# INTERNSHIP PROJECT

# COVID-19 DEATH RATE ANALYSIS

A Report Submitted to

Jawaharlal Nehru Technological University Kakinada,

Kakinadain partial fulfillment for the award of the degree of

## BACHELOR OF TECHNOLOGY
## IN
## COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

Submitted by:


**K.NAGA DEVI MOUNIKA**
(20KN1A4419)

**D.SAI SAMRAT**

(20KN1A4409)

**M.GIRISH KUMAR**

(21KN5A4405)


**UNDER THE ESTEEMED GUIDANCE OF**

**MR.P. SESHA SAI (DATA SCIENTIST)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE))**

**NRI INSTITUTE OF TECHNOLOGY**

**(Autonomous)**

**(Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada) Accredited by NBA (CSE, ECE & EEE), Accredited by NAAC with 'A' GradeISO 9001: 2015 Certified Institution**

Pothavarappadu (V), (Via) Nunna, Agiripalli (M), Krishna Dist., PIN: 521212, A.P, India.2022-2023

# NRI INSTITUTE OF TECHNOLOGY

(An Autonomous Institution, Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada)

Accredited by NBA (CSE, ECE & EEE), Accredited by NAAC with 'A' Grade
ISO 9001: 2015 Certified Institution

Pothavarappadu (V), (Via) Nunna, Agiripalli (M), Krishna Dist., PIN: 521212, A.P, India.

## CERTIFICATE

This is to certify that the "**Internship report"** submitted by **KANDIPALLI.NAGA DEVI MOUNIKA**(20KN1A4419),**DASARI.SAISAMRAT**(20KN1A4448**),MURALA.GIRISH KUMAR**(21KN5A4405) is work done by and submitted during 2022-2023 academic year, in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING,** at **BLACKBUCK ENGINEERS PVT LTD, Road No:36, Jubilee Hills, Hyderabad, Telangana.**

Kathyayani

**INTERNSHIP COORDINATOR**

(R.KATHYAYANI)

**Head of the Department**

(Dr. D. SUNEETHA)

**EXTERNAL EXAMINER**

BLACKBUCK ENGINEERS

# Internship Experience Letter

**Certificate ID: BBNR0291** **Issued Date: 01st November 2022**

**To Whom It May Concern:**

This is to certify that **KANDIPALLI NAGA DEVIMOUNIKA** has successfully completed internship at **Blackbuck Engineers Pvt Ltd**, and She worked with us from **13th June 2022** to **05th September 2022.**

She has worked on a project titled **Covid-19 Death Rate Analysis** by learning and incorporating Artificial Intelligence & Machine Learning concepts under the supervision of our project mentor.

We found that She is sincere, hardworking, technically sound and result oriented. She worked well as part of a team during the tenure.

We wish all the best for future endeavours.

Best regards,

**Kathyayani Rudravelli**
Project Head
Blackbuck Engineers Pvt Ltd

**Mounika Bezawada**
HR Manager
Blackbuck Engineers Pvt Ltd

+91-9391002762 Jubilee Hills, Hyderabad, Telangana contact@blackbucks.me

BLACKBUCK ENGINEERS

# Internship Experience Letter

**Certificate ID: BBNR0282** **Issued Date: 01st November 2022**

**To Whom It May Concern:**

This is to certify that **DASARI SAI SAMRAT** has successfully completed internship at **Blackbuck Engineers Pvt Ltd**, and He worked with us from **13th June 2022** to **05th September 2022.**

He has worked on a project titled **Covid-19 Death Rate Analysis** by learning and incorporating Artificial Intelligence & Machine Learning concepts under the supervision of our project mentor.

We found that He is sincere, hardworking, technically sound and result oriented. He worked well as part of a team during the tenure.

We wish all the best for future endeavours.

Best regards,

Kathyayani

**Kathyayani Rudravelli**
Project Head
Blackbuck Engineers Pvt Ltd

**Mounika Bezawada**
HR Manager
Blackbuck Engineers Pvt Ltd

+91-9391002762 Jubilee Hills, Hyderabad, Telangana contact@blackbucks.me

BLACKBUCK ENGINEERS

# Internship Experience Letter

**Certificate ID: BBNR0278** **Issued Date: 01st November 2022**

**To Whom It May Concern:**

This is to certify that **MURALA GIRISH KUMAR** has successfully completed internship at **Blackbuck Engineers Pvt Ltd**, and He worked with us from **13th June 2022** to **05th September 2022.**

He has worked on a project titled **Covid-19 Death Rate Analysis** by learning and incorporating Artificial Intelligence & Machine Learning concepts under the supervision of our project mentor.

