Heart Disease Prediction

CAPSTONE PROJECT

Submitted in partial fulfillment of the requirements of the

Post Graduate Certification Program
in
Artificial Intelligence and Machine Learning

By

Srinivasa Bharath K - 2019AIML557

Balasaravanan V - 2019AIML670

Jennifer Susa Sen - 2019AIML517

Pinky Singha Roy - 2019AIML574

Akhil John Mattam - 2019AIML583

Under the supervision of

Satyaki Dasgupta

Project work carried out at

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
Pilani (Rajasthan) INDIA

(Mar,2020-Apr, 2021)

PCAM ZC321 CAPSTONE PROJECT

Heart disease Prediction

Submitted in partial fulfillment of the requirements of the

PGP - Artificial Intelligence and Machine Learning

By

Srinivasa Bharath K - 2019AIML557

Balasaravanan V - 2019AIML670

Jennifer Susa Sen - 2019AIML517

Pinky Singha Roy - 2019AIML574

Akhil john Mattam - 2019AIML583

Under the supervision of

Satyaki Dasgupta

Project work carried out at

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI (RAJASTHAN)

(Mar,2020-Apr, 2021)

Acknowledgements

We would like to extend our deep sense of gratitude to Mr. Satyaki Das Gupta, Mentor, to encourage us to the highest peak and provide us with guidance and kind supervision in the completion of our project.

We would like to extend our sincere thanks to Prof. Sugata Ghosal, for his encouragement and guidance.

We would like to extend our deep sense of gratitude to Prof. Bhanu Murthy, Head of the Course for all the guidance.

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

CERTIFICATE

This is to certify that the Capstone Project entitled Heart Disease Prediction and submitted by Mr. SRINIVASA BHARATH K ID No. 2019AIML557, Mr. BALASARAVANAN V ID No. 2019AIML670, Ms. JENNIFER SUSA SEN ID No. 2019AIML517, Ms. PINKY SINGHA ROY ID No. 2019AIML574 and Mr. AKHIL JOHN MATTAM,ID No.2019AIML583 in partial fulfilment of the requirements of PCAM ZC321 Capstone Project, embodies the work done by him/her under my supervision.

Place : BENGALURU ,KOLKATA

Date : 11-04-2021

Signature of the Mentor

Name : Satyaki Dasgupta

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

CERTIFICATE

This is to certify that the Capstone Project entitled HEART DISEASE PREDICTION and submitted by

Mr. SRINIVASA BHARATH K ID No. 2019AIML557 in partial fulfilment of the requirements of

PCAM ZC321 Capstone Project, embodies the work done by him/her under my supervision.

Place : BENGALURU Signature of the Mentor

Date : 11-04-2021 Name : Satyaki Dasgupta

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

CERTIFICATE

This is to certify that the Capstone Project entitled HEART DISEASE PREDICTION and submitted by

Mr. BALASARAVANAN V ID No. 2019AIML670 in partial fulfilment of the requirements of

PCAM ZC321 Capstone Project, embodies the work done by him/her under my supervision.

Place : BENGALURU Signature of the Mentor

Date : 11-04-2021 Name : Satyaki Dasgupta

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

CERTIFICATE

This is to certify that the Capstone Project entitled HEART DISEASE PREDICTION and submitted by

Ms. JENNIFER SUSA SEN ID No. 2019AIML517 in partial fulfilment of the requirements of

PCAM ZC321 Capstone Project, embodies the work done by him/her under my supervision.

Place : BENGALURU Signature of the Mentor

Date : 11-04-2021 Name : Satyaki Dasgupta

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

CERTIFICATE

This is to certify that the Capstone Project entitled HEART DISEASE PREDICTION and submitted by

Ms. PINKY SINGHA ROY ID No. 2019AIML574 in partial fulfilment of the requirements of

PCAM ZC321 Capstone Project, embodies the work done by him/her under my supervision.

Place : KOLKATA

Date : 11-04-2021

Signature of the Mentor

Name : Satyaki Dasgupta

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

CERTIFICATE

This is to certify that the Capstone Project entitled HEART DISEASE PREDICTION and submitted by

Mr. AKHIL JOHN MATTAM ID No. 2019AIML583 in partial fulfilment of the requirements of

PCAM ZC321 Capstone Project, embodies the work done by him/her under my supervision.

Place : BENGALURU

Date : 11-04-2021

Signature of the Mentor

Name : Satyaki Dasgupta

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

SECOND SEMESTER 2020-21

PCAM ZC321 CAPSTONE PROJECT

Project Title : HEART DISEASE PREDICTION

Name of Mentor : SATYAKI DASGUPTA

Name of Student :Mr. SRINIVASA BHARATH K, Mr. BALASARAVANAN V, Ms. JENNIFER SUSA SEN, Ms. PINKY SINGHA ROY, Mr. AKHIL JOHN MATTAM

ID No. of Student : 2019AIML557 , 2019AIML670 , 2019AIML517 , 2019AIML574 , 2019AIML583

PCAM ZC321 Capstone Project – Heart Disease Prediction

Abstract

The World Health Organization has estimated 12 million deaths occur worldwide; every year due to Heart diseases. Half the deaths in the United States and other developed countries are due to cardiovascular diseases. The early prognosis of cardiovascular diseases can aid in making decisions on lifestyle changes in high-risk

patients and in turn reduce the complications. This research intends to pinpoint the most relevant/risk factors of heart disease as well as predict the overall risk

Key Words:

Blood pressure,Hypertension,diabetes,Total cholesterol,Body mass index (BMI), Heart Rate and Coronary heart disease

List of Symbols and Abbreviations Used

- AIML – Artificial Intelligence and Machine Learning
- ML – Machine Learning
- PCA – Principal Component Analysis
- SMOTE - Synthetic Minority Over-sampling Technique
- CHD- Coronary heart disease
- LR – Logistic Regression
- RF – Random Forest
- SVM – Support Vector Machine
- NB – Naïve Bayes
- XGB - eXtreme Gradient Boosting
- RFE – Recursive Feature Elimination
- GED – Graduate Equivalency Degree

List of Tables

List of Figures

Table of Contents

Problem Statement

The World Health Organization has estimated 12 million deaths occur worldwide; every year due to Heart diseases. Half the deaths in the United States and other developed countries are due to cardiovascular diseases. The early prognosis of cardiovascular diseases can aid in making decisions on lifestyle changes in high-risk patients and in turn reduce the complications. This research intends to pinpoint the most relevant/risk factors of heart disease as well as predict the overall risk

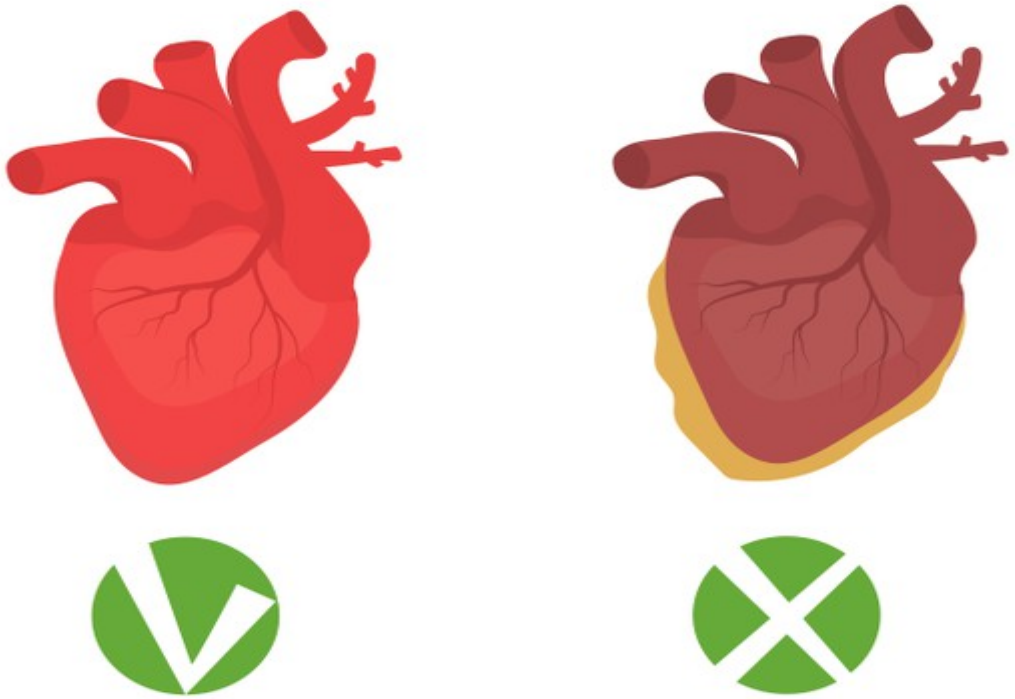

Figure 1 Problem Statement

PCAM ZC321 Capstone Project – Heart Disease Prediction

There are various types of heart diseases, some of them include Coronary artery disease, Arrhythmia, Dilated cardiomyopathy, Hypertrophic cardiomyopathy, Mitral valve regurgitation, Myocardial infarction and many others. The underlying cause of heart diseases are

- high blood pressure
- high cholesterol
- smoking
- a high intake of alcohol
- overweight and obesity
- diabetes
- a family history of heart disease
- dietary choices
- age
- a history of preeclampsia during pregnancy
- low activity levels
- high stress and anxiety levels

Objective of Project

- We have to predict whether the patient has 10-year risk of future coronary heart disease (CHD)
- We are using the data provided by the hospital to make necessary transformations and train the models to predict whether the patient in the future will have a coronary heart disease (CHD)
- Looking for patterns in the dataset or similar patients to help make predictions
- Heart disease describes a range of conditions that affect your heart.
- Heart disease is one of the biggest causes of morbidity and mortality among the population of the world
- Machine learning (ML) proves to be effective in assisting in making decisions and predictions for the healthcare industry

Background of previous work done in the chosen area

• As this problem has been around for quite some time there were few papers that provided some insight to the problem we were handling. The paper titled Prediction of heart disease and classifiers' sensitivity analysis by Khaled Mohamad Almustafa, provided a comparative analysis of different classifiers was performed for the classification of the Heart Disease dataset in order to correctly classify and or predict HD cases with minimal attributes

• Another worth mentioning paper is by Marimuthu Muthuvel, titled Analysis of Heart Disease Prediction using Various Machine Learning Techniques, helped understand how data is not always made use to the full extent and is often underutilized and how to tackle this problem.

