Project Report

On

Data Analysis and Visualization of Various Datasets

Master of Computer Applications

By

Namrata Singh(162120008)

Under the Guidance of

Dr. Sujoy Das

Session 2016-2019

Department of Mathematics & Computer Applications

Maulana Azad National Institute of Technology Bhopal(MP)

May 2019

Declaration

I, hereby declare that the work presented in this project entitled "**Data Analysis and Visualization**" presented in partial fulfillment for the award of the degree of Master of Computer Applications submitted in the Department Of Mathematics and Computer Applications, Maulana Azad National Institute of Technology ,Bhopal is authentic work carried out from 20th January 2017 to 26th April 2019 under the guidance of **Dr. Sujoy Das**, MANIT Bhopal.

The matter embodied in this project has not been submitted by me or anybody else to any institution for award of any other degree or diploma.

Namrata Singh(162120008)

Counter Signed by:

Supervisor:

Head, Department Of Mathematics and Computer Applications
MANIT, Bhopal.

Acknowledgement

Here, I gladly present this project report on **" Data Analysis and Visualization"** as part of the 6th semester MCA Master in Computer Applications. I take this occasion to thank God, almighty for blessing me with his grace and taking our endeavour to a successful culmination. I extend my sincere and heartfelt thanks to me esteemed guide, **Dr.Sujoy Das** for providing me with the right guidance and advice at the crucial junctures and for showing me the right way. I extend my sincere thanks to my respected head of the department **Dr. Sujoy Das**, for allowing me to use the facilities available.

I am highly indebted to **Dr.Sujoy Das** for his guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project. Last but not the least, I would like to express my gratitude towards my parents & my friends for the support and encouragement they have given me during the course of my work.

Submitted By -

Namrata Singh(162120008)

Certificate

2016-2019

DEPARTMENT OF MATHEMATICS AND COMPUTER APPLICATIONS

MAULANA AZAD NATIONAL INSTITUTE OF TECHNOLOGY, BHOPAL

(M.P.) – 462003

This is to certify that **Namrata Singh(162120008)** have carried out the project work in this report entitled **"Data Analysis and Visualization "** for the award of Master of Computer Application in Maulana Azad National Institute of Technology Bhopal (M.P) - 462003.

This report is the record of the candidates' own work carried out by them under our supervision and guidance. This project work is the part of their Master in Computer Applications in Information Technology curriculum.

Their performance was excellent and we wish them good luck for their future endeavors.

Signature of Project Guide **Signature of Head of Department**

Abstract

The purpose of this project is, that at the end the computer will give the accuracy among the data we have taken, like in iris flower data set the computer will give us the aggregate three different classification i.e to which class the flowers belong, the length of the sepals and petals.

Similarly we have taken 4 more different data sets, wine data set, in which we find the quality of wines good , bad and average.In the third we took heart disease data set which find the no of people suffering to heart disease. Then we have a pokemon data set on which we have used the Visualization technique to find how many pokemons of each type have, which pokemon is the most powerful, and which power does the pokemon have.and lastly we have a data set of a country where we have to analyze the birth date, total population, no. of Youths, children,men,women population and age of each person for each year from 1990 to 2005

To find these results we have applied different machine Learning Algorithms like KNN Algorithm , SVM, Gaussian Naive Bayes and Linear Regression.

CONTENTS

Chapter-1

1.1 Introduction [www.analyticalvidhya.com]

As we all know from the nature, most of creatures have the ability to recognize theobjects in order to identify food or danger.Similarly machines could also recognize objects just like us in the coming days. The Human brain process better informations using charts, grpahs to visualize large amounts of complex data is easier than doing it over spreadsheets and reports. Data visualization is a quick, easy way to convey concepts in a universal manner.In this thesis, the conception of algorithms are introduced like KNN Algorithm, Suport Vector Machine(SVM), Gaussian Naive Algorithm, to find the accuracy among the attributes and which Algorithm gives the most accurate result.

After the project has been settled, the computer should have the ability to aggregate different classifications of the data sets as we have used in this analysis

The aim of the case study is to visualization a system of pattern recognition for the 5 data sets. The data set was collected from an open source website of machine learning. The programming language used in this project was Python.

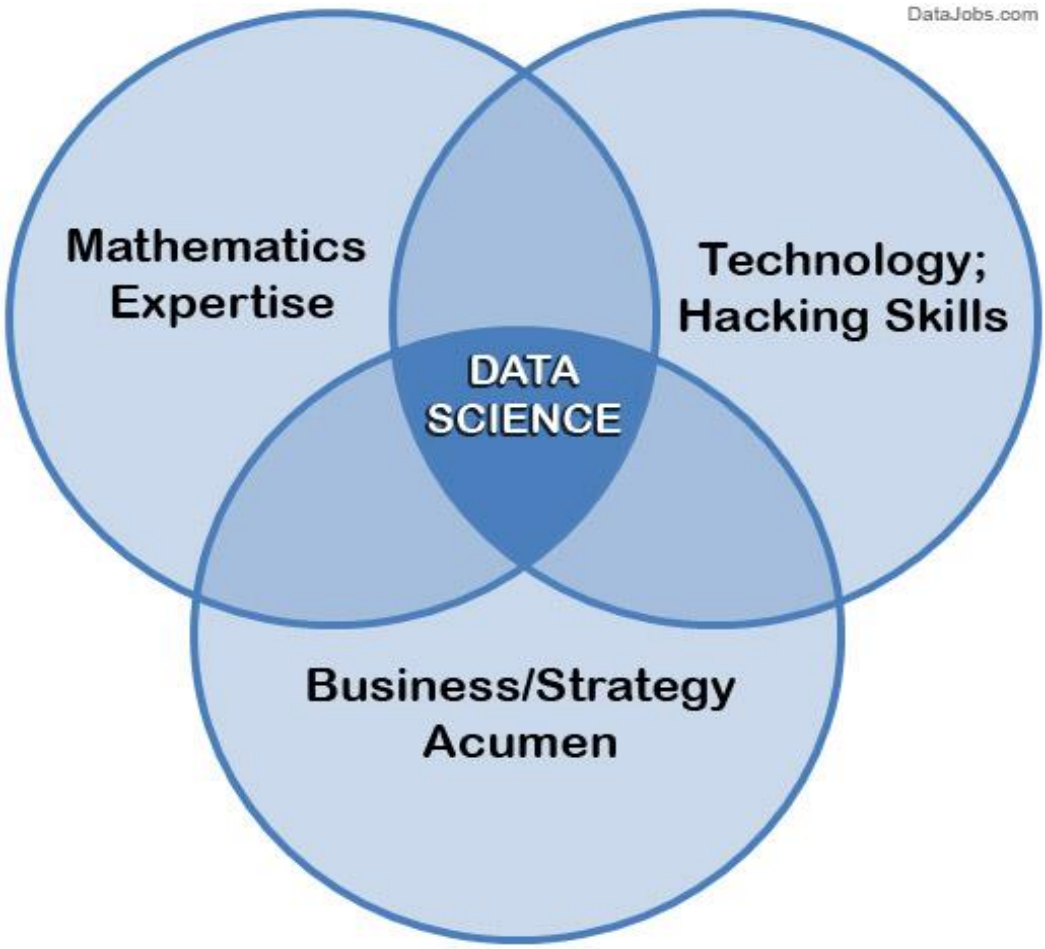
1.2 Data Science

Data science is a combination of data inference, algorithmm development, and technology in order to solve analytically complex problems.

Data is at the core. Collection of raw information, streaming in and stored in enterprise data warehouses and can learn a lot by mining it. We can build more advanced Capabilities with it. Data science is ultimately about using this data in creative ways to generate business value.

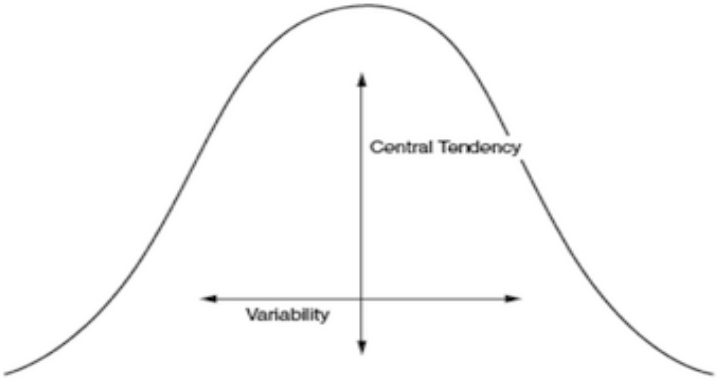
This aspect of data science is all about uncovering findings from data. Diving in at a granular level to mine and understand complex behaviors, trends and inferences. It's about surfacing hidden insight that can help enable companies to make smarter business decisions. For **example: [**www.tutorialspoint.com**]**

- Netflix data mines movie viewing patterns to understand what drives user interest, and uses that to make decisions on which Netflix original series to produce.
- Target identifies what are major customer segments within it's base and the unique shopping behaviors within those segments, which helps to guide messaging to different market audiences.



1.3 Descriptive Analysis

Descriptive statistics are used to describe or summarize data in ways that are meaningful and useful. Descriptive statistics implies a simple quantitative summary of a data set that has been collected. It helps us understand the experiment or data set in detail and tells us everything we need to put the data in perspective.



Measures of central tendency use a single value to describe the center of a data set. The mean, median, and mode are all the three measures of central tendency.

The **mean**, or average, is calculated by finding the sum of the study data and dividing it by the total number of data.

The **mode** is the number that appears most frequently in the set of data.

The **median** is the middle value in a set of data. It is calculated by first listing the data in numerical order then locating the value in the middle of the list.

Standard deviation tells how much deviation is present in the data, i.e. how spread out the numbers are from the mean value.

Minimum value smallest number in the data.

Maximum value largest number in the data. [www.geeksforgeeks.com]

1.4 Data Exploration and Visualization: [www.kaggle.com]

Data Exploration: Data exploration is an instuctive search used by data consumers to form true analysis from the information gathered. Often, data is gathered in a non-rigid or controlled manner in large bulks.

Data exploration is the first step in data analysis and typically involves summarizing the main characteristics of a data set, including its size, accuracy, initial patterns in the data and other attributes.

1.5 Role of Data Exploration:

Before it can conduct analysis on data collected by multiple data sources and stored in data warehouses, an organization must know how many cases are in a data set, what variables are included, how many missing values there are and what general hypotheses the data is likely to support.

Analysts commonly use automated tools such as data **visualization software** for **data exploration** because these tools allow users to quickly and simply view most of the relevant features of a data set.

1.6 Data Visualization: Data visualization is the presentation of data in a pictorial or graphical format. It enables decision makers to see analytics presented visually, so they can grasp difficult concepts or identify new patterns displays data/information in graphical charts, figures and bars.

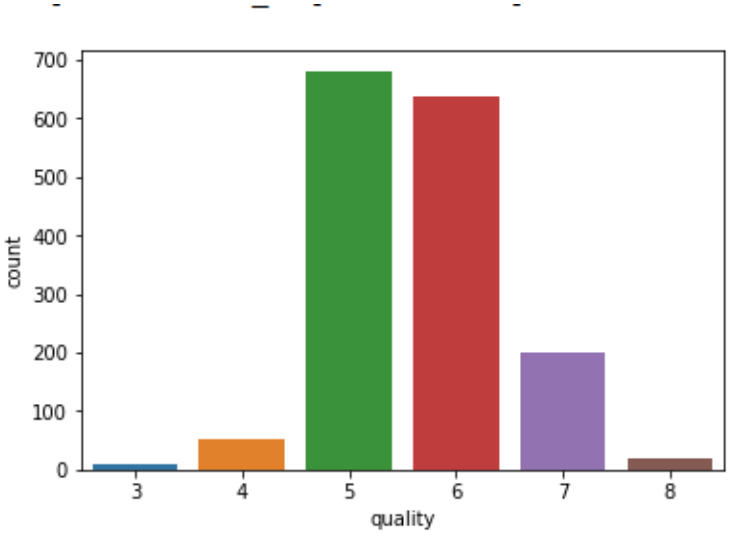
Data Visualization uses two libraries

1.6.1 Matplot : Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits

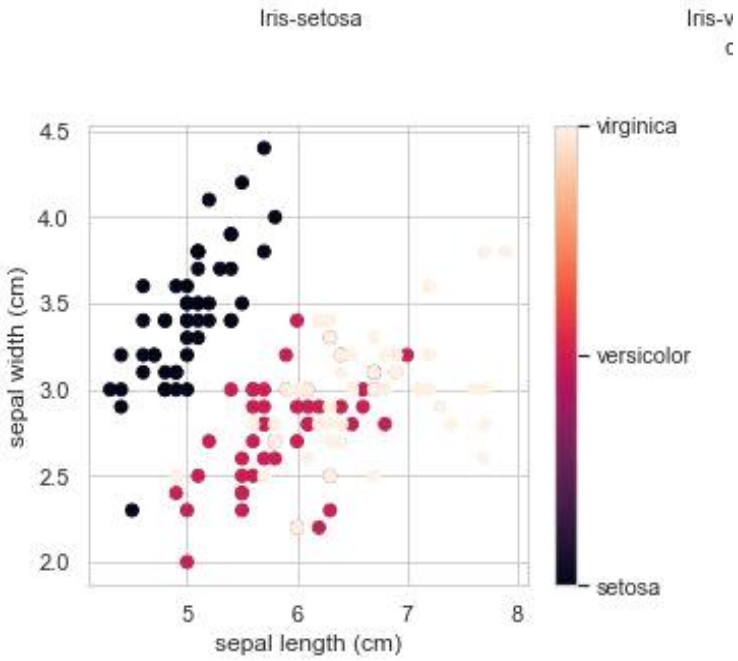
Seaborn: Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

1.6.2 Bar Graph: Bar charts are most commonly used for comparing the quantities of different categories or groups. Values of a category are represented using the bars, and they can be configured with either vertical or horizontal bars, with the length or height of each bar representing the value.

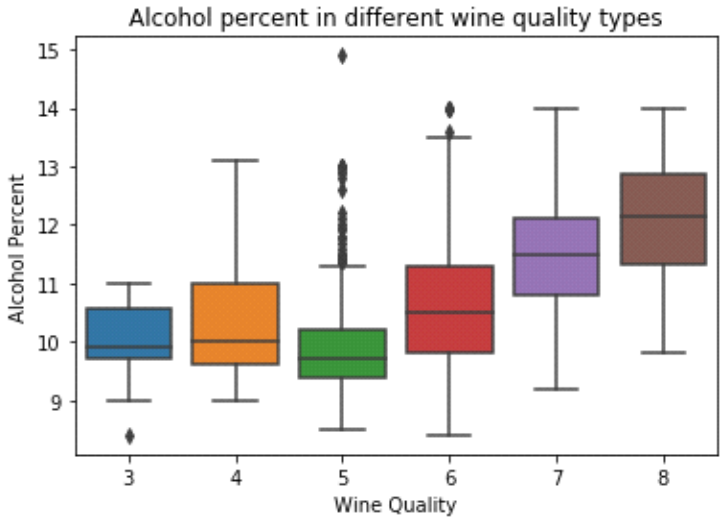
1.6.3 Histogram: A histogram is a data visualisation that uses rectangles with heights proportional to the count and widths equal to the “bin size” or range of small intervals.



1.6.4 Scatter Plot: A scatter plot is a type of plot or mathematical diagram using Cartesian coordinates to display values for typically two variables for a set of data. The scatter plot is simply a set of data points plotted on an x and y axis to represent two sets of variables. The shape those data points create tells the story, most often revealing correlation (positive or negative) in a large amount of data.

