Capstone Project

Machine Learning Engineer Nanodegree

Heart_Disease_Prediction

Usha Sree
February 10th, 2019

I. Definition

Project Overview

Heart disease (HD) is a major cause of morbidity and mortality in the modern society. Medical diagnosis is an important but complicated task that should be performed accurately and efficiently and its automation would be very useful. All doctors are unfortunately not equally skilled in every sub specialty and they are in many places a scarce resource. A system for automated medical diagnosis would enhance medical care and reduce costs.

Nowadays, diseases are increasing day by day due to life style, hereditary. Especially, heart disease has become more common these days, i.e. life of people is at risk. Among these are poor diet, lack of regular exercise, tobacco smoking, alcohol or drug abuse, and high stress.

Heart diseases are the leading cause of death globally, resulted in 17.9 million deaths (32.1%) in 2015, up from 12.3 million (25.8%) in 1990. It is estimated that 90% of disease is preventable. There are many risk factors for heart diseases that we will take a closer look at. The prediction of heart disease requires a huge size of data which is too complex and massive to process and analyze by conventional techniques.

The main objective of this study is to find out and build the suitable machine learning technique that is computationally efficient as well as accurate for the prediction of heart disease occurrence, based on a combination of features (risk factors) describing the disease. Different machine learning classification techniques will be implemented and compared upon standard performance metric such as accuracy and F-Score.

Reference Link:

https://www.researchgate.net/publication/328031918_Machine_Learning_Classification_Techniques_for_Heart_Disease_Prediction_A_Review#pf7

Problem Statement:

The main aim of this study is to build a model that can predict the heart disease occurrence, based on a combination of features (risk factors) describing the disease. Different machine learning

classification techniques will be implemented and compared upon standard performance metric such as accuracy and F-score.

Heart Disease Prediction predicts the likelihood of patients getting heart disease.

There are several factors that increase the risk of heart disease, such as smoking habit, body cholesterol level, and family history of heart disease, obesity, high blood pressure, and lack of physical exercise.

Metrics:

Accuracy:

It is the number of correct predictions made by the model over all kinds of predictions made

$$\text{Accuracy} = (TP+TN) / (TP+FP+FN+TN)$$

Accuracy is a good measure when the target variable classes in the data are nearly balanced.

F-score:

It is used to measure a test's accuracy and it balances the use of precision and recall to do it. It can provide a realistic measure of test's performance.

The following posts will provide some methods to evaluate the performance of a machine learning problem

- https://towardsdatascience.com/choosing-the-right-metric-for-evaluating-machine-learning-models-part-2-86d5649a5428
- https://towardsdatascience.com/choosing-the-right-metric-for-machine-learning-models-part-1-a99d7d7414e4

II.Analysis:

Data Exploration:

In this data set, I have used 14 attributes and around 500 trained and test data to evaluate accuracy and F-score.

Experiments with the Cleveland database have concentrated on endeavors to distinguish disease presence (values 1, 2, 3, 4) from absence (value 0). There are several missing attribute values, distinguished with symbol '?'. The header row is missing in this dataset, so the column names have to be inserted manually as shown in the below table

```
Number of records: 503
Number of variables: 14
```

	age	sex	chest_pain	blood_pressure	serum_cholestoral	fasting_blood_sugar	electrocardiographic	max_heart_rate	induced_angina	ST_depression
0	63.0	1.0	1.0	145.0	233.0	1.0	2.0	150.0	0.0	2.3
1	67.0	1.0	4.0	160.0	286.0	0.0	2.0	108.0	1.0	1.5
2	67.0	1.0	4.0	120.0	229.0	0.0	2.0	129.0	1.0	2.6
3	37.0	1.0	3.0	130.0	250.0	0.0	0.0	187.0	0.0	3.5
4	41.0	0.0	2.0	130.0	204.0	0.0	2.0	172.0	0.0	1.4
5	56.0	1.0	2.0	120.0	236.0	0.0	0.0	178.0	0.0	0.8
6	62.0	0.0	4.0	140.0	268.0	0.0	2.0	160.0	0.0	3.6
7	57.0	0.0	4.0	120.0	354.0	0.0	0.0	163.0	1.0	0.6
8	63.0	1.0	4.0	130.0	254.0	0.0	2.0	147.0	0.0	1.4
9	53.0	1.0	4.0	140.0	203.0	1.0	2.0	155.0	1.0	3.1
10	57.0	1.0	4.0	140.0	192.0	0.0	0.0	148.0	0.0	0.4
11	56.0	0.0	2.0	140.0	294.0	0.0	2.0	153.0	0.0	1.3
12	56.0	1.0	3.0	130.0	256.0	1.0	2.0	142.0	1.0	0.6
13	44.0	1.0	2.0	120.0	263.0	0.0	0.0	173.0	0.0	0.0
14	52.0	1.0	3.0	172.0	199.0	1.0	0.0	162.0	0.0	0.5
15	57.0	1.0	3.0	150.0	168.0	0.0	0.0	174.0	0.0	1.6

