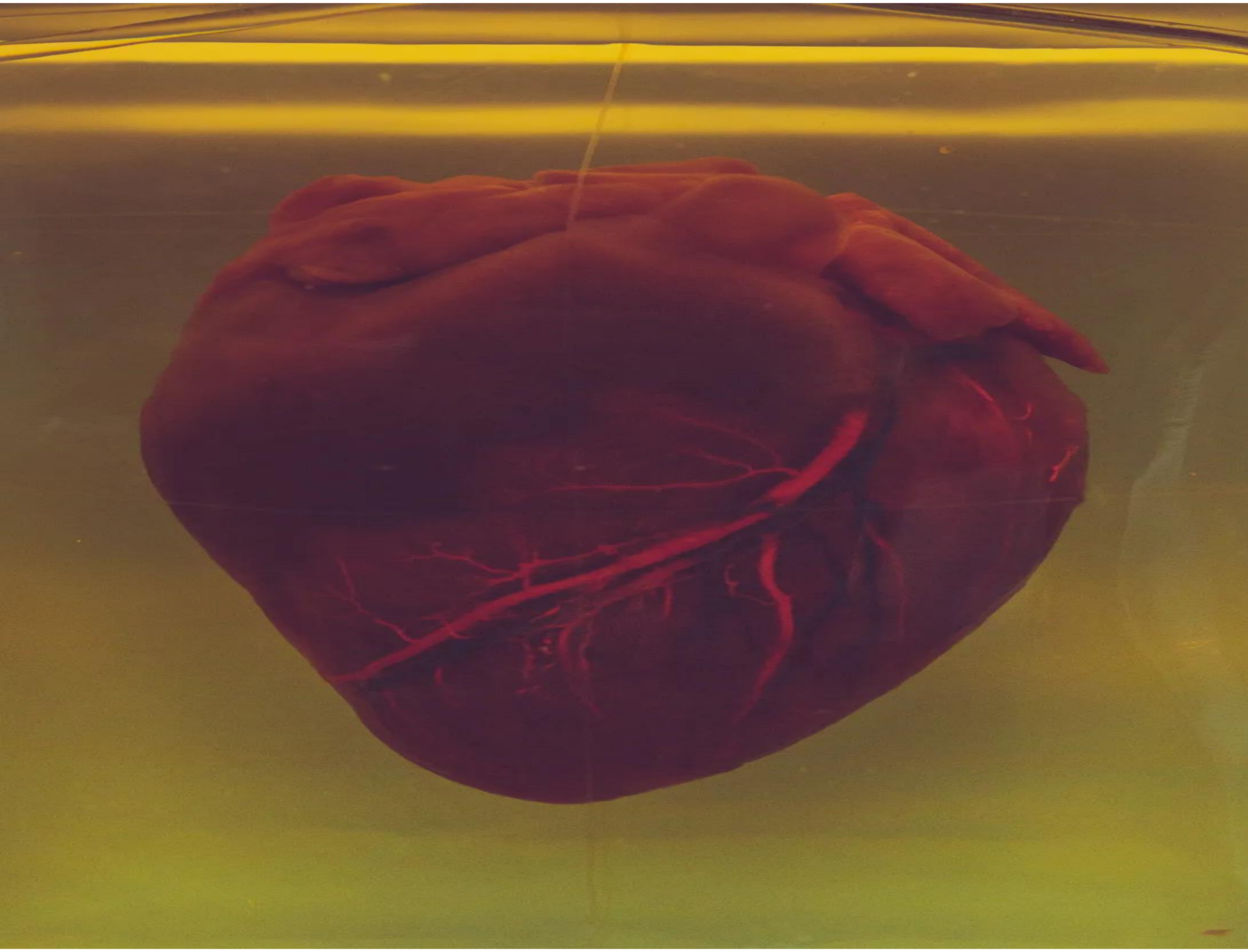
PREDICTING HEART DISEASE WITH CLASSIFICATION MACHINE LEARNING ALGORITHM

Chetan ashok Ahirrao
Mail id: -ahirraochetan1994@gmail.com
Mob: - 8408004897
GITHUB: - https://github.com/chetan456789
LINKEDIN: - https://www.linkedin.com/in/chetan-ahirrao-602305b5/



ABSTRACT

Heart disease is one of the leading causes of death worldwide. Early detection and diagnosis of heart disease is crucial in preventing serious complications and improving patient outcomes. Machine learning algorithms have shown great potential in predicting heart diseases by analyzing patient data and identifying patterns and risk factors.

Classification machine learning algorithms, such as logistic regression, decision trees, random forest, and support vector machines, have been used to predict heart diseases based on various patient attributes such as age, gender, blood pressure, cholesterol levels, and family history of heart disease. These algorithms work by learning from historical data to predict the likelihood- od of a patient developing heart disease in the future.