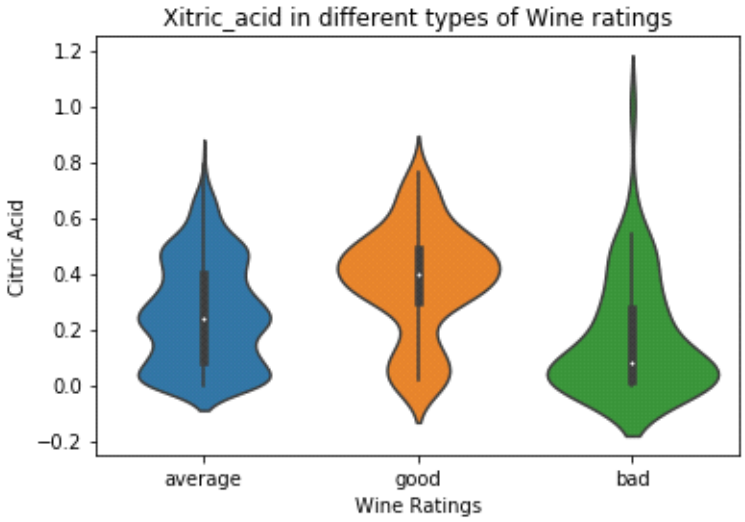


1.6.5 Box Plot: boxplot is a method for graphically depicting groups of numerical data through their quartiles. Box plots may also have lines extending vertically from the boxes (whiskers) indicating variability outside the upper and lower quartiles, hence the terms box-and-whisker plot and box-and-whisker diagram

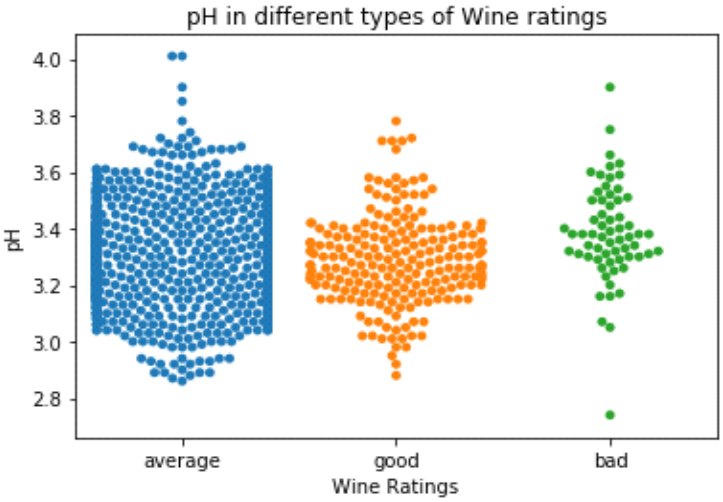


1.6.6 Violin Plot: Violin plot allows to visualize the distribution of a numeric variable for one or several groups. Each 'violin' represents a group or a variable. The shape represents the density estimate of the variable: the more data points in

a specific range, the larger the violin is for that range. It is really close to a boxplot, but allows a deeper understanding of the distribution.



1.6.7 Swarm plot: A swarm plot can be drawn on its own, but it is also a good complement to a box or violin plot in cases where you want to show all observations along with some representation of the underlying distribution.



1.7 Data Analysis and Data Preprocessing [www.codeacademy.com]

Data analysis is a process of inspecting, cleansing, transforming, and modeling data with the goal of discovering useful information, informing conclusions, and supporting decision-making.Data mining is a particular data analysis technique that focuses on modeling and knowledge discovery for predictive rather than purely descriptive purposes, while business intelligence covers data analysis that relies heavily on aggregation, focusing mainly on business information.

There are several phases of data pre processing:

1.8 Data Requirements: The data are necessary as inputs to the analysis, which is specified based upon the requirements of those directing the analysis or customers.The general type of entity upon which the data will be collected is referred to as an experimental unit (e.g., a person or population of people).

1.8.1 Data Collection: Data are collected from a variety of sources. The requirements may be communicated by analysts to custodians of the data, such as information technology personnel within an organization.

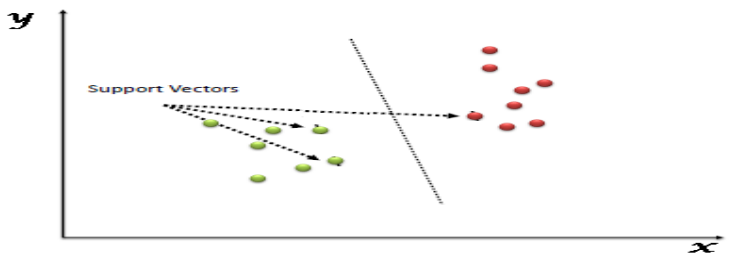
1.8.2 Data Processing: Data initially obtained must be processed or organised for analysis. For instance, these may involve placing data into rows and columns in a table format (i.e., structured data) for further analysis, such as within a spreadsheet or statistical software.

1.8.3 Data Cleaning: Once processed and organised, the data may be incomplete, contain duplicates, or contain errors. The need for data cleaning will arise from problems in the way that data are entered and stored. Data cleaning is the process of preventing and correcting these errors.

1.8.4 Exploratory Data Analysis: Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns,to spot anomalies,to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

1.9 Algorithm Used: [www.realpython.com]

1. Support Vector machine(SVM): "Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well .



Support Vectors are simply the co-ordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line).

2.KNN Algorithm: K nearest neighbors or KNN Algorithm is a simple algorithm which uses the entire dataset in its training phase. Whenever a prediction is required for an unseen data instance, it searches through the entire training dataset for k-most similar instances and the data with the most similar instance is finally returned as the prediction. KNN is often used in search applications where you are looking for similar items, like find items similar to this one.

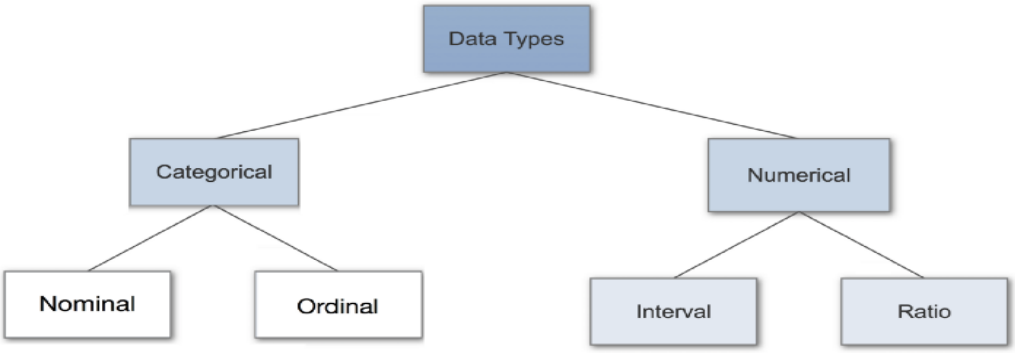
Algorithm suggests that if you're similar to your neighbours, then you are one of them. For example, if apple looks more similar to peach, pear, and cherry (fruits) than monkey, cat or a rat (animals), then most likely apple is a fruit.

3.Gausssian Naive Algorithm: A Gaussian Naive Bayes algorithm is a special type of NB algorithm. It's specifically used when the features have continuous values. It's also assumed that all the features are following a gaussian distribution i.e, normal distribution.

Using the Bayes theorem the naive Bayes classifier works. The naive Bayes classifier assumes all the features are independent to each other. Even if the features depend on each other or upon the existence of the other features.

1.10 Data Types [www.kaggle.com]

Data Types are an important concept of statistics, which needs to be understood, to correctly apply statistical measurements to your data and therefore to correctly conclude certain assumptions about it. To do proper exploratory data analysis (EDA), which is one of the most underestimated parts of a machine learning project.



1.10.1 Categorical Data: Categorical data represents characteristics. Therefore it can represent things like a person's gender, language etc. Categorical data can also take on numerical values (Example: 1 for female and 0 for male).

1.10.2 Nomial Data: Nominal values represent discrete units and are used to label variables, that have no quantitative value. Just think of them as „labels". Note that nominal data that has no order. Therefore if you would change the order of its values, the meaning would not change.

1.10.3 Ordinal Data: Ordinal values represent discrete and ordered units. It is therefore nearly the same as nominal data, except that it's ordering matters.

Numerical:

1.10.4 Discrete Data: Discrete data are those if there values are distinct and separate or if the data can only take on certain values. This type of data can't be measured but it can be counted. It basically represents information that can be categorized into a classification.

1.10.5 Conitnuous Data: Continuous Data represents measurements and therefore their values can't be counted but they can be measured. example the height of a person, which you can describe by using intervals on the real number line.

1.10.6 Interval Data: Interval values represent ordered units that have the same difference. Therefore we speak of interval data when we have a variable that contains numeric values that are ordered and where we know the exact differences between the values. eg temperature of a given place.

1.10.7 Ratio Data: Ratio values are also ordered units that have the same difference. Ratio values are the same as interval values, with the difference that they do have an absolute zero. examples are height, weight, length etc.

Chapter-2

Data Visualization on Wine Ratings Prediction

2.1 Introduction [www.codeacademmy.com]

We observed the key factors that determine and affects the quality of the red wine. Wine quality is ultimately a subjective measure. The ordered factor *'quality'* was not very helpful and to overcome this, so we created another variable called *'rating'*.The usage of this analysis will help to understand whether by modifying the variables, it is possible to increase the quality of the wine on the market. If you can control your variables, then you can predict the quality of your wine and obtain more profits.

2.2 Dataset

Name- Red Wine Quality Dataset

Source- UCI Machine Learning Repository

Input Variables

Fixed Acidity
Volatile Acidity
Citric Acid
Residual Sugar
Chlorides
Free Sulphur Dioxide
Total Sulfur Dioxide
Density
Ph
Sulphate
Alcohol

Number of Observations: 1599
Output Variables: Quality (score between 0 and 10)
Number of Attributes/Variables: 12
Missing Values: N/A

2.3 Setting up the development environment by importing required libraries and modules:

- *Numpy:* It will provide the support for efficient numerical computation.
- *Pandas:* It is convenient library that supports dataframes. Working with pandas will bring ease
- in many crucial data operations.
- *Matplotlib:* It provides a MATLAB-like plotting framework.
- *Seaborn:* It is a visualization library based on matplotlib which provides a high-level interface for drawing attractive statistical graphics.

- *Bokeh:* It is a interactive visualization library that targets modern web browsers for presentation.
- *Statsmodel:* It provides functions and classes for statistical tests and models.
- *Sklearn:* It is python library for data mining, data analysis and machine learning.

2.4 Loading the Red Wine dataset

- Lets read the red wine data set from the *'UCI Machine Learning Repository'*.
- Here, we can use the *read_csv()* from the *pandas* library to load data into dataframe from the remote url.

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

2.5 Exploring the Red Wine dataset:

The summary of Red Wine dataset looks perfect, there is no visible abnormality in data (invalid/negative values).
All the data seems to be in range (with different scales, which needs standardization).

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000

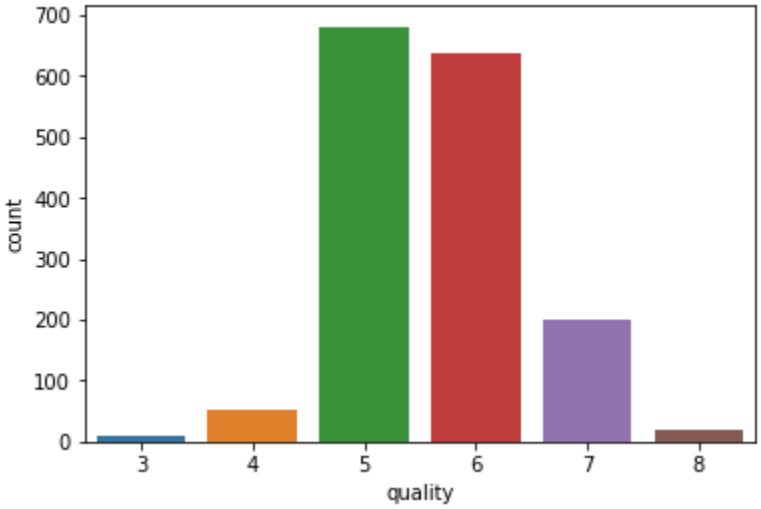
Missing Value

We will look at how we can identify and mark values as missing.We can use plots and summary statistics to help identify missing or corrupt data.We can load the dataset as a Pandas DataFrame and print summary statistics on each attribute. So in this dataset we filter the missing value with no errors.

```
fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
free sulfur dioxide     0
total sulfur dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality                 0
dtype: int64
```

- **2.6 Data Visualization**

2.6.1 Histogram



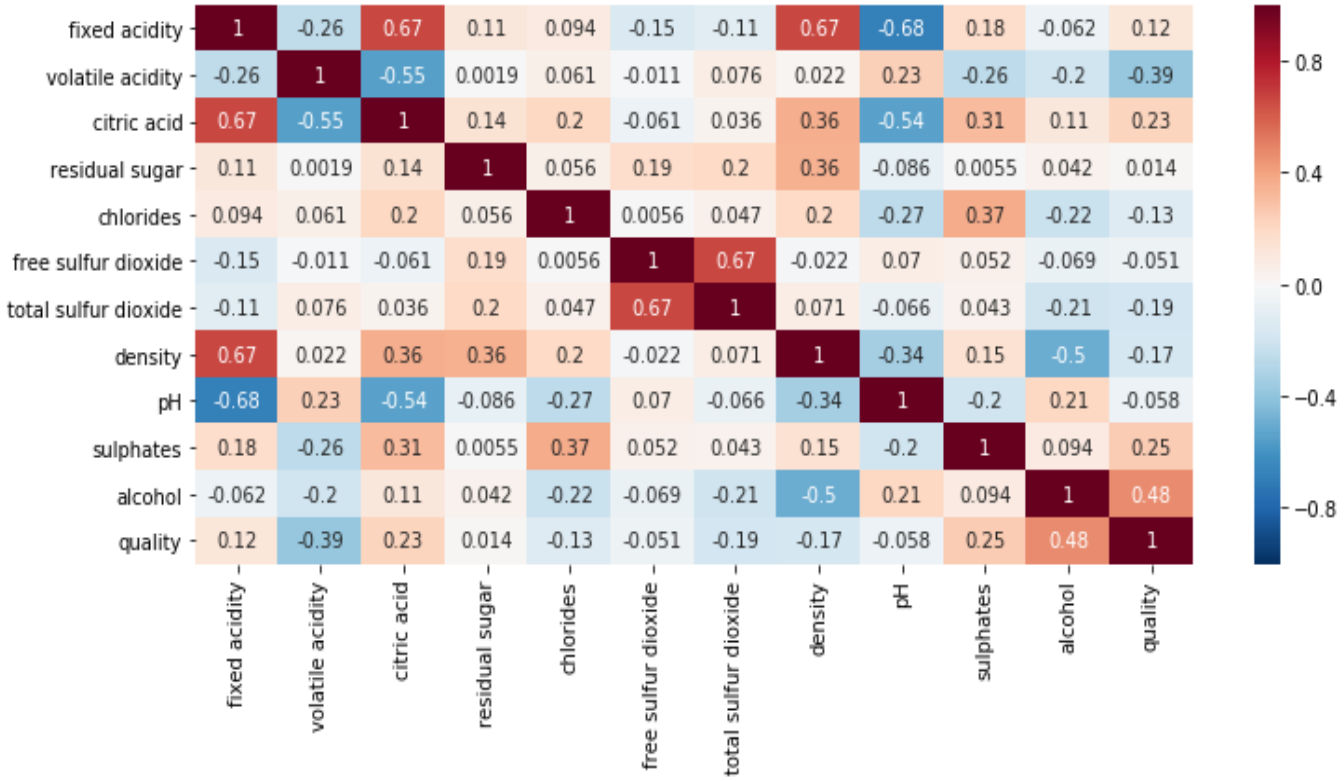
The above distribution shows the range for response variable (*quality*) is between 3 to 8.
Let's create a new discreet, categorical response variable/feature ('*rating*') from existing '*quality*' variable.
i.e. bad: 1-4
average: 5-6
good: 7-10

```
average    1319
good        217
bad          63
Name: rating, dtype: int64
```

rating	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
average	8.254284	0.538560	0.258264	2.503867	0.088973	16.368461	48.946929	0.996867	3.311296	0.647263	10.252717	5.483700
bad	7.871429	0.724206	0.173651	2.684921	0.095730	12.063492	34.444444	0.996689	3.384127	0.592222	10.215873	3.841270
good	8.847005	0.405530	0.376498	2.708756	0.075912	13.981567	34.889401	0.996030	3.288802	0.743456	11.518049	7.082949

- **2.6.2 Heatmap**

  It's check the correlation between the target variable and predictor variables.



