Kingdom of Saudi Arabia
Ministry of Education
Princess Nourah bint Abdulrahman University
Faculty of Computer and Information Sciences

Academic Year: 2021/2022
2nd Semester
Module: DCS 873-DIS 826
Program: MSc in computing
(Data Analytics)

Coronary Heart Disease Prediction

It is a project that combines the basic concepts of Data Management and Visualisation and Machine learning.

Maram Alqahtani – Hissah AlKanhal
Supervised by: Prof. Souham Meshoul, Dr. Romana Aziz

Table of Contents

Table of figures

Introduction

Coronary heart disease, also called coronary artery disease, is a chronic (long-lasting) disease and affects the blood vessels that supply blood to your heart. Coronary heart disease is the most common cause of death in Australia and the US, and although it cannot be cured, there are treatments that can reduce the risk of future heart problems and improve the symptoms. There are several risk factors that can increase the risk of developing CHD. These include high blood pressure, high cholesterol, diabetes, smoking, being overweight, and not doing enough physical activity. There are also risk factors that cannot control, including family history, and age.

Coronary heart disease (CHD) can be prevented by reducing or eliminating risk factors. Moreover, early diagnosis of CHD allows for the prevention of the worsening of CHD and its complications. Over the past several years, approaches that include machine learning (ML) have been making a significant impact on detecting and diagnosing diseases.

In general, the ML approach involves 'training' an algorithm with a control dataset for which the disease status (disease or no disease) is known and then applying this trained algorithm to a variable dataset in order to predict the disease status in patients for whom it is not yet determined. As larger data cohorts are introduced, the ML algorithm will be better trained as a predictor for disease status. More accurate disease prediction with ML would empower clinicians with improved detection, diagnosis, classification, risk stratification, and ultimately, management of patients, all while potentially minimizing required clinical intervention.

In this study, we will use the Cross-Industry Standard Process for Data Mining (CRISP-DM) to analyze, visualize, and formulate the problem as a learning problem. CRISP-DM is a process model with six phases that describes the data science life cycle naturally. It is like a set of guardrails to help plan, organize, and implement the data science (or machine learning) project.

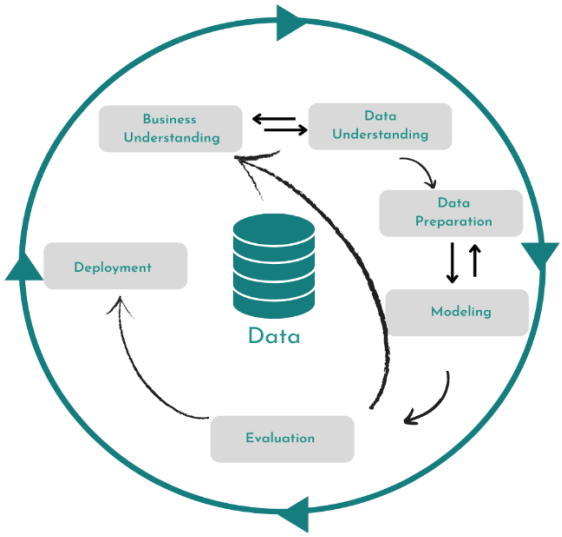


Figure 1: CRISP-dm process model

The study aims to predict whether a person has CHD or Not based on a set of health conditions and parameters related to them. Therefore, this project will follow the process illustrated below. More specifically, we will look at the topics listed below. To work through these topics, we will use Python, a high-level, general-purpose, and very popular programming language. Python programming language is being used in web development, Machine Learning applications, and all cutting-edge technology in the Software Industry. For the purpose of this project, we will use libraries such as Pandas, NumPy Matplotlib, and seaborn for data analysis and Scikit-Learn for machine learning and modeling tasks.

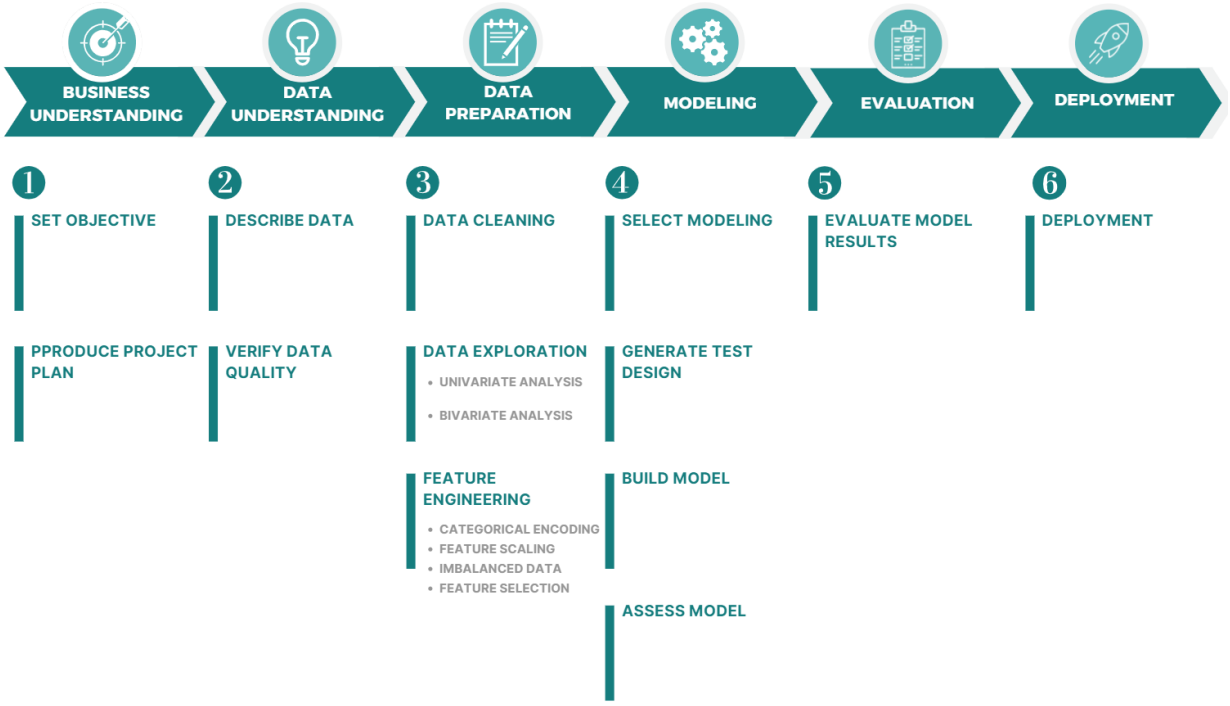


Figure 2: Project workflow

Chapter 1

Business Understanding

Business understanding is the initial phase of CRISP-DM. This phase focus on understanding the goal and the requirements of the project. This understanding will then be transformed into a machine learning problem to create a project plan for achieving the goals. This phase has two main tasks:

1. **Determine Business Objectives**

Determine business objectives can be done by understanding, from a business perspective, what the customer wants to accomplish and then defining business success criteria. For this project, our objective is to predict whether a patient has coronary heart disease or not based on his specific parameters such as systolic blood pressure, age, obesity, Etc.

2. **Project Plan**

The project plan is about selecting technologies and tools and defining detailed plans for each phase. For this project, it has six main phases; each phase will break down into sub-phases, and the project plan is as follows:

TASK NAME	ASSIGNED TO	START DATE	END DATE	Duration
Business understanding	Teamwork	27 March	30 March	4 days
Data understanding	Teamwork	31 March	6 April	7 days
Data Preparation	Teamwork	7 April	25 April	19 day
Modeling	Teamwork	26 April	8 May	14 day
Evolution	Teamwork	8 May	9 May	1 day
Deployment	Teamwork	10 May	11 May	2 days

Chapter 2

Data Understanding

The data understanding phase of CRISP-DM involves taking a closer look at the data available for analysis. This step is critical in avoiding unexpected problems during the next phase -data preparation- typically the longest part of a project. Data understanding involves accessing the data and exploring it using tables and graphics to determine the quality of the data and describe the results of these steps.

For this project, the dataset used is "Diseases" which targets people with a set of different health conditions and parameters; these parameters are used to predict whether or not they have coronary heart disease? It is a classic classification problem in machine learning in which we are trying to distinguish between one item or condition against another. We can only have one outcome; whether a person is dimmed to have coronary heart disease or not, the two options cannot occur together.

1. Describe data

The dataset contains information about the people. There are 462 observations and ten attributes in total. The dataset comprises ten features, nine descriptive features used to predict the target feature, and one target feature. Of the nine descriptive features, 8 are continuous features, while one feature, family history, is categorical. All features are discussed in the table below:

Feature	Feature type	Description
sbp	Continuous	Systolic Blood Pressure, measured in millimetres of mercury (mmHg)
tobacco	Continuous	Cumulative tobacco in kg
ldl	Continuous	Low Density Lipoprotein Cholesterol-measured in mmol/L
adiposity	Continuous	A condition of being severely overweight, or obese
famhist	Categorical	Family History whether there is a prevalence of heart disease in the family or not
typea	Continuous	Type of Behavior
obesity	Continuous	Measuring a person's body mass index (BMI)
alcohol	Continuous	Current Consumption of Alcohol
age	Continuous	Age of the patient
chd	Binary	Coronary Heart Disease, this is the target variable where 0 indicates "no disease" and 1 indicates "disease"

2. Verify Data Quality

Data quality is a measure of the condition of data based on factors such as accuracy, completeness, consistency, reliability, and whether it is up to date. Measuring data quality levels can help organizations identify data errors that need to be resolved and assess whether the data in their IT systems are fit to serve their intended purpose. For this project, we will discuss the data quality six dimensions in general and other issues.

1. *Accuracy* means that the data should reflect actual, real-world scenarios; the measure of accuracy can be confirmed with a verifiable source. For this dataset, and since we do not have the actual or the collected data, we will ensure that the values meet the real-world standards and measures. We summarize the accuracy of each feature as follows:

Feature	Range of values in dataset	Standards range in the real-world	Accurate or not? Why?
sbp	101 - 218 mm Hg	(120 – 180) mm Hg, It also can be lower or higher from this range, but this is the average range level	Yes, the dataset values met the real-world values range.
tobacco	0 - 31.2 kg	5.5 cigarettes per day which contain at least 6 gram of tobacco	Yes, considering it is cumulative tobacco
ldl	0.98 - 15.33 mmol/L	Less than 5 mmol/L	Yes, the dataset values met the real-world standards.
adiposity	6.74 – 42.49	(18.5 – 25), it can be less or higher from this range, which indicates a too thin person or too fat	Yes, the dataset values met the real-world standards.
typea	13 - 78		
obesity	14.7 - 46.58	(18.5 – 25), it can be less or higher from this range, which indicates a too thin person or too fat	Yes, the dataset values met the real-world standards.
alcohol	0 – 147.29	Less than 50 mg/dL, this is the normal level of alcohol consumption	Yes
age	15 - 64		Yes

2. *Completeness* measures the data's ability to deliver all the required available values effectively. For our dataset the table below summarize the missing values for each column:

Column Name	Count of Missing Values	Percentage of Missing Values
sbp	1	0.21645%
tobacco	0	0
ldl	0	0
adiposity	1	0.21645%
famhist	0	0
typea	0	0
obesity	0	0
alcohol	1	0.21645%
age	0	0
chd	0	0

The total number of missing values in the dataset is three, which indicates a tiny percentage considering the dataset size.

3. *Consistency* refers to the uniformity of data as it moves across networks and applications. The same data values stored in different locations should not conflict. For this dataset, because all continuous features fall in the same range, and the categorical feature has just two categories, we can state that our dataset is consistent.
4. *Timeliness*, Timely data is data that is available when it is required. Data may be updated in real-time to ensure that it is readily available and accessible.
5. *Validity*, Data should be collected according to defined business rules and parameters, conform to the proper format, and fall within the right range. For this dataset we generate box plot to make sure the numbers fall within the right range.

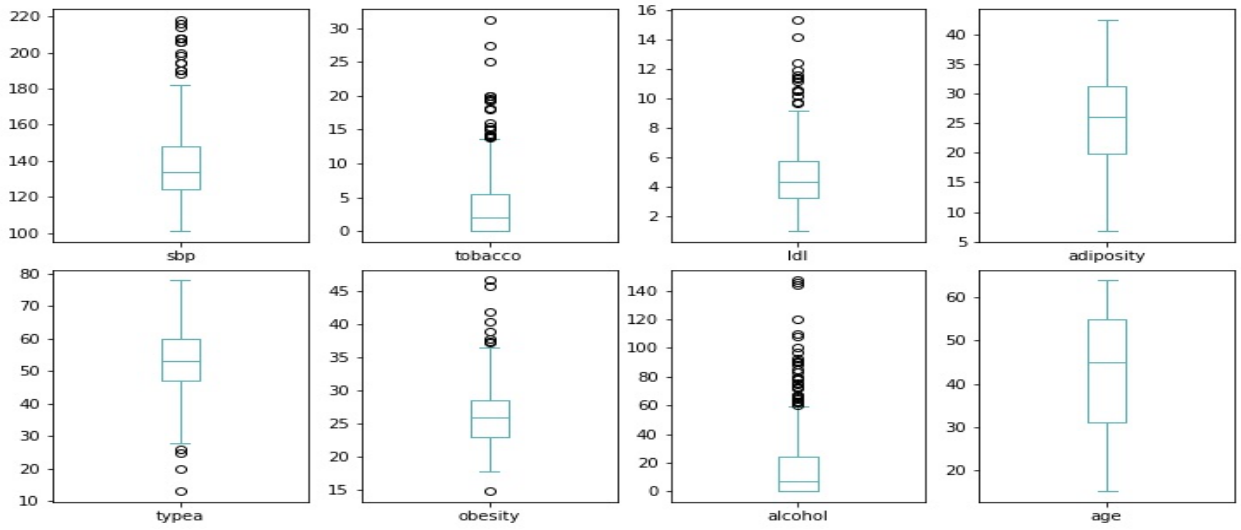


Figure 3: Dataset boxplot

For more details, we calculated the five-number summary, IQR and the upper and lower fences of the data set. Five-number summary simply consists of the smallest data value, the first quartile, the median, the third quartile. The interquartile range (IQR) measures the spread of the middle half of your data. It is the range for the middle 50% of your sample. Use the IQR to assess the variability where most of your values lie.

Interquartile range (IQR) formula: IQR= Q3–Q1

The upper and lower fences of the data set is the limits of a data set beyond which any scores should be treated as outliers.

The lower and upper inner fences calculated as following:

Lower inner fence= Q1–1.5(IQR)

Upper inner fence= Q3+1.5(IQR)

The lower and upper outer fences calculated as following:

Lower outer fence= Q1–3(IQR)

Upper outer fence= Q3+3(IQR)

Extreme outliers are data points that are more extreme than Q1–3(IQR) or Q3+3(IQR). Mild outliers are data points that are more extreme than Q1–1.5(IQR) or Q3+1.5(IQR). We summarize the statistics of each feature in the table below:

Feature	min	Q1	Q2	Q3	max	IQR	Lower inner fence	Upper inner fence	Lower outer fence	Upper outer fence	Number of outliers
sbp	101	124	134	148	218	24	88	184	52	220	15
tobacco	0	0.05	2	5.5	31.2	5.45	-8.1	13.6	-16.2	21.8	19
ldl	0.9	3.2	4.3	5.7	15.3	2.50	-0.47	9.5	-4.2	13.3	14
adiposity	6.7	19.8	26.1	31.2	42.4	11.4	2.69	48.4	-14.4	65.4	0
typea	13	47	53	60	78	13	27.5	79.5	8	99	4
obesity	14.7	22.9	25.8	28.4	46.5	5.5	14.7	36.7	6.4	45	9
alcohol	0	0.5	7.4	23.9	147.1	23.4	-34.68	59.16	-69.86	94.35	32
age	15	31	45	55	64	24	-5	91	-41	127	0

sbp: As shown in the box plot and the table above, 50% of observations between the interquartile range (148 - 124), 25% of the observations between (124-101), and 25% of the observations between (218-148). In addition, by applying the concept of lower and upper fences can be helpful to detects any outliers. Thus, as the results shows, any observations outside the range 88 and 184 will considered as Mild Outliers, and any observations outside the range 52 and 220 will considered as Extreme Outliers. After calculated outliers we found 15 outliers in the sbp feature.

tobacco: As shown in the box plot and the table above, 50% of observations between the interquartile range (5.5-0.05), 25% of the observations between (0.05-0), and 25% of the observations between (31.2-5.5). In addition, by applying the concept of lower and upper fences can be helpful to detects any outliers. Thus, as the results shows, any observations outside the range -8.1 and 13.6 will considered as Mild Outliers, and any observations outside the range -16 and 21.8 will considered as Extreme Outliers. After calculated outliers we found 19 outliers in the tobacco feature.
ldl: As shown in the box plot and the table above, 50% of observations between the interquartile range (5.7-3.2), 25% of the observations between (3.2-0.9), and 25% of the observations between (15.3-5.7). In addition, by applying the concept of lower and upper fences can be helpful to detects any outliers. Thus, as the results shows, any observations outside the range -0.47 and 9.5 will considered as Mild Outliers, and any observations outside the range -4.2 and 13.3 will considered as Extreme Outliers. After calculated outliers we found 14 outliers in the ldl feature.
adiposity: As shown in the box plot and the table above, 50% of observations between the interquartile range (31.2-19.8), 25% of the observations between (19.8-6.7), and 25% of the observations between (42.4-31.2). In addition, by applying the concept of lower and upper fences can be helpful to detects any outliers. Thus, as the results shows, any observations outside the range 2.69 and 48.4 will considered as Mild Outliers, and any observations outside the range -14.4 and 65.4 will considered as Extreme Outliers. After calculated outliers we found that the adiposity does not has any outliers.
typea: As shown in the box plot and the table above, 50% of observations between the interquartile range (60-47), 25% of the observations between (47-13), and 25% of the observations between (78-60). In addition, by applying the concept of lower and upper fences can be helpful to detects any outliers. Thus, as the results shows, any observations outside the range 27.5 and 79.5 will considered as Mild Outliers, and any observations outside the range 8 and 99 will considered as Extreme Outliers. After calculated outliers we found 4 outliers in the typea feature.
obesity: As shown in the box plot and the table above, 50% of observations between the interquartile range (28.4-22.9), 25% of the observations between (22.9-14.7), and 25% of the observations between (46.5-28.4). In addition, by applying the concept of lower and upper fences can be helpful to detects any outliers. Thus, as the results shows, any observations outside the range 14.7 and 36.7 will considered as Mild Outliers, and any observations outside the range 6.4 and 45 will considered as Extreme Outliers. After calculated outliers we found 9 outliers in the obesity feature.
alcohol: As shown in the box plot and the table above, 50% of observations between the interquartile range (23.9-0.5), 25% of the observations between (147.1-23.9), and 25% of the observations between (0.5-0). In addition, by applying the concept of lower and upper fences can be helpful to detects any outliers. Thus, as the results shows, any observations outside the range -34.68 and 59.16 will considered as Mild Outliers, and any observations

outside the range -69.86 and 94.35 will considered as Extreme Outliers. After calculated outliers we found 32 outliers in the alcohol feature. Which means alcohol feature has the highest number of outliers in the dataset.

age: As shown in the box plot and the table above, 50% of the ages between the interquartile range (55-31), 25% of the observations between (31-15), and 25% of the observations between (64-55). In addition, by applying the concept of lower and upper fences can be helpful to detects any outliers. Thus, as the results shows, any observations outside the range -5 and 91 will considered as Mild Outliers, and any observations outside the range -41 and 127 will considered as Extreme Outliers. After calculated outliers we found that the age does not has any outliers.

