Analysis And Future Prediction of Dengue Fever Using Machine Learning Algorithm

Reflective Report

Submitted in partial requirements for the degree of MSc Data Science

SAICHANDRA MUVVA

W9533735

Supervisor: Dr. Muhammad Zahid Iqbal

School of Computing, Engineering & Digital Technologies

Middlesbrough TS1 3BA

Abstract

The use of machine learning techniques to forecast dengue virus cases has improved the accuracy of disease prediction, which is crucial for effective disease control and prevention. With the help of These Algorithms in the medical sector helps to analyze the disease and patterns of change with respective to time and severeness of Illness can be controlled by these techniques and early prediction of disease helps to tackle the situation efficiently.

These methods play a vital role in recent outbreak of covid, and dengue fever is a leading worst scenario in the tropical areas of the world 390 million infections are occurring every year and there is no perfect treatment for this problem causing nightmare to the health sector. As a result, machine learning algorithms have become increasingly popular in recent times for predicting dengue virus cases. Choosing the most appropriate algorithm is important for ensuring accurate predictions. Typically, supervised, or unsupervised learning techniques are employed for dengue prediction.

To predict future outbreaks, a novel dengue virus prediction system was developed by combining a conventional supervised machine learning model with a random forest algorithm in this study and a few other algorithms. The model was trained on demographic data and historical dengue cases. The study used in this research is original patient data deployed by the America health agency. To analyze the virus behavior over the years and Entire Research is done on the reference to Ethics Report, and cybersecurity aspects were considered during the entire research process.

Acknowledgement

I would like to convey my sincere credit to my professor, Zahid Iqbal, for his guidance and support throughout my module for CIS4055 for master’s in data science. His encouragement and mentorship have been invaluable, and I am grateful for his willingness to go beyond in his role as an educator.

I would also like to thank my parents for their steady support and encouragement. Their love and guidance have been a constant source of strength and motivation, and I could not have achieved my academic goals without their unwavering belief in me. In addition, I would like to extend my appreciation to Teesside university for providing me with the opportunity to pursue my academic interests and providing the necessary resources for me to succeed. I would also like to acknowledge the contributions of my classmates and friends who have supported me throughout my academic journey. Their support, encouragement, and friendship have made the experience more rewarding. Finally, I would like to thank all the authors and researchers whose works have influenced my own research. Their contributions have been instrumental in shaping my academic interests and informing my research. I am deeply grateful to everyone who has played a role in my academic journey. Without their support, guidance, and encouragement, I would not have been able to achieve my academic goals.

Table of contents

List of Figures

List of Tables

Acronyms and Abbreviations

Word	Abbreviation
WBC	White blood cells
DENV	Dengue virus
WHO	World health organization
RT-PCR	Reverse Transcription polymerase chain reaction
NS1	Nonstructural protein 1
ELISA	Enzyme-linked immunosorbent assay
RNA	Ribonucleic acid

1.Introduction

1.1. Virus History

Dengue fever spread through a virus from **FLAVIVRIDAE** family, and it was first discovered in 1943 (M.s & S, 2015) by Japanese scientists in Nagasaki. Few studies Tell it was evolved Before humans and there is no convincing evidence to prove It, the microbe lies in the arthropods. This virus family is responsible for Zika virus, chikungunya, and Yellow Fever. This Infection Enters into the human body Through Blood stream and effects monocytes in The Immune system. Over the years it was evolved and formed four stereotypes Denv1, Denv2, Denv3, Denv4.

1.2. Spread

The infection expansion was through Mosquitoes. It spreads through a Female Mosquito of **AEDES AEGYTI** (Dash, et al., 2013) family, and its life span is 2 weeks. Male mosquitoes feed from nectar present in flowers of a plant and female mosquitoes feed from the blood of humans and animals and these are responsible for reproduction in their species, and it will feed maximum 3,4 times in lifetime. The mosquito sucks blood from a patient with dengue fever and sucks to a healthy person for blood to feed in the mean while the virus is transmitted through this process and causing spread of this virus. Mosquitoes are the main carrier of this disease.

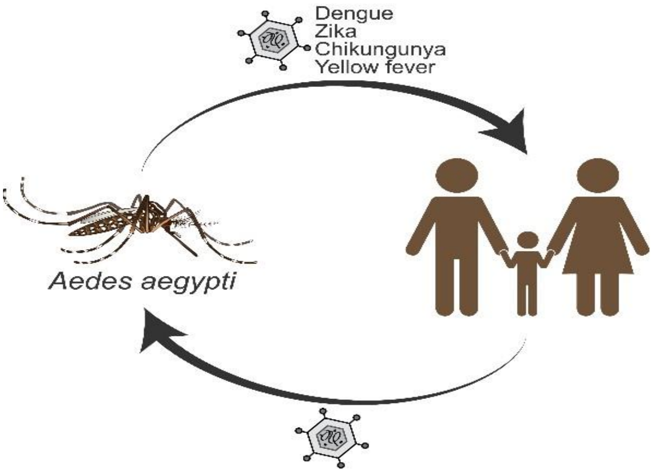


Figure 1: cycle of Dengue virus

1.3. Infection Rate

A study says Due to this virus nearly 390 million (Barbosa & Sandra, 2020) infections are taking place every year and among them 90 million are sick due to this virus 36000 deaths are caused every year. (Guha-Sapir, 2005) It normally spreads in tropical areas in the world like South America, North America, Asia, Australia, and Africa which is causing 40-50% of risk to the world population.

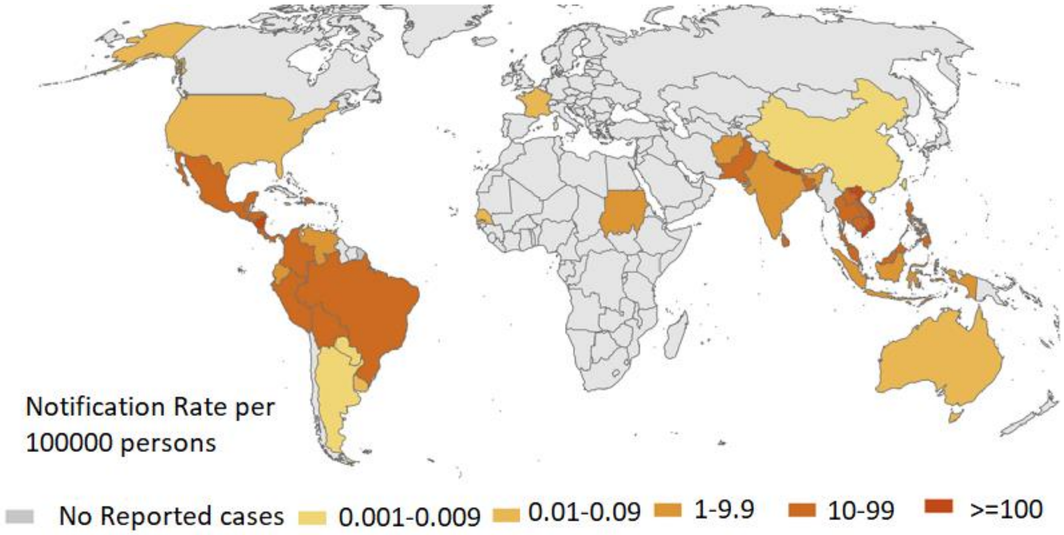


Figure 2: worldwide dengue infection

1.4. Vaccine and Treatment

Vaccine released and approved in eighteen countries. in 2016 WHO suggest avoiding the vaccine because it is not defending the virus and recurring chances are high and Mentioned about the Types of Dengue Fever Earlier. The vaccine of one variant will not be effective to another stereotype. This Virus is Similar to covid Virus due to Various Mutant viruses (Webster, 2009) the vaccine is not effective. Till now There is no effective Treatment for This fever only general treatment of using Paracetamol and Hydrated liquids to the body. If a patient is attacked with a variant and recovered, he is immune to the variant for the rest of his life and, but he does not have the immunity to other types of variants. children with age 5-9 and adolescents from age 12-19 having fatality. Denv2 (Renantha, 2022) is the most lethal type Which is having the highest Death toll.

1.5. Virus Mechanism

Even now it was a puzzle that why everyone will not be sick for dengue fever and only 20 percent of infection gets sick. the virus enters in to the blood stream by the mosquito bite and it takes white blood cells as the host and this germ creates a PH such that a pore is formed and virus penetrates in to the WBC and it uses its energy and metabolism to replicate the virus cells thus Virus is grown inside the body and effects the platelets and normal count is 450,000 (Ubol, 2010) per adult but this attacks on platelets results in decrease in count and leads to death.

1.6. Testing

- Rt-pcr – in which Virus is developed through enzymes and RNA is multiplied and virus is detected.
- NS1 Detection – NS1 is a protein produced by Dengue virus and Disease is Detected Through it.
- ELISA – it is used to detect antibodies with this technique used to Detect viruses inside the body.

Comparing Three Methods, Rt-par (Lai, 2007) is the most accurate method to detect the virus followed by Elisa and Ns1.but Ns1 is advisable because when the condition gets serious it doesn't discover the problem inside the patient it is only effective during early stages.

1.7. Research Question

1. Can we analyze the spread of dengue fever using various conditions and inputs?
2. How can machine learning algorithms help in early detection of dengue virus spread?

1.8. Objectives

Evaluate the current state of research in the medical field, including advancements and methods for evaluation. Examine the available literature on the spread of dengue fever that utilizes machine learning techniques. Use exploratory data analysis to gain valuable insights from the data. Assess the different approaches used to improve the existing model's performance and select the efficient one based on predicted outcomes. Perform an analysis on the proposed model and compare it with an existing model. Explore the benefits using these Techniques and suggest a secure workflow methodology to implement alongside the machine learning model. Finally, identify potential areas for future research in this field.

1.8.1. Expected Deliverables

The primary focus of this research is to analyze and detect chances of occurrence of dengue fever using machine learning techniques based on insights from the data. The proposed model incorporates Patient demographics to improve Health safety. By analyzing the disease patterns of patients, the algorithm proposed in this study can assist the Government, people to have awareness and help to fight against the virus spread. This research will provide a justification for the comparison analysis between the literature works of existing and the proposed recommender model. During complete study, any associated risks will be evaluated.

1.9. Ethical Considerations

1.9.1. Ethical issues

The proposed research is considered ethically sound as it will utilize publicly available datasets from Kaggle, eliminating any concerns related to data privacy. No personal demographic data will be collected independently. Additionally, all algorithms used in this project will be cited, and their implementation will be made publicly accessible for free, and the research would adhere to the policies of Teesside University.

1.9.2. Legal Issues

The data used in this study will be sourced from the domain of public, ensuring compliance with relevant legal guidelines. As this research involves no participation from external individuals or companies, there are no legal issues. Nevertheless, the research will comply with the legal policies set forth by Teesside University.

1.9.3. Professional Issues

The research will be conducted in a professional manner, with utmost respect for fairness and dignity, in accordance with the policies set forth by Teesside University.

1.9.4. Ethics Declaration

	True	False
My project is purely focused on technical or literature-based research without any practical implementation involved,	✓	
There are no participants involved in my project. It is solely based on data analysis,	✓	
My project is self-contained and does not require any external inputs, such as consultation or collaboration with industry experts,	✓	
My project can be completed entirely on campus or remotely, without the need to work off-campus, for example, at a company,	✓	
My project exclusively utilizes primary data sets and does not rely on secondary data sources.	✓	

Table 1: Ethics Declaration

2.Literature Review

Dengue fever is a significant public health concern in many tropical and subtropical regions around the world. It is caused by the FLAVIVRIDAE virus and is transmitted through a flies called Aedes mosquito. Dengue fever can cause severe flu-like symptoms, and in severe cases, it can lead to dengue hemorrhagic fever, which can be fatal. The prevention and control of dengue outbreaks depend on accurate and timely prediction of its occurrence.

