Machine Learning Nanodegree

Capstone Report

A Recommender System for Heart Diseases

Anirban Paul

June 9th, 2018

1 Definition

Project Overview

In today's world, heart diseases are the primary cause of the increasing death rate worldwide. Among the non-infectious diseases, the number of deaths due to heart disease is the highest. Huge amounts of data are generated every day in medical centers and hospitals. Valuable information is contained in these data that can be used to discover hidden patterns and arrive at meaning and accurate conclusions.

Much research has been carried out, in the past few years, in the assessment of risk due to heart diseases, to make decisions that are effective as well as safe. For the prediction of various diseases, data mining techniques and statistical tools have been used worldwide. However, till date, an analytical tool that is effective and has a high accuracy hasn't been proposed or implemented. To help fill the void, I am going to develop a heart disease recommender system. Recommender systems are very powerful tools that are user friendly and provide germane suggestions to users that are quite accurate, and can help create a positive impact.

I would be using three machine learning approaches for building the model for the prediction of the heart diseases. I will be using Random Forest, Neural Network and Gradient Boosting to predict the possibility of heart attack for given features of the patients. Amongst these three models the Neural Network Model will serve as a benchmark model and an attempt will be made to build at least one model which outperforms the benchmark model at least at par with the benchmark model, which I am looking forward to achieve with the other two model (Random Forest and Gradient Boosting). The dataset that I have used for my heart disease recommender system is Cleveland Heart Disease dataset that is provided by the University of California, Irvine (UCI) Machine Learning Repository, containing 14 features such as age, sex, type of chest pain etc. I will be using three different algorithms to classify the given data set.

Random forests is an ensemble machine learning technique for classification and regression it works by forming a various levels of decision trees at training time and outputs the class that is

the mode of the classes (classification) or mean prediction (regression) of the individual trees. It was created by Tin Kam Ho.

Gradient boosting is a machine learning method created by Leo Breiman for regression and classification related problems, which produce mostly decision tree in the form of an ensemble of weak models. . It builds stage-wise models like other boosting approach and it allows optimization of differentiable loss function.

Neural networks are an important machine learning method in which the algorithms are implemented by using the structure of neural networks which model the data using artificial neurons. Neural networks are like the functioning of the brain, which forms a biological neural network.

The motivation for this project was from an academic paper by Karayilan, "Prediction of Heart Disease Using Neural Network", IEEE 2017.

Problem Statement

In this project I will be using three machine learning technique to develop a recommender system for heart diseases. The three machine learning techniques are Random Forest, Neural Network and Gradient Boosting. These methods will be used to predict the chance of having a heart disease depending upon few features. Amongst these three models the Neural Network Model will serve as a benchmark model and an attempt will be made to build at least one model which outperforms the benchmark model at least at par with the benchmark model, which I am looking forward to achieve with the other two models (Random Forest and Gradient Boosting). The problem is a classification problem, in which the main task is to classify by predicting whether there is a chance of disease or not. There will various input features like Age, Serum cholesterol, Chest pain type etc. which will be obtained from the datasets and there will be predicting value which will output whether heart disease is present or not.

Metrics

This project will be evaluated with regards to the model's ability to predict the chance of having a heart disease or not. I expect that the two model that I am going to use other than the neural network which will be my benchmark model among them at least one model amongst them to perform better than the benchmark model. The project should be able to train successfully on random training data and then test the model and produce an accurate prediction based on features and the true positive and negative and false positives and negatives will help us to produce the F1 score. I will make an attempt to develop this model with better F1 score which than the benchmark model. So F1 score is the evaluation metrics.

2 Analysis

Data Exploration

The dataset that I have used for my heart disease recommender system is Cleveland Heart Disease dataset, which is provided by the University of California, Irvine (UCI) Machine

Learning Repository, containing 14 features such as age, sex, type of chest pain etc. I will be using three different algorithms to classify the given data set. The algorithms are known for its ability to classify complex data. Data from the UCI data repository is divided into training and testing data. The dataset used here is part of a database containing 14 features from Cleveland Clinic Foundation for heart disease. This dataset happens to be the most commonly used database by machine learning researchers. The dataset shows different levels of heart disease presence from 1 to 4 and 0 for the absence of the disease. It has 303 rows of people data with 13 continuous observations of different symptoms. The dataset has some missing values in it. Firstly, missing data are imputed in each feature. Then the data is split into training and testing data. I will be looking into different classic machine learning models, and their discoveries in diseases risks. I will be developing three classifications using Random Forest, Gradient Boosting and Neural Network, on Cleveland dataset.

Thirteen of the attributes listed below are used as input data for the algorithms. The last attribute, "num" which happens to be the predicting value, is used as output data for the algorithms. The "num" has values from zero to four. Number Zero means absence of a heart disease, and the other numbers show the presence and the level of heart disease. So, output of the algorithms is designed in a binary way in which zero would indicate that the heart disease is absent and one would indicate that the heart disease is present. The data is split 70 percentage of training data and 30 percentage of testing data and the split is done using train test split which randomly splits the data into train and test subsets. Description about all the features used for this project is given in the table below.

