**Springboard Data Science Career Track 2020 Consolidated Report**

**Heart Disease Prediction using various Machine Learning Algorithms Mukkul Kurwa**

**April 2020**

**Project Overview:** Heart disease is the leading cause of death for people of most racial and ethnic groups in the United States, including African American, American Indian, Alaska Native, Hispanic, and white men. For women from the Pacific Islands and Asian American, American Indian, Alaska Native, and Hispanic women, heart disease is second only to cancer.

* One person dies every 37 seconds just in the United States alone from cardiovascular disease.
* About 647,000 Americans die from heart disease each year—that’s 1 in every 4 deaths.
* Heart disease costs the United States about $219 billion each year from 2014 to 2015. This includes the cost of health care services, medicines, and lost productivity due to death.

**Origins of the Data Set:** The data used for training and testing is the [Heart Disease UCI](https://www.kaggle.com/ronitf/heart-disease-uci) downloaded from Kaggle. This database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. In particular, the Cleveland database is the only one that has been used by ML researchers to this date. The "goal" field refers to the presence of heart disease in the patient.

**Problem Statement:**

* Complete analysis of Heart Disease UCI data-set.
* To predict whether a person has a heart disease or not based on the various biological and physical parameters.

**Metrics:** Once the model is trained, I need to test its performance on the testing data-set. The model has never seen this information before; as a result, the testing data-set allows me to determine whether or not the model will be able to generalize to information that wasn't used during its training phase. I have used some of the metrics provided by Scikit-learn for this purpose such as classification reports and accuracy score.

# Data Exploration: Our data-set has the following 14 features which we will be using to train our model:

# **Data Wrangling:**

**1. What kind of cleaning steps did you perform?**

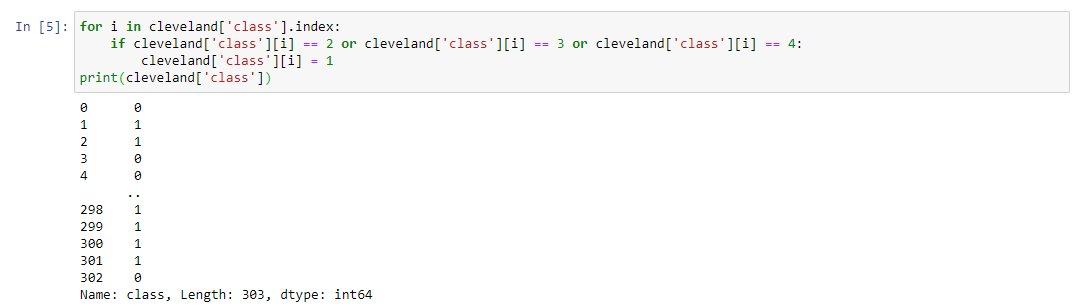
The data looks good and hence no cleaning required as there are no missing values nor there are any text values present in the dataset. Hence no dummy variables required. All the features are of the type float64 except for 'ca' and 'thal' which are object types. I'll concentrate on simply attempting to distinguish presence (values 1,2,3,4) from absence (value 0). So, I'll be replacing values 2,3,4 with the value 1.

**2. How did you deal with missing values, if any?**

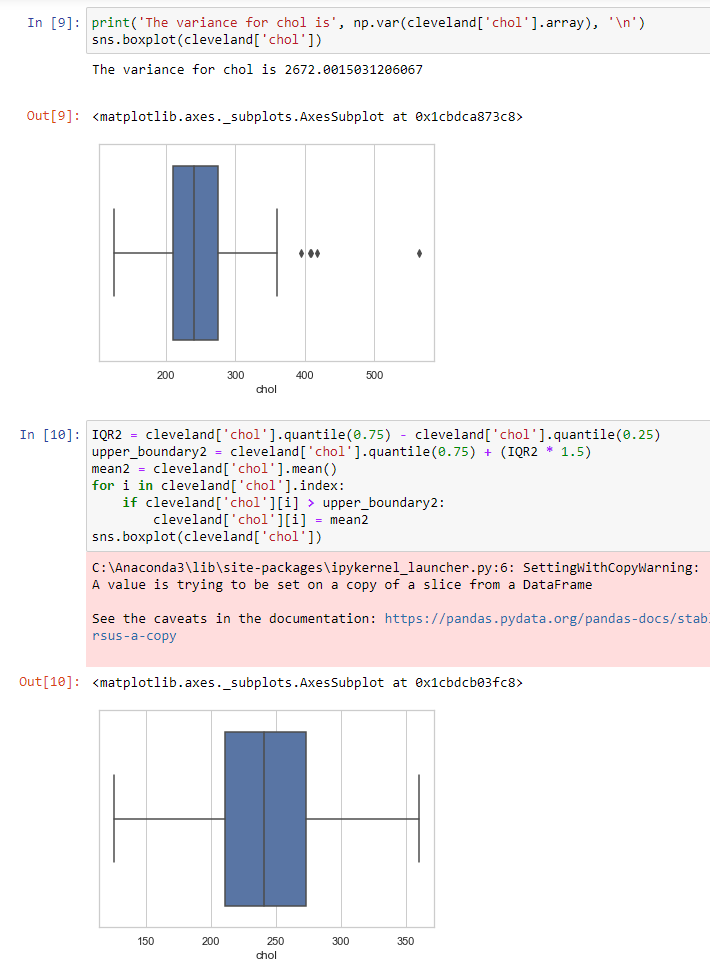
There are no missing values in the dataset.