Prediction algorithms have become a popular tool for predicting dengue outbreaks in recent years. The effectiveness of different prediction algorithms in predicting dengue outbreaks has been investigated in several studies. In this literature review, explore the different prediction algorithms that have been used for dengue fever analysis and their effectiveness in predicting outbreaks. Decision trees are widely used prediction algorithms in data mining. They are simple to understand and interpret, making them a useful tool in predicting dengue outbreaks (Rangarajan, 2019). conducted a study in Brazil using decision tree algorithms to predict dengue outbreaks. This study found that the decision tree algorithm was able to accurately predict outbreaks with a high degree of accuracy. Neural networks are another type of machine learning algorithm that have been used to predict dengue outbreaks (Harapan, 2018) used neural network algorithms to predict dengue outbreaks in Indonesia. This study demonstrated that neural network algorithms were able to accurately predict outbreaks with a high degree of accuracy. Support vector machines are a type of machine learning algorithm that has also been used to analyze dengue outbreaks (Singha, 2016)used a support vector machine algorithm to analyze dengue outbreaks in India. The study of this research helped to study the factors that influenced the spread of the disease. The random forest algorithm is a popular machine learning technique that has found widespread application in many fields, including predicting dengue outbreaks (Rangarajan, 2019)random forest is used to predict dengue outbreaks in Delhi, India. This research found that the random forest algorithm was able to accurately predict outbreaks with a high degree of accuracy.

In a comparative study conducted by (Alera, 2016)different machine learning algorithms were compared for their effectiveness in predicting dengue outbreaks. The study found that the random forest algorithm outperformed other algorithms such as support vector machine, decision trees and neural networks in predicting dengue outbreaks.

The use of these methods, particularly the random forest algorithm, can be a valuable tool for forecasting and controlling the spread of dengue fever. However, it is important to note that these algorithms should not be used in isolation, and other factors such as climate, population density, and other environmental factors should also be considered when predicting dengue outbreaks and may changes with change in environment, immunity of people.

All the above research conducted on the data of tropical regions of in Asia and South America but in this study deals with the population in North America This Research study includes the promotion of factors influencing the spread of the disease and is it possible that machine learning algorithms helps to tackle the predict of virus outbreak early.

The virus evolves with time and the features may vary with the mutation (Dowd, 2015) and this is the reason there is no treatment or the vaccine for this disease till now. and all the research done with parameters related to that area and limited to that of people living in that area, not for complete set of people.

The below figure illustrates how the workflow is done throughout the research. The first step includes picking the right dataset and preprocessing and then analyze the hidden patterns from the dataset and next step includes separating the data into test and train sets and finally compare the test cases accuracy by running the suitable machine learning algorithms and conclude the best model for the research by output accuracy.

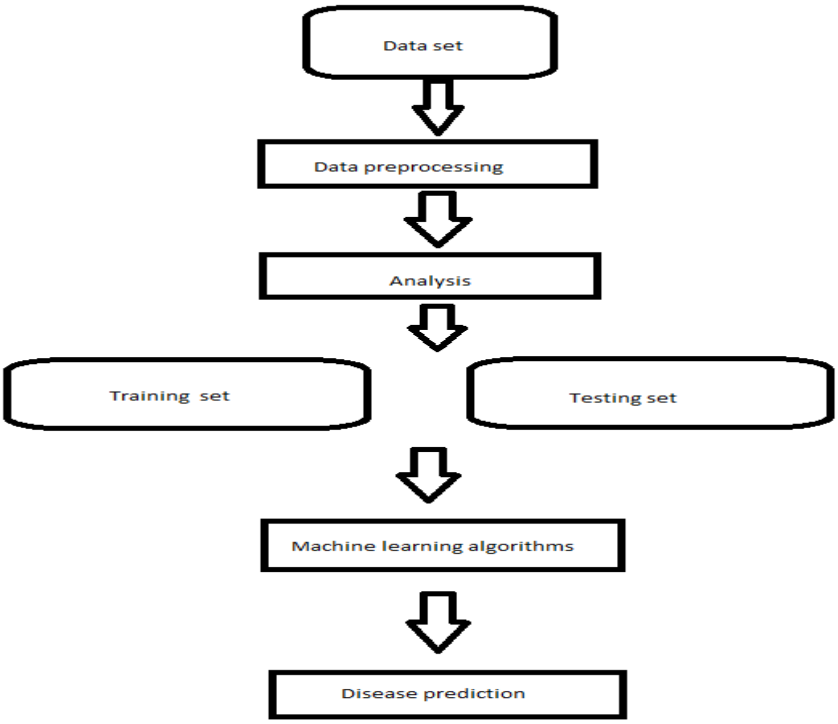


Fig 3: Flow chart for Disease Prediction

3.Software Requirements

3.1. Programming Language

This research utilized python as programming language, which is a high-level language compiled using interpreter (Van Rossum, 2009). Specifically, version 3.9.16 of Python was used for programming in this study.

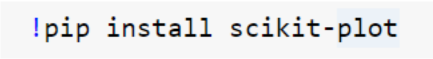
3.2. Integrated Development (IDE)

The Python program was coded using Google Colaboratory (Google Colab)[1] notebook as the integrated development environment (IDE). These Jupyter notebooks are hosted on the cloud and are seamlessly integrated with Google Drive, simplifying their creation, utilization, and sharing.

3.3. Library Installation

The commands below were required to install in Python libraries for this research, which were not readily available in Google Colab

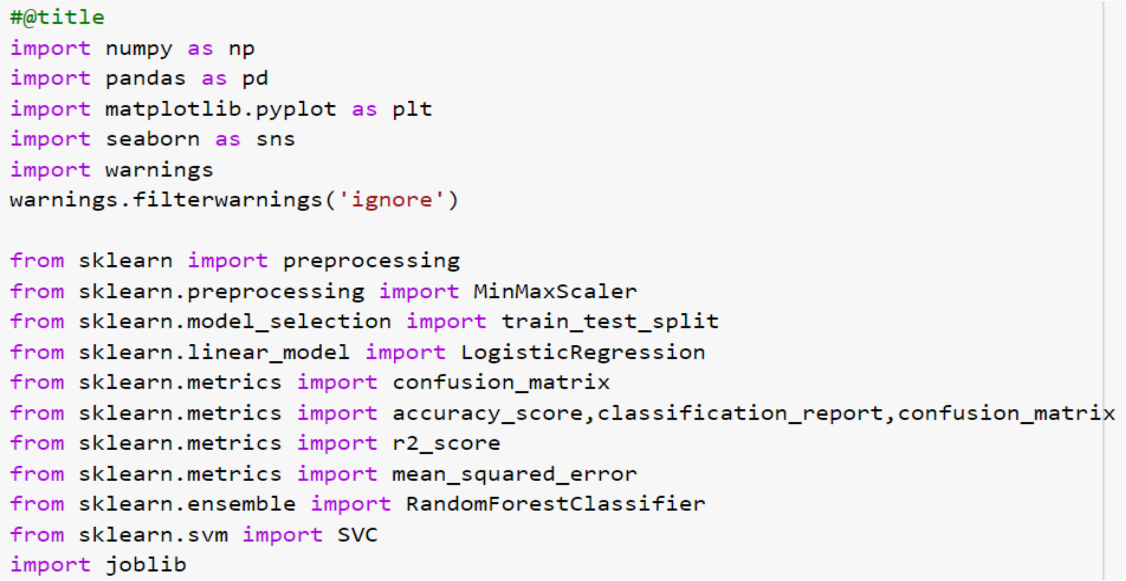
“pip” command stands for package manager for python is used install following libraies shown below



```
!pip install scikit-plot
```

3.4. Importing Libraries

The Following Libraries are imported as shown below diagram.



```
#@title
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')

from sklearn import preprocessing
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
import joblib
```

4.Research Methodology

Research methodology is important because it helps to make the research effective and helps to find the results accurate and meaningful conclusions.

[1] https://colab.research.google.com/

4.1. Dataset Information

The Dataset used for this research is taken from the Kaggle[2]. It was uploaded by the health department of united states of America. All these files have dengue cases registered in the cities of San Juan and Iquitos these two cities are near to south America and in tropical region and having the high dengue infection rate and the population of San Juan is 3.2laks and Iquitos is 4.9laks. This zip folder is downloaded. Having four distinct types of files inside the folder and all the formats of the files are "CSV" (comma separated values).

4.2. Dataset Description

There are three datasets used in this research and the columns of these datasets are below.

- City column consists of two values they are "SJ" for San Juan and "IQ" for Iquitos near to Cuba.
- Year column describes the record of dengue fever in those cities.
- The total number of cases column is for registration of total cases in that date.
- Week of year and week_start _date columns describe the week when virus is found on patient body.
- NDVI (normalized difference vegetation index) to determine the density of greenery in a piece of land and the values of ndvi are designed as below and ndvi_ne as NDVI in northeast similarly northwest, southeast, southwest in the dataset.

NDVI	**Value**
0	Dead plant
0-0.33	Unhealthy Plant
0.33-0.66	Moderate
0.66-1	Healthy Plant

Table 2: NDVI classification

- Precipitation_amt_mm column describes the number of liters of water per meter area.
- Station_min_temp and station_max_temp columns are the max and min temperatures of dengue case record in that city.
- The term "Dew point" is used to describe the measure of moisture in air. If dewpoint < Temperature feels more humid, so values of dewpoint and relative humidity are in their respective columns of dataset.
- Station_avg_temp_c column describes the temperature of the dengue record in that area in Celsius scale.
- Specific humidity and Relative humidity columns speak about the values that are recorded during dengue case register.
- The average temperature column is added in the dataset it is the mean from hot temperature and low temperature on a particular day.

[2] https://www.kaggle.com/datasets/arashnic/epidemy.

4.3. Dataset Loading

All the three datasets are in the same format. The datasets are uploaded into google colab using the syntax below fig. The syntax “pd” stands for panda data frame object and “read” command is used to read the data frame of path where the file is located.

4.4. Data Preprocessing

The process of data preprocessing enhances the accuracy and dependability of the dataset. By removing erroneous and inconsistent data values caused by human or machine error, the quality of the dataset can be improved, resulting in more reliable data. Additionally, data preprocessing helps to make the data consistent.

4.4.1. Read Data

The dataset is uploaded into the google colab by the command mentioned earlier about the data format as csv. the count of rows in a data frame represents the total number of records contained in it, while the columns count indicates the number of features present in the data frame, the header of each data file is displayed on the first row and the shape of the file is obtained by using “df. head ()” with the help of this to see the data frame structure.

4.4.2. Dataset Merging

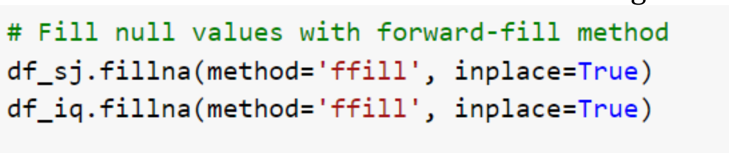
As mentioned earlier, the dataset and these are merged into a single dataset using “concat” function with the help of this function these datasets are combined as single dataset easy for further process.

4.4.3. Missing values

There are some missing data, and these are found by applying the syntax “isnull.sum ()” so the table columns containing the count of null values in dataset columns are shown .by ignoring these values in the further procedure leads to greater impact on the output. The results of missing values in the data frame columns are obtained by use of above syntax.

4.4.4. Removal of missing values

All these null values are filled using techniques called forward filling, backward filling, mean value, median value based upon the requirement these are followed. for this analysis I prefer forward filling and the syntax used is given below and check the null values after filling it.



