Capstone Project Proposal

Robson Rocha
October 9th, 2020.

Machine Learning Engineer Nanodegree

1 Domain Background

This project consists in classify an Electrocardiogram (ECG) in normal or abnormal, but different to the others kind of work at this area, it will try to do that by mesures these are possible to extract from an ECG, not reading its signal. This study will show that if its possible or not.

Also, this work will include an analyse of two algorithms to reach this goal, it will be compared two algorithms' answers and it will be check if they can detect a healthy (or not) signal, just understanding its measures.

Beyond, if they reach the goal, will be try to improve the result of those to algorithms just passing as parameters the most important 5 features that are in correspondence with the classification.

2 Problem statement

The electrocardiogram is a very important resource for cardiac medicine. With the graph drawn through an appropriate device, it is possible to view important information about the conditions of the patient's cardiovascular system.

The heartbeat is controlled by an electrical activity, that is, in order for the heart muscle to contract, electrical currents must pass through it. This electrical action in which the heart is inserted is closely linked to the question of being immersed in an electrically conductive saline solution. The principle of electrocardiography lies in the fact that the electrical currents generated in the heartbeat are likely to be registered on the body surface.

The heart has its electrical stimulus originated in a structure called the Sinus or Sinoatrial Node. This stimulus does not contract the heart in a uniform way, because if it were, there would be no blood pumping to the body.

Each contraction stage generates a type of wave that can be recorded on the ECG, this work consists in interpret and transformed in measures these waves and pass them to those algorithms classify them.

Analyse a ECG's graph is a complex work, it requires a good filter and algorithms that can read a long array of floats like a unique feature. The idea here is transform this kind of array in a set of features and based on it to classify the ECG in normal or sick.

This process is summarized in figure 1.

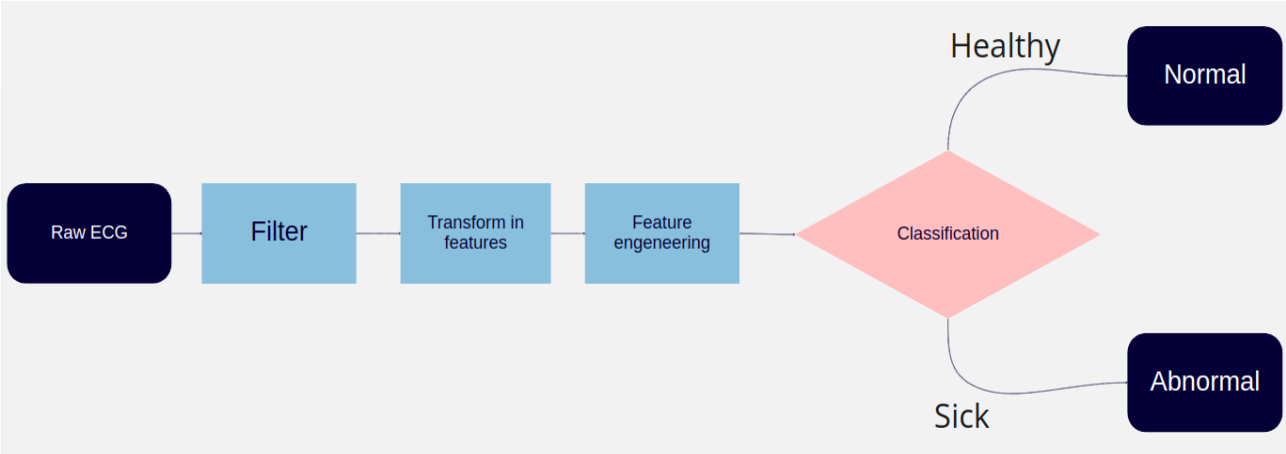

Figure 1. The process.

3 Dataset and inputs

The raw data is download at Kaggle[1]. This dataset is divided in 4 files, 2 sets from the MIT-BIH Arrhythmia Dataset[2] and 2 from the PTB Diagnostic Database[3]. As PTB is divided in Normal and Abnormal classification, it will be the option. The MIT database is divided in a lot classes that is not in the plan of this work.