• Many risk factors in heart disease and a need of the time to get accurate, reliable, and sensible approaches to make an early diagnosis to achieve prompt management of the disease requires basic knowledge about what vitals might potentially play a part in the prediction of a Coronary Heart Disease

• Our primary aim to have the results presented in this study be the baseline for any future work to compare.

Machine Learning process flow

Explaining the various steps involved

1. With some foundational knowledge about various health related terms like Blood Sugar, Blood Pressure used by doctors to analyse the various functions of the body and its compositions.
2. The data in its raw form is not suitable for modeling. In order to make it fit for use, we clean the data. Some of the techniques used to clean the data are imputation, Scaling, Encoding, Outlier Detection, etc
3. visualization is important for various reasons, some are listed below
    a. Identify areas that need attention or improvement.
    b. Clarify which factors influence the outcome.
    c. Help you understand the data.
    d. Handy for Predict.
4. This data is then passed through various models that are trained to recognize certain types of patterns.
5. Evaluation phase aims to estimate the generalization accuracy of a model on future data

Figure 2 ML Workflow

Resources needed for the Project

Hardware:

Basic Laptop/Desktop

16 GB RAM

Software:

Python 3 & Above

PyCharm/Jupyter Notebook/Google Colab

OS Windows or Mac

Roles in the Machine Learning Project:

- Business Analyst:
    - Business analysts are responsible for converting particular business challenges into well-defined analysis plans.
- Data Architect:
    - Data architects are responsible for designing and operating the underlying platform and infrastructure that supports data science work.
- Machine learning engineer/Data Engineer:
    - Data engineers are responsible for data ingestion, processing, and storage. They build data pipelines which make data accessible for data scientists to work with.
- Data Scientist:
    - A Data Scientist uses advanced analytics technologies, including Machine Learning and Predictive Modeling to collect, analyze and interpret large amounts of data and produce actionable insights. These are then used to make business decisions by the company executives.
- Software Engineer:
    - The developer is responsible for taking a deployed model and embedding it into an application that end-users can interact with, or in a product or system that can consume the model.

Potential data challenges & risks

- No access to the original data as summarized data has been provided.
- Lack of the necessary data, the dataset was imbalanced, missing pieces of data that was required for this project
- Being medical data, the handling of missing values was a major challenge. The options available like imputation or dropping the values are difficult to decide as the data was probably missing because that feature was not measured but it could have something to offer while training the data.

Detailed Plan of Work

Table 1 Detailed Plan of Work

Activities	W 1	W 2	W 3	W 4	W 5	W 6	W 7	W 8
BUSINESS UNDERSTANDING								
Explain and Discuss the problem statement								
DATA UNDERSTANDING								
Data Source								
Understanding of data structure								
Environment								
Upload data								
Data Cleanup								
Split dataset between training and test in the ratio of 80:20								
Drop columns which are not required								
Find missing value								
Data Visualization								
Visualize missing value								
Impute missing value								
Reviews along with BITS Faculty								
MODELING								
Feature Engineering								
Preprocessing								
Encoding								
Scaling								
Outlier								
Visualize Outlier								
Correlation								
Visualize Correlation								
Preparation								

Principal Component Analysis								
Perform PCA								
Draw scree plot to find how many components should be selected for								
Find contribution of columns on each dimension								
Find Correlation between columns and dimensions								
Find contribution of rows on each dimension								
Find Correlation between rows and dimensions								
Visualize PCA								
Reviews along with BITS Faculty								
Model Training								
Apply following model								
Logistic Regression								
Random Forest								

SVM								
Reviews along with BITS Faculty								
XGBoost								
Random Forest with GridSearch								
Naive Bayes								
Model Evaluation								
Predict								
Run the model against test data								
Calculate the Accuracy								
Generate confusion matrix								
Draw AUC for each model								
Display accuracy of each model in a tabular form								
Preparation & submission of Solution								
Review of solution								
Viva Voce								

PCAM ZC321 Capstone Project – Heart Disease Prediction

Pre-Processing Steps

Observations of dataset

- Attribute “male” is a nominal attribute. 1 represents patient is male and 0 represents patient is female.
- Attribute “age” is a discrete attribute where minimum is 32 and maximum age is 70 years.
- Attribute “education” is an ordinal attribute. This column data is categorized into four.
  1 = Some High School; 2 = High School or GED; 3 = Some College or Vocational School; 4 = college
- Attribute “currentSmoker” is a nominal attribute, 1 represents the person is current smoker and 0 represents person is not current smoker.
- Attribute “cigsPerDay” is a discrete attribute where the minimum number of cigarettes per day is 0 and maximum number of cigarettes per day is 70.
- Attribute “BPMeds” is a nominal attribute. If the value is 1 then the patient is on blood pressure medication, if it is 0 the patient is not on blood pressure medication.
- Attribute “prevalentStroke” is a nominal attribute.if the value is 1 then person had previously heart stroke.If it is 0 then person not suffered from the stroke.
- Attribute “prevalentHyp” is a nominal attribute.If it is 1 then the person is hypertensive if 0 then the person is not hypertensive.
- Attribute “diabetes” is a nominal attribute.1 represent the patient is diabetic and if the value is 0 patient is not a diabetic.
- Attributes totChol(Total cholesterol),sysBP,diaBP,BMI,heartRate,glucose are continuous attributes.
- Attribute TenYearCHD(Coronary Heart Disease) is a discrete attribute.This is the target attribute for this data.

Data processing

Dataset given contains:4238 records with 16 columns		
Feature Name	**Description & Unique Values**	**Missing Data**
male	Gender	NIL
age	Min age is 32 and maximum is 70	NIL
education	1 = Some High School; 2 = High School or GED; 3 = Some College or Vocational School; 4 = college	105(2.4%)
currentSmoker	Person is current smoker or not	NIL
cigsPerDay	How many cigarette smokes per day	29(.6%)
BPMeds	Blood pressure of the person	53(1.25%)
prevalentStroke	Having Prevalent stroke or not	NIL
prevalentHyp	Having prevalent Hypertension or not	Nil
diabetes	Having diabetes or not	NIL
totChol	Total cholesterol of a person	50(1.15%)
sysBP	Blood pressure of a person	NIL
diaBP	Current diabetes value	NIL
BMI	Body mass index of a person	19(.4%)
heartRate	Heart rate of a person	1(0.02%)
glucose	Glucose levels of a person	388(9.15%)
TenYearCHD	Coronary heart disease	NIL

Table 2 Data Analysis

Feature Pre-processing

- In the given dataset checking that any feature has relatively large missing values.Checking the threshold value of 30%.If any feature/column has more than 30% missing data then deleting that column.

```
colsToDrop = missingCols[missingCols > threshold].index.values
```

- Used MICE(Multiple Imputation by Chained Equations) technique to fill the missing value data of Numerica datal.

```
imputedData = missingValueObj.imputeByMice(cleanData, numericCols)
```

Encoding

- In the given dataset there exists seven categorical features.In this except education remaining features have only two unique values education has 4 unique values.
- For all this categorical features are label encoded.

```
labelEncodedImputedData = encoderObj.labelEncoder(imputedData, categoricalCols)
```

Scaling

- Encoded data is now scaled using MinMaxScaler of sklearn preprocessing module.

```
scaler = MinMaxScaler(feature_range=(0, 1))
scaledData = scaler.fit_transform(d1)
scaledData = pd.DataFrame(scaledData, columns=d1.columns)
```

- Hence all the features will take values in range of [0-1]

Outlier Detection

- The scaled data is now used to detect outliers
- Outliers are first visualized using seaborn boxplot to get a better idea

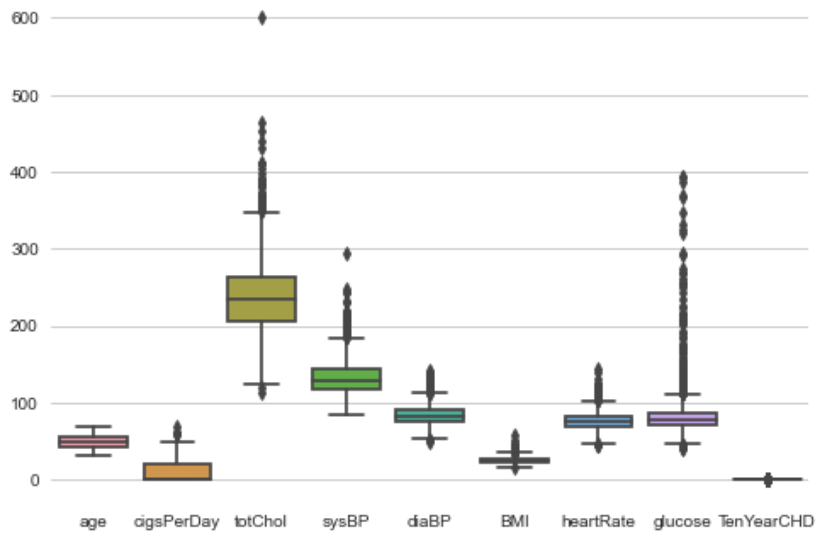

Figure 3 Box Plot - Outliers

- Outliers are detected in 3 different ways
  a. Z-score based outlier detection
  b. IQR (Inter Quartile Range) based outlier detection
  c. DB Scan (Density based scan) outlier detection