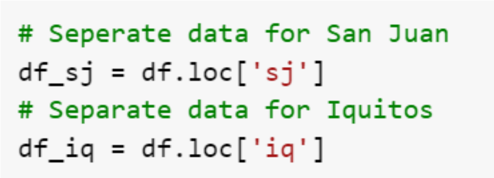
```
# Fill null values with forward-fill method
df_sj.fillna(method='ffill', inplace=True)
df_iq.fillna(method='ffill', inplace=True)
```

4.4.5. Rename Data in column.

After looking into the column of the city. the names of cities are named “San Juan” and “Iquitos” as “sj” and “Iq” in the table it looks odd to read so the syntax below is used to change the desire column name and thus data in the city column is renamed and easily identified for further data analysis.

4.4.6. Splitting of Data

The data is primarily divided between two cities, so we need to separate these features for the cities and analyze them accordingly so ". loc" function used to split the data and the below fig illustrates how the function is used for both the cities and successfully created two data frames.



```
# Seperate data for San Juan
df_sj = df.loc['sj']
# Separate data for Iquitos
df_iq = df.loc['iq']
```

4.4.7. Unit Conversion

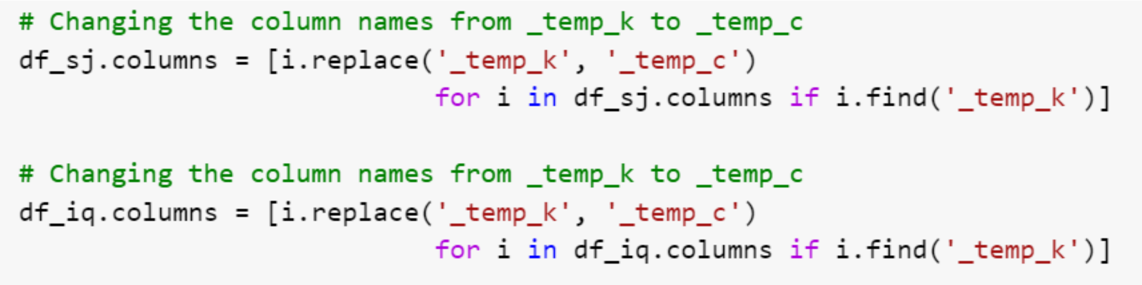
In the dataset there are a few temperature columns such as max_temp, min_temp, reanalysis, and dewpoint. All these values are noted in the kelvin scale and kelvin scale has bigger numerical scale compared to Celsius scale by using the formula below is used to change the format to Celsius.

$$\boxed{c = k - 273.15}$$

Where c denotes Celsius, k is kelvin.

4.4.8. Changing Column Names

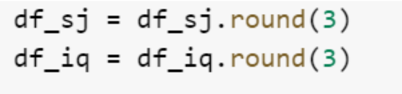
All the temperature columns are changed to Celsius scale, column name is changed such that it looks clear for further process with the help of "replace" command shown below all the temp parameters in the data frames of the both the cities are modified.in the below visual "I " is a variable checks for " _temp_k " if it is found replaces with " _temp_c ".



```
# Changing the column names from _temp_k to _temp_c
df_sj.columns = [i.replace('_temp_k', '_temp_c')
                           for i in df_sj.columns if i.find('_temp_k')]

# Changing the column names from _temp_k to _temp_c
df_iq.columns = [i.replace('_temp_k', '_temp_c')
                           for i in df_iq.columns if i.find('_temp_k')]
```

4.4.9. Rounding Values

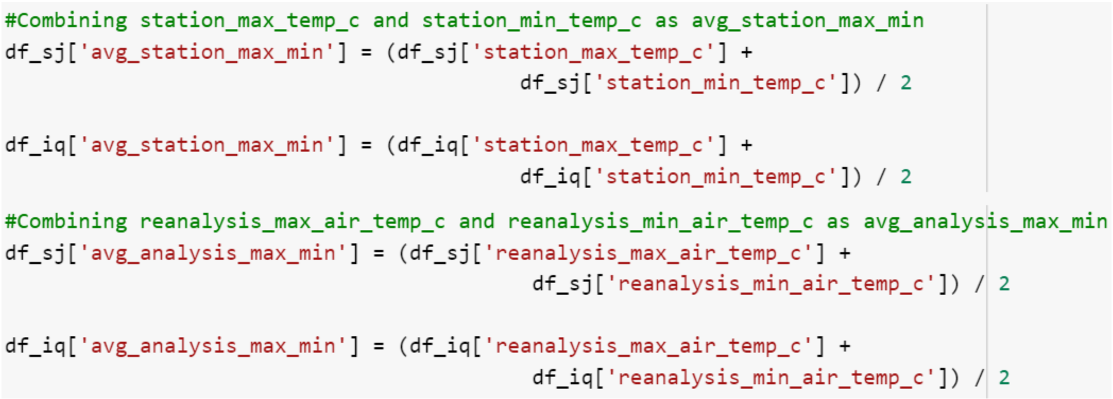
Some float values in two data frames of cities have big decimal values which will have a higher impact on analysis and all these values are shortened by using "round" syntax on the data frames. The fig below shows how the syntax is written on two cities data frames.



```
df_sj = df_sj.round(3)
df_iq = df_iq.round(3)
```

4.4.10. Average Column

We have max and min temperature for a station where the dengue case is recorded for that day, To average temperature for both the cities of Iquitos and San Juan and for that day the below syntax is used to get the values for required column and similarly average of reanalysis air temperature for both the cities is done. The syntax below is helpful for retrieving the required values.



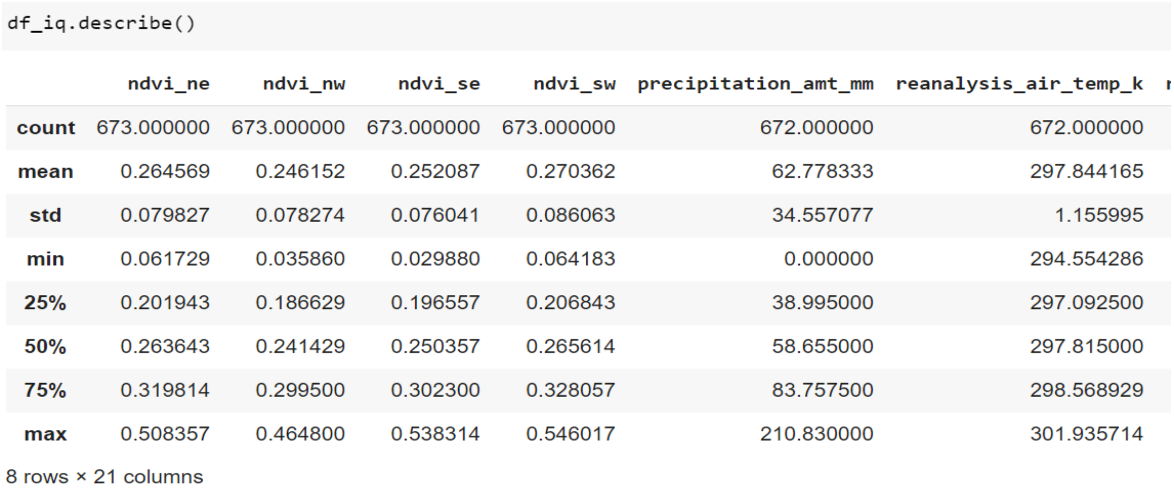
```
#Combining station_max_temp_c and station_min_temp_c as avg_station_max_min
df_sj['avg_station_max_min'] = (df_sj['station_max_temp_c'] +
                                            df_sj['station_min_temp_c']) / 2

df_iq['avg_station_max_min'] = (df_iq['station_max_temp_c'] +
                                            df_iq['station_min_temp_c']) / 2
#Combining reanalysis_max_air_temp_c and reanalysis_min_air_temp_c as avg_analysis_max_min
df_sj['avg_analysis_max_min'] = (df_sj['reanalysis_max_air_temp_c'] +
                                            df_sj['reanalysis_min_air_temp_c']) / 2

df_iq['avg_analysis_max_min'] = (df_iq['reanalysis_max_air_temp_c'] +
                                            df_iq['reanalysis_min_air_temp_c']) / 2
```

5.Exploratory Data Analysis

Exploratory Data Analysis helps to examine data without any preconceived notions or biases. It assists in identifying apparent mistakes and provides a better understanding of data patterns, highlighting outliers or unusual occurrences, and revealing noteworthy relationships among variables.

5.1. Descriptive statistics

Data was split and the command "describe" below gives the basic static summary of the cities Iquitos and San Juan. With the help of below command parameters such as mean, median, count and a few others are obtained with help of these some insights on the data will be obtained.



```
df_iq.describe()
```

	ndvi_ne	ndvi_nw	ndvi_se	ndvi_sw	precipitation_amt_mm	reanalysis_air_temp_k
count	673.000000	673.000000	673.000000	673.000000	672.000000	672.000000
mean	0.264569	0.246152	0.252087	0.270362	62.778333	297.844165
std	0.079827	0.078274	0.076041	0.086063	34.557077	1.155995
min	0.061729	0.035860	0.029880	0.064183	0.000000	294.554286
25%	0.201943	0.186629	0.196557	0.206843	38.995000	297.092500
50%	0.263643	0.241429	0.250357	0.265614	58.655000	297.815000
75%	0.319814	0.299500	0.302300	0.328057	83.757500	298.568929
max	0.508357	0.464800	0.538314	0.546017	210.830000	301.935714

8 rows × 21 columns

The above visual describes the descriptive statistics of Iquitos city with the parameters and analyzes the columns with help of these and this gives basic overview of the data.

5.2. Graphical Analysis

Graphical analysis plays a crucial role in machine learning by providing a visual representation of complex data sets, which allows for easier interpretation and understanding of patterns and relationships. It enables researchers and data scientists to identify trends, outliers, and correlations that might not be apparent through numerical or statistical analysis alone. Furthermore, graphical analysis can help with model selection, hyperparameter tuning, and feature engineering, as it can highlight which variables are most influential in predicting the outcome. Overall, graphical analysis is an essential tool for optimizing the accuracy and efficiency of machine learning models. A few graphs are plotted by changing parameters and input values. graphs hide the pattern of information and investigate it.

5.2.1. Total Cases Distribution for Both cities

The below fig illustrates how the dengue total cases are registered with the frequency and the visual speaks that city San Juan has the highest number of cases compare to Iquitos. Mostly the frequency 0-20 cases are registered and cannot see any cases above one hundred in Iquitos city, but it was not same in case of San Juan few cases like 200-400 frequency are registered. San Juan has the highest frequency because the cases are recorded from 1990 but Iquitos from two thousand.