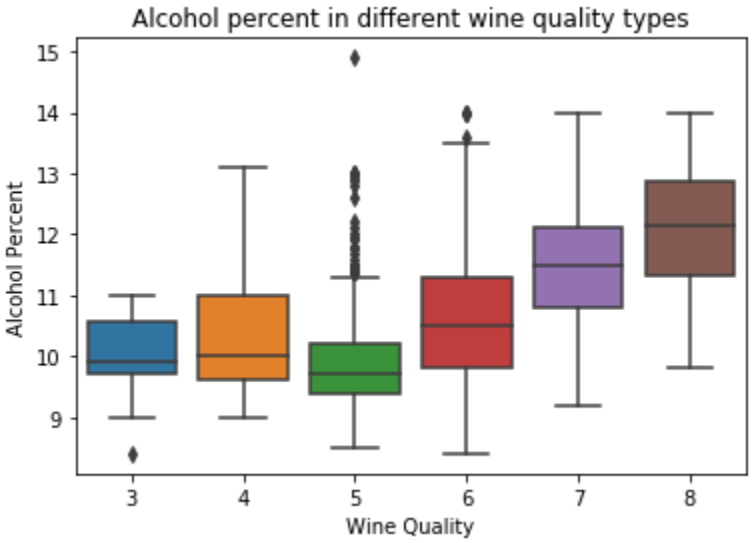
We can observe that, the *'alcohol, sulphates, citric_acid & fixed_acidity'* have maximum corelation with response variable '*quality*'.
This means that, they need to be further analysed for detailed pattern and corelation exploration.
Hence, we will use only these 4 variables in our future analysis.

```
quality                1.000000
alcohol                0.476166
sulphates              0.251397
citric acid            0.226373
fixed acidity          0.124052
residual sugar         0.013732
free sulfur dioxide   -0.050656
pH                    -0.057731
chlorides             -0.128907
density               -0.174919
total sulfur dioxide  -0.185100
volatile acidity      -0.390558
Name: quality, dtype: float64
```

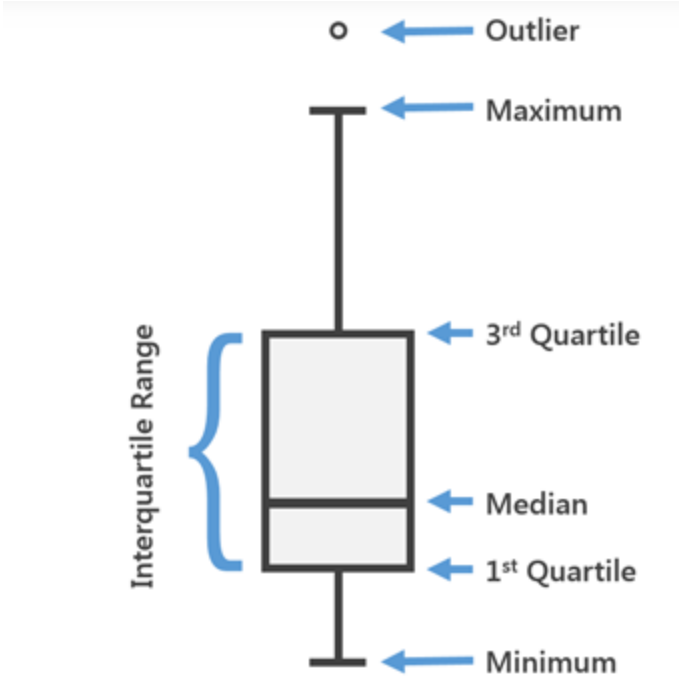
- ## 2.6.3 Boxplot

  - Analysis of alcohol percentage with wine quality.

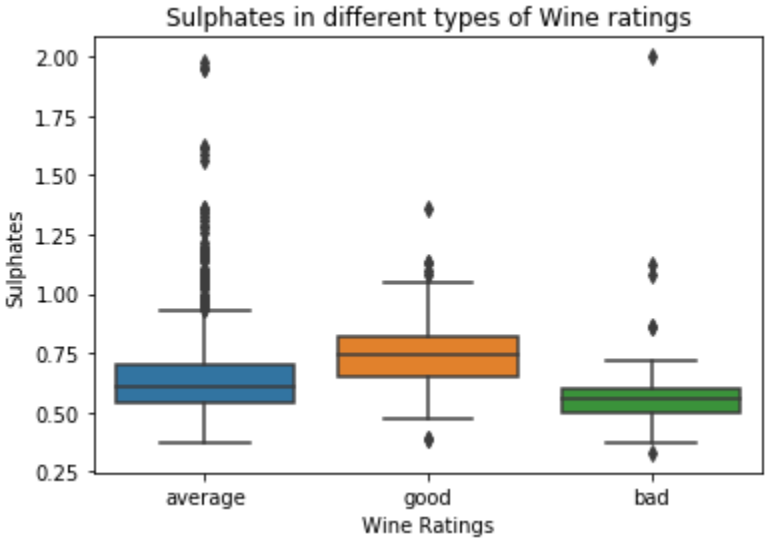
In this box plot method we are graphically depicting groups of numerical data through their quartiles. The box extends from the Q1 to Q3 quartile values of the data, with a line at the median (Q2).



The straight lines at the maximum and minimum are also called as **whiskers**. Points outside of whiskers will be inferred as an outliers. The box plot gives us a representation of 25th, 50th ,75th quartiles.From box plot we can also see the Interquartile range(IQR) where maximum details of the data will be present. It also gives us a clear overview of outlier points in the data.



Analysis of sulphates & wine ratings

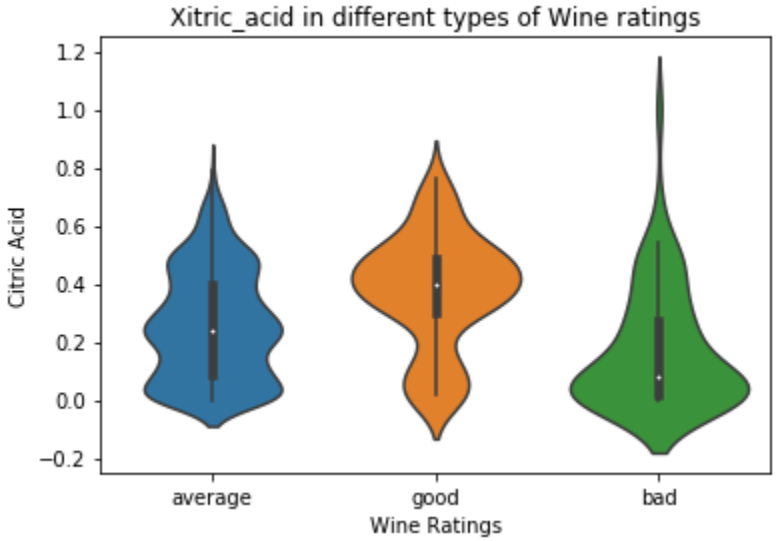


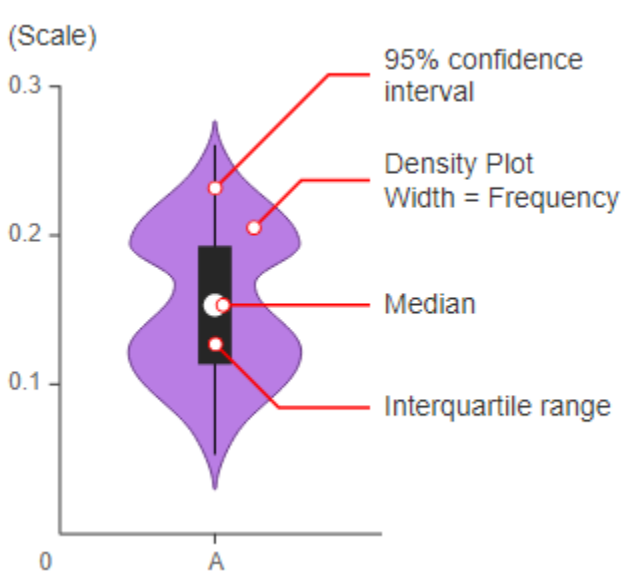
The whiskers extend from the edges of box to show the range of the data. The **matplotlib** axes to be used by **boxplot** to visualize the distribution of values within each column. Here is a boxplot representing trials of observations of a uniform random variable.

- ## 2.6.4 Violin Plot

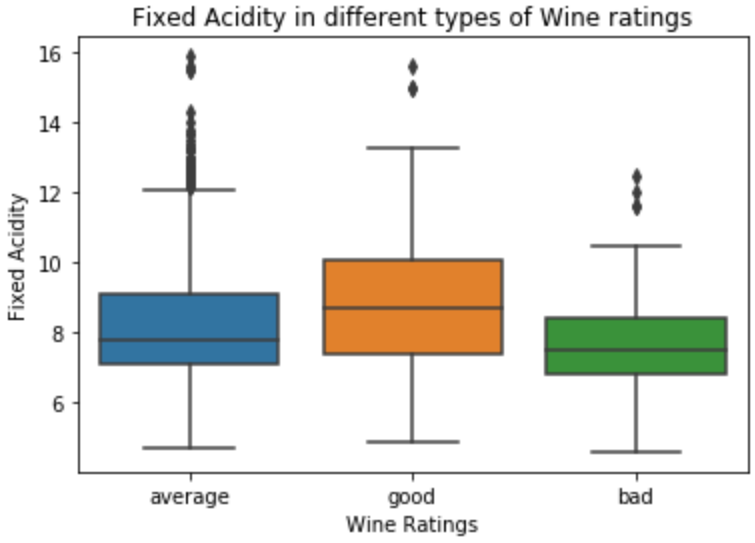
Analysis of Citric Acid & wine ratings

The violin plots can be inferred as a combination of Box plot at the middle and distribution plots (Kernel Density Estimation) on both side of the data. This can give us the details of distribution like whether the distribution is mutimodal, Skewness etc. It also give us the useful info like 95% confidence interval. The below image help us grasp some important parts from a violin plot.





Here we are finding the quality of wine by the ratings which is based on good, average and bad prediction.Using the box plot, as mentioned above we can see how much of data is present in 1st quartile and how much points are outliers etc. From the above plot for the **class 1** we can see that there are very few/no data is present between median and the 1st quartile. Also there are more number of outlier points for **class 1** in feature **axil_nodes**.

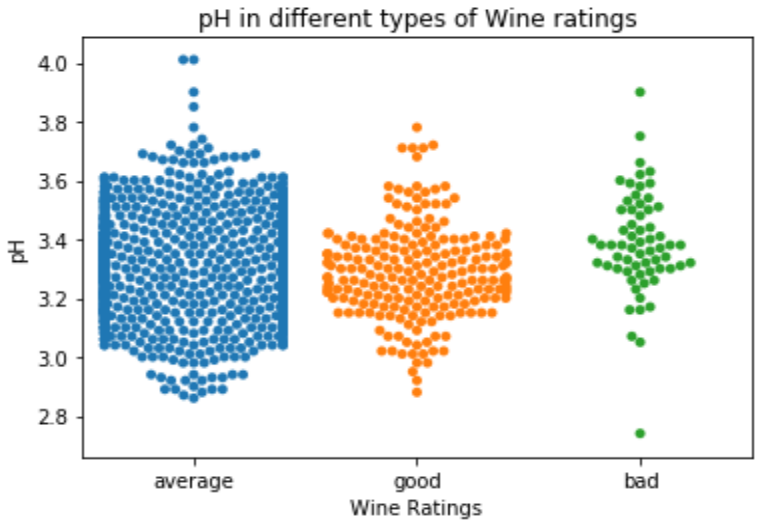


From the above violin plot we can infer that median of both the classes are around 63 and also the maximum number of persons with class 2 has op_yr value as 65 whereas for persons in class1, the maximum value is around 60.The 3rd quartile to median has lesser points than the median to 1st quartile and so on.

- ## 2.6.5 Swarm Plot

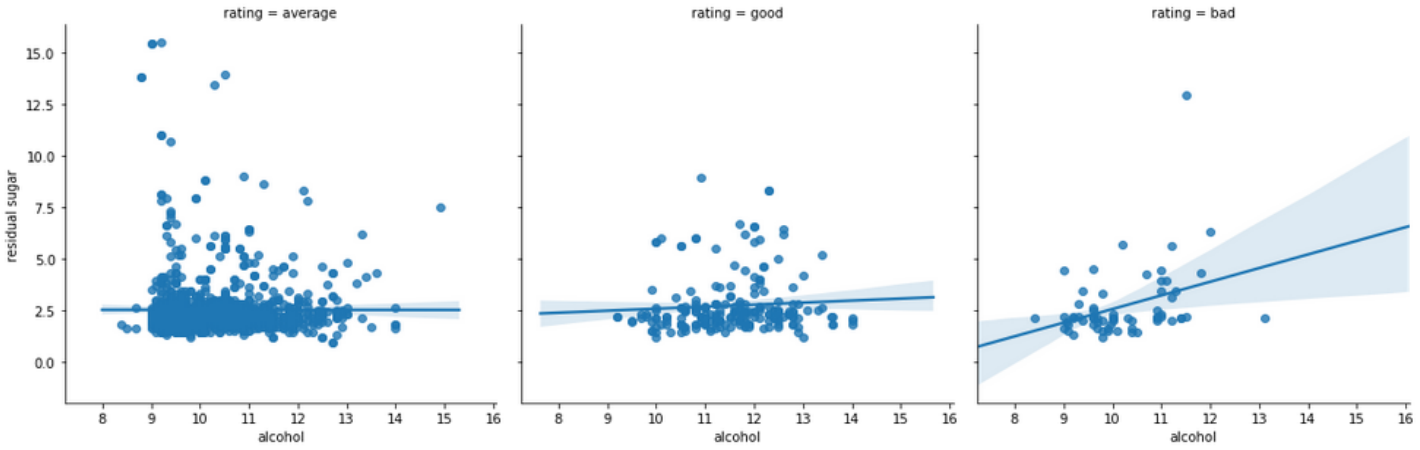
Analysis of pH & wine ratings

As you can see, one potential problem with a strip plot is that you could have very dense grouping of data points, leading to data points being plotted over top of one another on the chart and obscuring the data. Each plotting defines the density of each attribute data with higher or lower level of dots. The above wine quality vs alcohol content regression model's result shows that, the minimum value for quality is 1.87 and there will be increment by single unit for wine quality for every change of 0.360842 alcohol units.



2.7 Linear Regression

Below graphs for different quality ratings shows a linear regression between residual_sugar & alcohol in red wine. This analysis can help in manufacturing the good quality wine with continuous monitoring and contrilling the alcohol and residual sugar content of the red wine

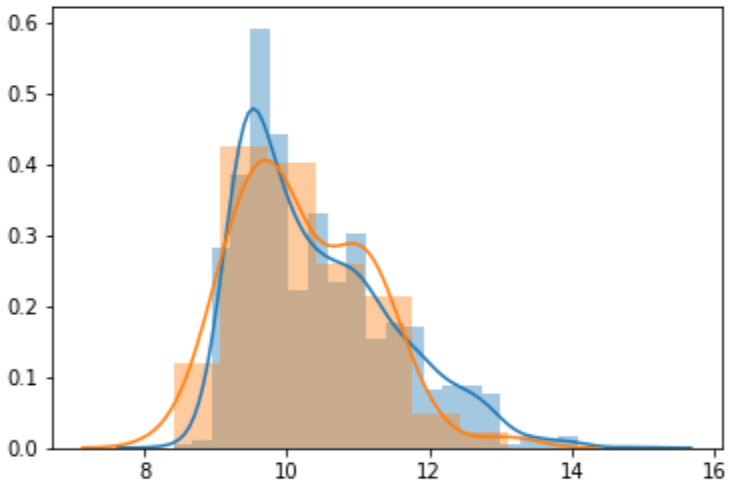


The linear regression plots above for different wine quality ratings (bad, average & good) shows the regression between alcohol and residual sugar content of the red wine.We can observe from the trendline that, for good and average wine types the residual sugar content remains almost constant irrespective of alcohol content value. Whereas for bad quality wine, the residual sugar content increases gradually with the increase in alcohol content.

- ## 2.8 Classification

Classification using Statsmodel

We will use statsmodel for this logistic regression analysis of predicting good wine quality (>4).Let's create a new categorical variable/column (rate_code) with two possible values (good = 1 & bad = 0).