Z-score outliers

- The data points which are 3 standard deviations away from mean are considered as outliers

```
outlierData = data[(zScore < 3).all(axis=1)]
```

- Hence the data points which have z_score >3 (calculated from scipy module stats function) are considered as outliers
- With the scaled data 431 outliers are detected in the given data

IQR outliers

- For each feature, 25th quartile and 75th quantile is determined and IQR is calculated using

$$IQR = 75th\ Quartile - 25th\ Quartile$$

- For each feature the upper limit is considered as 75th Quartile + IQR x 1.5 and lower limit as 25th Quartile - IQR x 1.5

```
outlierData = data[~((data < (quantile1 - 1.5 * IQR)) | (data > (quantile3 + 1.5 * IQR))).any(axis=1)]
```

- The data points outside this limit are considered as outliers

DB Scan Outliers

- Choose a value for eps and MinPts
- For a particular data point (x) calculate its distance from every other datapoint.
- Find all the neighbourhood points of x which fall inside the circle of radius (eps) or simply whose distance from x is smaller than or equal to eps.
- Treat x as visited and if the number of neighbourhood points around x are greater or equal to MinPts then treat x as a core point and if it is not assigned to any cluster, create a new cluster and assign it to that.
- If the number of neighbourhood points around x are less than MinPts and it has a core point in its neighbourhood, treat it as a border point.
- Include all the density connected points as a single cluster. (What density connected points mean is described later)
- Repeat the above steps for every unvisited point in the data set and find out all core, border and outlier points.

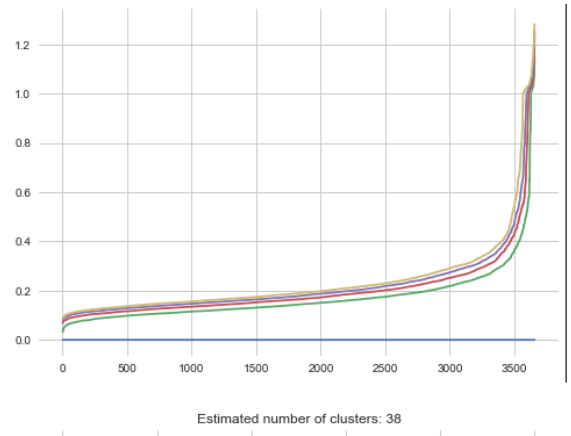

Figure 4 Eps for DB Scan

- With this approach we noticed that there are 355 outliers in the provided data.

Figure 5 DB Scan outliers

Principal Component Analysis (PCA)

It is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set

Steps -

● Scaled data should be used for PCA. So fit the scaled data . Then transform and shape the columns of the scaled data. Target feature is dropped and passed to sklearn decomposition PCA module

```
       PCA0      PCA1      PCA2      PCA3      PCA4      PCA5      PCA6  \
0  0.092669 -0.086193 -0.868281  0.636677 -0.194255  0.037603  0.007445
1 -0.545206 -0.471654 -0.164493 -0.019403 -0.089576  0.014315 -0.009261
2  0.799802  0.022705 -0.127254 -0.380037  0.010552 -0.017296  0.031086
3 -0.019101  0.342885  0.907740  0.405880  0.253579 -0.130124 -0.091099
4  0.274503 -0.555707  0.480372  0.299015  0.064127  0.002662 -0.010294

       PCA7      PCA8      PCA9     PCA10     PCA11     PCA12     PCA13  \
0  0.012300 -0.110026 -0.047147  0.047146 -0.087810 -0.000231 -0.000425
1  0.198647 -0.074903 -0.010558  0.068672 -0.038781  0.000888 -0.021004
2  0.029037  0.018951 -0.011229  0.010305  0.031399 -0.002853 -0.016978
3 -0.072359  0.102220  0.211211 -0.033680 -0.097743 -0.006833  0.051197
4  0.147532  0.003397  0.075119  0.010181  0.087008  0.008406  0.012861

      PCA14
0  0.017535
1 -0.012541
2  0.019931
3 -0.060449
4 -0.012079
        male       age  education  currentSmoker  cigsPerDay    BPMeds  \
0   0.569178 -0.112378   0.049292       0.729840    0.225041 -0.043288
1   0.514932  0.156679  -0.097365      -0.059717    0.019888  0.081136
2  -0.635795  0.009531  -0.066086       0.624466    0.127032  0.056626
3  -0.018162 -0.104631   0.982130      -0.012069   -0.006931  0.022218
4  -0.009424  0.941101   0.129001       0.096676   -0.008712  0.093346
```

Figure 6 PCA - 1

	BMI	heartRate	glucose
0	-0.027888	-0.013030	-0.008087
1	0.068051	0.017867	0.013689
2	-0.003563	0.058543	0.000564
3	-0.025128	-0.012655	-0.005467
4	-0.011407	-0.038830	0.055017
5	0.052403	0.071807	0.236169
6	-0.002762	-0.086057	-0.119946
7	0.237237	0.737308	0.002981
8	0.372749	-0.625884	-0.007960
9	0.037015	-0.190853	-0.017063
10	0.629494	0.005994	-0.012165
11	-0.628459	-0.089691	-0.000918
12	-0.017669	0.021562	0.008348
13	0.015927	-0.038962	0.900273
14	0.051770	0.003374	-0.339785

Figure 7 PCA - 2

● Variance are plotted against principal components using **Scree plot**

The **scree plot** is used to determine the number of factors to retain in an exploratory factor analysis (FA) or principal components to keep in a **principal component analysis** (**PCA**).

Percentage of Variance Explained as label at Y-axis. Principal Component as label at X-axis.

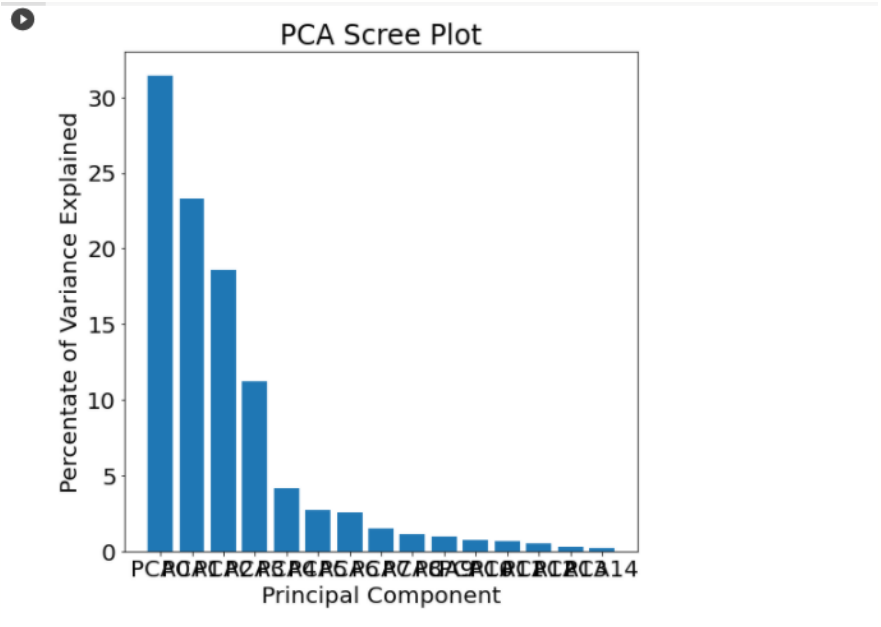

Figure 8 PCA - Scree Plot

Data Visualization

- Heart disease data has been visualized using various data visualization techniques:
  1. Categorical scatter plot with hue
     a. - Swarm plot
  2. Categorical distribution plot
     a. - Box
     b. - Violin
     c. - Boxen
  3. Categorical Estimate Plots
  4. Numerical scatterplot
  5. Pair plot
  6. Pandas plot
  7. Dist plot
  8. Rug plot
  9. Line plot

Categorical scatter plot with hue:-
Swarm Plot

```
def plot_swarm(data,numericCols,categoricalCols,response='TenYearCHD'):
    for i in numericCols:
                for j in categoricalCols:
                    #sns.jointplot(x=response, y=i, data=data)
                    #sns.catplot(x=j, y=i, hue=response,   data=data, kind='bar')
                    sns.kdeplot(data=data, x=i, hue=response, fill=True,
                                common_norm=False, palette="crest", alpha=.5, linewidth=0,)
                    sns.swarmplot(x=j, y=i, hue=response, data=data)
                    plt.show()
```

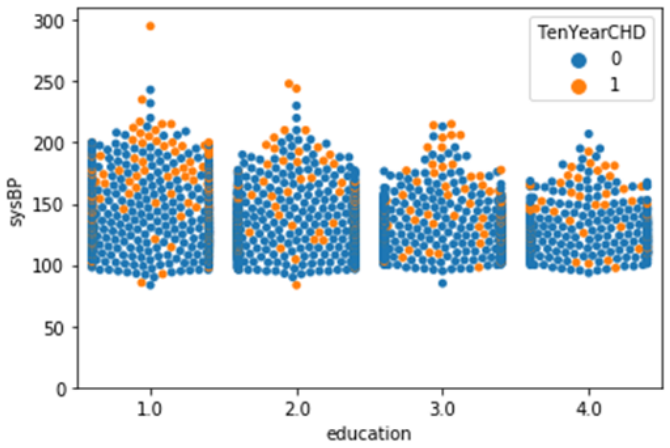

Figure 9 Swarm plot

- **swarm plot** is a graphical data analysis technique for summarizing a univariate data set.
- Swarm plot helps in visualizing different categorical variables