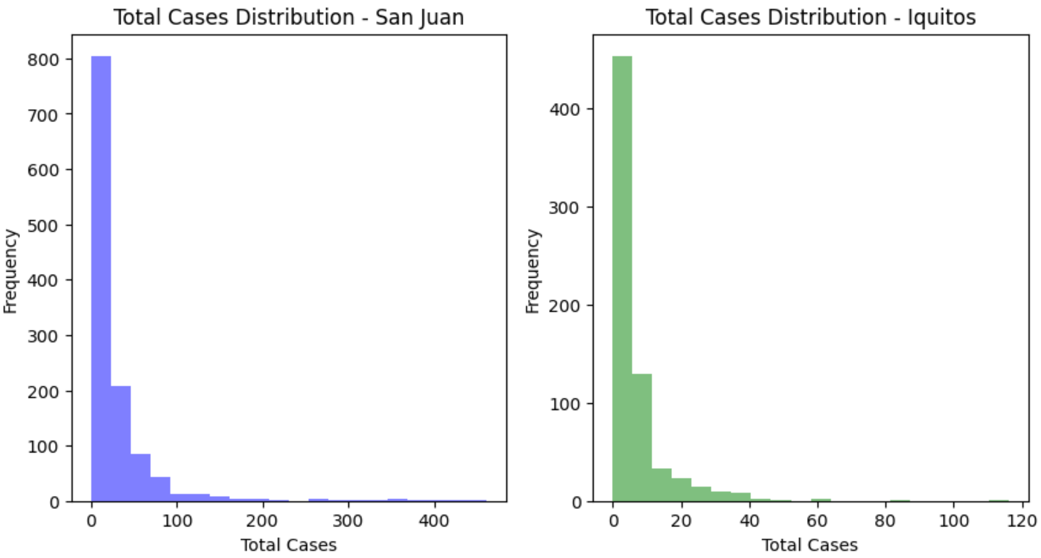


Figure 4: Total Dengue cases Distribution Frequency

5.2.2. Finding the year maximum number of cases

Iquitos city has dengue case registration from 2000 to 2013 and San Juan has from 1990 to 2013 and Iquitos city has the greatest number of cases are registered in 2008 and least number in 2000 but 1994 is the peak year for another city and 2013 is the least for San Juan. By observing the below graphs the first city has maximum peak and it is below one thousand cases over the years but other one is more than 6500 and over the years the total dengue cases are declining.

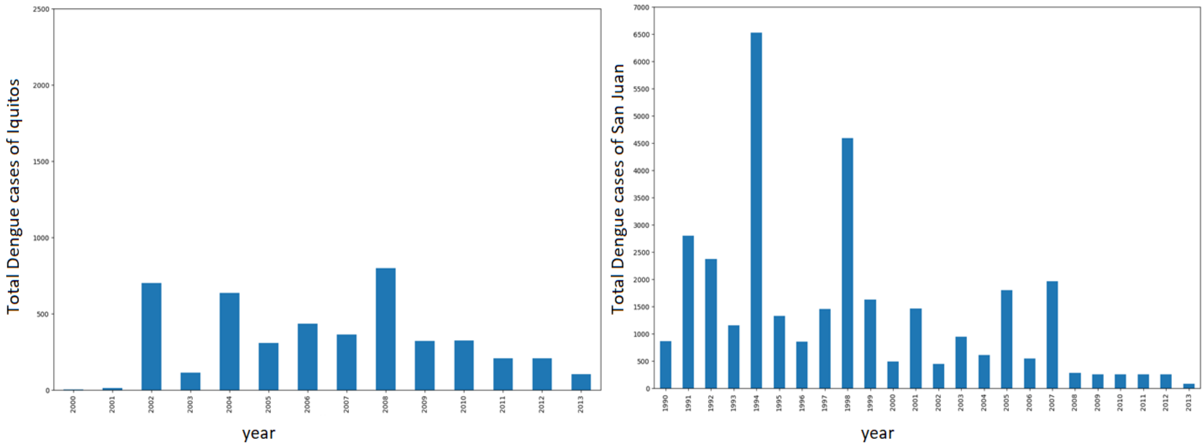


Figure 5: Total Dengue cases over the years

5.2.3. Correlation Matrix

Correlation analysis is a statistical method helps to determine the relationship between different variables. It helps in understanding how one variable affects another and to what extent. The correlation function generates scores that can range from -1 to +1, which reflect the degree of correlation between two variables. A score of +1 points as perfect positive correlation, while a score of -1 points as perfect negative correlation. A score of zero indicates no correlation.

One of the most common tools used to visualize correlation data is the correlation matrix, which is a table that displays correlation coefficients between different variables. The resulting correlation score reveals the strength of association between the variables under analysis. Heat maps are often used to visually represent correlation matrices, where the intensity of color represents the correlation between two variables.

Correlation matrices are important because they allow us to identify relationships between variables that may not be immediately apparent. By understanding the correlations between variables, we can make more informed decisions and predictions based on the data. This is especially important in fields such as finance, where understanding the relationships between different financial variables is critical to making effective investment decisions. Overall, correlation matrices are a valuable tool for understanding the complex relationships between variables and making informed decisions based on data.

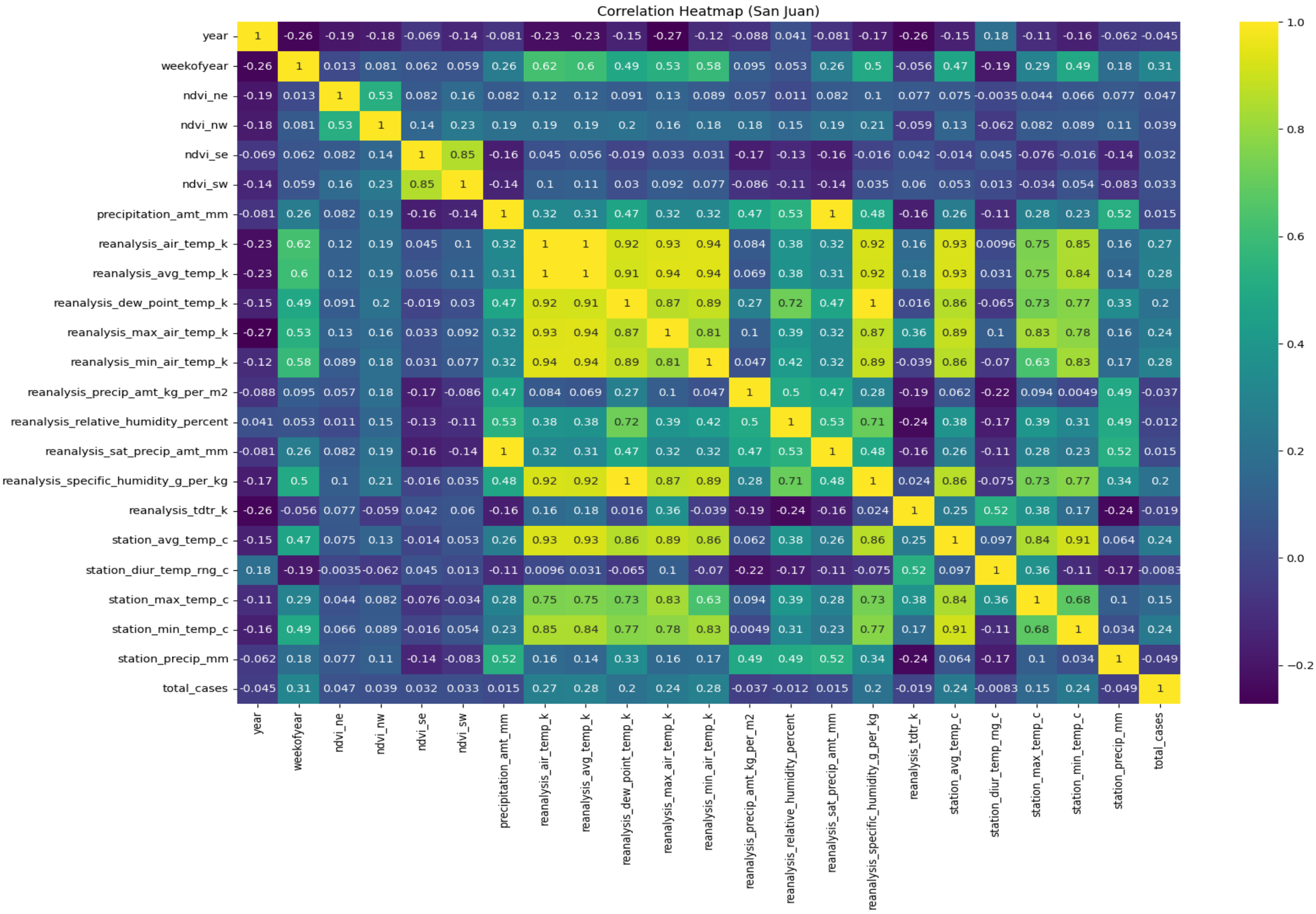


Figure 6: Correlation Matrix for San Juan city

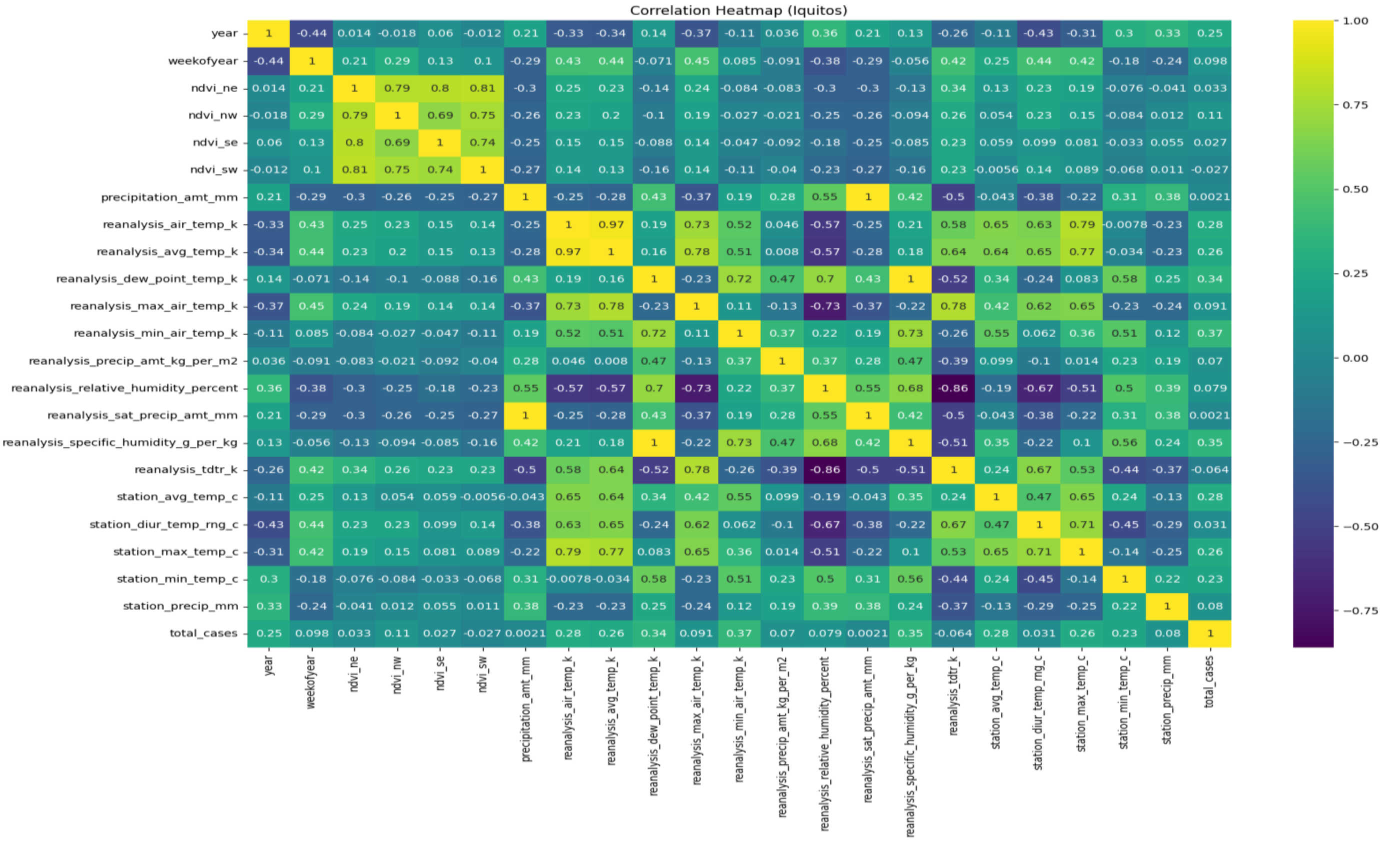


Figure 7: Correlation Matrix for Iquitos city

5.2.4. Average Temperature Distribution

The below graph was plotted between the distribution of temperature of dengue cases record versus frequency the first graph is San Juan city and other one is Iquitos by observing both the graphs data the fever registrations are done from the temperature ranging from greater than 20 degrees Celsius and less than 35 degrees .other than this range there was no record of cases .both the graphs have one common point that frequency of cases are peaking near at 28 degrees Celsius and declining after this temperature

