Predicting Heart Failure with Random Forest

Kevin De La Torre
Computer Science Department
Cal Poly Pomona
Email: ktorre@cpp.edu

ABSTRACT

In this paper I will be building a random forest classifier using a "Heart Failure" data set found on Kaggle. A bruteforce approach will be used to generate classifiers and find an optimal classifier. I will be using 13 features of the data set to train the classifier to try to predict a possible heart failure in the time after a patients first visit to the doctor.

I. Introduction

Heart Failure is a major issue plaguing the world as it is the leading cause of death in the world so being able to predict if a person is in higher risk of being affected by it would help save innumerable lives. There are many major risk factors that can lead to heart failure such as diabetes, drug use, age, sex, etc.[3] There are of course many risk factors involved in heart disease but our dataset makes use of 12 features that could be considered risk factors, these features are outlined in the following section. The problem I will be solving in this paper is to try to predict a heart failure of a patient aged 40 and above, based on results after an appointment with their medical provider. I will be training a Random Forest classifier to try to predict if a person is likely to experience a heart failure in the time period before their next doctors appointment.

II. Data Set

The data set [1] I will be using was found on Kaggle. It has 12 features and a target feature that can be used to try to predict death by heart failure. These features include:

A. Age

An integer representing the age of the relevant patient.

B. Anemia

A Boolean [0,1] representing whether the patient has Anemia or not.

C. High Blood Pressure

A Boolean [0,1] representing whether the patient has high blood pressure.

D. Creatinine Phosphokinase (CPK)

An integer representing the level of the CPK enzyme in the blood.

E. Diabetes

A Boolean [0,1] representing whether the patient has diabetes or not.

F. Ejection Fraction

An integer representing the percentage of blood leaving the heart at each contraction.

G. Sex

A Boolean [0,1] representing the sex of the patient.

H. Platelets

A float representing the volume of platelets in the blood.

I. Serum Creatinine

A float representing the level of serum creatinine in the blood.

J. Serum sodium

An integer representing the level of serum sodium in the blood.

K. Smoking

A Boolean [0,1] representing whether the patient smoked or not.

L. Time

An integer representing the length of the follow up period until the next doctor's appointment.

M. [Target] Death Event

A Boolean [0,1] representing whether the patient died within the "Time"/follow-up period.

III. Machine Learning Model

My approach to this problem was to use the Heart Failure dataset and train a large amount of randomly seeded random forest classifiers (n=1000) with different starting parameters to achieve the greatest F1-score. I measured accuracy of this classifier by it's F1-score because when predicting for heart failure I wanted to minimize false positives and especially false negatives as errors like these could be life-changing for the patient while measuring for just accuracy might allow for more errors to arise. For training I created an array of 1000 randomly generated integers as the randomization seeds. the

approach to find an optimal random forest was a bruteforce approach of testing 40 different values for every seeded tree and keep track of best F1-score I measured from the trees. As for the bruteforce approach it involved creating 20 trees with different number of estimators (number of trees) and choosing the random forest with the highest F1-score and lowest estimator count as the base estimator value for the next step in the process. The next step in the model bruteforced another 20 trees with the estimator count from before but in this step I changed the 'max features' parameter that controls how many features are considered when splitting on every node. I repeated these steps for every seed generated from earlier, I then compiled the best F1-scores from all the seeded random forests and chose the classifier showing the highest F1-score.

IV. Results

After building 40k trees a highly accurate tree was found for this dataset. Using the random seed [7022146] to seed the classifier I was able to build a random forest classifier using 120 estimators and 0.5 max features ratio to get an F1-score of .96 and an accuracy score of .97.

V. Conclusion

In conclusion in this paper I used an extremely bruteforce-centric method of finding an optimal random forest classifier. Although using this method was very computationally expensive it was able to find a pretty good classifier at F1-score of .96, with an accuracy of .97. This paper showed that even though a rudimentary bruteforce method was used a performant classifier was constructed.

References

[1] Larxel, (2020, June). Heart Failure Prediction, Version 1. Retrieved March 20, 2020 from https://www.kaggle.com/andrewmvd/heart-failure-clinical-data

[2] A. Géron, "Chapter 7. Ensemble Learning and Random Forests," in Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems, Sebastopol, CA: O'Reilly Media, Inc., 2019, pp. 189–199.

[3] MayoClinic, "Heart disease," Mayo Clinic, 09-Feb-2021. [Online]. Available: https://www.mayoclinic.org/diseases-conditions/heart-disease/symptoms-causes/syc-20353118. [Accessed: 11-May-2021].