There are a lot of toolkits that provide a way to transform an ECG in measures. Heartpy[4] is one of them that has a lot of functions that handle with ECG very well and it extracts the measures that will be used to predict. heartpy has a function that filter noise from this dataset. And, after that, it is served as input to other function that transform it in a set of features that interest to this project, these features are discribed below:

- BPM - heart rate (BPM), is calculated as the average beat-beat interval across the entire analysed signal (segment).
- IBI - interbeat interval.
- SDNN - standard deviation of RR intervals.
- SDSD - standard deviation of successive differences. See figure 2.
- RMSSD - root mean square of successive differences. See figure 2.
- PNN20 - proportion of successive differences above 20ms.
- PNN50 - proportion of successive differences above 50ms.
- HR_MAD - median absolute deviation of RR intervals.
- SD1 – standard deviation perpendicular to identity line (Poincaré parameters[5]).
- SD2 – standard deviation a long identtiy line.
- S – area of ellipse described by SD1 and SD2.
- SD1/SD2 – ratio.
- BREATHING RATE – that is the frequency with which the heart beats is strongly

influenced by.

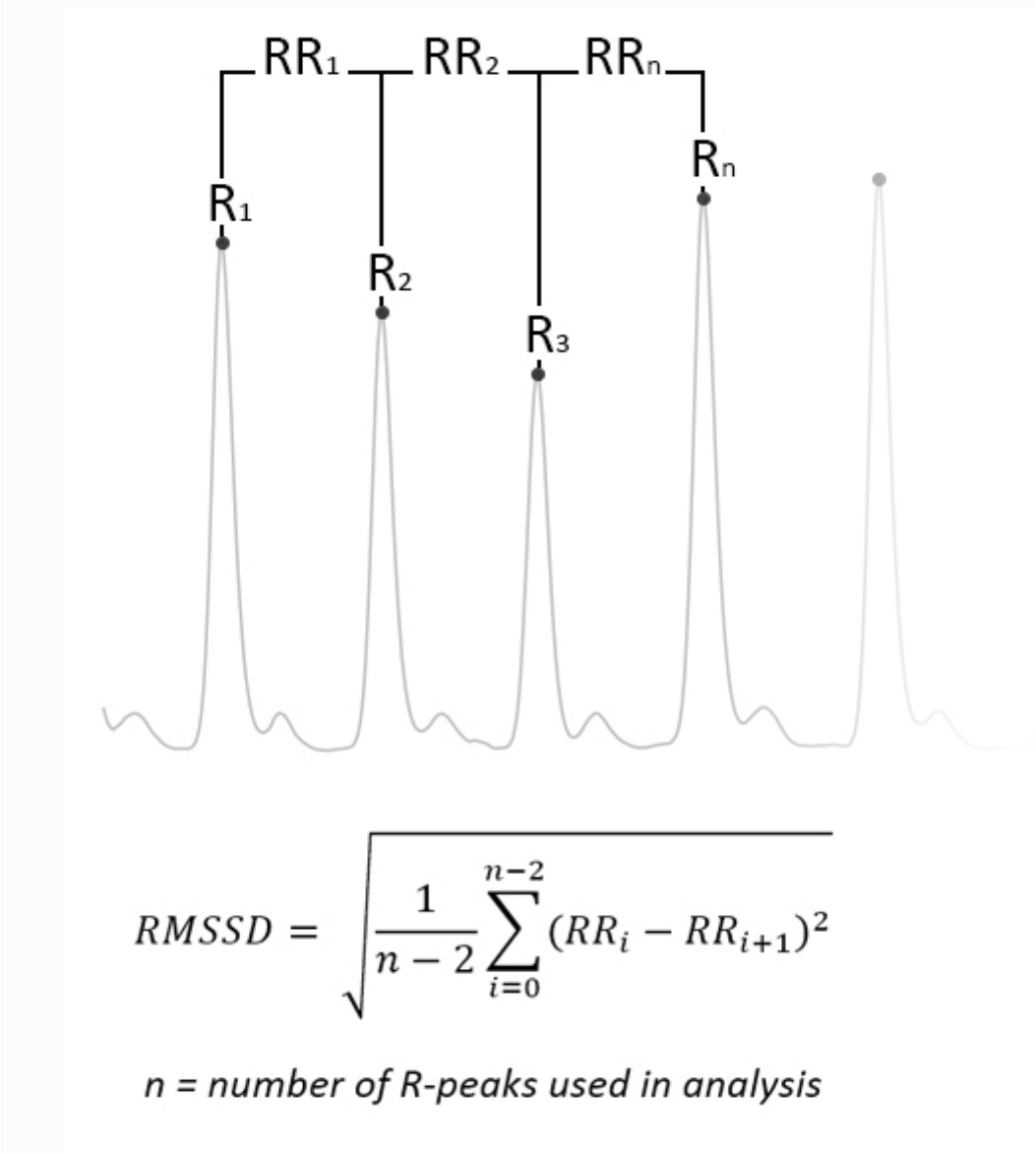

Figure 2. Image displaying the desired peak detection result, as well as the calculation of the RMSSD measure. The SDSD measure is the standard deviation between successive differences[6]

The Kaggle dataset includes 14552 samples of ECG from Physionet's PTB Diagnostic Database with 2 categories in csv file. There are 3754 nornal ECGs and 10798 abnormal ECGs.