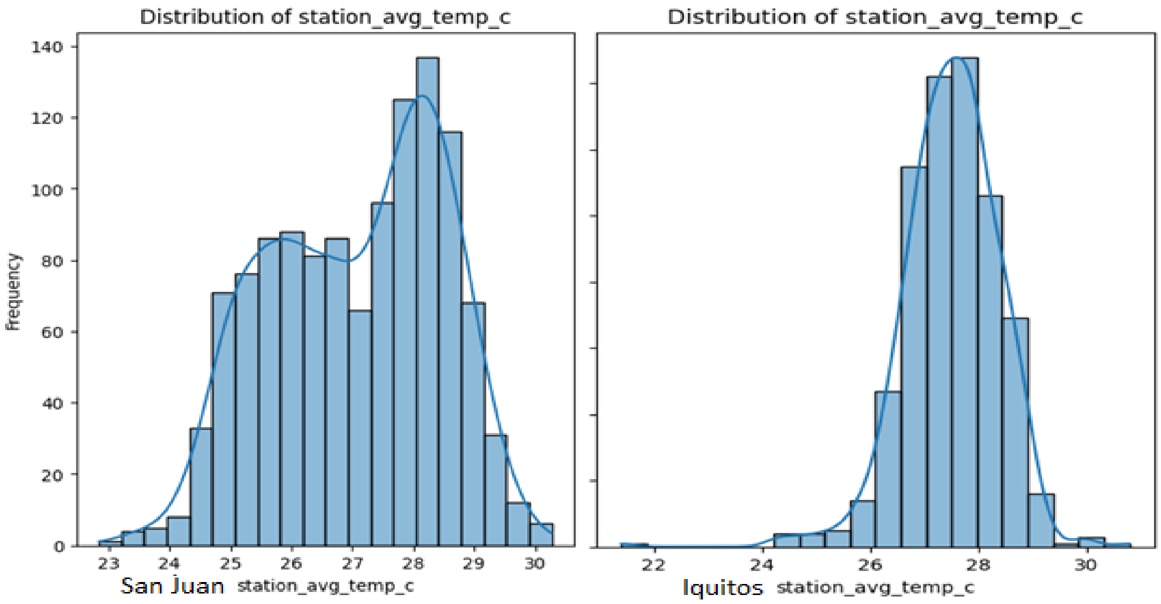


Figure 8: Dengue cases record Frequency On average Temperature of a day

5.2.5. Precipitation amount distribution

The term "precipitation amount per millimeter" pertains to the volume of water that falls from the atmosphere as precipitation in a specific location in relation to the unit of measurement, which is typically millimeters. Precipitation can come in the form of rain, snow, hail, or sleet and is quantified by measuring the amount of water that falls per millimeter. This metric is crucial in comprehending the weather trends of an area and plays a vital role in agriculture, hydrology, and water supply. The amount of precipitation per millimeter serves as a gauge to quantify the quantity of water that falls from the atmosphere and reaches the ground surface. It is a fundamental parameter in determining the water balance of a specific location as it impacts the moisture level of the soil, groundwater recharge, and availability of surface water. The quantity of precipitation per millimeter can significantly vary between different regions, seasons, and types of precipitation and can be influenced by several factors such as pressure, temperature, humidity, and topography. precipitation amount per millimeter is a crucial metric in comprehending the water cycle and climate patterns of an area. It is essential in managing water resources and predicting weather conditions, which is why meteorologists, hydrologists, and climate scientists consider it as a significant parameter. The below graph shows how dengue cases frequency are recorded with this parameter precipitation amount per mm .by observing both the graphs max dengue cases are recorded from 0-200 units after that there is negligible number of cases are recorded the above range is the key factor for promoting the surge of dengue cases.

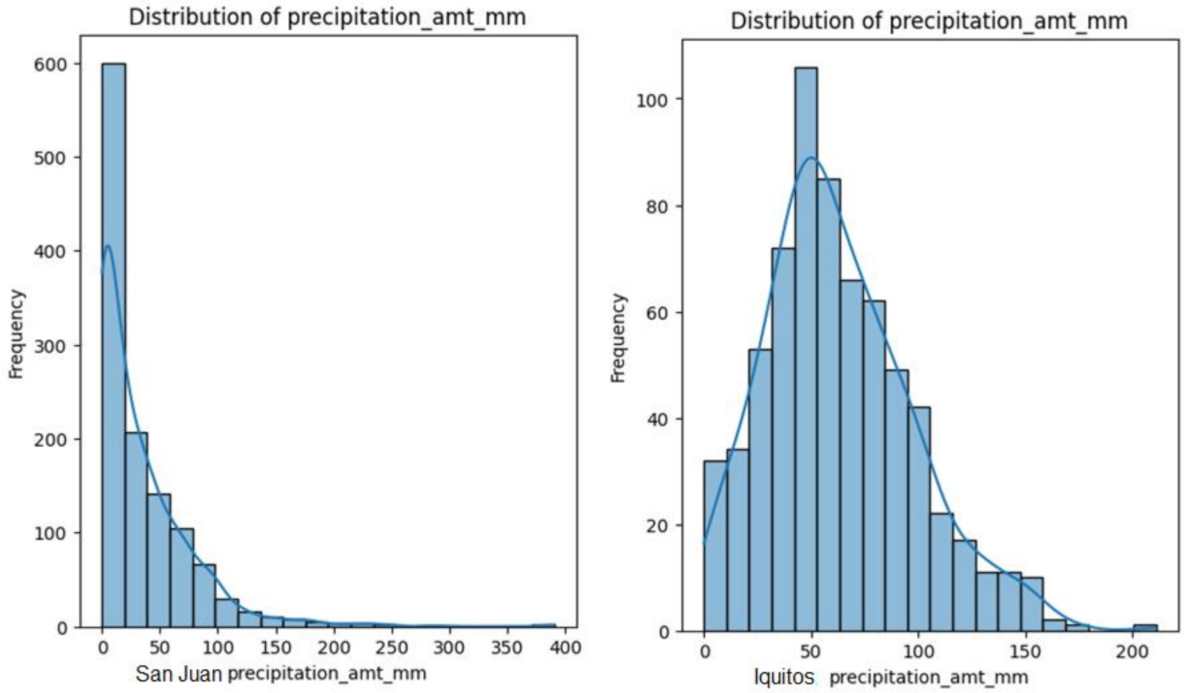


Figure 9: Dengue Registration frequency for precipitation

5.2.6. Week of the year with Total Dengue cases

In a year there are 52 weeks, but leap years have fifty-three. The below figure is plotted between the total dengue cases and week. The below visual shows that cases are recorded every week in the year. Start of the year there is a surge in the cases, but it is declined up to week twenty and from week20 to 40 there is a huge increment in the visual and then to end of the graph the diagram speaks that decrement of the total cases.

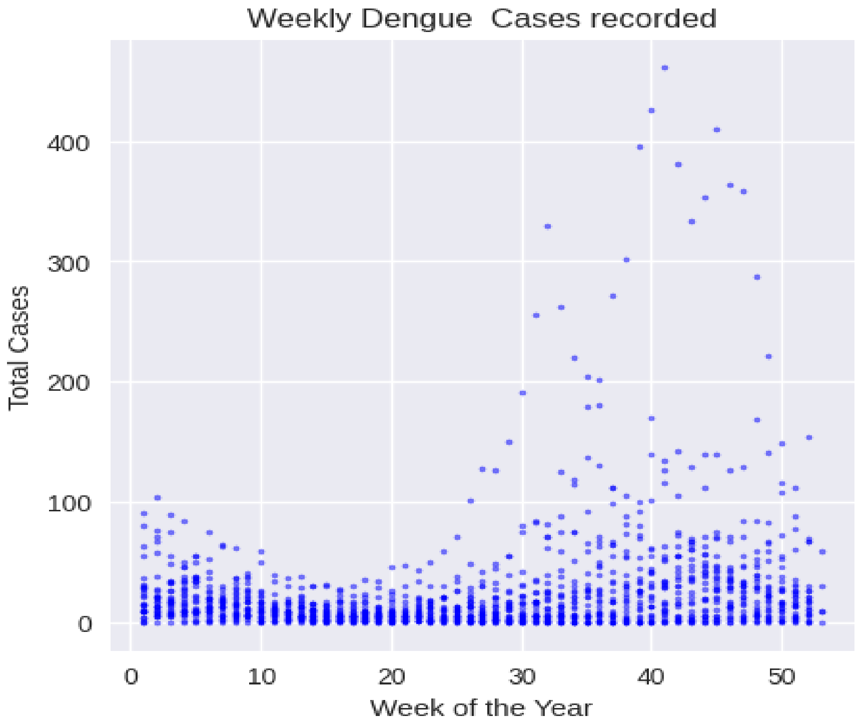


Figure 10: total dengue cases recorded on basis of every week in year.

6. Data preparing and model Building

6.1. Standardization

Machine learning techniques have been used to predict and control the spread of dengue fever. One crucial step in machine learning is the standardization of datasets.

Standardization is the process of transforming the data, so it has zero as mean and one as standard deviation. This transformation is important because many machine learning algorithms (Li, 2022)require that input data be standardized. Standardization can also help to improve the performance of some algorithms by reducing the impact of outliers and making the data easier to interpret.

The standardization of datasets for dengue fever has several uses. One of the most important uses is in predicting the occurrence and spread of dengue fever. Machine learning algorithms can be used on standardized datasets to identify patterns and relationships between different variables, such as weather conditions, population density, and mosquito abundance, and the occurrence and spread of dengue fever. These algorithms can then be used to predict the likelihood of an outbreak of dengue fever in a particular area, which can help public health officials to take preventive measures such as mosquito control and vaccination campaigns.

Another use of standardized datasets in dengue fever research is in identifying risk factors for severe disease outcomes. Some people who are infected with dengue fever develop severe symptoms such as dengue hemorrhagic fever or dengue shock syndrome, which can be life-threatening. Machine learning algorithms can be trained on standardized datasets to identify factors that increase the risk of severe disease outcomes, such as age, gender, and comorbidities. This data can be used to develop targeted interventions to reduce the risk of severe disease outcomes in high-risk populations.

In addition to these uses, standardized datasets can also help to improve the accuracy and reliability of machine learning models. Standardization can reduce the impact of outliers, which are data points that are significantly different from the rest of the data. Outliers can skew the results of machine learning algorithms and reduce their accuracy. Standardization can also make the data easier to interpret, which can help researchers to identify patterns and relationships more easily.

The advantages of standardizing datasets for dengue fever research are numerous. First, standardization can help to improve the machine learning algorithms performance. Many machine learning algorithms require that input data be standardized, and failure to standardize data can lead to inaccurate results. Standardization can also help to reduce the impact of outliers, which can improve the accuracy and reliability of the model.

Second, standardization can make the data easier to interpret. By transforming data so that it means is zero and a standard deviation of one, the data can be more easily compared and analyzed. This can help researchers to identify patterns and relationships between different variables, which can lead to new insights and discoveries.

Third, standardized datasets can facilitate collaboration and data sharing among researchers. By standardizing datasets, researchers can ensure that their data is compatible with other datasets, which can facilitate data sharing and collaboration. This can help to accelerate the pace of research and lead to new discoveries and insights.

