BIOSTAT 546: Machine Learning for Biomedical and Public Health Data

Final Project Report

BY: Venkat Srinidhi Vaddy (2173456, vvaddy@uw.edu)

Problem Statement:

Performing statistical and exploratory data analysis on ECG data containing signals of single heartbeats and deploying multivariate predictive models to predict and classify heart arrhythmias, that is, whether the heartbeat is normal or abnormal with the highest accuracy.

Introduction:

The heart contracts in a rhythmical manner to pump blood throughout the body with contractions beginning at the atrial sine node and propagates through the rest of the muscle. This electrical signal propagation has a pattern and because of this, electrical currents are generated on the surface of the body causing variations in the electrical potential of the skin surface. These signals can be captured with electrodes and appropriate equipment and are called ECG signals.

Thus, an ECG signal or an electrocardiogram measures the electrical activity of the heart to detect heart functioning and heart diseases. By analyzing the combination of action impulse waveforms generated by different specialized cardiac tissues in the heart, it is possible to detect some abnormality. The abnormal beating is called an arrhythmia and each type of arrhythmia is associated with a pattern that can be seen in the ECG signal.

The process of analyzing, identifying and classifying arrhythmias can be very troublesome and requires analysis of each second and every curve of each heartbeat of the ECG records. Therefore, in our data, we have variables that have the impulse value at each second of a heartbeat which are analyzed and modeled to predict and classify any abnormality.

About the dataset:

1. The dataset consists of 12552 observations of single heartbeats.
2. The predictor vector is of length p = 187, wherein these 187 variables describe the ECG signal impulse value at each consecutive second.
3. The diagnostic response vector Y_train is a categorical vector depicting 0 for normal heartbeat and 1 for abnormal heartbeat.

Statistical Analysis:

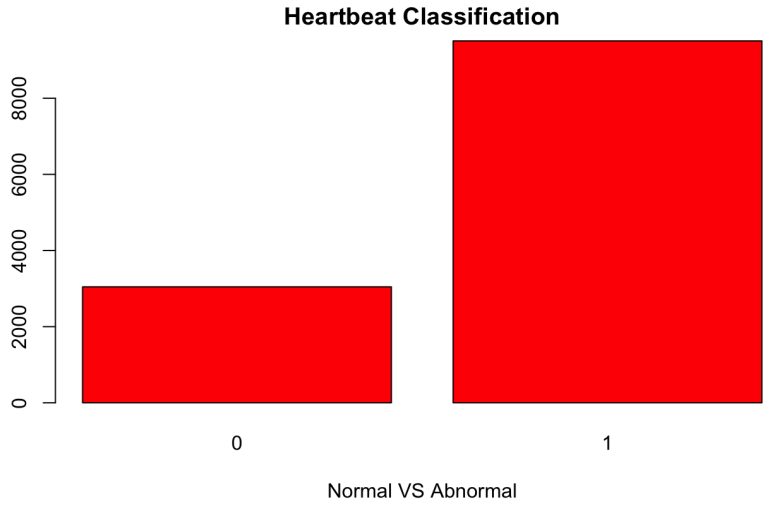
1. **Data Exploration : Loading and describing the data**
    - The dataset X_train has 12552 observations and 187 variables. The vector y_train has 12552 rows of categorical values of 0 and 1 for normal and abnormal heartbeats. This forms our training data to train the different models on.
    - The dataset X_test has 2000 observations and 187 variables but there is no y_test. This is unlabelled data and the model's accuracy can be checked on this unseen dataset.
    - table(y_train) function is used to check the number of observations in each class.

- There are 3046 observations in 0/normal class
- There are 9506 observations in 1/abnormal class

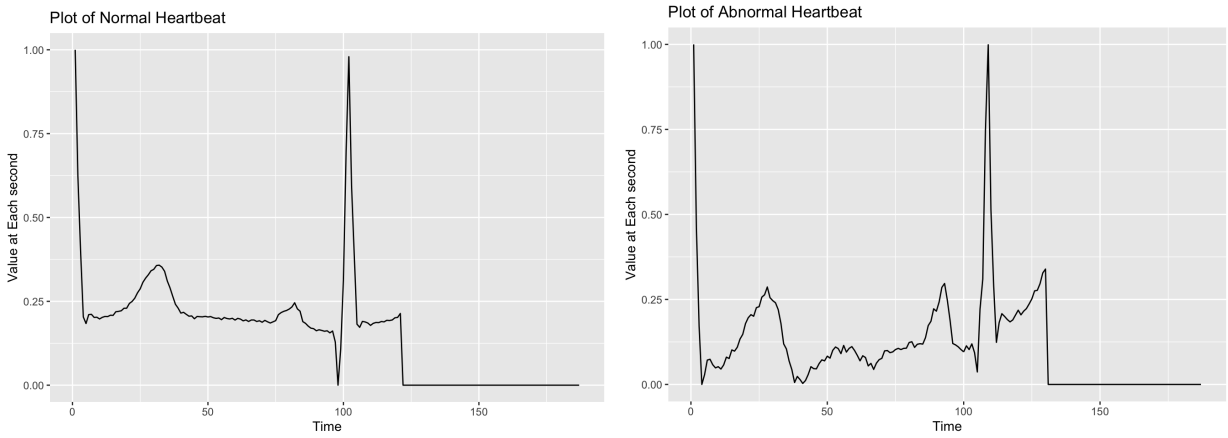
- colSums(is.na(X_train)) function is used to check for any missing values in any of the variables. There were no missing values in any of the columns.

2. Data Visualization:

Plotting the distribution of the observations in the two classes - normal and abnormal using Barplot. There are 3 times more observations in the abnormal class than normal class.