In this study, we have analyzed a dataset of patient information collected from various sources to predict the occurrence of heart disease. The dataset contains various patient attributes such as age, sex, blood pressure, cholesterol levels etc. We have applied different classification algorithms on this dataset and compared their performance based on various metrics such as accuracy, sensitivity, specificity.

Our results show that the random forest algorithm outperforms other algorithms in predicting heart disease, achieving an accuracy of 81%.We have also identified the most important features that contribute to the prediction of heart disease, which can help healthcare providers to focus on these risk factors and take preventive measures.

In conclusion, machine learning algorithms can be powerful tools for predicting heart disease and improving patient outcomes. Our study demonstrates the potential of classification algorithms to accurately predict heart disease using patient data and highlights the importance of early detection and prevention of heart disease.

1.INTRODUCTION

Machine learning is a rapidly growing field that has gained a lot of attention in recent years due to its ability to extract insights and predictions from large datasets. A machine learning project involves building and training models that can learn patterns in data and make predictions or decisions based on those patterns.

The goal of a machine learning project may vary depending on the specific problem being addressed. It could involve predicting customer behavior, diagnosing a disease.

In this project introduction, we will diagnose the patient if he or she is having diseases or not based on the various parameters such as below.

1.age
2.**sex**: 1= Male, 0= Female (*Binary*)
3. (**cp**)chest pain type (4 values -*Ordinal*): Value 1: typical angina, Value 2: atypical angina, Value 3: non-anginal pain, Value 4: asymptomatic
4. (**trestbps**) resting blood pressure
5. (**Chol**) serum cholesterol in mg/dl
6. (**fbs**)fasting blood sugar > 120 mg/dl(*Binary*) (1 = true; 0 = false)
7. (**restecg**) resting electrocardiography results (values 0,1,2)
8. (**thalach**) maximum heart rate achieved. Etc. First, we need to identify the problem and gather relevant data. This may involve obtaining data in our project we got a data from amazon s3. Data preprocessing is the next step, which involves cleaning, transforming, and normalizing the data to make it suitable for training the machine learning models.

.

Once the data is preprocessed, we move on to feature selection and engineering. This involves identifying the most important features that are relevant to the problem and engineering new features from the existing ones.

The next step is to select the appropriate machine learning algorithm that is best suited for the problem at hand
The machine learning model is then trained using the preprocessed data, and the performance of the model is evaluated using various performance metrics. Once the model is trained and validated, it can be deployed in a production environment to make predictions on new data.

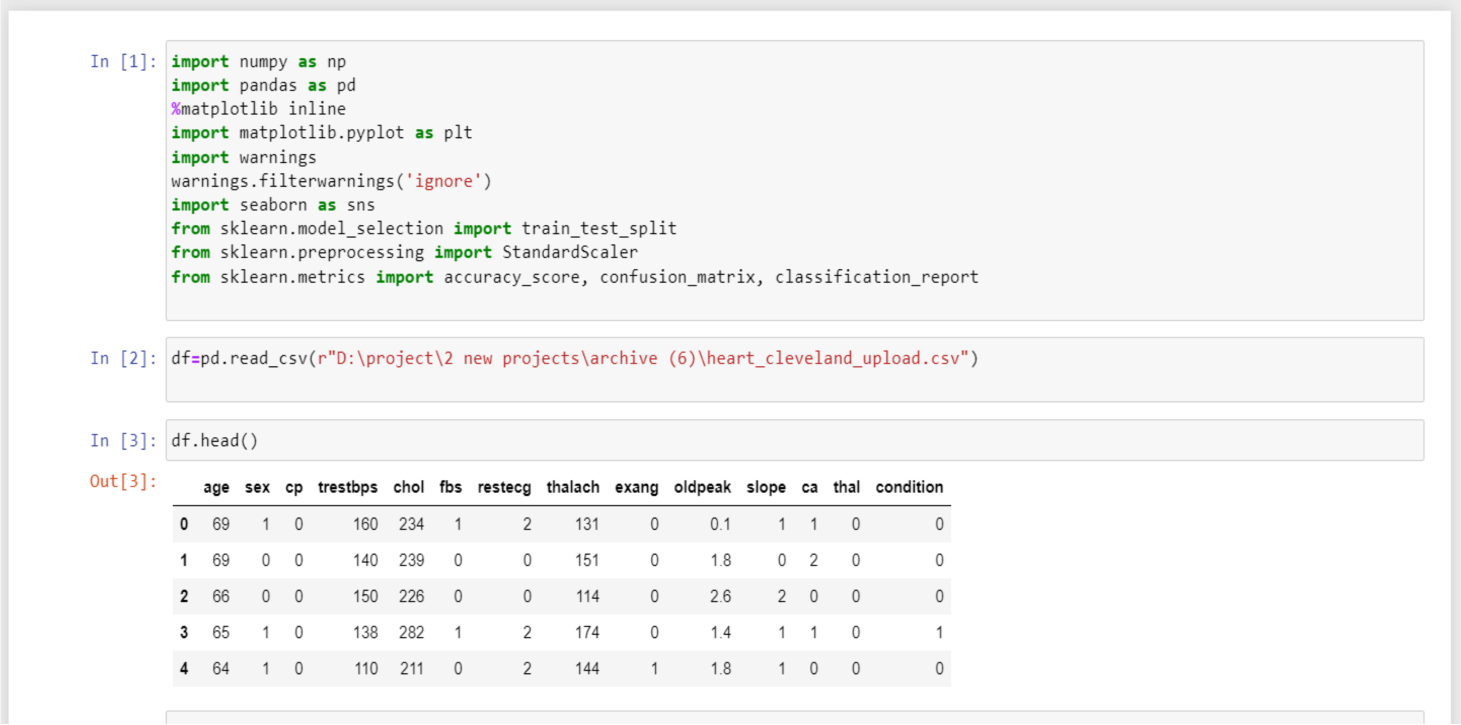
Finally, it is essential to continuously monitor the performance of the model and update it as needed. This may involve retraining the model with new data, tuning hyperparameters, or even switching to a different algorithm altogether.