We found that He is sincere, hardworking, technically sound and result oriented. He worked well as part of a team during the tenure.

We wish all the best for future endeavours.

Best regards,

**Kathyayani Rudravelli**
Project Head
Blackbuck Engineers Pvt Ltd

**Mounika Bezawada**
HR Manager
Blackbuck Engineers Pvt Ltd

+91-9391002762 Jubilee Hills, Hyderabad, Telangana contact@blackbucks.me

# ACKNOWLEDGEMENT

we take this opportunity to thank all who have rendered their full support to my work. The pleasure, the achievement, the glory, the satisfaction, the reward, the appreciation and the construction of my project cannot be expressed with a few words for their valuable suggestions.

we are expressing our heartfelt thanks to **Head of the Department, Dr. D. SUNEETHA** garu for her continuous guidance for completion of our Project work.

we are extending our sincere thanks to **Dean of the Department, Dr. K. V.SAMBASIVA RAO** for his continuous guidance and support to complete our project successfully.

we are thankful to the **Principal, Dr. C. NAGA BHASKAR** garu for his encouragement to complete the Project work.

We are extending our sincere and honest thanks to the **Chairman, Dr. R. VENKATA RAOgaru & Secretary, Sri K. Sridhar** garu for their continuous support in completing the Project work.

We are thankful to **Blackbucks CEO, Ms. ANURADHA THOTA, Academic Administrator Ms. KATHYAYANI R** and Project Mentors for their continuous guidance and support to complete our project successfully.

Finally, we thank the Administrative Officer, Staff Members, Faculty of Department of CSE, NRI Institute of Technology and my friends, directly or indirectly helped us in the completion of this project

**K .NAGA DEVI MOUNIKA**

**(20KN1A4419)**

**D.SAI SAMRAT**

**(20KN1A4409)**

**M.GIRISH KUMAR**

**(21KN5A4405)**

# ABSTRACT

Death rates of coronavirus disease-2019 (COVID-19) continue to rise across the world. The impact of several risk factors on coronavirus mortality has been previously reported in several meta-analyses limited by small sample sizes. In this systematic review, we aimed to find the number of deaths,confirmed and recovered cases based on the data that we have collected.some insights could be-change in number of affected cases over time,change in cases overtime at country level,latest number of affected cases.We followed 4 steps: 1.Data Collection 2.Data Pre-processing 3.Exploratory Data Analysis 4.Fitting the Model

## Organization Information:

Blackbuck Engineers is started in 2013 with the aim of creating a great ecosystem of academia, research, industry, and individuals. Blackbucks is a premier partner to Govts International Institute of Digital Technologies, and IITs. Blackbuck delivers the TAPTAP AI Driven employability platform to transform the journey of students towards their dream goals while helping **HRs hire right students**.

Blackbuck has the largest chain of excellence in emerging tech across India.

Blackbucks runs post graduation programs in AI, ML and Data Science

www.theblackbucks.com

## Programs and opportunities:

This ground-up approach helps us deliver not only the solution to our clients but also add value to at the core Blackbuck Engineers which operates in Five specific domains namely TapTap - AI Driven, Post Graduation Programs, Center of Excellence, Virtual Programming Labs and Happie Days - A social Networking site for the students. TapTap offer services in Campus Recruitment drives for the Employers as well as College authorities. Recruiters can Conduct Customized Online Assessments secured with Best-in-class Proctoring and Schedule the end-to-end hiring process. Under each division we further provide specific industry solutions on focused domains with cutting edge technologies. Blackbuck Engineers emphasize on building relationships with our clients by delivering projects on time and within budget.

**INDEX**

## Learning Objectives/Internship Objectives

- Internships are generally thought of to be reserved for college students looking to gain experience in a particular field. However, a wide array of people can benefit from Training Internships in order to receive real world experience and develop their skills.

- An objective for this position should emphasize the skills you already possess in the area and your interest in learning more

- Internships are utilized in a number of different career fields, including architecture, engineering, healthcare, economics, advertising and many more.

- Some internship is used to allow individuals to perform scientific research while others are specifically designed to allow people to gain first-hand experience working.

