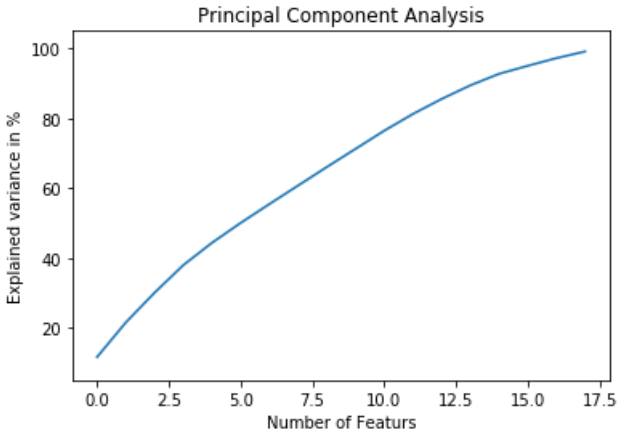

Figure 7: PCA for feature Extraction.

MACHINE LEARNING

K Nearest Neighbor approach

There are multinumber of features used to describe a label and it is nearly 323 features. The features are labelled and then the next task is to find the best model which fits and predicts the label [3]. Methods like Support vector machines seems to be computationally expensive and it is running for more than 7 hours. A supervised machine learning method that relies on labelled input data to learn a function and when given unlabelled data it produces an appropriate output. In our dataset KNN classifies vectors(1x323) according to the most common label among its K Nearest Neighbours. *Figure 8* shows the plots of accuracy and the K value. Because of the feature scaling and principal component Analysis implemented in data wrangling part KNN seems to be perform well in high dimension data.
From *figure 8* K =1 is the best value obtained and the accuracy found to be 80%. The odd number of K values seems to be tiebreaker

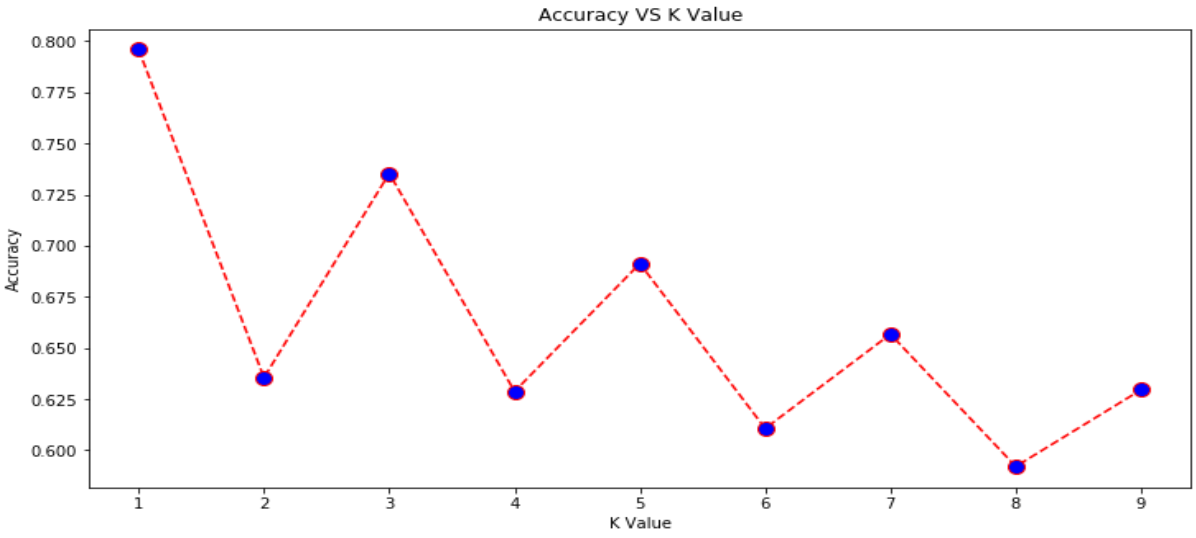

Figure 8: Plot of accuracy and the K value for KNN

From *figure 9* shows the error rate of k value. The error rate is overall optimised between the values k =1 and K=9

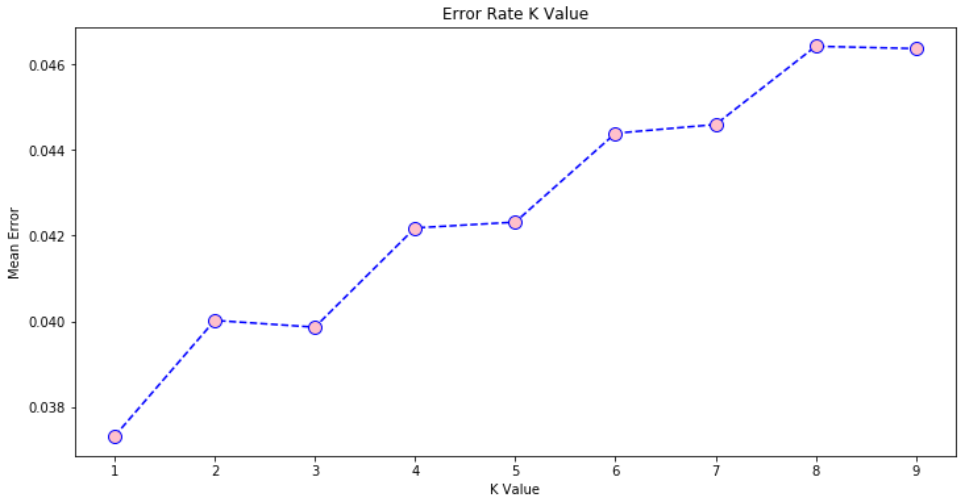

Figure 9: Error rate of k value.