Clinical Features	Description
Age	Age
Ca	Number of major vessels (0-3) colored by flourosopy
Chol(mg/dl)	Serum cholesterol
Cp	Chest pain type
Exang	Exercise induced angina
Fbs	Fasting blood sugar
Num	Diagnosis of heart disease
Oldpeak	ST depression induced by exercise relative to rest
Restecg	Resting electrocardiographic results
Sex	Gender
Slope	The slope of the peak exercise ST segment
Thal	3=normal ; 6 = fixed defect; 7= reversible defect
Thalach	Maximum heart rate achieved
Trestbps(mmHg)	Resting Blood Pressure

The first five rows of the initial data are given below:

	age	sex	cp	trestbps	cholestrol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
0	63	1	1	145	233	1	2	150	0	2.3	3	0	6	0
1	67	1	4	160	286	0	2	108	1	1.5	2	3	3	2
2	67	1	4	120	229	0	2	129	1	2.6	2	2	7	1
3	37	1	3	130	250	0	0	187	0	3.5	3	0	3	0
4	41	0	2	130	204	0	2	172	0	1.4	1	0	3	0

The link to the dataset is http://archive.ics.uci.edu/ml/datasets/heart+disease

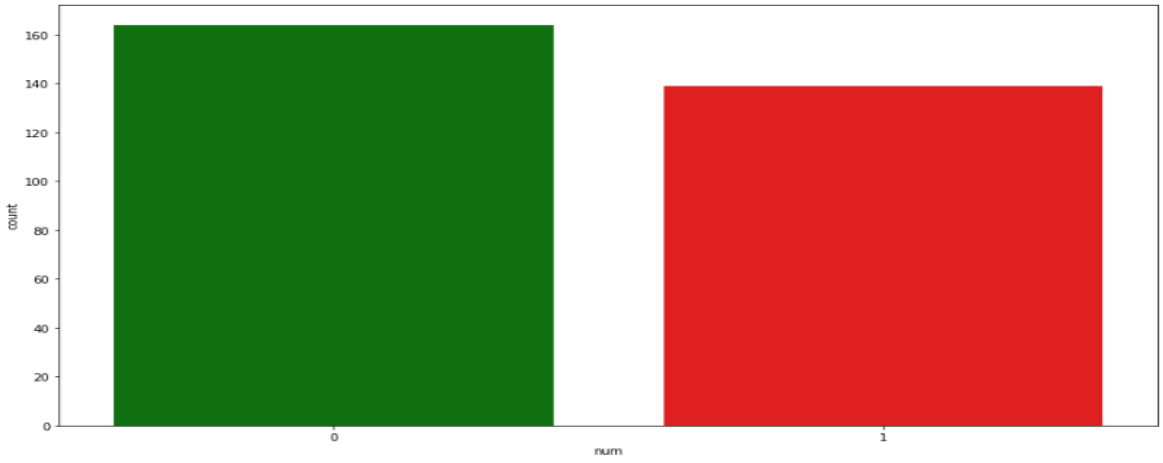
Exploratory Visualization

The dataset has 14 features such as age, sex, type of chest pain etc. From the above data exploration it is clear that the last column ‘num’ in the data is the indication of heart disease present or not with ‘0’ being absence of heart disease and ‘1’,’2’,’3’ and ’4’ indicates number pf heart disease present. The main aim of the project is to predict heart disease there or not so the values ‘1’,’2’,’3’ and ’4’ all are converted into ‘1’ which indicates heart disease is present. So after replacing the first five rows of the data looks like this:-

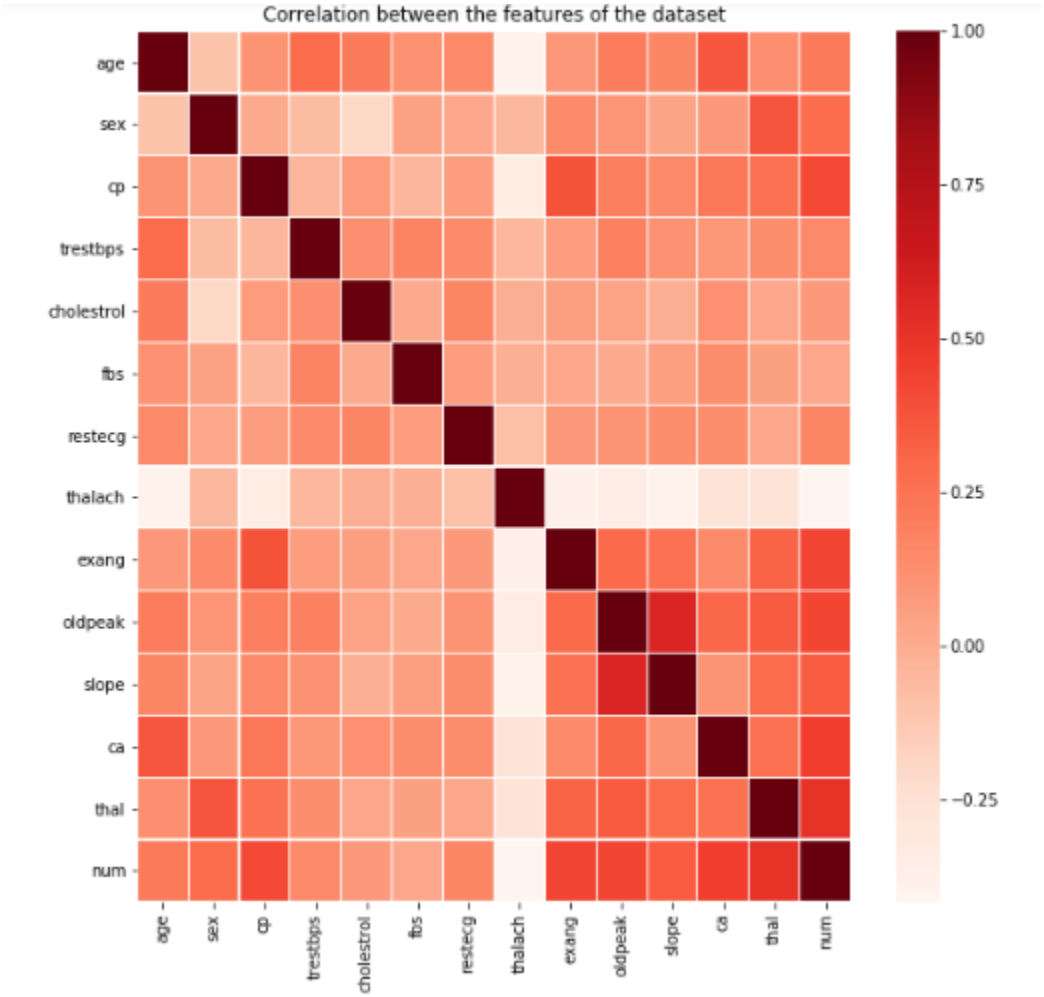
	age	sex	cp	trestbps	cholestrol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
0	63	1	1	145	233	1	2	150	0	2.3	3	0	6	0
1	67	1	4	160	286	0	2	108	1	1.5	2	3	3	1
2	67	1	4	120	229	0	2	129	1	2.6	2	2	7	1
3	37	1	3	130	250	0	0	187	0	3.5	3	0	3	0
4	41	0	2	130	204	0	2	172	0	1.4	1	0	3	0