Utilizing internships is a great way to build your resume and develop skills that can be emphasized in your resume for future jobs. When you are applying for a TrainingInternship, make sure to highlight any special skills or talents that can make you stand apart from the rest of the applicants so that you have an improved chance of landing the position.

## WEEKLY OVERVIEW OF INTERNSHIP ACTIVITIES

| | DATE | DAY | NAME OF THE TOPIC/MODULE COMPLETED |
|---|---|---|---|
| 1st WEEK | 13.06.2022 | Monday | Introduce the Topic & the Problem Statement |
| | 14.06.2022 | Tuesday | Introduce the Topic & the Problem Statement |
| | 15.06.2022 | Wednesday | Introduce the Topic & the Problem Statement |
| | 16.06.2022 | Thursday | Introduce the Topic & the Problem Statement |
| | 17.06.2022 | Friday | Introduce the Topic & the Problem Statement |

| | DATE | DAY | NAME OF THE TOPIC/MODULE COMPLETED |
|---|---|---|---|
| 2nd WEEK | 20.06.2022 | Monday | Abstract Building |
| | 21.06.2022 | Tuesday | Abstract Building |
| | 22.06.2022 | Wednesday | Abstract Building |
| | 23.06.2022 | Thursday | Abstract Building |
| | 24.06.2022 | Friday | Abstract Submission |

| | DATE | DAY | NAME OF THE TOPIC/MODULE COMPLETED |
|---|---|---|---|
| 3rd WEEK | 27.06.2022 | Monday | Abstract Submission |
| | 28.06.2022 | Tuesday | Abstract Submission |
| | 29.06.2022 | Wednesday | Explain your Approach to Solving Problem |
| | 30.06.2022 | Thursday | Explain your Approach to Solving Problem |
| | 01.07.2022 | Friday | Explain your Approach to Solving Problem |

| | DATE | DAY | NAME OF THE TOPIC/MODULE COMPLETED |
|---|---|---|---|
| 4th WEEK | 04.07.2022 | Monday | Explain your Approach to Solving Problem |
| | 05.07.2022 | Tuesday | Explain Structure of Project |
| | 06.07.2022 | Wednesday | Explain Structure of Project |
| | 07.07.2022 | Thursday | Explain Structure of Project |
| | 08.07.2022 | Friday | Explain Structure of Project |

| 5th WEEK | DATE | DAY | NAME OF THE TOPIC/MODULE COMPLETED |
|---|---|---|---|
| | 11.07.2022 | Monday | Data Preprocessing |
| | 12.07.2022 | Tuesday | Data Preprocessing |
| | 13.07.2022 | Wednesday | Data Preprocessing |
| | 14.07.2022 | Thursday | Data Preprocessing |
| | 15.07.2022 | Friday | Data Preprocessing |

| 6th WEEK | DATE | DAY | NAME OF THE TOPIC/MODULE COMPLETED |
|---|---|---|---|
| | 18.07.2022 | Monday | Perform Analysis |
| | 19.07.2022 | Tuesday | Perform Analysis |
| | 20.07.2022 | Wednesday | Perform Analysis |
| | 21.07.2022 | Thursday | Perform Analysis |
| | 22.07.2022 | Friday | Perform Analysis |

| 7th WEEK | DATE | DAY | NAME OF THE TOPIC/MODULE COMPLETED |
|---|---|---|---|
| | 25.07.2022 | Monday | PPT Preparation |
| | 26.07.2022 | Tuesday | PPT Preparation |
| | 27.07.2022 | Wednesday | PPT Preparation |
| | 28.07.2022 | Thursday | PPT Preparation |
| | 29.07.2022 | Friday | PPT Preparation |

| 8th WEEK | DATE | DAY | NAME OF THE TOPIC/MODULE COMPLETED |
|---|---|---|---|
| | 01.08.2022 | Monday | PPT Submission |
| | 02.08.2022 | Tuesday | PPT Submission |
| | 03.08.2022 | Wednesday | Mid Review |
| | 04.08.2022 | Thursday | Mid Review |
| | 05.08.2022 | Friday | Mid Review |

| 9th WEEK | DATE | DAY | NAME OF THE TOPIC/MODULE COMPLETED |
|---|---|---|---|
| | 08.08.2022 | Monday | Mid Review |
| | 10.08.2022 | Tuesday | Mid Review |
| | 11.08.2022 | Wednesday | Building & Applying Algorithm |
| | 12.08.2022 | Thursday | Building & Applying Algorithm |