**3. Were there outliers, and how did you handle them?**

To detect the outliers we plot the boxplot for the quantitative features 'age', 'trestbps', 'chol', 'thalach' and 'oldpeak' as below. No chance of outliers for categorical features.









We can see that features 'trestbps', 'chol', 'thalach' and 'oldpeak' have outliers. We have dealt with the outliers by replacing them with mean.

**Apply Inferential Statistics:**

**Are there strong correlations between pairs of independent variables or between an independent and a dependent variable?**

**Numerical Variables**

We use correlation to see how the numerical features are related to each other.

corrMatrix **=** cleveland

​

corrMatrix **=** corrMatrix.corr()

​

fig, ax **=** plt.subplots(figsize**=**(15,10))

​

ax **=** sns.heatmap(corrMatrix,

annot**=True**,

linewidths**=**0.5,

fmt**=**".2f",

cmap**=**"YlGnBu");



**INSIGHT:** There are no strong correlations between numerical independent features.

**What are the most appropriate tests to use to analyse these relationships?**

**Numerical Variables v/s Categorical Variables**

We use ANOVA to see how the numerical features are related to categorical features.

The one-way analysis of variance (ANOVA) is used to determine whether there are any statistically significant differences between the means of two or more independent (unrelated) groups (although you tend to only see it used when there are a minimum of three, rather than two groups).

cleveland\_anova **=** cleveland

​

cleveland\_anova **=** cleveland\_anova.drop(['sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca', 'thal'], axis**=**1)

cleveland\_anova

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |

**import** scipy.stats **as** stats

​

*# Significance level*

alpha **=** 0.05

*# Perform One Way ANOVA*

outcome **=** stats.f\_oneway(cleveland\_anova['age'], cleveland\_anova['trestbps'], cleveland\_anova['chol'],

cleveland\_anova['thalach'], cleveland\_anova['oldpeak'])

*# gettin p-value of test*

p\_value**=** outcome[1]

​

*#Null hypothesis mu\_1 = mu\_2 = mu\_3 = mu\_4 = mu\_5*

**if** p\_value **<=** alpha:

*# we reject null hypothesis*

print('Null hypothesis is unlikely to except.')

**else**:

*# we reject alternative hypothesis*

print('Null hypothesis cannot be rejected.')

Null hypothesis is unlikely to except.

**INSIGHT:** We reject null hypothesis.

**Categorical Variables**

We use Chi Square to see how the numerical features are related to categorical features.

**Sex**

fig, ax **=** plt.subplots(figsize**=**(10,6))

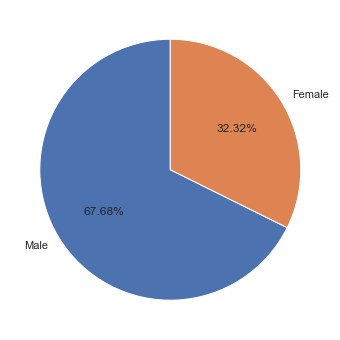
​

male **=** len(cleveland[cleveland['sex'] **==** 'Male'])

female **=** len(cleveland[cleveland['sex'] **==** 'Female'])

ax **=** plt.pie(x**=**[male, female], explode**=**(0, 0), labels**=**['Male', 'Female'], autopct**=**'%1.2f%%', startangle**=**90)

plt.show()



absence **=** cleveland[cleveland["class"] **==** 'Absence']["sex"].sort\_values()

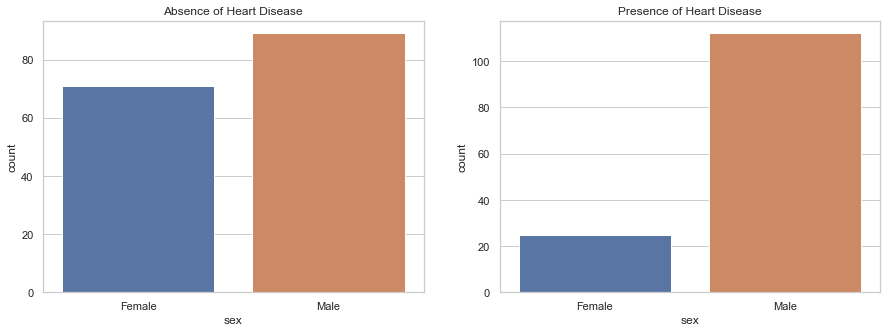
presence **=** cleveland[cleveland["class"] **==** 'Presence']["sex"].sort\_values()

f, axes **=** plt.subplots(1, 2, figsize **=** (15,5))

sns.countplot(absence, data**=**cleveland,ax**=**axes[0]).set\_title('Absence of Heart Disease')

sns.countplot(presence, data**=**cleveland,ax**=**axes[1]).set\_title('Presence of Heart Disease')

plt.show()



*# Chi-square test of independence of variables*

cont **=** pd.crosstab(cleveland['sex'], cleveland['class'])

chi\_stat **=** stats.chi2\_contingency(cont)

print(f'Chi statistics is {chi\_stat[0]} and p value is {chi\_stat[1]}')

*# Null hypothesis : Gender is not associated with Goal*

*# Alternate hypothesis : Gender is associated with Goal*

Chi statistics is 21.851612168613475 and p value is 2.945690038078843e-06

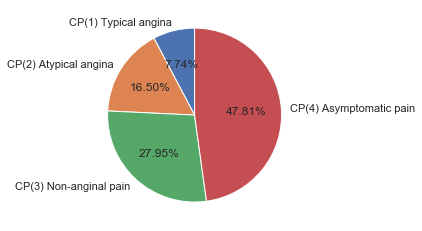
**INSIGHT:** As expected, given the low p-value(2.666712348180942e-06), so we reject null hypothesis and the test result detect a significant relationship between Sex and Class.