In summary, machine learning algorithms can be powerful tools for predicting heart disease and improving patient outcomes. involves identifying the problem, gathering and preprocessing the data, selecting the appropriate algorithm, training and evaluating the model, and deploying and maintaining the model in healthcare domain.

2.DATA WRANGLING:

This step is very essential in the data analysis. correcting errors, removing duplicates, and formatting data in a way that it is suitable for analysis below are some steps.

First, we imported python libraries NumPy pandas matplotlib for visualization and sklearn our dataset is label data so for categorical data we use supervised machine learning algorithms next step is to read the file. Using head () and tail () function we can see top five records and last 5 records and we can see our data.



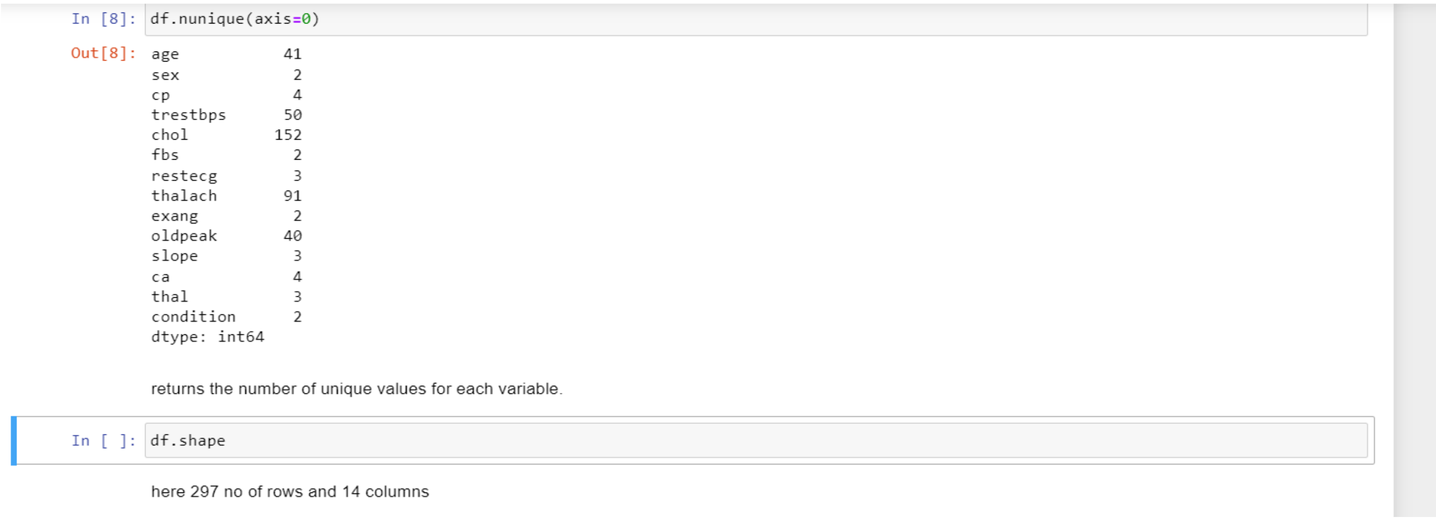
In [1]:
```
import numpy as np
import pandas as pd
%matplotlib inline
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

In [2]:
```
df=pd.read_csv(r"D:\project\2 new projects\archive (6)\heart_cleveland_upload.csv")
```

In [3]:
```
df.head()
```

Out[3]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	condition
0	69	1	0	160	234	1	2	131	0	0.1	1	1	0	0
1	69	0	0	140	239	0	0	151	0	1.8	0	2	0	0
2	66	0	0	150	226	0	0	114	0	2.6	2	0	0	0
3	65	1	0	138	282	1	2	174	0	1.4	1	1	0	1
4	64	1	0	110	211	0	2	144	1	1.8	1	0	0	0



```
In [8]: df.nunique(axis=0)