| 10th WEEK | DATE | DAY | NAME OF THE TOPIC/MODULE COMPLETED |
|---|---|---|---|
| | 16.08.2022 | Tuesday | Building & Applying Algorithm |
| | 17.08.2022 | Wednesday | Building & Applying Algorithm |
| | 19.08.2022 | Friday | Building & Applying Algorithm |

| 11th WEEK | DATE | DAY | NAME OF THE TOPIC/MODULE COMPLETED |
|---|---|---|---|
| | 22.08.2022 | Monday | Concluding Project |
| | 23.08.2022 | Tuesday | Concluding Project |
| | 24.08.2022 | Wednesday | Concluding Project |
| | 25.08.2022 | Thursday | Concluding Project |
| | 26.08.2022 | Friday | Concluding Project |

| 12th WEEK | DATE | DAY | NAME OF THE TOPIC/MODULE COMPLETED |
|---|---|---|---|
| | 29.08.2022 | Monday | Final Review |
| | 30.08.2022 | Tuesday | Final Review |
| | 01.09.2022 | Wednesday | Final Review |
| | 02.09.2022 | Thursday | Final Review |
| | 05.09.2022 | Friday | Final Review |

# CHAPTER 1
# INTRODUCTION

# INTRODUCTION

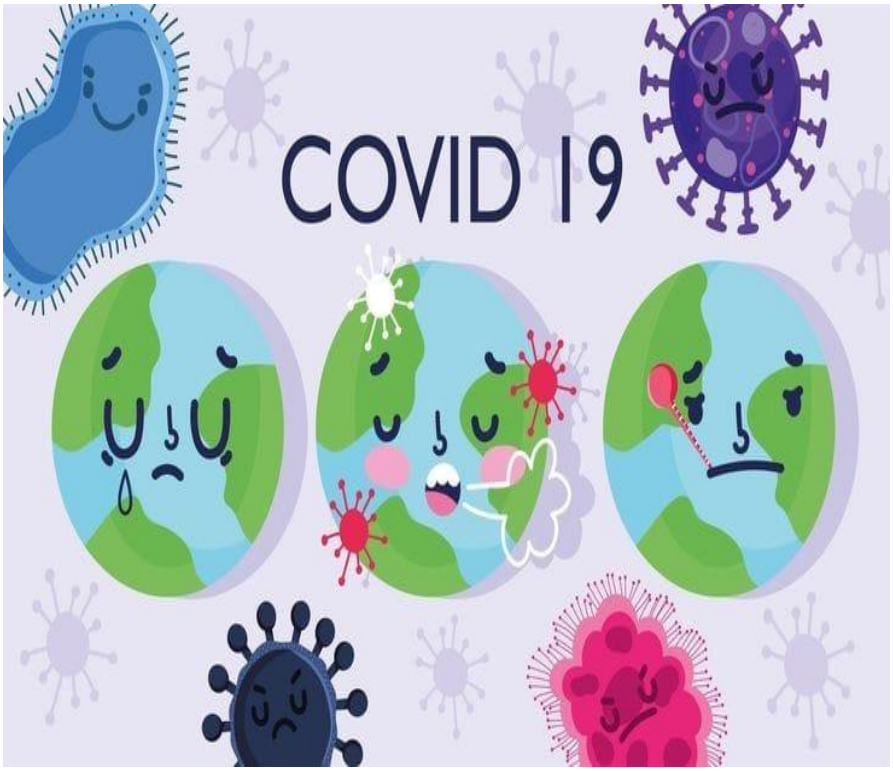
Coronavirus disease 2019 (COVID-19) is an infectious disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). It was first identified in December 2019 in Wuhan, China, and has resulted in an ongoing pandemic. The first case may be traced back to 17 November 2019. As of 8 June 2020, more than 6.98 million cases have been reported across 188 countries and territories, resulting in more than 401,000 deaths. More than 3.13 million people have recovered.

The virus is primarily spread between people during close contact, most often via small droplets produced by coughing, sneezing, and talking. The droplets usually fall to the ground or onto surfaces rather than travelling through air over long distances. Less commonly, people may become infected by touching a contaminated surface and then touching their face. It is most contagious during the first three days after the onset of symptoms, although spread is possible before symptoms appear, and from people who do not show symptoms

From World Health Organization

- On 31 December 2019, WHO was alerted to several cases of pneumonia in Wuhan City, Hubei Province of China. The virus did not match any other known virus. This raised concern because when a virus is new, we do not know how it affects people.

