Trenton Potgieter
January 12, 2017

A Three-Way Consensus Pipeline for Stress Level Detection

1. Background

As one gets older, an increasingly difficult awareness of our parent's mortality becomes a serious concern. Personally, my parents are both in their early 70's and according to a study [^1] done in **2015** by the **American Heart Association**, around **370,000** people die of heart attacks each year and is the **No. 1** cause of in the United States. In **2014**, around **356,500** people experienced heart attacks out of the hospital. Of that amount only **12%** survived due to emergency medical services intervention. Personally, I would not like my parents to be one the **88%** who suffered from a fatal heart attack and didn't survive due to the fact that there was no intervention by emergency medical services. According to the study, there is a prevalence of almost *third* of the population at risk of *Heart Disease* leading to a *Heart Attack* as one approaches **80+** years of age. Having no personal experience in the Coronary Field of Medical research, it would be difficult for me to diagnose any potential warning signs, but with the advent of wearable technology, the mechanisms are in place to potentially aid in this early warning and detection of heart attacks. The majority of wearable technology today has the built-in ability to monitor heart rates. Therefore in this project, I proposed that this information can be uploaded or sent to a **data ingestion pipeline** that this capable of interpreting, analyzing and detecting an the patterns that could be classified as symptoms of a heart attack.

Additionally, since one of the potential symptoms is the increase in heart rates. There are a number of potential factors that influence the increase in heart rate, but there are well published guidelines [^2] that can be used to determine anomalous patterns. If these anomalies occur, the the **data ingestion pipeline** could proactively determine if a heart attack is about to *or* has occurred and alert the appropriate emergency medical response. Thus proactively preventing a fatal or near-fatal heart attack. As am added benefit, the **pipeline** mechanism can be used to monitor patients who are in *Cardiac Rehabilitation* [^3].

2. Problem Statement

For this Project, I propose creating a classification pipeline that ingests heart-rate signal data (from a simulated wearable monitor) and classifies whether the subject is in a stressful situation that could lead to *Cardiac Unrest*. Additionally, in order to prevent a cry-wolfßcenario or *false-positives*, the pipeline employs a consensus mechanism where three classifiers must all agree on the classification.

3. Datasets and Inputs

The dataset used for this Project was obtained as part of a *Proof of Concept (POC)* project in the **Dell IoT Solutions Lab** [^6] in Santa Clara, California, where a PPG [^4] Pulse sensor was used to measure Heart Rate Variability (HRV) [^7] reading, similar to those found on current wearables like the **Fitbit Charge 2** [^8]. The scope of the original POC is simply to verify if the data can be extracted and filtered to detect peaks in the PPG signal for a one minute data segment. Four separate test subjects (between the ages of 68 and 76) were subjected to different stimuli to induce *stress* and *relaxing* scenarios. The one minute observations (**300** in total) are stored in a `data.csv` file. Each observation has **8** specific features of the PPG waveform, namely:

1. Time $\rightarrow$ Time Stamp of the observation.
2. AVRR $\rightarrow$ Average nnormal"hert beats.

3. AVHR → Average total heart beats.
4. SDRR → Standard Deviation of nnormal"heart beats.
5. RMSRR → Root Mean Squared of nnormal"hear beats.
6. ppNN50 → Proportion of NN50 (50 successive nnormal"heart beats) divided by total number of nnormal"heart beats.
7. ppNN20 → Proportion of NN20 (20 successive nnormal"heart beats) divided by total number of nnormal"heart beats.
8. Label → Stressed or Relaxed.

A sample of the data set is as follows:

```
                Time        AVHR        AVRR       SDRR     RMSSD     ppNN50  \
0  07:16:37 25-08-15  74.393829  25.203703  1.883951  1.710125  16.666666

      ppNN20  State
0  22.222221  relax
```

To address the scope of this project however, I propose training three separate supervised machine learning models by applying the following methodology to create a pipeline:

1. Separate the input data into two separate repositories. One for the observations and one for the labeled output.
2. Apply **normalization** and/or **standardization** techniques to pre-process the data.
3. Define three separate models to evaluate the the data.

