Machine Learning Engineer Nanodegree

Capstone Project

Heba Ghonaemy
March 10th, 2019

I. Definition

Project Overview

Heart disease is the number one killer of men and women in the United States today. The Centers for Disease Control and Prevention (CDC) estimate that heart disease causes about 1 in 4 deaths in the United States each year. That's 610,000 people per year. About 735,000 people in the United States have a heart attack each year.

Problem Statement

Heart disease is considered one of the top preventable causes of death in the United States. It is believed some genetic factors can contribute, however the disease is largely attributed to poor lifestyle habits. *By discovering patterns in data like* (Blood Glucose Level / Blood Pressure / Cholesterol / etc.). We can detect the presence of heart disease. The goal of this project is early detection of cardiovascular disease which can be the difference between life and death.

Metrics

For this project our main focus is on Classifiers being sensitive to false negatives (FN). For this dataset, false negative is a person that has heart disease but classifier decided the person does not have any heart diseases. In other words, classifier said that the ill person is healthy.

Because of this, I believe the **recall** score should be used as the most appropriate measure for classifier validation.

$$\begin{aligned}\text{sensitivity} &= \frac{\text{number of true positives}}{\text{number of true positives} + \text{number of false negatives}} \\ &= \frac{\text{number of true positives}}{\text{total number of sick individuals in population}} \\ &= \text{probability of a positive test given that the patient has the disease}\end{aligned}$$

II. Analysis

Data Exploration

The dataset used is the Heart disease UCI from kaggle. This database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. In particular, the Cleveland database is the only one that has been used by ML researchers to this date. The "goal" field refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4.

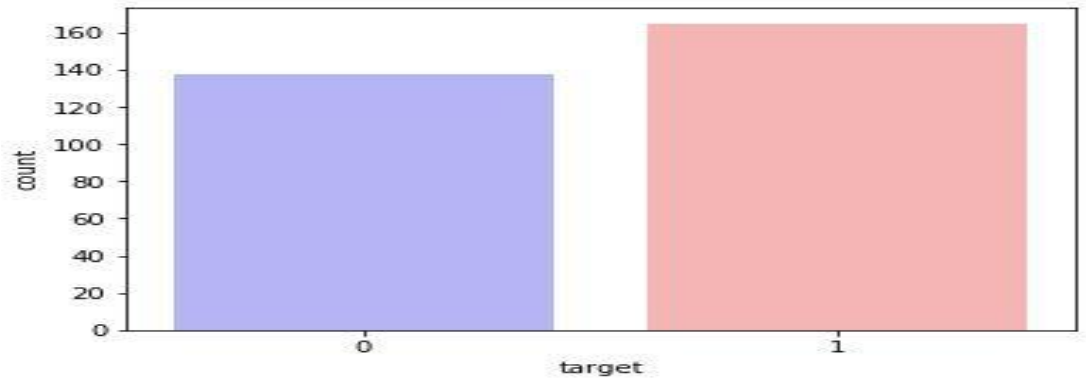
Details like the names and social security numbers of the patients were removed from the database. Only 14 attributes used:

- Age : age in years
- Sex : (1 = male; 0 = female)
- CP : chest pain type
- Trestbps : resting blood pressure (in mm Hg on admission to the hospital)
- Chol : serum cholesterol in mg/dl
- Fbs : (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
- Restecg : resting electrocardiographic results
- Thalach : maximum heart rate achieved
- Exang : exercise induced angina (1 = yes; 0 = no)
- Oldpeak : ST depression induced by exercise relative to rest
- Slope : the slope of the peak exercise ST segment
- Ca : number of major vessels (0-3) colored by fluoroscopy
- Thal : 3 = normal; 6 = fixed defect; 7 = reversible defect