**Chest Pain Type**

x **=** [len(cleveland[cleveland['cp'] **==** 'Typical angina']),len(cleveland[cleveland['cp'] **==** 'Atypical angina']), len(cleveland[cleveland['cp'] **==** 'Non-anginal pain']), len(cleveland[cleveland['cp'] **==** 'Asymptomatic pain'])]

plt.pie(x, data**=**cleveland, labels**=**['CP(1) Typical angina', 'CP(2) Atypical angina', 'CP(3) Non-anginal pain', 'CP(4) Asymptomatic pain'], autopct**=**'%1.2f%%',startangle**=**90)

plt.show()



f, axes **=** plt.subplots(1,2,figsize**=**(15,5))

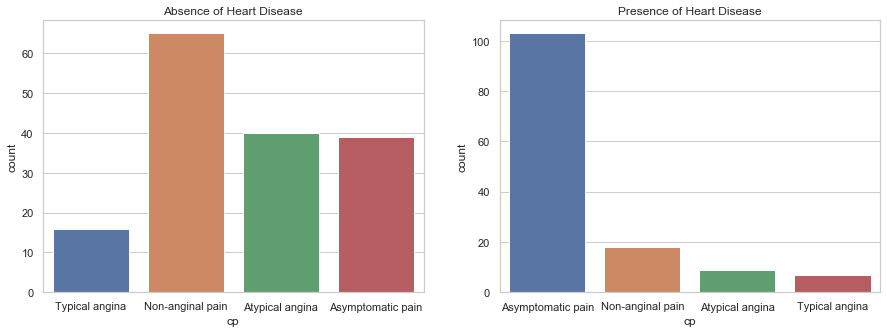
absence **=** cleveland[cleveland["class"]**==**'Absence']["cp"]

presence **=** cleveland[cleveland["class"]**==**'Presence']["cp"]

sns.countplot(absence, data**=**cleveland,ax**=**axes[0]).set\_title('Absence of Heart Disease')

sns.countplot(presence, data**=**cleveland,ax**=**axes[1]).set\_title('Presence of Heart Disease')

plt.show()



*# Chi-square test of independence of variables*

cont **=** pd.crosstab(cleveland['cp'], cleveland['class'])

chi\_stat **=** stats.chi2\_contingency(cont)

print(f'Chi statistics is {chi\_stat[0]} and p value is {chi\_stat[1]}')

*# Null hypothesis : Chest Pain is not associated with Goal*

*# Alternate hypothesis : Chest Pain is associated with Goal*

Chi statistics is 77.27579978222383 and p value is 1.1782838465918115e-16

**INSIGHT:** As expected, given the low p-value, so we reject null hypothesis and the test result detect a significant relationship between cp and class. Presence of disease graph have 104 patients with Chest pain type 4 much higher as compared to other chest pain. So, Asymptomatic pain can high predictive power.

**Fasting Blood Sugar**

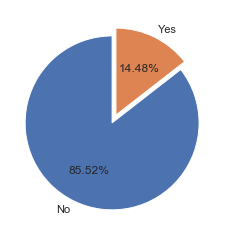
sizes **=** [len(cleveland[cleveland['fbs'] **==** 'No']), len(cleveland[cleveland['fbs'] **==** 'Yes'])]

labels **=** ['No', 'Yes']

plt.pie(x**=**sizes, labels**=**labels, explode**=**(0.1, 0), autopct**=**"%1.2f%%", startangle**=**90)

plt.show()

*# Fbs: fasting blood sugar > 120 mg/dl (1 = true; 0 = false)*



f, axes **=** plt.subplots(1,2,figsize**=**(15,5))

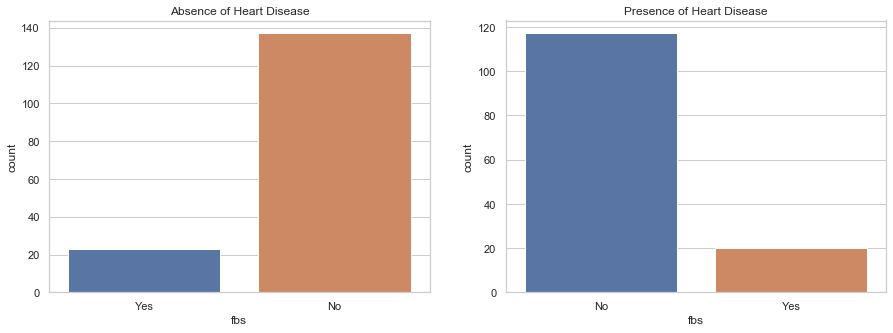
absence **=** cleveland[cleveland["class"]**==**'Absence']["fbs"]

presence **=** cleveland[cleveland["class"]**==**'Presence']["fbs"]

sns.countplot(absence, data**=**cleveland,ax**=**axes[0]).set\_title('Absence of Heart Disease')

sns.countplot(presence, data**=**cleveland,ax**=**axes[1]).set\_title('Presence of Heart Disease')

plt.show()