Features information:

- age - age in years
- sex - sex(1 = male; 0 = female)
- chest_pain - chest pain type (1 = typical angina; 2 = atypical angina; 3 = non-anginal pain; 4 = asymptomatic)
- blood_pressure - resting blood pressure (in mm Hg on admission to the hospital)
- serum_cholestoral - serum cholestoral in mg/dl
- fasting_blood_sugar - fasting blood sugar > 120 mg/dl (1 = true; 0 = false)
- electrocardiographic - resting electrocardiographic results (0 = normal; 1 = having ST-T; 2 = hypertrophy)
- max_heart_rate - maximum heart rate achieved
- induced_angina - exercise induced angina (1 = yes; 0 = no)
- ST_depression - ST depression induced by exercise relative to rest
- slope - the slope of the peak exercise ST segment (1 = upsloping; 2 = flat; 3 = downsloping)
- no_of_vessels - number of major vessels (0-3) colored by flourosopy
- thal - 3 = normal; 6 = fixed defect; 7 = reversable defect
- diagnosis - the predicted attribute - diagnosis of heart disease (angiographic disease status) (Value 0 = < 50% diameter narrowing; Value 1 = > 50% diameter narrowing)

Types of features:

1. **Categorical features** (Has two or more categories and each value in that feature can be categorised by them): sex, chest_pain
2. **Ordinal features** (Variable having relative ordering or sorting between the values): fasting_blood_sugar, electrocardiographic, induced_angina, slope, no_of_vessels, thal, diagnosis
3. **Continuous features** (Variable taking values between any two points or between the minimum or maximum values in the feature column): age, blood_pressure, serum_cholestoral, max_heart_rate, ST_depression

Description of Dataset:

```
# Display a description of the total dataset
display(df.describe())
```

	age	sex	chest_pain	blood_pressure	serum_cholestoral	fasting_blood_sugar	electrocardiographic	max_heart_rate	induced_angina
count	503.000000	503.000000	503.000000	503.000000	503.000000	503.000000	503.000000	503.000000	503.000000
mean	54.258449	0.675944	3.176938	131.262425	248.552684	0.147117	1.035785	149.648111	0.333996
std	9.029434	0.468487	0.948108	18.046937	51.780612	0.354575	0.995364	23.208166	0.472108
min	29.000000	0.000000	1.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000
25%	47.500000	0.000000	3.000000	120.000000	212.000000	0.000000	0.000000	133.000000	0.000000
50%	55.000000	1.000000	3.000000	130.000000	245.000000	0.000000	2.000000	153.000000	0.000000
75%	61.000000	1.000000	4.000000	140.000000	277.000000	0.000000	2.000000	166.000000	1.000000
max	77.000000	1.000000	4.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000

Information of Data set:

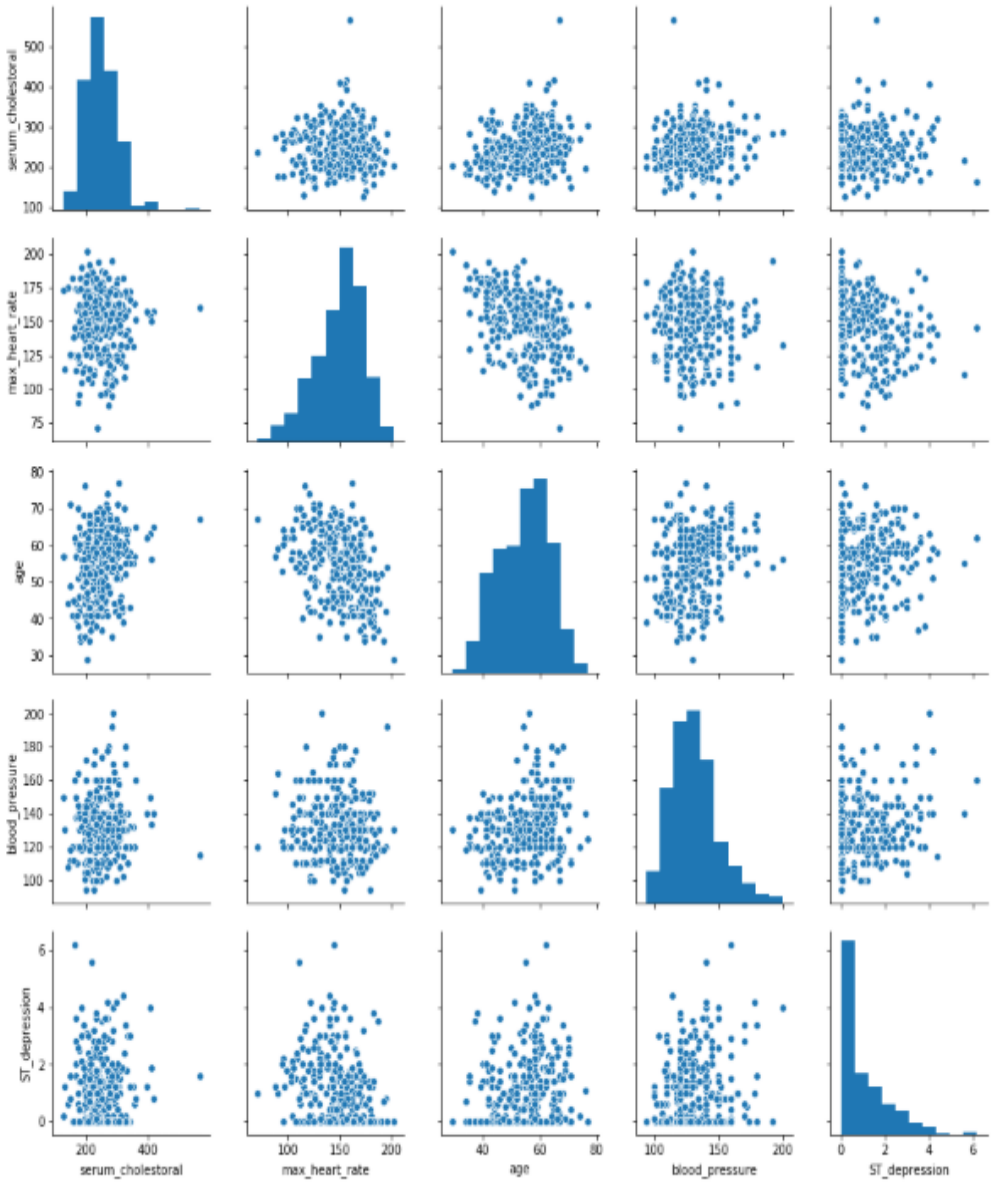
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 503 entries, 0 to 502
Data columns (total 14 columns):
age                    503 non-null float64
sex                    503 non-null float64
chest_pain             503 non-null float64
blood_pressure         503 non-null float64
serum_cholestoral      503 non-null float64
fasting_blood_sugar    503 non-null float64
electrocardiographic   503 non-null float64
max_heart_rate         503 non-null float64
induced_angina         503 non-null float64
ST_depression          503 non-null float64
slope                  503 non-null float64
no_of_vessels          497 non-null float64
thal                   500 non-null float64
diagnosis              503 non-null int64
dtypes: float64(13), int64(1)
memory usage: 55.1 KB
```

There are only 9 missing values in this dataset and all variables are recognized as numeric. From dataset description, we know, however, that most of features are categorical and it's necessary to distinguish them.

Data Visualization

Let us visualize the absolute correlation coefficient of target variable with all the other variables. Higher absolute correlation coefficient means the variable can provide more information about how the target variable moves as shown in below figure

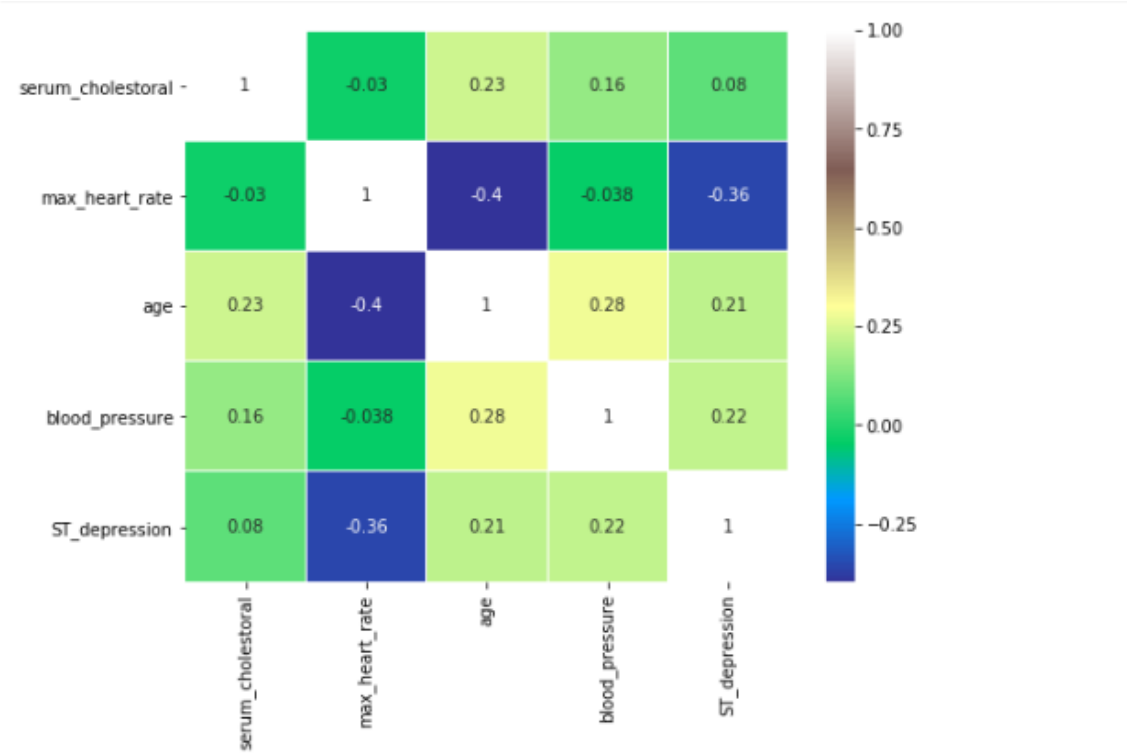


Seeing above plots, I infer that none of displayed value pairs is having an explicitly high correlation, so there is no necessity to ditch any feature at this stage. I also notice a negative correlation between 'age' and 'max_heart_rate' and positive correlation between 'age' and 'blood_pressure', what is intuitive.

A correlation matrix will make us sure whether the above is correct.

Heat Map:

The heat map is a 2-D representation of data in which values are represented by colours. A simple heat map provides the immediate visual summary of information. More elaborate heat maps allow the user to understand complex data.



Apart from two aforementioned relations, there is one more important dependency: 'max_heart_rate' and 'ST_depression'. The conclusion comes up that both features, 'age' and 'max_heart_rate', will play an important role in predicting heart disease.

Observations:

- Men are much more prone to get a heart disease than women.
- The higher number of vessels detected through fluoroscopy, the higher risk of disease.
- While soft chest pain may be a bad symptom of approaching problems with heart (especially in case of men), strong pain is a serious warning!
- Risk of getting heart disease might be even 3x higher for someone who experienced exercise-induced angina.
- The flat slope (value=2) and downslope (value=3) of the peak exercise indicates a high risk of getting disease

Modelling and Predicting with Machine Learning:

The main goal of the entire project is to predict heart disease occurrence with the highest accuracy. In order to achieve this, we will test several classification algorithms. This section includes all results obtained from the study and introduces the best performer according to accuracy metric. I have chosen several algorithms typical for solving supervised learning problems throughout classification methods.
First of all, let's equip ourselves with a handy tool that benefits from the cohesion of SciKit Learn library and formulate a general function for training our models. The reason for displaying accuracy on both, train and test sets, is to allow us to evaluate whether the model overfits or underfits the data (so-called bias/variance tradeoff).