So daily level information on the affected people can give some interesting insights when it is made available to the broader data science community.


COVID 19

# CHAPTER 2
# ATTRIBUTE INFORMATION

## 2.ATTRIBUTE INFORMATON:

- S no: - Serial number
- Observation Date - Date of the observation in MM/DD/YYYY
- Province/State - Province or state of the observation (Could be empty when missing)
- Country/Region - Country of observation
- Last Update - Time in UTC at which the row is updated for the given province or country. (Not standardised and so please clean before using it)
- Confirmed - Cumulative number of confirmed cases till that date
- Deaths - Cumulative number of of deaths till that date
- Recover ed - Cumulative number of recovered cases till that date

# CHAPTER 3
# SOFTWARE REQUIREMENTS SPECIFICATION

# 3.SOFTWARE REQUIREMENTS SPECIFICATIONS

## System configurations

The software requirement specification can produce at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by established a complete information description, a detailed functional description, a representation of system behavior, and indication of performance and design constrain, appropriate validate criteria, and other information pertinent to requirements.

## Software Requirements:

- Operating system : Windows 7 Ultimate.
- Coding Language : MVC 4 Razor
- Front-End : Visual Studio 2012 Professional.
- Data Base : SQL Server 2008.

## Hardware Requirement:

- System : Pentium IV 2.4 GHz.
- Hard Disk : 1TB.
- Ram : 4GB

# CHAPTER 4
# LITERATURE REVIEW

# LITERATURE REVIEW

**Python** is a high-level, interpreted, general-purpose programming language. Itsdesign philosophy emphasizes code readability with the use of significant indentation.

Python is dynamically-typed and garbage-collected. It supports multiple

**Programming** including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

**JUPYTER NOTBOOK**: The Jupyter Notebook is an open-source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter.

If so, then you can use a handy tool that comes with Python called pip to install Jupyter Notebook like this:

```
$ pip install jupyter
```

# CHAPTER 5
# KEYWORDS AND DEFINITIONS

## 5.KEYWORDS and DEFINITION

| COMMAND | DEFINATION |
|---|---|
| PIP | pip is the standard package manager for Python. It allows you to install and manage additional packages that are not part of the Python standard library. |
| from | The from keyword is used to import only a specified section from a module. |
| Import | import imports the whole library. from import imports a specific member or members of thelibrary. |
| Import* | It just means that you import all (methods, variables, ...) in a way so you don't need to prefixthem when using them |
| 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. |
| Updater | The update() method inserts the specified items tothe dictionary. The specified items can be a dictionary, or an iterable object with key value pairs. |
| Pandas | 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. |
| 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, |

| | |
|---|---|
| Machine Learning | Machine Learning is a subfield of Artificial Intelligence, which is broadly defined as the capability of a machine to imitate intelligent human behaviour. |
| Regression | Regression is a technique for investigating relationship between dependent variable or outcome and independent features or variables. |
| Dataset | A collection of related sets of information that is composed of separate elements but can be manipulated as a unit by a computer. |
| Algorithm | A process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer. |

# CHAPTER6
# METHADALOGIES

# 6.METHODOLOGIES

We follow a structured methodology for our projects which starts from designing the solution to the implementation phase. Well planned Project reduces the time to deliver the project and any additional ad-hoc costs to our clients, hence we dedicate majority of our time understanding our client's business and gather requirements. This ground up approach helps us deliver not only the solution to our clients but also add value to your investments.

**Data Preprocessing:**

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

It involves the following steps:

1. Data collection
2. Importing Libraries
3. Importing Dataset
4. Finding missing data
5. Encoding Categorical values
6. Exploratory Data Analysis
7. Splitting Dataset into train-test data
8. Fitting the model
9. Model Evaluation

**Data Collection:**

Data collection is an essential part of exploratory data analysis. It refers to the process of finding and loading data into our system. Good, reliable data can be found on various public sites or bought from private organizations. Some reliable sites for data collection are Kaggle, GitHub, Machine Learning Repository, etc.

The data depicted below represents the supermarket sales dataset that is available on Kaggle. It contains information on supermarket and its sales.