Categorical Distribution plot :-

Boxen Plot

```
def Boxen_plot(data,numericCols,response='TenYearCHD'):
    sns.set(rc={'figure.figsize':(11.7,8.27)})
    for i in range(len(numericCols)):

                for j in range(len(numericCols)-1):
                    #sns.jointplot(x=response, y=i, data=data)

                    sns.boxenplot(data=data, x=numericCols[i], y=numericCols[j+1], scale="linear")
                    #plt.figure(figsize=(16,6))
                    #ax.legend(title=response)  # add a title to the legend
                    plt.show()
```

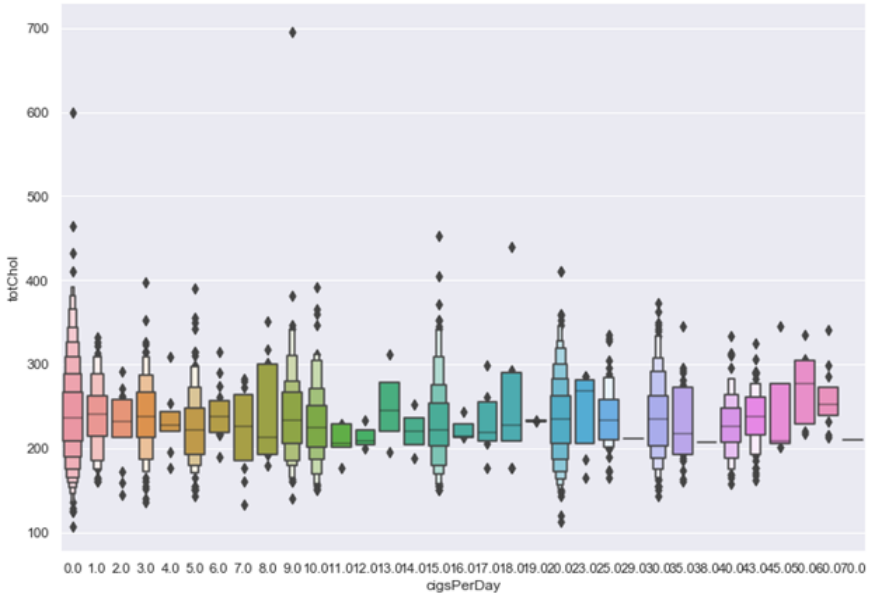

Figure 10 Boxen Plot

- By plotting different quartile values, It helps us to understand the shape of the distribution particularly in the head end and tail end

Box Plot

- boxplot is a method for graphically depicting groups of numerical data through their quartiles
- **Box plots** visually show the distribution of numerical data and skewness through displaying the data quartiles

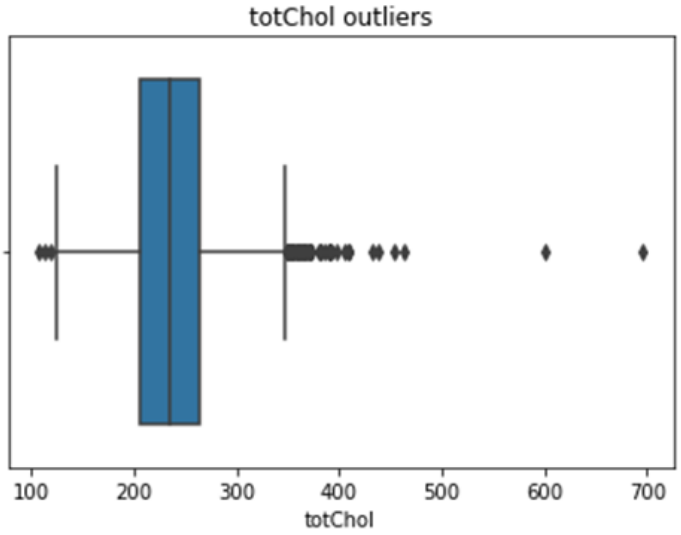

Figure 11 Box Plot

Violin Plot

- A violin plot is a method of plotting numeric data. It is similar to a box plot, with the addition of a rotated kernel density plot on each side
- While a box plot only shows summary statistics such as mean/median and interquartile ranges, the violin plot shows the full distribution of the data.

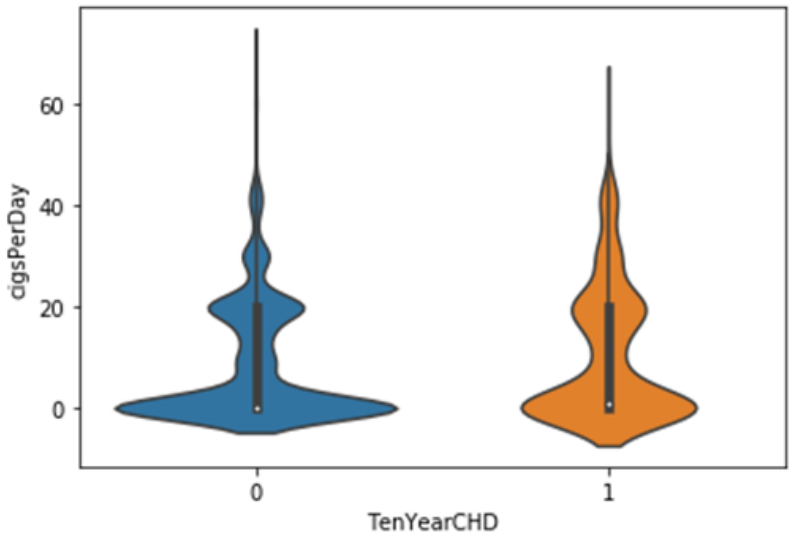

Figure 12 Violin Plot

Categorical Estimate Plots

Countplot

- A countplot basically counts the categories and returns a count of their occurrences. It is one of the most simple plots provided by the seaborn library

```
def plot_bar(data,numericCols,categoricalCols,response='TenYearCHD'):
    for i in numericCols:
                for j in categoricalCols:
                    #sns.jointplot(x=response, y=i, data=data)
                    sns.catplot(x=j, y=i, hue=response,   data=data, kind='bar')
                    #sns.kdeplot(data=data, x=i, hue=response, fill=True,
                                #common_norm=False, palette="crest", alpha=.5, linewidth=0,)
                    #sns.swarmplot(x=j, y=i, hue=response, data=data)
                    plt.show()
```

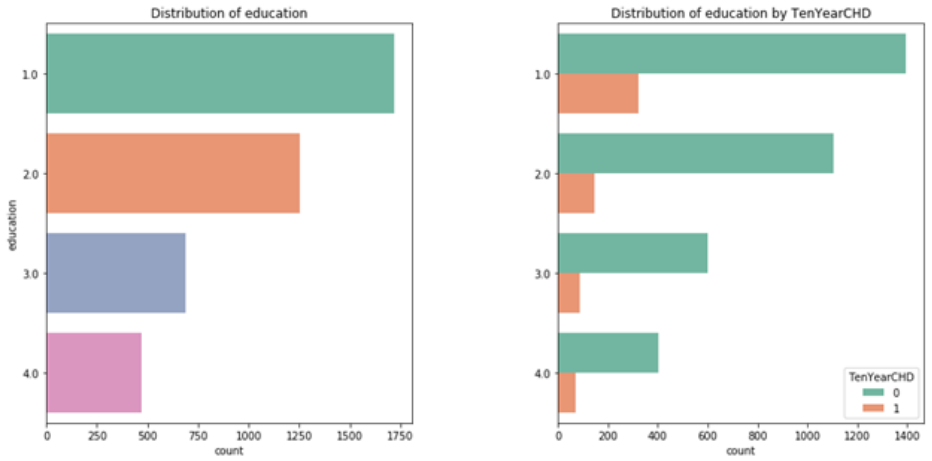

Figure 13 Count Plot

Barplot

- A **barplot** is basically used to aggregate the categorical data according to some methods and by default it's the mean.
- It can also be understood as a visualization of the group by action.
- To use this plot we choose a categorical column for the x axis and a numerical column for the y axis and we see that it creates a plot taking a mean per categorical column.

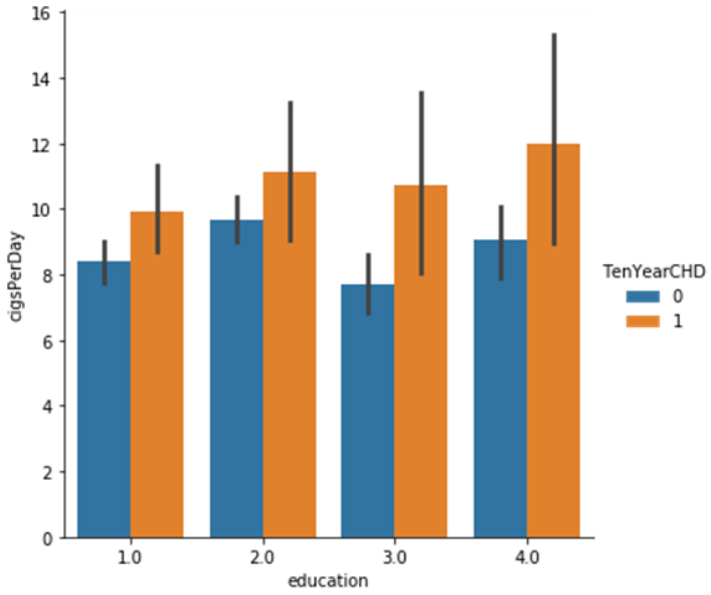

Figure 14 Bar Plot

Numerical Scatter plot

- A scatter plot is a useful visual representation of the relationship between two numerical variables (attributes) and is usually drawn before working out a linear correlation or fitting a regression line.
- The resulting pattern indicates the type (linear or nonlinear) and strength of the relationship between two variables.

```
def scatter_plot(data,numericCols,response='TenYearCHD'):
    for i in range(len(numericCols)):
                for j in range(len(numericCols)-1):
                    #sns.jointplot(x=response, y=i, data=data)
                    sns.scatterplot(data=data, x=numericCols[i], y=numericCols[j+1], palette="deep", legend="full",
                                    hue=data[response].tolist())
                    #ax.legend(title=response)  # add a title to the legend
                    plt.show()
```

```
scatter_plot(data,numerical)
```

