“COVID-19 OUTBREAK PREDICTION”

A

Project Report

submitted

in partial fulfillment

for the award of the Degree of

Bachelor of Technology

in Department of Computer Science and Engineering

Project Coordinator:
Mr. Ankit Kumar
Asooc. Prof.

Submitted By :
Ankit Saini
17ESKCS027

Department of Computer Science and Engineering
Swami Keshvanand Institute of Technology, M & G, Jaipur
Rajasthan Technical University, Kota
Session 2020-2021

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that Mr Ankit Saini, a student of B.Tech (Computer Science & Engineering) VIIth semester has submitted his Project Report entitled Covid-19 Outbreak Prediction under my guidance.

Project Coordinator

Mr. Ankit Kumar

Assoc. Prof.

DECLARATION

We hereby declare that the report of the project entitled Covid-19 Outbreak Prediction is a record of an original work done by us at Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur under the coordination of Mr. Ankit Kumar (Dept.of Computer Science and Technology). This project report has been submitted as the proof of original work for the partial fulfillment of the requirement for the award of the degree of Bachelor of Technology (B.Tech) in the Department of Computer Science and Technology.It has not been submitted anywhere else, under any other program to the best of our knowledge and belief.

Ankit Saini

17ESKCS027

Acknowledgement

It is my pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced my thinking, behavior, and acts during the course of study.

I express my sincere gratitude to **Prof. Dr. Mukesh kumar Gupta**, HOD,(Computer Science) for providing me an opportunity to work in a consistent direction and providing all necessary means to complete my presentations and report thereafter.

I would like to thank my Project Coordinator **Mr. Ankit Kumar** ,Department of Computer Science & Engineering ,Swami Keshvanand Institute of Technology, Management and Gramothan ,Jaipur for her valuable suggestion,keen interest,constant encouragement,incessant inspiration and continuous help throughout this work.Her excellent guidance has been instrumental in making this work a success.

I express my sincere heartfelt gratitude to all the staff members of Department of Computer Science & Engineering who helped me directly or indirectly during this course of work.

I would also like to express my thanks to my parents for their support and blessings. A special thank goes to all my friends for their support in completion of this work.

Ankit Saini
17ESKCS027

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction to Project

Machine learning (ML) has proved itself as a prominent field of study over the last decade by solving many very complex and sophisticated real-world problems. The application areas included almost all the real-world domains such as health-care, autonomous vehicle , business applications, natural language processing , intelligent robots, gaming, climate modeling, voice, and image processing. One of the most significant areas of ML is forecasting , numerous standard ML algorithms have been used in this area to guide the future course of actions needed in many application areas including weather fore-casting, disease forecasting, stock market forecasting as well as disease prognosis. Various regression and neural network models have wide applicability in predicting the conditions of patients in the future with a specific disease. There are lots of studies performed for the prediction of different diseases using machine learning techniques such as coronary artery disease , cardiovascular disease prediction , and breast cancer prediction. In particular, the study is focused on live forecasting of COVID-19 confirmed cases and study is also focused on the forecast of COVID-19 outbreak and early response. These prediction systems can be very helpful in decision making to handle the present scenario to guide early interventions to manage these diseases very effectively.

1.2 Objective

This study aims to provide an early forecast model for the spread of novel coronavirus, also known as SARS-CoV-2,officially named as COVID-19 by the World Health Organization (WHO) . COVID-19 is presently a very serious threat to human life all over the world. At the end of 2019,the virus was first identified in a city of China called Wuhan,when a large number of people developed symptoms like pneumonia . It has a diverse effect on the human body,including severe acute respiratory syndrome and multiorgan failure which can ultimately lead to death in a very short duration . Millions of people are affected by this pandemic throughout the world with thousands of deaths every coming day. Thousands of new people are reported to be positive every day from countries across the world. The virus spreads primarily through close person to person physical contacts, by respiratory droplets, or by touching the contaminated surfaces. The most challenging aspect of its spread is that a person can possess the virus for many days without showing symptoms. The causes of its spread and considering its danger, almost all the countries have declared either partial or strict lockdowns throughout the affected regions and cities. Medical researchers throughout the globe are currently involved to discover an appropriate vaccine and medications for the disease. Since there is no approved medication till now for killing the virus so the governments of all countries are focusing on the precautions which can stop the spread. Out of all precautions, “be informed” about all the aspects of COVID-19 is considered extremely important. To contribute to this aspect of information, numerous researchers are studying the different dimensions of the pandemic and produce the results to help humanity.

1.3 Proposed Solution

The forecasting is done for the two important variables of the disease : 1) the number 0f New con-firmed cases. 2) the number of death cases . This problem of forecasting has been considered as a regression problem in this study, so the study is based on some state of the art supervised ML regression models such as Random Forest and XGBoost. The learning models have been trained using the COVID-19 patient stats dataset provided by Johns Hopkins. The dataset has been preprocessed and divided into two subsets: training set (85 percent records) and testing set (15 percent records). The performance evaluation has been done in terms of important measures including R-squared-score (R2score) and root mean square error (RMSE).

1.4 Scope of the Project

The scope of the project include forecasting the number of covid-19 cases and number of fatality for a particular day in a specific region. This Project also aims at comparing different Machine Learning algorithms and techniques to get the best preditions possible.

Chapter 2

Technology Stack

Following are the technologies used in this project :

2.1 Python

Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a ”batteries included” language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980's, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.1. Python 2.0 was released in 2000 and introduced new features, such as list comprehensions and a garbage collection system using reference counting and was discontinued with version 2.7.18 in 2020. Python 3.0 was released in 2008 and was a major revision of the language that is not completely backward-compatible and much Python 2 code does not run unmodified on Python 3.

2.2 Numpy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors.

2.3 Pandas

In computer programming, pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term ”panel data”, an econometrics term for data sets that include observations over multiple time periods for the same individuals. Its name is a play on the phrase ”Python data analysis” itself. Wes McKinney started building what would become pandas at AQR Capital while he was a researcher there from 2007 to 2010.

2.4 Scikit Learn

Scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to inter-operate with the Python numerical and scientific libraries NumPy and SciPy.

2.5 Matplotlib

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 like Tkinter, wxPython, Qt, or GTK+. Matplotlib was originally written by John D. Hunter. Since then it has an active development community and is distributed under a BSD-style license. Matplotlib 2.0.x supports Python versions 2.7 through 3.6. Python 3 support started with Matplotlib 1.2. Matplotlib 1.4 is the last version to support Python 2.6. Matplotlib has pledged not to support Python 2 past 2020 by signing the Python 3 Statement.