The count percentage of ‘num’ value has been visualized with the help of a bar chart using Count Plot.

```
Heart Disease Present Percentage       : 54.12541254125413
Heart Disease Absent Percentage        : 45.87458745874587
```



The data after preprocessing is ready for exploration. The following visualization is used to show the correlation between the features.



The above matrix is used to show the strong correlation between the features that is to be used for the dataset. ‘Num’ is strongly related with ‘thal’, ‘ca’ and even ‘oldpeak’ is also strongly related with ‘slope’.

Algorithms and Techniques

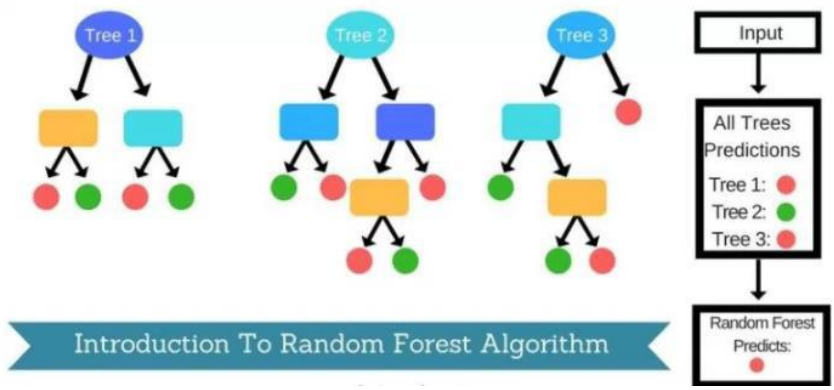
The three algorithms that I will be using to classify the dataset are:

a)Random Forest Algorithm

Random forest is a supervised classification algorithm. It is an ensemble machine learning technique for classification and regression it works by forming a various levels of decision trees at training time and outputs the class that is the mode of the classes

(classification) or mean prediction (regression) of the individual trees. More number of trees more accuracy of the algorithm.

Layman explanation of how random forest works:



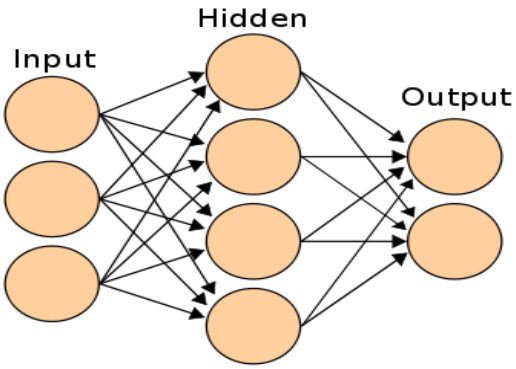
Suppose you would like to watch a movie in your laptop. You would like to watch something that will interest you. So you ask your best friend about any suggestions. Naturally, he will ask you the kind or genre of movies you liked in the past. Based on your answer, he will recommend a movie to you. Your best friend may recommend his best movie to you as a friend. The model will be biased with the closeness of your friendship. So you decide to ask few more friends to recommend a movie for you. Now, each one of them will ask you about the movies you liked, and finally recommend a movie they might think that you would like. Finally, you as a person will watch a movie that has the highest number of votes from your friends. In the above movie planning, two algorithms were used, decision tree and random forest.

b) Gradient Boosting Algorithm

Gradient boosting is a machine learning method created by Leo Breiman for regression and classification related problems, which produce mostly decision tree in the form of an ensemble of weak models. . It builds stage-wise models like other boosting approach and it allows optimization of differentiable loss function.

c) Neural Network(Benchmark Model)

Neural networks are an important machine learning method in which the algorithms are implemented by using the structure of neural networks which model the data using artificial neurons. Neural networks are like the functioning of the brain, which forms a biological neural network.



Benchmark Model

I will be looking into different classic machine learning models, and their discoveries in diseases risks. I will be developing three classifications using Random Forest, Gradient Boosting and Neural Network, on Cleveland dataset. Amongst these three models the Neural Network Model will serve as a benchmark model with F score of 0.791 and accuracy of 79.12% and an attempt will be made to build at least one model which outperforms the benchmark model at least at par with the benchmark model, which I am looking forward to achieve with the other two models (Random Forest and Gradient Boosting). In simple terms the neural network model on the project data is going to act as a benchmark model for my project and an attempt will be made to build a model with better accuracy and F1 score than the simple neural network model.

Benchmark model neural network F score=0.791

3 Methodology

Data Preprocessing

As it has been mentioned in Data exploratory that the dataset that I have used for my heart disease recommender system is Cleveland Heart Disease dataset, that is provided by the University of California, Irvine (UCI) Machine Learning Repository, containing 14 features such as age, sex, type of chest pain etc. The last column ‘num’ in the data is the indication of heart disease present or not with ‘0’ being absence of heart disease and ‘1’,’2’,’3’ and ’4’ indicates number pf heart disease present. The main aim of the project is to predict heart disease there or not so the values ‘1’,’2’,’3’ and ’4’ all are converted into ‘1’ which indicates heart disease is present. There are some missing values and ‘Not a number’ case in the dataset which are replace with ‘0’ in the following ways.