*# Chi-square test of independence of variables*

cont **=** pd.crosstab(cleveland['fbs'],cleveland['class'])

chi\_stat **=** stats.chi2\_contingency(cont)

print(f'Chi statistics is {chi\_stat[0]} and p value is {chi\_stat[1]}')

*# Null hypothesis : FBS is not associated with Goal*

*# Alternate hypothesis : FBS is associated with Goal*

Chi statistics is 0.01228173970336428 and p value is 0.91175669783748

**INSIGHT:** As expected, given the high p-value, so we fail to reject null hypothesis and the test result detect a non-significant relationship between Fbs and Goal. Most individuals did not have fasting blood sugar levels greater than 120 mg/dL. This did not change greatly when the data was divided based on the presence of disease.So, FBS is not a predictive feature.

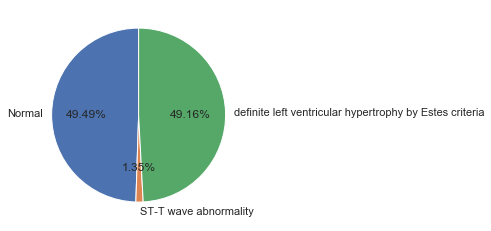
**Resting ECG Results**

sizes **=** [len(cleveland[cleveland['restecg'] **==**'Normal']), len(cleveland[cleveland['restecg']**==**'Abnormality']), len(cleveland[cleveland['restecg']**==**'Hypertrophy'])]

labels **=** ['Normal', 'ST-T wave abnormality', 'definite left ventricular hypertrophy by Estes criteria']

plt.pie(x**=**sizes, labels**=**labels, explode**=**(0, 0, 0), autopct**=**"%1.2f%%", startangle**=**90)

plt.show()



f, axes **=** plt.subplots(1,2,figsize**=**(15,5))

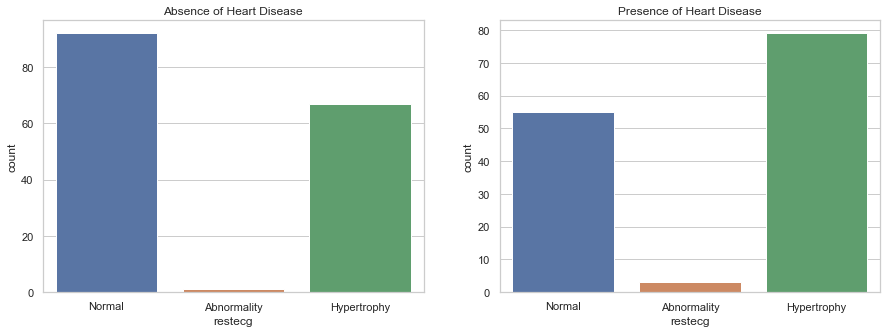
absence **=** cleveland[cleveland["class"]**==**'Absence']["restecg"]

presence **=** cleveland[cleveland["class"]**==**'Presence']["restecg"]

sns.countplot(absence, data**=**cleveland,ax**=**axes[0],order**=**['Normal', 'Abnormality', 'Hypertrophy']).set\_title('Absence of Heart Disease')

sns.countplot(presence,ax**=**axes[1],order**=**['Normal', 'Abnormality', 'Hypertrophy']).set\_title('Presence of Heart Disease')

plt.show()



print(f'Probability of Hypertropy in disease cohorts {presence[presence**==**"Hypertrophy"].value\_counts()**/**len(presence)}')

print(f'Probability of Hypertropy in non-disease cohorts {absence[absence**==**"Hypertrophy"].value\_counts()**/**len(absence)}')

Probability of Hypertropy in disease cohorts Hypertrophy 0.576642

Name: restecg, dtype: float64

Probability of Hypertropy in non-disease cohorts Hypertrophy 0.41875

Name: restecg, dtype: float64

cont **=** pd.crosstab(cleveland['restecg'],cleveland['class'])

chi\_stat **=** stats.chi2\_contingency(cont)

print(f'Chi statistics is {chi\_stat[0]} and p value is {chi\_stat[1]}')

*#Null hypothesis : Exang is not associated with Goal*

*#Alternate hypothesis : Exang is associated with Goal*

Chi statistics is 9.575507229251564 and p value is 0.008331151353680854

**INSIGHT**: Most patients exhibited normal resting electrocardiograhic results . However, a higher proportion of diseased patients had hypertropy suggesting that this feature may contribute some predictive power.

