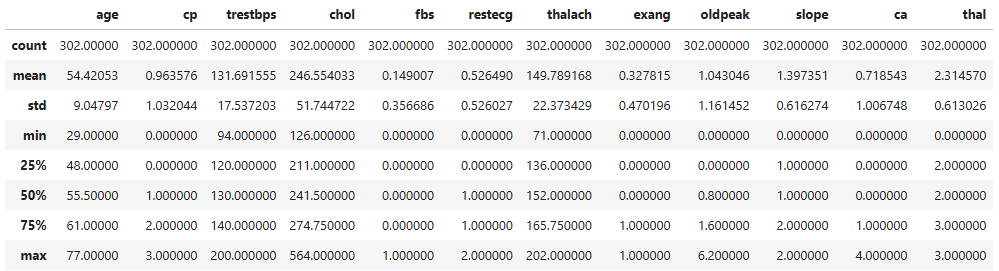
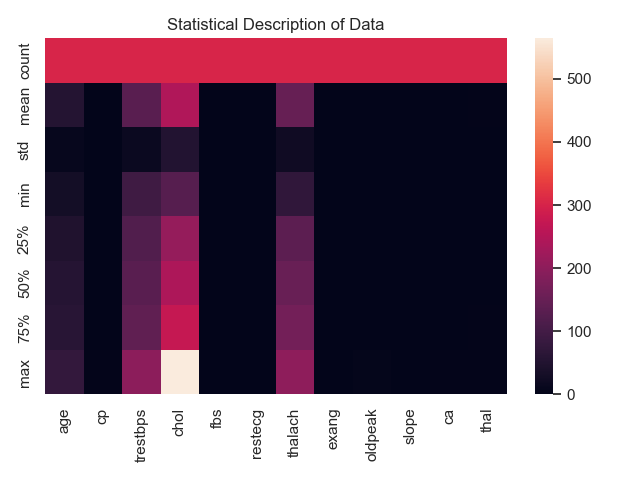
**Dataset Insights**

* Dataset is of Heart Disease Classification.
* It has 303 observations out of which 2 are identical (duplicated). First observation has been kept in the data while last one has been discarded.
* There are three columns that have null values which have been imputed with respective mean value.

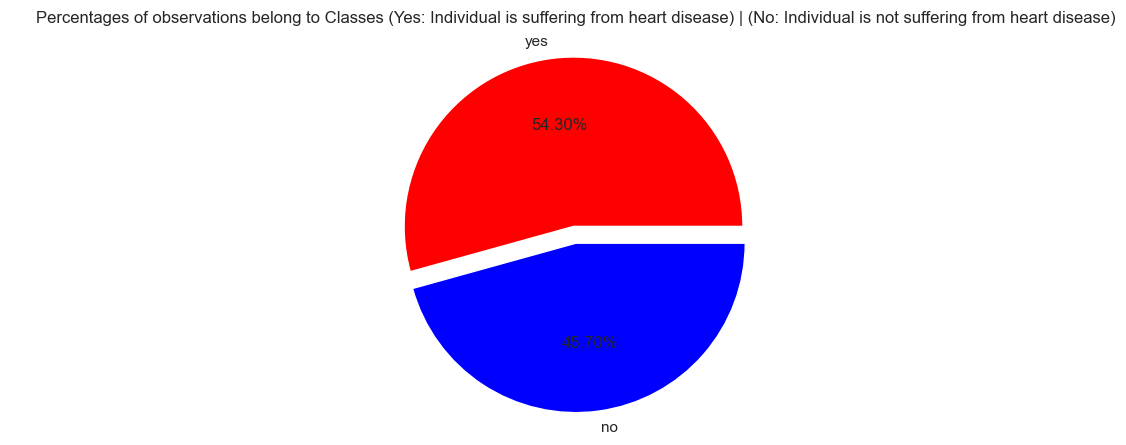
|  |  |
| --- | --- |
| **Column Name** | **Number of Null Values** |
| Trestbps | 4 |
| chol | 1 |
| Thalach | 5 |

* There are 206 entries of male and 96 entries of female in the dataset.
* Some basic statistics of the dataset have been given below:





* There are 164 patients suffering from heart disease while 138 patients who are not suffering from heart disease. There is a class imbalance problem.

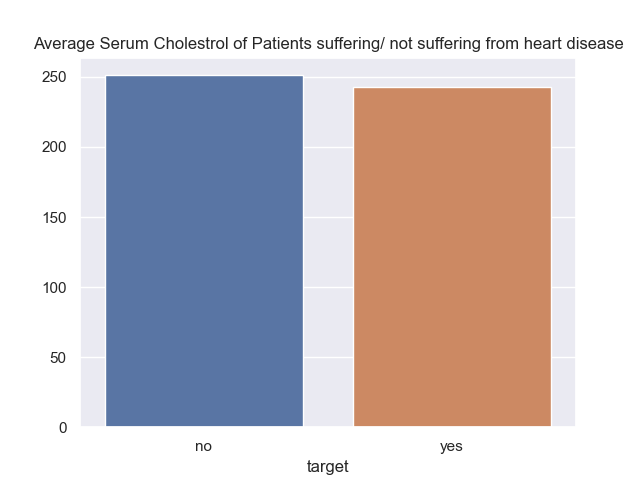


## Gender Distribution for the Patients suffering from heart disease

* Among the people 164 suffering from heart disease, 92 are male, 72 are female

## Average Serum Cholestrol of Patients suffering/ not suffering from heart disease

* Mean of Serum Cholestrol of Patients having heart disease: 242.640244
* Mean of Serum Cholestrol of Patients not having heart disease: 251.205202