   Note: There are two concerns with the above dataset. The *first* is that fact that it has only 300 observations, thus making it a relatively small data set. The *second* is the fact that there are significantly more observations labeled as `relaxed` then there are those labeled as `stressed`. To address this *imbalance* and verify the accuracy of the predictions, I propose leveraging **k-fold cross validation** to split the the data into a **60%** training set and a **30%** testing set. This process will be executed **10** times (10 Folds). The advantage of this technique is that it can treat each test set uniquely, thus addressing the fact that the data set used is relatively small, and provide an average prediction result across the 10 folds. This process will be used for each of the three models.

4. Apply the models and measure their performance on a completely **separate** and as yet **unseen** dataset. This dataset is exactly the same as the training dataset except it is has no `State` label.

4. Solution

Once created, the pipeline (see Section 7) will be used to test and deploy the models on a sample unseen data from the test subjects and hence predict their stress levels.

To accomplish this, the pipeline is comprised of three stages:

1. **Ingesting signal data**. → Collect already filtered PPG [^4] signal data with symbolic peaks (and other features) have been collected for a one-minute time segment. Each one-minute time segment is considered an observation labeled with the class `relax` or `stress`.
2. **Classification model training**.

3. **Classification model application on new, unseen data**. → The final classification is Implementing a Weighted Majority Rule Ensemble Classifier [^5] based on the probability of the time segment observation belonging to either class, using the following:

$$\hat{y} = \arg\max_i \sum_{j=1}^{m} w_j p_{ij},$$

where wj is the weight that can be assigned to the j^{th} classifier.

It is the objective of this project to re-apply the resulting pipeline to a set of new test subjects and hopefully provide a viable prototype that can preemptively warn of potential heart attacks. Based on this final classification, additional future actions can be implemented that are currently outside the scope of this project.

5. Benchmark Model

Since there isn't another comparable methodology for the proposed pipeline and hence there isn't a comparable model implementation to serve as a benchmark, the pipeline methodology will be compared to a simple **Linear Classifier**. The evaluation criteria (see section 6) will be leveraged to compare each individual model's performance as well as the final ensemble model's performance against the *Linear Classifier's* baseline.

6. Evaluation Metrics

Since the success criteria of the project is based on the overall **probability** of the time segment observation belonging to either class (`stressed` or `relaxed`), each individual model as well as the overall consensus pipeline will be evaluated using the following metrics:

1. **Confusion Matrix:** → A tabular breakdown of predictions into a table showing the predictions that are correctly classified as well and the predictions are made incorrectly.

2. **Recall:** → The measure of completeness of the classifier. In other words, if the label is `stressed`, how well does the model predict that the subject is `stressed`. Basically, the ratio of the number of observations the model can correctly recall, to the number of all correct observations.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

3. **Precision:** → The number of positive predictions divided by the total positive class values. So, precision is the ratio of a number of observations the model can correctly predict to a number of all observations the model can recall. In other words, it is how precise the model's recall is.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

4. **F1 Score:** → If the models are good at *Recall*, that doesn't necessarily mean that they are good at *Precision*. The *F1 Score* is the balanced average of the the two. This balanced *F1 Score* is necessary as an overal performance metric due to the fact that if there is a misclassification that the subject is under stress, but isn't, then the emergency medical services are called out unnecessarily. If however, there is a misclassification that the subject isn't stressed, but actually is, then this could result in a fatality. Having the *F1 Score* will allow us to allocate more weight to *Precision* or *Recall*.