2.6 Plotly

Plotly is a technical computing company headquartered in Montreal, Quebec, that develops online data analytics and visualization tools. Plotly provides online graphing, analytics, and statistics tools for individuals and collaboration, as well as scientific graphing libraries for Python, R, MATLAB, Perl, Julia, Arduino, and REST.

Chapter 3

Project Details

3.1 Data Preprocessing

Machine Learning algorithms don’t work so well with processing raw data. Before we can feed such data to an ML algorithm, we must preprocess it. We must apply some transformations on it. With data preprocessing, we convert raw data into a clean data set. There are many techniques of data processing :

3.1.1 Data Preparation

Machine Learning depends largely on test data.Data preparation involves data selection, filtering, transformation, etc.

Most importantly in this we divide data into three parts :

- Training Data
- Testing Data
- Validation Data

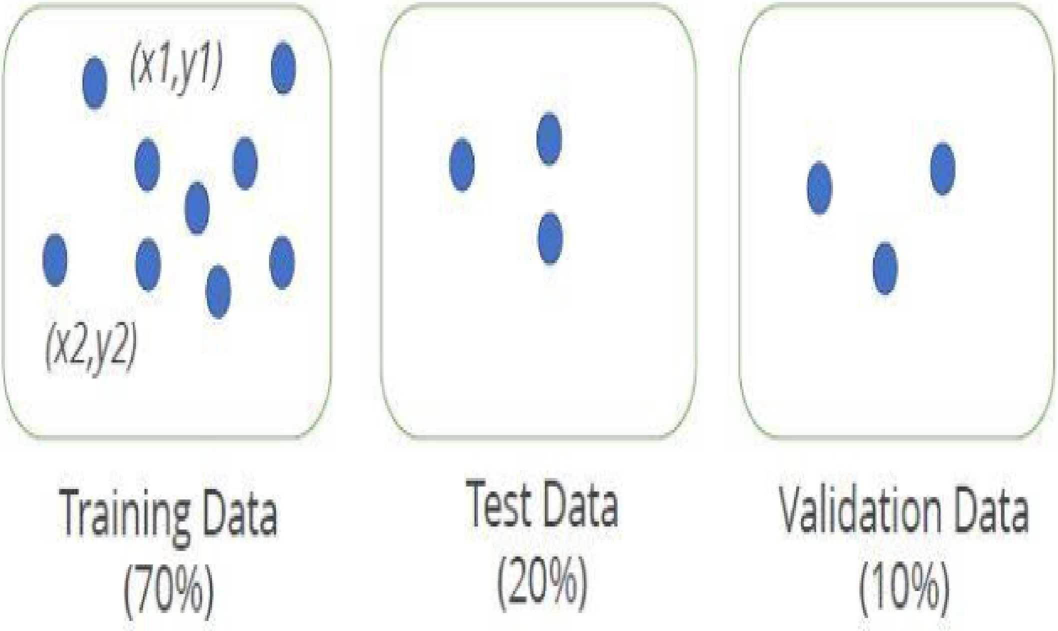


Figure 3.1: Data Selection

3.1.2 Feature Engineering

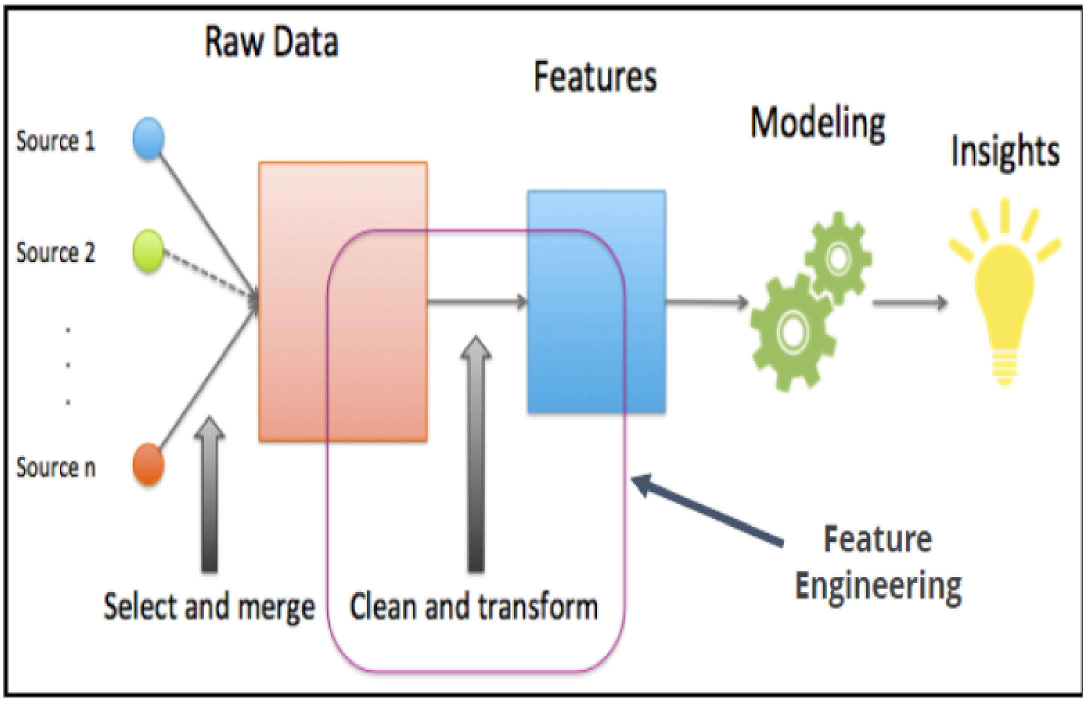


Figure 3.2: Feature Engineering

Feature Engineering refers to selecting and extracting right features from the data that are relevant to the task and model in consideration.

Following are the aspects of feature engineering :

- Feature Selection
- Feature Extraction
- Feature Filter

3.1.3 Feature Scaling

Feature scaling is an important step in the data transformation stage of data preparation process.Feature Scaling is a method used in Machine Learning for standardization of independent variables of data features.