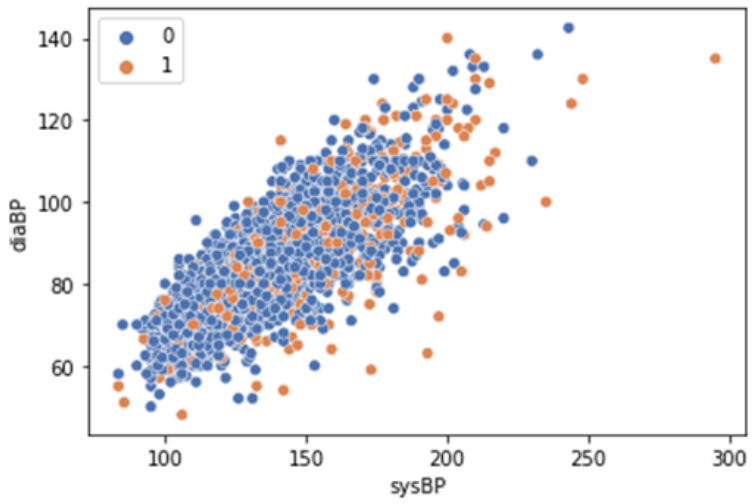

Figure 15 Scatter Plot

Pair Plot

- Pair plot visualizes pairwise relationships in a dataset.
- This function will create a grid of Axes such that each numeric variable in data will be shared across the y-axes across a single row and the x-axes across a single column.
- Pair plot is used to plot multiple pairwise bivariate distributions in a dataset, we can **use** the **pairplot**() function.

```
def snsPlot(data, response=['TenYearCHD']):
    try:

        d1 = data.copy()
        categoricalCols = d1.select_dtypes(include=["category"]).columns
        numericCols = d1.select_dtypes(include=["number"]).columns

        for i in numericCols:
            for j in categoricalCols:
                sns.catplot(x=j, y=i, hue=response,   data=data, kind='bar')

        for i in range(len(numericCols)):
            for j in range(len(numericCols)-1):
                sns.scatterplot(data=data, x=numericCols[i], y=numericCols[j+1], palette="deep", legend="full")
                plt.show()

        #TODO : Education is a category. Why does the graph not show this ?
        for i in categoricalCols:
            for j in [response]:
                plt.figure(figsize=(15, 7))
                plt.subplot(121)
                graph = sns.countplot(y=data[i],
                                      palette="Set2",
                                      order=data[i].value_counts().index[:100])
                plt.title("Distribution of " + i)

                plt.subplot(122)
                sns.countplot(y=data[i],
                              hue=data[j], palette="Set2",
                              order=data[i].value_counts().index[:100])
                plt.ylabel("")
```

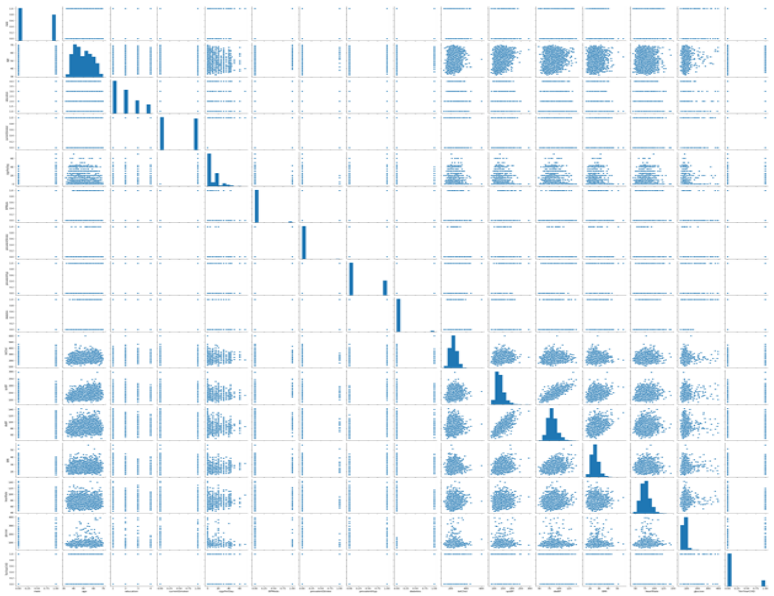

Figure 16 Pair Plot

Pandas Plot

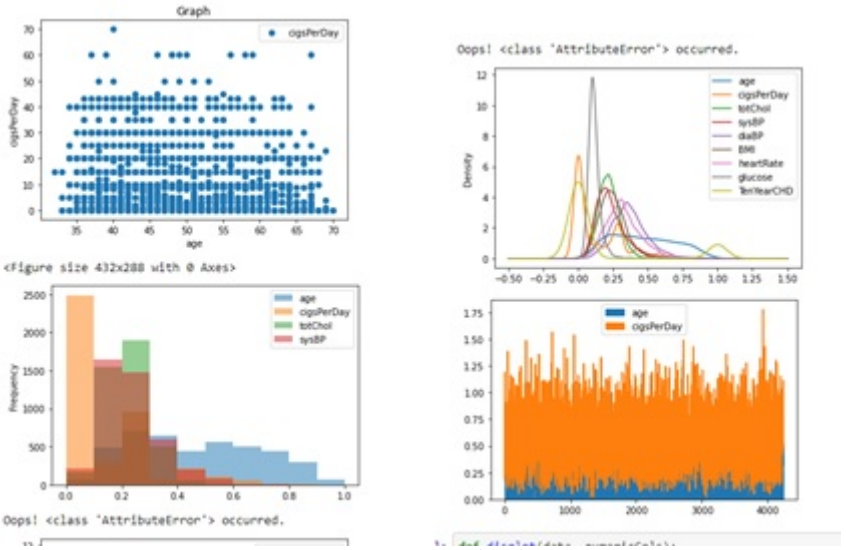

Figure 17 Panda Plot

Dist Plot

- A distplot plots a univariate distribution of observations.
- The **distplot()** function combines the matplotlib hist function with the seaborn **kdeplot()** and **rugplot()** functions.

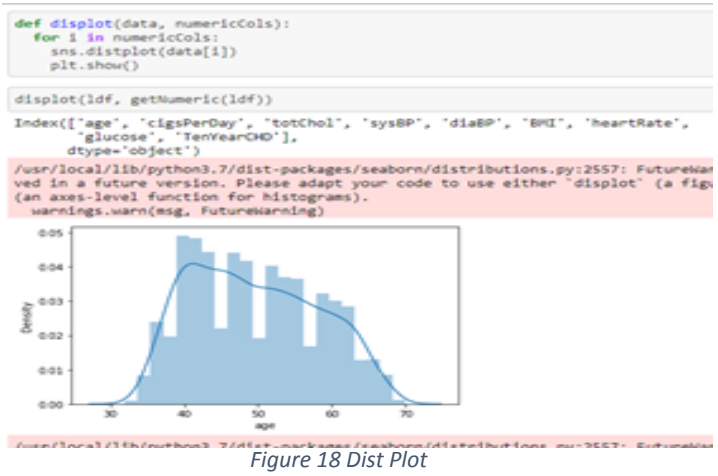

Figure 18 Dist Plot

Rug Plot

- Data distribution of a variable against the density distribution
- Overall distribution of continuous data can be concluded from rug plot

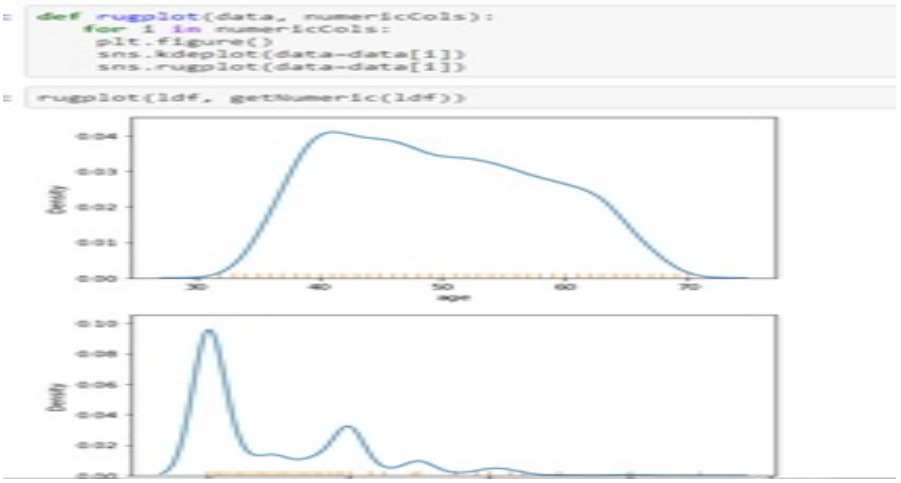

Figure 19 Rug Plot

Visualize Multiple Facets

- To visualize 3 dimensional data using collection of graphs
- Two Variables per graph and a third categorical variable
- Produces a visual representation of two dependent variables and their relationship with the third categorical variable