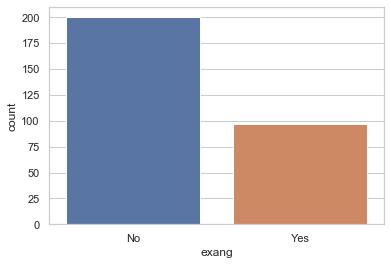
**Exercise Induced Angina**

sns.countplot(data **=** cleveland , x **=** 'exang')

*# exercise induced angina (1 = yes; 0 = no)*

Out[42]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x1e913f88448>



f, axes **=** plt.subplots(1,2,figsize**=**(15,5))

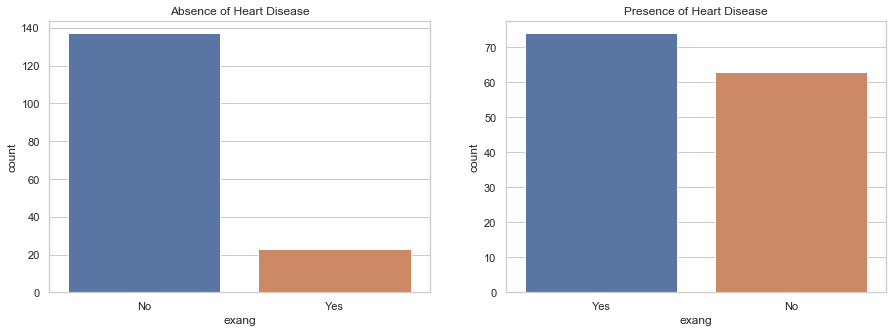
absence **=** cleveland[cleveland["class"]**==**'Absence']["exang"]

presence **=** cleveland[cleveland["class"]**==**'Presence']["exang"]

sns.countplot(absence, data**=**cleveland,ax**=**axes[0]).set\_title('Absence of Heart Disease')

sns.countplot(presence, data**=**cleveland,ax**=**axes[1]).set\_title('Presence of Heart Disease')

plt.show()



*# Chi-square test of independence of variables*

cont **=** pd.crosstab(cleveland['exang'], cleveland['class'])

chi\_stat **=** stats.chi2\_contingency(cont)

print(f'Chi statistics is {chi\_stat[0]} and p value is {chi\_stat[1]}')

*# Null hypothesis : Exang is not associated with Goal*

*# Alternate hypothesis : Exang is associated with Goal*

Chi statistics is 50.9425597633616 and p value is 9.510884265909016e-13

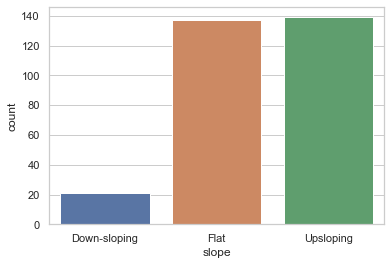
**INSIGHT:** As expected, given the low p-value, so we reject null hypothesis and the test result detect a significant relationship between Exang and Goal. Significantly more patients in the diseased cohort displayed exercise induced angina. This feature should be strongly predictive.

**Peak Exercise ST Segment**

sns.countplot(data **=** cleveland , x **=** 'slope')

Out[45]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x1e9139fbac8>



f, axes **=** plt.subplots(1,2,figsize**=**(15,5))

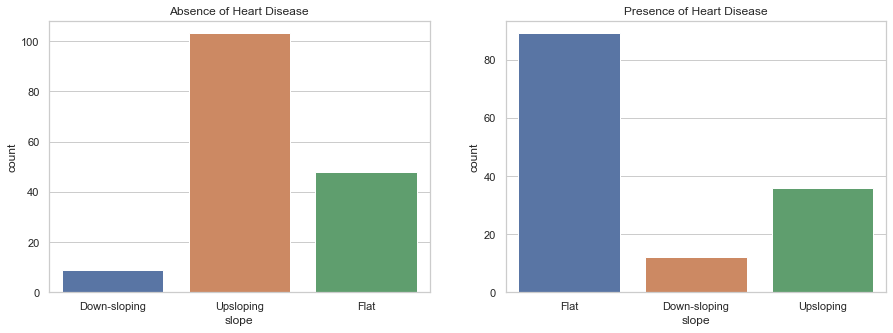
absence **=** cleveland[cleveland["class"]**==**'Absence']["slope"]

presence **=** cleveland[cleveland["class"]**==**'Presence']["slope"]

sns.countplot(absence, data**=**cleveland,ax**=**axes[0]).set\_title('Absence of Heart Disease')

sns.countplot(presence, data**=**cleveland,ax**=**axes[1]).set\_title('Presence of Heart Disease')

plt.show()



*# Chi-square test of independence of variables*

cont **=** pd.crosstab(cleveland['slope'],cleveland['class'])

chi\_stat **=** stats.chi2\_contingency(cont)

print(f'Chi statistics is {chi\_stat[0]} and p value is {chi\_stat[1]}')

*#Null hypothesis : Slope is not associated with Goal*

*#Alternate hypothesis : Slope is associated with Goal*

Chi statistics is 43.47317755212573 and p value is 3.630107106911135e-10

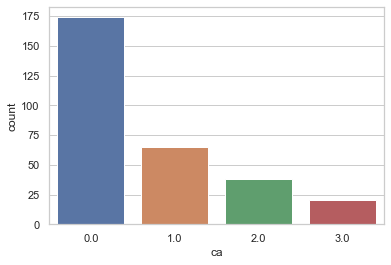
**INSIGHT:** As expected, given the low p-value, so we reject null hypothesis and the test result detect a significant relationship between Slope and Goal. Significantly more patients in the non-diseased cohort displayed Slope-Flat. This feature could be strongly predictive. The slope of the peak exercise ST segment differed between the non-disease and diseased cohorts with the majority of cardiac disease patients exhibiting a flat ST slope(value = 2).This can also have good predictive power.