Plotting the heartbeat with time for each of the classes by using the Unlist function.



The normal heartbeat has less variability than the abnormal heartbeat and the curve is smoother. The abnormal heartbeat also has higher peaks, which means it goes beyond the normal range in P wave, PR interval and QRS duration.

3. Data Processing and Transformation

Since the data in all the variables are in the same scale, normalization isn't required. The testing dataset provided is unlabelled. Therefore, to check the accuracy of the trained models, it is

important to split the given training dataset into train and test sets of fixed size. Setting 11000 rows in the training dataset and the rest in the test dataset to check for model performance.

4. Predictive Modeling and Accuracy Evaluation:

a. **Logistic Regression:** It is a classification algorithm used to assign the observations into discrete classes. It is easy to implement, interpret and easy to train. It performs well on the datasets which are linearly separable. It is the simplest of all for classification and therefore, was the first model to train and run. It has given an accuracy of **85% on the created training dataset and 83.69% on created testing dataset**. The model has performed poorly on both training and test datasets. This is probably because of high variance and overfitting due to high dimensionality of the dataset. Therefore, it is important to deploy a more complex model with regularization.

```
[1] "Training data - confusion matrix"
    Predicted
True    0    1
   0 1548 1118
   1  530 7804
[1] "Training data - Model accuracy"
[1] 0.8501818
Warning in predict.lm(object, newdata, se.fi
  prediction from a rank-deficient fit may b
[1] "Testing data - confusion matrix"
    Predicted
True    0    1
   0  218  162
   1   91 1081
[1] "Training data - Model accuracy"
[1] 0.8369845
```

Logistic Regression

```
[1] "Training Data Accuracy"
[1] 0.8486364
[1] "Testing Data Accuracy"
[1] 0.8337629
[1] "Training Confusion Matrix"
    Predicted
True    0    1
   0 1519 1147
   1  518 7816
[1] "Test Confusion Matrix"
    Predicted
True    0    1
   0  211  169
   1   89 1083
```

Ridge Regression

```
[1] "Training Data Accuracy"
[1] 0.8446364
[1] "Testing Data Accuracy"
[1] 0.8363402
[1] "Training Confusion Matrix"
    Predicted
True    0    1
   0 1479 1187
   1  522 7812
[1] "Testing Confusion Matrix"
    Predicted
True    0    1
   0  207  173
   1   81 1091
```

Lasso Regression

b. **Ridge Regression:** This model is a regularization algorithm that shrinks the parameters towards 0. It is mostly to eliminate multicollinearity in the dataset. By shrinking the coefficients using L2 regularization, it reduces the model complexity. It has given an accuracy of **84.8% and 83.3% on training and testing datasets**. It has performed worse than Logistic Regression. It is generally used to eliminate multicollinearity and probably hasn't performed well because there is no high correlation between the variables. Despite regularization, Ridge regression can still lead to high variance and low bias.

c. **Lasso Regression:** This model is a regularization algorithm that eliminates variables by making some parameters 0 using L1 regularization. It does automatic feature selection when we have more features. It has high interpretability as it shows which features are more significant. Lasso model has given an accuracy of **84.4% and 83.6% on training and testing datasets**. The performance is similar to that of Ridge Regression and is worse than Logistic Regression. The poor performance even after regularization models shows that the linear models aren't working on the dataset and non-linear models should be deployed.

d. **Decision tree:** It uses the tree representation to solve the problem in which each leaf node corresponds to a class label and attributes are represented on the internal node of the tree. Attributes are the variables used to split the data. It is suitable for non-linear data. It has shown an accuracy of **89% on training dataset**. Higher than all the other previous models. But the

accuracy on the created testing dataset is 86.9% and on X_test is 82.45%. This shows that the decision tree might be overfitting on the training data. Decision trees are prone to overfitting, especially when a tree is particularly deep. This is due to the amount of specificity we look at leading to smaller samples.

```
[1] "Training data - confusion matrix"
    Predicted
True    0    1
   0 2024  642
   1  558 7776
[1] "Training data - Model accuracy"
[1] 0.8909091
[1] "Testing data - confusion matrix"
    Predicted
True    0    1
   0  280  100
   1  102 1070
[1] "Training data - Model accuracy"
[1] 0.8698454
```

Decision Tree

```
Call:
 randomForest(formula = y ~ ., data = train_df, importance = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 13

        OOB estimate of  error rate: 2.68%
Confusion matrix:
     0    1 class.error
0 2437  229 0.085896474
1   66 8268 0.007919366
[1] "Testing Data Accuracy"
[1] 0.9697165
```

Random Forest

e. **Random Forest:** A random forest is a non-linear meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. It uses bagging and features randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree. In this model, the random forest has 500 trees and at each split, tries 13 variables together to split the data.

   The random forest model has shown an **accuracy of 96.9% on the testing dataset**. This is the highest accuracy achieved from all the models, also depicting that the non-linear assumption was correct. This model is not overfitting as the accuracy is on the testing data. Moreover, as we increase the number of trees, the algorithm is less likely to overfit. The testing performance of Random Forests does not decrease (due to overfitting) as the number of trees increases.

   Thus, learning from the interpretations drawn from other models, Random forest is the most suitable method of prediction and classification for this problem statement with 96.9% accuracy on created test data. The model has given an **accuracy of 94.9% on X_test** (2000 observations) dataset.

REFERENCE:

Mohammad Kachuee, Shayan Fazeli, and Majid Sarrafzadeh. "ECG Heartbeat Classification: A Deep Transferable Representation." arXiv preprint arXiv:1805.00794 (2018).