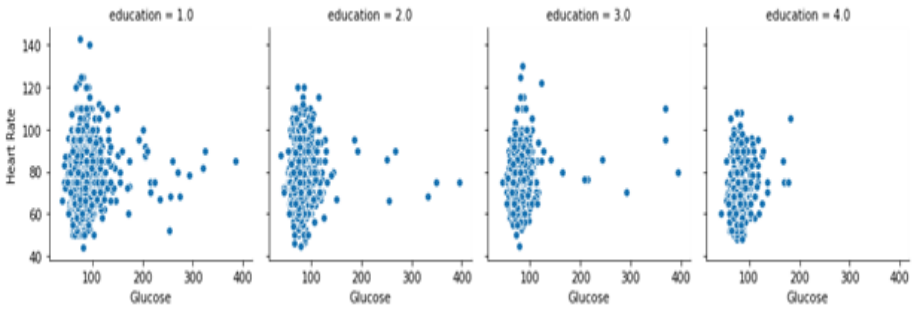

Figure 20 Multiple Facets

Observations

- There is a positive correlation between **sysBp** and **diaBp**.
- Low educated people are in take of cigperday is less compare with more educated and resulting more chances of getting heart diseases
- There is a relation between education and health parameters.In all the education plots, the lower educated people have health issues.In **sysBP** vs **education** , **diaBP** vs **education**,**education** vs **BMI** and **glucose** vs **education** plots
- In between **40-50** age people are smoking more Cigarettes per day
- Surprisingly there is very little impact of getting the CHD with current smoking habit
- Impact of male patients is more when compared with female.

Machine learning modelling and techniques applied

The classification modelling activity is dependent on the following 3 main methods

1. Hyper parameters finder method (hyperparam_finder) . This method performs a grid search on the parameters that correspond to each of the classifier algorithms. The parameters along with their values are passed a dictionary to this method. Using the classifier model object and the parameter dictionary, this method returns the best classifier algorithm parameter with respect to accuracy.
2. Run model method (run_model) . This method is a simple wrapper method to the fit method of the corresponding classifier algorithm. This method is responsible for performing the test + train split and fitting (X, y) values of the classifier.
3. Summarise results method (summarise_results) . This method is a helper method for the reporting and summarisation functions. The metrics corresponding to the classifier score, ROC curve and a heat map of the confusion matrix are reported via this function.

The general approach to model the classifier is split into two steps.

1. The first step is to pass the data to the hyperparam_finder along with the parameters dictionary of the corresponding classifier model object.
2. With the optimal algorithm parameters found in the hyperparam_finder method, the actual model building method run_model will commence. The results will also be reported in this method.

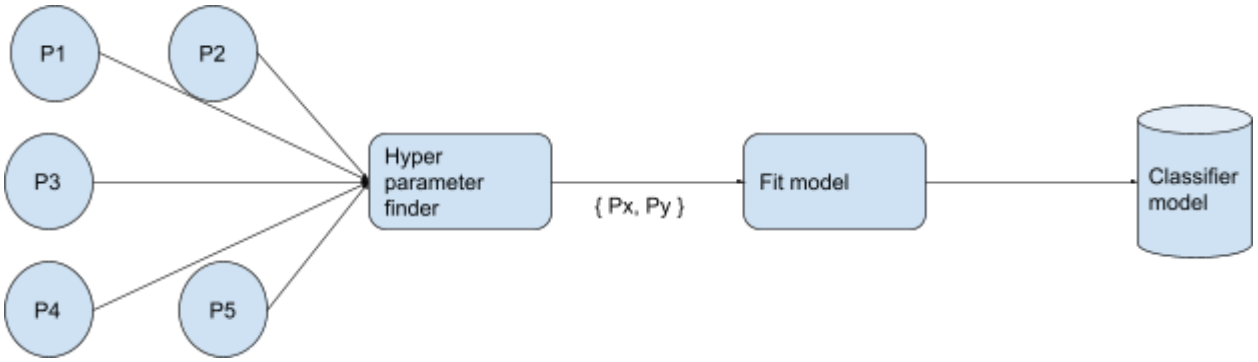

Figure 21 Modelling Flow

The classification models that were studied are

1. Logistic regression
2. SVM
3. Random forest
4. XGBoost
5. Gaussian naive bayes
6. Multilayer perceptron

Logistic regression

Logistic regression is a modelling technique by which the probability of each class is transformed into a logistic function. A logistic function is a sigmoid function that has a value between 0 and 1. The coefficients of the logistic regression algorithm are estimated from the training data. This estimation is done using maximum likelihood estimation.

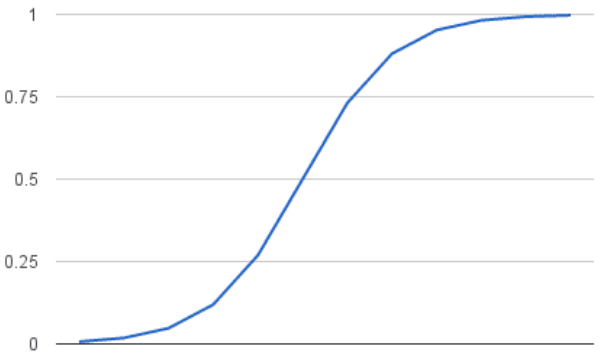

Figure 22 Logistic Regression

In the hyperparameter search step the logistic regression parameters that were evaluated are

Parameter	Values
solver	newton-cg, lbfgs, liblinear
penalty	none, l1, l2, elastic net
C	1e-5, 1e-4, 1e-3, 1e-1, 1, 10, 100
multi_class	auto, ovr, multinomial

Table 3 Logistic Regression

The hyperparameter search was carried out for 30 folds for each of the 288 candidate parameter combinations. The total number of search iterations was 8640. From the grid search the best performance was found to be

```
Best Score: 0.6696192696192697
Best Hyperparameters: {'C': 0.1, 'multi_class':
'auto', 'penalty': 'l2', 'solver': 'newton-cg'}
```

Figure 23 Logistic Regression - Best Hyperparameter

Using this parameter, the ROC curve and confusion matrix obtained from the training are presented below.

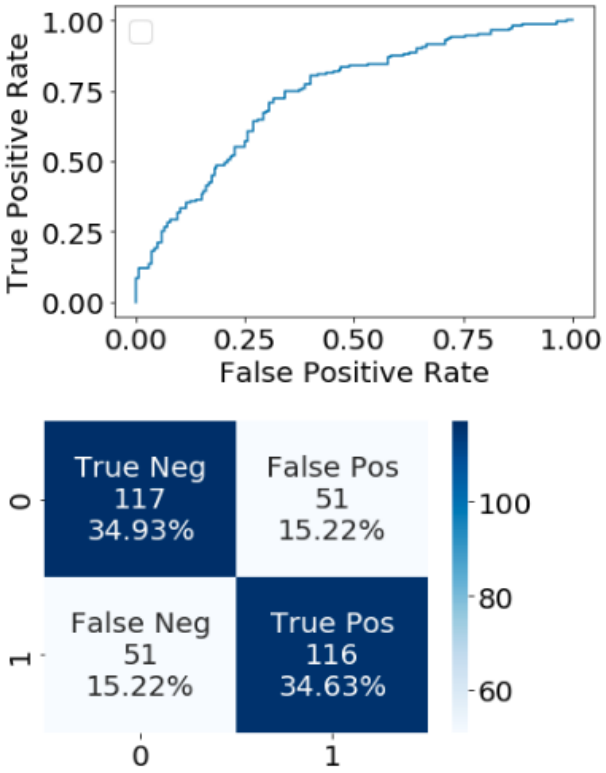

Figure 24 Logistic Regression - Evaluation

The precision, recall and F1 score from the model are as mentioned below

Score=0.696

	precision	recall	f1-score	support
Yes	0.70	0.70	0.70	168
No	0.69	0.69	0.69	167
accuracy			0.70	335
macro avg	0.70	0.70	0.70	335
weighted avg	0.70	0.70	0.70	335

ROC AUC=0.739

Figure 25 Logistic Regression - Classification Report

SVM classifier

The SVM algorithm finds a hyperplane decision boundary that best splits the training data into two classes. The SVM is a kernel based method where the algorithm could represent the training data in a different dimension that best splits the data.

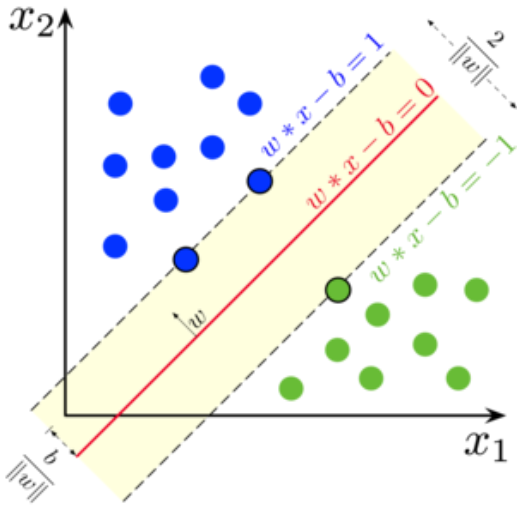

Figure 26 Soft Margin Classifier

In the hyperparameter search step the logistic regression parameters that were evaluated are

Parameter	Values
kernel	linear, poly, rbf, sigmoid
decision_fuction_shape	ovo, ovr
C	1e-5, 1e-4, 1e-3, 1e-1, 1, 10, 100

Table 4 SVC Hyperparameters

The hyperparameter search was carried out for 30 folds for each of the 64 candidate parameter combinations. The total number of search iterations was 1920. From the grid search the best performance was found to be

```
Best Score: 0.676012876012876
Best Hyperparameters: {'C': 1, 'decision_function_shape':
'ovo', 'kernel': 'linear'}
```

Figure 27 SVC Best Hyperparameter

Using this parameter, the ROC curve and confusion matrix obtained from the training are presented below.