4 Solution Statement

In order to face the problem stated above, this project proposes to apply machine learning techniques to study the heart's behavior by analyzing the extracted measures and discover if it is possible classify them.

More specifically, the XGBoost[7] will be trained to show if its possible and its hyperparameters will tuning for improve its training. Beyond, it will be used Pycaret[8] to show if another algorithm that can perform iqual or better than XGBoost. That algorithm will be discovered on the fly.

If their behaviors will be satifatory(>= 80% accuracy), it will be used a random forest[9] to discover the top 5 features' correlation with the classification and another round of training, testing and validation will be carried out to try to improve these algorithms.

The final result will be discover if its possible to classify using the measures and propose another way to predict an disease in an ECG, as an auxiliar to medical decision.

5 Bechmark Model

Like the dataset is very unbalanced, it was chosen XGBoost as first algorithm to classify this kind of data, because it can handle very well with this issue.

It will be used PyCaret to choose the better algorithm that matches with this kind of data.

“PyCaret is an open source low-code machine learning library in Python that aims to reduce the hypothesis to insights cycle time in a ML experiment. It enables data scientists to perform end-to-end experiments quickly and efficiently. In comparison with the other open source machine learning libraries.”[8]

6 Evolution Metrics

It will be compared the confusion matrix and the measures that is possible to extract from it.

Confusion matrix[10] returns the true positives(TP), true negatives(TN), false positives(FP) and false negatives(FN), with these measures, it is possible to extract a lot of others metrics that work like a list of rates of performance that allows to create a ranking and to compare algorithms more thoroughly.

Those other metrics are:

- Accuracy = (TP+TN)/Total
- AUC = Area under curve
- Precision = TP/(TP + FP)
- Recall = TP/(TP+FN)

- F1-Score = 2*Precision*Recall/(Precision+Recall)

The final benchmark reached will be considered satisfatory if all algorithms pass 80% of accurancy.

7 Project Design

It was possible to see in session above, that the dataset had to be filtered by heartpy function and after that it will extract features that will passed to the algorithms as input to training, testing and validation as intent to allow the models classify between normal and abnormal ECG.

The set of features will be normalized using z-score to facilitate the classification of algorithms.

The software requirement for the implementation is as followed:

- Python >= 3.6
- numpy>=1.19.0
- pandas>=1.0.5
- pycaret>=2.1.2
- heartpy>=1.2.6
- sklearn>=0.19.1
- xgboost>=1.1.1

The workflow:

1. To download the dataset fom kaggle
2. To analyse the ECG signals
3. To filter the ECG signals
4. To pass the filtered ECG signals to function that will transform it in features
5. To generate a complete dataframe with the class of each row of features
6. To normalize this dataframe and split it in training, testing and validation sets
7. To tuning XGBoost and training it
8. To use PyCaret to discover a concurrent algorithm and train, test and valid it.
9. To compare XGBoost vs PyCaret's proposal
10. To discover the top 5 features correlation
11. To run steps 7, 8 and 9 again with the features discovered in step 10

8 References

[1] ECG Heartbeat Cathegorization Dataset - https://www.kaggle.com/shayanfazeli/heartbeat
[2] MIT-BIH Arrhythmia Database - https://www.physionet.org/content/mitdb/1.0.0/
[3] PTB Diagnostic ECG Database - https://www.physionet.org/content/ptbdb/1.0.0/
[4] HeartPy – Python Heart Rate Analysis Toolkit's documentation - https://python-heart-rate-analysis-toolkit.readthedocs.io/en/latest/
[5] Shaffer, F., Ginsberg, J.P. (2017), An Overview of Heart Rate Variability Metrics and Norms.

[6] Heart Rate Analyse by heartpy - https://python-heart-rate-analysis-toolkit.readthedocs.io/en/latest/heartrateanalysis.html
[7] XGBoost: A Scalable Tree Boosting System - http://dmlc.cs.washington.edu/data/pdf/XGBoostArxiv.pdf

[8] PyCaret - https://pycaret.org/

[9] Random forest - https://link.springer.com/content/pdf/10.1023/A:1010933404324.pdf

[10] Simple guide to confusion matrix terminology - https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/