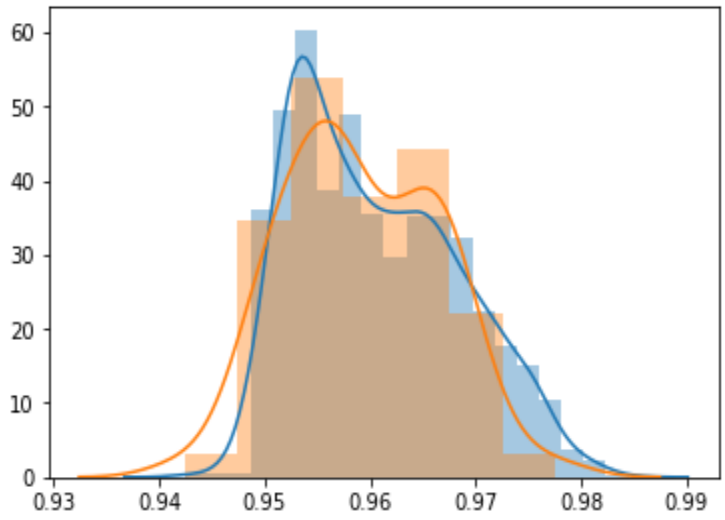
- The above plot shows the higher probability for red wine quality will be good if alcohol percentage is more than equal to 12, whereas the same probability reduces as alcohol percentage decreases.

Model:	Logit	No. Iterations:	8.0000
Dependent Variable:	rate_code	Pseudo R-squared:	0.005
Date:	2018-05-26 22:53	AIC:	532.3386
No. Observations:	1599	BIC:	543.0928
Df Model:	1	Log-Likelihood:	-264.17
Df Residuals:	1597	LL-Null:	-265.48
Converged:	1.0000	Scale:	1.0000

	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
Intercept	1.0456	1.3628	0.7673	0.4429	-1.6253	3.7166
alcohol	0.2082	0.1327	1.5685	0.1168	-0.0519	0.4683

The above distribution plot displays the overlapped outcomes for the good and bad quality plots of the red wine.We can observe that the precision for the good wine prediction is almost 96% accurate, where as for bad wine its only 4%, which is not good. But overall there is 92% average precision in wine quality rate prediction.

2.9 Classification using Sklearn's LogisticRegression



From the above plot, we can see that we created a distribution plot on the feature ‘Age’(input feature) and we used different colors for the Survival status(dependent variable/output) as it is the class to be predicted and we can see there is a huge overlap between their PDFs. The sharp block like structures are histograms and the smoothened curve is called Probability density function(PDF). The pdf of a curve can help us to identify the underlying distribution of that feature which is one major takeaway from Data visualization/EDA

2.9.1 Sklearn's LogisticRegression

	precision	recall	f1-score	support
0.0	0.04	0.32	0.07	63
1.0	0.96	0.69	0.80	1536
micro avg	0.67	0.67	0.67	1599
macro avg	0.50	0.50	0.44	1599
weighted avg	0.92	0.67	0.77	1599

The accuracy matrix for sklearn's logistics regression model for red wine quality prediction shows the overall 92% precision which is similar to previous statsmodel's average precision.Also the precision for good wine (1) prediction is almost 96%.The code is pretty straight forward, We will be using bar function from Matplotlib to achieve it. The below code will give you a bar plot, where the position of bars are mentioned by values in x and length/height of the bar as values in y.But the precision is almost 0% for the bad type of wine (0) with sklearn's linear regression model. Which is not a good sign for the analysis.

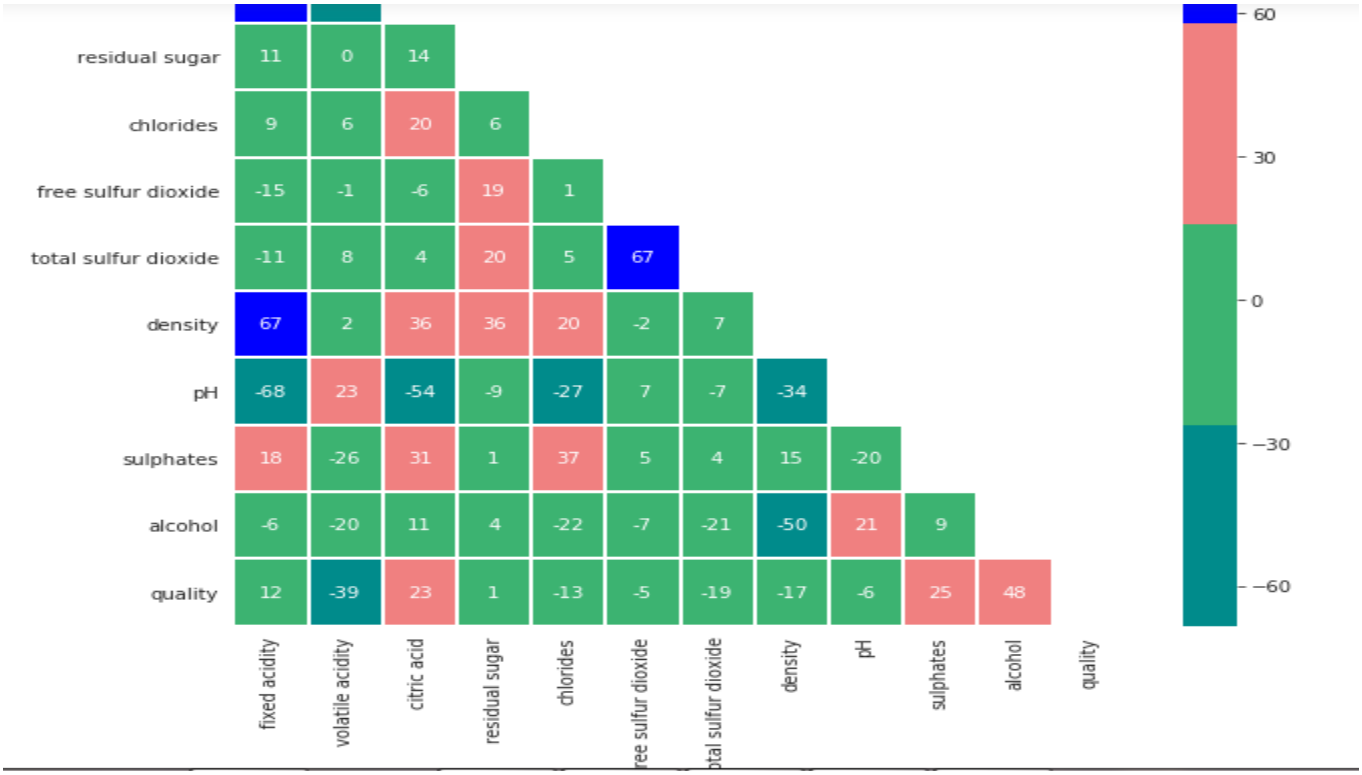
2.9.2 Classification using Sklearn's Random Forest Classifier

	precision	recall	f1-score	support
0.0	0.67	0.03	0.06	63
1.0	0.96	1.00	0.98	1536
micro avg	0.96	0.96	0.96	1599
macro avg	0.81	0.52	0.52	1599
weighted avg	0.95	0.96	0.94	1599

Here, with the accuracy matrix for sklearn's random forest classifier model for the prediction of red wine quality, we can observe that the values have been improved significantly.The precision for the prediction of bad quality wine (0) is almost 100% where as the precision for prediction of good quality wine (1) is approximately 96%.This sklearn's random forest classifier model also has the overall precision around 96%, which is far better than the previous two models (i.e. statsmodel and sklearn's linear regression model)

▪ 2.10 Correlation

Correlation matrix plot, to view the correlations between different variables in a dataframe. Let's take a look at a positive correlation. Numpy implements a `corrcoef()` function that returns a matrix of correlations of x with x, x with y, y with x and y with y. We're interested in the values of correlation of x with y (so position (1, 0) or (0, 1)).



Chapter-3

Data Analysis & Visualization on Iris Dataset

3.1 Introduction

The Iris flower data set or Fisher's Iris data set is a multivariate data set introduced by the British statistician and biologist Ronald Fisher in his 1936 paper "The use of multiple measurements in taxonomic problems" as an example of linear discriminant analysis. This famous iris data set gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are Iris setosa, versicolor, and virginica.

3.2 About Iris dataset

The iris dataset contains the following data

50 samples of 3 different species of iris (150 samples total)

Measurements: sepal length, sepal width, petal length, petal width

The format for the data: (sepal length, sepal width, petal length, petal width)

3.3 Data Import

Import the iris.csv using the panda library and examine first few rows of data. Each row is an **observation** (also known as: sample, example, instance, record)

Each column is a **feature** (also known as: predictor, attribute, independent variable, input, regressor, covariate)

3.4 Display Iris Dataset

This data sets consists of 3 different types of irises' (Setosa, Versicolour, and Virginica) petal and sepal length, stored in a 150x4 numpy.ndarray

The rows being the samples and the columns being: Sepal Length, Sepal Width, Petal Length and Petal Width. The data set consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters.

The use of this data set in cluster analysis however is not common, since the data set only contains two clusters with rather obvious separation. One of the clusters contains Iris setosa, while the other cluster contains both Iris virginica and Iris versicolor and is not separable without the species information Fisher used. This makes the data set a good example to explain the difference between supervised and unsupervised techniques in data mining: Fisher's linear discriminant model can only be obtained when the object species are known: class labels and clusters are not necessarily the same.

```
(150, 5)
    sepal-length  sepal-width  petal-length  petal-width        class
0            5.1          3.5           1.4          0.2  Iris-setosa
1            4.9          3.0           1.4          0.2  Iris-setosa
2            4.7          3.2           1.3          0.2  Iris-setosa
3            4.6          3.1           1.5          0.2  Iris-setosa
4            5.0          3.6           1.4          0.2  Iris-setosa
5            5.4          3.9           1.7          0.4  Iris-setosa
6            4.6          3.4           1.4          0.3  Iris-setosa
7            5.0          3.4           1.5          0.2  Iris-setosa
8            4.4          2.9           1.4          0.2  Iris-setosa
9            4.9          3.1           1.5          0.1  Iris-setosa
10           5.4          3.7           1.5          0.2  Iris-setosa
11           4.8          3.4           1.6          0.2  Iris-setosa
12           4.8          3.0           1.4          0.1  Iris-setosa
13           4.3          3.0           1.1          0.1  Iris-setosa
14           5.8          4.0           1.2          0.2  Iris-setosa
15           5.7          4.4           1.5          0.4  Iris-setosa
16           5.4          3.9           1.3          0.4  Iris-setosa
```

3.5 Requirements for working with data in scikit-learn

1. Features and response are **separate objects**
   - In this case, data and target are separate
2. Features and response should be **numeric**
   - In this case, features and response are numeric with the matrix dimension of 150 x 4
3. Features and response should be **NumPy arrays**
   - The iris dataset contains NumPy arrays already
   - For other dataset, by loading them into NumPy
4. Features and response should have **specific shapes**
   - 150 x 4 for whole dataset
   - 150 x 1 for examples
   - 4 x 1 for features
   - you can convert the matrix accordingly using np.tile(a, [4, 1]), where a is the matrix and [4, 1] is the intended matrix dimensionality.

3.6 Algorithms

3.6.1 SUPPORT VECTOR MACHINE (SVM CLASSIFIER) IMPLEMENATION

The main objective is to segregate the given dataset in the best possible way. The distance between the either nearest points is known as the margin. The objective is to select a hyperplane

with the maximum possible margin between support vectors in the given dataset. In this dataset we are finding the accuracy of all datasets that which algorithm predict the accurate result and this SVM Algo will find that in which class this flowers belongs.

```
0.96
[[14  0  0]
 [ 0 16  2]
 [ 0  0 18]]
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        14
Iris-versicolor       1.00      0.89      0.94        18
 Iris-virginica       0.90      1.00      0.95        18

      micro avg       0.96      0.96      0.96        50
      macro avg       0.97      0.96      0.96        50
   weighted avg       0.96      0.96      0.96        50
```

3.6.2 KNN Algorithm

Basically, the result of clustering algorithm is to find the same classification of different data in the whole data sets. For example, the data set contains monkey, lion, banana, apple, four different data units. After clustering, these four data will be divided into two main sections. One section includes monkey and lion representing the class of animals. The other section includes apple and banana, this section representing the class of fruits.

```
0.92
[[14  0  0]
 [ 0 17  1]
 [ 0  3 15]]
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        14
Iris-versicolor       0.85      0.94      0.89        18
 Iris-virginica       0.94      0.83      0.88        18

      micro avg       0.92      0.92      0.92        50
      macro avg       0.93      0.93      0.93        50
   weighted avg       0.92      0.92      0.92        50
```

3.6.3 Gaussian Algorithm

This time I want to talk about the Gaussian Naive Bayes algorithm, which is a simple classification algorithm which is based on the Bayes' theorem.As we can see the classifier is very fast, even it is no big data set, and the accuracy is perfect. Of course with other permutations and other ratio between training and testing set there will be other results.

```
0.92
[[14  0  0]
 [ 0 17  1]
 [ 0  3 15]]
```

	precision	recall	f1-score	support
Iris-setosa	1.00	1.00	1.00	14
Iris-versicolor	0.85	0.94	0.89	18
Iris-virginica	0.94	0.83	0.88	18
micro avg	0.92	0.92	0.92	50
macro avg	0.93	0.93	0.93	50
weighted avg	0.92	0.92	0.92	50

Now as we can that in these three algorithms, the SVM is giving the more accurate result than the other algorithm.

3.7 Calculating Statistics and Basic Visualization

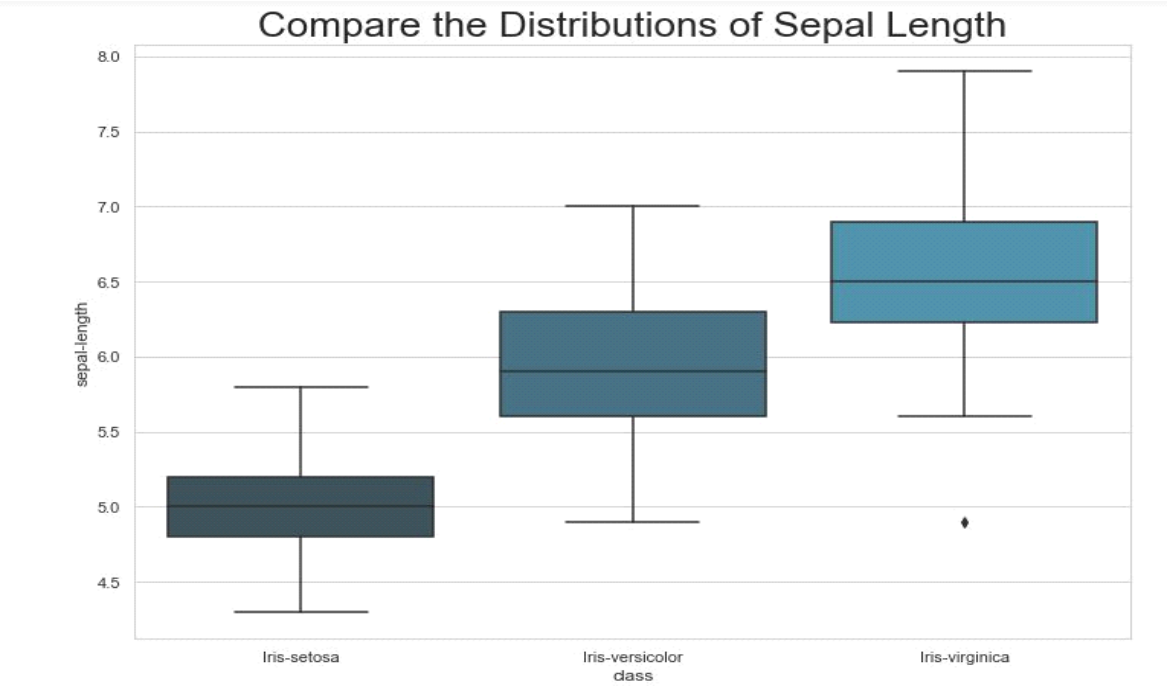
Min, Max, Mean, Median and Standard Deviation