6. *Duplication* ensures no duplications or overlapping of values across all data sets. Data cleansing and deduplication can help remedy a low uniqueness score. For our dataset the table below summarize the duplicated values for all features:

Column Name	Count of Duplicated Values
sbp	0
tobacco	0
ldl	0
adiposity	0
famhist	0
typea	0
obesity	0
alcohol	0
age	0
chd	0

The total number of duplicated values in the dataset is zero, which means the majority of the values are completely unique.

Chapter 3

Data Preparation

Data preparation is the process of cleaning and transforming raw data prior to processing and analysis. It is an essential step before processing and often involves reformatting data, making corrections to data, and combining data sets to enrich it.

1. Data Cleaning

Data cleaning or cleansing is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data. For this project, we perform the following analysis for cleaning data:

1.1 Dealing with missing values

When dealing with large datasets missing values are a common occurrence. Strategies must be taken to determine how to deal with missing values in order for them not to impact our modeling negatively. Several strategies are usually employed to handle and tackle missing data, and this may include:

- Replacing or filling up the missing values: this is done by using the mean of the feature that contains the missing values, mode, or not treating them at all.
- Deleting missing values: missing values can be omitted from the dataset using various strategies. These include:
  - Listwise deletion, listwise deletion is preferred when there is a Missing Completely at Random case. In Listwise deletion entire rows (which hold the missing values) are deleted. It is also known as complete-case analysis as it removes all data that have one or more missing values.
  - Pairwise Deletion is used if missingness is missing completely at random i.e MCAR. Pairwise deletion is preferred to reduce the loss that happens in Listwise deletion. It is also called an available-case analysis as it removes only null observation, not the entire row.
  - Dropping complete columns, If a column holds a lot of missing values, say more than 80%, and the feature is not meaningful, that time we can drop the entire column.

In our case, and as shown in table X. The data contained only missing values in the sbp, adiposity, and alcohol. Since these features are very few to impact our analysis and machine learning model, we dropped the rows that contain the missing values.

1.2 Dealing with Irregular Data (Outliers)

Outliers are data values that are extremely high or low and differ significantly from most of the dataset, and they fall outside the interquartile range of the dataset. For purposes of our project, we used Interquartile Range (IQR) and Boxplot to visualize outliers. If not treated, outliers can have a negative adverse effect during the training of our machine learning model, leading to longer training durations and low accuracy, among other negative effects. There are several methods to handle outliers in a dataset, below some of these methods:

- Trimming/removing the outlier, in this technique, we remove the outliers from the dataset. Although it is not a good practice to follow.
- Mean/Median imputation, As the mean value is highly influenced by the outliers, it is advised to replace the outliers with the median value.
- Capping and Flooring, Capping is replacing all higher side values exceeding a certain theoretical maximum or upper inner fence by the upper inner fence, and Flooring is replacing all values falling below a certain theoretical minimum or lower inner fence by the lower inner fence value.

For our dataset, we will treat outliers with flooring and capping techniques. Therefore, we generated a boxplot to detect the outliers. As shown in figure 3 all features have outliers except age and adiposity. We also calculated the lower and upper inner fences for each feature (as shown in table below). Then for each feature, any feature values exceed the upper inner fence, we will replace them with the upper inner fence value, and if any values exceed the lower inner fence, we will replace them with the lower inner fence value. The table below lists all the feature with lower and upper fences and how to treat them:

Feature Name	Lower inner fence	Upper inner fence	Treatment
sbp	88	184	Since all the outliers are exceeded the upper inner fence, then we use capping to replace them with the value 184.
tobacco	-8.1	13.67	Since all the outliers are exceeded the upper inner fence, then we use capping to replace them with the value 13.67.
ldl	-0.47	9.5	Since all the outliers are exceeded the upper inner fence, then we use capping to replace them with the value 9.55.
type	27.5	79.5	Since all the outliers are exceeded the lower inner fence, then we use flooring to replace them with the value 27.5.
obesity	14.7	36.7	Since obesity have outliers are exceeded the upper and lower inner fences, we use capping and flooring to replace them with the value 36.7 and 14.7.
alcohol	-34.68	59.16	Since all the outliers are exceeded the upper inner fence, then we use capping to replace them with the value 59.16.

Therefore, we generated a second boxplot to display the data points distribution after treating outliers.

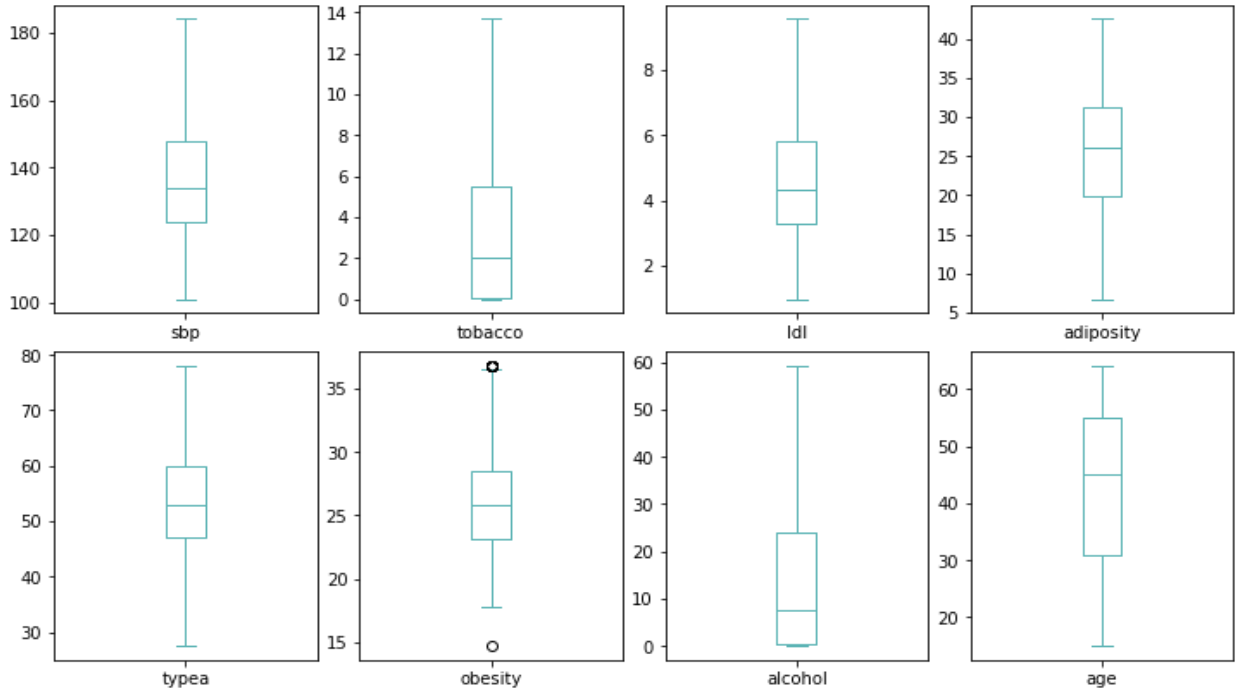


Figure 4: Dataset boxplot after treating outliers

Now, after we ensure that we have a cleaned dataset, we can move to the next task in the data preparation phase, which is Exploration Data Analysis (EDA), to gain more insights about the dataset.

2. Exploratory Data Analysis

Exploratory Data Analysis (EDA) refers to the critical process of performing initial investigations on data to discover patterns, spot anomalies, test hypotheses, and check assumptions with the help of summary statistics and graphical representations. To better understand the data and all features, we carried out exploratory data analysis, and below are the various analysis we undertook.

2.1 Univariate Analysis

Univariate analysis refers to the analysis of one variable. It does not deal with causal effects or relationships; instead, its primary purpose is to describe data and find any pattern within data. For this analysis, we carried out the below analysis: Summary Statistics, Frequency Distribution Tables, Frequency Polygons, Histogram, Bar Charts, and Pie Charts. Summary statistics help us understand the dataset better before undertaking any machine learning activities. The main summary statistics we will be interested in include the measures of dispersion of our dataset, which enables us to understand the degree of spread in the data we are using. Dispersion will tell us if the observations within the datasets fall far from each other or far from the mean value of the dataset for a highly dispersed dataset. For a dataset with low dispersion, the observations will tend to fall closer to the central value of the dataset. Summary statistics can also enable us to detect any anomalies present in our dataset. The graphical representations for different feature types help us look at the data distribution to analyze and find patterns. The following section will interpret each feature in the dataset.

- **SBP**

The tables below display the summary statistics for sbp feature:

Descriptive statistics		Quantile statistics	
Standard deviation	18.683835	**Minimum**	101
Variance	349.085702	**Q1 (25%)**	124
Skewness	0.720674	**Median (50%)**	134
Mean	137.793926	**Q3 (75%)**	148
Mode	134	**Maximum**	184
Sum	63523	**Range**	64

The count of sbp feature is 461, the average of the data is 137.79 with a standard deviation of 18.683, which indicates a large spread in the data around the mean considering the maximum value is 184 and the minimum value is 101. Moreover, it has a slight positive skewness of 0.654514 to ensure this point; we plotted a histogram to determine the shape of the data distribution.

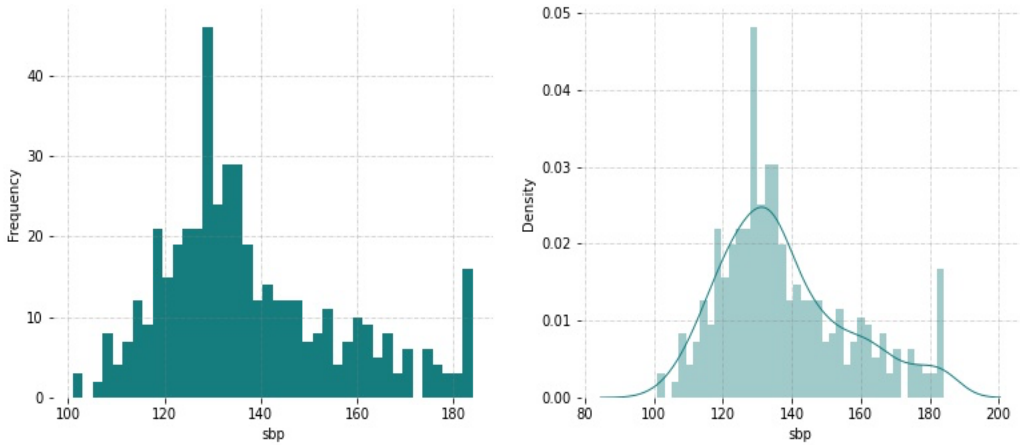


Figure 5: SBP Histogram

As shown in the figures above, systolic blood pressure(sbp) exhibits a right skewed or positive distribution. As can be seen from the histogram majority of the data falls to the right side of the histogram peak. It indicates that the data is not normally distributed.

- **Tobacco**

The tables below display the summary statistics for Tobacco feature:

Descriptive statistics		Quantile statistics	
Standard deviation	3.911996	**Minimum**	0
Variance	15.303716	**Q1 (25%)**	0.050000
Skewness	1.196790	**Median (50%)**	2
Mean	3.444585	**Q3 (75%)**	5.5
Mode	0	**Maximum**	13.671250
Sum	1587.93	**Range**	13.671250

The average of the data is 3.444585 with a standard deviation of 3.911996, which indicates a good spread in the data around the mean considering the maximum value is 13.67 and the minimum value is 0. Moreover, it has a slight positive skewness of 1.196790 to ensure this point; we plotted a histogram to determine the shape of the data distribution.

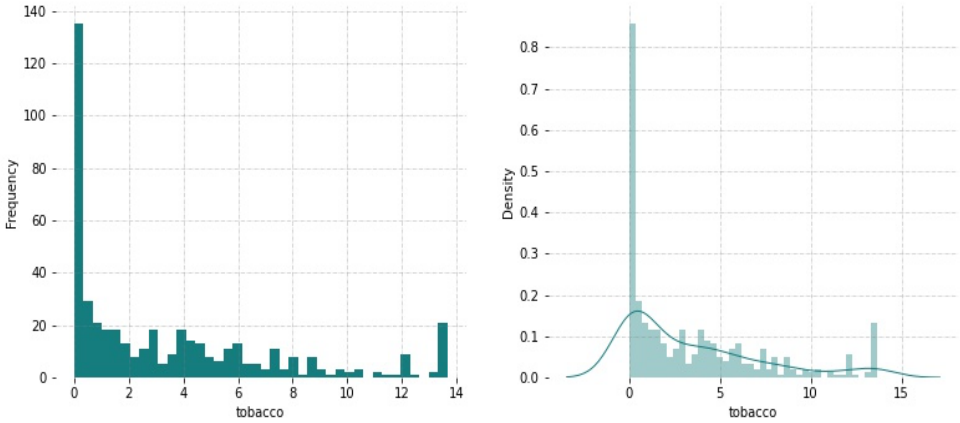


Figure 6: Tobacco Histogram

As shown in the figures above, tobacco intake exhibits a positive distribution with the data heavily skewed to the right. This trend indicates that majority of the datapoints in the tobacco variable fall towards zero or there is little to no tobacco present.

- **LDL**

The tables below display the summary statistics for ldl feature:

Descriptive statistics		Quantile statistics	
Standard deviation	1.878771	**Minimum**	0.98
Variance	3.029047	**Q1 (25%)**	3.29
Skewness	0.719637	**Median (50%)**	4.34
Mean	4.688671	**Q3 (75%)**	5.8
Mode	9.551250	**Maximum**	9.551250
Sum	2161.477500	**Range**	8.571250

The average of the data is 4.68 with a standard deviation of 1.87, which indicates a good spread in the data around the mean considering the maximum value is 9.55 and the minimum value is 0.98. Moreover, it has a slight positive skewness of 0.719637 to ensure this point; we plotted a histogram to determine the shape of the data distribution.

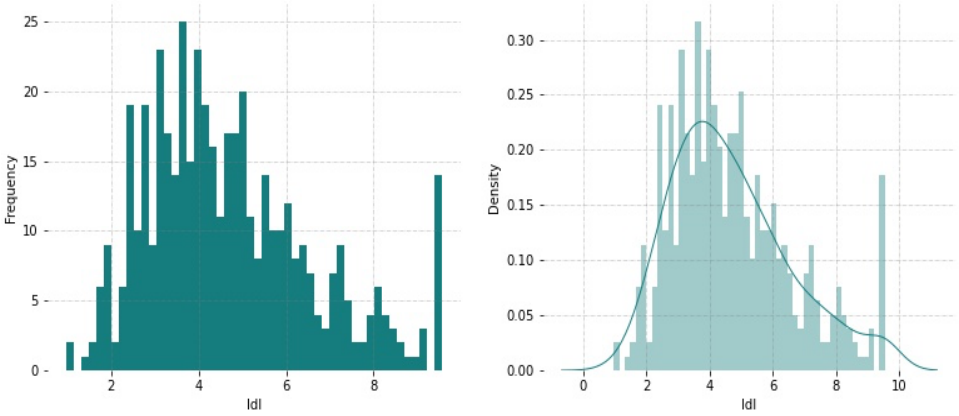


Figure 7: LDL Histogram

As shown in the figures above, low density lipoprotein cholesterol is also right skewed with majority of the data points falling to the right.

- **Adiplosity**

The tables below display the summary statistics for Adiposity feature:

Descriptive statistics		Quantile statistics	
Standard deviation	7.765163	**Minimum**	6.74
Variance	60.297760	**Q1 (25%)**	19.85
Skewness	-0.216972	**Median (50%)**	26.13
Mean	25.435119	**Q3 (75%)**	31.29
Mode	21.1	**Maximum**	42.49
Sum	11725.59	**Range**	35.75

The tables above shows the statistics of the adiposity feature. It gives us an approximate visualization of the data, as we can see the minimum value in the adiposity is 6.74 and the maximum is 42.49. The first quartile (Q1) = 19.85, the second quartile (Q2)= 26.13, the third quartile (Q3)= 31.29, range= 35.75, and Interquartile range (IQR) = 11.44. In addition, standard deviation = 7.765163 which indicates a good spread in the data. Moreover, it has a negative skewness of 0.216972 which means the data distribution has a skewness to the left.