- Target : 1 or 0

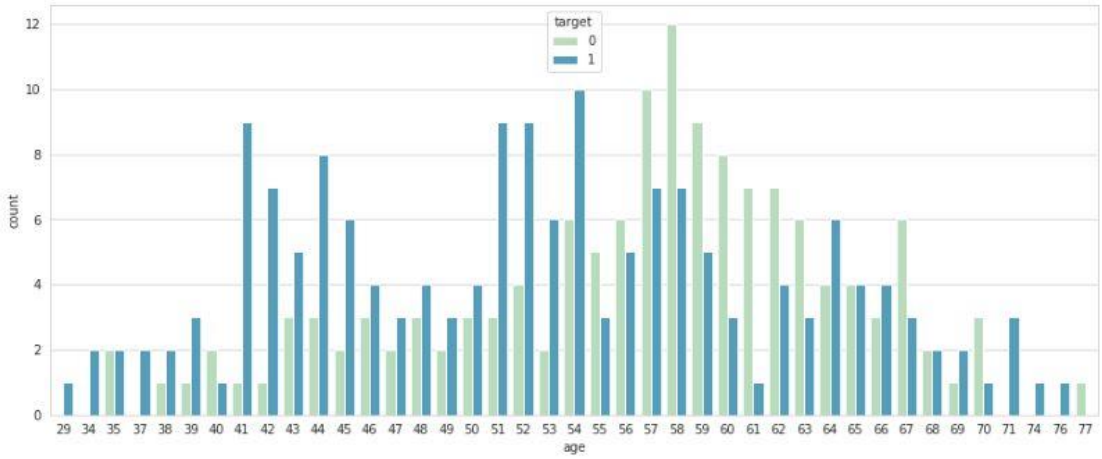
Exploratory Visualization

The plot below shows how the target is distributed. This is helpful for predicting how balanced the classes will be, whether the data is skewed and also for choosing appropriate metric.