Random Forest Approach

Random forest seems to be computationally cheap compared to other methods. Random forest consists of large number of decision trees that operate as an ensemble. In the random forest each individual tree spit out a class prediction and the class with most votes becomes our prediction for model [3]. In our dataset based on the no of estimators the training data yields a better accuracy(78% -86.7%).Because of the ensemble estimators and underlying decision trees both prediction and training are very fast .This method is extremely flexible and performing better well in task that are underfit by other estimators. *Figure 10* shows the relation between mean error rate and no of estimators. The error rate decreases as the no of estimators decreases

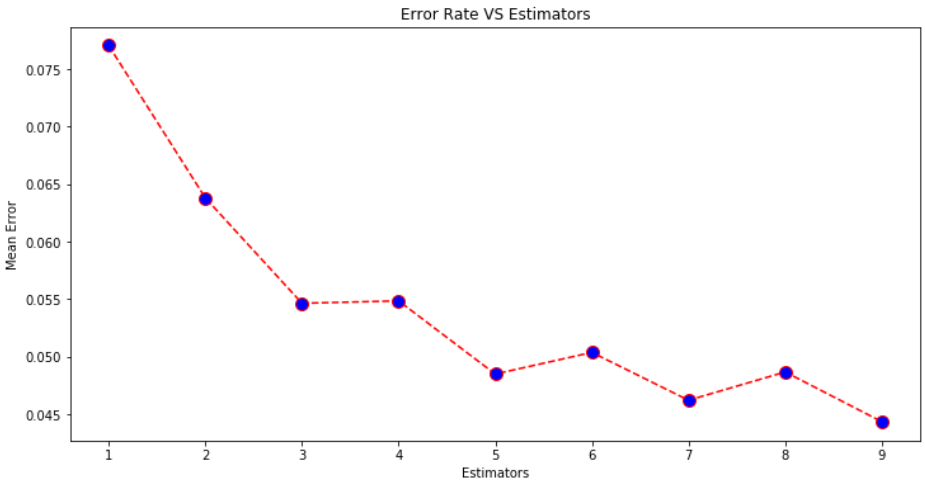

Figure 10: Relation between mean error rate and no of estimators

MLP Classifier

Multilayer perceptron (MLP) make powerful classifiers that may provide superior performance compared with other classifiers. The Multi-Layer Perceptron (MLP) is widely applied recognition processes, especially in human activity recognition [4]. We create a instance of the model, where we define the hidden_layer_sizes, for this model we choose 3 layers and the learning rate of 0.001. Figure 11, shows the main classification metrics precision, recall and f1-score. The micro average for precision, recall and f1-score is 0.69, 0.17 and 0.28 respectively.

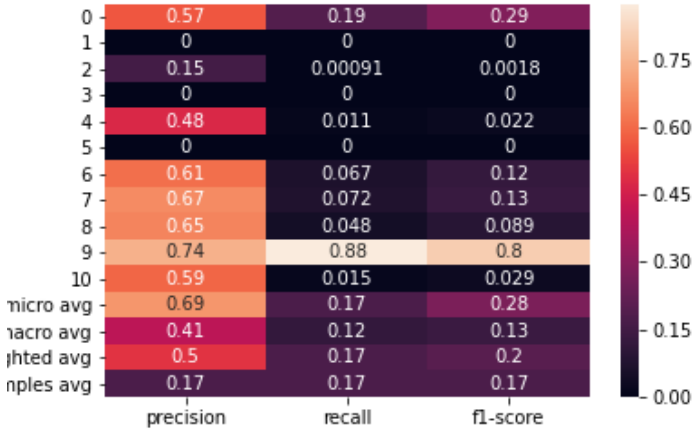

Figure 11: Visualization of classification report of MLP Classifier

Conclusion

This report concludes by encompassing the basic steps which are to be followed for gathering data, analyzing it and using Machine Learning algorithms to make better predictions. Specifically, data analysis and feature extraction techniques in the subject of human activity recognition plays an important role in making multi-label classifications. Using Random Forest, K Nearest Neighbor and Multi-Layer Perceptron as classification algorithms for training the model and performing classifications can prove to be beneficial for different activity recognition from the sensor data.

Techniques such as outlier detection, standardization and Principal component analysis are among a few essential tools to detect data anomalies and find the correlation and variance of the predictor variables.

The model accuracy can be improved modifying and choosing the optimal set of hyperparameters. This involves using Grid Search and Cross Validation techniques while analyzing the error rates calculated by the model while training.

References

[1] https://bbdc.csl.uni-bremen.de/index.php/2019h/25-bbdc-2019.

[2] Guo, Gongde, et al. "KNN model-based approach in classification." *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, Berlin, Heidelberg, 2003.

[3] Biau, GÃŠrard. "Analysis of a random forests model." *Journal of Machine Learning Research* 13.Apr (2012): 1063-1095

[4] Myo, Win Win, Wiphada Wettayaprasit, and Pattara Aiyarak. "Designing Classifier for Human Activity Recognition Using Artificial Neural Network." *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*. IEEE, 2019.

Appendix A

Code Repository

All the code implemented and utilized during the execution of the process described in this report is available at the GitHub repository https://github.com/sumansanyukta/Data_Mining