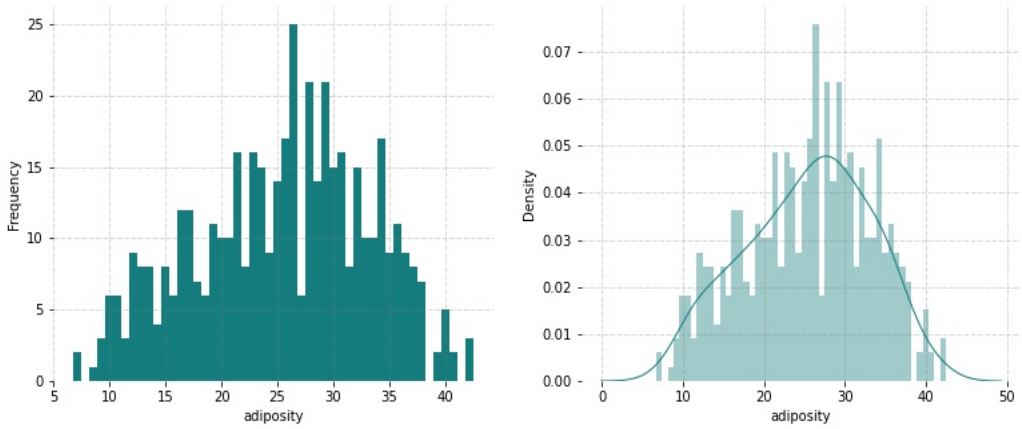


Figure 8: Adiposity Histogram

As shown in the figures above, adiposity is left skewed indicating that from the data we have high cases of people with too much fatty tissue in the body.

- **Famhist**

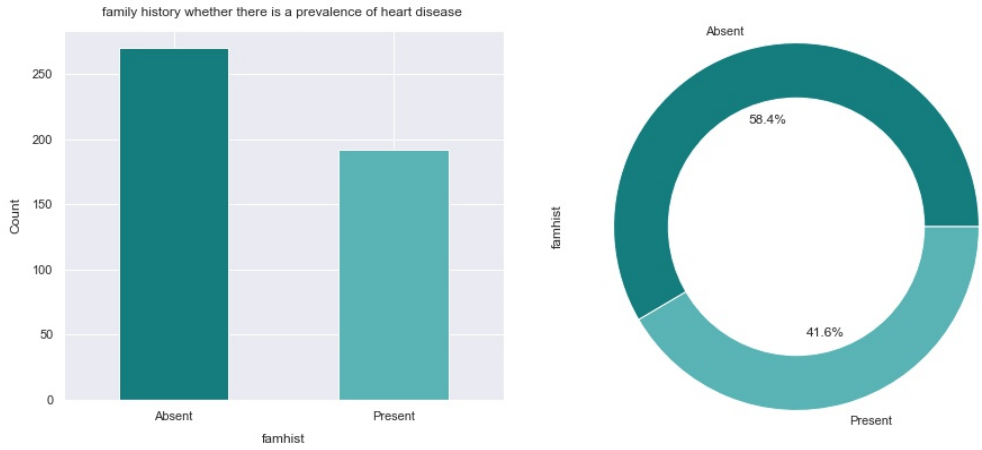


Figure 9: Famhist Histogram

The most common ways to display a categorical variable is bar charts and pie charts. As shown, the most people in the dataset and they are represent 58.4% dose not has prevalence of heart disease in their family history. On the another hand 41.6% has prevalence of heart disease in their family.

- **Type A**

The table below display the summary statistics for type A:

Descriptive statistics		Quantile statistics	
Standard deviation	9.599582	**Minimum**	27.5
Variance	92.151971	**Q1 (25%)**	47
Skewness	-0.210424	**Median (50%)**	53
Mean	53.114967	**Q3 (75%)**	60
Mode	52	**Maximum**	78
Sum	24486	**Range**	50.5

As shown in the tables above, the sum of all values is 24486. The center or the average of the data is 53. 114967 with a standard deviation of 9. 599582, which indicates a good spread in the data considering the maximum value is 78 and the minimum value is 27.5. Moreover, it has a negative skewness of 0.210424. Since the skewness is too small, we can say that the data distribution is approximately normal, with the most values accruing on the left. To ensure this point, we plotted a histogram to determine the shape of the data distribution.

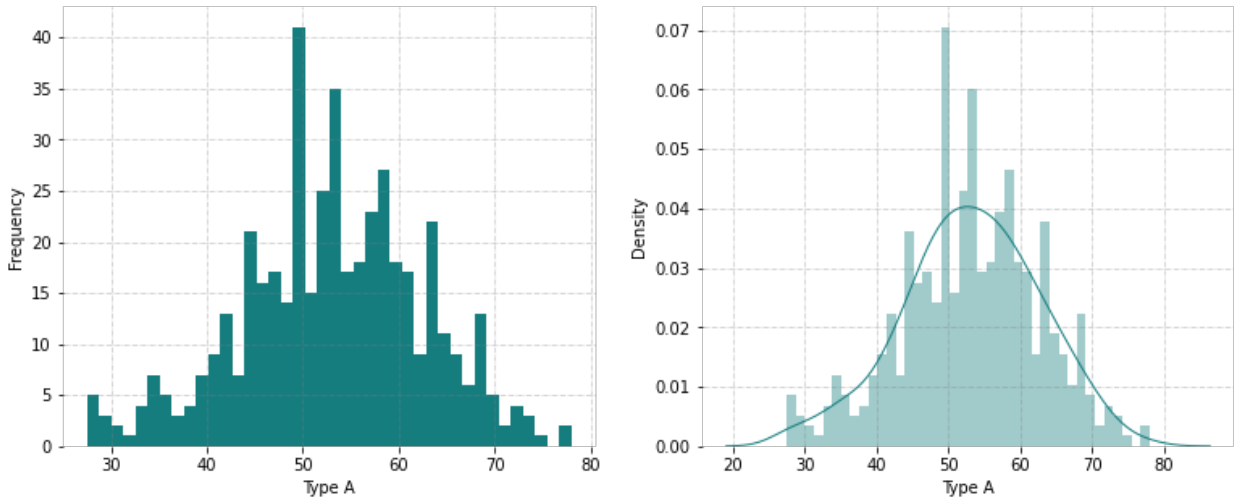


Figure 10: Type A Histogram

The type of behavior is approximately normal, as seen in the graph above.

- **Obesity**

The table below display the summary statistics for obesity:

Descriptive statistics		Quantile statistics	
Standard deviation	3.973589	**Minimum**	14.716250
Variance	15.789412	**Q1 (25%)**	23.09
Skewness	0.442454	**Median (50%)**	25.81
Mean	25.987085	**Q3 (75%)**	28.5
Mode	36.766250	**Maximum**	36.766250
Sum	11981.046250	**Range**	22.05

The average of the data is 25.987085 with a standard deviation of 3.973589, which indicates a small spread in the data around the mean considering the maximum value is 36.766250 and the minimum value is 14.716250.
Moreover, it has a slight positive skewness of 0.442454 to ensure this point; we plotted a histogram to determine the shape of the data distribution.

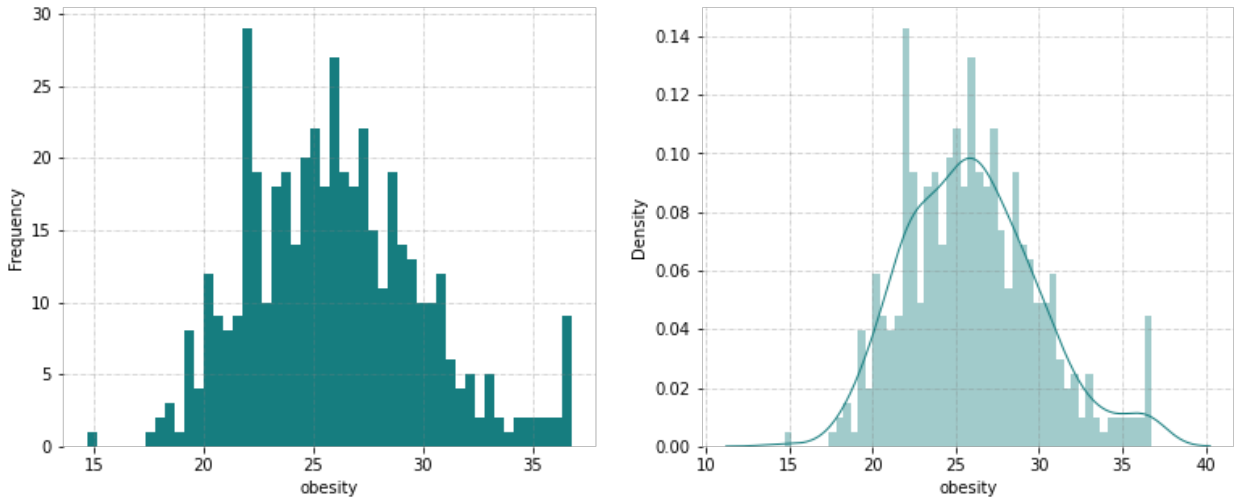


Figure 11: Obesity Histogram

Obesity tends to show an approximately normal distribution with slight positive skewness, as we can see most of the large values are located at the Centre and the right of the distribution. This indicates that most people are neither obese nor not too obese but fall within the middleweight destitution.

- **Alcohol**

The table below display the summary statistics for alcohol:

Descriptive statistics		Quantile statistics	
Standard deviation	18.517020	**Minimum**	0
Variance	342.880042	**Q1 (25%)**	0.51
Skewness	1.250455	**Median (50%)**	7.41
Mean	15.144165	**Q3 (75%)**	23.97
Mode	0	**Maximum**	59.16
Sum	6981.46	**Range**	59.16

The average of the data is 15.144165 with a standard deviation of 18.517020, which means that the data points have a large spread from the mean point. Moreover, it has a positive skewness of 1.250455 to ensure this point; we plotted a histogram to determine the shape of the data distribution.

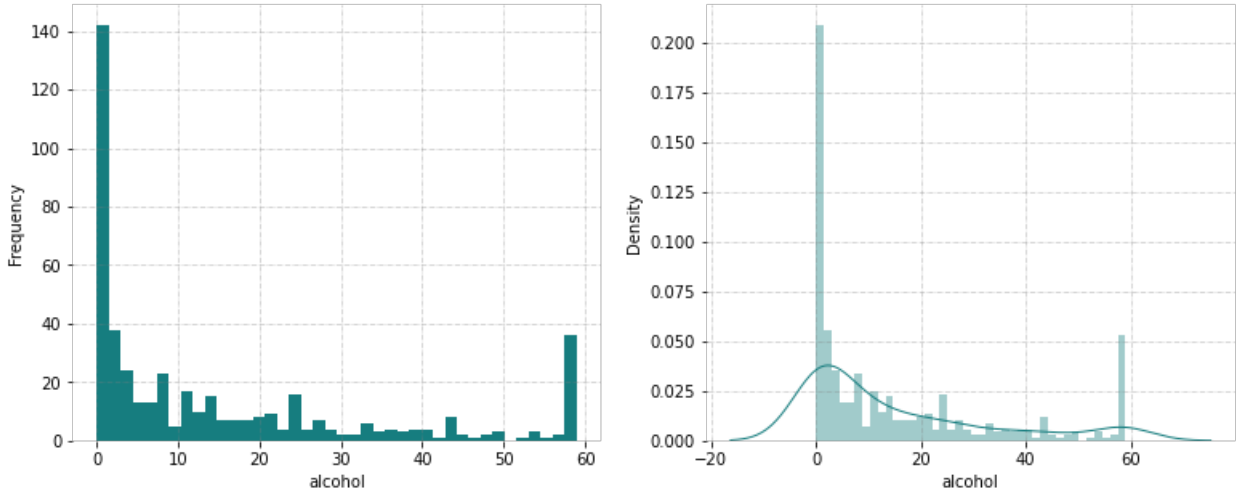


Figure 12: Alcohol Histogram

As shown, current alcohol consumption is heavily skewed to the right with most values falling to the right. This indicates that most people in the dataset had low alcohol consumption.

- **Age**

The table below display the summary statistics for alcohol:

Descriptive statistics		Quantile statistics	
Standard deviation	14.586001	**Minimum**	15
Variance	212.751438	**Q1 (25%)**	31
Skewness	-0.386244	**Median (50%)**	45
Mean	42.865510	**Q3 (75%)**	55
Mode	16	**Maximum**	64
Sum	19761	**Range**	49

The average value of data is 42.865510 with a standard deviation of 14.586001, which indicates a slightly large spread in the data considering the maximum value is 64 and the minimum value is 15. Moreover, it has a negative skewness of 0.386244 to ensure this point, we plotted a histogram to determine the shape of the data distribution.

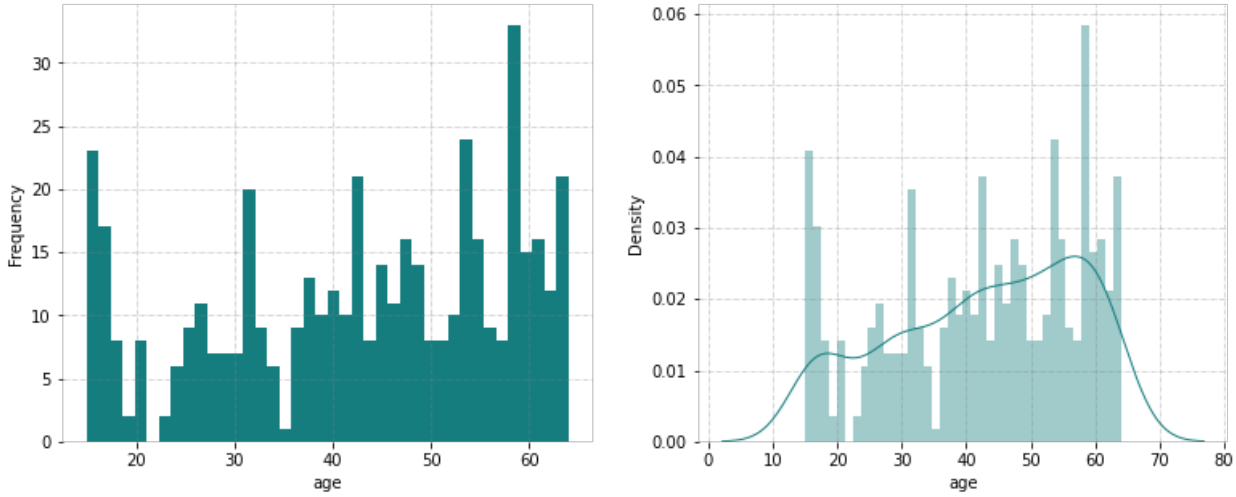


Figure 13: Age Histogram

As shown, Age exhibits Multi-Modal Distribution. This indicates that the dataset has great variations of age with no particular age group being dominantly represented in the dataset.

- **CHD**

CHD, coronary heart disease this is the target variable where 0 indicates "no disease" and 1 indicates "disease.

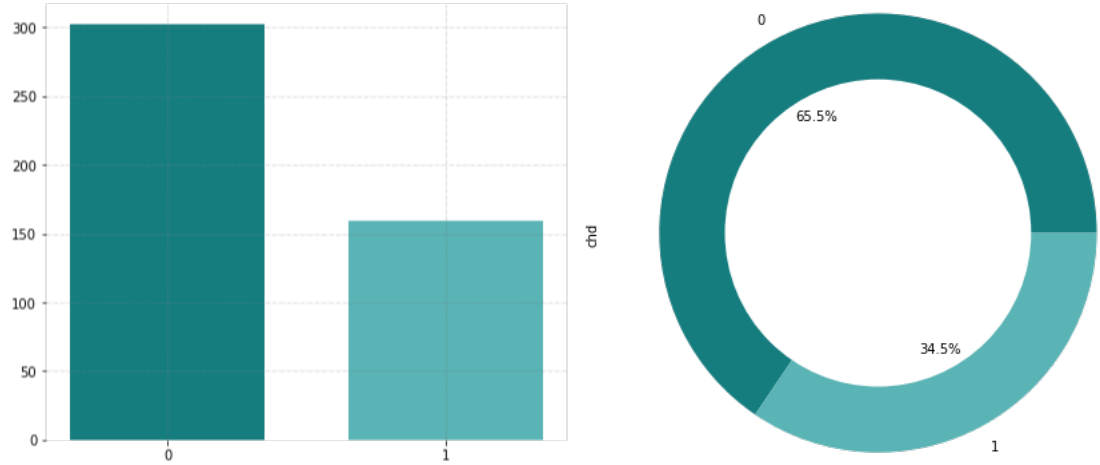


Figure 14: CHD Histogram

As shown, 65.4% of people in the sample have CHD, and 34.5% do not have CHD.

2.2 Bivariate Analysis

Bivariate analysis is one of the simplest quantitative (statistical) analysis forms. It involves the analysis of two variables (often denoted as X and Y) for the purpose of determining the empirical relationship between them. Bivariate analysis can be useful in testing the simple association between two variables. It can also help us define and predict the values of a dependent variable based on the changes occurring in the independent variable. For this project, we undertook the below bivariate analysis.

2.2.1 Impact of numerical features on target feature

Since our dataset contains eight numerical features from nine, it is essential to understand the data distribution for each one in the target feature levels. Therefore, we generated a violin plot that depicts distributions of numeric data for one or more groups using density curves.