**Number of Blood Vessels**

sns.countplot(data **=** cleveland , x **=** 'ca')

Out[48]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x1e913f1d4c8>



f, axes **=** plt.subplots(1,2,figsize**=**(15,5))

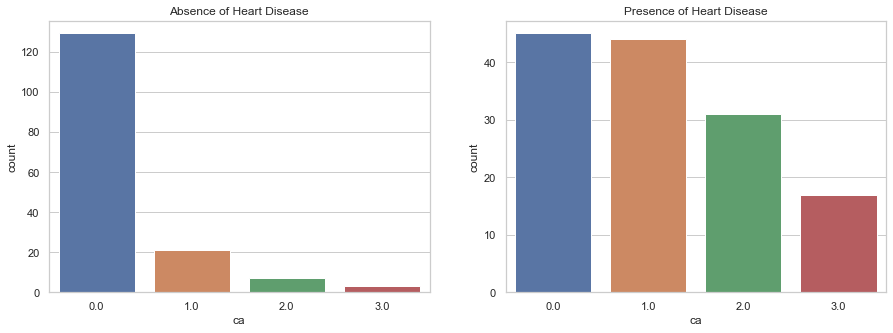
absence **=** cleveland[cleveland["class"]**==**'Absence']["ca"]

presence **=** cleveland[cleveland["class"]**==**'Presence']["ca"]

sns.countplot(absence, data**=**cleveland,ax**=**axes[0]).set\_title('Absence of Heart Disease')

sns.countplot(presence, data**=**cleveland,ax**=**axes[1]).set\_title('Presence of Heart Disease')

plt.show()



*# Chi-square test of independence of variables*

cont **=** pd.crosstab(cleveland['ca'],cleveland['class'])

chi\_stat **=** stats.chi2\_contingency(cont)

print(f'Chi statistics is {chi\_stat[0]} and p value is {chi\_stat[1]}')

*# Null hypothesis : CA is not associated with Goal*

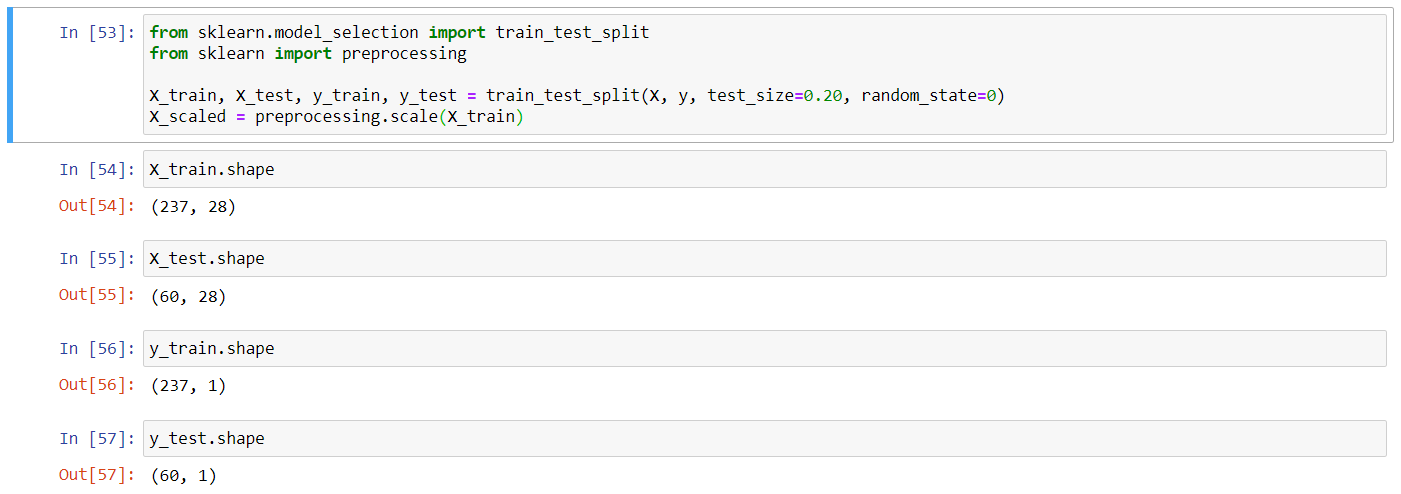
*# Alternate hypothesis : CA is associated with Goal*