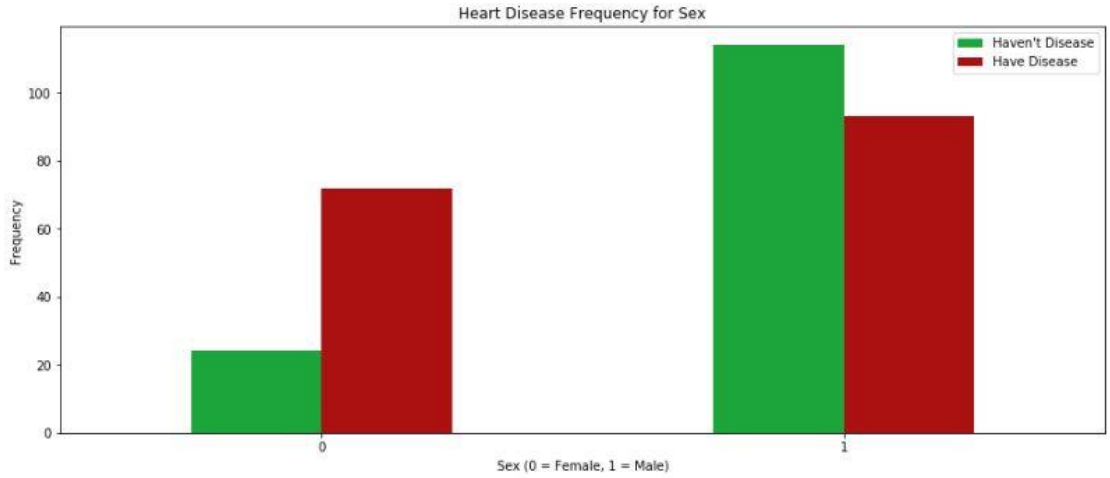


Checking features of various attributes and their relationship with target

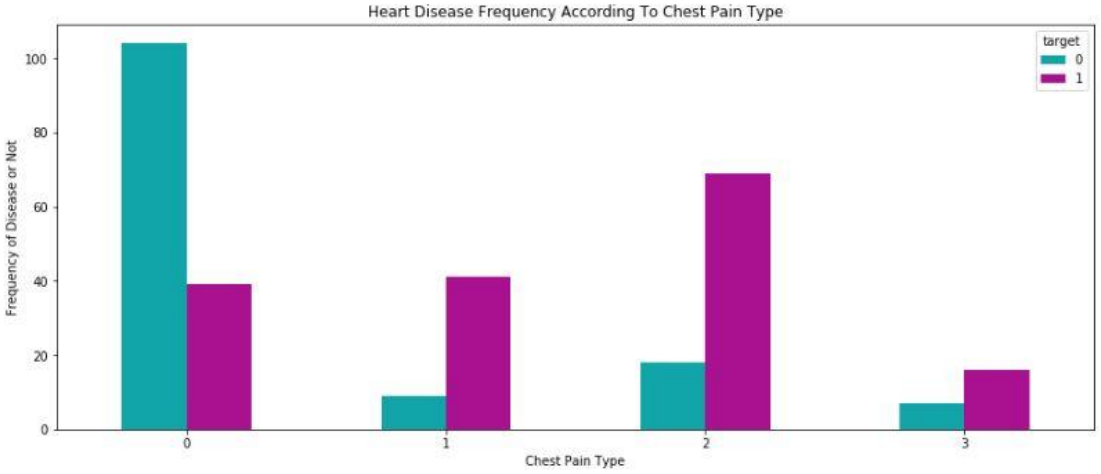
1- Age



2- **Sex**



3- Chest pain type



Algorithms and Techniques

Since this is a binary classification problem. I decided to use supervised learning techniques. I have built four supervised learning models to compare results and decide on the optimal model for this specific problem.

- K-Nearest Neighbors (KNeighbors).

  The model is very easy to understand, and often gives reasonable performance without a lot of adjustments. Using this algorithm is a good baseline method to try before considering more advanced techniques. Building the nearest neighbors model is usually very fast with such small dataset.

- Support Vector Machine (SVM).

  SVM's can model non-linear decision boundaries, and there are many kernels to choose from. They are also fairly robust against over fitting. Though it is tricky to tune the correct kernel.

- GaussianNB.

  It is not only a simple approach but also a fast and accurate method for prediction. Naive Bayes has low computation cost. And when the assumption of independence holds, a Naive Bayes classifier performs better compared to other models. And even if the NB assumption doesn't hold, a NB classifier still often does a great job in practice.

- Decision Tree Classifier.

  Perform very well in practice. They are robust to outliers, scalable, and able to naturally model non-linear decision boundaries thanks to their hierarchical structure. Yet they are prone to over fitting.

Benchmark

Some of the tests used to determine cardiovascular disease are known to produce many false positive results. For example exercise stress test, despite its low sensitivity and specificity (67% and 72%, respectively), exercise testing has remained one of the most widely used noninvasive tests to determine the prognosis in patients with suspected or established coronary disease.

III. Methodology

Data Preprocessing

The preprocessing done in the notebook consists of the following steps:

1- Checked the data for any null values.
2- Changed categorical variables to numerical variables using one hot encoding scheme.
3- Shuffle and split data into training and testing sets.

Implementation

1- Created 4 different supervised learning models
2- Trained the models using the grid search technique to optimize the hyper parameters for each model. To ensure we are producing the optimal version for each one.
3- Used cross validation with 10 shuffled sets. For each shuffled sets, and for each shuffle, 20% (`'test size'`) of the data will be used as the *validation set*.
4- Applied both previous points in the train_predict function. Which I used to return the optimal version of each model.
5- Fitted all the four models on the data. Where each model returned its recall and accuracy results for comparison.

Refinement

Almost all of the models reported good numbers for both accuracy and recall. At least when comparing the results with current test used for detecting heart disease. One of the reason why is applying grid search with almost all of them.

SVM was one of the trickiest algorithms. At first it reported a low recall score. (Around 65% which is the same as the existing tests). Also couldn't apply grid search with it. So had to try different hyper parameters manually. After trying different kernels I finally got to such good results with recall. Though it is noticeable that it affected the accuracy of the model.

IV. Results

Model Evaluation and Validation

During development, a validation set was used to evaluate the models. The final architecture and hyper parameters were chosen because they performed the best among the tried combinations.

Unfortunately the data set we have is relatively small. So even though all of the models performed relatively well. It is not guaranteed that they will return similar results with different (larger) data set.

Also some of the features we have are skewed. For example the (sex) feature has over 200 males and less than 100 female. Even though the analysis showed a relationship between gender and heart disease. We cannot trust these results.