In conclusion, the standardization of datasets for dengue fever research is a major step in machine learning. Standardization can help to increase the performance of machine learning models, reduce the impact of outliers, and make the data easier to interpret. Standardized datasets can also be used to predict the occurrence and spread of dengue fever, identify risk factors for severe disease outcomes, and improve the accuracy and reliability of machine learning algorithms. The advantages of standardizing datasets for dengue fever research are numerous and can lead to new insights and discoveries in the field.

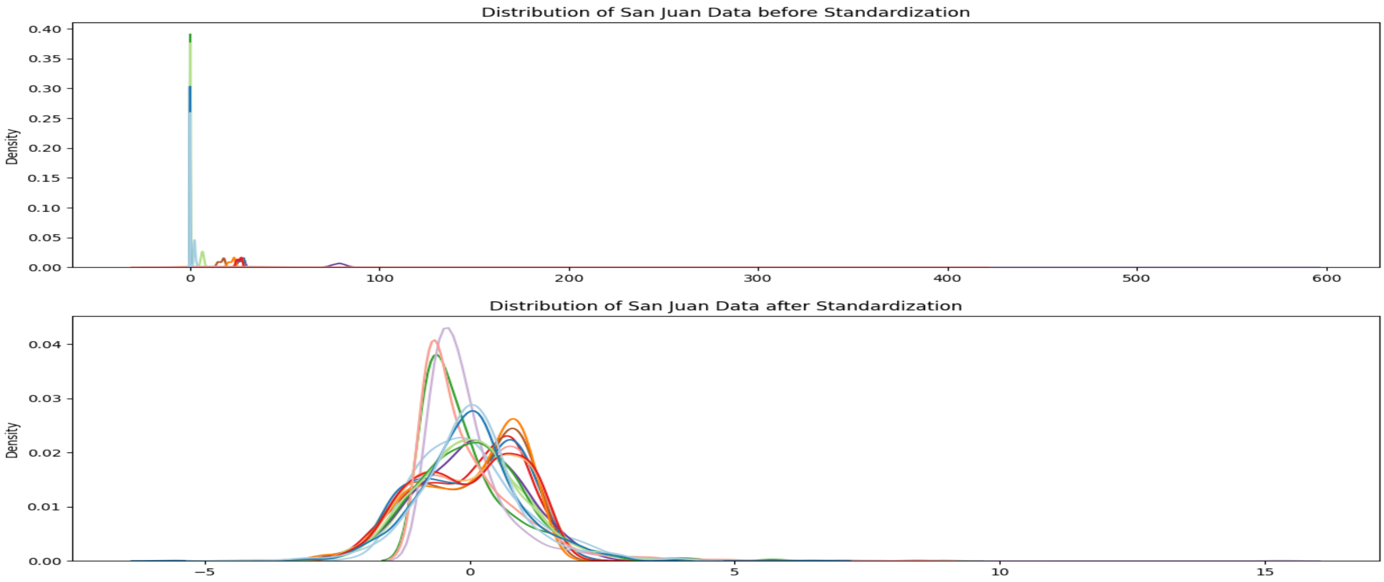


Figure 11: San Juan City Data before and after Standardization

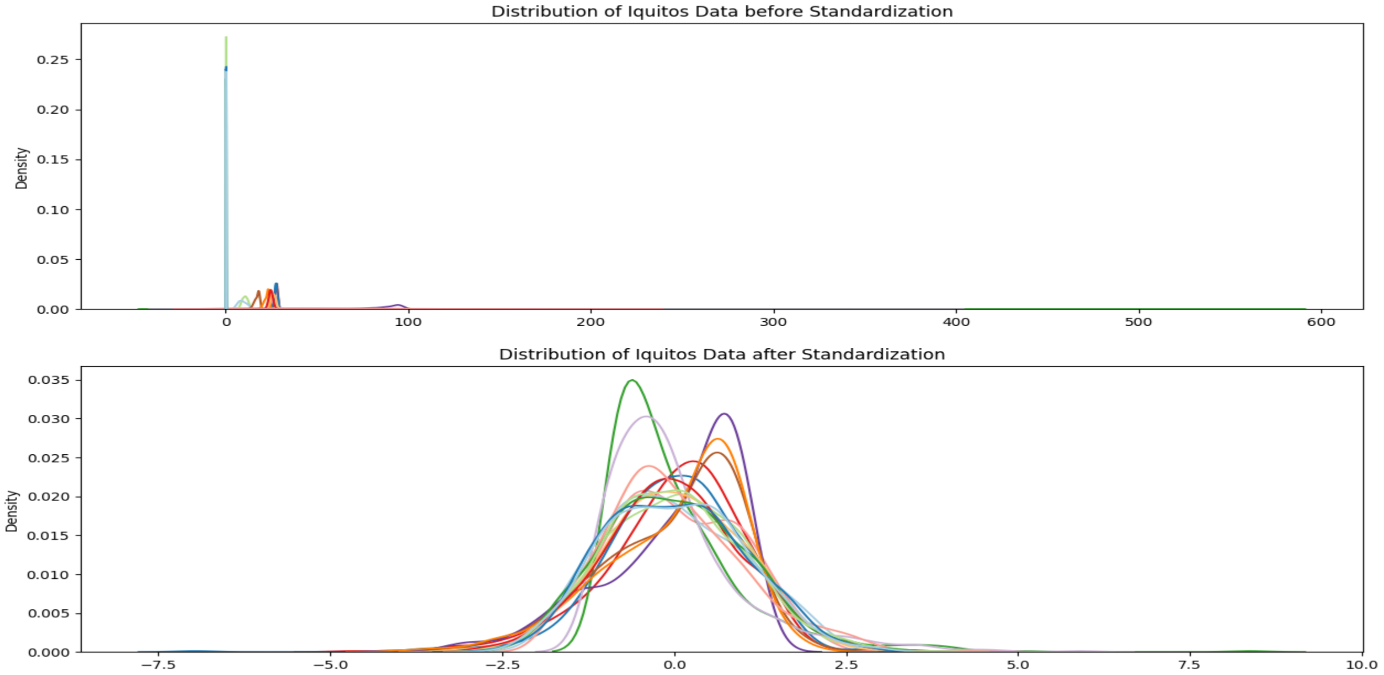


Figure 12: Iquitos City Data before and after Standardization

The above figures are the data of both the cities San Juan and Iquitos, because of the outliers and huge variance in the data we can see the data difference in both the cities visual. The chance error in the result is high so Standardization Technique is used to tackle this issue. After applying the procedure, the data is completely transformed.

6.2. Dependent and independent variables

In this process, X has the information is viewed as the dimensions of the data, while the y has the information depicts the behavior of the independent variable’s dimensions. The target variable in a dataset could be any column that the machine learning model seeks to predict. In the case of the given dataset, the target variable is 'Total cases.' Therefore, the y value is assigned to the 'Total cases' column. The 'Total cases' column is removed from the dataset, which then forms a new data frame that can be assigned to X, representing the dimensions of the independent variables. This division of data into dependent and independent variables is essential as it allows the machine learning model to distinguish between the input and the output variables. This separation helps the model to focus on the independent variables, which have a direct impact on the output variable. The machine learning model uses independent variables to create a mathematical model that predicts the target variable, which is the dependent variable. By providing the input data, which is the independent variables, to the model, the model can produce an output that predicts the target variable accurately. Furthermore, the splitting of the dataset into dependent and independent variables helps in the interpretation of the model's results. It allows us to identify which independent variables have the most significant impact on the target variable. This information is crucial in understanding the relationships between different variables in the dataset and can assist in making informed decisions.

6.3. Data split - Train and Test

Dividing a dataset into a train set and a test set is an essential technique to evaluate the performance of a machine learning model. This method is particularly useful to assess about the model ability to generalize to new data and to prevent overfitting issues. To apply this technique to the dengue fever dataset, so to utilize the train_test_split method available in the scikit-learn library.

This method requires three arguments: the independent variables (features), the dependent variable (target), and the size of the test set. By specifying the random_state parameter, to ensure that the data is split consistently and reproducibly. In this case, to split the dengue fever cities dataset into an 80:20 ratio, where 80 percentage of the data will be used for train and the remaining 20 percentage is for test. This approach can help to train the model on a substantial amount of data and evaluate its performance using a considerable number of data points.

7.Building Models

In machine learning, building models is essential for making accurate predictions and automated decision-making. Models are trained on data sets to recognize patterns and make informed predictions on new data. These models can be used for a wide range of applications, such as image classification, natural language processing, and recommendation systems. Models help automate decision-making and improve accuracy, efficiency, and scalability of various tasks in machine learning.

7.1 Random Forest model

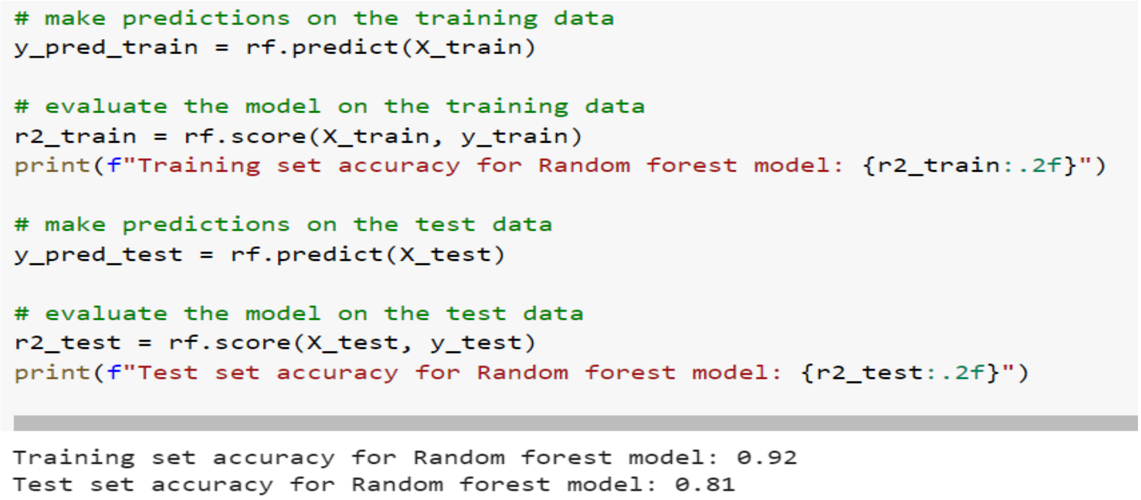
Random forest is a popular machine learning algorithm that is widely used for classification and regression tasks. It is an ensemble learning method that combines multiple decision trees to improve the accuracy and robustness of the predictions (Speiser, 2019). The algorithm works by building a forest of decision trees on different subsets of the training data and then combining the results of each tree to make a final prediction.

To build an accurate and dependable random forest model, it is important to evaluate and train the data. The training data is used to build the decision trees and the testing data is used to evaluate the performance of the model. One of the key aspects of testing and training data is to ensure that they are independent and identically distributed.

For a random forest model to be effective, it is important that the samples used for training and testing accurately represent the same underlying population and are not biased in any way. To assess the performance of the model, one common technique is to employ cross-validation methods like k-fold cross-validation. This approach involves splitting the dataset into k equal parts, with k-1 parts being used for training and the remaining part for testing. The process is then repeated k times, with each part being used for testing once.

The model performance is then evaluated by calculating the average error or accuracy over all k iterations. In conclusion, testing and training data are critical components of building an effective random forest model. Selecting and preparing the data can significantly impact the performance and accuracy of the model. Cross-validation techniques can be used to evaluate the model and evaluate its performance, ensuring that the model is robust and dependable.