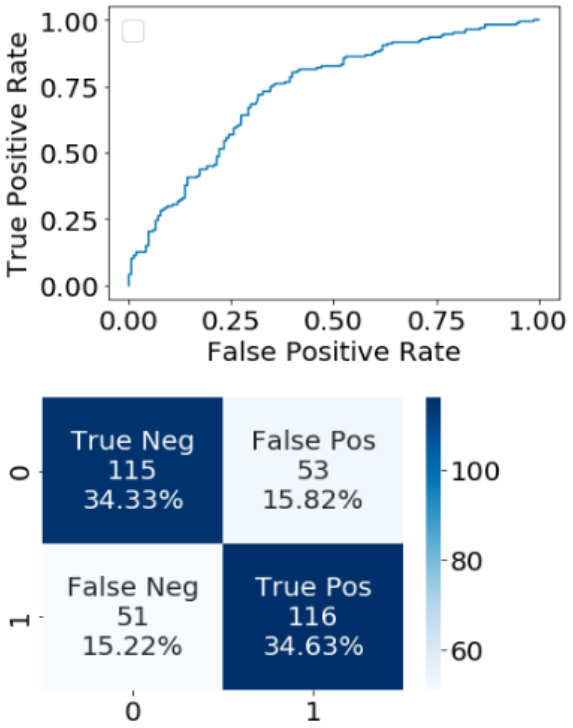

Figure 28 SVC Evaluation

The precision, recall and F1 score from the model are as mentioned below

Score=0.690

	precision	recall	f1-score	support
Yes	0.69	0.68	0.69	168
No	0.69	0.69	0.69	167
accuracy			0.69	335
macro avg	0.69	0.69	0.69	335
weighted avg	0.69	0.69	0.69	335

ROC AUC=0.734

Figure 29 SVC Classification Report

Random forest classifier

The random forest classifier is a meta classifier that is an ensemble of multiple decision trees. The decision trees are able to perform a classification operation using predicting modelling of the outcome based on the training data.

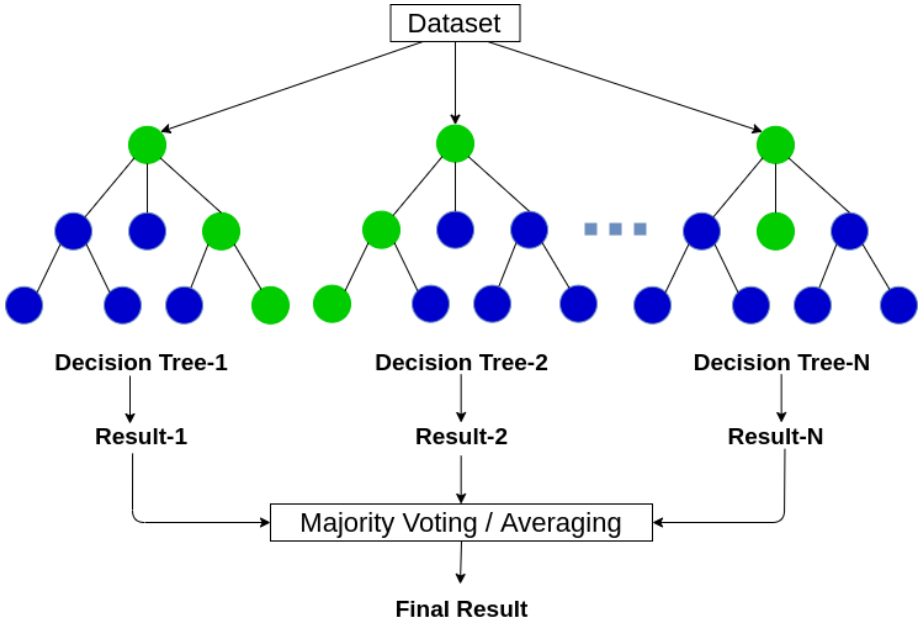

Figure 30 Random forest

In the hyperparameter search step the logistic regression parameters that were evaluated are

Parameter	Values
n_estimators	25, 50, 75, 100, 125
max_samples	0.1, 0.2, 0.3, 0.4, 0.5
max_features	1, 2, 3, 4, 5, 6

Table 5 Random Forest - Hyperparameters

The hyperparameter search was carried out for 30 folds for each of the 150 candidate parameter combinations. The total number of search iterations was 4500. From the grid search the best performance was found to be

```
Best Score: 0.6610556110556111
Best Hyperparameters: {'max_features': 3, 'max_samples':
0.2, 'n_estimators': 125}
```

Figure 31 Random forest - Best Hyperparameter

Using this parameter, the ROC curve and confusion matrix obtained from the training are presented below.

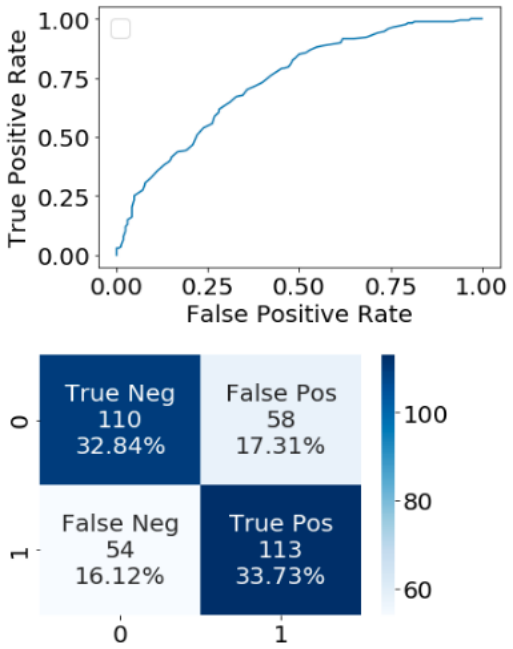

Figure 32 Random forest - Evaluation

The precision, recall and F1 score from the model are as mentioned below

Score=0.666

	precision	recall	f1-score	support
Yes	0.67	0.65	0.66	168
No	0.66	0.68	0.67	167
accuracy			0.67	335
macro avg	0.67	0.67	0.67	335
weighted avg	0.67	0.67	0.67	335

ROC AUC=0.737

Figure 33 Random Forest - Classification Report

XGBoost classifier

The XGBoost classifier is a recent algorithm that has been very useful in tabular data scenarios as in another improvement on decision trees technique. It stands for extreme gradient boosted random forest classifier. We used the vanilla option of the classifier with default options and did not use the hyperparameter search method. The precision, recall and F1 score from the model are as mentioned below

Score=0.651

	precision	recall	f1-score	support
Yes	0.66	0.64	0.65	168
No	0.65	0.66	0.65	167
accuracy			0.65	335
macro avg	0.65	0.65	0.65	335
weighted avg	0.65	0.65	0.65	335

ROC AUC=0.704

Figure 34 XGBoost - Classification Report

The ROC curve and confusion matrix obtained from the training are presented below.

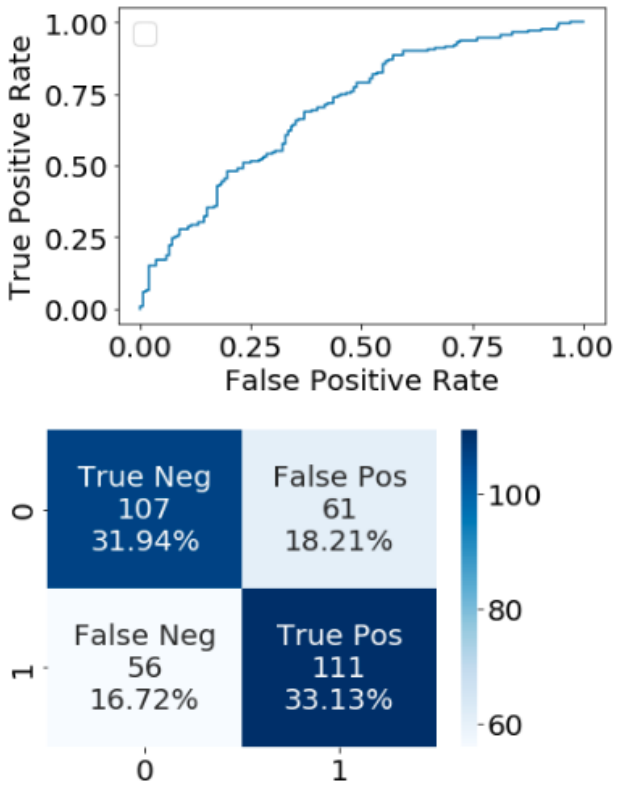

Figure 35 XGBoost - Evaluation

Gaussian naive bayes classifier

The Naive Bayes is a classification algorithm for binary (two-class) and multi-class classification problems. This method assumes that the attributes do not interact with each other and the classifier is modelled based on the probabilities of the variables on the data set. We used the vanilla option of the classifier with default options and did not use the hyperparameter search method.

The precision, recall and F1 score from the model are as mentioned below

Score=0.549

	precision	recall	f1-score	support
Yes	0.53	0.96	0.68	168
No	0.77	0.14	0.23	167
accuracy			0.55	335
macro avg	0.65	0.55	0.46	335
weighted avg	0.65	0.55	0.46	335

ROC AUC=0.747

Figure 36 Gaussian Naive Bayes - Classification Report

The ROC curve and confusion matrix obtained from the training are presented below.