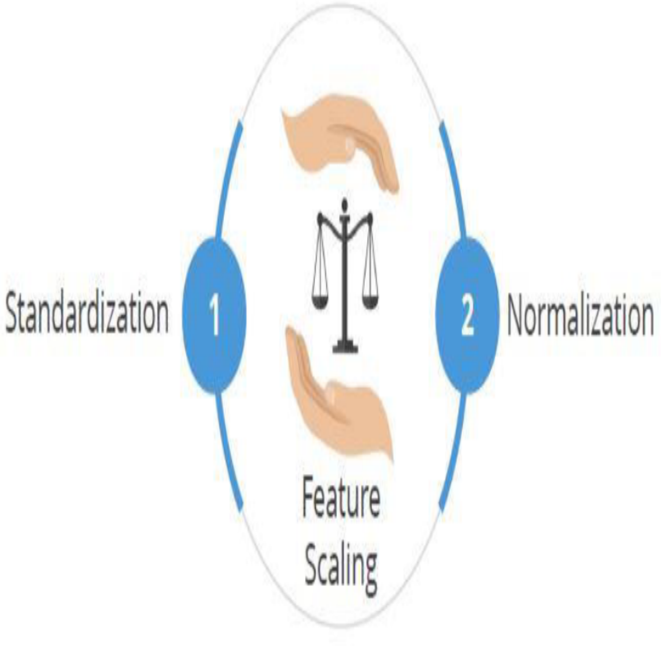


Figure 3.3: Feature Scaling

There are two techniques of feature scaling :

- **Standardization :** With standardizing, we can take attributes with a Gaussian distribution and different means and standard deviations and transform them into a standard Gaussian distribution with a mean of 0 and a standard deviation of 1.
- **Normalization :** In this task, we rescale each observation to a length of 1 (a unit norm). For this, we use the Normalize class.

3.2 Model Training

I used three different Machine learning algorithms to train the model :

3.2.1 Decision Tree

A decision tree falls under supervised Machine Learning Algorithms in Python and comes of use for both classification and regression-although mostly for classification. This model takes an instance, traverses the tree, and compares important features with a determined conditional statement. Whether it descends to the left child branch or the right depends on the result. Usually, more important features are closer to the root.Decision Tree, a Machine Learning algorithm in Python can work on both categorical and continuous dependent variables. Here, we split a population into two or more homogeneous sets. Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees.

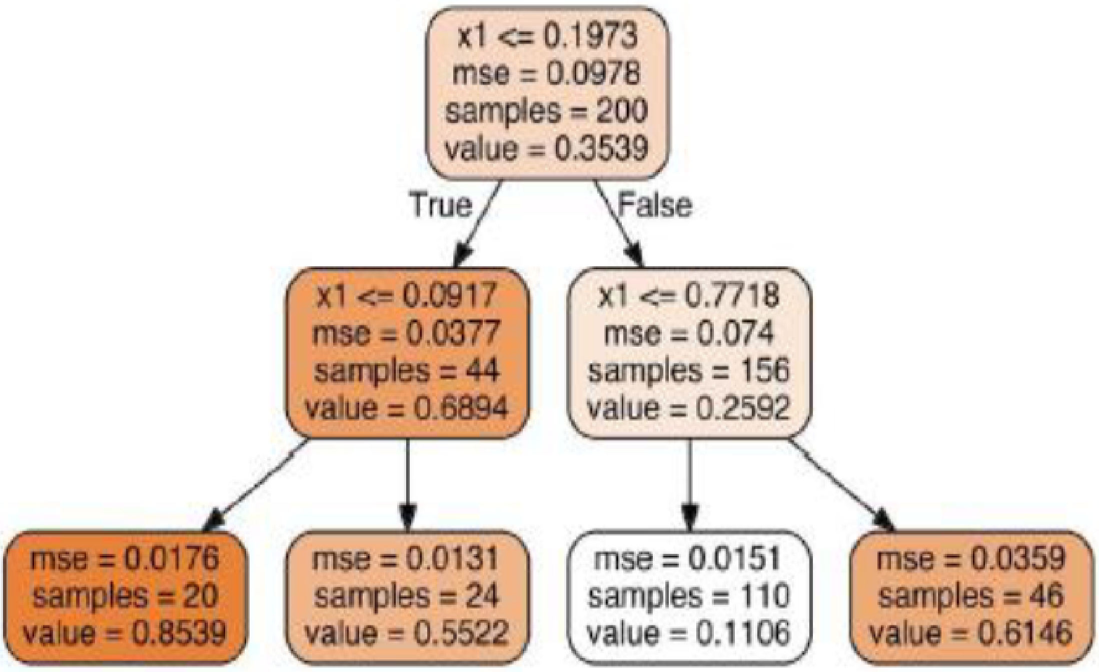


Figure 3.4: Decision Tree

3.2.2 Random Forest

A random forest is an ensemble of decision trees. In order to classify every new object based on its attributes, trees vote for class-each tree provides a classification. The classification with the most votes wins in the forest. Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision tree sat training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

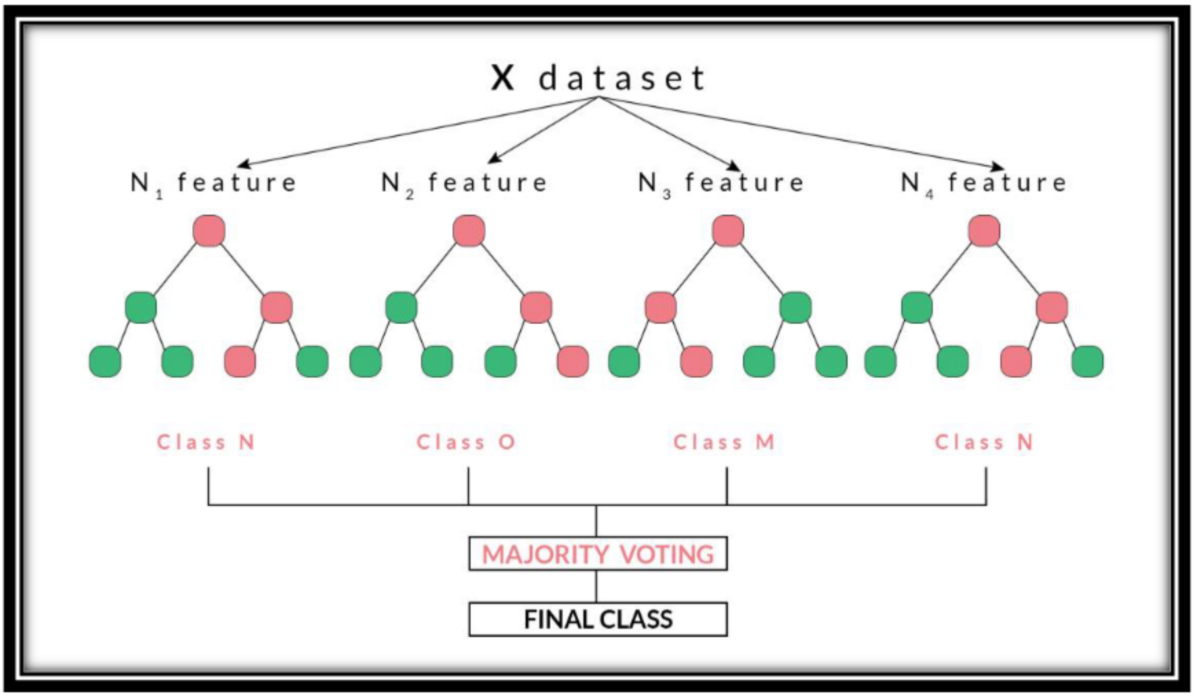


Figure 3.5: Random Forest

3.2.3 XGBoost

The XGBoost library implements the gradient boosting decision tree algorithm. This algorithm goes by lots of different names such as gradient boosting, multiple additive regression trees, stochastic gradient boosting or gradient boosting machines.Boosting is an ensemble technique where new models are added to correct the errors made by existing models. Models are added sequentially until no further improvements can be made. A popular example is the AdaBoost algorithm that weights data points that are hard to predict.Gradient boosting is an approach where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction. It is called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models.This approach supports both regression and classification predictive modeling problems.