**Importing Libraries:**

To import the required libraries, we use the following code:

```
import pandas as pd , numpy as np
from sklearn.utils import resample
from sklearn.preprocessing import StandardScaler , MinMaxScaler
from collections import Counter
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns

import plotly.express as px
import plotly.figure_factory as ff
import plotly
```

# CHAPTER – 7
# CODING

## 7.CODING:

```
#Data processing functions

from sklearn.model_selection import train_test_split
from sklearn import model_selection
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()

import warnings
warnings.filterwarnings("ignore")
```

**Importing Dataset:**

To import the dataset, use the following code:

```
df=pd.read_csv("covid_19_data.csv")
```

To get a brief look into our Dataset we used the following code:

Out[4]:

| | SNo | ObservationDate | Province/State | Country/Region | Last Update | Confirmed | Deaths | Recovered |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 01/22/2020 | Anhui | Mainland China | 1/22/2020 17:00 | 1.0 | 0.0 | 0.0 |
| 1 | 2 | 01/22/2020 | Beijing | Mainland China | 1/22/2020 17:00 | 14.0 | 0.0 | 0.0 |
| 2 | 3 | 01/22/2020 | Chongqing | Mainland China | 1/22/2020 17:00 | 6.0 | 0.0 | 0.0 |
| 3 | 4 | 01/22/2020 | Fujian | Mainland China | 1/22/2020 17:00 | 1.0 | 0.0 | 0.0 |
| 4 | 5 | 01/22/2020 | Gansu | Mainland China | 1/22/2020 17:00 | 0.0 | 0.0 | 0.0 |
| 5 | 6 | 01/22/2020 | Guangdong | Mainland China | 1/22/2020 17:00 | 26.0 | 0.0 | 0.0 |
| 6 | 7 | 01/22/2020 | Guangxi | Mainland China | 1/22/2020 17:00 | 2.0 | 0.0 | 0.0 |
| 7 | 8 | 01/22/2020 | Guizhou | Mainland China | 1/22/2020 17:00 | 1.0 | 0.0 | 0.0 |
| 8 | 9 | 01/22/2020 | Hainan | Mainland China | 1/22/2020 17:00 | 4.0 | 0.0 | 0.0 |
| 9 | 10 | 01/22/2020 | Hebei | Mainland China | 1/22/2020 17:00 | 1.0 | 0.0 | 0.0 |
| 10 | 11 | 01/22/2020 | Heilongjiang | Mainland China | 1/22/2020 17:00 | 0.0 | 0.0 | 0.0 |

To get more information we can use this code :

```
df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20574 entries, 0 to 20573
Data columns (total 8 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   SNo             20574 non-null  int64
 1   ObservationDate 20574 non-null  object
 2   Province/State  10026 non-null  object
 3   Country/Region  20574 non-null  object
 4   Last Update     20574 non-null  object
 5   Confirmed       20574 non-null  float64
 6   Deaths          20574 non-null  float64
 7   Recovered       20574 non-null  float64
dtypes: float64(3), int64(1), object(4)
memory usage: 1.3+ MB
```

To check if there are any null values, we use this code :

```
In [8]: df.isnull().sum()

Out[8]: SNo                   0
        ObservationDate       0
        Province/State    10548
        Country/Region        0
        Last Update           0
        Confirmed             0
        Deaths                0
        Recovered             0
        dtype: int64
```

```
In [9]: import datetime

In [10]: pd.to_datetime(df["ObservationDate"])

Out[10]: 0       2020-01-22
         1       2020-01-22
         2       2020-01-22
         3       2020-01-22
         4       2020-01-22
         5       2020-01-22
         6       2020-01-22
         7       2020-01-22
         8       2020-01-22
         9       2020-01-22
         10      2020-01-22
         11      2020-01-22
         12      2020-01-22
         13      2020-01-22
         14      2020-01-22
         15      2020-01-22
         16      2020-01-22
         17      2020-01-22
         18      2020-01-22
```

**changing observation date into hours**

```
In [11]: df["ObservationDate"]=pd.to_datetime(df["ObservationDate"])-datetime.timedelta(hours=4)
```

:

**Since our data contains date it will show non-null inorder to change we use the above code**

**We split the date into year, month and day by using this code :**

### splitting Observation date into year, month and day. ¶

In [12]:
```
df["year"]=df.ObservationDate.dt.year
df["month"]=df.ObservationDate.dt.month
df["day"]=df.ObservationDate.dt.day
```