```
def train_model(X_train, y_train, X_test, y_test, classifier, **kwargs):

    """
    Fit the chosen model and print out the score.

    """

    # instantiate model
    model = classifier(**kwargs)

    # train model
    model.fit(X_train,y_train)

    # check accuracy and print out the results
    fit_accuracy = model.score(X_train, y_train)
    test_accuracy = model.score(X_test, y_test)
    predictions_train = model.predict(X_train)
    predictions_test = model.predict(X_test)


    print "Train accuracy: {:0.2f}%".format(fit_accuracy*100)
    print "Test accuracy:{:.2f}%" .format(test_accuracy*100)
    print("Train F-Score:{}" .format(fbeta_score(y_train,predictions_train,0.5)))
    print("Test F-Score:{}" .format(fbeta_score(y_test,predictions_test,0.5)))

    return model
```

Algorithms and Techniques:

1. K-Nearest Neighbours
2. Decision Trees
3. Logistic Regression
4. Gaussian Naïve Bayes
5. Support Vector Machines
6. Random Forests

1. K-Nearest Neighbours (KNN)

K-Nearest Neighbours algorithm is a non-parametric method used for classification and regression. The principle behind nearest neighbour methods is to find a predefined number of training samples closest in distance to the new point and predict the label from these.

Advantages:

- The K-Nearest Neighbor (KNN) Classifier is a very simple classifier that works well on basic recognition problems.

Disadvantages:

- The main disadvantage of the KNN algorithm is that it is a *lazy learner*, i.e. it does not learn anything from the training data and simply uses the training data itself for classification.
- To predict the label of a new instance the KNN algorithm will find the *K* closest neighbors to the new instance from the training data, the predicted class label will then be set as the most common label among the *K* closest neighboring points.
- The algorithm must compute the distance and sort all the training data at each prediction, which can be slow if there are a large number of training examples.
- Another disadvantage of this approach is that the algorithm does not learn anything from the training data, which can result in the algorithm not generalizing well and also not being robust to noisy data.

2. Decision Tree

Given a data of attributes together with its classes, a decision tree produces a sequence of rules that can be used to classify the data.

Advantages: Decision Tree is simple to understand and visualize, requires little data preparation, and can handle both numerical and categorical data.

Disadvantages: Decision tree can create complex trees that do not generalize well, and decision trees can be unstable because small variations in the data might result in a completely different tree being generated.

3. **Logistic Regression**

Logistic regression is a machine learning algorithm for classification. In this algorithm, the probabilities describing the possible outcomes of a single trial are modelled using a logistic function.

Advantages: Logistic regression is designed for this purpose (classification), and is most useful for understanding the influence of several independent variables on a single outcome variable.

Disadvantages: Works only when the predicted variable is binary, assumes all predictors are independent of each other, and assumes data is free of missing values.

4. **Naïve Bayes**

Naive Bayes algorithm based on Bayes' theorem with the assumption of independence between every pair of features. Naive Bayes classifiers work well in many real-world situations such as document classification and spam filtering.

Advantages: This algorithm requires a small amount of training data to estimate the necessary parameters. Naive Bayes classifiers are extremely fast compared to more sophisticated methods.

Disadvantages: Naive Bayes is known to be a bad estimator.

5**. Support Vector Machine**

Support vector machine is a representation of the training data as points in space separated into categories by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

Advantages: Effective in high dimensional spaces and uses a subset of training points in the decision function so it is also memory efficient.

Disadvantages: The algorithm does not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

6.**Random Forest**

Random forest classifier is a meta-estimator that fits a number of decision trees on various sub-samples of datasets and uses average to improve the predictive accuracy of the model and controls over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement.