## Age Range for patient suffering from heart disease[¶](http://localhost:8889/notebooks/Desktop/BootcampAI_WP_ICODE/Assignment2_Hassan Aftab.ipynb" \l "Age-Range-for-patient-suffering-from-heart-disease" \t "http://localhost:8889/notebooks/Desktop/BootcampAI_WP_ICODE/_self)

* Minimum Age of a patient suffering from heart disease: 29 years
* Maximum Age of a patient suffering from heart disease: 76 years
* Average Age of a patient suffering from heart disease: 52.58536585365854 years

**Feature Engineering**

* After doing the train test split, the column “sex” has been encoded using one hot encoding by removing the first column after being encoded to avoid multicolinearity.
* The targets y\_train and y\_test have been encoded too by Label Encoder.
* Standard scaler has been used to scale the input features.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1-score |
| KNN Classifier | 0.918033 | 0.922484 | 0.918033 | 0.918033 |
| Random Forest Classifier | 0.868852 | 0.870862 | 0.868852 | 0.868923 |
| Decision Tree Classifier | 0.770492 | 0.776666 | 0.770492 | 0.770245 |

* The results of models on the testing data have been given in Table below:

Here we can see that KNN classifier achieved the highest accuracy score.

**Overfitting Scenario:**

Now we look towards the overfitting scenario:

* KNN Classifier Training Accuracy: 0.8630705394190872
* KNN Classifier Testing Accuracy: 0.9180327868852459
* Decision Tree Classifier Training Accuracy: 1.0
* Decision Tree Classifier Testing Accuracy: 0.7704918032786885
* Random Forest Classifier Training Accuracy: 1.0
* Random Forest Classifier Testing Accuracy: 0.8688524590163934

It is apparent from above that results of Decision Tree Classifier and Random Forest Classifier are showing over-fitting.

**Fine Tuning using Random Search CV:**



In the case of KNN, With mentioned dictionary of parameters, you can say search space: Both the training as well as testing accuracy decreased but again overfitting doesn’t see to take place.

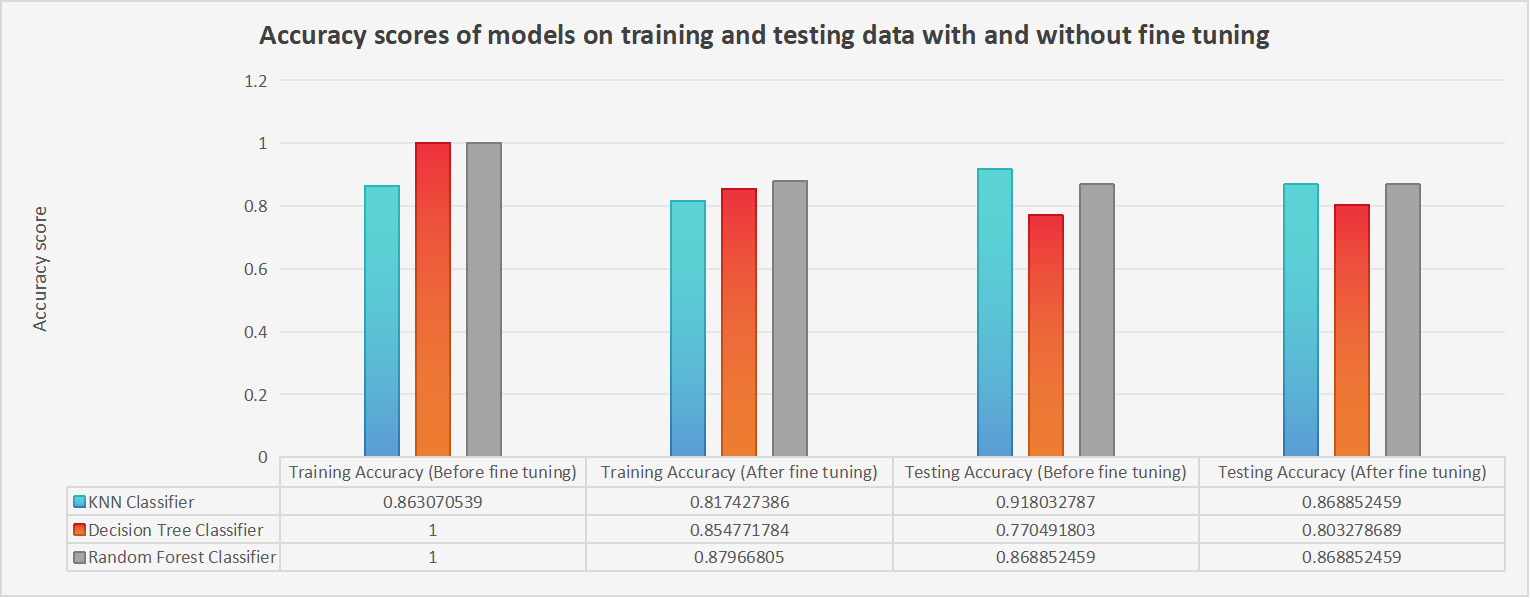
**Decision Tree Classifier**



In the case of Decision Tree Classifier, the training accuracy saw a dip and the testing accuracy saw a rise. A big drop in the overfitting took place.

**Random Forest Classifier**



In the case of random forest classifier, earlier accuracy scores were 1.0 and 0.888 for training and testing data respectively. But now you can see here that overfitting is quite reduced as there is a very small gap between the training and accuracy scores after fine tuning of model.  
  
  
  
  
  


Best Performance on test Data: KNN Classifier before fine tuning

KNN is the only algorithm that is giving the testing accuracy greater than the training accuracy in either for the cases: with or without fine tuning.