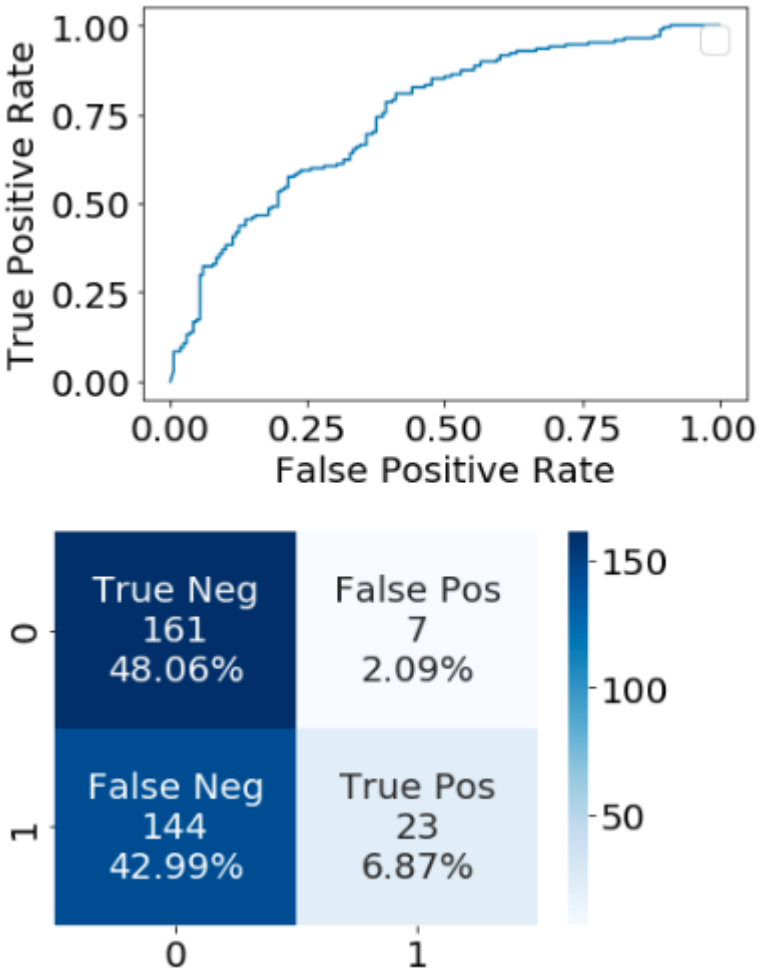

Figure 37 Gaussian Naive Bayes - Evaluation

Interpretation

Justification of Measures / Metrics used

Classification Accuracy:

Classification Accuracy is what we usually mean, when we use the term accuracy. It is the ratio of the number of correct predictions to the total number of input samples.It works well only if there are an equal number of samples belonging to each class.

We can conclude based on the results , SVM classifier returned with best score of 0.676012876012876 in comparison to logistic regression which returned with its score of 0.6696192696192697 , and Random forest with score of 0.6610556110556111

Confusion Matrix:

Confusion Matrix as the name suggests gives us a matrix as output and describes the complete performance of the model. We have a binary classification problem and have some samples belonging to two classes : YES or NO. Also, we have our own classifier which predicts a class. There are 4 important terms:

True Positives: The cases in which we predicted YES and the actual output was also YES. **True Negatives**: The cases in which we predicted NO and the actual output was NO. **False Positives**: The cases in which we predicted YES and the actual output was NO. **False Negatives**: The cases in which we predicted NO and the actual output was YES.

We can conclude that our heart disease prediction based on the classifier, highest rate of True Positive are found using :

SVM classifier with rate as 34.63%,

Random Forest classifier as 33.73%

XBoost Classifier as 33.12%

Similarly, the highest rate of True negative are found using:

Logistic regression with rate as 34.93%

Gaussian naive bayes classifier with rate as 48.06%

Confusion Matrix forms the basis for the other types of metrics.

Area Under Curve:

Area Under Curve(AUC) is one of the most widely used metrics for evaluation. It is used for binary classification problem. *AUC* of a classifier is equal to the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative example. Before defining *AUC*, let us understand two basic terms :

- **True Positive Rate (Sensitivity)** : True Positive Rate is defined as *TP/ (FN+TP)*. True Positive Rate corresponds to the proportion of positive data points that are correctly considered as positive, with respect to all positive data points.

- **True Negative Rate (Specificity)** : True Negative Rate is defined as *TN / (FP+TN)*. False Positive Rate corresponds to the proportion of negative data points that are correctly considered as negative, with respect to all negative data points.

- **False Positive Rate** : False Positive Rate is defined as *FP / (FP+TN)*. False Positive Rate corresponds to the proportion of negative data points that are mistakenly considered as positive, with respect to all negative data points.

We can conclude that the best ROC AUC are found with *Gaussian naive bayes with 0.747 and rest are*,

Logistic regression with 0.739, SVM with 0.734, Random Forest with 0.737,XBoost Classifier with 0.704

False Positive Rate and *True Positive Rate* both have values in the range **[0, 1]**. *FPR* and *TPR* both are computed at varying threshold values such as (0.00, 0.02, 0.04, …., 1.00) and a graph is drawn. *AUC* is the area under the curve of plot *False Positive Rate vs True Positive Rate* at different points in **[0, 1]**. **F1 Score:**

We can conclude that the best score found with Logistic regression as 0.696, and rest classifiers scored as SVM with 0.690, Random Forest with 0.666, XBoost Classifier with 0.651, Gaussian naive bayes with 0.549.

F1 Score is used to measure a test's accuracy.

F1 Score is the Harmonic Mean between precision and recall. The range for F1 Score is [0, 1]. It tells you how precise your classifier is (how many instances it classifies correctly), as well as how robust it is (it does not miss a significant number of instances).

High precision but lower recall, gives you an extremely accurate, but it then misses a large number of instances that are difficult to classify. The greater the F1 Score, the better is the performance of our model. Mathematically, it can be expressed as:

F1 Score tries to find the balance between precision and recall.

- **Precision:** It is the number of correct positive results divided by the number of positive results predicted by the classifier.

- **Recall:** It is the number of correct positive results divided by the number of ***all*** relevant samples (all samples that should have been identified as positive).

Project output in terms of above measures/metrics:

Best Metrics of 7 models developed are mentioned in the table shown below

Table 6 - Models

SI	Name of the Model	Accuracy (%)	AUC (%)	Recall (0/1)	Precision (0/1)	F1-score (0/1)	
1	Logistic Regression	0.70	0.739	0.70	0.70	0.70	
2	Gaussian Naïve Bayes	0.55	0.747	0.64	0.65	0.65	
3	SVM	0.69	0.69	0.68	0.69	0.69	
4	Random Forest	0.67	0.737	0.65	0.67	0.66	
5	XG Boost	0.65	0.704	0.64	0.66	0.65	

Future Work & Extension or Scope of improvements

- Our research suggests that applying a machine learning approach to a larger feature set as well as novel approaches to model diversity and model blending can improve on simpler readmission models such as LACE, potentially improving patient outcomes and lowering inpatient cost to hospitals.
- The LACE index identifies patients that are at risk for readmission or death within thirty days of discharge.
- The highest performing models were those developed around age groups rather than a general “all” age group
- This study targets diabetic patients only; however, we believe this early work sets the stage for further research to improve the accuracy of readmission risk for other top health conditions like heart disease, COPD, etc.
- Additional discovery may exist in modelling by condition group name (circulatory, respiratory, diabetes) as a primary condition. Also, suggesting a “next step” in transitions of care (home health, SNF, rehab facility) for a patient’s optimal outcome may prove useful within healthcare.
- From technical point of view, we can improve models by removing outliers rows and train models and predict the readmission rate. This is not performed due to time constraints

Conclusions / Recommendations

- Through this project, we created a machine learning model that is able to predict the patients with diabetes with highest risk of being readmitted.
- The best model is XG boost classifier with optimized hyper parameters with balanced target dataset
- The best model is SVM Classifier with unbalanced target dataset

Bibliography / References

https://www.ripublication.com/acst17/acstv10n7_13.pdf

https://thesai.org/Downloads/Volume10No12/Paper_36-Cardiovascular_Disease_Diagnosis.pdf

https://towardsdatascience.com/diagnostic-for-heart-disease-with-machine-learning-81b064a3c1dd

https://www.nature.com/articles/s41598-020-72685-1

Appendix-1

The detailed references to the code and techniques used are available in the Project Code submission folder

- **Heart Disease Prediction.ipynb** is our main python notebook file which has the all the feature engineering, scaling, encoding, visualization, PCA, modelling, Classifier, Accuracy calculations are performed on original data set `heart_disease.csv` and generates the required accurate and predictive results.

- **Deployment** done using Streamlite from the command prompt from c:/path using below syntax-

  Streamlit run heart_disease_prediction.py

  Note - Download the program in .py extension and save it in c:/ path

The detailed references to the additional files and code and techniques used are available in the Project Code submission folder

PCAM ZC321 Capstone Project – Heart Disease Prediction

Duly Completed Checklist

a) Is the Cover page in proper format? Y

b) Is the Title page in proper format? Y

c) Is the Certificate from the Mentor in proper format? Has it been signed? Y

d) Is Abstract included in the Report? Is it properly written? Y

e) Does the Table of Contents page include chapter page numbers? Y

f) Does the Report contain a summary of the literature survey? Y
- i. Are the Pages numbered properly? Y
- ii. Are the Figures numbered properly? Y
- iii. Are the Tables numbered properly? Y
- iv. Are the Captions for the Figures and Tables proper? Y
- v. Are the Appendices numbered? Y

g) Does the Report have Conclusion / Recommendations of the work? Y

h) Are References/Bibliography given in the Report? Y

i) Have the References been cited in the Report? N/A

j) Is the citation of References / Bibliography in proper format? Y