CSE 512 Machine Learning, Spring 2020, Final Project: "Learning to Detect Heavy Drinking Episodes Using Smartphone Accelerometer Data" (Implementation)

Priyanka Nath (SBU ID: 112715634)

Abhijat Shrivastava (SBU ID: 112584928)

1 Introduction

Heavy consumption of alcohol often leads to reckless erratic behaviour. Interventions are often required so that the user can be made aware of when to stop drinking. However, if the interventions are too frequent, the user tends to pay less attention to the warnings and continues to consume alcohol.

This can be resolved by Just-In-Time Adaptive Interventions (JITAIs). The paper we have selected makes use of JITAIs to select an ideal time to give a warning to the user if their predicted alcohol value – TAC value (Transdermal Alcohol Content) is greater than a given threshold. The authors of the paper collected the accelerometer data of the devices of the participants in a study. The study involved participants who participated in a 'Bar Crawl' event. The dataset consists of the following:

1) The accelerometer data of the mobile devices of the users.
2) The actual TAC value of each user at an interval of 30 minutes.

This dataset can be used to make a regressive model which reads the accelerometer data of each user and predicts the TAC value at different intervals based on this data. However by taking inspiration from the paper[1] we are treating this as a classification problem by classifying a TAC value over 0.08 as intoxicated and otherwise sober.

The paper researched several classification models with a combination of features. According to their analysis they reported the random forest classifier as the best model with 77.5% accuracy.

The classification approach we implemented as per the paper has two major advantages:

- It makes the use of user data which is non-sensitive thus protecting the user's privacy.
- It makes use of JITAIs (Just-In-Time Adaptive Interventions) to ensure that the interventions are as less as possible.

We took inspiration from their analysis and reimplemented a random forest classifier using the python scikit library with different forest depths and our own set of features extracted from the Bar Crawl dataset. We improved on their initial findings. Our random forest classifier successfully labelled when a person was intoxicated and when they were sober with a 82.27 % average accuracy.

2 Implementation Details

We used the UCI Machine Learning Repository Bar Crawl dataset and extracted features using the time series data in both the time and frequency domains. We then scaled the data and used it with the Random Forest Classifier. We split the data into testing and training datasets in a 25:75 ratio. We achieved an accuracy of 82.27% which is actually an improvement on what the paper achieved (approx. 77.5%)

2.1 Data Pre-Processing

The raw data consists of two datasets. The accelerometer data and the TAC readings. All the people who participated in the survey are marked with a unique ID. There is a separate TAC reading file for each ID.

The accelerometer data and the TAC reading data are not perfectly mapped. The TAC reading data consists of some timestamps which are not present in the accelerometer data. First, to convert our problem into a classification problem we convert the continuous TAC reading into discrete values – we labeled any TAC reading above 0.08 as 1 (intoxicated) and a reading below 0.08 is 0 (sober). We segmented the time series accelerometer data into 1-second windows. Using the segmented-data we extracted the time domain and frequency domain features as shown in the table.

We then segment the data into larger 10-second windows and use the 10-second windows to generate 4 summarizing statistics (mean, variance, maximum, minimum). Our final dataset comprises around 30,800 data points and 180 features. The dataset was randomized and split into training and testing datasets in a 75: 25 ratio.

2.2 Features Extracted

We extracted 15 features, some in the time domain and some in the frequency domain [2] for each X, Y, Z value. These features were calculated using the 1-second short window. By binning them into 1-second windows and applying 4 summarizing statistics we obtained a total of 15 x 3 x 4 i.e. 180 features.
The table below explains the 15 features we extracted and used in our implementation.

Feature Name	Definition
Mean	Average of raw signal
Standard Deviation	Standard deviation of raw signal
Median	Median of raw signal
Zero Crossing Rate	Number of times signal changed signs
Max (Raw)	Max of raw signal
Min (Raw)	Min of raw signal
Max (Absolute)	Max of absolute signal
Min (Absolute)	Min of absolute signal
Spectral Entropy	Entropy of energy in both the frequency and time domain i.e 2 features, 1 for each domain.
Spectral Centroid	Weighted mean of frequencies
Spectral Spread	Measure of variance about the centroid
Spectral Roll-Off	Frequency under which 90% of energy is contained
Max Frequency	Maximum value in the frequency domain
Gait	Difference between max and min of one stride

We used methods available in the PyAudioAnalysis library[2] to extract the features in the frequency domain and the standard numpy and pandas libraries to obtain the others.