In [5]:
```
def replace_q(x):
    if x == '?':
        x = 0
    else:
        return x
```

In [6]:
```
train_df['thal'] = train_df['thal'].apply(lambda x : replace_q(x))
```

In [7]:
```
train_df['ca'] = train_df['ca'].apply(lambda x : replace_q(x))
train_df['slope'] = train_df['slope'].apply(lambda x : replace_q(x))
train_df['oldpeak'] = train_df['oldpeak'].apply(lambda x : replace_q(x))
train_df['exang'] = train_df['exang'].apply(lambda x : replace_q(x))
train_df['thalach'] = train_df['thalach'].apply(lambda x : replace_q(x))

train_df['restecg'] = train_df['restecg'].apply(lambda x : replace_q(x))

train_df['fbs'] = train_df['fbs'].apply(lambda x : replace_q(x))

train_df['cholestrol'] = train_df['cholestrol'].apply(lambda x : replace_q(x))
train_df['trestbps'] = train_df['trestbps'].apply(lambda x : replace_q(x))

train_df['cp'] = train_df['cp'].apply(lambda x : replace_q(x))
```

In [8]:
```
train_df['trestbps'] = train_df['trestbps'].fillna('0')
train_df['cholestrol'] = train_df['cholestrol'].fillna('0')
train_df['fbs'] = train_df['fbs'].fillna('0')
train_df['restecg'] = train_df['restecg'].fillna('0')
train_df['thalach'] = train_df['thalach'].fillna('0')
train_df['exang'] = train_df['exang'].fillna('0')
train_df['oldpeak'] = train_df['oldpeak'].fillna('0')
train_df['slope'] = train_df['slope'].fillna('0')
train_df['ca'] = train_df['ca'].fillna('0')
train_df['thal'] = train_df['thal'].fillna('0')
train_df['cp'] = train_df['cp'].fillna('0')
```

Implementation

This project comprises of three different machine learning algorithms they are Random Forest, Neural Network and Gradient Boosting to predict the possibility of heart attack for given features of the patients. Each of these algorithms will be composed of these steps

Data Exploration: Visualizing the dataset, detect outliers, remove null values, cleaning the dataset, check relevance of every column to the target column for example the dataset has some missing values in it. Firstly, missing data are imputed in each feature. Then the data is split into training and testing data. I will be looking into different classic machine learning models, and their discoveries in diseases risks. I will be developing three classifications using Random Forest, Gradient Boosting and Neural Network, on Cleveland dataset. Amongst these three models the Neural Network Model will serve as a benchmark model and an attempt will be made to build at least one model which outperforms the benchmark model at least at par with the benchmark model, which I am looking forward to achieve with the other two models (Random Forest and Gradient Boosting After data preprocessing I will be splitting the training dataset into training(70 percent) and testing(30 percent) sets using train test split etc. The split is done in the following ways:

Train Test Split

In [15]:
```
from sklearn.model_selection import train_test_split
```

In [16]:
```
train,test = train_test_split(train_df,test_size=0.3,random_state=42)
len(train)
```

Out[16]:
```
212
```

In [17]:
```
features = train[ftcol].values
label = train['num'].values
```

Before applying the algorithms the initial data frame is modified by removing the last column which contains whether heart disease is present or not.

Training: After randomly splitting the dataset into training and testing sets, I trained the models on the samples.

The benchmark model is a simple neural network model which generates tensor with normal distribution. It has 13 features as input and then I am going to use various hidden layers followed by regularization of the data using Dropout to prevent overfitting of the data and sigmoid activation function in final layer to obtain a classified output. Other two models will involves Random Forrest classifier and Gradient Boosting classifier which will be developed with an attempt to produce a better F1 score than the simple neural network model.

For gradient boosting the algorithm is applied to the dataset in the following way.

Grading Boosting Classifier

```
In [18]: from sklearn.ensemble import GradientBoostingClassifier

In [19]: gbc = GradientBoostingClassifier(n_estimators=1250,learning_rate=0.05,verbose=1,max_depth=10,max_features = 0.8,min_samples_leaf=

In [20]: pred = gbc.fit(features, label)
         pred = pred.predict(test[ftcol].values)
```

For random forest the algorithm is applied to the dataset in the following way.

```
In [33]: from sklearn.ensemble import RandomForestClassifier

In [34]: rfc = RandomForestClassifier(n_estimators=1500,random_state=42,max_depth=None,n_jobs=3,verbose=1)

In [35]: predrfc = rfc.fit(features,label).predict(test[ftcol].values)
[Parallel(n_jobs=3)]: Done  44 tasks      | elapsed:    0.0s
[Parallel(n_jobs=3)]: Done 194 tasks      | elapsed:    0.2s
[Parallel(n_jobs=3)]: Done 444 tasks      | elapsed:    0.4s
[Parallel(n_jobs=3)]: Done 794 tasks      | elapsed:    0.7s
[Parallel(n_jobs=3)]: Done 1244 tasks     | elapsed:    1.0s
[Parallel(n_jobs=3)]: Done 1500 out of 1500 | elapsed:    1.2s finished
[Parallel(n_jobs=3)]: Done  44 tasks      | elapsed:    0.0s
[Parallel(n_jobs=3)]: Done 194 tasks      | elapsed:    0.0s
[Parallel(n_jobs=3)]: Done 444 tasks      | elapsed:    0.1s
[Parallel(n_jobs=3)]: Done 794 tasks      | elapsed:    0.1s
[Parallel(n_jobs=3)]: Done 1244 tasks     | elapsed:    0.3s
[Parallel(n_jobs=3)]: Done 1500 out of 1500 | elapsed:    0.3s finished
```

Testing: After successful training of the model I tested them to determine the F1 score of the models using the model info. While testing the target label column name for gradient boosting is 'predictgradient' and for random forest is 'rfcPredict'.