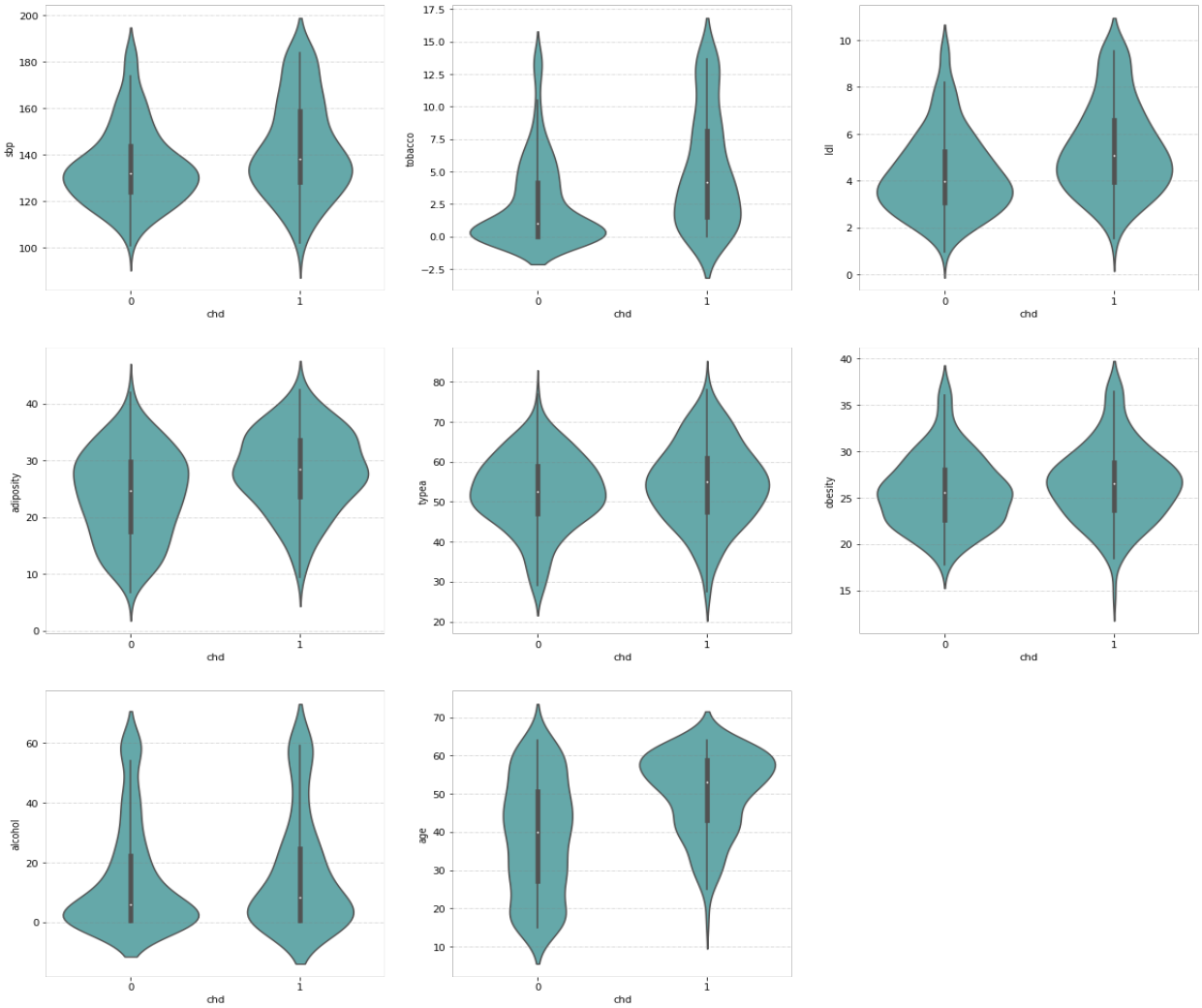


Figure 15: violin plot for all features

The average systolic blood pressure for people with no CHD is between 120 and 140 mmHg, representing the normal range levels for SBP, and for the people that have CHD is spread from 120 mmHg and increases to higher levels. For tobacco, the average tobacco consumption for people with no CHD is between 0 and 2, representing a low level of consumption, but people with CHD are spread to higher levels of consumption.
The average level of low-density lipoprotein cholesterol for people with no CHD is between 2 and 5 mmo/L, which typically falls in the normal range, but for the people that have CHD is spread from 4 mmo/L to higher levels. A high level of LDL is an important factor for having CHD. For adiposity, the average level of adiposity for people with no CHD is between 20 and 30, representing almost the normal levels or overweight, and for people with CHD is between 25 and 40, representing higher levels of BMI (Morbidly obese).

The average level of type A for people with no CHD is between 40 and 60, and for people with CHD, the level is spread to higher units. For obesity, the average level of obesity for people with no CHD is increased from 20 and decreased to 25, representing almost the normal levels, and for people with CHD, it is increased from 25 to 30, representing higher levels of BMI means that these people are overweight.
The average level of alcohol consumption for people with no CHD indicates an insignificant level, from 0 to 20, and approximately the same for people with CHD but with a bit of increase in consumption. People with no CHD have different ages, but for people with CHD, their ages are between 50 and 70 years, which means that age is a risk factor in causing CHD. Furthermore, the table below lists the average values for each feature in target feature levels:

Feature	Average for a person that does not has CHD	Average for a person that has CHD
sbp	135.175497	142.767296
tobacco	2.564627	5.115951
ldl	4.312438	5.403278
adiposity	23.969106	28.219623
typea	52.428808	54.418239
obesity	25.683262	26.564159
alcohol	14.297219	16.752830
age	38.854305	50.484277

2.2.3 The relationship between two variables

Bivariate associations are the associations between pairs of variables in a dataset. An association is any relationship between two variables that makes them dependent, i.e., knowing the value of one variable gives us some information about the possible values of the second variable. This task is essential in building high-performing machine learning algorithms and depends on identifying relationships between variables. This helps in feature engineering as well as for deciding on the machine learning algorithm. For this project, we performed the following analysis techniques.

A. Relationship between numerical features

Relationship Visualization

One widely used plot to present measurements of two or more related variables is Scatterplot. It is particularly useful when values of the variables of the y-axis are thought to be dependent on values of the variable of the x-axis. In a scatterplot, the data points are plotted but not joined. The resulting pattern indicates the type and strength of the relationship between two or more variables. The pattern of the data points on the Scatterplot reveals the relationship between the variables. Scatterplots can illustrate various patterns and relationships, such as linear or non-linear relationships, positive (direct) or negative (inverse) relationships, the concentration or spread of data points, and the presence of outliers. We will generate a scatterplot for each feature in our dataset to detect any possible relationship between these numerical features.

- **SBP**

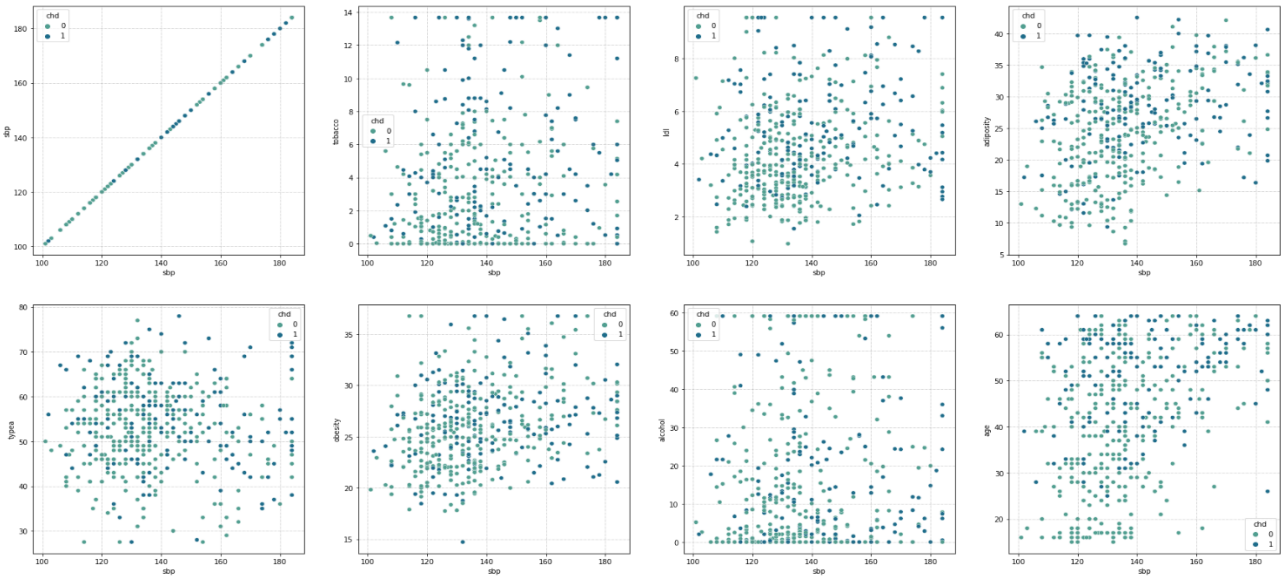


Figure 16: SBP scatterplot

It seems there is no linear relationship between sbp and other numerical features. Although, sbp and adiposity and obesity tend to have a positive relationship.

- **Tobacco**

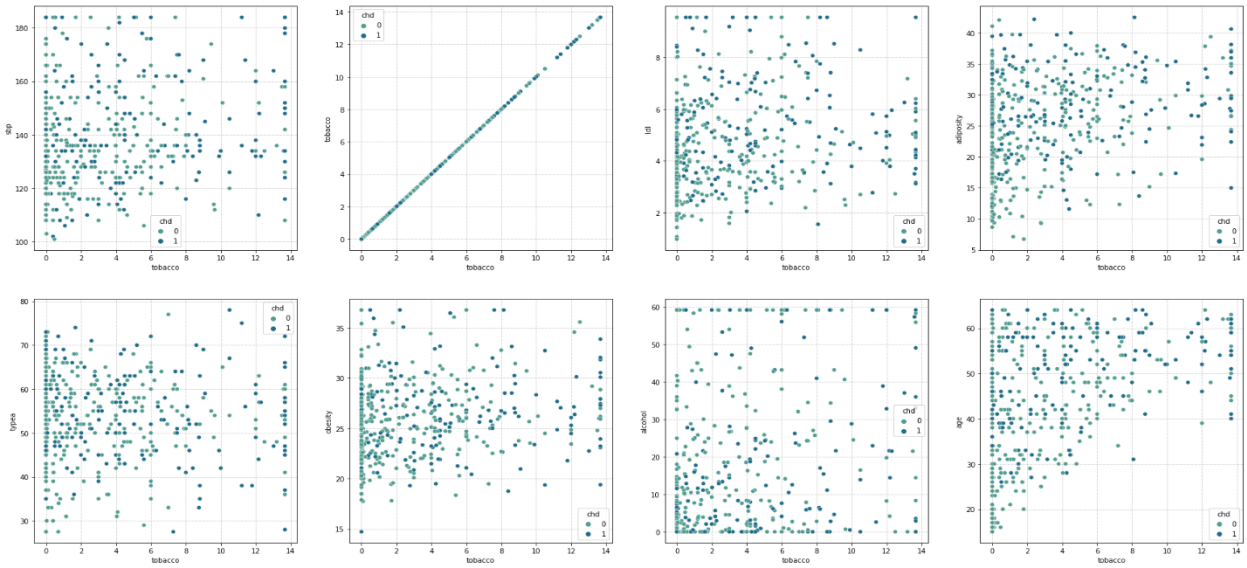


Figure 17: Tobacco scatterplot

Based on the data points distribution, there is no relationship between tobacco with other features.

- **LDL**

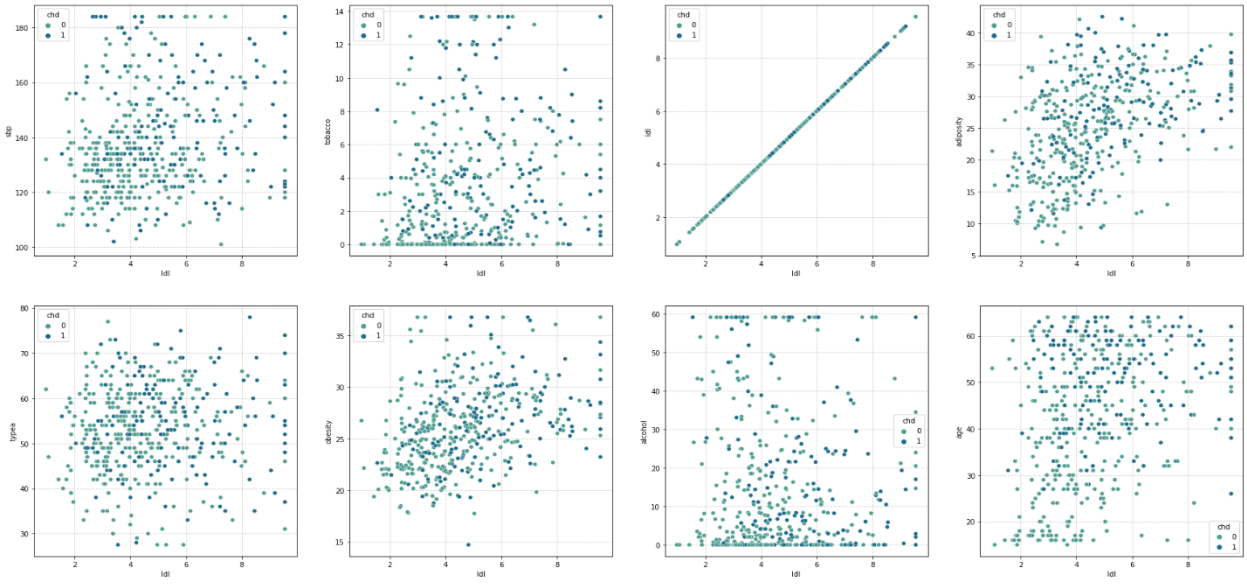


Figure 18: LDL scatterplot

In general, LDL has a positive relationship with other features, which means that any increase in LDL will equally increase the other factors like age, SBP, consumption of alcohol, Etc. Although LDL and adiposity show a tighter relationship than others, this makes sense; having extra weight raises the chances of having too much LDL.

- **Adiposity**

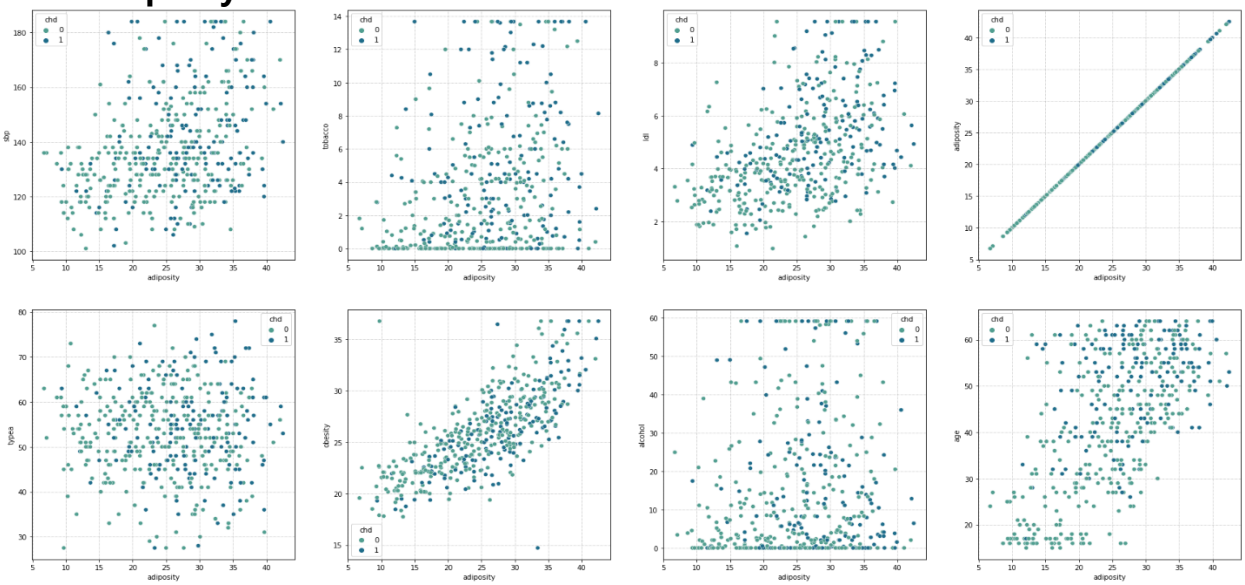


Figure 19: Adiposity scatterplot

In general, adiposity has a positive relationship with most of the features, except with Type A, the data distribution tends to be natural. To be noted, adiposity and obesity have a strong positive linear relationship.

- **Type A**

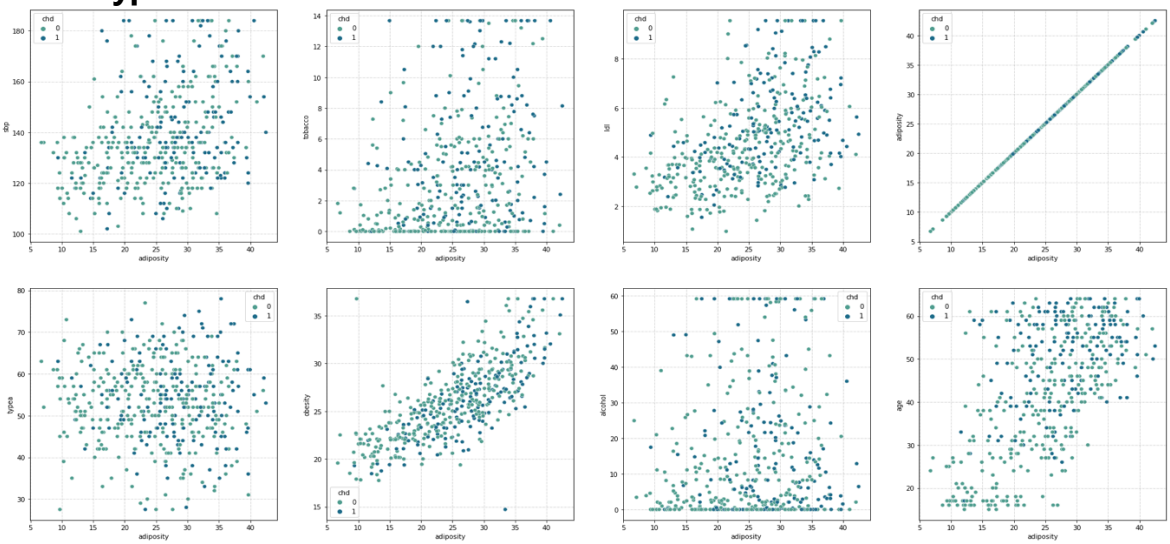


Figure 20: Type A scatterplot

Type A data points distribution have no relationship with the other features.