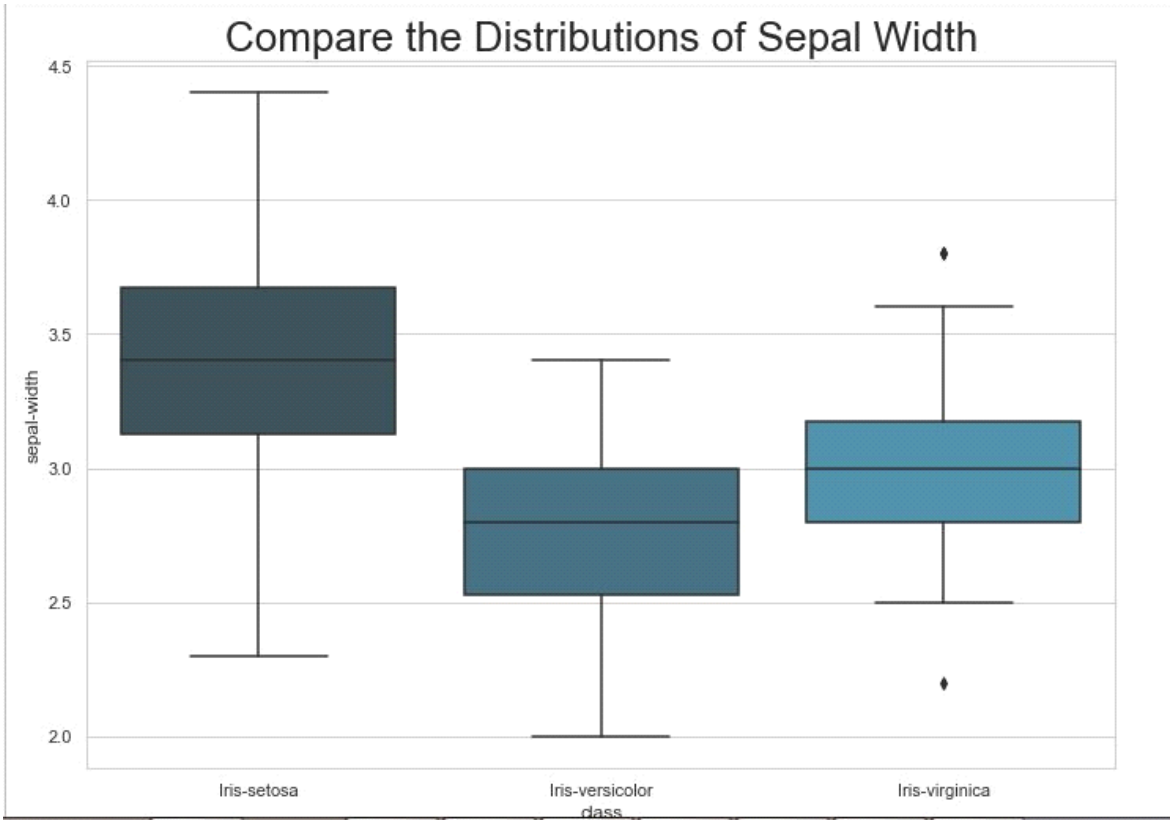
	sepal-length	sepal-width	petal-length	petal-width
count	50.00000	50.000000	50.000000	50.00000
mean	5.00600	3.418000	1.464000	0.24400
std	0.35249	0.381024	0.173511	0.10721
min	4.30000	2.300000	1.000000	0.10000
25%	4.80000	3.125000	1.400000	0.20000
50%	5.00000	3.400000	1.500000	0.20000
75%	5.20000	3.675000	1.575000	0.30000
max	5.80000	4.400000	1.900000	0.60000
	sepal-length	sepal-width	petal-length	petal-width
count	50.000000	50.000000	50.000000	50.000000
mean	5.936000	2.770000	4.260000	1.326000
std	0.516171	0.313798	0.469911	0.197753
min	4.900000	2.000000	3.000000	1.000000
25%	5.600000	2.525000	4.000000	1.200000
50%	5.900000	2.800000	4.350000	1.300000
75%	6.300000	3.000000	4.600000	1.500000
max	7.000000	3.400000	5.100000	1.800000

3.8 Data Visualization

3.8.1 Boxplots

The boxplot is a quick way of visually summarizing one or more groups of numerical data through their quartiles. Comparing the distributions of: Sepal Length, Sepal width. It is displaying the distribution of data based on a five number summary ("minimum", first quartile (Q1), median, third quartile (Q3), and "maximum").

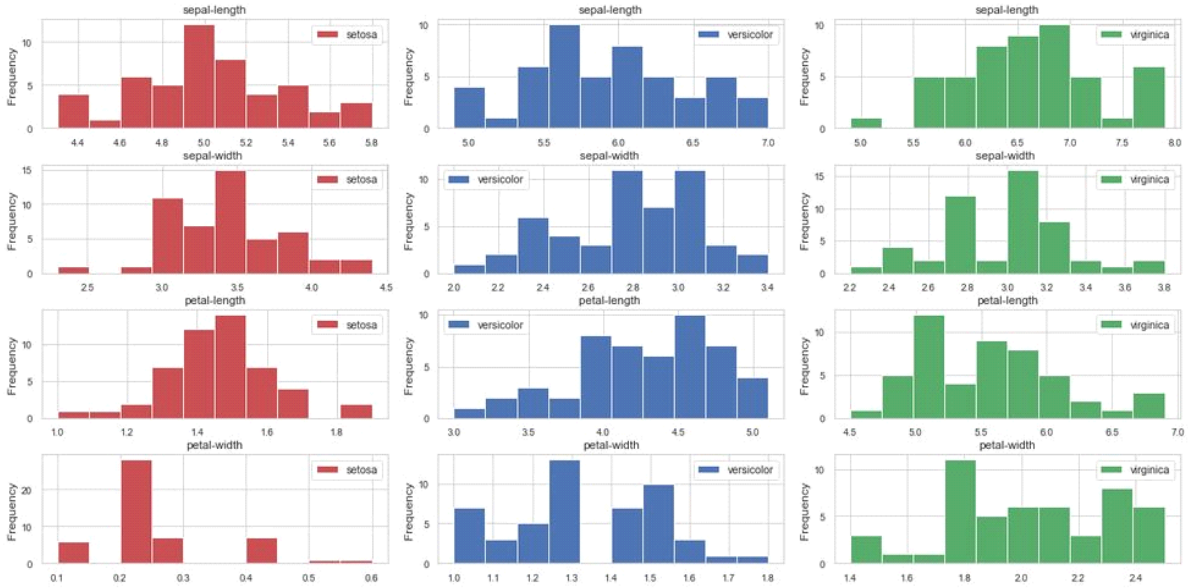




median (Q2/50th Percentile): the middle value of the dataset. **first quartile (Q1/25th Percentile)**: the middle number between the smallest number (not the "minimum") and the median of the dataset. **third quartile (Q3/75th Percentile)**: the middle value between the median and the highest value (not the "maximum") of the dataset.**whiskers (shown in black line). outliers (shown as horizontal line).The dark blue color quartile shows species of Setosa with minimum outliers with range below 4.5 to 6.0, The medium blue color quartile shows species of Versicolor with average outliers with range 5.0 to 7.0, The medium blue color quartile shows species of Virginica with minimum outliers with range below 6.0 to 8.0, which is based on the modules of datasets.**

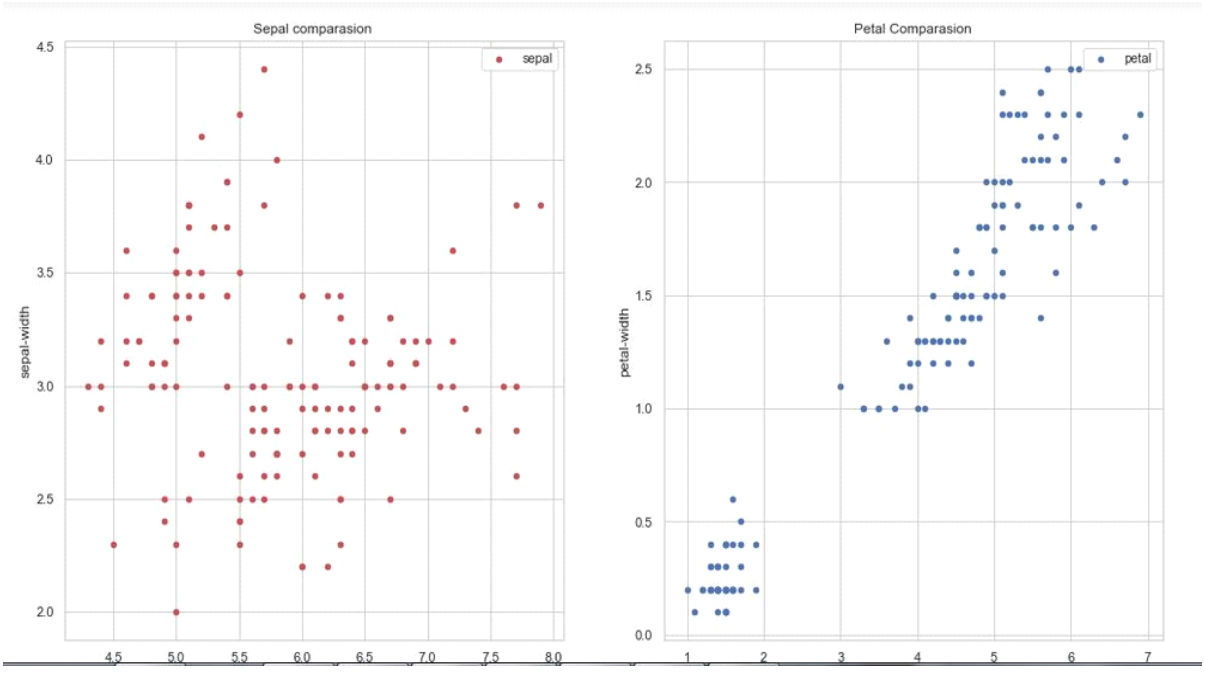
3.8.2 Histogram

The best part about this type of plot is that it just takes a single command to draw the plots and it provides so much information in return. Just use `dataset.hist()`.



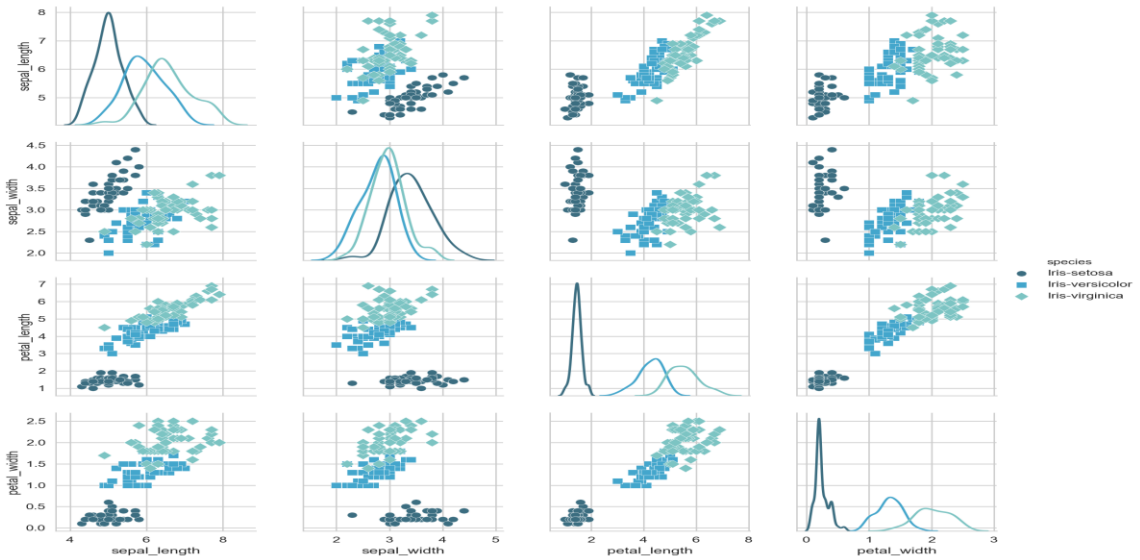
3.8.3 Scatterplots

Here we can use to variables to show that there is distinct difference in sizes between the species. Firstly, we look at the Petal width and Petal length across the species. Is it clear to see that the iris Setosa has a significantly smaller petal width and petal length than the other two species. This difference occurs again for the Petal width and Sepal length. And in both cases we can see that the Iris Viginica is the largest species.



3.8.4 Pairplot

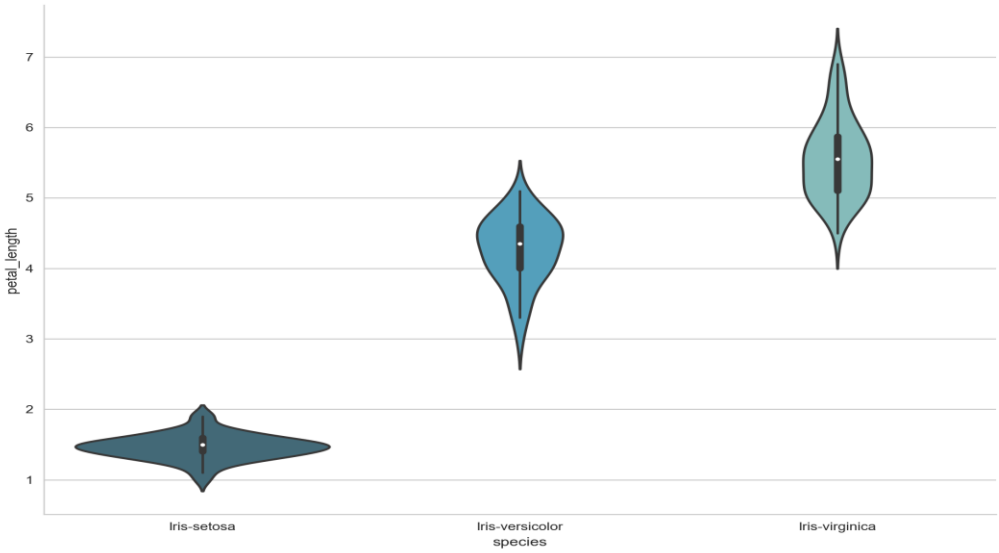
This chart enables us to quickly see the relationships between variables across multiple dimensions using scatterplots and histograms.



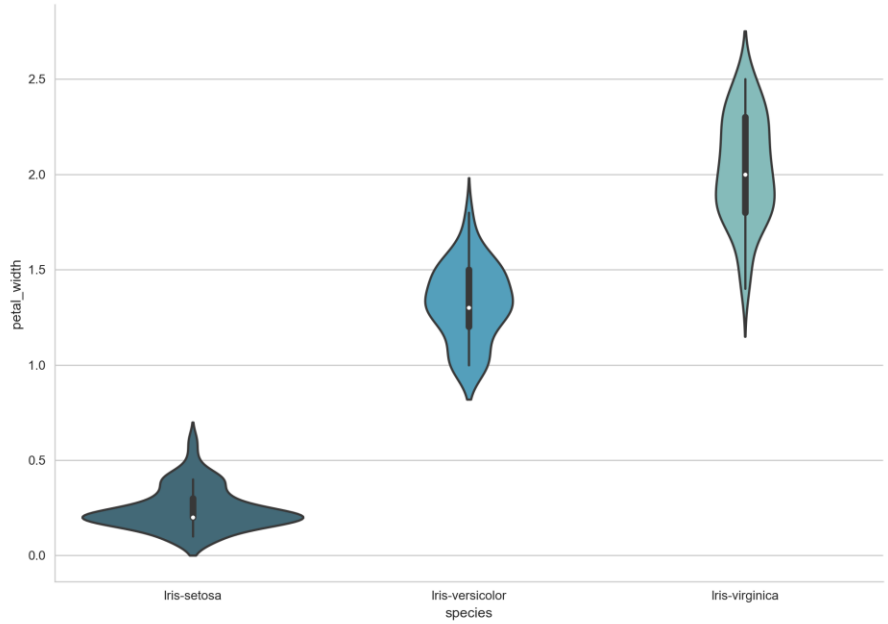
3.8.5 Violin Plot

A voilin plot is used to visualise the distribution of the data and its probability density. The thick black bar in the center represents the interquartile range, the thin black line extended from it represents the 95% confidence intervals, and the white dot is the median.