	age	sex	cp	trestbps	cholestrol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num	predictgradient	rfcPredict
179	53	1	3	130	246	1	2	173	0	0.0	1	3	3	0	0	0
228	54	1	4	110	206	0	2	108	1	0.0	2	1	3	1	1	1
111	56	1	4	125	249	1	2	144	1	1.2	2	1	3	1	1	1
246	58	1	4	100	234	0	0	156	0	0.1	1	1	7	1	1	1
60	51	0	4	130	305	0	0	142	1	1.2	2	0	7	1	1	1

As discussed I am using the dataset which is a part of a database containing 14 features from Cleveland Clinic Foundation for heart disease. I am using Jupyter Notebook for the development of the project. The code has been written in Python language. The data have bern preprocessed. Missing values and not a number case have been taken care of. The preprocessing involved converting all the missing value data to be replaced with '0' and not a number case is also to be replaced with zero. Then we classified the data according to the three algorithms discussed earlier. The F1 score of each algorithm is then observed.

This system is fast and extremely reliable and would help in overcoming the problems of today's world. The algorithms discussed and used for achieving the aim of the project could motivate researchers and fellow data scientists to use these algorithms in their research of heart disease in future.

Refinement

After running the project first with the benchmark neural network model, I got F score value of 0.7912 and 79.12% of accuracy. Then I applied Gradient Boosting with 'n_estimators'=1250 and 0.05 learning rate and got F score value of 0.8235 and 83.51% of accuracy which is higher than the benchmark model. Then I applied Random Forest with 'n_estimators'=1500 and got F score value of 0.847 and 85.71% of accuracy which is higher than the benchmark model and Gradient boosting. So the final model for this system is Random Forests Algorithm.

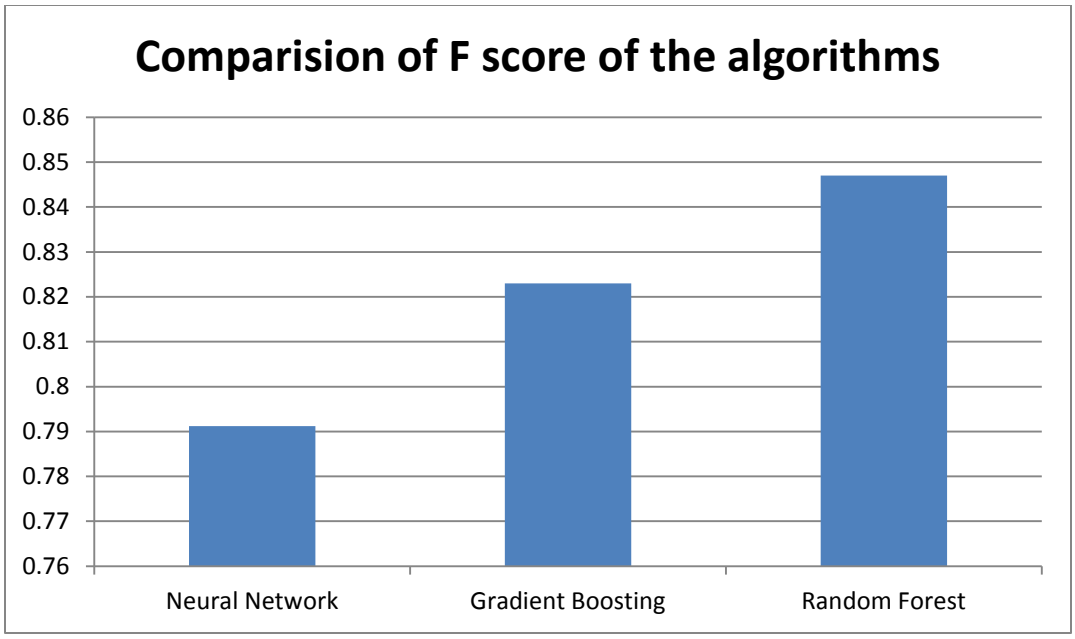
4 Results and discussion

Model Evaluation and Validation

I have evaluated each classification model based on their F-score. From the beginning of the project my the aim of the project was to make a recommender for heart diseases which will have higher F-score than the benchmark model(Neural Network), so that being the important aspect of the project. On the basis of their F-score I have provided a bar chart. F-score results for all three algorithms are as follows:
a) Neural Network: 0.791