Advantages: Reduction in over-fitting and random forest classifier is more accurate than decision trees in most cases.

Disadvantages: Slow real time prediction, difficult to implement, and complex algorithm.

Benchmark Model:

Here we compare the final model with the remaining models to see if it got better or same or worse. The accuracy and F-scores are compared among the models and the optimal one is selected. I choose Naive Bayes model as the benchmark model. Now we will try and achieve better accuracy than this model by using the above mentioned classification models.

III. Methodology:

Pre-Processing:

In this step we will pre-process the data. Data pre-processing is considered to be the first and foremost step that is to be done before starting any process. We will read the data by using read_csv. Then we will know the shape of the data. And by using the info () we will know the information of the attributes. Then we will check whether there are any null values by using isnull (). After that we will divide the whole data into training and testing data. We will assign 70% of the data to the training data and the remaining 30% of the data into testing data. We will do this by using train_test_split from sklearn.model_selection.

Implementation:

Out of the chosen algorithms we will start with KNN classification model. We will take a classifier and fit the training data. After that we will predict that by using predict (X_train). Now we will predict the accuracy of the testing data by using accuracy score (y_test, pred) and F-score by importing fbeta_score from sklearn.metrics. By doing so for, the KNN will giive us the accuracy of 0.834.We will continue the same procedure on Naïve Bayes, SVM, Decision tree ,Logistic Regression and Random Forest . By following the same procedure above that is fitting, predicting and finding the accuracy score and F-score we will get the accuracy score and F-score as bellow.

	Accuracy	F-Score
KNN	0.8344	0.8
Decision Tree	0.8344	0.9
Logistic Regression	0.827	0.79
Naïve Bayes	0.8211	0.78
SVM	0.8476	0.82
Random Forests	0.9139	0.92

From the above reports Random Forest seems to be performing well.

Refinement

I found out random forest as the best classifier out of the chosen classifiers. Now we will perform tuning of random forest classifier in order to achieve the better accuracy. For doing refinement we will just tune

the parameter. Here we will assign n_estimators=[110,]. Now will find out the new accuracy. The new accuracy will be 0.9139.

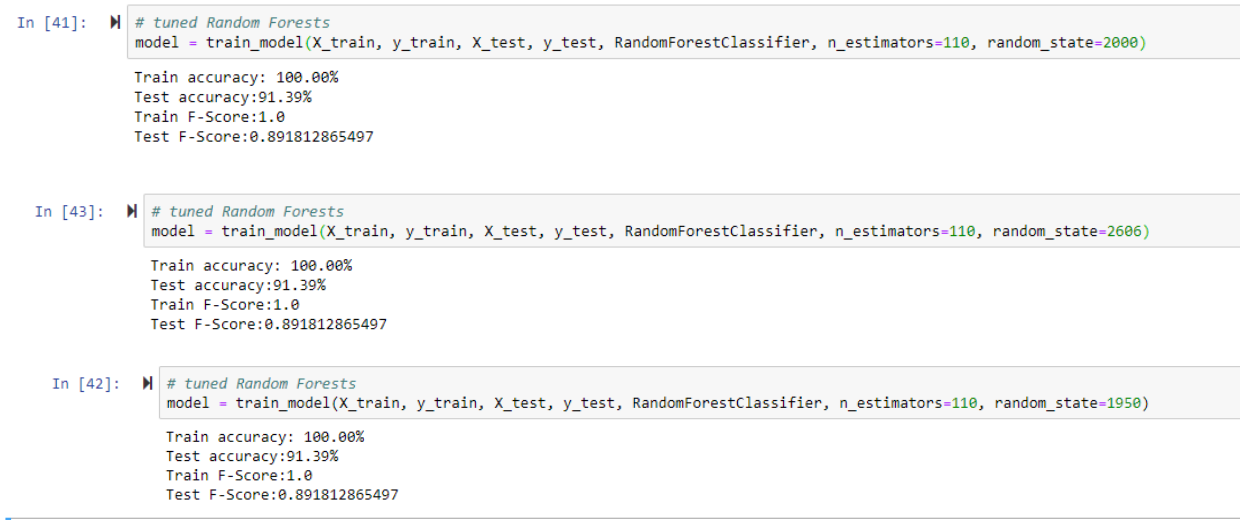
Complications:

- Chest pain (angina): When your coronary arteries narrow, your heart may not receive enough blood when demand is greatest — particularly during physical activity.
- Heart attack: If a cholesterol plaque ruptures and a blood clot forms, complete blockage of your heart artery may trigger a heart attack.
- Heart failure: If some areas of your heart are chronically deprived of oxygen and nutrients because of reduced blood flow, or if your heart has been damaged by a heart attack, your heart may become too weak to pump enough blood to meet your body's needs.
- Abnormal heart rhythm (arrhythmia): Inadequate blood supply to the heart or damage to heart tissue can interfere with your heart's electrical impulses, causing abnormal heart rhythms.
  These four factors have higher issues to get heart disease and increase the complication level. These attributes play a major role for diagnosis of heart disease.

IV. Result

Model evaluation and validation

The final model we have chosen is tuned random forest which gave us more accuracy that is 0.9139 and F-score value is 0.89. In order to achieve this accuracy we assigned n_estimators=[110] and F-score by importing fbeta_score from sklearn.metrics . Here we can say that the solution is reasonable because we are getting much more less accuracy while using other models. The final model that is tuned random forest has been tested with various inputs to evaluate whether the model generalises well. This model is also robust enough for the given problem. We can say this by testing it over different random sates. From this we can say that Small changes in the training data will not affect the results greatly. So the results found from this model can be trusted.



