**Voting Classifier Machine Learning Model for Predicting Heart Failure**

**Predicting Heart Failure Dataset**

**https://www.kaggle.com/datasets/whenamancodes/heart-failure-clinical-records**

Justin Mattix, [jmattix@bellarmine.edu](mailto:jmattix@bellarmine.edu)

**ABSTRACT**

For this project I decided to use the Heart Failure dataset to construct a model using voting classification to predict whether a patient is at risk of heart failure. Within the Voting Classifier model, I used the following estimators, Logistic, Random Forest, SVC, DT (gini and entropy). Using this classifier, predictions can be made on whether a person is at high risk of heart failure.

1. **INTRODUCTION**

The dataset I used for this project is the heart failure dataset. In heart disease early detection of the disease is important to correct any habits, such as smoking, that could improve a person’s chances of survival. This is where a machine learning algorithm could come in handy. Most cardiovascular diseases can be prevented by addressing a patient’s behaviors. Using this model medical professionals as well as patients can see whether a patient is at risk and what factors they could change / eliminate to help their chances.

1. **BACKGROUND**
   1. *Predicting Heart Failure*

The dataset I have selected using 13 clinical features to predict heart failure. First is age of the patient in years as more elderly patients are at a higher risk of heart failure. Anemia is a decrease of red blood cells or hemoglobin. Next is if the patient has high blood pressure. Creatinine phosphokinase (CPK) is, and enzyme found in the heart. Diabetes is a condition that has been linked in an increased risk in heart failure. Ejection fraction is a measurement of the percentage of blood leaving your heart each time it squeezes. High blood pressure, also linked to heart failure, is defined as systolic pressure from 120 – 129 and a diastolic pressure less than 80. Platelets are the smallest of the blood cells and a low amount of them can lead to heart failure. Serum creatinine is based on a blood test that measures the amount of creatinine in your blood. Serum sodium is a way for a doctor to measure the amount of sodium in your blood. Sex is a feature because women have been found to be at a higher risk of heart failure than men. Smoking is also taken into consideration as smoking increases risk. This dataset has 300 entries with no missing data.

* 1. *Machine Learning Model*

For my model I used the Voting Classifier with the following estimators Logistic, Random Forest, SVC, and Decision Tree classifiers. Logistic classification is a type of statistical model often used in classification and predictive analytics. Logistic classification estimates the probability of an event as in this case whether a person is at risk of heart failure. This type of classification maximizes the log likelihood function to determine the beta coefficients the model will use. Logistic classification uses gradient descent to find the global maximum. Decision Trees are a supervised learning approach for classification. In these tree structures, leaves represent classes (targets). Its goal is to predict the target by starting a root and traversing down the tree going through each feature to arrive at the terminal nodes at the bottom which will decide in this case whether a person is at risk for heart failure or not. These nodes in the middle of the root and the terminal nodes represent splits based on the classification features. Random forests are an ensemble learning method for the classification. It is a meta estimator that fits multiple decision tree classifiers on different samples of the dataset. It uses the average to improve the accuracy and to control the over-fitting typically found in decision trees. The implementation of SVC is based on the libsvm library. It is often considered as more accurate than logistic regression and decision trees. The classifier separates the data on a hyperplane. It finds an optimal hyperplane to help predict new data points.

1. **EXPLORATORY ANALYSIS**

This data set contains 300 samples with 13 columns with various data types. All the data types of the dataset were of type integer except for Serum Creatinine which was of type float. All the rows had sufficient data and there were no missing values. This distplots shown in the collab file of the various datatypes to not show any unusual distributions except those that are binary such as Diabetes where the only possible values are 0 and 1. The heatmap is similar as shown in the collab file there is no irregularities that would cause concern.

**Table 1: Data Types**

|  |  |
| --- | --- |
| *Variable Name* | *Data Type* |
| *Age* | *integer* |
| *Anemia* | *integer* |
| *High Blood Pressure* | *integer* |
| *Creatinine Phosphokinase* | *integer* |
| *Diabetes* | *integer* |
| *Ejection Fraction* | *integer* |
| *Platelets* | *integer* |
| *Serum Creatinine* | *float* |
| *Serum Sodium* | *integer* |
| *Sex* | *integer* |
| *Smoking* | *integer* |
| *Time* | *integer* |
| *DEATH\_EVENT* | *integer* |

1. **METHODS**

In this section, describe how you prepared the data for your model and performed multiple experiments using different parameters for the model.