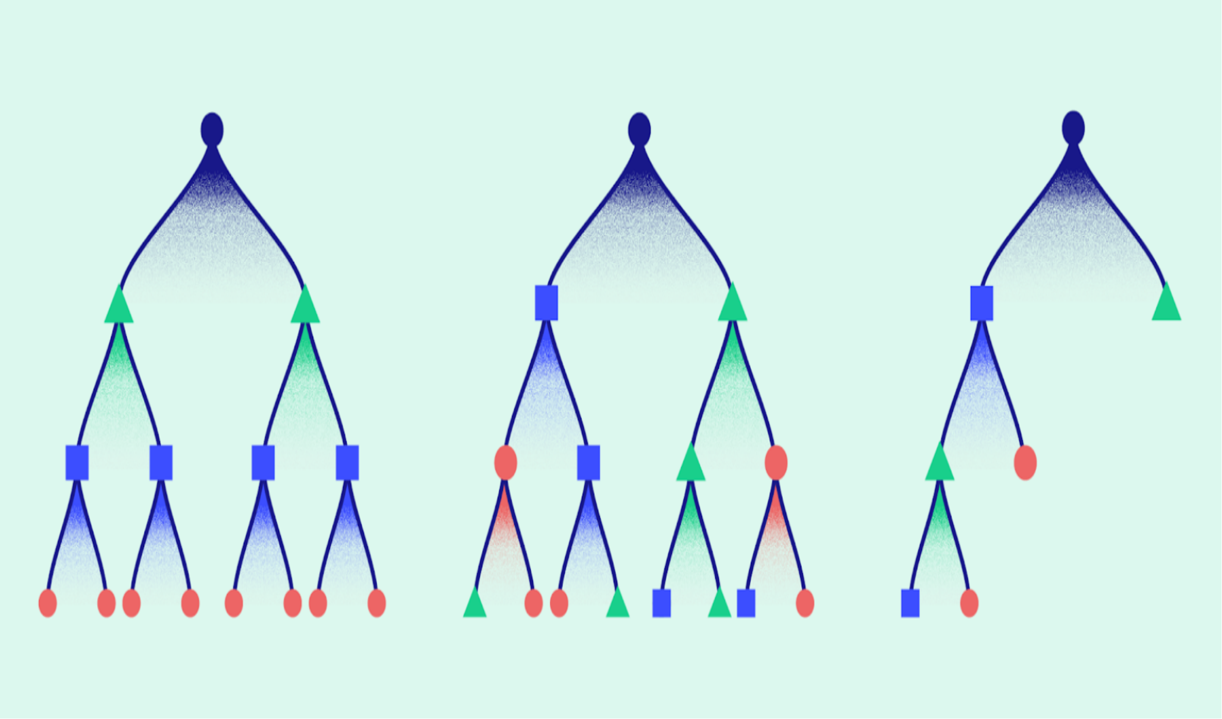
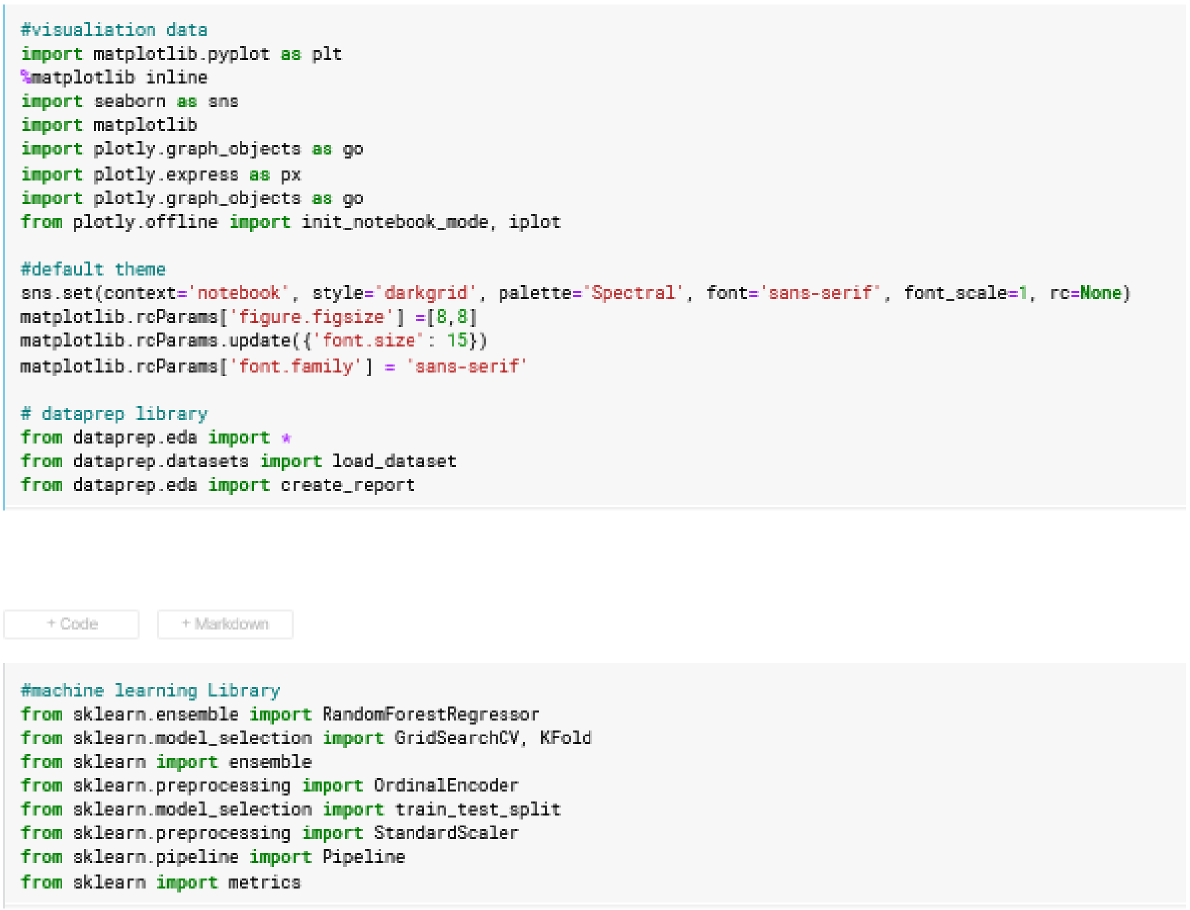