- **Obesity**

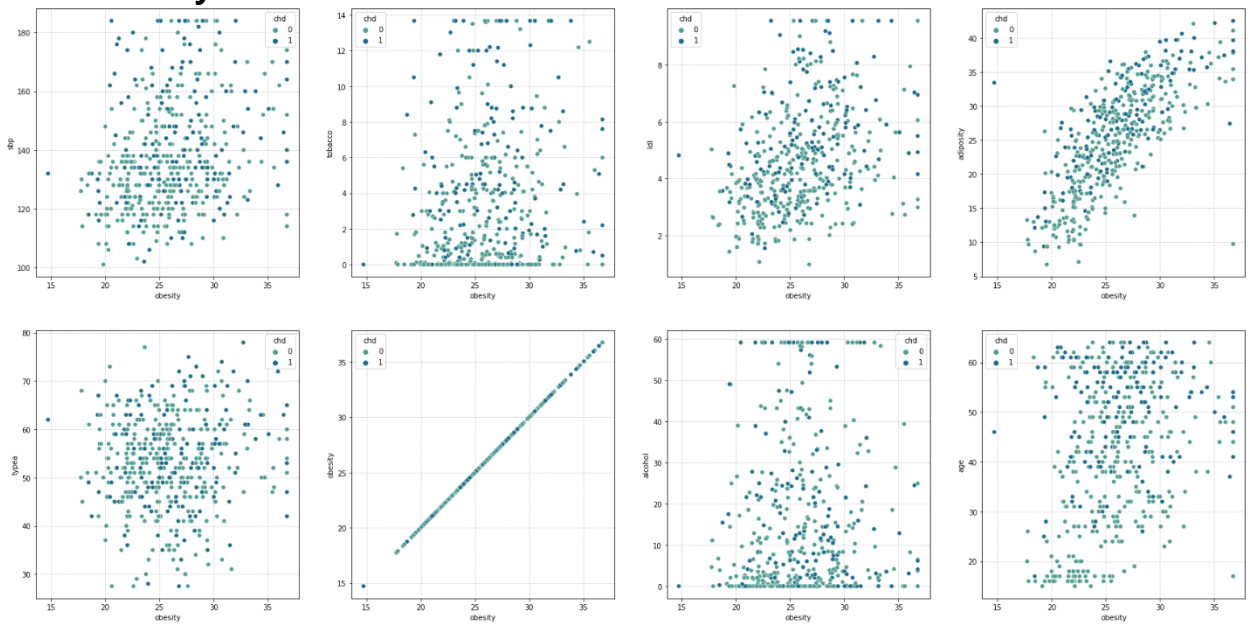


Figure 21: Obesity scatterplot

As has been said before, obesity and adiposity tend to have a strong positive relationship.

- **Alcohol**

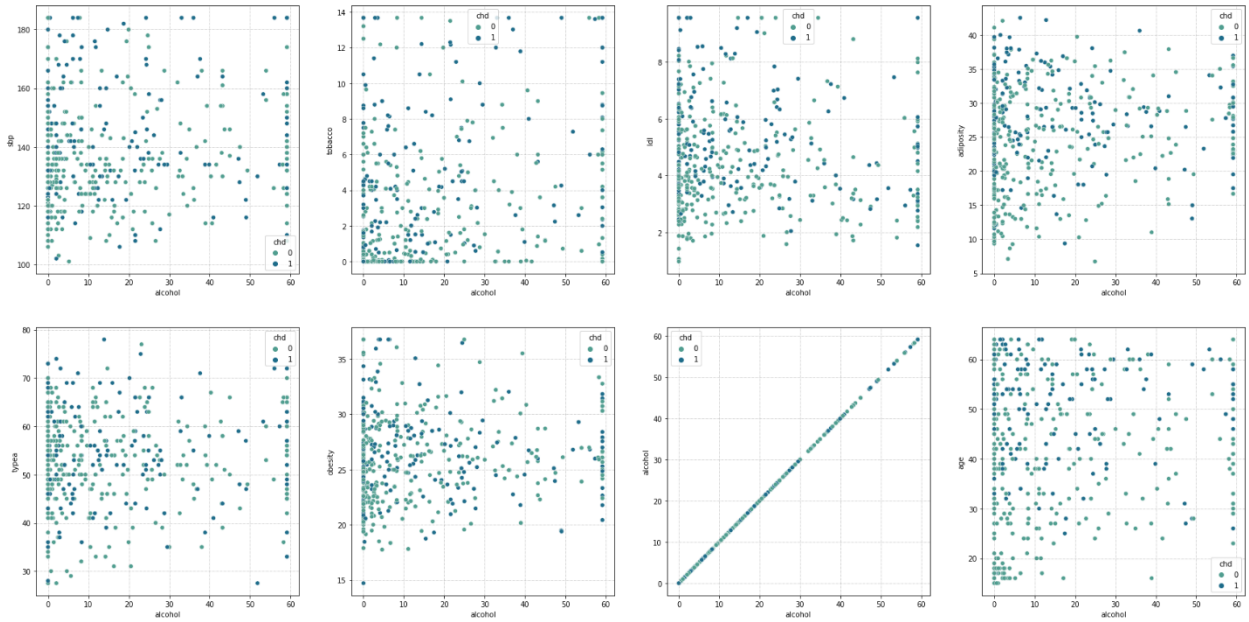


Figure 22: Alcohol scatterplot

Data points distribution for alcohol tends to have no relationship with the other features.

- **Age**

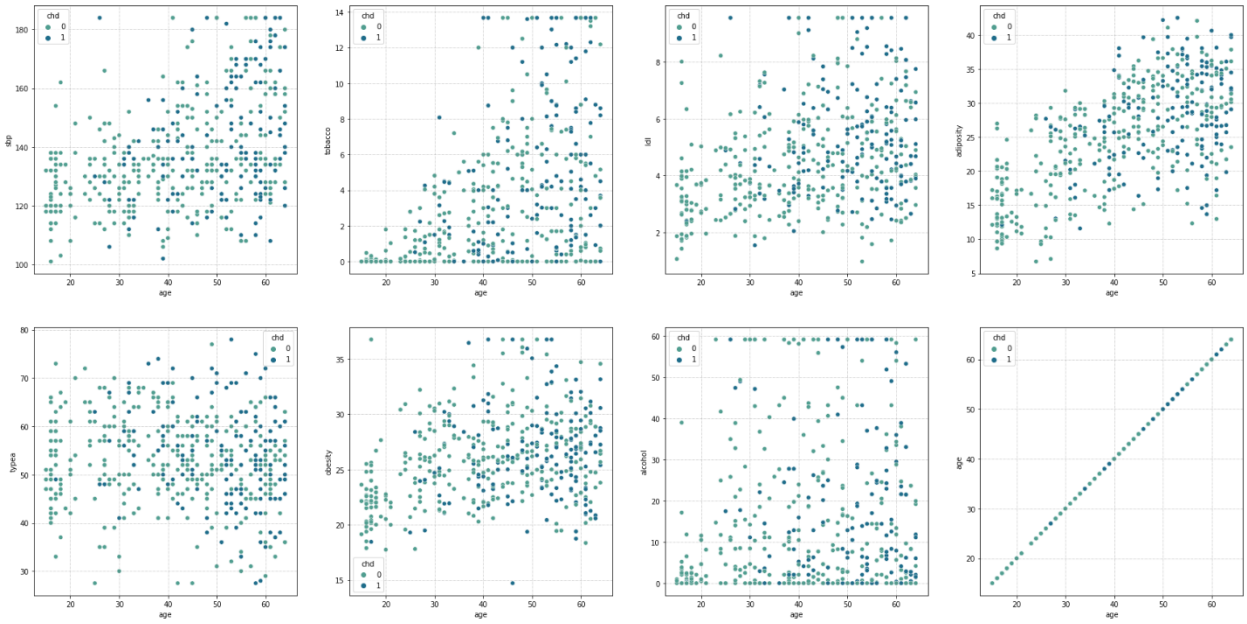


Figure 23: Age scatterplot

In general, age tends to have a neutral relationship with other features, but the relationship is positive for adiposity. After we get a general view of the data points distribution, we want to better understand the relationship between the features. Thus, we will find Correlation coefficients.

Correlation Coefficients

Correlation is used to measure the monotonic association between two variables in a dataset. Correlations are used to quantify, visualize and interpret bivariate (linear) relationships among measured variables. In correlated data, changes in one variable are usually associated with or driven by changes in another variable; this can be in the positive or negative correlation direction for both variables.

For purposes of this project, we undertook two correlation measures, Pearson correlation, and Spearman correlation. Pearson correlation coefficient is typically used to measure the association between variables in a normally distributed dataset, while the Spearman correlation coefficient is used in measuring the correlation between ordinal or skewed data that is not normally distributed. Both Spearman and Pearson correlation coefficients range between -1 and +1. +1 indicates a strong positive relationship between the variables under observation,0 indicates no linear association between the variables, while -1 indicates a strong negative correlation between the variables that are as one variable increases, the other variable decreases. Below is the correlation heatmap among the various variables in the dataset.

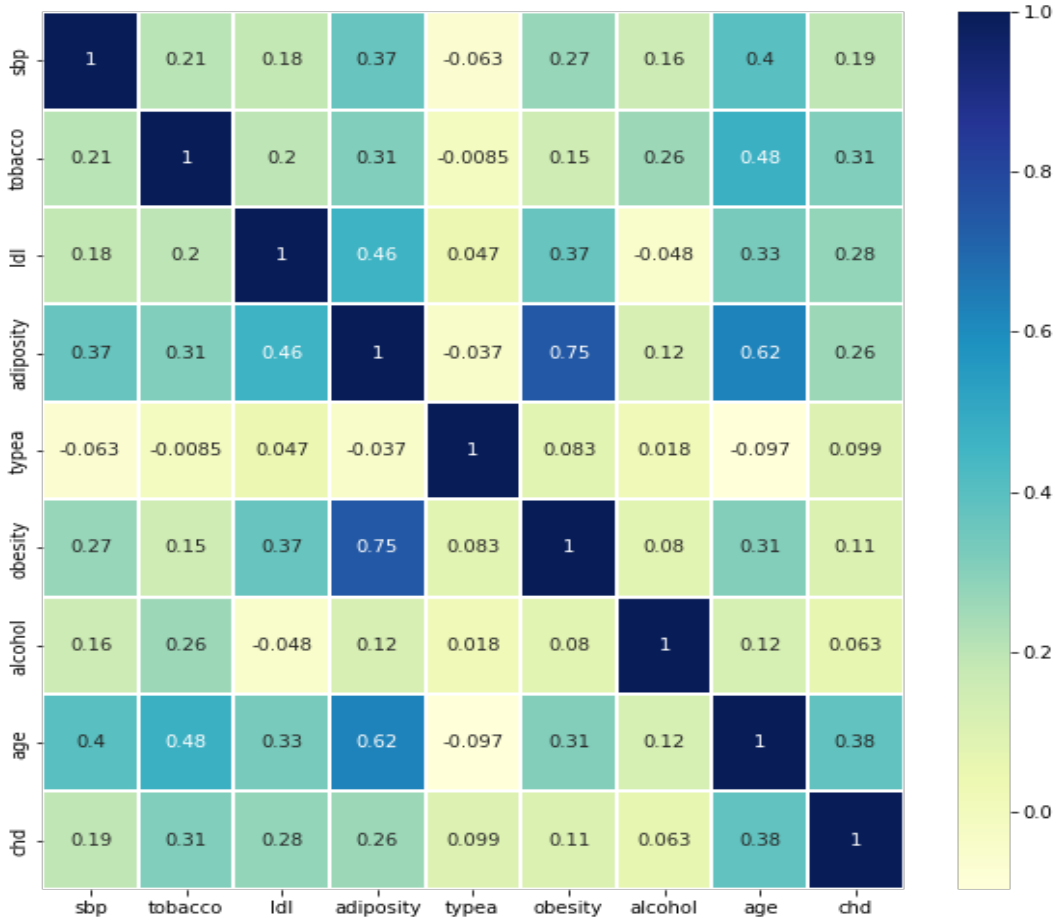


Figure 24:Correlation Coefficients

The correlation plot shows that some variables are highly positively correlated while others are negatively correlated. Below are the highly correlated variables:

- Adiposity and Obesity with a correlation coefficient of (0.75).

There are also moderate correlated variables such as:

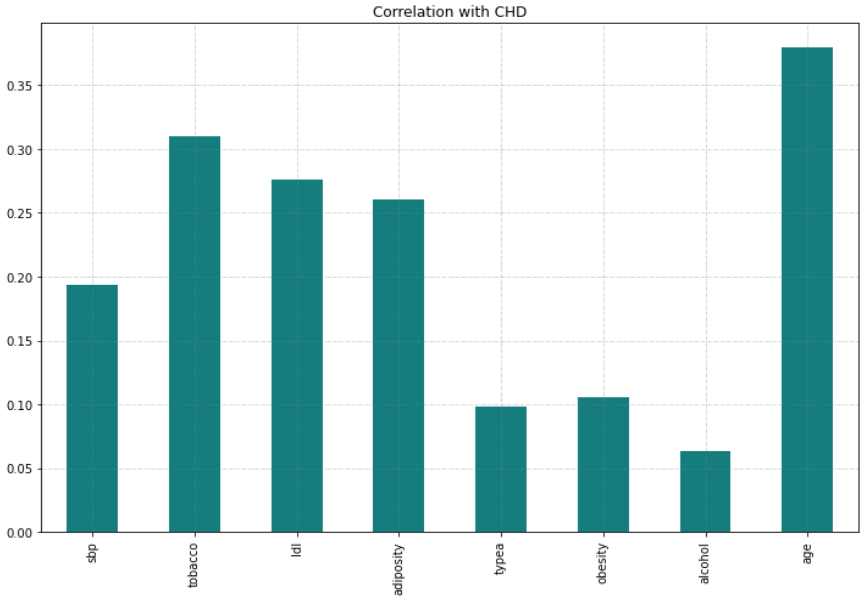
- Adiposity and Age with a correlation coefficient of (0.62).
- Age and tobacco are positively correlated with (0.48).
- Adiposity and LDL are positively correlated with (0.46).

The rest we consider weakly correlated variables. We also have negatively correlated variables, implying that as one variable increases, the other variable decreases. These variables are highlighted below:

- Alcohol and ldl are negatively correlated with a correlation negative coefficient of (-0.048).
- Age and typea have a negative correlation of (-0.097).
- Adiposity and typea have a negative correlation of (-0.037).
- Tobacco and type A have a negative correlation of (0.0085).
- Typea and sbp have a negative correlation of (-0.063).

We also looked at the correlation between the target variables i.e., coronary heart disease chd and all the independent variables. The results show in the below table:

Feature	Correlation
sbp	0.193353
tobacco	0.310341
ldl	0.276286
adiposity	0.260474
typea	0.098616
obesity	0.105491
alcohol	0.063105
age	0.379416
chd	1.000000



We can conclude that all variables have a low or medium correlation with CHD, with age having the highest correlation of 0.37 while typea has the lowest correlation of 0.06.

B. Relationship between categorical features

After we understand the association between two numerical features in our dataset, it is time to deal with the last thing in bivariate analysis: the correlation between family history and CHD (both are categorical). Therefore, we generated a contingency table that displays how two categorical variables are related in a table with how many individuals fall in each combination of categories. The table below:

CHD	Family History	Absent	Present
0		206	96
1		63	96

Moreover, we will perform chi-square test of independence to test association between two family history and CHD. The output gives us p-value, degrees of freedom and expected values.
Chi-square test formulate hypotheses:
Ho — There is no relationship between Family History and CHD
Ha — There is relationship between Family History and CHD
After we performed the test, The *p-value* of the chi-square test was: *.000000005907439*. Hence, it is less than 0.05; we reject Ho and accept Ha means that family history and CHD has a statistically significant association.

3. Feature Engineering

Feature engineering is the process of improving a machine model's accuracy by using domain knowledge to select and transform raw data's most relevant variables into features of predictive models that better represent the underlying problem. Feature engineering aims to improve the way statistical models and machine learning (ML) algorithms perform. For this project, we will perform four critical tasks for feature engineering: Categorical Encoding, Feature Scaling, Feature Selection, and Handling imbalanced data.

3.1 Categorical Encoding

Our dataset contains one categorical feature, which is family history. Most machine learning algorithms cannot work with categorical data, which must be transformed into numeric data. Thus, we used encoding process to convert family history. Encoding categorical data is a process of converting categorical data into integer format so that the data with converted categorical values can be provided to the different models to give and improve the predictions. Encoding has a multiple of techniques such as Label Encoding and One-Hot Encoding.

- Label encoding converts the data in machine-readable form, it replace the categorical value with a numeric value between 0 and the number of classes minus 1. If the categorical variable value contains 5 distinct classes, we use (0, 1, 2, 3, and 4).
- One-Hot encoding technique is used when the features are nominal(do not have any order) and for each category of a feature, create a new column (sometimes called a dummy variable) with binary encoding (0 or 1).

We used one-hot encoding to convert our categorical data to numerical data because family history is a binary variable. After, we converted family history into a numeric

variable 0 represents the absence, and 1 represents the presence of the heart disease in the family history.

3.2 Feature Scaling

Feature scaling is done owing to the sensitivity of some machine learning algorithms to the scale of the input values. This technique of feature scaling is sometimes referred to as feature normalization. The commonly used processes of scaling include:

- Normalization, the process involves the rescaling of all values in a feature in the range 0 to 1, also known as Min-Max Scaling. to In other words, the minimum value in the original range will take the value 0, the maximum value will take 1 and the rest of the values in between the two extremes will be appropriately scaled.
- Standardization/Variance scaling: All the data points are subtracted by their mean and the result divided by the distribution's variance to arrive at a distribution with a 0 mean and variance of 1.

Since our independent variables do not have the same range, we have to implement feature scaling to improve our machine learning algorithm training efficiency. Therefore, we implemented by Min-Max scaling, which subtracting the minimum value of the feature then dividing by the range. The table below list the five first rows form our dataset after implementing normalization:

	sbp	tobacco	ldl	adiposity	famhist	typea	obesity	alcohol	age
0	0.710843	0.877754	0.554178	0.457902	1.0	0.425743	0.479989	1.000000	0.755102
1	0.518072	0.000731	0.400175	0.611748	0.0	0.544554	0.641893	0.034821	0.979592
2	0.204819	0.005852	0.291673	0.714406	1.0	0.485149	0.654138	0.064402	0.632653
3	0.831325	0.548597	0.633513	0.875245	1.0	0.465347	0.783390	0.410074	0.877551
4	0.397590	0.994788	0.294006	0.588531	1.0	0.643564	0.511281	0.969236	0.693878

3.3 Handling imbalanced data

Imbalanced data sets are a special case for classification problem where the class distribution is not uniform among the classes. Typically, they are composed by two classes: The majority (negative) class and the minority (positive) class.