Chi statistics is 72.30053062466945 and p value is 1.37257765344901e-15

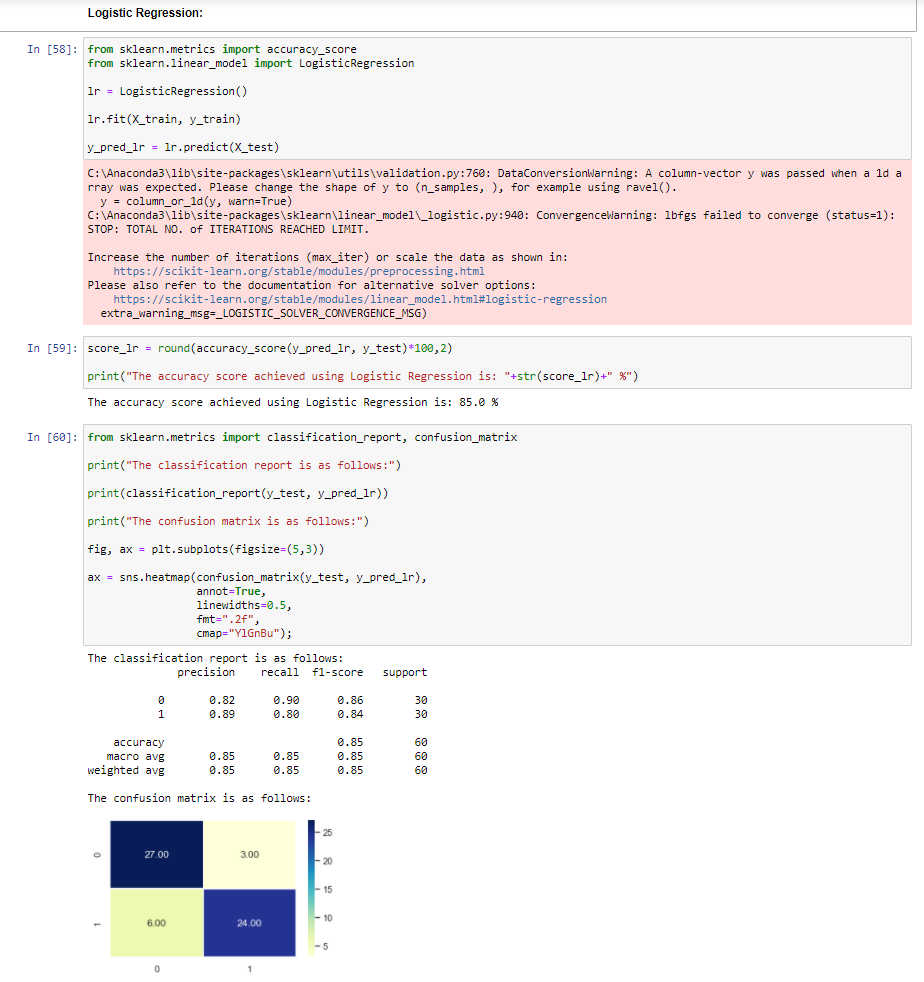
**INSIGHT:** As expected, given the low p-value, so we reject null hypothesis and the test result detect a significant relationship between CA and Goal. Significantly more patients in the diseased cohort has number of blood vessels greater than 1. This feature should be strongly predictive.

### In-Depth Analysis:

First of all I have split the data into 80/20 ratio i.e. 80% of the data has been used to train the model and remaining 20% used for the testing purpose.

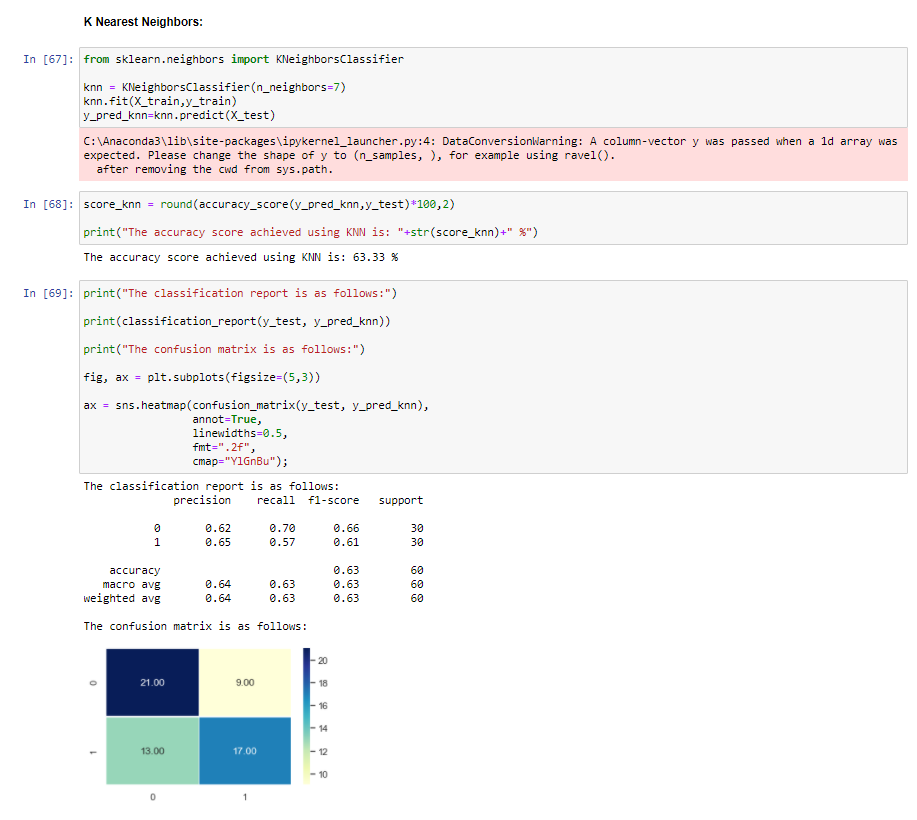


I have applied various algorithms like Linear Regression, Naive Bayes, SVM, KNN, Decision Tree, XGBoost, Neural Neural and Random Forest on the data-set and tested them for their accuracy. Finally I have compared their accuracy. The accuracy can be further increased by performing hyper parameter tuning.