Figure 3.6: XGBoost

Chapter 4

PROJECT SCREENSHOTS



```
#visualiation data
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import matplotlib
import plotly.graph_objects as go
import plotly.express as px
import plotly.graph_objects as go
from plotly.offline import init_notebook_mode, iplot

#default theme
sns.set(context='notebook', style='darkgrid', palette='Spectral', font='sans-serif', font_scale=1, rc=None)
matplotlib.rcParams['figure.figsize'] =[8,8]
matplotlib.rcParams.update({'font.size': 15})
matplotlib.rcParams['font.family'] = 'sans-serif'

# dataprep library
from dataprep.eda import *
from dataprep.datasets import load_dataset
from dataprep.eda import create_report
```

```
#machine learning Library
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import GridSearchCV, KFold
from sklearn import ensemble
from sklearn.preprocessing import OrdinalEncoder
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
from sklearn import metrics
```

Figure 4.1: Importing Libraries

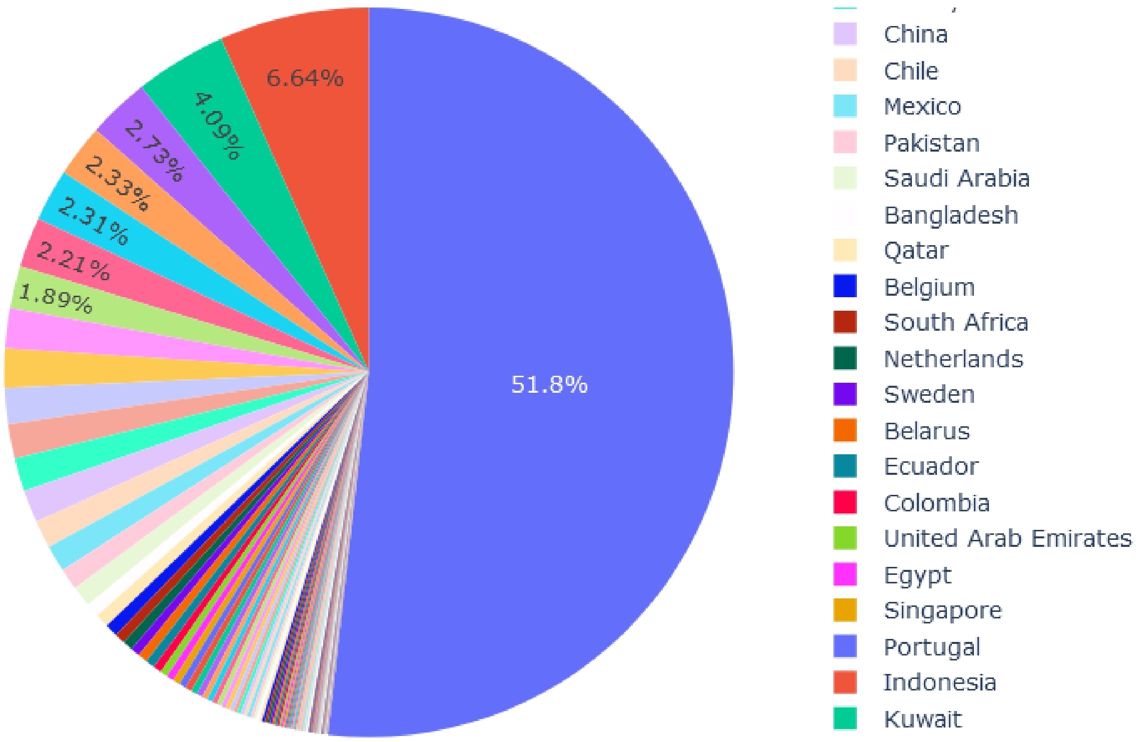


Figure 4.2: Pie Chart

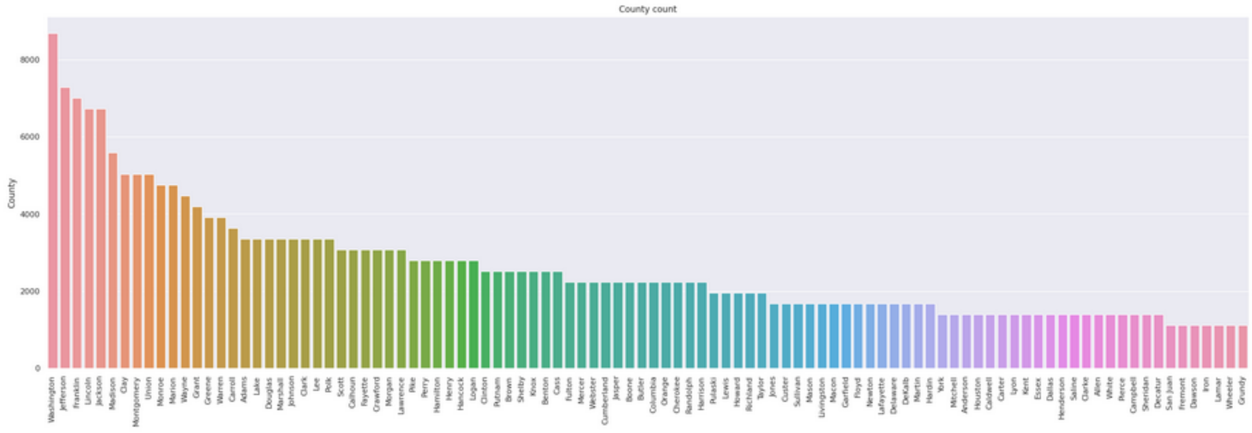


Figure 4.3: Bar Plot

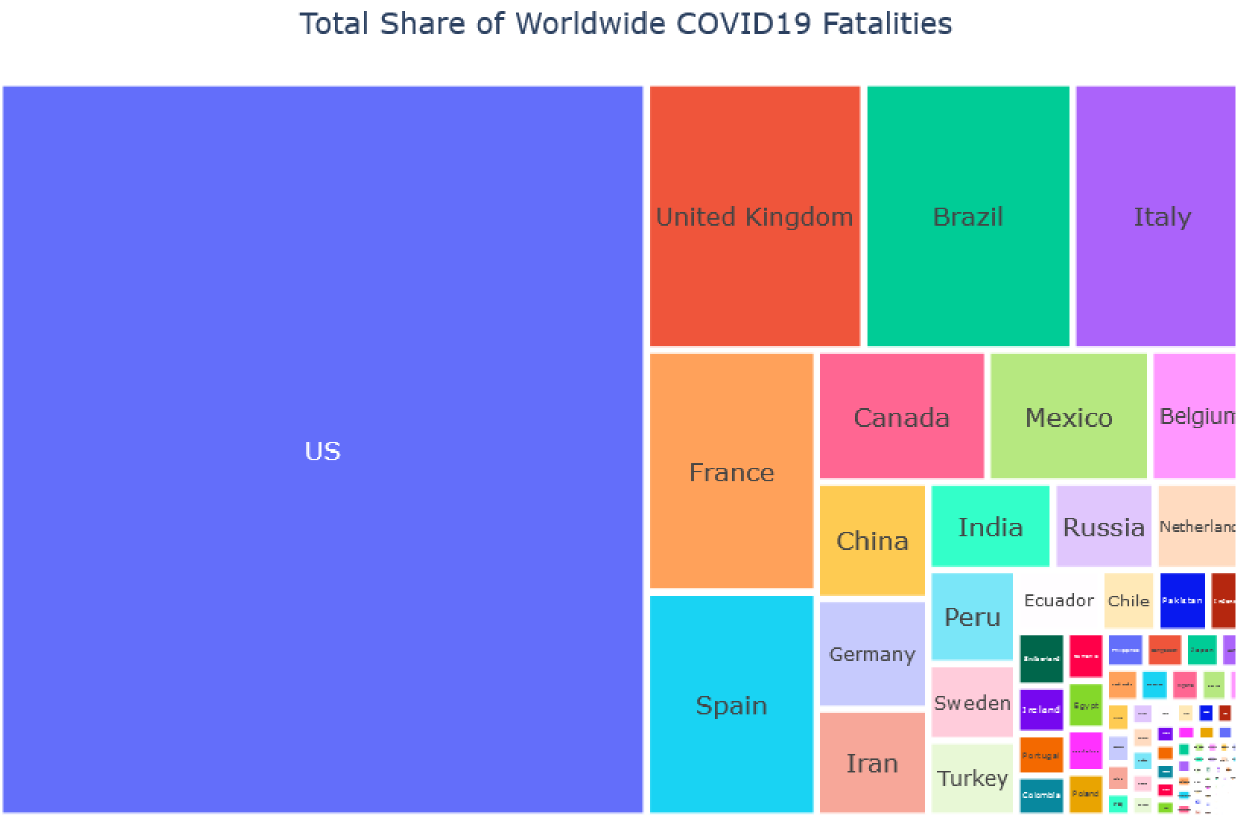


Figure 4.4: Tree Map

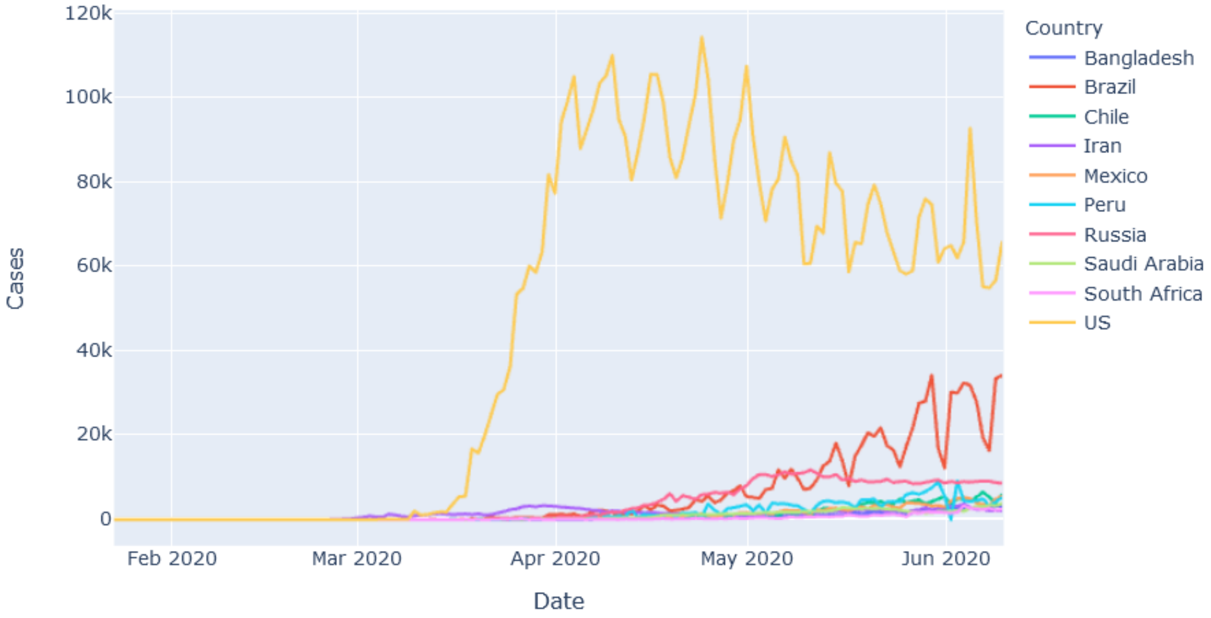
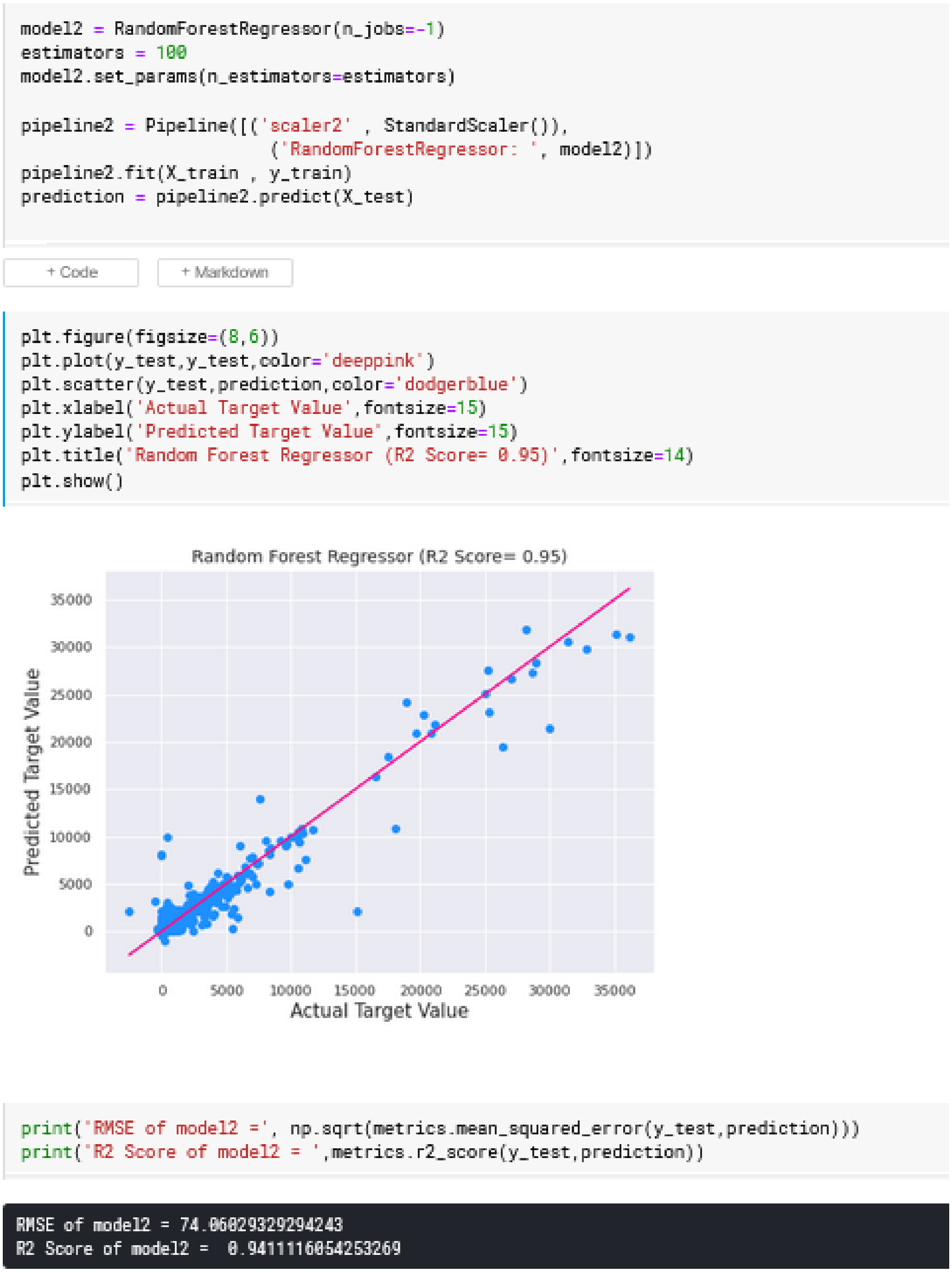


