# HEART DISEASE PATIENTS' CLASSIFICATION

**Dharmit Dalvi**
**Project for CS677: Data Science with Python**
**April 2019**

## Dataset

The dataset is taken from Kaggle datasets, the link for which is as follows: https://www.kaggle.com/ronitf/heart-disease-uci

This dataset contains 14 attributes, it's a subset of the original dataset that contains 76 attributes. These are the 14 main ones used for experiments and research purposes. The original dataset can be found on UCI datasets, the link for the same is as follows: https://archive.ics.uci.edu/ml/datasets/Heart+Disease

Every tuple in the dataset stores data for a patient. The dataset contains following attributes for every patient tuple:

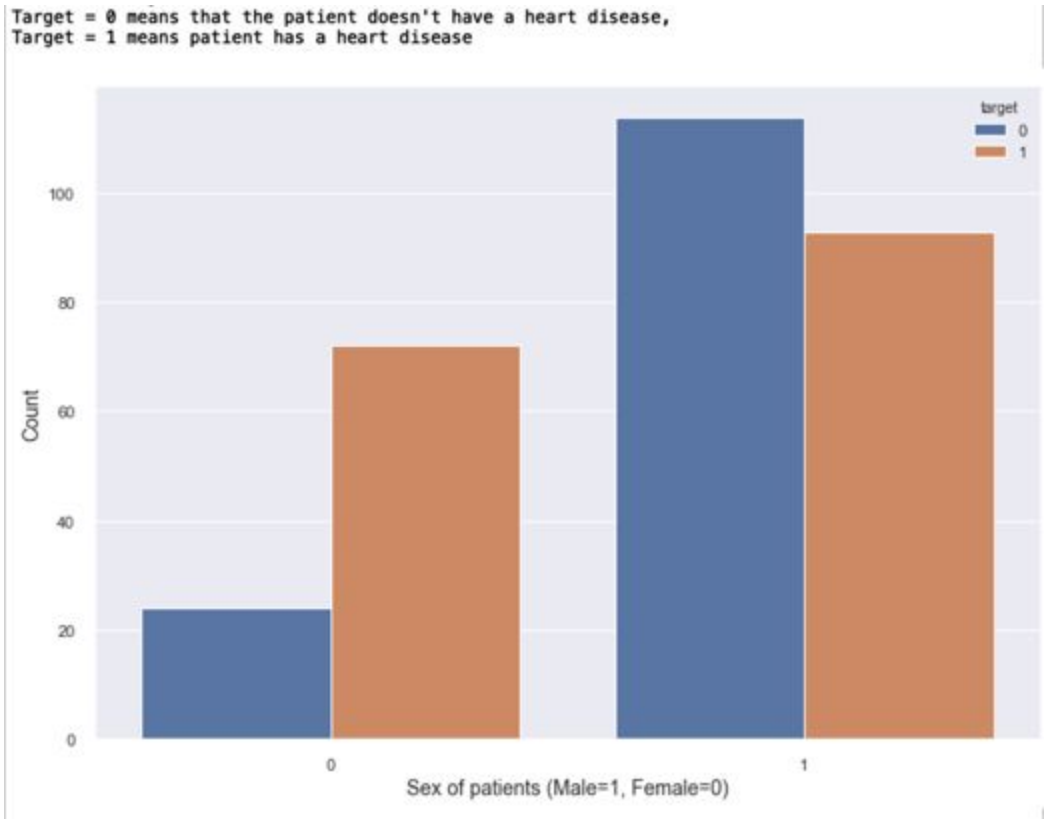
1. age
2. sex
3. chest pain type
4. resting blood pressure
5. serum cholesterol in mg/dl
6. fasting blood sugar > 120 mg/dl
7. resting electrocardiographic results
8. maximum heart rate achieved
9. exercise induced angina
10. oldpeak = ST depression induced by exercise relative to rest
11. the slope of the peak exercise ST segment
12. number of major vessels (0-3) colored by flourosopy
13. thalessemia

The goal is to predict whether the patient has a heart disease or no. So our label is the 14th attribute, which is "target". Target attribute contains values 1 or 0. Value =1 means

that the patient has a heart disease, 0 means that the patient doesn't have a heart disease.

**Visualization**

Firstly, I created plots to visualize how each of the 13 attributes in the dataset affects our label, the target attribute. I created plots using the Seaborn library. I created countplots for each of the attributes with respect to the label. Following is an example graph of sex of patients v/s the label, which describes how sex affects the label:




From the above plot, we can infer that males are more than females among the patient entries in this dataset. We can also infer that the number of females having a heart disease is higher than the number of females not having the disease. For males however, it's the other way round.