b) Gradient Boosting Algorithm: 0.823

c) Random Forest Algorithm: 0.847

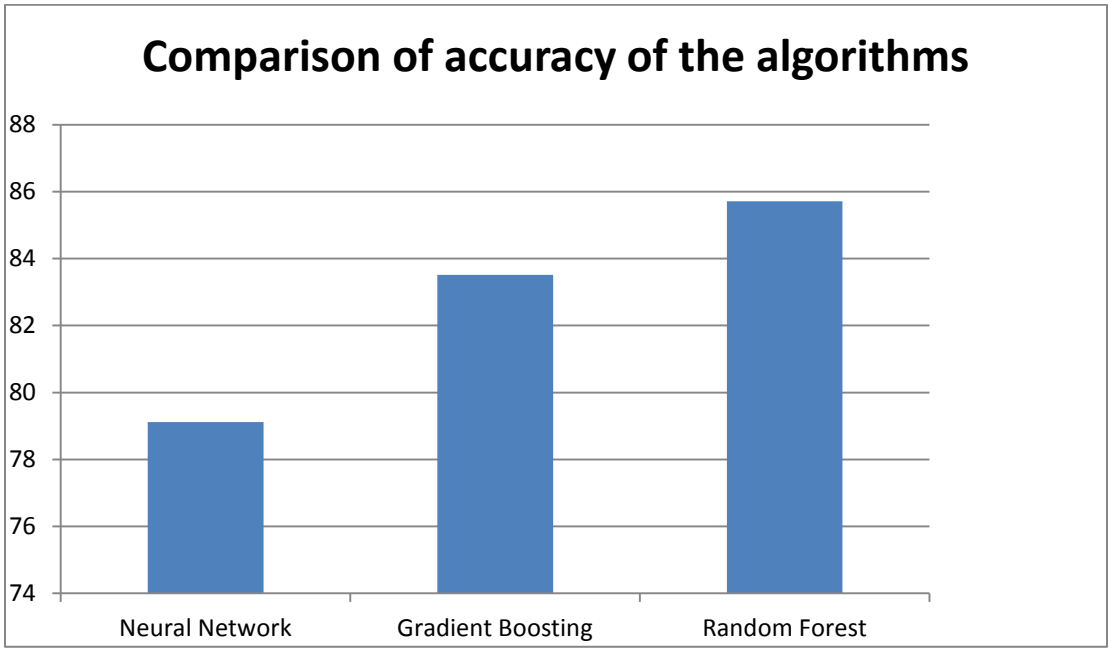


Accuracy results for all three algorithms are as follows:

a) Gradient Boosting Algorithm: 83.51%

b) Random Forest Algorithm: 85.71%

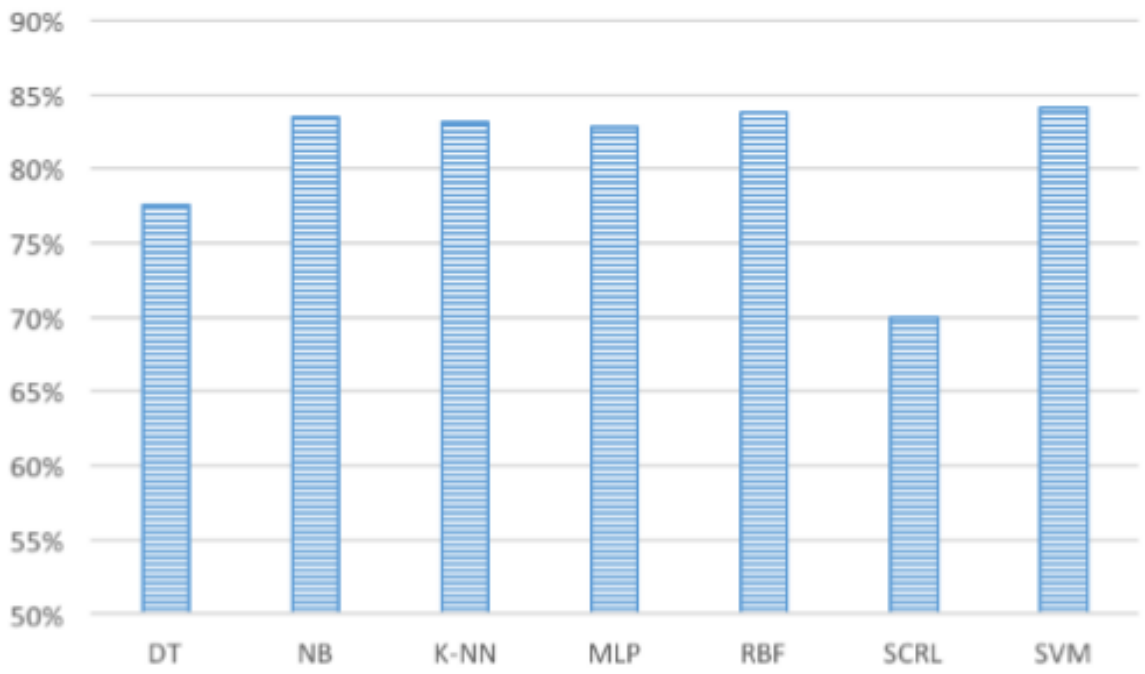
c) Neural Network: 79.12%

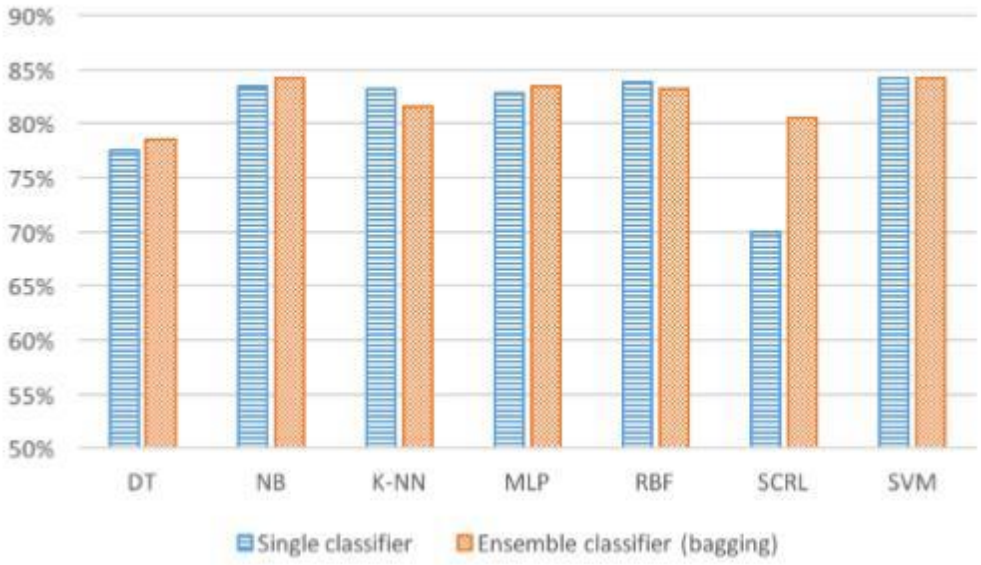


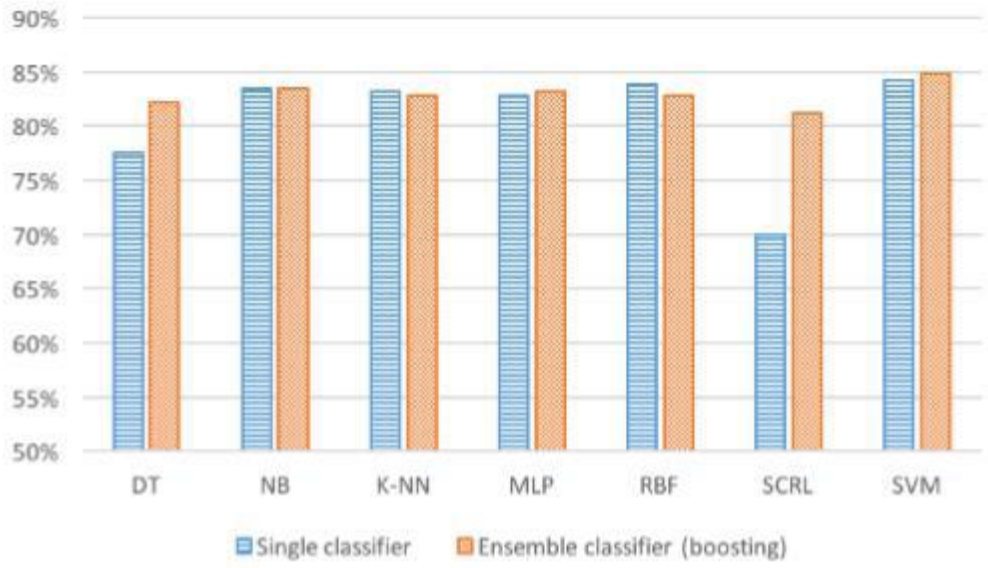
Justification

I would like to justify my project implementation results by referring to the paper "A Comprehensive Investigation and Comparison of Machine Learning Techniques in the Domain of Heart Disease" by Pouriyeh, the author presented a comparison between the different classification methods used on the same dataset that I have used in this project. The comparison results are as follows:

Classifier	Precision	Recall	F-Measure	ROC Area	Accuracy (%)
Decision Tree(DT)	0.774	0.830	0.801	0.800	77.55
Naïve Bayes (NB)	0.836	0.867	0.851	0.904	83.49
K Nearest Neighbor (K-NN, K=1)	0.782	0.782	0.782	0.752	76.23
K Nearest Neighbor (K-NN, K=3)	0.821	0.836	0.829	0.838	81.18
K Nearest Neighbor (K-NN, K=9)	0.848	0.842	0.845	0.898	83.16
K Nearest Neighbor (K-NN, K=15)	0.847	0.836	0.841	0.904	82.83
MultiLayer Perceptron (MLP)	0.824	0.824	0.824	0.894	82.83
Radial Basis Function (RBF)	0.845	0.861	0.853	0.892	83.82
Single Conjunctive Rule Learner (SCRL)	0.734	0.703	0.718	0.707	69.96
Support Vector Machine (SVM)	0.827	0.897	0.860	0.836	84.15







Based on these results the author claims that SVM algorithm with boosting technique is the best. However, my Random Forest Algorithm has achieved an accuracy of 85.71%. **Hence, my final algorithm Random Forest has outperformed all the seven algorithms, individually and also when boosting and bagging was applied to each one of them.** 24

In the paper 'An Intelligent Recommender System based on Short-term Risk Prediction for Heart Disease Patients' by Lafta, the author has claimed that his accuracy ranges from 75% to 100%. It does not have a fixed accuracy every time. Its accuracy can be put for a debate. **My algorithms perform consistently, giving accurate results every time.**

5 Conclusion

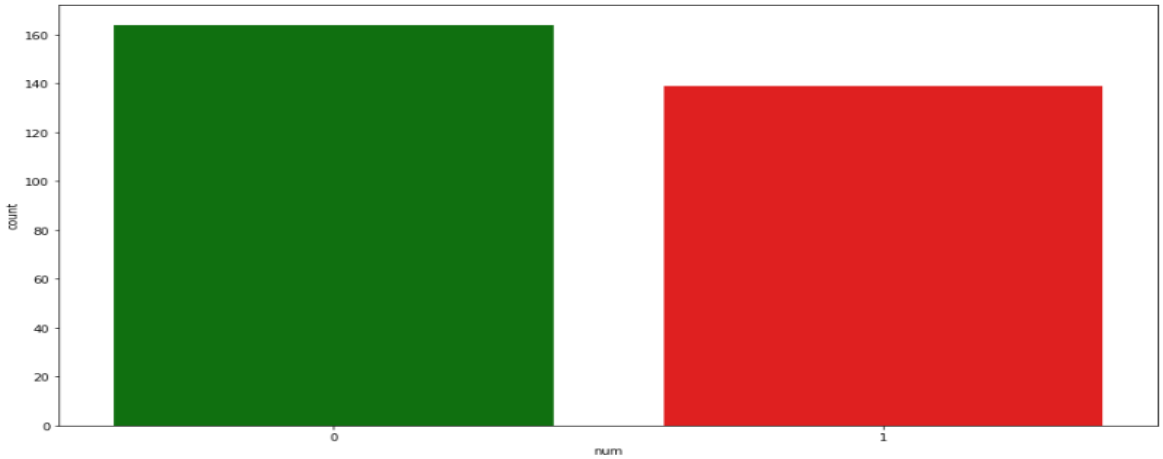
Free Form Visualization

	age	sex	cp	trestbps	cholestrol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num	predictgradient	rfcPredict
179	53	1	3	130	246	1	2	173	0	0.0	1	3	3	0	0	0
228	54	1	4	110	206	0	2	108	1	0.0	2	1	3	1	1	1
111	56	1	4	125	249	1	2	144	1	1.2	2	1	3	1	1	1
246	58	1	4	100	234	0	0	156	0	0.1	1	1	7	1	1	1
60	51	0	4	130	305	0	0	142	1	1.2	2	0	7	1	1	1