Petal Length

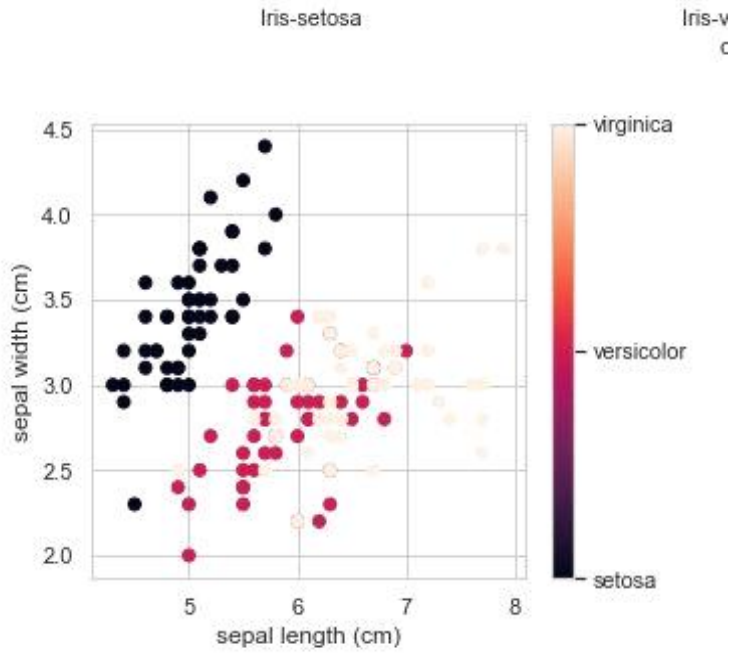


Petal Width

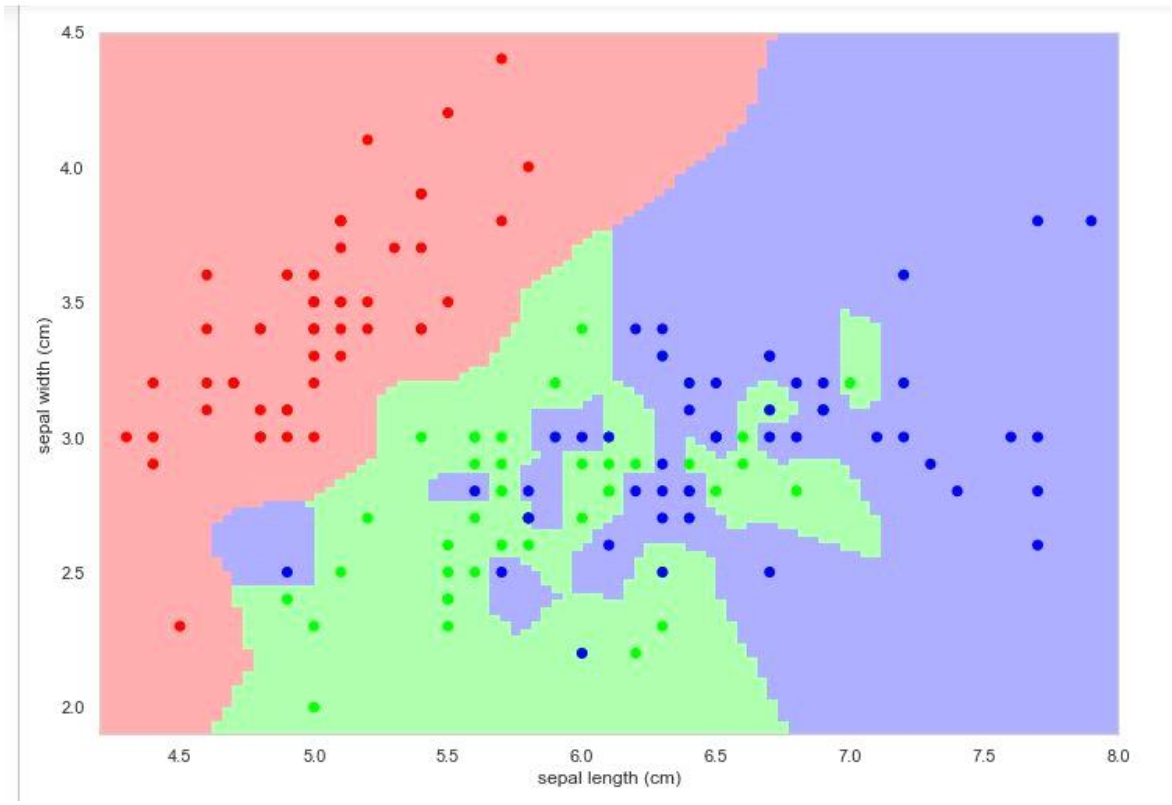


3.8.6 Scatter Plot

This data is four-dimensional, but we can visualize two of the dimensions at a time using a scatter plot.



A plot of the sepal space and the prediction of the KNN


4.5
4.0
3.5
3.0
2.5
2.0
sepal width (cm)
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
sepal length (cm)

Chapter-4

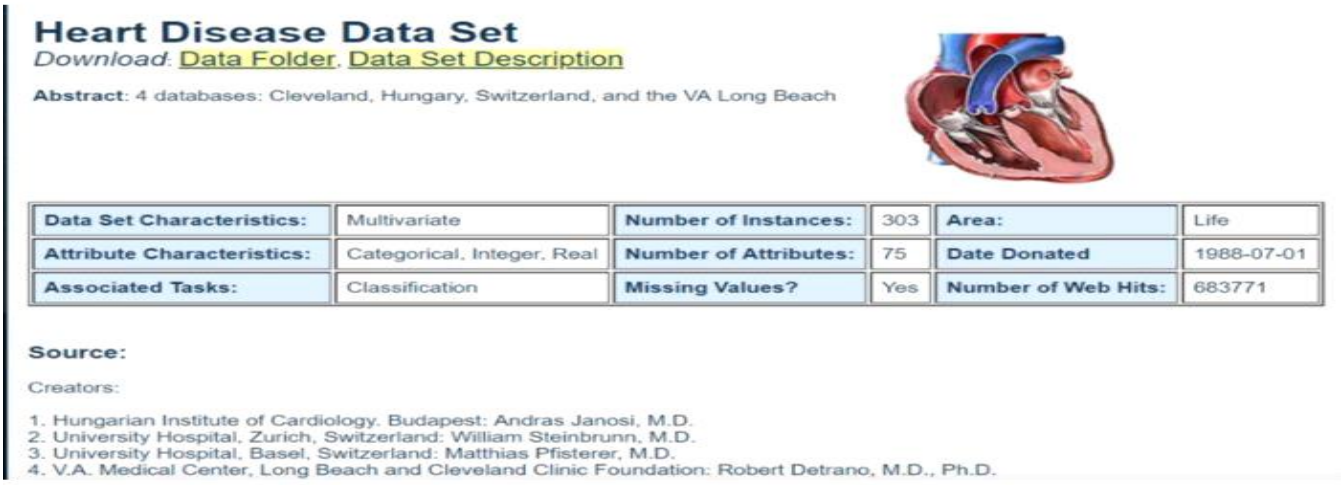
Data Analysis & Visualization of Heart Disease

4.1Introduction

In this, I'll discuss a project where I worked on predicting potential Heart Diseases in people using Machine Learning algorithms. The algorithms included `K Neighbors Classifier`, `Support Vector Classifier`, `Gaussian`. The dataset has been taken from Kaggle. My complete project is available at Heart Disease Prediction.

4.2 Dataset

This database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. In particular, the Cleveland database is the only one that has been used by ML researchers to this date. The "goal" field refers to the presence of heart disease in the patient.



Heart Disease Data Set

Download: Data Folder, Data Set Description

Abstract: 4 databases: Cleveland, Hungary, Switzerland, and the VA Long Beach

Data Set Characteristics:	Multivariate	**Number of Instances:**	303	**Area:**	Life
Attribute Characteristics:	Categorical, Integer, Real	**Number of Attributes:**	75	**Date Donated**	1988-07-01
Associated Tasks:	Classification	**Missing Values?**	Yes	**Number of Web Hits:**	683771

Source:

Creators:

1. Hungarian Institute of Cardiology. Budapest: Andras Janosi, M.D.
2. University Hospital, Zurich, Switzerland: William Steinbrunn, M.D.
3. University Hospital, Basel, Switzerland: Matthias Pfisterer, M.D.
4. V.A. Medical Center, Long Beach and Cleveland Clinic Foundation: Robert Detrano, M.D., Ph.D.

After downloading the dataset from Kaggle, I saved it to my working directory with the name `dataset.csv`. Next, I used `read_csv()` to read the dataset and save it to the `dataset` variable.

4.2.1 Import Dataset

As you can see from the output above, there are a total of 13 features and 1 target variable. Also, there are no missing values so we don't need to take care of any null values. Next, I used `describe()` method.

```
(303, 14)
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak
0	63	1	3	145	233	1	0	150	0	2.3
1	37	1	2	130	250	0	1	187	0	3.5
2	41	0	1	130	204	0	0	172	0	1.4
3	56	1	1	120	236	0	1	178	0	0.8
4	57	0	0	120	354	0	1	163	1	0.6
5	57	1	0	140	192	0	1	148	0	0.4
6	56	0	1	140	294	0	0	153	0	1.3
7	44	1	1	120	263	0	1	173	0	0.0
8	52	1	2	172	199	1	1	162	0	0.5
9	57	1	2	150	168	0	1	174	0	1.6
10	54	1	0	140	239	0	1	160	0	1.2
11	48	0	2	130	275	0	1	139	0	0.2
12	49	1	1	130	266	0	1	171	0	0.6
13	64	1	3	110	211	0	0	144	1	1.8
14	58	0	3	150	283	1	0	162	0	1.0
15	50	0	2	120	219	0	1	158	0	1.6
16	58	0	2	120	340	0	1	172	0	0.0

As you can see from the output above, there are a total of 13 features and 1 target variable. Also, there are no missing values so we don't need to take care of any null values. Next, I used `describe()` method.

4.2.2 Investigating the data:

Min, Max, Mean, Median and Standard Deviation

	age	sex	cp	trestbps	chol	fbs	\
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	

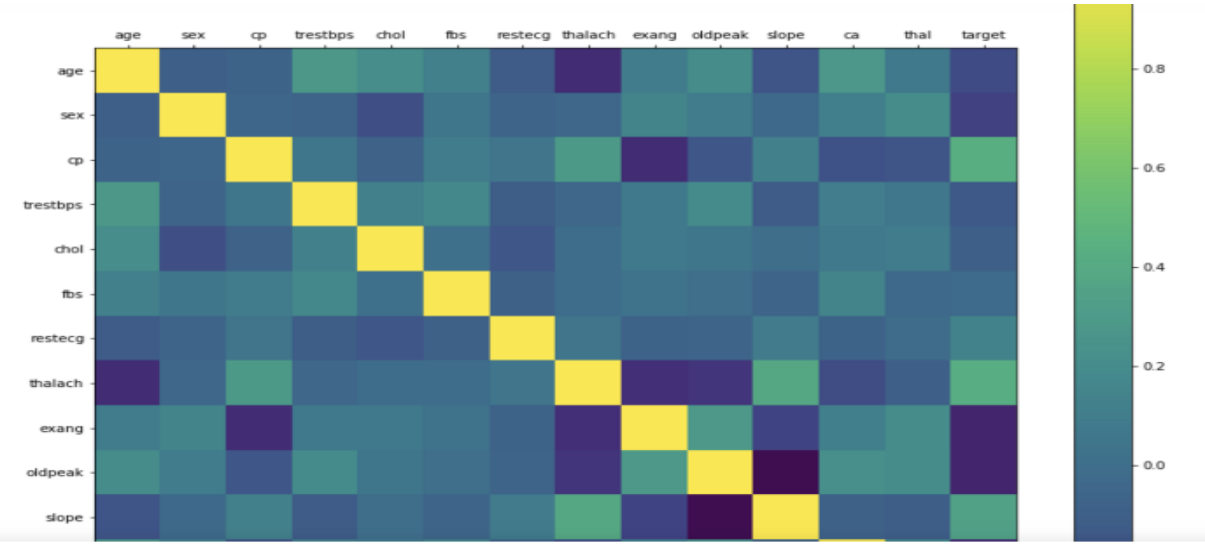
	restecg	thalach	exang	oldpeak	slope	ca	\
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	
mean	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	
std	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	
min	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	
50%	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	
75%	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	
max	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	

The method revealed that the range of each variable is different. The maximum value of `age` is 77 but for `chol` it is 564. Thus, feature scaling must be performed on the dataset.

4.3 Understanding the data

4.3.1 Correlation Matrix

To begin with, let's see the correlation matrix of features and try to analyse it. The figure size is defined to 12 x 8 by using `rcParams`. Then, I used pyplot to show the correlation matrix. Using `xticks` and `yticks`, I've added names to the correlation matrix. `colorbar()` shows the colorbar for the matrix.



It's easy to see that there is no single feature that has a very high correlation with our target value. Also, some of the features have a negative correlation with the target value and some have positive.

4.4Algorithms

4.4.1Support Vector Machine(SVM)

This classifier aims at forming a hyperplane that can separate the classes as much as possible by adjusting the distance between the data points and the hyperplane. There are several `kernels` based on which the hyperplane is decided. I tried four kernels namely, *linear*, *poly*, *rbf*, and *sigmoid*.

```
max       3.000000    1.000000
0.74
[[34 14]
 [12 40]]
              precision    recall  f1-score   support

         0.0       0.74      0.71      0.72        48
         1.0       0.74      0.77      0.75        52

   micro avg       0.74      0.74      0.74       100
   macro avg       0.74      0.74      0.74       100
weighted avg       0.74      0.74      0.74       100

0.62
[[24 24]
 [14 38]]
```

4.4.2 KNN Algorithm:

This classifier looks for the classes of K nearest neighbors of a given data point and based on the majority class, it assigns a class to this data point. However, the number of neighbors can be varied. I varied them from 1 to 20 neighbors and calculated the test score in each case.

```
0.62
[[24 24]
 [14 38]]
              precision    recall  f1-score   support

         0.0       0.63      0.50      0.56        48
         1.0       0.61      0.73      0.67        52

   micro avg       0.62      0.62      0.62       100
   macro avg       0.62      0.62      0.61       100
weighted avg       0.62      0.62      0.61       100
```

4.4.3 Gaussian NB Algorithm:

Here, we can vary the maximum number of features to be considered while creating the model. I range features from 1 to 30 (the total features in the dataset after dummy columns were added).

```
0.62
[[24 24]
 [14 38]]
              precision    recall  f1-score   support

         0.0       0.63      0.50      0.56        48
         1.0       0.61      0.73      0.67        52

   micro avg       0.62      0.62      0.62       100
   macro avg       0.62      0.62      0.61       100
weighted avg       0.62      0.62      0.61       100
```

4.5 Conclusion

The project involved analysis of the heart disease patient dataset with proper data processing. Then, 3 models were trained and tested with maximum scores as follows

K Neighbors Classifier: 62%

Support Vector Classifier: 74%

Chapter-5

Comparison of data produced within a country by national agencies with those published by international agencies

5.1 Introduction

In this article, I have used Pandas to analyze data on Country Data.csv file from UN public Data Sets of a popular 'statweb.stanford.edu' website.
As I have analyzed the Indian Country Data, I have introduced Pandas key concepts as below. Before going through this article, have a rough idea of basics from matplotlib and csv.

The first step is to read the data. The data is stored as a comma-separated values, or csv, file, where each row is separated by a new line, and each column by a comma (,). In order to be able to work with the data in Python, it is needed to read the csv file into a Pandas DataFrame. A DataFrame is a way to represent and work with tabular data.