```
In [45]:  # tuned Random Forests
          model = train_model(X_train, y_train, X_test, y_test, RandomForestClassifier, n_estimators=110, random_state=160000)

          Train accuracy: 100.00%
          Test accuracy:91.39%
          Train F-Score:1.0
          Test F-Score:0.891812865497
```

```
In [81]:  # tuned Random Forests
          model = train_model(X_train, y_train, X_test, y_test, RandomForestClassifier, n_estimators=110, random_state=2500)

          pd.Series(model.feature_importances_,X.columns).sort_values(ascending=True).plot.barh()

          Train accuracy: 100.00%
          Test accuracy:90.73%
          Train F-Score:1.0
          Test F-Score:0.887573964497
```

Justification:

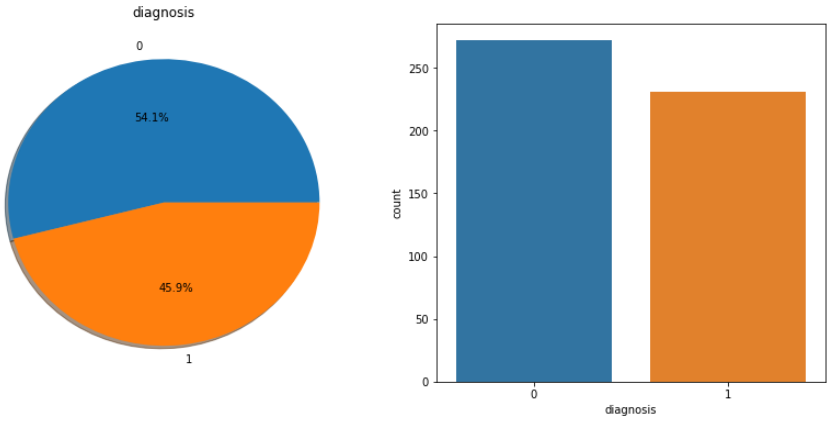
My final model's solution is better than the benchmark model.

Tuned	Random Forest	Benchmark Model
Accuracy	0.9139	0.8344

From the above we can conclude that the results for the final model are stronger than the benchmark model. Hence we can say that tuned random forest provides the significant to solve the problem of predicting heart diseases.

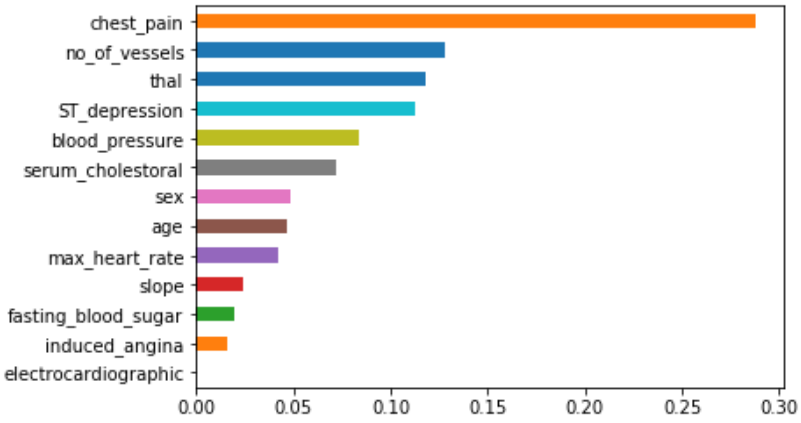
V. Conclusion:

The goal of the project was to compare different machine learning algorithms and predict if a certain person, given various personal characteristics and symptoms, will get heart disease or not. Here are the final results.



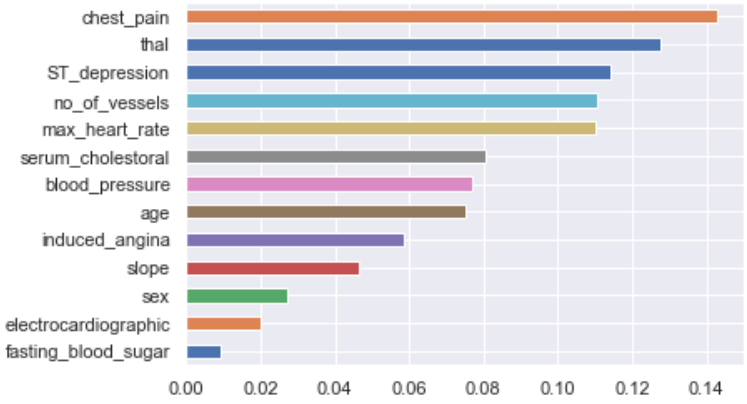
Now the distribution of target value is almost equal, so using standard metrics in further machine learning modelling like *accuracy* and *AUC* is justified