After splitting the data into test and train to predict the total cases as y variable and all the dependent variables as x. With the help of columns year and week of year so that to predict the total cases. The below syntax is helpful for running trained data and test data on the algorithm.



```
# make predictions on the training data
y_pred_train = rf.predict(X_train)

# evaluate the model on the training data
r2_train = rf.score(X_train, y_train)
print(f"Training set accuracy for Random forest model: {r2_train:.2f}")

# make predictions on the test data
y_pred_test = rf.predict(X_test)

# evaluate the model on the test data
r2_test = rf.score(X_test, y_test)
print(f"Test set accuracy for Random forest model: {r2_test:.2f}")
```

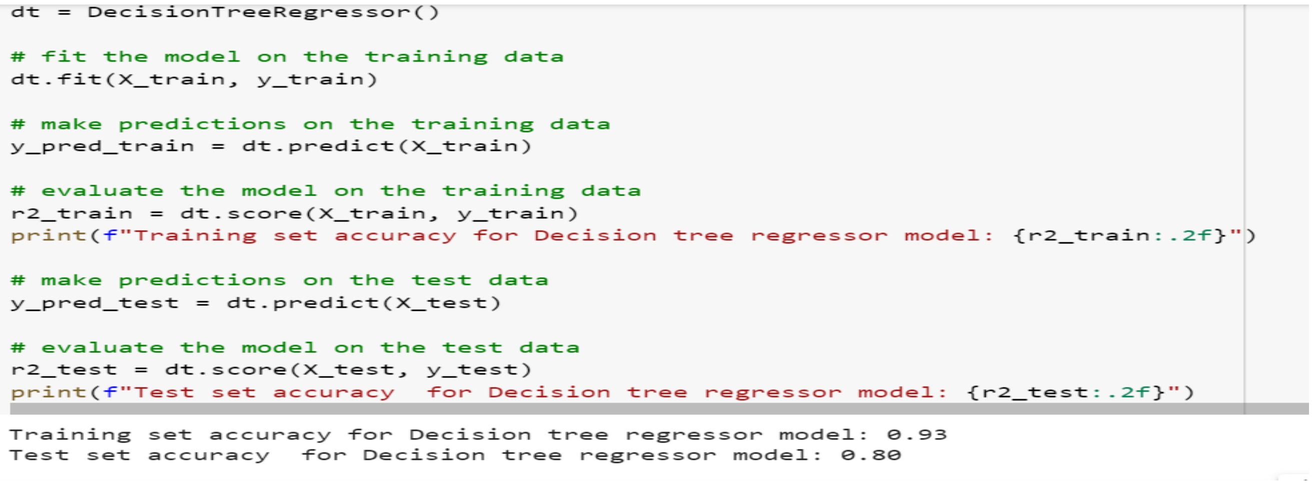
```
Training set accuracy for Random forest model: 0.92
Test set accuracy for Random forest model: 0.81
```

7.2 Decision Tree classifier

A decision tree classifier is a machine learning approach that can perform both classification and regression tasks. The classifier builds a tree model that captures decisions and their respective outcomes. The tree comprises nodes that represent individual decisions and branches that depict the possible outcomes of those decisions. The process involves dividing the data into subsets based on the most relevant features that can effectively differentiate the classes or forecast the target variable, which is performed recursively. To train a decision tree classifier (Safavian, 1991), a dataset with labeled examples is needed. The dataset is split into training and testing sets, with most of the data being used for training and a smaller portion being used for testing. The training data is used to build the decision tree by recursively splitting the data based on the features that best separate the classes or predict the target variable. The splitting process continues until the data cannot be further subdivided, or a stopping criterion is met.

It is important to avoid building a decision tree classifier that is overly complex and captures noise in the training data instead of the underlying patterns. Techniques like pruning and setting stopping criteria can help prevent this. Once the decision tree is constructed, it can be utilized to make predictions on new data. The algorithm works by traversing the tree from the root node to a leaf node based on the input feature values. At each node, a decision is made based on a specific feature value, and the traversal continues down the corresponding branch until a leaf node is reached. The leaf node represents a predicted target value or class, and this serves as the output of the decision tree classifier.

Decision tree classifiers are a powerful and widely used machine learning algorithm that can be used for both classification and regression tasks. Selecting and preparing the training and testing data is crucial for building an effective decision tree classifier. The algorithm works by constructing a tree-like model of decisions and their consequences, and it can be used to make predictions on new data by traversing the tree from the root node to a leaf node. The below syntax is used on trained and testing to get the accuracy of model on the data.



```
dt = DecisionTreeRegressor()

# fit the model on the training data
dt.fit(X_train, y_train)

# make predictions on the training data
y_pred_train = dt.predict(X_train)

# evaluate the model on the training data
r2_train = dt.score(X_train, y_train)
print(f"Training set accuracy for Decision tree regressor model: {r2_train:.2f}")

# make predictions on the test data
y_pred_test = dt.predict(X_test)

# evaluate the model on the test data
r2_test = dt.score(X_test, y_test)
print(f"Test set accuracy  for Decision tree regressor model: {r2_test:.2f}")
```

```
Training set accuracy for Decision tree regressor model: 0.93
Test set accuracy  for Decision tree regressor model: 0.80
```

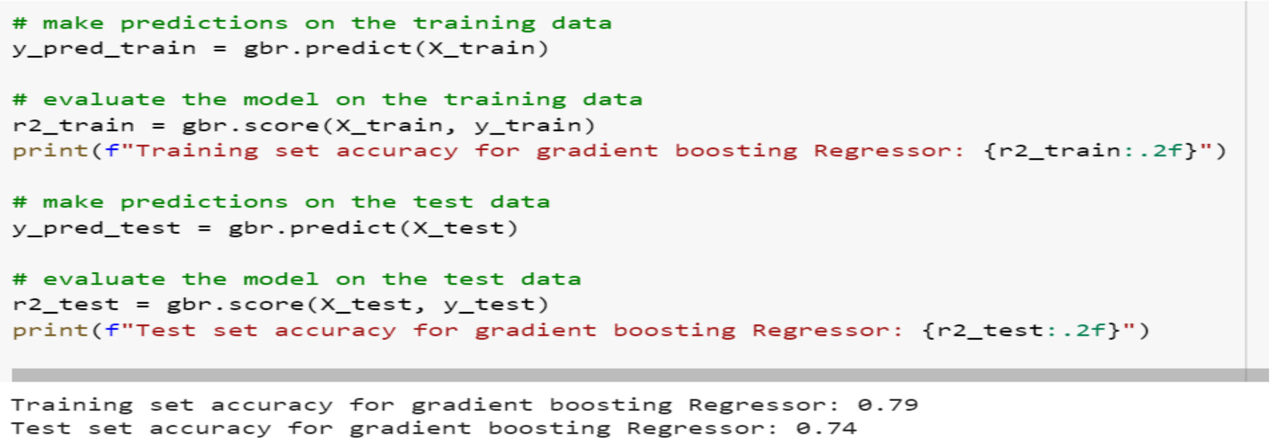
7.3 Gradient Boosting Regressor

Gradient boosting is a powerful and popular machine learning technique that is used to improve the accuracy and predictive power of models. It works by combining multiple weak learners to create a strong learner. (El.Elhola, 2022) The algorithm is based on the concept of gradient descent, which is used to minimize the loss function of the model. At the beginning of the process, the algorithm initializes the model with a weak learner, such as a decision tree or a linear regression model. The model is trained on the training data, and the errors of the predictions are calculated using a loss function.

In the next iteration, the algorithm trains a new weak learner on the errors of the previous learner. The new learner is trained to minimize the errors of the previous learner, rather than the original loss function. This process is repeated until the desired number of learners are added to the model.

The resulting model is an amalgamation of all the weak learners, with each learner assigned a weight based on its performance on the training data. The algorithm is referred to as gradient boosting because it employs gradient descent to minimize the loss function.

One of the significant benefits of using gradient boosting is its capability to manage intricate and non-linear relationships in the data. Additionally, it is less prone to overfitting, which may arise when the model is too intricate and captures the noise in the training data.



```
# make predictions on the training data
y_pred_train = gbr.predict(X_train)

# evaluate the model on the training data
r2_train = gbr.score(X_train, y_train)
print(f"Training set accuracy for gradient boosting Regressor: {r2_train:.2f}")

# make predictions on the test data
y_pred_test = gbr.predict(X_test)

# evaluate the model on the test data
r2_test = gbr.score(X_test, y_test)
print(f"Test set accuracy for gradient boosting Regressor: {r2_test:.2f}")
```

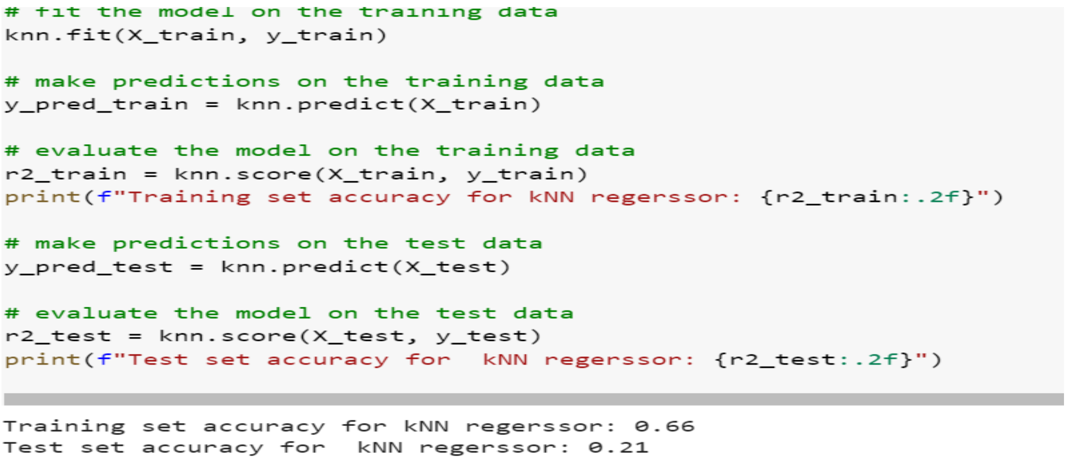
```
Training set accuracy for gradient boosting Regressor: 0.79
Test set accuracy for gradient boosting Regressor: 0.74
```

7.4 KNN Regressor

KNN (k-nearest neighbors) is a popular machine learning algorithm used for both classification and regression tasks. The KNN regressor (Song, 2017)is used for regression tasks, where the goal is to predict a continuous target variable. The algorithm works by finding the k closest neighbors to a new data point in the training set, and then predicting the target value as the average or weighted average of the target values of those neighbors.

To utilize the KNN regressor, a labeled training dataset is necessary. The dataset is partitioned into training and testing sets, with most of the data employed for training and a smaller subset used for testing. The algorithm operates by calculating the distances between the novel data point and all the points within the training set. Subsequently, the k nearest points to the new data point are chosen, and their target values are utilized to forecast the target value of the new data point. One of the key challenges in using the KNN regressor is selecting the value of k. A small value of the k may result with overfitting, where the model is too complex and captures the noise in the training data. The large value of k may result in underfit, where the model is too simple and does not capture the underlying patterns in data.

The KNN regressor is a simple but powerful machine learning algorithm used for regression tasks. The algorithm works by finding the k closest neighbors to a new data point in the training set, and then predicting the target value as the average or weighted average of the target values of those neighbors. Selecting the value of k is important for building an effective KNN regressor. The below syntax is used to obtain the accuracy score of both testing and training data sets.



```
# fit the model on the training data
knn.fit(X_train, y_train)

# make predictions on the training data
y_pred_train = knn.predict(X_train)

# evaluate the model on the training data
r2_train = knn.score(X_train, y_train)
print(f"Training set accuracy for kNN regerssor: {r2_train:.2f}")

# make predictions on the test data
y_pred_test = knn.predict(X_test)

# evaluate the model on the test data
r2_test = knn.score(X_test, y_test)
print(f"Test set accuracy for  kNN regerssor: {r2_test:.2f}")
```

```
Training set accuracy for kNN regerssor: 0.66
Test set accuracy for  kNN regerssor: 0.21
```

8.Results and Analysis

Data is split into train and test cases and a few machine learning algorithms are run on these models after the execution some results and accuracy are shown below. The below Tables show the Model and their accuracy score.