Figure 4.5: Line Plot



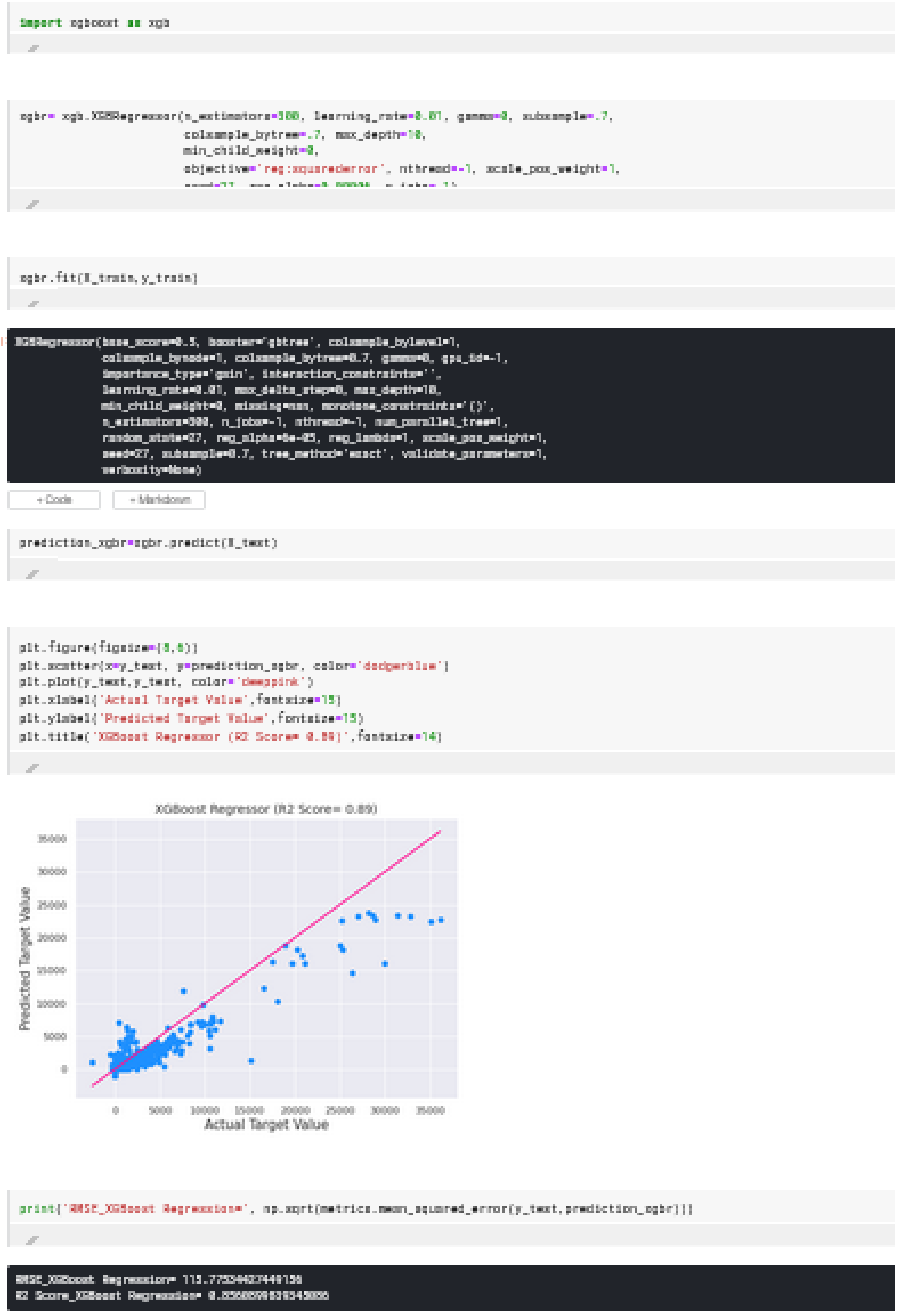
```
model2 = RandomForestRegressor(n_jobs=-1)
estimators = 100
model2.set_params(n_estimators=estimators)

pipeline2 = Pipeline([('scaler2' , StandardScaler()),
                        ('RandomForestRegressor: ', model2)])
pipeline2.fit(X_train , y_train)
prediction = pipeline2.predict(X_test)
```

```
plt.figure(figsize=(8,6))
plt.plot(y_test,y_test,color='deeppink')
plt.scatter(y_test,prediction,color='dodgerblue')
plt.xlabel('Actual Target Value',fontsize=15)
plt.ylabel('Predicted Target Value',fontsize=15)
plt.title('Random Forest Regressor (R2 Score= 0.95)',fontsize=14)
plt.show()
```

```
print('RMSE of model2 =', np.sqrt(metrics.mean_squared_error(y_test,prediction)))
print('R2 Score of model2 = ',metrics.r2_score(y_test,prediction))
```

```
RMSE of model2 = 74.06029329294243
R2 Score of model2 =  0.9411116054253269
```