For our dataset, people with no CHD are 302 and people with CHD are 159. Thus, we has Imbalanced dataset. To deal with that, we use Oversampling technique to resampling our dataset which adds examples of the minority class to balance the dataset. The basic

approach of random sampling with replacement from the minority class called Simple random oversampling. We'll begin with simple random oversampling. This is a straightforward approach. We simply take copies/samples with replacement from the minority class, until the minority class has the same number of examples as the majority class. In the end, we have the new training dataset with the two classes balanced: both with 604 observations.

3.4 Feature Selection

Feature Selection is the process of selectively reducing the number of input variables; this is always desirable to reduce the computational costs of modeling, and it often enhances model performance. Features are sometimes more or less critical to model accuracy or may lose relevance in the context of other features. Feature selection algorithms analyze features for relevance and functionality and then determine which features are most useful and deserve to be prioritized and which should be removed for redundancy. Feature selection Techniques can be classified into four categories: filter methods, wrapper methods, embedded methods, and hybrid methods. For our dataset, we will use the correlation coefficient to select the essential descriptive features for the model. The logic behind using correlation for feature selection is that the good variables are highly correlated with the target. Furthermore, variables should be correlated with the target but uncorrelated among themselves. If two variables are correlated, we can predict one from the other. Therefore, if two features are correlated, the model only needs one of them, as the second one does not add additional information.

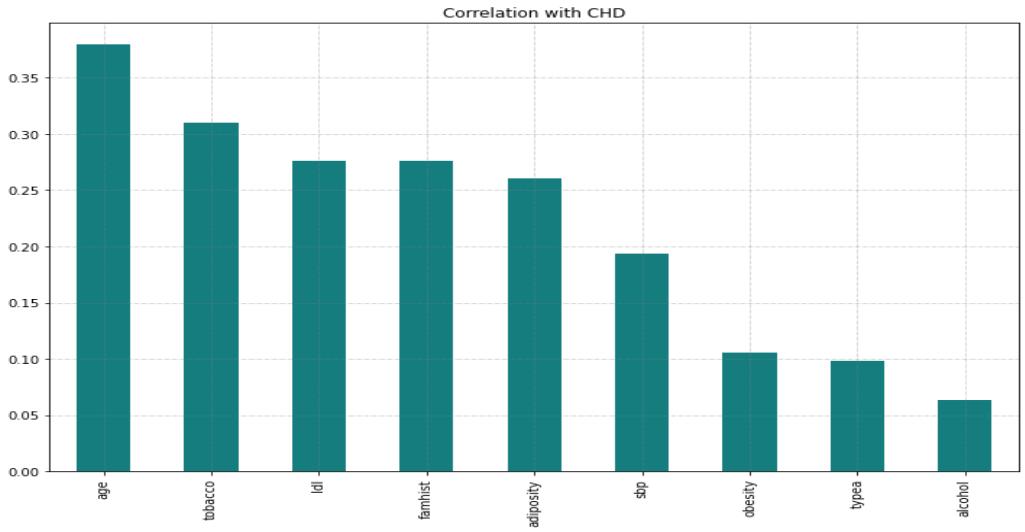


Figure 25:Correlation with CHD

Based on the correlation results, the most important features highly correlated with CHD are Age, Tobacco, LDL, SBP, Adiposity, and Family History. Furthermore, we already used Pearson's correlation to calculate the correlation score between the features; obesity has a 0.75 score with adiposity representing a high correlation. Therefore, we can drop obesity because it will not improve the model performance, and it may affect the linear models.

Chapter 4

Modeling

In this phase, various modeling techniques are selected and applied and their parameters are calibrated to optimal values. Typically, several techniques exist for the same problem type. Some techniques have specific requirements on the form of data. Therefore, stepping back to the data preparation phase may be necessary. Modeling steps include the selection of the modeling technique, the generation of test design, the creation of models, and the assessment of models.

1. Model Selection

In order to select the prop model for the dataset, first, it is important to understand what machine learning is and how it works. Machine learning (ML) is a core sub-area of Artificial Intelligence (AI). ML applications learn from experience (or to be accurate data) as humans do without direct programming.

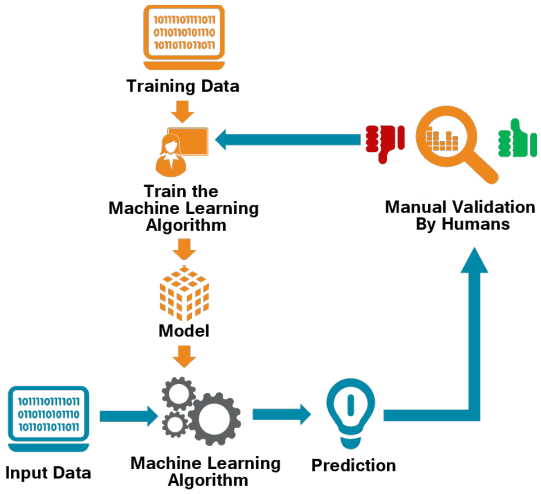


Figure 26: Machine learning Process

The process of ML starts with inputting training data into the selected algorithm. New input data is fed into the ML algorithm to test whether the algorithm works correctly. The prediction and results are then checked against each other. If the prediction and results do not match, the algorithm is re-trained multiple times until it gets the desired outcome. ML algorithms can be broadly classified into four broad groups:

- **Supervised learning**

Supervised learning is a type of machine learning that uses labeled data to train machine learning models. In labeled data, the output is already known. The model just needs to

map the inputs to the respective outputs. Supervised machine learning algorithms are categorized into two types depending on the problem to be addressed. These are:

- Classification Methods these are machine learning tasks that make use of machine learning algorithms to assign or group data to specific categories or classes accurately.
- Regression is supervised machine learning algorithms used to draw out the relationship between the independent and dependent variables. They are usually employed when we are interested in predicting values based on a given data point, such as sales revenue for a given quarter or a specific business.

- **Unsupervised Learning**

Unsupervised learning is a type of machine learning that uses unlabeled data to train machines. Unlabeled data doesn't have a fixed output variable. The model learns from the data, discovers the patterns and features in the data, and returns the output. Unsupervised machine learning are usually employed for three main tasks that include:

- Clustering: This unsupervised learning model involves grouping of data that is unlabelled based on their feature similarities. Algorithms used in clustering include k-means clustering, which groups similar items into k groups with K being the size and number of groups.
- Association: this algorithm uses rules to unearth relationships and associations between variables for a given dataset, a good example of association is its implementation in search engine recommendations and also shopping basket analysis.
- Dimensionality reduction: This is a machine learning algorithm used when the number of features or variables in a dataset of interest is too high. It reduces the number of input features to a manageable number while not losing important features. Dimensionality reduction is used in the pre-processing stage or feature engineering.

- **Semi-supervised Learning**

Semi-supervised learning is similar to supervised learning, but instead uses both labelled and unlabeled data. Labelled data is essentially information that has meaningful tags so that the algorithm can understand the data, whilst unlabeled data lacks that information. By using this combination, machine learning algorithms can learn to label unlabeled data.

- **Reinforcement Learning**

Reinforcement Learning trains a machine to take suitable actions and maximize its rewards in a particular situation. It uses an agent and an environment to produce actions and rewards. The agent has a start and an end state.

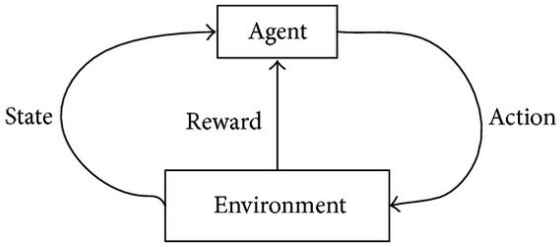


Figure 27: Reinforcement Learning

1.1 Choosing The Learning Task

Various factors determine when choosing the learning task for the problem. These factors include:

1. Nature and knowledge of data: understanding the data structure and complexity will help dictate which algorithm to use. A dataset with hundreds of features will use a different algorithm from one with five features.
2. Labeled or unlabeled data: if the dataset under consideration is labeled, supervised machine learning algorithms are chosen, unlike if the data is not labeled.
3. Processing speed: Different machine learning algorithms have different training speeds depending on the number of parameters; this makes our choice of algorithm dependent on time constraints needed for a given algorithm training based on these parameters.

Based on the above considerations, we settled on classification as our algorithm because our dataset has input and output labels, as we have nine input features and one target feature, CHD. This makes supervised learning to be the appropriate choice, and since our output will be based on classes, i.e., whether someone has coronary heart disease or not, the classification algorithms became the appropriate algorithm for our task. We briefly discuss classification algorithms below before deciding on the specific machine learning algorithm that falls under classification to use. In machine learning, classification refers to a predictive modeling problem where a class label is predicted for a given set of input data. It is a supervised learning algorithm that is used to classify observations into specific categories based on the training data.

1.2 Classification Algorithms

Classification algorithms take labeled input data. Classification algorithm involves doing some mathematical processing on input variable(x) and mapping this input to a discrete output function(y). classification algorithms need a training dataset with inputs and outputs from which the algorithm learns from. The algorithm uses the training dataset to model how to map the input data to a specific class label in our case whether a patient has coronary heart disease or not based on input features. There four major types of

classification algorithms: binary classification, multi-class classification, multi-label classification, and imbalanced classification

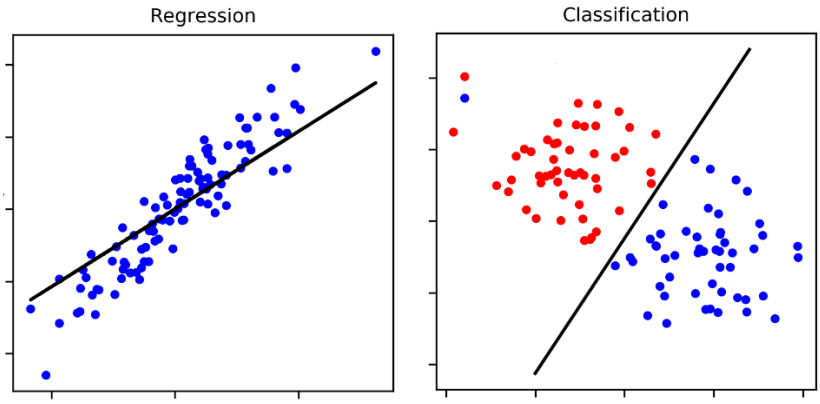


Figure 28:Classification Algorithms

For our task, we employed binary classification algorithms. Binary classification refers to classification tasks that have two class labels, such as whether one has coronary heart disease or not. Typically, binary classification tasks involve one class that is the normal state and another class that is the abnormal state, for in the coronary heart disease, no coronary heart disease is the normal state, and having coronary heart disease is the abnormal state. The class for the normal state is assigned the class label 0, and the class with the abnormal state is assigned the class label 1. For purposes of this task, we used the below three classification algorithms:

- Support Vector Machine (SVM)
- Decision Trees
- Logistic Regression

2. Test Design

For this project, we used scikit-learn Python library to conduct the selected models, support vector machines (SVM), decision trees, and logistic regression. Those models will be applied to the dataset after being cleaned and transformed, containing 461 observations, eight descriptive features (considering obesity was dropped), and one target feature, CHD. We used the Hold-out Validation method to split the dataset into training and testing sets.

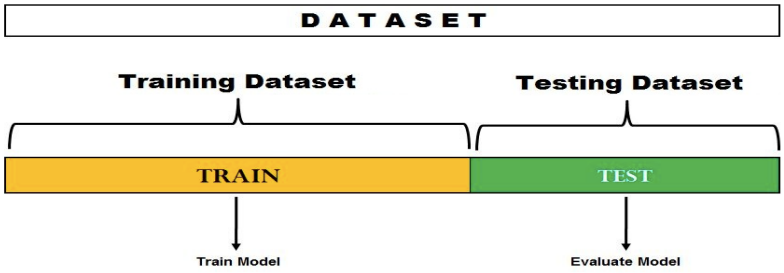


Figure 29:Hold-out Validation

The hold-out method for model evaluation represents the mechanism of splitting the dataset into training and test datasets. The model is trained on the training set and then tested on the testing set to get the most optimal model. Our dataset was split into training and testing sets; 70% of the original data will be used for training and 30% for testing. Therefore, 422 observations were used for training and 182 for testing.

Furthermore, for the purpose of assess the models, the following performance metrics will be introduce and discuss on each model:

1. **Confusion Matrix**

Confusion Matrix is a tabular visualization of the ground-truth labels versus model predictions. Each row of the confusion matrix represents the instances in a predicted class and each column represents the instances in an actual class. Confusion Matrix is not exactly a performance metric but sort of a basis on which other metrics evaluate the results.

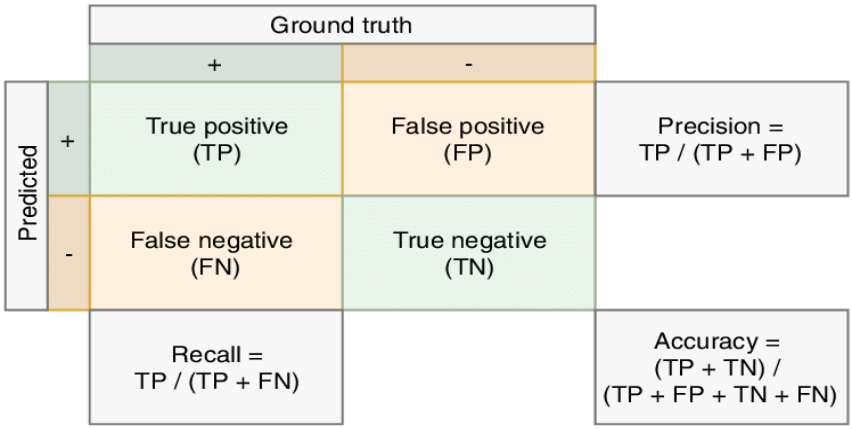


Figure 30: Confusion Matrix

True Positive(TP) signifies how many positive class samples your model predicted correctly.

True Negative(TN) signifies how many negative class samples your model predicted correctly.

False Positive(FP) signifies how many negative class samples your model predicted incorrectly. This factor represents Type-I error in statistical nomenclature.

False Negative(FN) signifies how many positive class samples your model predicted incorrectly. This factor represents Type-II error in statistical nomenclature.

2. Accuracy

Classification accuracy is perhaps the simplest metric to use and implement and is defined as the number of correct predictions divided by the total number of predictions, multiplied by 100.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN}$$

3. Precision

Precision is the ratio of true positives and total positives predicted.

$$Precision = \frac{TP}{TP+FP}$$

4. Recall/Sensitivity

A Recall is essentially the ratio of true positives to all the positives in ground truth.

$$Recall = \frac{TP}{TP+FN}$$

5. F1-score

The F1-score metric uses a combination of precision and recall. In fact, the F1 score is the harmonic mean of the two. The formula of the two essentially is:

$$F1\text{-}score = \frac{2}{\frac{1}{Precision}+\frac{1}{Recall}}$$

6. Specificity

The number of negatives returned by ML model.

$$Specificity = \frac{TN}{TN+FP}$$

7. AUC (Area Under ROC curve)

AUC (Area Under Curve)-ROC (Receiver Operating Characteristic) is a performance metric, based on varying threshold values, for classification problems. As name suggests, ROC is a probability curve and AUC measure the separability. In simple words, AUC-ROC metric will tell us about the capability of model in distinguishing the classes. Higher the AUC, better the model.

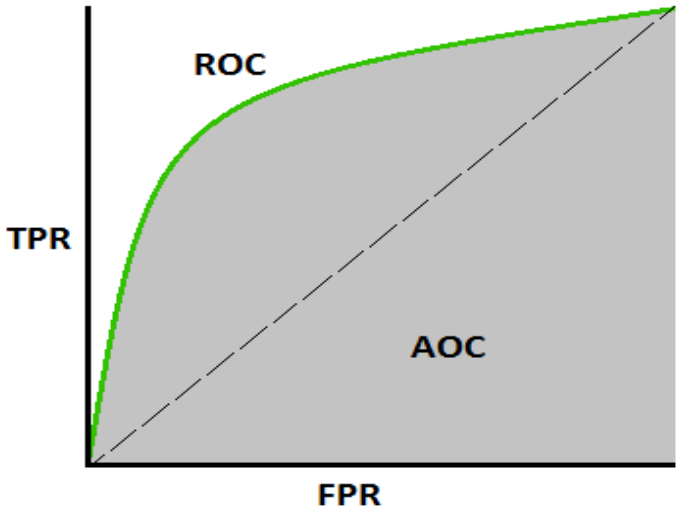


Figure 31:Area Under ROC curve

Mathematically, it can be created by plotting TPR (True Positive Rate) i.e. Sensitivity or recall vs FPR (False Positive Rate) i.e. 1-Specificity, at various threshold values. Following is the graph showing ROC, AUC having TPR at y-axis and FPR at x-axis.'

3. Model Building and Assessment

3.1 Support Vector Machines

Support Vector Machines (SVM) is considered a classification approach, but it can be employed in both classification and regression problems. It can easily handle multiple continuous and categorical variables. SVM constructs a hyperplane in multidimensional space to separate different classes. SVM generates optimal hyperplane in an iterative manner, used to minimize an error.

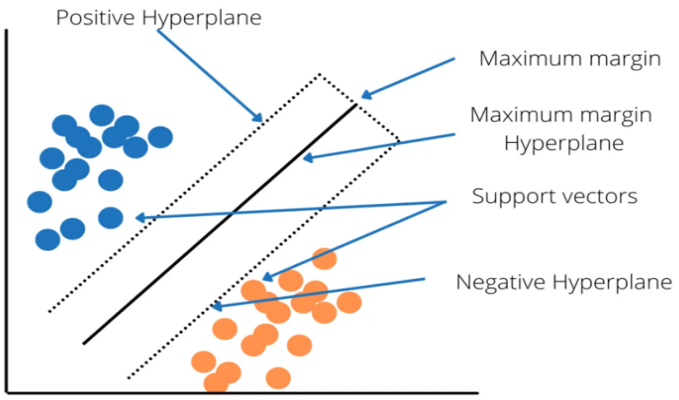


Figure 32:Support Vector Machines

Data points closest to the hyperplane are called **support vectors**. These points will define the separating line better by calculating margins and are more relevant to the construction of the classifier. A **hyperplane** is a decision plane that separates between a set of objects having different class memberships. **Margin** is the distance between the two lines on the class points closest to each other. It is calculated as the perpendicular distance from the line to support vectors or nearest points.