Out[8]: age          41
        sex           2
        cp            4
        trestbps     50
        chol        152
        fbs           2
        restecg       3
        thalach      91
        exang         2
        oldpeak      40
        slope         3
        ca            4
        thal          3
        condition     2
        dtype: int64
```

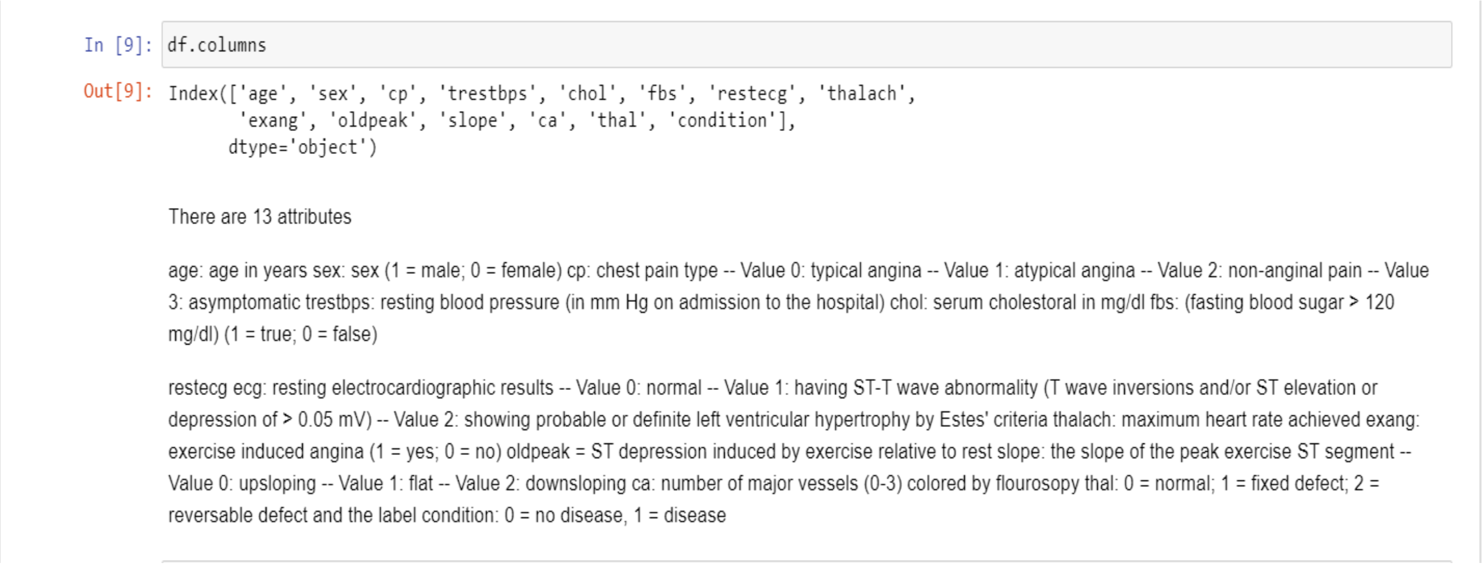
returns the number of unique values for each variable.

```
In [ ]: df.shape
```

here 297 no of rows and 14 columns

df.nunique(axis=0) represents unique values for each variable.

And in this dataset, we have 297 no of rows and 14 columns.

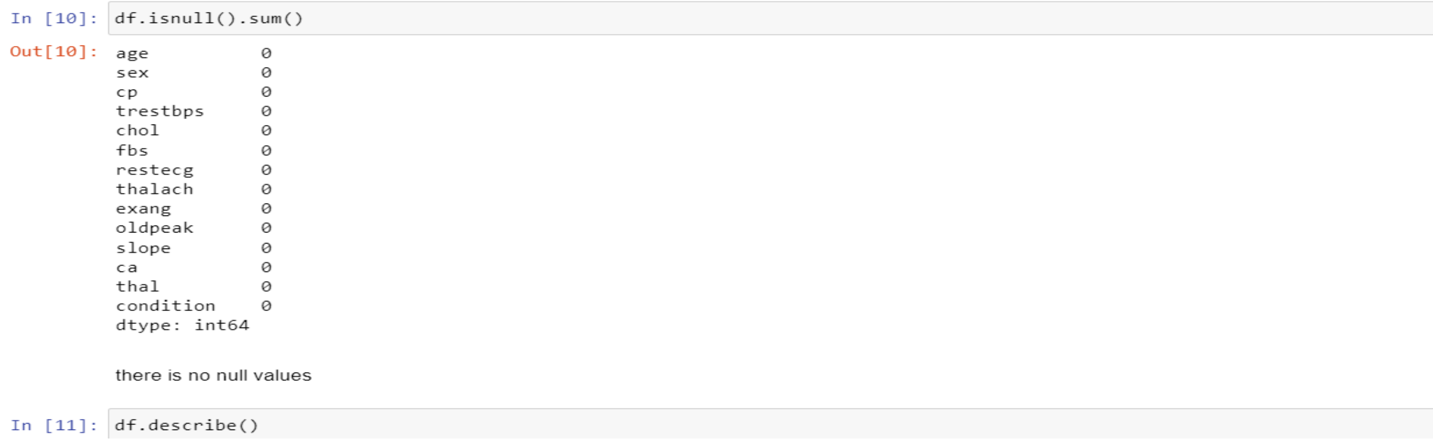