Justification

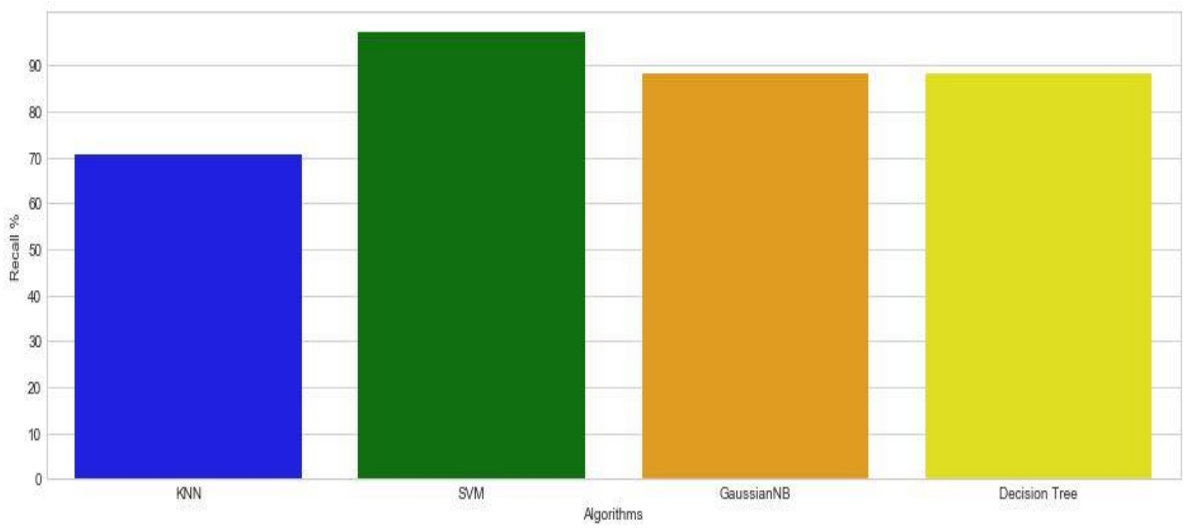
The final results for the models I created are much better than that of the results mentioned earlier. Where sensitivity was only at 67%. The worst results of the four models scored at 70.5%. The best one scored 97%.

While this data set is too small to get a result to compare. It is noticeable that we can improve on the current tests used to detect heart disease.

V. Conclusion

Free-Form Visualization

Comparing the sensitivity (recall) results for each optimized model.



- ***SVM has the highest recall score. Near to 97%.***

- ***GaussianNB and Decision Tree both come in second. With exactly the same results around 88%***

- ***KNN has the worst results when it comes to Recall. Still it scores around 70% which is higher than some of the used tests for detecting heart disease.***

Reflection

The process used for this project can be summarized using the following steps:

1- An initial problem and relevant, public datasets were found
2- The data was downloaded and preprocessed (segmented)
3- A benchmark was created for the classifier.
4- Define the models used. More than one classifier. This gave us the chance to compare results deciding on the better model.
5- The models were trained using the data (multiple times, until the optimal parameters were found for each model).
6- Comparing recall results for all models.

Step 5 was by far the most challenging. First I wanted to run each model, Find the one giving best results. Then only work towards optimizing this one model. Later I decided against this approach. Each model has its own advantage and dis-advantage. For a real life application. It would be helpful to have different models to choose from. Where the model to be used can be decided based on the situation.

Improvement

Most medical tests are evaluated based on both sensitivity and specificity. I decided to solve the issue of low sensitivity since it's considered the most urgent one in the medical field. By offering an alternative with better results in this one aspect. The model can be further improved to provide a better specificity result. This would make it a perfect test with real life application.

$$\begin{aligned}\text{specificity} &= \frac{\text{number of true negatives}}{\text{number of true negatives} + \text{number of false positives}} \\ &= \frac{\text{number of true negatives}}{\text{total number of well individuals in population}} \\ &= \text{probability of a negative test given that the patient is well}\end{aligned}$$

Again the data used for this project is relatively small. To be able to use this with in a real life situation. We have to first try the models with much larger dataset.