In [13]:
```
df.head(10)
```

Out[13]:

| | SNo | ObservationDate | Province/State | Country/Region | Last Update | Confirmed | Deaths | Recovered | year | month | day |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2020-01-21 20:00:00 | Anhui | Mainland China | 1/22/2020 17:00 | 1.0 | 0.0 | 0.0 | 2020 | 1 | 21 |
| 1 | 2 | 2020-01-21 20:00:00 | Beijing | Mainland China | 1/22/2020 17:00 | 14.0 | 0.0 | 0.0 | 2020 | 1 | 21 |
| 2 | 3 | 2020-01-21 20:00:00 | Chongqing | Mainland China | 1/22/2020 17:00 | 6.0 | 0.0 | 0.0 | 2020 | 1 | 21 |
| 3 | 4 | 2020-01-21 20:00:00 | Fujian | Mainland China | 1/22/2020 17:00 | 1.0 | 0.0 | 0.0 | 2020 | 1 | 21 |
| 4 | 5 | 2020-01-21 20:00:00 | Gansu | Mainland China | 1/22/2020 17:00 | 0.0 | 0.0 | 0.0 | 2020 | 1 | 21 |
| 5 | 6 | 2020-01-21 20:00:00 | Guangdong | Mainland China | 1/22/2020 17:00 | 26.0 | 0.0 | 0.0 | 2020 | 1 | 21 |
| 6 | 7 | 2020-01-21 20:00:00 | Guangxi | Mainland China | 1/22/2020 17:00 | 2.0 | 0.0 | 0.0 | 2020 | 1 | 21 |
| 7 | 8 | 2020-01-21 20:00:00 | Guizhou | Mainland China | 1/22/2020 17:00 | 1.0 | 0.0 | 0.0 | 2020 | 1 | 21 |
| 8 | 9 | 2020-01-21 20:00:00 | Hainan | Mainland China | 1/22/2020 17:00 | 4.0 | 0.0 | 0.0 | 2020 | 1 | 21 |
| 9 | 10 | 2020-01-21 20:00:00 | Hebei | Mainland China | 1/22/2020 17:00 | 1.0 | 0.0 | 0.0 | 2020 | 1 | 21 |

**For more inference we group our data by using this code :**

### Grouping the covid cases for more inference.

In [14]:
```
grouped_df = df[["Country/Region","Confirmed","Deaths","Recovered"]]
grouped_df
```

Out[14]:

| | Country/Region | Confirmed | Deaths | Recovered |
|---|---|---|---|---|
| 0 | Mainland China | 1.0 | 0.0 | 0.0 |
| 1 | Mainland China | 14.0 | 0.0 | 0.0 |
| 2 | Mainland China | 6.0 | 0.0 | 0.0 |
| 3 | Mainland China | 1.0 | 0.0 | 0.0 |
| 4 | Mainland China | 0.0 | 0.0 | 0.0 |
| 5 | Mainland China | 26.0 | 0.0 | 0.0 |
| 6 | Mainland China | 2.0 | 0.0 | 0.0 |
| 7 | Mainland China | 1.0 | 0.0 | 0.0 |
| 8 | Mainland China | 4.0 | 0.0 | 0.0 |
| 9 | Mainland China | 1.0 | 0.0 | 0.0 |
| 10 | Mainland China | 0.0 | 0.0 | 0.0 |

In [15]:
```
grouped_df = grouped_df.sort_values(by="Confirmed",ascending=False)
grouped_df = grouped_df.reset_index(drop=True)
grouped_df
```

Out[15]:

| | Country/Region | Confirmed | Deaths | Recovered |
|---|---|---|---|---|
| 0 | US | 308314.0 | 24039.0 | 0.0 |
| 1 | US | 304372.0 | 23587.0 | 0.0 |
| 2 | US | 299691.0 | 23477.0 | 0.0 |
| 3 | US | 295106.0 | 22912.0 | 0.0 |
| 4 | US | 291996.0 | 22668.0 | 0.0 |
| 5 | US | 288045.0 | 22269.0 | 0.0 |
| 6 | US | 282143.0 | 22009.0 | 0.0 |
| 7 | US | 271590.0 | 21411.0 | 0.0 |
| 8 | US | 263460.0 | 20973.0 | 0.0 |
| 9 | US | 263292.0 | 19413.0 | 0.0 |
| 10 | US | 258361.0 | 19104.0 | 0.0 |