Model	Train Accuracy	Test Accuracy
Random Forest Model	94%	80%
Decision Tree classifier	93%	75%
Gradient Boosting Regressor	79%	62%
KNN Regressor	66%	37%

Table 3: Machine Learning models Accuracy

After looking into the accuracy table Random Forest method has the most accuracy among them the below graph is drawn between test cases of actual versus predicted blue line stroke indicates the actual test values and orange stroke indicates the predicted value. most cases of predicted values are aligning with original cases, so this method is the best among other algorithms.

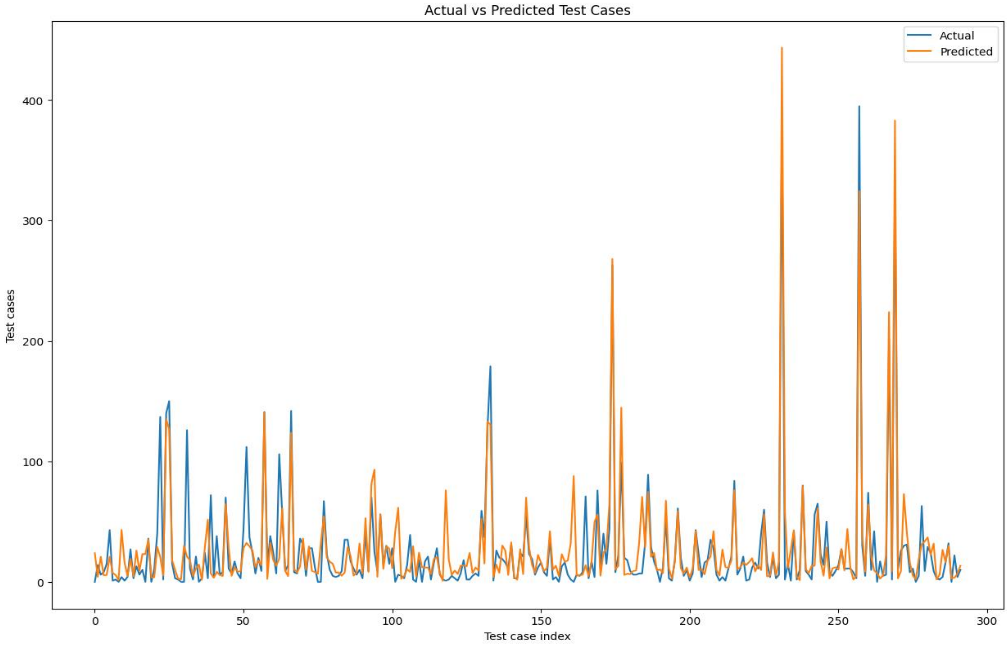


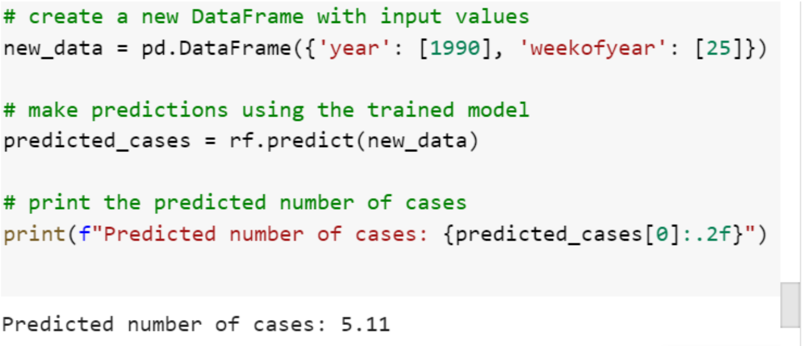
Figure 13: Actual cases vs predicted cases using Machine learning.

8.1 Test cases

Random forest model has high accuracy so run the testing data on this model.

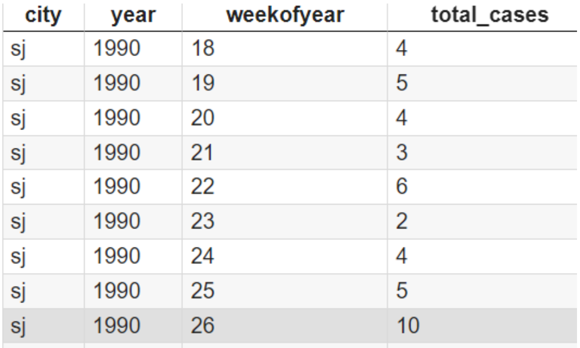
By applying the input values shown in the below figure, for the first case the ML model predicted ‘5.11’ for 5 cases inputs year as ‘1990’ and week of year as ‘25’ and 29.43 for 29 cases inputs given are year as ‘1992’ and week of year as ‘13’ which is almost correct in both the test cases worked perfectly

Test case1



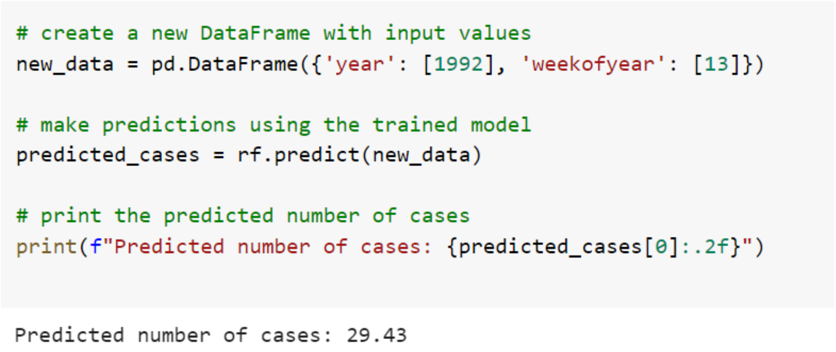
```
# create a new DataFrame with input values
new_data = pd.DataFrame({'year': [1990], 'weekofyear': [25]})

# make predictions using the trained model
predicted_cases = rf.predict(new_data)

# print the predicted number of cases
print(f"Predicted number of cases: {predicted_cases[0]:.2f}")
```

```
Predicted number of cases: 5.11
```



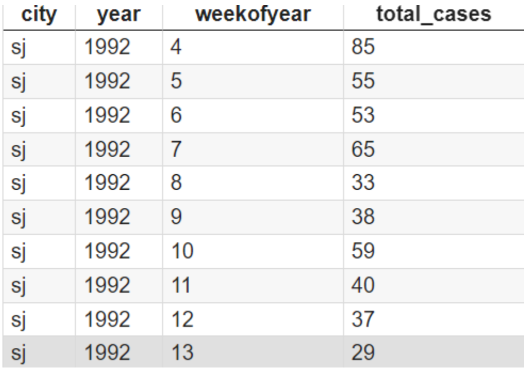
city	year	weekofyear	total_cases
sj	1990	18	4
sj	1990	19	5
sj	1990	20	4
sj	1990	21	3
sj	1990	22	6
sj	1990	23	2
sj	1990	24	4
sj	1990	25	5
sj	1990	26	10

Test case 2



```
# create a new DataFrame with input values
new_data = pd.DataFrame({'year': [1992], 'weekofyear': [13]})

# make predictions using the trained model
predicted_cases = rf.predict(new_data)

# print the predicted number of cases
print(f"Predicted number of cases: {predicted_cases[0]:.2f}")
```

```
Predicted number of cases: 29.43
```



city	year	weekofyear	total_cases
sj	1992	4	85
sj	1992	5	55
sj	1992	6	53
sj	1992	7	65
sj	1992	8	33
sj	1992	9	38
sj	1992	10	59
sj	1992	11	40
sj	1992	12	37
sj	1992	13	29

9.Conclusion

This epidemic disease causing not only hospitalization and deaths but also making victims suffer for extended period and recovery is also slow and making economic burden in the developing countries of tropical region and this makes obstacle for the socio-economic sector this is why research on this fever is necessary. The above analysis gave the factors influencing the spread of disease. The first discovery of the virus was in 20th century and then from years of mutation the infection changed its structure and evolved into many forms like Corona virus in 2019 to 2023 more than 21 mutations are seen similarly so many evolutions are seen in this case too and few research studies says that new infection virus of DENV5 also evolved. The above prediction of dengue cases done in this research using random forest from the data of 1990 to 2010 in the cities of Iquitos and San Juan united states of America. With the above Data analysis, we have seen the factors promoting the virus spread, but these are not fully evident because the new updated virus may have varied reasons to spread and still scientists have not proved why it is not fatal to all the patients. This Research is helpful for set of people Living in those conditions and applicable to those for a set of time and conditions but not globally. In future to predict the future spread of virus the previous data of patients and their living scenario, environmental conditions must be updated and research on this topic must be continued to see the breakthrough for this virus treatment or the manufacture of vaccine.

References

Alera, M. T., 2016. Incidence of Dengue Virus Infection. *PLOS.*

Barbosa, H. G. & Sandra, M., 2020. Dengue Infections in Colombia: Epidemiological Trends of a Hyperendemic Country. *MDPI,* 5(4).

Dash, P. k., Sharma, S. & soni, M., 2013. Complete genome sequencing and evolutionary analysis of DENV2. *Biochemical and Biophysical Research Communication,* 436(5), pp. 478-485.

Dowd, K. A., 2015. Genotypic Differences in Dengue Virus Neutralization Are Explained by a Single Amino Acid Mutation That Modulates Virus Breathing. *American society for micro biology,* 6(6).

El.Elhola, W., 2022. Performance Analysis of Brain Tumor Detection. *ieee,* pp. 23-25.

Guha-Sapir, D., 2005. new paradigms for a changing epidemiology. *Bio-medical central.*

Harapan, 2018. Knowledge, attitude, and practice regarding dengue virus infection. *Bio medical central infectious diseases,* 18(96).

Lai, Y.-L., 2007. Cost-Effective Real-Time Reverse Transcriptase PCR (RT-PCR) To Screen for Dengue Virus followed by Rapid Single-Tube Multiplex RT-PCR for Serotyping of the Virus. *Journal of clinical biology,* 45(3).

Li, J., 2022. standardized use inspection of workers' personal protective equipment based on deep learning. *safety science,* Volume 150.

M.s, M. & S, J., 2015. Discovery of fifth serotype of dengue virus. *sciencedirect,* pp. 67-70.

Rangarajan, P., 2019. Forecasting dengue and influenza incidences using a sparse representation of Google trends. *Plos computational biology.*

Renantha, R. R., 2022. Flavonoids as potential inhibitors of dengue virus 2 (DENV2). *Journal of Pharmacy & Pharmacognosy Research,* 10(4), pp. 660-675.

Safavian, S., 1991. A survey of decision tree classifier methodology. *ieee,* 21(3), pp. 671-674.

Singha, 2016. Expanding antigen-specific regulatory networks to treat auto immunity. *Nature,* pp. 434-440.

Song, Y., 2017. An efficient instance selection algorithm for k nearest neighbor regression. *Nuero computing,* Volume 251, pp. 26-34.

Speiser, J. L., 2019. A comparison of random forest variable selection methods for classification prediction modeling. *science direct,* Volume 134, pp. 93-101.

Ubol, S., 2010. Mechanisms of Immune Evasion Induced by a Complex of Dengue Virus and Preexisting Enhancing Antibodies. *Infectious diseases,* 201(6), pp. 923-935.

Webster, D. P., 2009. progress towards a dengue vaccine. *Lancet infectious diseases,* 9(11), pp. 678-687.