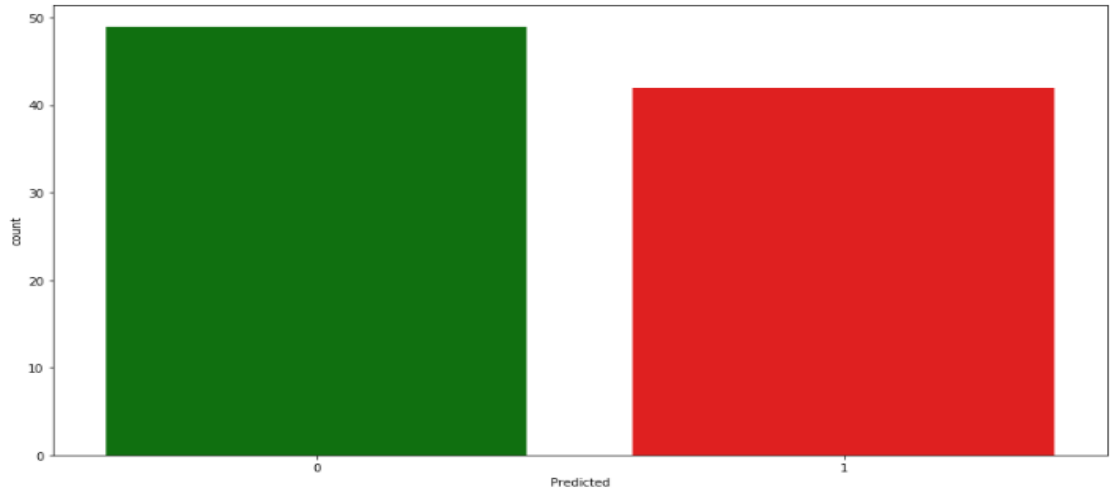
Since the dataset is randomly divided into 30% of test sets and 70% training data we can see that after the split the value predicted by Gradient Boost (‘predictgradient’ column) and Random forest(‘rfcPredict’ column) to be almost same with the actual value ‘num’, still the Random Forests is a better algorithm because its F-score is better than that of Gradient Boost Classifier.

The comparison of the percentage of heart disease present and absent predicted in the initial dataset and after the final algorithm Random forests is shown below.

```
Heart Disease Present Percentage       : 54.12541254125413
Heart Disease Absent Percentage        : 45.87458745874587
```




```
Heart Disease Present Percentage       : 53.84615384615385
Heart Disease Absent Percentage        : 46.15384615384615
```



The first visualization was on the whole data and the second visualization was on the test samples. The values were quite close.

Reflection

In this project, I had started my work by extensively studying various research papers of the existing work carried out in heart disease prediction. I then identified my aim. This was followed by research on which algorithm or classifier would be best suited for my project among which them I selected one as a benchmark model. The dataset that was to be used was identified and studied in depth. The dataset was subjected to preprocessing. The classification models I thought to be appropriate and have used are Random Forest, Gradient Boosting and Neural Networks(Benchmark model). I have discussed why I have chosen these three classifiers and how they can help in fulfilling the objective of the project. I have then evaluated each classification model based on their F-score. The entire framework of the project was also discussed.

From the F-score results, Random Forest algorithm was found to be the most accurate algorithm. Gradient Boosting Algorithm was found to be less than Random Forest but more than the benchmark model Neural Network algorithm. One can infer from these observations that ensemble techniques are very powerful and cannot be ignored in the prediction of important data. A recommender system has been developed and the objective of the project has been achieved.

Improvement

I have used a simple neural network and compared it with algorithms that involve boosting and bagging. My bagging and boosting techniques proved to be more efficient than neural network and boosting was found to be at slightly better than neural network this proves that bagging and boosting are very effective techniques and cannot be ignored in the prediction of important data.

Obviously there is little scope to make improvement to achieve more accuracy, like using various powerful tools like pruning and principal component analysis the F-score and the accuracy value can be improved. Even complex neural networks can also be used to make a recommender with more accuracy with the real world.

6 References

Web links:

1. https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/

2. https://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/

3. http://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/

4. http://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html#sphx-glr-auto-examples-model-selection-plot-confusion-matrix-py

5. https://machinelearningmastery.com/compare-machine-learning-algorithms-python-scikit-learn/

6. https://towardsdatascience.com/boosting-in-machine-learning-and-the-implementation-of-xgboost-in-python-fb5365e9f2a0

7. https://www.analyticsvidhya.com/blog/2017/03/imbalanced-classification-problem/

8. https://www.analyticsvidhya.com/blog/tag/gradient-boosting/

9. https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d

10. https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6

11. https://pandas.pydata.org/pandas-docs/stable/categorical.html

12. https://pandas.pydata.org/pandas-docs/stable/generated/pandas.Series.astype.html

13. https://stackoverflow.com/questions/39623429/what-is-the-difference-between-dtype-and-astype-in-numpy

14. https://machinelearningmastery.com/metrics-evaluate-machine-learning-algorithms-python/

15. https://www.analyticsvidhya.com/blog/2016/02/complete-guide-parameter-tuning-gradient-boosting-gbm-python/

16. http://pandas-ml.readthedocs.io/en/latest/conf_mat.html

17. https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/

18. https://www.youtube.com/watch?v=ErDgauqnTHk

19. http://dataaspirant.com/2017/06/26/random-forest-classifier-python-scikit-learn/

20. https://saintlad.com/install-tensorflow-on-windows/

Journal:

1. Pouriyeh, “A Comprehensive Investigation and Comparison of Machine Learning Techniques in the Domain of Heart Disease”, IEEE 2017.

2. Karayilan, “Prediction of Heart Disease Using Neural Network”, IEEE 2017.

3. Raid Lafta, ‘An Intelligent Recommender System based on Short-term Risk Prediction for Heart Disease Patients’, IEEE 2015.