SVM algorithm

The objective of SVM is to draw a line that best separates the two classes of data points. SVM produces a line that cleanly divides the two classes. The margins and support vectors are used to construct the hyperplane. Some problems cannot be solved using a linear hyperplane because they are non-linearly separable. SVM uses a kernel trick to transform the input space into a higher-dimensional space in such a situation. Different types of the kernel include:

- **Linear Kernel** is a regular dot product for two observations. The sum of the multiplication of each pair of input values is the product of two vectors.
- **Polynomial Kernel** is a more generalized form of Linear Kernel. The polynomial Kernel can tell if the input space is curved or nonlinear. It has one parameter, degree, and if degree = 1, it is similar to the linear transformation.
- **Radial Basis Function Kernel** can map an input space into an infinite-dimensional space. It has gamma as a parameter which ranges from 0 to 1. A higher gamma value will perfectly fit the training dataset, which causes over-fitting. The gamma = 0.1 is considered to be a good default value.

For implementing SVM into our dataset, the SVM model from scikit-learn Python library was used. The SVM model has two hyperparameters; they need to be set before training the model. These parameters are:

- Regularization parameter (C) is mainly used for the Penalty parameter of the error term. It considers the degree of correct classification that the algorithm must meet or the degree of optimization the SVM has to meet.
- Gamma decides how much curvature wants in a decision boundary.

Kernel, C, and Gamma were tuned using GridSearchCV, and they were set to 'poly', '1', and '1'. Then, the training and testing set were fitted to the model, and the last step was to determine the model performance. Therefore, the table below lists the performance metrics for SVM:

	Accuracy	Precision	Recall	F1-score	Specificity	AUC
0	0.82	0.84	0. 79	0.81	0.79	0.824
1		0.81	0.86	0.83		3

The model performs well for the dataset, showing high accuracy with 82% and a true positive rate of 86%, which is an excellent rate and it is important because of the classification objective, which is to classify people based on their health information into having CHD or not having CHD. To be more precise, if they are a true positive case where a person has CHD and by error is classified as negative, it means he/she has not have CHD, it is a huge problem. Therefore, true positive rate or recall should be paid attention to in the medical field. If we looked to the confusion matrix to determine how the model predict each level at CHD feature:

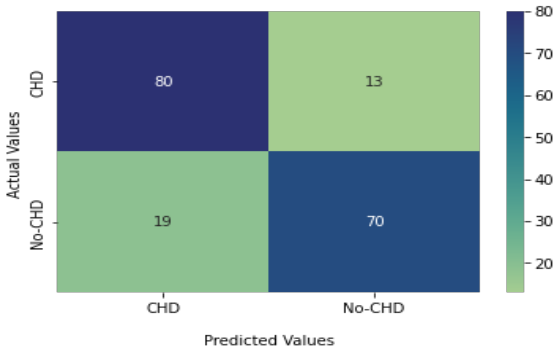


Figure 33: SVM confusion matrix

The model is predicted well for both levels, with slightly better performance for people with CHD class. Another way used to measure the model performance is the ROC curve and AUC. The SVM curve is closer to the top-left corner, indicating good performance. Moreover, the AUC is 0.8243, which means there is an 80% chance that the model will be able to distinguish between the CHD class and No-CHD class, representing an excellent score.

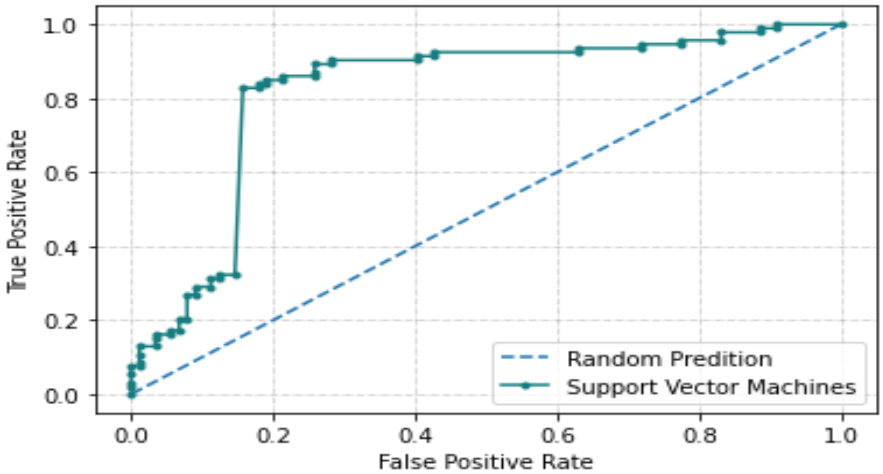


Figure 34: SVM ROC curve and AUC

3.2 Decision Tree

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees. The figure below shows the concept of the Decision Tree.

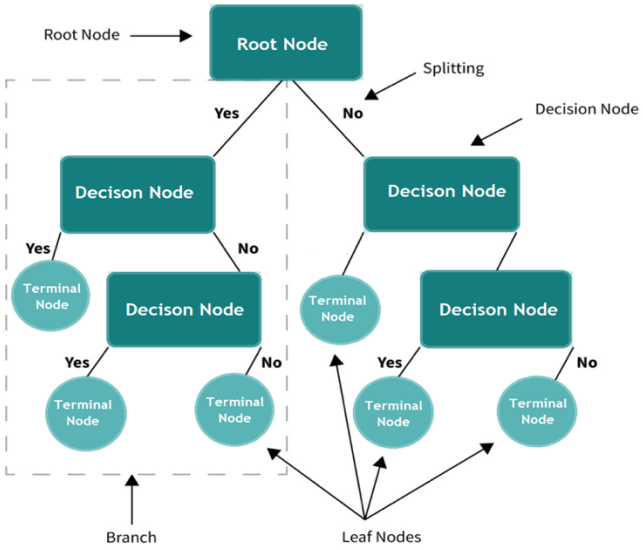


Figure 35:Decision Tree

- Root Node: It represents the entire population or sample and this further gets divided into two or more homogeneous sets.
- Splitting: It is a process of dividing a node into two or more sub-nodes.
- Decision Node: When a sub-node splits into further sub-nodes, then it is called the decision node.
- Leaf / Terminal Node: Nodes do not split is called Leaf or Terminal node.
- Pruning: When we remove sub-nodes of a decision node, this process is called pruning. You can say the opposite process of splitting.
- Branch / Sub-Tree: A subsection of the entire tree is called branch or sub-tree.
- Parent and Child Node: A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of a parent node.

Decision Tree Classifier algorithm

The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by Information based learning. In order to build a tree, we used the CART algorithm, which stands for Classification and Regression Tree

algorithm. In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node. For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree. Thus, to start from the root node we first needs to calculate the best attribute. There are two popular techniques for select best attribute Entropy and Gini index.

- Gini Index, also known as Gini impurity, calculates the amount of probability of a specific feature that is classified incorrectly when selected randomly. If all the elements are linked with a single class then it can be called pure. the Gini index varies between values 0 and 1, where 0 expresses the purity of classification.

$$Gini(t,D)=1-\sum_{1\in levels(t)}p(t=l)^2$$

- Entropy: It is used to measure the impurity or randomness of a dataset. Entropy is calculated between 0 and 1.

$$H(t)=-\sum_{i=1}(p(t=i)*log_2(P(t=i)))$$

For implementing Decision Tree into our dataset, used the Decision Tree Classifier model from scikit-learn Python library. The Decision Tree Classifier model has three hyperparameters; they need to be set before training the model. These parameters are:

- The first parameter to tune is max_depth. This indicates how deep the tree can be. The deeper the tree, the more splits it has and it captures more information about the data. We fit a decision tree with 4 max depth ranging.
- The second parameter to tune is criterion, it's to measure the quality of a split. Supported criteria are "gini" for the Gini impurity and "entropy" for the information gain.
- The third parameter to tune is min_samples_split which is the minimum number of samples required to split an internal node.

criterion, max_depth, and min_samples_split were tuned using GridSearchCV, and they were set to 'gini, '4', and '2'. Then, the training and testing set were fitted to the model, and draw the model:

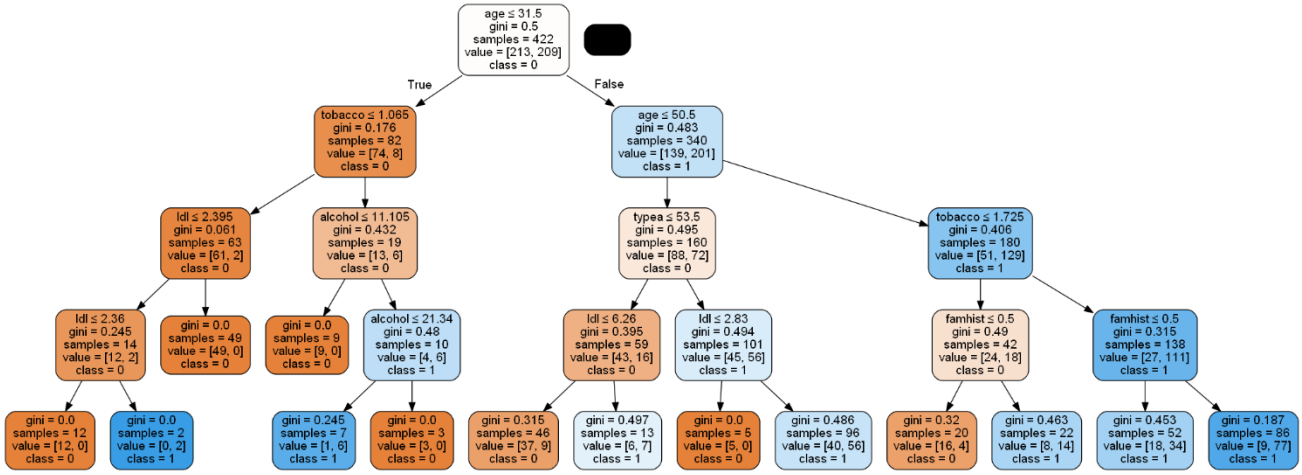


Figure 36: Decision Tree Model

The last step was to determine the model performance. Therefore, the table below lists the performance metrics for Decision Tree Classifier:

	Accuracy	Precision	Recall	F1-score	Specificity	AUC
0	0.77	0.86	0.63	0.73	0.62	0.80
1		0.72	0.90	0.80		

In the table above, we can see the accuracy of the model is 77%. Precision equal 0.72 which means the model can predicts that a patient has heart disease and it is correct around 72%. Recall or Sensitivity for our model equal 0.90 which is means the model is able to identify the patients who actually have heart disease and it is correct with 90%. If we looked to the confusion matrix to determine how the model predict each level at CHD feature:

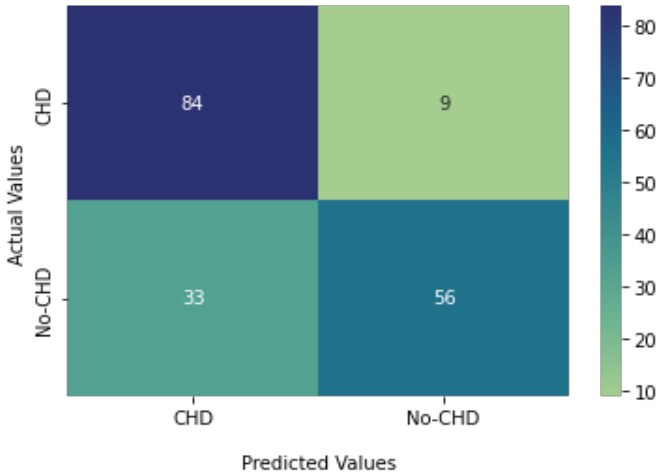


Figure 37: Decision Tree confusion matrix

In the confusion matrix, we see the 84 + 9 = 93 people have Coronary Heart Disease, 84(90%) were correctly classified. And of the 33 + 56 = 89 people that did not have Coronary Heart Disease 56(63%) were correctly classified. Another way used to measure the model performance is the ROC curve and AUC.

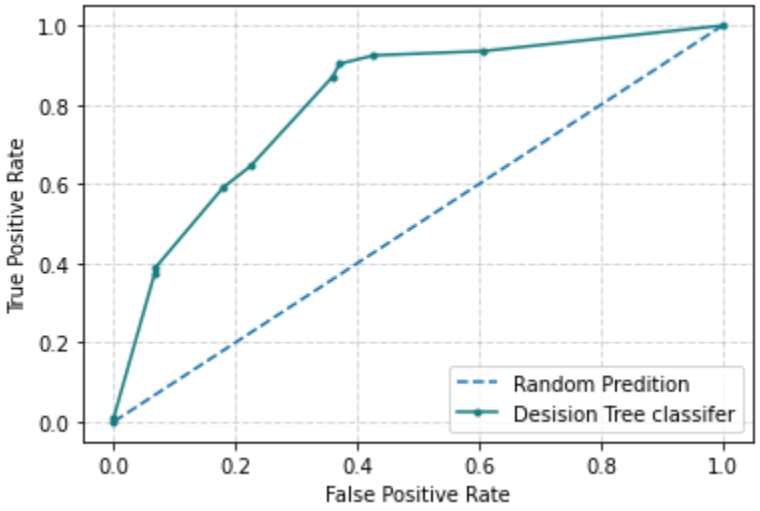


Figure 38: Decision Tree ROC curve and AUC

As shown in plot the AUC is 0.80, it means there is a 80% chance that the model will be able to distinguish between CHD class and No-CHD class.

3.3 Logistic Regression

Logistic regression is a classification algorithm that assigns observations to a discrete set of classes. Unlike linear regression, which outputs continuous number values, logistic regression transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to two or more discrete classes. It is an extension of the linear regression model for classification problems. However, instead of fitting a straight line or hyperplane, the logistic regression model uses the logistic function to squeeze the output of a linear equation between 0 and 1.

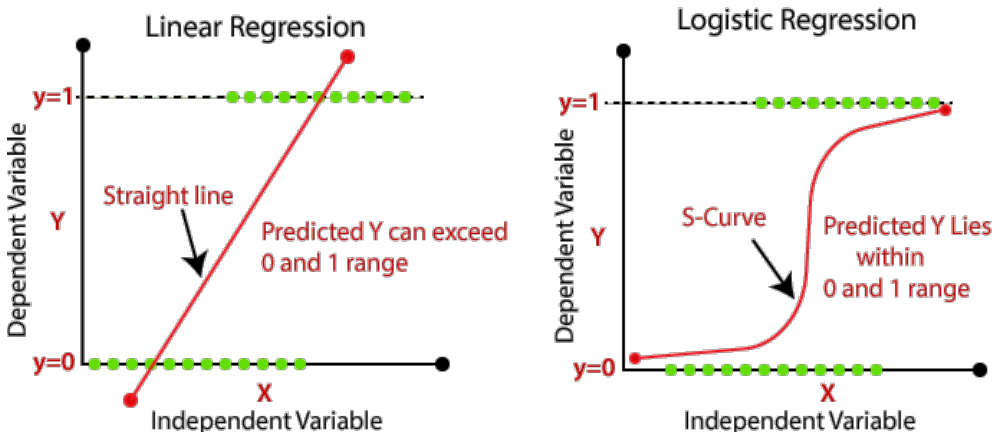


Figure 39: Linear Regression Vs. Logistic Regression

The step from linear regression to logistic regression is kind of straightforward. In the linear regression model, we have modeled the relationship between outcome and features with a linear equation:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \cdots + \beta_m x_m$$

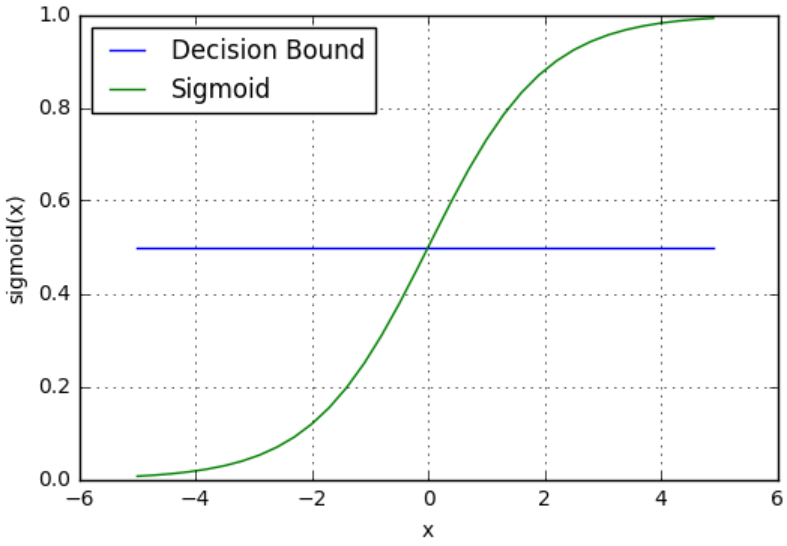
Here, β_0 is the y-intercept, β_1 is the slope of the line, x_1 is the value of the x coordinate, and $\hat{y}$ is the value of the prediction. For classification, we prefer probabilities between 0 and 1, so we wrap the right side of the equation into the logistic function (also called the sigmoid function); this forces the output to assume only values between 0 and 1.

$$p(\hat{y} = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \cdots + \beta_m x_m)}}$$

The logistic function returns a probability score between 0 and 1. In order to map this to a discrete class (true/false, 1/0), we select a threshold value or tipping point above which we will classify values into class 1, and below which we classify values into class 2. It is also called a decision boundary.

$$p \geq 0.5,\ class=1$$
$$p < 0.5,\ class=0$$

For example, if our threshold was .5 and our prediction function returned .7, we would classify this observation as positive. If our prediction were .2, we would classify the observation as negative.



Logistic Regression Algorithm

A prediction function can be written using the knowledge of logistics function and decision boundary concepts. A prediction function in logistic regression returns the probability of an observation being positive (1 or have CHD). As the probability gets closer to 1, our model is more confident that the observation is in class 1. Therefore, mathematically we can use the multiple linear regression equation to formulate our prediction function:

$$\hat{y} = \beta_0 + \beta_1 sbp + \beta_2 tobacco + \beta_3 ldl + \beta_4 adiposity + \beta_5 famhist + \beta_6 typea + \beta_7 alcohol + \beta_8 age$$

However, we will transform the output using the logistic function to return a probability value between 0 and 1.

$$p(\hat{y} = 1) = \frac{1}{1 + e^{-\hat{y}}}$$

Where is $\hat{y}$ is the expressed by the previous multiple linear regression equation.