```
In [9]: df.columns

Out[9]: Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
               'exang', 'oldpeak', 'slope', 'ca', 'thal', 'condition'],
              dtype='object')
```

There are 13 attributes

age: age in years sex: sex (1 = male; 0 = female) cp: chest pain type -- Value 0: typical angina -- Value 1: atypical angina -- Value 2: non-anginal pain -- Value 3: asymptomatic trestbps: resting blood pressure (in mm Hg on admission to the hospital) chol: serum cholestoral in mg/dl fbs: (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)

restecg ecg: resting electrocardiographic results -- Value 0: normal -- Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV) -- Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria 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 -- Value 0: upsloping -- Value 1: flat -- Value 2: downsloping ca: number of major vessels (0-3) colored by flourosopy thal: 0 = normal; 1 = fixed defect; 2 = reversable defect and the label condition: 0 = no disease, 1 = disease



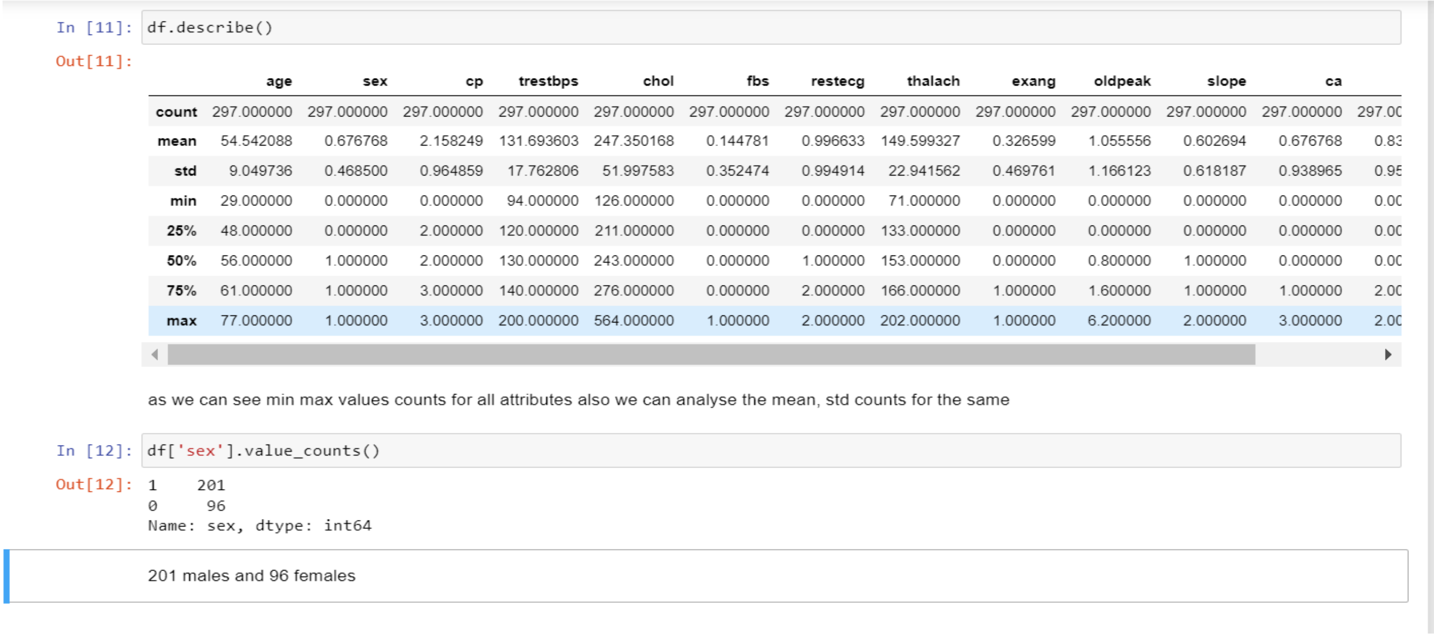
```
In [10]: df.isnull().sum()

Out[10]: age          0
         sex          0
         cp           0
         trestbps     0
         chol         0
         fbs          0
         restecg      0
         thalach      0
         exang        0
         oldpeak      0
         slope        0
         ca           0
         thal         0
         condition    0
         dtype: int64
```

there is no null values

```
In [11]: df.describe()
```

as we can see there is no null value present in our dataset.

Next, we describe the data here we have values of variable like min, max, std, mean etc. we can see. Also, we can understand how well our data is classified in 25%, 50% and 75%. Also in dataset 201 are males and 96 are females.