I have plotted countplots similarly for all the 12 other attributes as well in my code.

Next, I plotted violinplots. Following is the example of a violinplot that shows the relationship between sex of the patients and the label:


The violinplot clearly tells us that the number of patients having a heart disease is much more than the number of patients not having the disease.

Again, I have plotted violinplots similarly for all the 12 other attributes as well in my code.

**Classification**

I further went on to build classification models on the dataset. I primarily used scikit-learn library for the same.
First, I split the dataset into training and testing sets. On splitting, the training set now has the label attribute, and the testing set doesn't. We are predicting a label for every tuple in our testing set based on our training set and classification algorithms. I built models based on the following algorithms:

- Classifiers:
    - Naive Bayes
    - LinearSVC

    - Logistic Regression
- Ensemble methods:
    - Adaboost
    - Random Forest

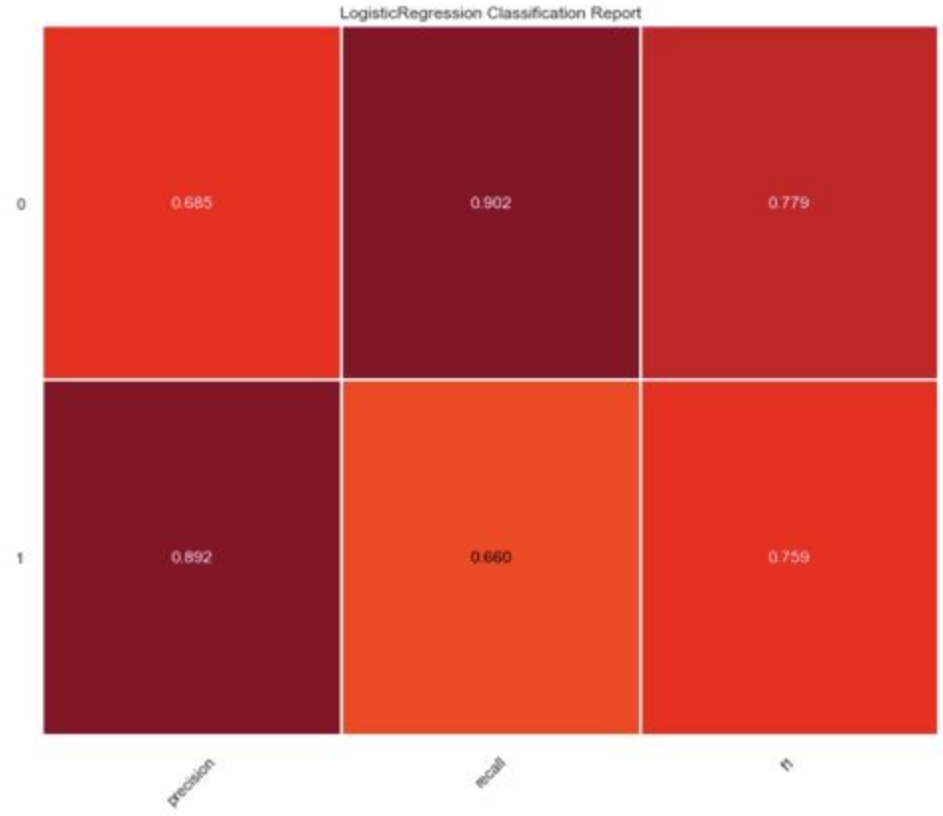
Following are the accuracies I obtained for each of the above models:

| Classifier | Accuracy (in %) |
|---|---|
| Naive Bayes | 79.12 |
| LinearSVC | 76.92 |
| Logistic Regression | 76.92 |
| Adaboost | 78.02 |
| Random Forest | 82.41 |

**Evaluation of classifiers**

I evaluated each of the classifier models using the Yellowbrick python library. I generated a graph that shows the precision, recall and f1 score values for both our labels. This is an effective evaluation method, since it eliminates the need to manually calculate these parameters, and we can visually view how the model has performed. For instance, following is the Yellowbrick evaluation plot for Logistic Regression classifier:

```
Accuracy using Logistic Regression: 0.7692307692307693
```




I have similarly evaluated the other models as well in my code.

**Conclusion**

Random forest ensemble method works best on the heart disease dataset, with an accuracy of 82.41 %.