For implementing logistic regression into our dataset, used the Logistic Regression model from scikit-learn Python library. It has four hyper-parameters to consider, below the list of these parameters:

- Penalty: This hyper-parameter is used to specify the type of normalization used. Few of the values for this hyper-parameter can be l1, l2 or none. The default value is l2.
- Inverse of regularization: This hyper-parameter is denoted as C. Smaller values of this hyper-parameter indicates a stronger regularization. Default value is 1.0

- Solver: This indicates which algorithm to use in the optimization problem. Default value is lbfgs. other possible values are newton-cg, liblinear, sag, saga.
- Max iter : max_iter represents maximum number of iterations taken for the solvers to converge a training process.

Penalty, C, solver, and max-iter were tuned using GridSearchCV, and they were set to 'l2', '0.23357214690901212', 'lbfgs', and 'l2'. Then, the training and testing set were fitted to the model, the following table list the intercept and coefficients for each feature:

feature	sbp	tobacco	ldl	adiposity	famhist	typea	alcohol	age
coefficient	0.224337	0.988278	0.606052	0.105902	0.842035	0.362246	0.072248	1.470710
intercept	-2.251505762514177							

Therefore, we can update our prediction function considering the coefficients and intercept as follows:

$$\hat{y} = (-2.251505762514177) + (0.224337)sbp + (0.988278)tobacco + (0.606052)ldl + (0.105902)adiposity + (0.842035)\ famhist + (0.362246)typea + (0.072248)alcohol + (1.470710)age$$

The last step was to determine the model performance. Therefore, the table below lists the performance metrics for logistic regression:

	Accuracy	Precision	Recall	F1-score	Specificity	AUC	Log Loss
0	0.74	0.76	0.67	0.71	0.67	0.815	0.54
1		0.72	0.80	0.76			

As shown, the model have accuracy with 74%, and a true positive rate with 80%, but it has fewer score for a true negative rate with 67%.

Another vital classification metric for logistic regression is Log Loss. Log loss, aka logistic loss or cross-entropy loss, is the loss function used in logistic regression and extensions such as neural networks, defined as the negative log-likelihood of a logistic model that returns probabilities for its training data.

Log-loss indicates how close the prediction probability is to the corresponding actual/true value (0 or 1 in the case of binary classification). For any given problem, a lower log loss value means better predictions. Our model offers a log loss of 0.54, not bad considering the size of the dataset.

Furthermore, we looked to the confusion matrix to determine how the model predict each level at CHD feature:

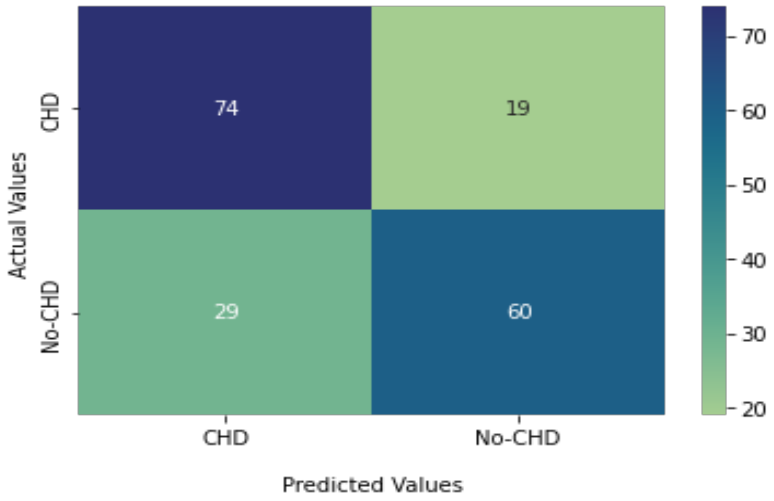


Figure 40: Logistic Regression confusion matrix

The model has many wrong predictions for both levels, but it performs better for people with CHD class.

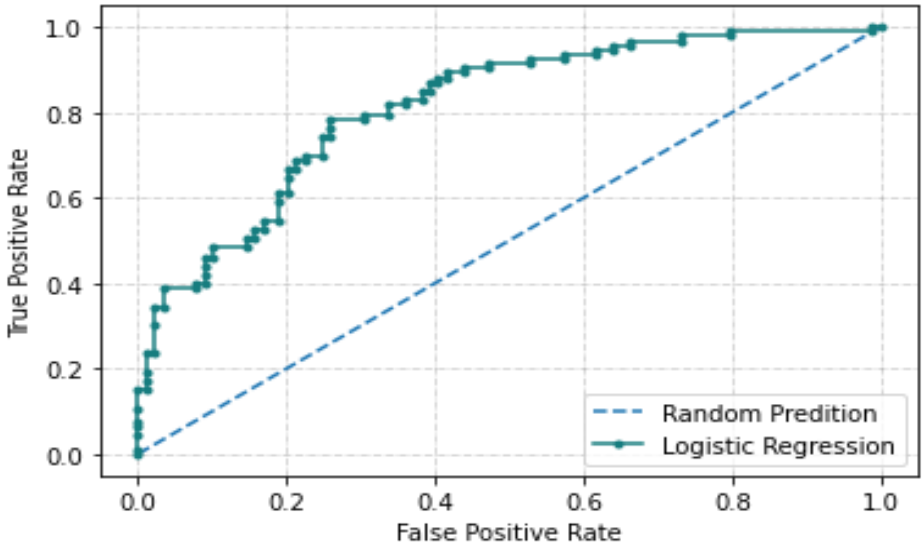


Figure 41: Logistic Regression ROC curve and AUC

Another metric to consider is the ROC curve and AUC. As seen below, the logistic regression curve is in between the false positive rate and true positive. Moreover, the AUC is 0.82, which means there is an 80% chance that the model will be able to distinguish between the CHD class and the No-CHD class.

After we have been building different models, the next step is to evaluate the results and choose the best model that will fit the project's purpose.

Chapter 5

Evaluation

Before the model is used, it is important to evaluate it. It must be questioned whether the model offers the quality to meet the project's objective and whether the model satisfies the project's objectives. Some phases may have to be run through if the goals cannot be achieved.
Our objective for this project is to determine whether a person has CHD or not, based on their health parameters. Thus, the model that will be selected and obtained should be predicted the results with higher accuracy to avoid misclassification of patients as much as possible.
Therefore, the table below lists the performance metrics for each model in order to choose the best model based on its results.

Model	CHD Levels	Accuracy	Precision	Recall	F1-score	Specificity	AUC
SVM	0	**0.82***	0.84	0. 79	0.81	**0.79***	**0.8243***
	1		**0.81***	0.86	**0.83***		
Decision Tree	0	0.77	0.86	0.63	0.73	0.62	0.80
	1		0.72	**0.90***	0.80		
Logistic Regression	0	0.74	0.76	0.67	0.71	0.67	0.815
	1		0.72	0.80	0.76		

Furthermore, a ROC curve has been plotted to determine the performance of each model.

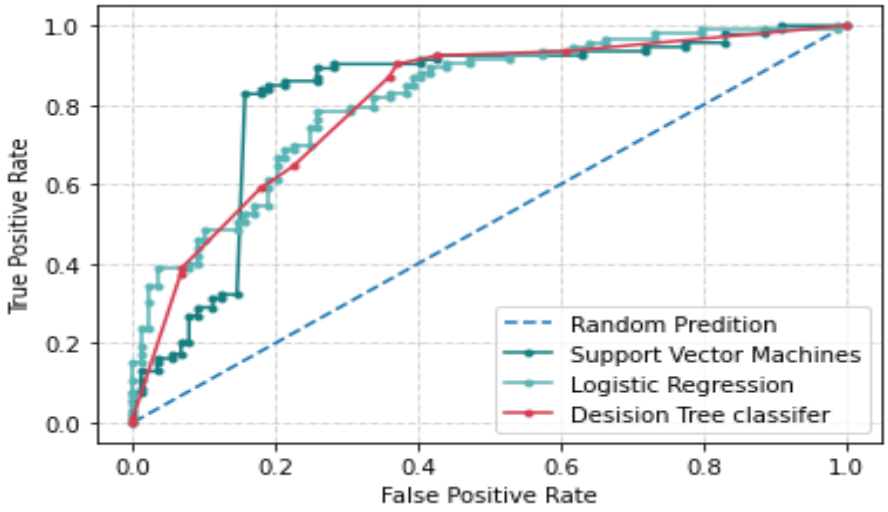


Figure 42: ROC curve for all models

Based on the models' performance, we can see that SVM predicts with higher accuracy with 82%, it has a good score for both true negative and positive rates; also, it is able with an 82% to distinguish between the CHD and the No-CHD class. Considering all of this, we can state that the SVM model is the best model that could solve our problem.

Chapter 6

Deployment

Deployment is the process of using new insights to make improvements within an organization. It is also the method of integrating a machine learning model into an existing production environment to make practical business decisions based on data. Deployment is one of the last stages in the machine learning life cycle and can be one of the most cumbersome.

For this project, the SVM model was integrated as a website for predicting CHD, using Flask, which is a micro web framework written in python. The web service has two pages: the first one is the form by which a user inserts his/her health parameters and submits them, and the second one is the results page, which displays his/her results as having CHD or not.

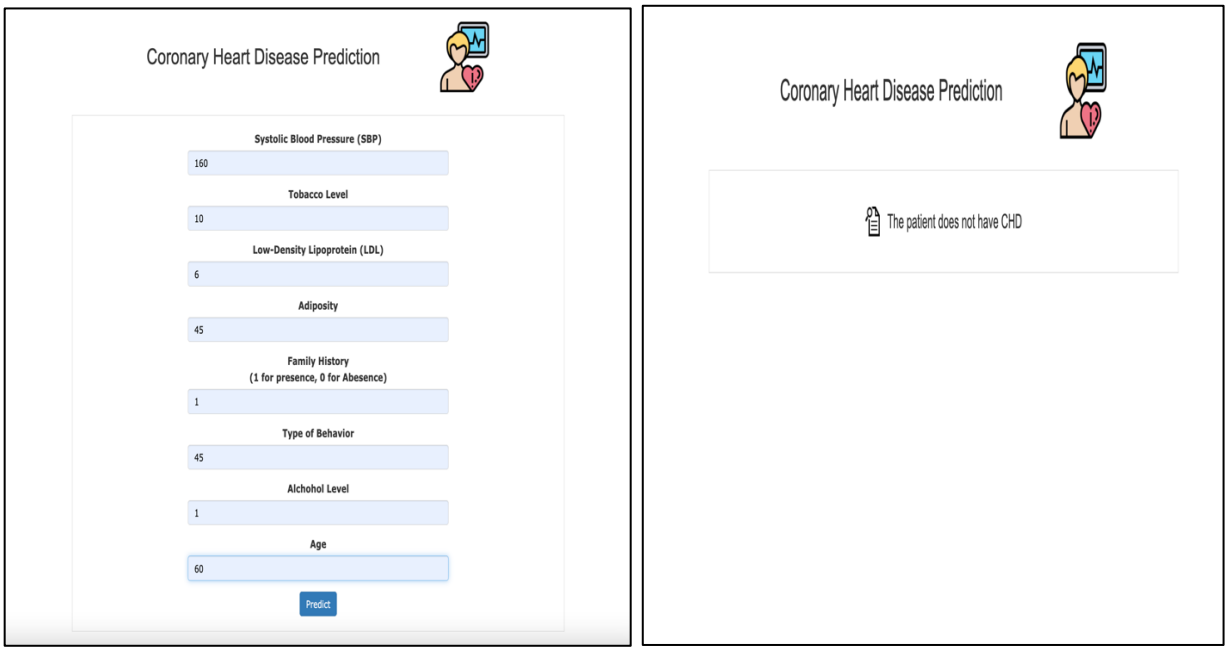


Figure 43: website to predict coronary heart disease

Conclusion

This project has been introducing the basic concepts of data management and visualization and machine learning models with the adoption of CRISP-DM and employed it for a real-world problem predicting whether a person has a CHD. The project covered different types of problems, such as missing data and outliers. Also, it covered some of the basic tasks in feature engineering like imbalanced data, categorical encoding, and feature selection. Different models have been implemented and evaluated, such as SVM, Decision tree, and logistic regression. Based on the evaluation results, SVM has been chosen as the best model to predict this type of problem. As the last step in the project, a web service has been introduced by integrating the SVM model with a flask framework. Thus, it can be used for business purposes. For future work, we can be developed the web service with more frontend flexible libraries such as React, Angular, or Vue js.

References

[1] CRISP-DM. (2022, May 10). Data Science Process Alliance. https://www.datascience-pm.com/crisp-dm-2/.

[2] Adiposity Medical Definition Written by Doctors. (2021, March 29). MedicineNet. https://www.medicinenet.com/adiposity/definition.htm.

[3] C.Rodriguez et al., "Systolic Blood Pressure Levels Among Adults With Hypertension and Incident Cardiovascular Events", *JAMA Internal Medicine*, vol. 174, no. 8, p. 1252, 2014. Available: 10.1001/jamainternmed.2014.2482.

[4] WHO Report On The Global Tobacco Epidemic", new york, 2013.

[5] R. Catapano and D. G, "Corrigendum to: '2016 ESC/EAS Guidelines for the Management of Dyslipidaemias'", *European Heart Journal*, vol. 39, no. 15, pp. 1254-1254, 2017. Available: 10.1093/eurheartj/ehx180.

[6] S. Kotz and N. Balakrishnan, Encyclopedia of statistical sciences. Hoboken, N.J.: Wiley-Interscience, 2006.

[7] B. Everitt and A. Skrondal, The Cambridge dictionary of statistics. Cambridge: Cambridge University Press, 2010.

[8] E. Babbie, The practice of social research. Boston: Cengage learning, 2016.

[9] J. Mandrekar, "Receiver Operating Characteristic Curve in Diagnostic Test Assessment", Journal of Thoracic Oncology, vol. 5, no. 9, pp. 1315-1316, 2010. Available: 10.1097/jto.0b013e3181ec173d.

[10] P. Schober, C. Boer and L. Schwarte, "Correlation Coefficients", Anesthesia & Analgesia, vol. 126, no. 5, pp. 1763-1768, 2018. Available: 10.1213/ane.0000000000002864.

[11] J. Brownlee, "How to Use ROC Curves and Precision-Recall Curves for Classification in Python", Machine Learning Mastery, 2022. [Online]. Available: https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/. [Accessed: 03- Apr- 2022].

[12] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition. [S.l.]: O'Reilly Media, Inc., 2019.

[13] Vaughan, J. (2019, November 1). Data quality. SearchDataManagement. https://www.techtarget.com/searchdatamanagement/definition/data-quality.

[14] What is Data Quality? Definition and FAQs | HEAVY.AI. (2022). Heavy.AI. https://www.heavy.ai/technical-glossary/data-quality.

[15] Bandgar, S. (2022, January 6). Data Preparation in Data Science - Analytics Vidhya. Medium. https://medium.com/analytics-vidhya/data-preparation-in-data-science-16f9311760.

[16] Kaur, J. (2022, March 16). Data Preprocessing and Data Wrangling in Machine Learning. XenonStack. https://www.xenonstack.com/blog/data-preparation

[17] Lianne & Justin @ Just into Data. (2022, January 3). Data Cleaning in Python (2020): the Ultimate Guide | Towards Data Science. Medium. https://towardsdatascience.com/data-cleaning-in-python-the-ultimate-guide-2020-c63b88bf0a0d.

[18] Bonthu, H. (2021, May 27). Detecting and Treating Outliers | How to Handle Outliers. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/05/detecting-and-treating-outliers-treating-the-odd-one-out/.

[19] Gawali, S. (2021, October 26). Missing Data Handling |How to Deal with Missing Data using Python. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/10/how-to-deal-with-missing-data-using-python/.

[20] Patil, P. (2021, December 18). What is Exploratory Data Analysis? - Towards Data Science. Medium. https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15.
[21] Jakhotia, R. (2021, February 24). Outlier Treatment in Python and R. K2 Analytics. https://www.k2analytics.co.in/outlier-treatment-in-python-and-r/.
[22] Singh, D. (2019, November 12). Finding Relationships in Data with Python. Pluralsight. https://www.pluralsight.com/guides/finding-relationships-data-with-python.
[23] Feature Engineering - The Ultimate Guide. (2022, March 15). Explorium. https://www.explorium.ai/wiki/feature-engineering/.
[24] Reddy, E. P. K. (2021, March 25). Feature Engineering Step by Step | Feature Engineering in ML. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/03/step-by-step-process-of-feature-engineering-for-machine-learning-algorithms-in-data-science/#h2_4.
[25] Alam, B. (2022, January 15). Implementation of Support Vector Machine (SVM) using Python. Hands-On-Cloud. https://hands-on.cloud/implementation-of-support-vector-machine-svm-using-python/.
[26] Bajaj, A. (2022, March 18). Performance Metrics in Machine Learning [Complete Guide]. Neptune.Ai. https://neptune.ai/blog/performance-metrics-in-machine-learning-complete-guide.
[27] Menon, K. (2021, September 14). An Introduction to the Types Of Machine Learning. Simplilearn.Com. https://www.simplilearn.com/tutorials/machine-learning-tutorial/types-of-machine-learning.
[28] Charbuty, B. and Abdulazeez, A., 2021. Classification Based on Decision Tree Algorithm for Machine Learning. Journal of Applied Science and Technology Trends, 2(01), pp.20-28.
[29] Molnar, C. (2022, March 29). 5.2 Logistic Regression | Interpretable Machine Learning. Christophm. https://christophm.github.io/interpretable-ml-book/logistic.html.
[30] Medium. 2022. Data Quality for Everyday Analysis. [online] Available at: <https://towardsdatascience.com/data-quality-for-everyday-analysis-d3aa1442c31>.
[31] DataRobot AI Cloud. 2022. Machine Learning Model Deployment. [online] Available at: <https://www.datarobot.com/wiki/machine-learning-model-deployment/>.