```
In [11]: df.describe()
Out[11]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	
count	297.000000	297.000000	297.000000	297.000000	297.000000	297.000000	297.000000	297.000000	297.000000	297.000000	297.000000	297.000000	297.00
mean	54.542088	0.676768	2.158249	131.693603	247.350168	0.144781	0.996633	149.599327	0.326599	1.055556	0.602694	0.676768	0.83
std	9.049736	0.468500	0.964859	17.762806	51.997583	0.352474	0.994914	22.941562	0.469761	1.166123	0.618187	0.938965	0.95
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.00
25%	48.000000	0.000000	2.000000	120.000000	211.000000	0.000000	0.000000	133.000000	0.000000	0.000000	0.000000	0.000000	0.00
50%	56.000000	1.000000	2.000000	130.000000	243.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	0.00
75%	61.000000	1.000000	3.000000	140.000000	276.000000	0.000000	2.000000	166.000000	1.000000	1.600000	1.000000	1.000000	2.00
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	3.000000	2.00

as we can see min max values counts for all attributes also we can analyse the mean, std counts for the same

```
In [12]: df['sex'].value_counts()
Out[12]: 1    201
         0     96
         Name: sex, dtype: int64
```

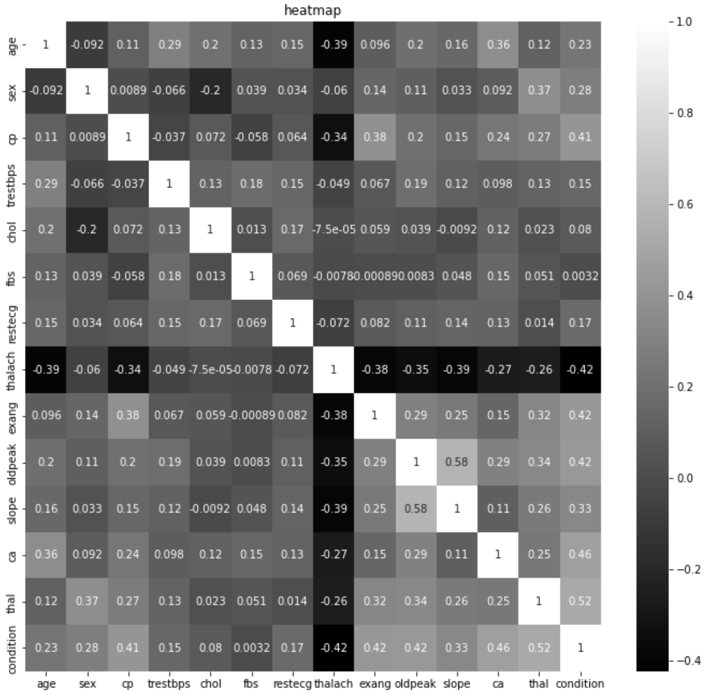
201 males and 96 females

3.EXPLORATORY DATA ANALYSIS

3.1 CORRELATIONS

Correlation Matrix- lets you see **correlations** between all variables.

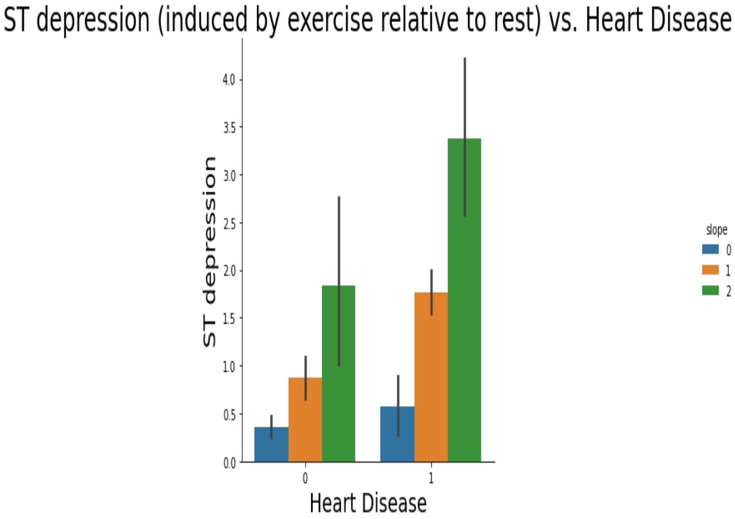
Within seconds, you can see whether something is positively or negatively correlated with our **predictor (target)**.