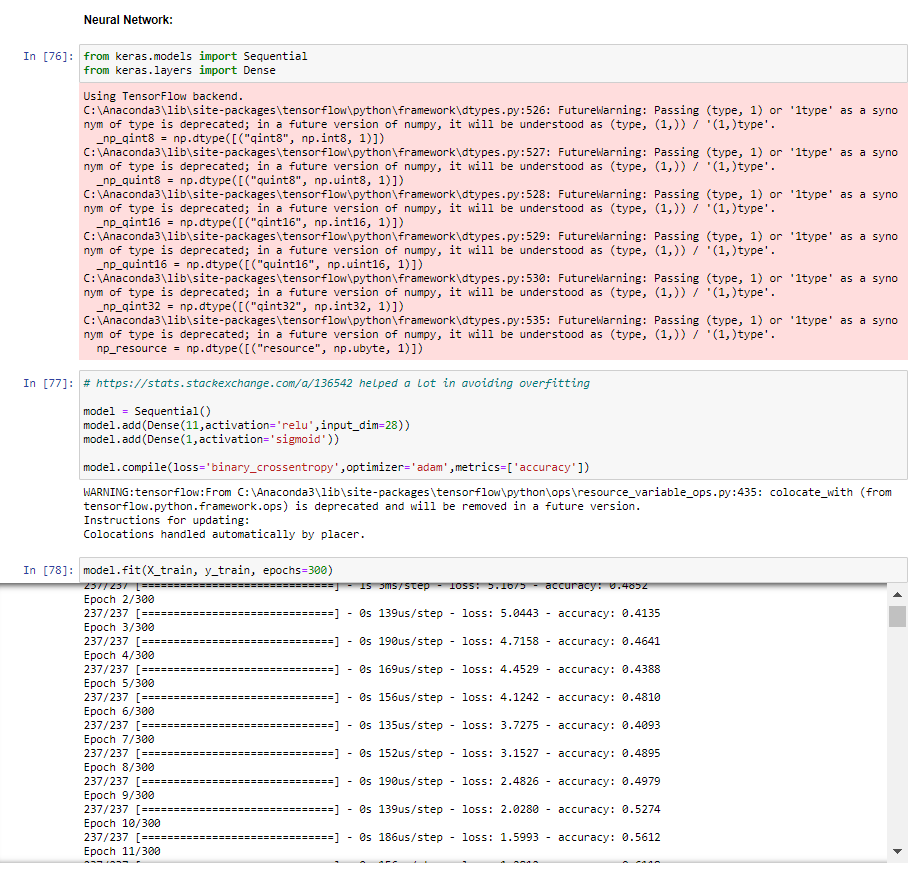




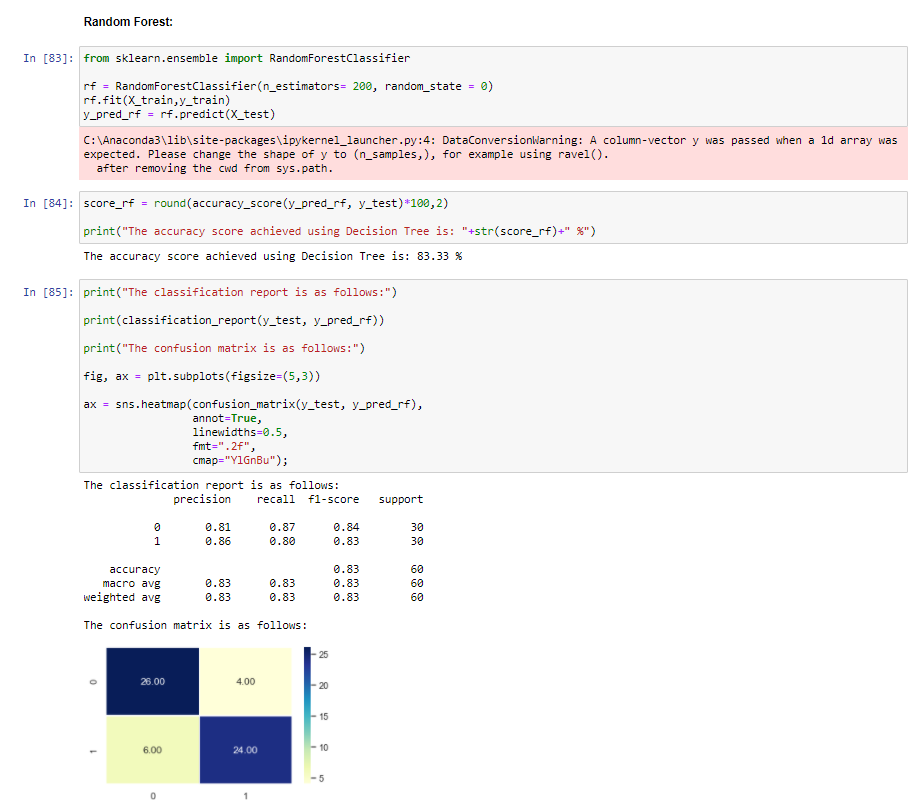












Finally the accuracy of all the models have been compared against one another.