$$F1\ Score = \frac{2 \cdot Precision}{Precision + Recall}$$

7. Overall Design

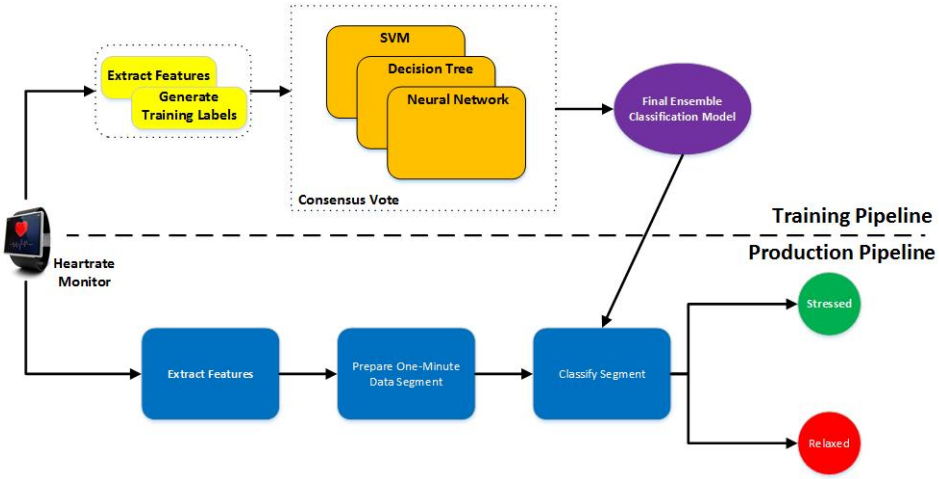

Figure 1: Training/Testing Pipeline

Figure 1 provides an overview of the proposed pipeline that address the solution scope and is separated into two specific workflows:

1. Training
2. Production

7.1 The Training Pipeline

The Training Pipeline is comprised of *three* specific stages.

7.1.1 Feature Extraction The first process - **Feature Extraction** - separates the incoming signal data from the heart rate monitor into two separate training data sets. The first data set are the signal observations, while the second data set are the training labels associated with each observation. The labels are further converted to a binary integer value, demarcating 1 for `relaxed` and 0 for `stressed`. Additionally, in order to account for outlier variables and overfitting, the data is further standardized and scaled using the following:

Standardize:

$$X = \frac{\sum_{i=1}^{n}(x_i - \mu)}{\sigma}$$

Scale:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

7.1.2 Model Training Once the data has been pre-processed, three separate classifieds are trained on the data.

1. Decision Tree
2. Support Vector Machines (SVM)
3. Neural Network

Once each of these models have been trained, there resultant classification probability undergoes a consensus vote to determine the the final classification probability by by using the **Weighted Average Probabilities** ensemble method.

7.1.3 Final Model The last stage of the Training Pipeline is an optimized classification model that can be used for new data.

7.2 Testing/Production Pipeline

Like the Training Pipeline, the Production/Testing Pipeline also comprises of three stages.

7.2.1 Feature Extraction Unlike the first stage of the Training Pipeline, the data from the heart rate monitor is not separated into two data sets. Rather, the signal data is pre-processed; scaled and normalized.

7.2.2 Observation Segmentation The pre-processed data is then split into one-minute segments based on the time stamp of the data. These one-minute segments are established as a single observation of the test individuals stress level at the given time.

7.2.3 Classification The final model from the Training Pipeline is then executed against each one-minute observation segment to classify whether the test subject is `stressed` or `relaxed`.

8. References

1. https://www.heart.org/idc/groups/ahamah-public/@wcm/@sop/@smd/documents/downloadable
2. http://www.heart.org/HEARTORG/HealthyLiving/PhysicalActivity/FitnessBasics/Target-Heart-Rates_UCM_434341_Article.jsp#.WHEiXbGZNE4)
3. https://www.nhlbi.nih.gov/health/health-topics/topics/rehab
4. https://en.wikipedia.org/wiki/Photoplethysmogram
5. http://scikit-learn.org/stable/modules/ensemble.html#weighted-average-probabilities-soft-voting
6. https://www.dell.com/en-us/work/learn/internet-of-things-labs
7. http://www.myithlete.com/what-is-hrv/
8. https://www.fitbit.com/charge2