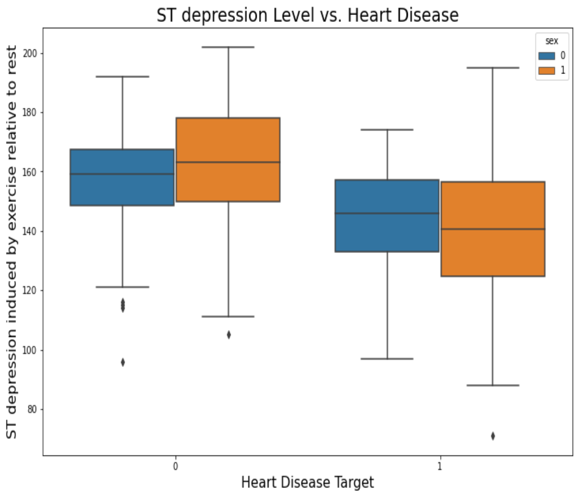
We can see there is a **positive correlation** between chest pain (cp) & target (our predictor). This makes sense since, the greater amount of chest pain results in a greater chance of having heart disease. Cp (chest pain), is an ordinal feature with 4 values: Value 1: typical angina, Value 2: atypical angina, Value 3: non-anginal pain, Value 4: asymptomatic. In addition, we see a negative correlation between exercise induced angina (exang) & our predictor. This makes sense because when you exercise, your heart requires more blood, but narrowed arteries slow down blood flow.

Pairplots are also a great way to immediately see the correlations between all variables. But you will see me make it with only continuous columns from our data, because with so many features, it can be difficult to see each one. So instead, I will make a pairplot with only our continuous features.

Catplot: ST segment depression occurs because when the ventricle is at rest and therefore repolarized. If the trace in the ST segment is abnormally low below the baseline, this can lead to this Heart Disease. This is supports the plot above because low ST Depression yields people at greater risk for heart disease. While a high ST depression is considered normal & healthy. The “slope” hue, refers to the peak exercise ST segment, with values: 0: upsloping, 1: flat, 2: downsloping). Both positive & negative heart disease patients exhibit equal distributions of the 3 slope categories.



Boxplot: Positive patients exhibit a heightened median for ST depression level, while negative patients have lower levels. In addition, we don’t see many differences between male & female target outcomes, expect for the fact that males have slightly larger ranges of ST Depression.


ST depression Level vs. Heart Disease
ST depression induced by exercise relative to rest
200
180
160
140
120
100
80
0
1
Heart Disease Target
sex
0
1

4.MODEL BUILDING

Next, we have to prepare data for model building for that I have used StandardScaler to scale the data.
the main reason to use standardscaler are normalization, standardscaler scaler the data such that it has a mean of 0 and a standard deviation of 1. This is useful because it makes the data more consistent and easier to compare across different feature. Also, standardscaler is less sensitive to outliers than other scaling significant impact on other scaling techniques, but standardscaler uses the mean and standard deviation, which are less affected by extreme values.
Then we split the dataset in train set and testing set training set is used to fit the model, i.e., to learn relationship between feature and target variable. The test set is used to evaluate the performance of the model by measuring how well it can predict the target variable for new unseen data.

4.1 JUSTIFICATION OF FOUR ALGORITHMS:

Model1:

4.1.1 SVC (support vector classifier):

SVC (Support Vector Classification) is a type of supervised learning algorithm used in machine learning for binary and multi-class classification. It is based on the concept of finding the optimal hyperplane that separates the different classes in a high-dimensional space.

In SVC, the training data is used to find the optimal hyperplane that maximizes the margin, which is the distance between the hyperplane and the closest data points of each class. The hyperplane is chosen such that it maximizes the margin and correctly classifies the training data.

In cases where the data is not linearly separable, SVC uses a kernel function to map the input data into a higher-dimensional space where it is more likely to be linearly separable. The most common kernel functions used in SVC are the linear, polynomial, and radial basis function (RBF) kernels.

The hyperparameters of the SVC algorithm, such as the regularization parameter (C) and the kernel parameters, can be tuned using cross-validation or other model selection techniques to optimize the performance of the algorithm on the given dataset.

SVC has been shown to be effective in a wide range of applications, including image classification, text classification, and bioinformatics.

model2:

4.1.2 NAIVE BAYS CLASSIFIER:

Naive Bayes is a type of probabilistic classification algorithm used in machine learning. It is based on Bayes' theorem, which describes the probability of an event occurring based on prior knowledge of conditions that might be related to the event. Naive Bayes is called "naive" because it assumes that the features used for classification are independent of each other, which is often not the case in practice.

In Naive Bayes, the algorithm builds a probabilistic model of the training data, assuming that the distribution of each feature is independent of the other features. Then, for a new observation, the algorithm calculates the probability of each class given the observed values of the features, using Bayes' theorem. The class with the highest probability is assigned as the predicted class for the new observation.

Model3:

4.1.3 Logistic Regression

Logistic regression is a statistical method used for binary classification tasks, where the goal is to predict a binary outcome (e.g., true/false, yes/no, 0/1) based on a set of input features. It is a type of generalized linear model that models the probability of the binary outcome as a function of the input features.

In logistic regression, the output is a logistic function, which maps any real-valued input to a value between 0 and 1. The logistic function is a sigmoid function that can be expressed as:

$$P(y=1|x) = 1 / (1 + \exp(-z))$$

where $P(y=1|x)$ is the probability of the positive class ($y=1$) given the input features (x), and z is a linear combination of the input features and their associated weights:

$$z = w0 + w1x1 + w2x2 + ... + wn*xn$$

where w0, w1, w2, ..., wn are the weights assigned to each feature.