```
(29, 16)
    Series Code Frequency                                          Series  \
0    IT_USE_ii99    Annual                                  Internet users
1    IT_USE_ii99    Annual                                  Internet users
2    IT_USE_ii99    Annual                                  Internet users
3    SE_ADT_1524    Annual                                   Literacy rate
4    SE_ADT_1524    Annual                                   Literacy rate
5    SH_DYN_MORT    Annual                      Under-five mortality rate
6    SH_DYN_MORT    Annual                      Under-five mortality rate
7    SH_DYN_MORT    Annual                      Under-five mortality rate
8    SH_STA_BRTC    Annual  Births attended by skilled health personnel
9    SH_STA_BRTC    Annual  Births attended by skilled health personnel
10   SH_STA_BRTC    Annual  Births attended by skilled health personnel
11   SH_STA_MORT    Annual                       Maternal mortality ratio
12   SH_STA_MORT    Annual                       Maternal mortality ratio
13   SH_STA_MORT    Annual                       Maternal mortality ratio
14   SH_TBS_PREV    Annual                  Tuberculosis prevalence rate
15   SH_TBS_PREV    Annual                  Tuberculosis prevalence rate
16   SH_TBS_PREV    Annual                  Tuberculosis prevalence rate
```

5.2 Indexing DataFrames with Pandas

Indexing can be possible using the pandas.DataFrame.iloc method. The iloc method allows to retrieve as many as rows and columns by position.

Creation of dataframe is done by passing multiple Series into the DataFrame class using **pd.Series** method. Here, it is passed in the two Series objects, s1 as the first row, and s2 as the second row.

	Series Code	Frequency	Series	Units of measurement	Location
5	SH_DYN_MORT	Annual	Under-five mortality rate	Per 1,000 live births	Total (national level)
6	SH_DYN_MORT	Annual	Under-five mortality rate	Per 1,000 live births	Total (national level)
7	SH_DYN_MORT	Annual	Under-five mortality rate	Per 1,000 live births	Total (national level)
8	SH_STA_BRTC	Annual	Births attended by skilled health personnel	Percent	Total (national level)
9	SH_STA_BRTC	Annual	Births attended by skilled health personnel	Percent	Total (national level)
10	SH_STA_BRTC	Annual	Births attended by skilled health personnel	Percent	Total (national level)
11	SH_STA_MORT	Annual	Maternal mortality ratio	Per 100,000 live births	Total (national level)
12	SH_STA_MORT	Annual	Maternal mortality ratio	Per 100,000 live births	Total (national level)
13	SH_STA_MORT	Annual	Maternal mortality ratio	Per 100,000 live births	Total (national level)
14	SH_TBS_PREV	Annual	Tuberculosis prevalence rate	Per 100,000 population	Total (national level)
15	SH_TBS_PREV	Annual	Tuberculosis prevalence rate	Per 100,000 population	Total (national level)
16	SH_TBS_PREV	Annual	Tuberculosis prevalence rate	Per 100,000 population	Total (national level)
17	SH_TBS_PREV	Annual	Tuberculosis prevalence rate	Per 100,000 population	Total (national level)
18	SH_TBS_PREV	Annual	Tuberculosis prevalence rate	Per 100,000 population	Total (national level)
19	SH_TBS_PREV	Annual	Tuberculosis prevalence rate	Per 100,000 population	Total (national level)
20	SI_POV_NAHC	Annual	Population below national poverty line	Percent	Total (national level)

Creation of dataframe is done by passing multiple Series into the DataFrame class using **pd.Series** method. Here, it is passed in the two Series objects, s1 as the first row, and s2 as the second row.

5.2 .1 Importing Data With Pandas

The first step is to read the data. The data is stored as a comma-separated values, or csv, file, where each row is separated by a new line, and each column by a comma (,). In order to be able to work with the data in Python, it is needed to read the csv file into a Pandas DataFrame. A DataFrame is a way to represent and work with tabular data.

	Observation Value	Unit multiplier	Nature of data points
0	0.2	Units	Country Data
1	3.2	Units	Country Data
2	7.0	Units	Country Data
3	48.5	Units	Country Data
4	76.4	Units	Country Data
5	109.3	Units	Country Data
6	94.9	Units	Country Data
7	74.3	Units	Country Data
8	33.0	Units	Country Data
9	42.4	Units	Country Data
10	48.2	Units	Country Data
11	398.0	Units	Country Data
12	327.0	Units	Country Data
13	301.0	Units	Country Data
14	503.0	Units	Country Data
15	412.0	Units	Country Data
16	371.0	Units	Country Data
17	343.0	Units	Country Data

5.3 Indexing Using Labels in Pandas

Indexing can be worked with labels using the **pandas.DataFrame.loc** method, which allows to index using labels instead of positions. The above doesn't actually look much different from df.iloc[0:5,:]. This is because while row labels can take on any values, our row labels match the positions exactly. But column labels can make things much easier when working with data.

	Time period details	Source details	Footnotes
0	1999	Ministry of Information Technology	NaN
1	2002	Ministry of Information Technology	NaN
2	2006	Ministry of Information Technology	NaN
3	1991	Planning Commission	NaN
4	2001	Planning Commission	NaN
5	1992	National Family Health Survey	NaN
6	1998	National Family Health Survey	NaN
7	2005	National Family Health Survey	NaN
8	1992	National Family Health Survey	NaN
9	1998	National Family Health Survey	NaN
10	2005	National Family Health Survey	NaN
11	1997	Sample Registration System	NaN
12	2000	Sample Registration System	NaN
13	2002	Sample Registration System	NaN
14	1990	NaN	NaN
15	2000	NaN	NaN
16	2001	NaN	NaN
17	2002	NaN	NaN

5.4 Investigating Data

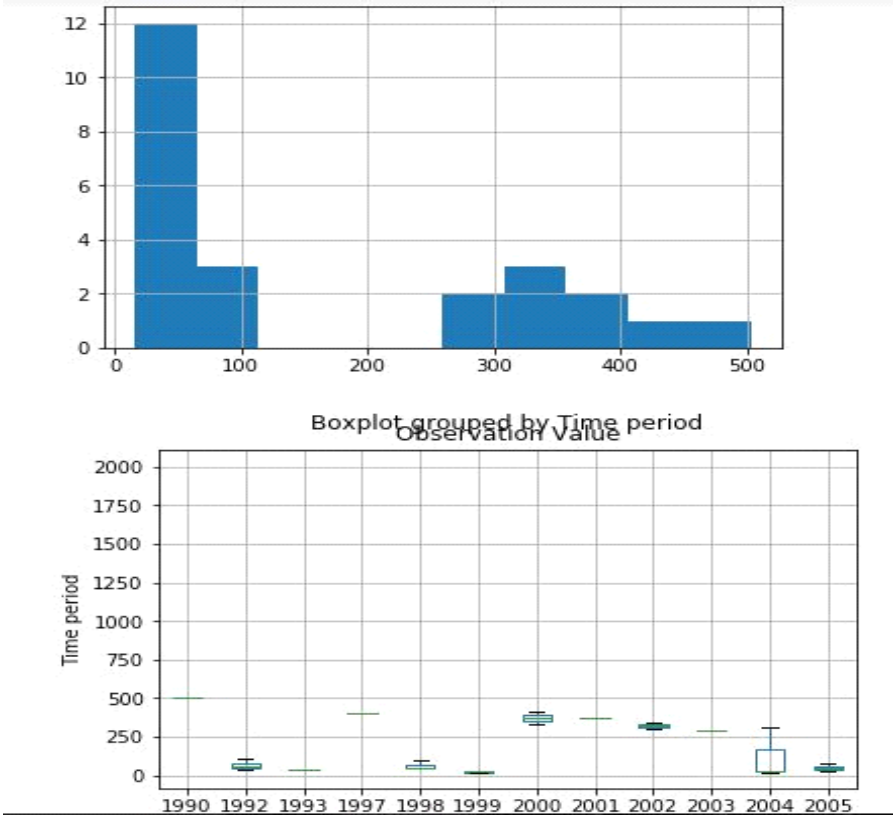
	Time period	Observation Value	Time period details
count	29.000000	29.000000	29.000000
mean	1999.551724	140.806897	1999.551724
std	4.807629	156.922569	4.807629
min	1990.000000	0.200000	1990.000000
25%	1998.000000	27.500000	1998.000000
50%	2000.000000	48.500000	2000.000000
75%	2004.000000	301.000000	2004.000000
max	2006.000000	503.000000	2006.000000

5.5 Computation of data frames can be done by using Statistical Functions of pandas tools

	Series Code	Frequency	Series	Units of measurement	Location	Age group	Sex	Reference Area	Source type	Time period	Observation Value	Unit multiplier	Nature of data points	Time period details	Source details	Footnotes
5	2.0	12.5	23.0	2.0	12.5	21.5	8.5	12.5	12.5	3.0	15.0	12.5	12.5	3.0	5.0	NaN
6	2.0	12.5	23.0	2.0	12.5	21.5	8.5	12.5	12.5	8.0	14.0	12.5	12.5	8.0	5.0	NaN
7	2.0	12.5	23.0	2.0	12.5	21.5	8.5	12.5	12.5	22.5	13.0	12.5	12.5	22.5	5.0	NaN
8	5.0	12.5	2.0	18.5	12.5	17.0	23.0	12.5	12.5	3.0	6.0	12.5	12.5	3.0	5.0	NaN
9	5.0	12.5	2.0	18.5	12.5	17.0	23.0	12.5	12.5	8.0	8.0	12.5	12.5	8.0	5.0	NaN
10	5.0	12.5	2.0	18.5	12.5	17.0	23.0	12.5	12.5	22.5	11.0	12.5	12.5	22.5	5.0	NaN
11	8.0	12.5	8.0	5.0	12.5	8.0	19.0	12.5	12.5	6.0	22.0	12.5	12.5	6.0	17.0	NaN
12	8.0	12.5	8.0	5.0	12.5	8.0	19.0	12.5	12.5	12.5	19.0	12.5	12.5	12.5	17.0	NaN
13	8.0	12.5	8.0	5.0	12.5	8.0	19.0	12.5	12.5	15.5	17.0	12.5	12.5	15.5	17.0	NaN
14	12.5	12.5	18.5	9.5	12.5	8.0	8.5	12.5	12.5	1.0	24.0	12.5	12.5	1.0	NaN	NaN
15	12.5	12.5	18.5	9.5	12.5	8.0	8.5	12.5	12.5	12.5	23.0	12.5	12.5	12.5	NaN	NaN

5.6 Pandas Plotting:

Plots in these examples are made using standard convention for referencing the matplotlib API which provides the basics in pandas to easily create decent looking plots.


12
10
8
6
4
2
0
0
100
200
300
400
500
Boxplot grouped by Time period
Observation Value
2000
1750
1500
1250
1000
750
500
250
0
Time period
1990
1992
1993
1997
1998
1999
2000
2001
2002
2003
2004
2005

Chapter-6

Pokemon Analysis

6.1 Introduction

Pokémon is a video game where creatures (known as Pokémon) of different types battle each other for glory. To commemorate the release of the newest Pokémon games for the Nintendo Switch (Let's Go Pikachu and Let's Go Eevee), we will aggregate and analyze Pokémon data in order to answer the following questions:

1. **How many Pokémon of each type are there?**
2. **Which Pokémon type is the most powerful?**
   First, we will complete our analysis using spreadsheets because spreadsheets are the most widely used tool for data analysis. However, Python programming provides more flexible and more scalable analysis options than spreadsheets, so we will complete the analysis using Python and the Pandas library.

6.2 Data :

link: https://www.kaggle.com/abcsds/pokemon

Name Name of the Pokemon
Type 1 First ability
Type 2 Second ability
Total Sum of all power points
HP Health Points
Attack Attack points
Defense Defense points
Sp. Atk Speed points on attack
Sp. Def Speed points on defense
Speed Speed points
Generation Number of generation (1-6)
Legendary True/False

Why analyse Pokémon?

I wanted to start off with a dataset that was relatively small and not too complicated. I found this dataset on Kaggle: "Pokémon with stats". I have a fair understanding of the columns and the data in the CSV file from that page so I thought, why not? The file consists of 800 rows and 13 columns, detailing the features of each Pokémon spanning 6 Generations.

6.3 Objective :

Exploratory data analysis on Pokémons
- Pokémon type
- Legendary vs Non-Legendary
- Power (HP, Attack, Defense, Sp. Atk, Sp. Def, Speed) range per Generation

K-Nearest Neighbors (KNN) model using power variables to classify between Legendary vs. Non-Legendary Pokémon.

6.4 Dataset

As you can see, from taking the first instance in the data frame, there are two indices to identify the Pokémon, one formed when the rows were entered in the data frame, and one from the file itself. After looking at the head and tail of the Dataframe, there is unnecessary text in front of some Pokémon names. This needs to be removed using regular expression (Regex).

#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False
2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	False
3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	False
3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	False
4	Charmander	Fire	NaN	309	39	52	43	60	50	65	1	False

6.5 Investigating Data

Since there is no type that is definitively the “strongest”, we will look at the strongest type for each stat. This would be useful for Pokémon players who are building balanced teams with both offensive;y- and defensively-inclined Pokémon. The dataset we have consists of Pokémon from 6 Generations. Conventionally, generations work independent of each other so an option would be to potentially analyse Pokémon with respect to their region.

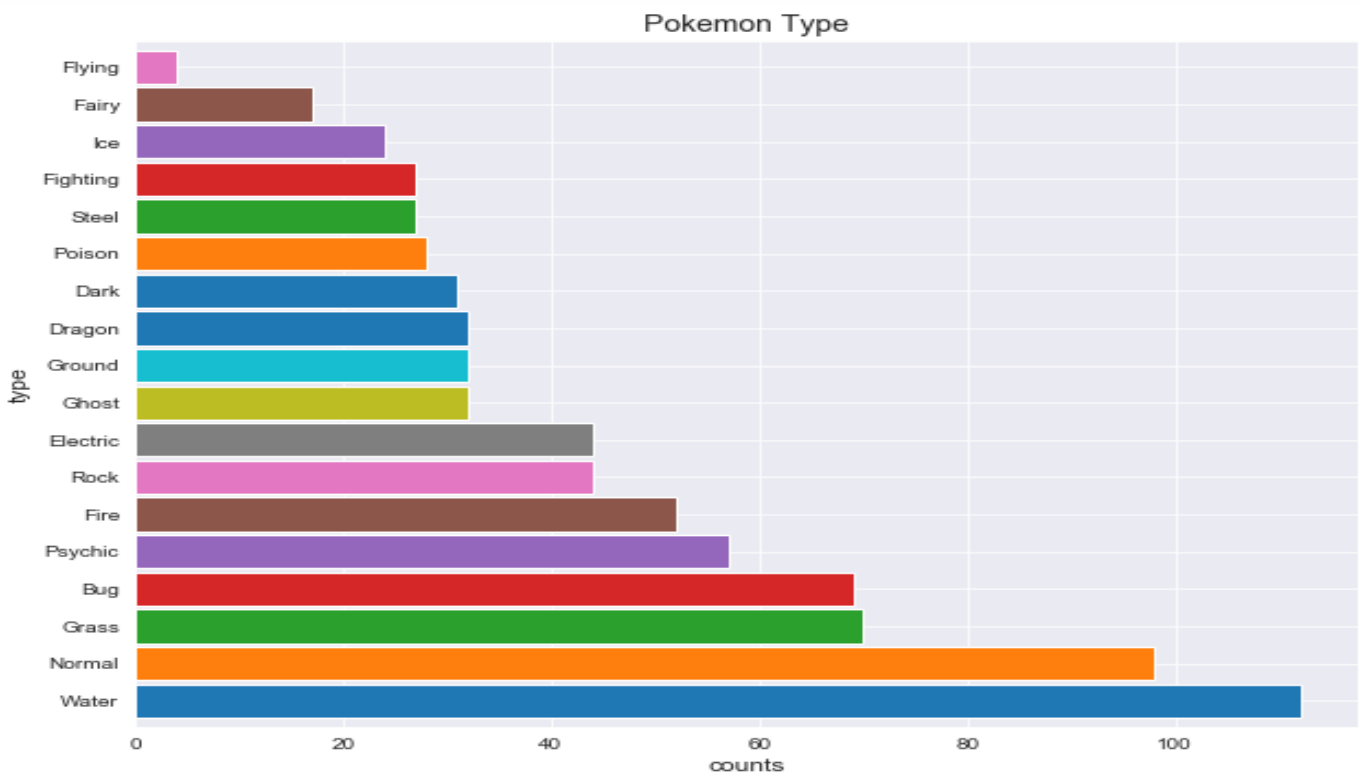
	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation
count	800.00000	800.000000	800.000000	800.000000	800.000000	800.000000	800.000000	800.00000
mean	435.10250	69.258750	79.001250	73.842500	72.820000	71.902500	68.277500	3.32375
std	119.96304	25.534669	32.457366	31.183501	32.722294	27.828916	29.060474	1.66129
min	180.00000	1.000000	5.000000	5.000000	10.000000	20.000000	5.000000	1.00000
25%	330.00000	50.000000	55.000000	50.000000	49.750000	50.000000	45.000000	2.00000
50%	450.00000	65.000000	75.000000	70.000000	65.000000	70.000000	65.000000	3.00000
75%	515.00000	80.000000	100.000000	90.000000	95.000000	90.000000	90.000000	5.00000
max	780.00000	255.000000	190.000000	230.000000	194.000000	230.000000	180.000000	6.00000

6.6 Pokemon Type

Legendary Pokémon are Pokémon that feature in myths in the Pokémon world; two of these take a Primal form. Primal Reversion is a transformation affecting Legendary Pokémon Kyogre and Groundon. Generation 6 saw the introduction of Mega Pokémon. This evolution is not applicable to all Pokémon. An example of this evolution is the Mega Evolution of Charizard. In this case, Charizard has two Mega forms, where they both have a Total base stat of 634, as opposed to Charizard's base form which has a Total base stat of 534.

Mega, Primal and Legendary Pokémon

All of the above are very powerful and therefore, their base stats are expected to be of the highest level amongst the dataset. Since not all Pokémon can take these forms, it would be a good idea to omit these types of Pokémon from our analysis.

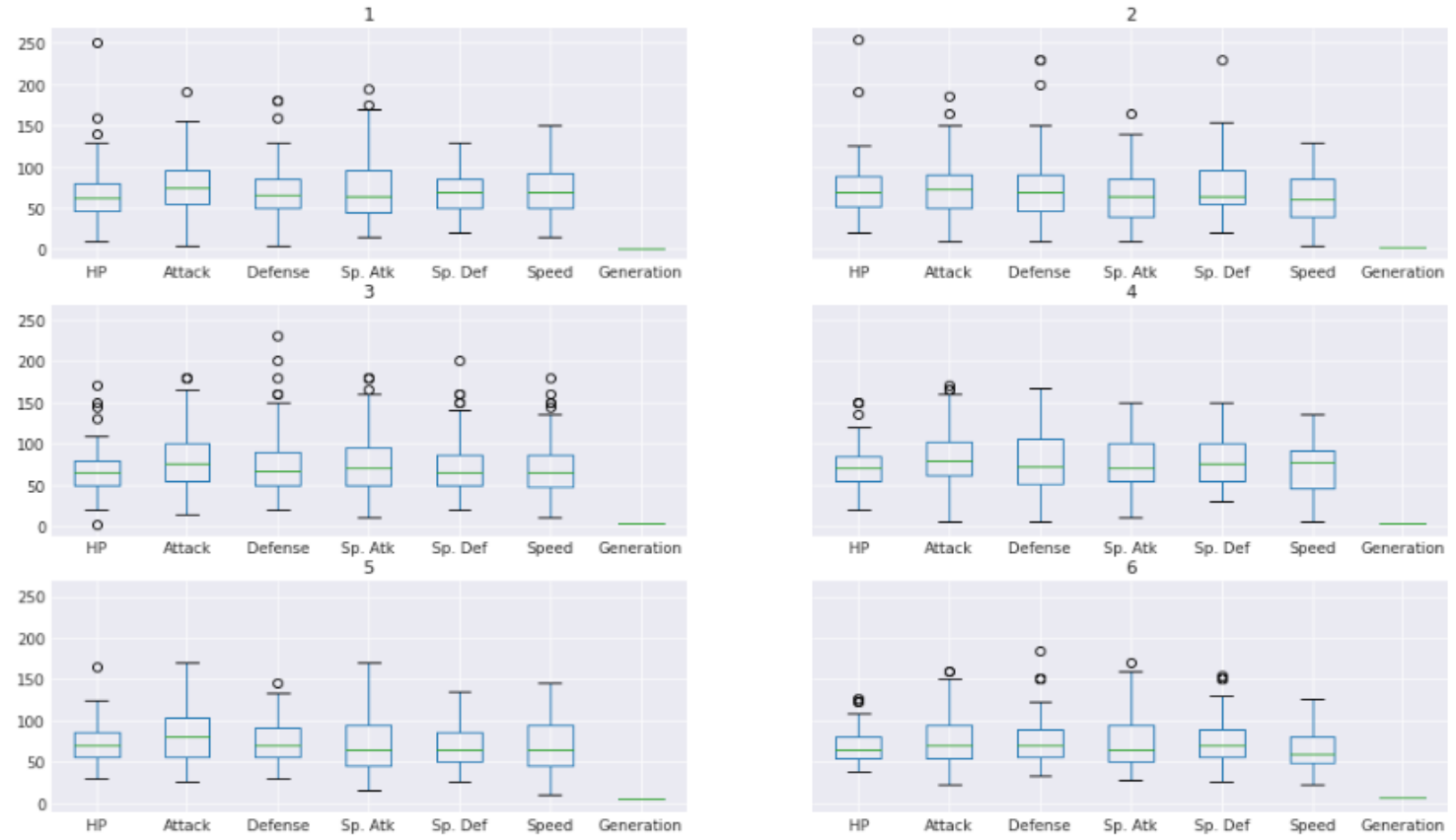


6.7 Data Visualization

6.7.1 BoxPlot

From this, we can see instantly that Generation 3 has the Pokémon with the highest total base stat. From printing the max stats using our UDF, we know that Pokemon is Slaking, with a base stat total of 670. All the other Generations share a high

base stat total of 600.



6.8 Algorithm Prediction

6.8.1 KNN Algorithm Analysis

Remove object dtypes: Name, Type 1, Type 2

Remove 'Total' column since it totals power points that are variables in the data

	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
#								
1	45	49	49	65	65	45	1	False
2	60	62	63	80	80	60	1	False
3	80	82	83	100	100	80	1	False
3	80	100	123	122	120	80	1	False
4	39	52	43	60	50	65	1	False