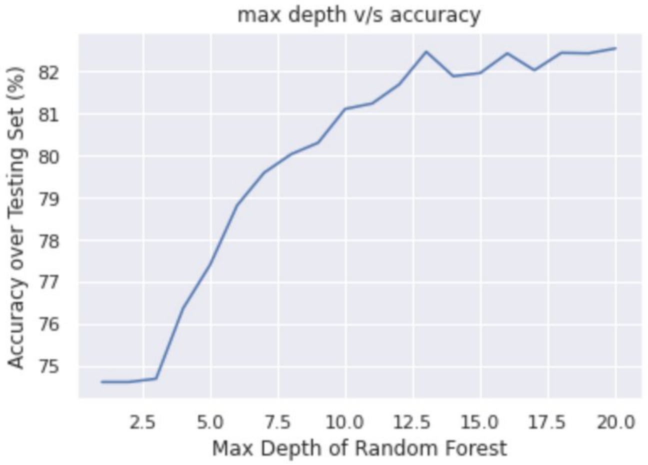
Since the TAC data and the accelerometer data are not mapped 1 is to 1, we combined the two datasets using the timestamps. Our final dataset using the 2-window approach had a shape of 30,874 x 180. The dataset was dumped into a CSV file named final_data.csv.

2.3 Training and Classification

The resultant data frame obtained was used for training the model. We scaled our dataset and then randomly split the data into 75% training data and 25% testing data. using sklearn's train_test_split functionality.

The paper[1] observed its best results on using the random forest classifier so that is where we started. We implemented a random forest classifier and trained it on the training data with different max_depth parameters.

The graph below shows that the accuracy of our model increased with an increase in maximum depth the classifier was allowed to reach. We saw that the classifier performed the best with max_depth = 20 so we ran the classifier with random seeds and max_depth=20. The average accuracy of the model thus came out to be 82.27 % which is an improvement on the paper.



3 Future work

There are some leads worth exploring in this problem. For example the author extracted well over 1000 features which they then reduced. In our implementation we used our own set of features using the author's work as a guide. We could extract more features like the author and experiment with different dimensionality reduction methods starting with simple ones like Principal Component Analysis and extending to t-SNE, etc.

The problem is essentially a binary classification problem. It would be interesting to see how the dataset fares with clustering algorithms. In our assignments we have used the Breast Cancer dataset extensively and even though it was not linearly separable it proved to be effective for clustering.

The dataset used consists of only tri-axial (X, Y, Z) accelerometer data. This might contribute negatively to the performance of the classifier, since the orientation data of the device also plays a major role in detecting the sobriety of the user, and this dataset lacks that. Including gyroscope readings of each timestamp will include the orientation of the device in the training of the classifier, which might result in a more accurate classifier. A dataset containing accelerometer data as well as gyroscope data can be a better dataset for training the classifier for our task.

We reported an improvement in the accuracy of the classifier on the dataset. The improvement is probably due to the concise set of features we selected which we then normalized but it would be interesting to work towards further improvements.

4 Conclusion

In our implementation we used our own set of features using the author's work as a guide. We extracted 180 features in the time and frequency domain from the continuous time series data. We scaled these features so that the values are normalized. We trained our random forest classifier with different parameters to achieve an average accuracy of 82.27 %. This is an improvement on the accuracy noted by the paper using the same classifier.

References

[1] Jackson A Killian, Kevin M Passino, Arnab Nandi, Danielle R Madden, John Clapp "Learning to Detect Heavy Drinking Episodes Using Smartphone Accelerometer Data"

[2] Hsin-Liu Cindy Kao,Bo-Jhang Ho, Allan C Lin, and Hao-Hua Chu "Phone-based gait analysis to detect alcohol usage"

[3] Inbal Nahum-Shani, Shawna N Smith, Bonnie J Spring, Linda M Collins, Katie Witkiewitz, Ambuj Tewari, and Susan A Murphy "Just-in-time adaptive interventions (jitais) in mobile health: key components and design principles for ongoing health behavior support"

[4] PyAudioAnalysis source code: https://github.com/tyiannak/pyAudioAnalysis

[5] Sklearn: https://scikit-learn.org/stable/

[6] Bar Crawl: Detecting Heavy Drinking Data Set https://archive.ics.uci.edu/ml/datasets/Bar+Crawl%3A+Detecting+Heavy+Drinking