The logistic regression model is trained using a maximum likelihood estimation method, which seeks to find the weights that maximize the likelihood of the observed data. The weights can be learned using optimization algorithms such as gradient descent or Newton's method.

Logistic regression has several advantages, including its simplicity, interpretability, and ability to handle non-linear relationships between the input features and the output variable. It is commonly used in applications such as marketing, finance, and healthcare, where binary classification is a common task.

However, logistic regression also has some limitations, such as its assumption of linear relationships between the input features and the output variable, and its inability to handle complex interactions between features. In such cases, more complex models such as decision trees, random forests, or neural networks may be more appropriate

model4:

4.1.4 Decision trees:

Decision trees are a type of algorithm that make predictions by partitioning the data into smaller subsets based on a set of rules or conditions.

A decision tree is a type of supervised learning algorithm used in machine learning for both classification and regression tasks. It is a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. Each internal node of the tree represents a decision on a feature, and each leaf node represents a class label or a numerical value. The tree is built by recursively partitioning the feature space into smaller and smaller regions, based on the values of the features, until the regions are pure or meet some other stopping criterion.

In decision tree classification, the goal is to classify a new observation by traversing the decision tree based on the observed values of the features, starting from the root node and proceeding down the tree until a leaf node is reached.

In decision tree regression, the goal is to predict a numerical value by traversing the decision tree and computing the value at the leaf node reached by the observation.

The decision tree algorithm works by selecting the feature that provides the most information gain, or reduction in impurity, at each node. Information gain is typically measured by entropy or Gini impurity, which are measures of the degree of randomness or uncertainty in the class distribution at a given node.

5. EVALUATION METRICS

5.1 Precision

Precision shows the ratio of true positives to the sum of true positives and false positives.

$$\text{Precision} = \frac{\text{True\_Positives}}{\text{True Positives + False Positives}}$$

From the equation above, it is clear that for a good model, False Positives should be as small as possible. Precision lies between 1(good) and 0(bad).

5.2 Recall

This is the ratio of true positives to the sum of true positives and false negatives.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives + False Negatives}}$$

Here, for a good model, False Negatives should be as small as possible. Recall also lies between 1(good) and 0(bad).

5.3 Accuracy

Accuracy summarises the whole model. It is the ratio of the correctly classified prediction to the entire prediction. Mathematically:

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{All Predictions}}$$

6. CONCLUSION:

1. Out of the 13 features we examined, the top 4 significant features that helped us classify between a positive & negative Diagnosis were chest pain type (cp), maximum heart rate achieved (thalach), number of major vessels (ca), and ST depression induced by exercise relative to rest (oldpeak).

2. Our machine learning algorithm can now classify patients with heart disease. Now we can properly diagnose patients, & get them the help they need to recover. By diagnosing detecting these features early, we may prevent worse symptoms from arising later.

3. Our Support Vector Classifier yields the highest accuracy, 81%. Any accuracy above 70% is considered good, but be careful because if your accuracy is **extremely** high, it may be too good to be true (an example of Over fitting). Thus, 80% is the ideal accuracy!

7. RESULTS

classifier	Class	labels	PREC	REC	F1-SCORE	SUPPORT	ACC

Support vector classifier	Heart Disease with classification ml algorithm	0(Negative heart disease)
		1(positive heart disease)
			0.86
			0.77
				0.77
				0.86
					0.81
					0.81
						48
						42
							81.11%

Naïve bayes	Heart Disease with classification ml algorithm	0(Negative heart disease)
		1(positive heart disease)
			0.81
			0.76
				0.77
				0.83
					0.80
					0.80
						48
						42
							80.00%

Logistic Regression	Heart Disease with classification ml algorithm	0(Negative heart disease)
		1(positive heart disease)
			0.82
			0.76
				0.77
				0.81
					0.80
					0.78
						48
						42
							78.00%

Decision tree algorithm	Heart Disease with classification ml algorithm	0(Negative heart disease)
		1(positive heart disease)
			0.77
			0.74
				0.77
				0.74
					0.77
					0.74
						48
						42
							75.55%

8. REFERENCES:

1."Pattern Recognition and Machine Learning" by Christopher M. Bishop
"Machine Learning: A Probabilistic Perspective" by Kevin P. Murphy

2."Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville

3."Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Géron

4."Python Machine Learning" by Sebastian Raschka and Vahid Mirjalili
"The Hundred-Page Machine Learning Book" by Andriy Burkov

5."Introduction to Machine Learning with Python" by Andreas C. Müller and Sarah Guido

6"Reinforcement Learning: An Introduction" by Richard S. Sutton and Andrew G. Barto

7."Machine Learning Yearning" by Andrew Ng
"Bayesian Reasoning and Machine Learning" by David Barber

8. https://www.kaggle.com/datasets/cherngs/heart-disease-cleveland-uci