```
HP              int64
Attack          int64
Defense         int64
Sp. Atk         int64
Sp. Def         int64
Speed           int64
Generation      int64
Legendary        bool
dtype: object
```

6.9 Base Stat analysis

6.9.1 HeatMap

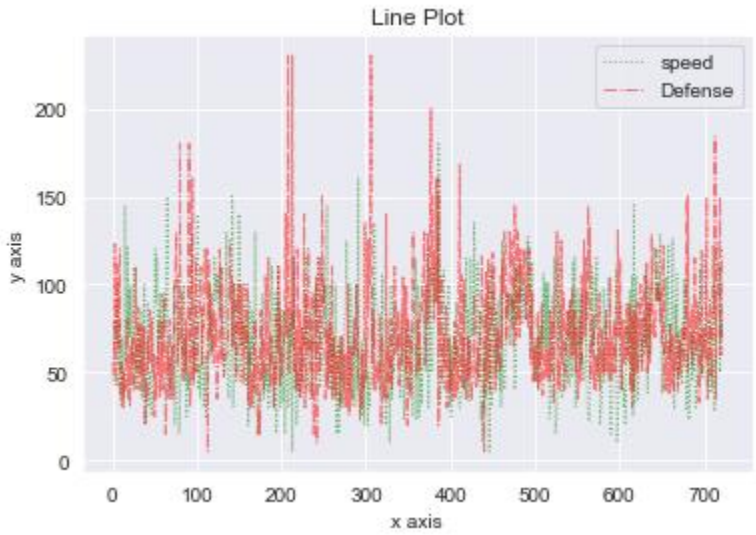
The following cell and graphic will express the correlation between each of the base stats against each other. From the heat map, we can see that the correlation between the Sp.Def and Total is 0.68, which is the highest in the matrix (excluding the diagonal). We can go one step further and create a scatter plot of the Sp.Def and Total.

```
                          Name Type 1   Type 2   Total   HP   Attack   Defense
#
1                    Bulbasaur   Grass   Poison     318   45       49        49
2                      Ivysaur   Grass   Poison     405   60       62        63
3                     Venusaur   Grass   Poison     525   80       82        83
3       VenusaurMega Venusaur   Grass   Poison     625   80      100       123
4                   Charmander    Fire      NaN     309   39       52        43
5                   Charmeleon    Fire      NaN     405   58       64        58
6                    Charizard    Fire   Flying     534   78       84        78
6   CharizardMega Charizard X    Fire   Dragon     634   78      130       111
6   CharizardMega Charizard Y    Fire   Flying     634   78      104        78
7                     Squirtle   Water      NaN     314   44       48        65

    Sp. Atk   Sp. Def   Speed   Generation   Legendary
#
1         65        65      45            1       False
2         80        80      60            1       False
3        100       100      80            1       False
3        122       120      80            1       False
4         60        50      65            1       False
5         80        65      80            1       False
6        109        85     100            1       False
6        130        85     100            1       False
6        159       115     100            1       False
7         50        64      43            1       False
Index(['Name', 'Type 1', 'Type 2', 'Total', 'HP', 'Attack', 'Defense',
       'Sp. Atk', 'Sp. Def', 'Speed', 'Generation', 'Legendary'],
      dtype='object')
```

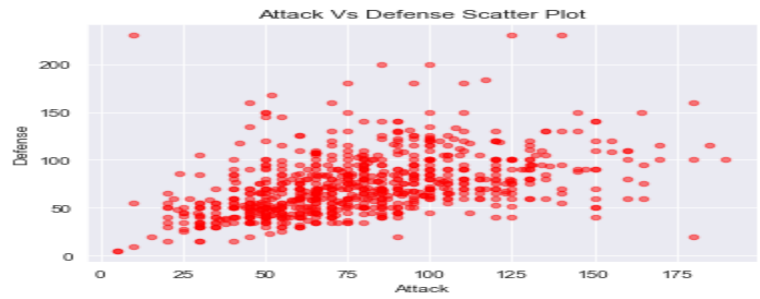
6.9.2 Line Plot

Here the line plot visualize the relationship between two modules on different axis, i.e x-axis and y-axis. The green color lines displays the information of speed and red color line visualize the defense as a series of datapoints, where the time interval is varied for each attributes based on our dataset.



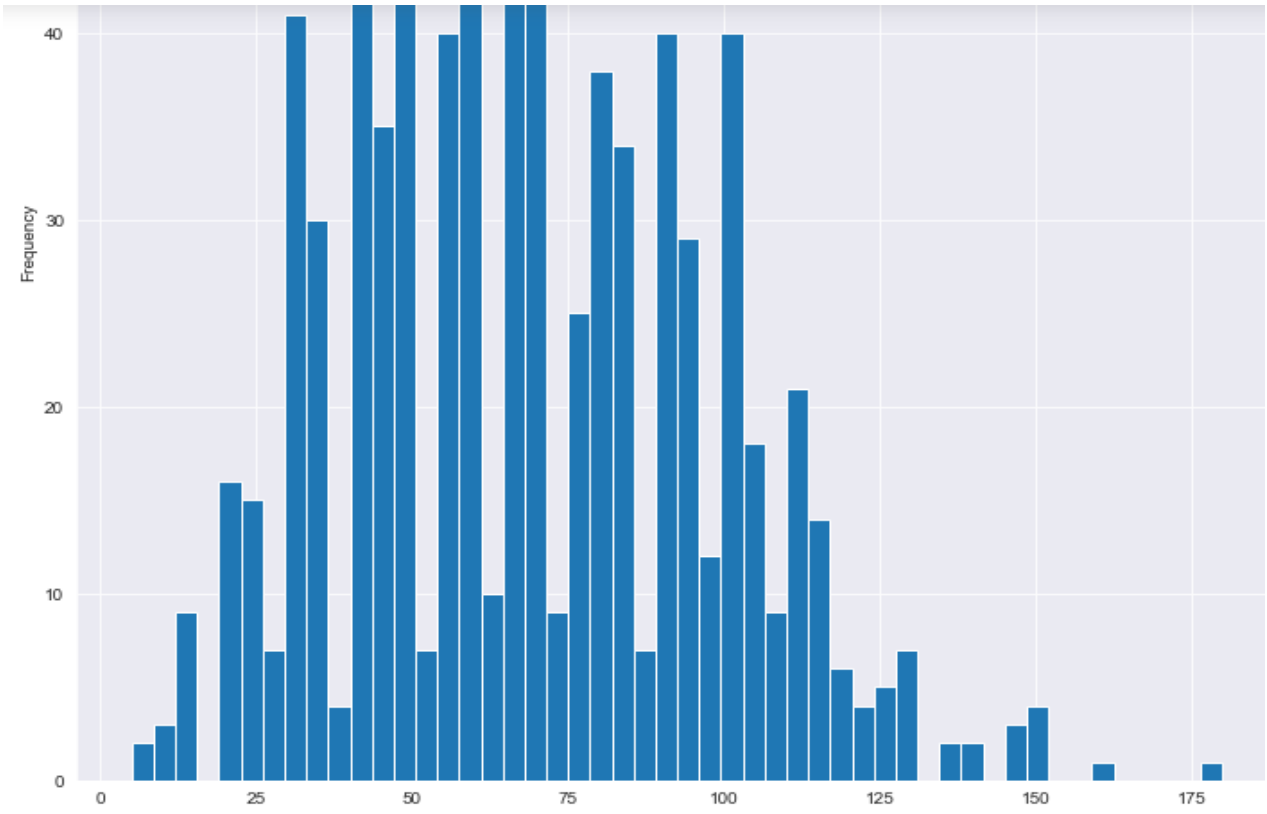
6.9.3 Scatter Plot

In this plot, the data is classified into two groups i.e attack and defense. Here the plot describe data as a collection of point where the dense part visualize that the attack and defense are on the same track, with dimension value where each value is a position on either horizontal and vertical.



6.9.4 Histogram

In this plot, it displays the accurate graphical representation of distribution of numerical data.Each bar shows the frequency on the vertical and horizontal axis.It predict the frequency of attack and defense of each module.



Conclusion

In this project, we have analyzed the accuracy of all the five datasets in terms of applying data analysis algorithm [KNN, SVM, Gaussian Naïve Algo] through classification technique, and visualized each our datasets through graphical representation for better understanding using Matplotlib library through Numpy packages.

Sometimes data does not make sense until you can look at in a visual form, such as with charts and plots.So that human mind can easily visualize things by seeing any objects in pictorial form, so here we are making our datasets flexible to understand.

However, we have summarize data distributions with histograms and box plots, the relationship between variables with scatter plots. When exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and so on.

References

Web Sites:-

- www.kaggle.com
- www.analyticsvidhya.com
- UCI Machine Learning Repository
- www.codeacademy.com

Appendix

- **NumPy**. It provides some advance math functionalities to python.
  matplotlib. A numerical plotting library. It is very useful for any data scientist or any data analyzer.

- **Pandas** is a popular Python library for data analysis. It is not directly related to Machine Learning. As we know that the dataset must be prepared before training. In this case, Pandas comes handy as it was developed specifically for data extraction and preparation. It provides high-level data structures and wide variety tools for data analysis. It provides many inbuilt methods for groping, combining and filtering data.

- **matplotlib.pyplot** is a plotting library used for 2D graphics in python programming language.

- **class sklearn.model_selection.KFold(n_splits='warn', shuffle=False, random_state=None)**

K-Folds cross-validator
Provides train/test indices to split data in train/test sets. Split dataset into k consecutive folds (without shuffling by default).
Number of folds. Must be at least 2.

Changed in version 0.20: n_splits default value will change from 3 to 5 in v0.22.

shuffle : boolean, optional
Whether to shuffle the data before splitting into batches.

random_state : int, RandomState instance or None, optional, default=None
If int, random_state is the seed used by the random number generator; If RandomState instance, random_state is the random number generator; If None, the random number generator is the RandomState instance used by np.random. Used when shuffle == True.

- **split(X, y=None, groups=None)**

Parameters: X : array-like, shape (n_samples, n_features)
Training data, where n_samples is the number of samples and n_features is the number of features.

y : array-like, shape (n_samples,)
The target variable for supervised learning problems.

groups : array-like, with shape (n_samples,), optional
Group labels for the samples used while splitting the dataset into train/test set.

- **Seaborn** is a library for making statistical graphics in Python. It is built on top of matplotlib and closely integrated with pandas data structures.

- **patsy.dmatrix(formula_like, data={}, eval_env=0, NA_action='drop', return_type='matrix')**

Parameters: **formula_like** – An object that can be used to construct a design matrix. .
data – A dict-like object that can be used to look up variables referenced in formula_like.
eval_env – Either a EvalEnvironment which will be used to look up any variables referenced in formula_like that cannot be found in data, or else a depth represented as an integer which will be passed to EvalEnvironment.capture(). eval_env=0 means to use the context of the function calling dmatrix() for lookups. If calling this function from a library, you probably want eval_env=1, which means that variables should be resolved in your caller's namespace.

NA_action – What to do with rows that contain missing values. You can "drop" them, "raise" an error, or for customization, pass an NAAction object. See NAAction for details on what values count as ‘missing’ (and how to alter this).
return_type – Either "matrix" or "dataframe".

- **patsy.dmatrices(formula_like, data={}, eval_env=0, NA_action='drop', return_type='matrix')**

Construct two design matrices given a formula_like and data.

This function is identical to dmatrix(), except that it requires (and returns) two matrices instead of one. By convention, the first matrix is the “outcome” or “y” data, and the second is the “predictor” or “x” data.