1. Data Exploration

- Sentiment Analysis Dataset Consist of total 18,389 records
- Dataset is divided into training and test set in the ratio of 80-20%
- There are two attributes in which Phrases represent review of movies and Sentiment represents the target corresponding to reviews.
- Targets are 0,1,2,3 and 4 where 0,1 are considered as negative reviews, 2 is considered as neutral and 3 and 4 are considered as positive reviews.

❖ **df['target'].value_counts()**

1 165(present of heart disease)

0 138(absence of heart disease)

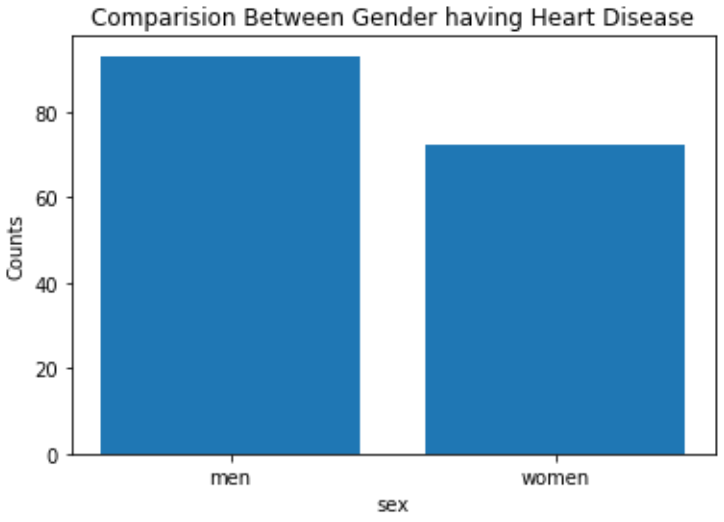

Fig: Counts of male and female having Heart Disease which clearly shows that risk of heart disease in male is higher than the female.

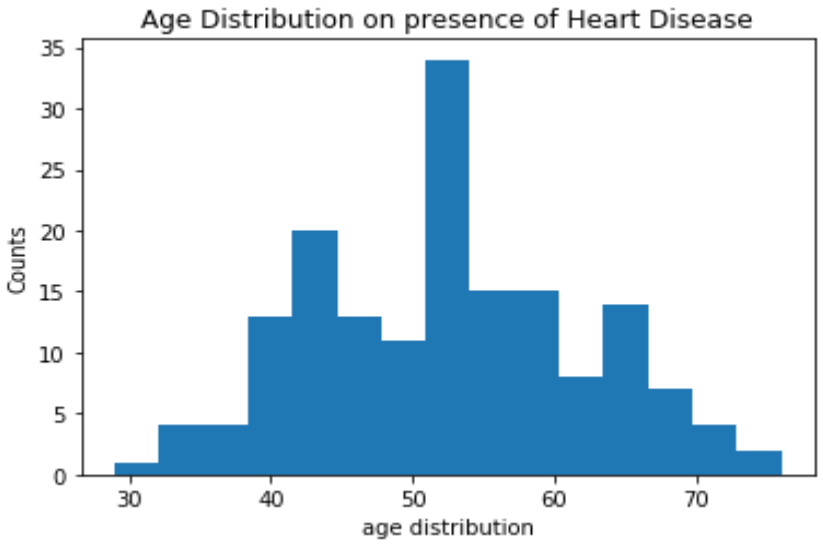

Fig: above figure is the age distribution according to the presence of Heart Disease. It clearly shows below the age 40 there is a minimal chance of Heart Disease whereas between the ages 50 to 55 there is a higher chance of having Heart Disease.

2. Feature Extraction and preprocessing

- **df.isnull().sum()**

  There were no null values in all the records of the variables.

- Training and test data are splitted to X_train, y_train, X_test and y_test where X_train and X_test hold the features and y-train and y_test hold the target variable.

- Since before training we have encoded the categorical datas using OneHotEncoder.

- OneHotEncoder encodes categorical features as a one-not numerical array.

- Later we do the same procedure to preprocess the categorical features of test data.

- After encoding, the total number of both test and training dataset columns contain 23 features.

3. Grid Search

- Here, we used the Grid search for tuning the hyperparameter.
- Grid Search uses the Cross Validation technique to find the best hyperparameter.

Different hyperparameters defined for each classifier is shown below:

S.No.	Classifier	Hyperparameters
1	SVC	**Kernel**: linear, rbf **C**: 1,10,5
2	DecisionTreeClassifier	**max_depth**: 2,3,5,8,6,4 **min_samples_leaf** : 12, 8, 2,14
3	RandomForestClassifier	**N_estimators**: 10,50,100 **max_features**: 1,0.5,0.8, 0.1 **Min_samples_leaf**: 12,1,5,8
4	ExtraTreesClassifier	**N_estimators**: 10,50,100 **max_features**: 1,0.5,0.8, 0.1, **Min_samples_leaf**: 12,1,5,8
5	AdaBoostClassifier	**N_estimators:** 5,10,20,30 **Learning_rate**: 0.5,0.1,0.001,0.0001
6	GradientBoostingClassifier	**N_estimators**: 10,20,40,45 **learning_rate**: 0.05,0.1, 0.01, 0.2, 0.5 **Min_samples_split**: 2,3,5

S.No.	Classifier	Best Hyperparameters
1	SVC	**Kernel**: linear **C**: 1
2	DecisionTreeClassifier	**max_depth**: 3 **min_samples_leaf** : 14
3	RandomForestClassifier	**N_estimators**: 50 **max_features**: 1 **Min_samples_leaf**: 5
4	ExtraTreesClassifier	**N_estimators**: 10 **max_features**: 0.1, **Min_samples_leaf**:5
5	AdaBoostClassifier	**N_estimators:** 30 **Learning_rate**: 0.1
6	GradientBoostingClassifier	**N_estimators**: 10 **learning_rate**: 0.1 **Min_samples_split**: 2

4. Model Evaluation and Comparison

Below table shows the accuracy of each classifier for the dataset of Heart Disease.

Classifier	F1-macro
SVC	
DecisionTreeClassifier	
RandomForestClassifier	
ExtraTreesClassifier	
AdaBoostClassifier	
GradientBoostingClassifier	

The accuracy rate of GradientBoostingClassifier is higher compared to other five classifiers. Hence, GradientBoostingClassifier with best hyperparameter **N_estimators**: 10, **learning_rate**: 0.1 and **Min_samples_split**: 2 is chosen as the best classification algorithm for the Heart Disease Dataset.