```
Out[30]: <matplotlib.axes._subplots.AxesSubplot at 0x130a87b8>
```



- Variable 'thal' turns out to be a significantly important feature.
- Remember my hypothesis that 'fasting_blood_sugar'' is a very weak feature? Above graph confirms this clearly.
- Decision tree model learns the train set perfectly, and at the same time is entirely overfitting the data, what results in poor prediction. Other values of 'max_depth' parameter need to be tried out.

```
Out[78]: <matplotlib.axes._subplots.AxesSubplot at 0xfb1c748>
```



While it is typical for Random Forests to perfectly learn and fit into training data, the test accuracy achieved outstanding 91.39%!

```
# initialize an empty list
accuracy1 = []
accuracy2 = []
# list of algorithms names
classifiers = ['KNN', 'Decision Trees', 'Logistic Regression', 'Naive Bayes', 'SVM', 'Random Forests']

# list of algorithms with parameters
models = [KNeighborsClassifier(n_neighbors=5), DecisionTreeClassifier(max_depth=6, random_state=2606), LogisticRegression(),
        GaussianNB(), SVC(C=0.05, kernel='linear'), RandomForestClassifier(n_estimators=110, random_state=2606)]

# loop through algorithms and append the score into the list
for i in models:
    model = i
    model.fit(X_train, y_train)
    score = model.score(X_test, y_test)
    score1=model.score(X_train,y_train)
    accuracy1.append(score1)
    accuracy2.append(score)
```

In [117]:
```
# create a dataframe from accuracy results
summary = pd.DataFrame({ 'Train_Accuracy':accuracy1, 'Test_Accuracy':accuracy2}, index=classifiers)
summary
```

Out[117]:

	Test_Accuracy	Train_Accuracy
KNN	0.834437	0.906250
Decision Trees	0.834437	0.934659
Logistic Regression	0.827815	0.872159
Naive Bayes	0.821192	0.877841
SVM	0.847682	0.857955
Random Forests	0.913907	1.000000

It does not come as a surprise that the more complex algorithms like SVM and Random Forests generated better results compared to the basic ones. It is worth to emphasize that in most cases hyperparameter tuning is essential to achieve robust results out of these techniques. By producing decent results, simpler methods proved to be useful as well.

Machine learning has absolutely bright future in medical field. Just imagine a place where heart disease experts are not available. With just basic information about a certain patient's medical history, we may quite accurately predict whether a disease will occur or not.

Reflection:

1. I have learnt how to visualize and understand the data.

2. I have learnt that the data cleaning place a very vital role in data analytics.

3. Removing the data features which are not necessary in evaluating model is very important.

4. I got to know how to use the best technique for the data using appropriate ways

5. I got to know how to tune the parameters in order to achieve the best score.

6. On a whole I learnt how to graph a dataset and applying cleaning techniques on it and to fit the best techniques to get best score.

Improvement:

Artificial Neural Network (ANN) performed well in most models for predicting heart disease as well as Decision Tree (DT). Finally, the field of using machine learning for diagnosing heart disease is an important field, and it can help both healthcare professionals and patients. It is still a growing field, and despite the massive availability of patient data in hospitals or clinics, not much of it is published.

Since the quality of the dataset is an essential factor in the prediction's accuracy, more hospitals should be encouraged to publish high-quality datasets (while protecting the privacy of patients) so that researchers can have a good source to help them develop their models and obtain good results.