* 1. *Data Preparation*

For my preparation first I checked the distributions of all my independent variables. This showed that all my variables were distributed well enough to work with. Second, I checked the correlation with a heatmap, and nothing seemed out of the ordinary. Next, I examined the columns. I found that the dataset did not have any missing values that needed to be imputed. Next, I split, scaled, and stratified the dataset using train\_test\_split and StandardScaler. I did not drop any columns from the dataset as there was only 13 columns and they all felt relevant to predicting the target variable.

* 1. *Experimental Design*

You will run your model several times with different parameters to see what different results you get. In a table, describe your experimental parameters. Three or four experiments are sufficient. This is where you will describe how you divided your data into train, validate and test data sets. For example:

Table X: Experiment Parameters

|  |  |
| --- | --- |
| **Experiment Number** | **Parameters** |
| 1 | All 13 raw features with 80/10/10 split for train, validate, and test |
| 2 | All 13 normalized features with 80/10/10 split for train, validate, and test |
| 3 | All 13 normalized features with 70/15/15 split for train, validate, and test |
| 4 | All 13 normalized features with 60/20/20 split for train, validate, and test |

* 1. *Tools Used*

The following tools were used for this analysis: Python running in the Google Collaboratory environment for a Windows computer were used for all analysis and implementation. In addition to base Python, the following libraries were used Pandas 0.18.1, Numpy 1.11.3, Matplotlib 1.5.3, Seaborn 0.7.1, SKLearn 0.18.1. I chose these libraries for their use in graphics in matplotlib and seaborn as they allow me to display matrices and histograms for the data. Pandas is useful for reading in the dataset from csv. Matplotlib is useful for everything to do with preprocessing and creating and training the model.

1. **RESULTS**
   1. *Mean square Error and R-Square calculation*

Experiment 1: MSE – 0.183 R-Sqaure – 0.164

Experiment 2: MSE – 0.21 R-Sqaure – -0.001

Experiment 3: MSE – 0.18 R-Square – 0.13

Experiment 4: MSE – 0.183 R-Square – 0.16

* 1. *Discussion of Results*

The most accurate was Experiment 3. This makes sense as it has the normalization to help with the accuracy of the model. This is shown by the low MSE and R-Square values. The worst was experiment 2. This surprised me as I assumed the raw data would prefer the worst.

* 1. *Problems Encountered*

I think the largest problem I faced during this experiment was finding relevant data. During my initial search I came across datasets that were either too small and the model would overfit or were so large and complicated I could not make sense of the independent variables that were contributing to the target.

* 1. *Limitations of Implementation*

Discuss the limitations of your model. Is there is reason it might not be the best way to model the data? What other models might work better?

* 1. *Improvements/Future Work*

I think that improving the dataset would improve the model. Adding to the number of features and the number of samples would contribute to a more solid model. Possibly adding a blood pressure number value instead of a 0 or 1 for high blood pressure for example. Another possible addition could be like a BMI or weight value. These additions I feel would improve my model and be more useful to the medical field.

1. **CONCLUSION**

I think this model is satisfactory for what it is. I would have liked to have a more detailed dataset but with what I was given I feel I made the best model I could. My problem was searching for a dataset where I could first understand what the features meant, second had a good number of samples to avoid overfitting, and lastly one that didn’t have huge gaps of missing data within the dataset. However, the model I chose was a good compromise to make a fairly accurate model.

**REFERENCES**

“SVM Classifier Using Sklearn: Code Examples.” *Data Analytics*, 6 May 2022, https://vitalflux.com/svm-classifier-scikit-learn-code-examples/#:~:text=SVC%2C%20or%20Support%20Vector%20Classifier,the%20data%20into%20two%20classes.

Taboga, Marco. *Logistic Classification Model (Logit or Logistic Regression)*, https://www.statlect.com/fundamentals-of-statistics/logistic-classification-model#:~:text=The%20logistic%20model%20(or%20logit,StatLect%20official%20YouTube%20channel.

Yiu, Tony. “Understanding Random Forest.” *Medium*, Towards Data Science, 29 Sept. 2021, https://towardsdatascience.com/understanding-random-forest-58381e0602d2.