Figure 4.6: Random Forest



```
import xgboost as xgb
```

```
xgbr= xgb.XGBRegressor(n_estimators=300, learning_rate=0.01, gamma=0, subsample=.7,
                       colsample_bytree=.7, max_depth=10,
                       min_child_weight=0,
                       objective='reg:squarederror', nthread=-1, scale_pos_weight=1,
                       seed=27, reg_alpha=0.00006, n_jobs=-1)
```

```
xgbr.fit(X_train,y_train)
```

```
XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,
             colsample_bynode=1, colsample_bytree=0.7, gamma=0, gpu_id=-1,
             importance_type='gain', interaction_constraints='',
             learning_rate=0.01, max_delta_step=0, max_depth=10,
             min_child_weight=0, missing=nan, monotone_constraints='()',
             n_estimators=300, n_jobs=-1, nthread=-1, num_parallel_tree=1,
             random_state=27, reg_alpha=6e-05, reg_lambda=1, scale_pos_weight=1,
             seed=27, subsample=0.7, tree_method='exact', validate_parameters=1,
             verbosity=None)
```

```
prediction_xgbr=xgbr.predict(X_test)
```

```
plt.figure(figsize=(8,6))
plt.scatter(x=y_test, y=prediction_xgbr, color='dodgerblue')
plt.plot(y_test,y_test, color='deeppink')
plt.xlabel('Actual Target Value',fontsize=15)
plt.ylabel('Predicted Target Value',fontsize=15)
plt.title('XGBoost Regressor (R2 Score= 0.89)',fontsize=14)
```

```
print('RMSE_XGBoost Regression=', np.sqrt(metrics.mean_squared_error(y_test,prediction_xgbr)))
```

```
RMSE_XGBoost Regression= 115.775344027440158
R2 Score_XGBoost Regression= 0.85580898383345086
```

Figure 4.7: XGBoost

Chapter 5

CONCLUSIONS

The global pandemic of the severe acute respiratory syndrome Covid-19 (SARS-CoV-2) has become the primary national security issue of many nations. Advancement of accurate prediction models for the outbreak is essential to provide insights into the spread and consequences of this infectious disease. Due to the high level of uncertainty and lack of crucial data, standard epidemiological models have shown low accuracy for long-term prediction. In this study, an ML-based prediction system has been proposed for predicting the risk of COVID-19 out-break globally. The system analyses dataset containing the day-wise actual past data and makes predictions for upcoming days using machine learning algorithms. This paper presents a comparative analysis of ML and soft computing models to predict the COVID-19 outbreak. The results of two ML models (Random Forest and XGBoost)reported a high generalization ability for long-term prediction. With respect to the results reported in this paper and due to the highly complex nature of the COVID-19 outbreak and differences from nation-to-nation, this study suggests ML as an effective tool to model the time series of outbreak.We should note that this paper provides an initial benchmark to demonstrate the potential of machine learning for future research.

For the advancement of higher performance models for long-term prediction, future research should be devoted to comparative studies on various ML models for individual countries.Due to the fundamental differences between the outbreak in various countries,advancement of global models with generalization ability would not be feasible.As observed and reported in many studies,it is unlikely that an individual outbreak will be replicated elsewhere.

Although the most difficult prediction is to estimate the maximum number of infected patients, estimation of the individual mortality rate ((no of deaths) / (no of infecteds)) is also essential. The mortality rate is particularly important to estimate accurately the number of patients and the required beds in intensive care units. For future research,modeling the mortality rate would be of the utmost importance for nations to plan for new facilities.For future research integration of machine learning and SIR/SEIR models is suggested to enhance the existing standard epidemiological models in terms of accuracy and longer lead time.

It was very difficult to choose different hyper-parameters between the given values of the dataset. Overall we conclude that model predictions according to the current scenario are correct which may be helpful to understand the upcoming situation. The study forecasts thus can also be of great help for the authorities to take timely actions and make decisions to contain the COVID-19 crisis. This study will been hanced continuously in the future course, next we plan to explore the prediction methodology using the updated dataset and use the most accurate and appropriate ML methods for forecasting. Real-time live forecasting will be one of the primary focuses in our future work.

Chapter 6

FUTURE SCOPE

- Can use more advanced machine learning techniques and more advance algorithms to make better prediction.
- Can be upgraded for providing suitable solution according to different regions based on it's attribute.
- can be upgraded to automatically provide the necessary resource required to better deal with the Covid-19 pandemic.
- Can be used in making of better decision and plan to handle Covid-19 pandemic.
- Can be upgraded for future use of handling different pandemic.
- can utilize available resources much better.
- Gives different result based on different regions.
- Output based on real time input data.

Chapter 7

REFERENCES

- Ivanov, D. Predicting the impacts of epidemic outbreaks on global supply chains : Asimulation-based analysis on the coronavirus outbreak (COVID-19/SARS-CoV-2) case. Transp. Res. Part E Logist. Transp. Rev. 2020, 136, doi:10.1016/j.tre.2020.101

- Rypdal, M.; Sugihara, G. Inter-outbreak stability reflects the size of the susceptible pool and forecasts magnitudes of seasonal epidemics. Nat. Commun. 2019, 10, doi:10.1038/s41467-019-10099-y.

- R. Kaundal, A. S. Kapoor, and G. P. Raghava, “Machine learning techniques in disease forecasting: A case study on rice blast prediction,”BMCBioinf., vol. 7, no. 1, p. 485, 2006.

- `https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series`

- `https://github.com/CSSEGISandData/COVID-19`