In [16]:
```
grouped=df[["Confirmed","Deaths","Recovered","Country/Region"]]
grouped.head()
```

Out[16]:

| | Confirmed | Deaths | Recovered | Country/Region |
|---|---|---|---|---|
| 0 | 1.0 | 0.0 | 0.0 | Mainland China |
| 1 | 14.0 | 0.0 | 0.0 | Mainland China |
| 2 | 6.0 | 0.0 | 0.0 | Mainland China |
| 3 | 1.0 | 0.0 | 0.0 | Mainland China |
| 4 | 0.0 | 0.0 | 0.0 | Mainland China |

In [17]:
```
df.head()
```

Out[17]:

| | SNo | ObservationDate | Province/State | Country/Region | Last Update | Confirmed | Deaths | Recovered | year | month | day |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2020-01-21 20:00:00 | Anhui | Mainland China | 1/22/2020 17:00 | 1.0 | 0.0 | 0.0 | 2020 | 1 | 21 |
| 1 | 2 | 2020-01-21 20:00:00 | Beijing | Mainland China | 1/22/2020 17:00 | 14.0 | 0.0 | 0.0 | 2020 | 1 | 21 |
| 2 | 3 | 2020-01-21 20:00:00 | Chongqing | Mainland China | 1/22/2020 17:00 | 6.0 | 0.0 | 0.0 | 2020 | 1 | 21 |
| 3 | 4 | 2020-01-21 20:00:00 | Fujian | Mainland China | 1/22/2020 17:00 | 1.0 | 0.0 | 0.0 | 2020 | 1 | 21 |
| 4 | 5 | 2020-01-21 20:00:00 | Gansu | Mainland China | 1/22/2020 17:00 | 0.0 | 0.0 | 0.0 | 2020 | 1 | 21 |

**Exploratory Data Analysis:**

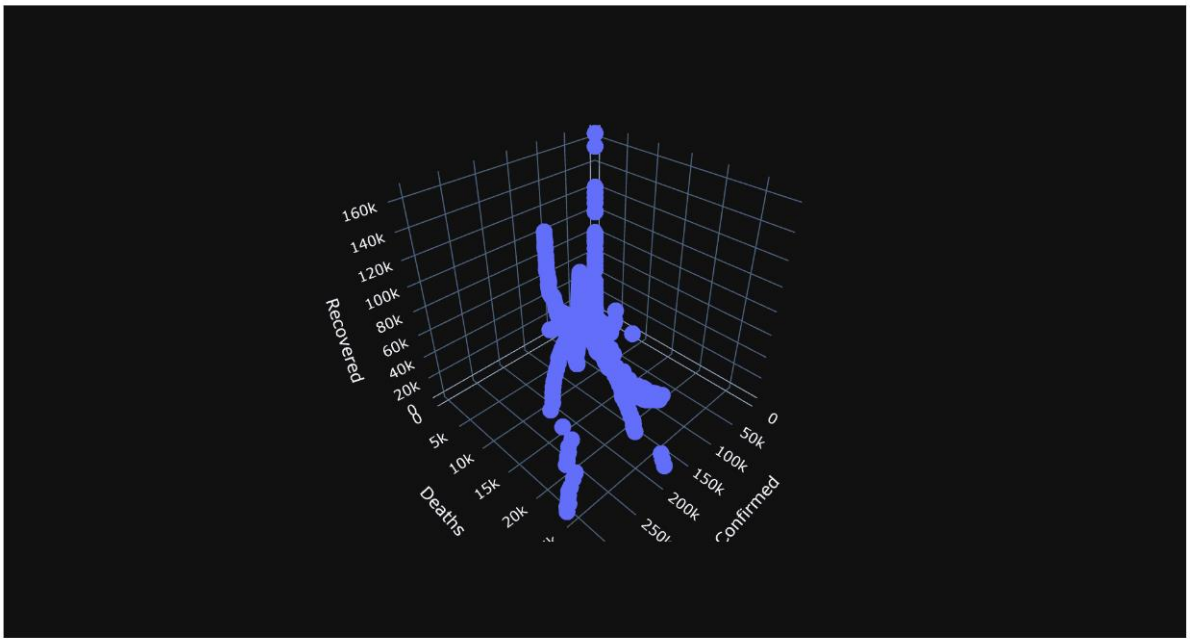
The visualization techniques that are used in this project are

1. Scatter plot
2. Bar Plot
3. histogram
4. piePlot

1. scatter plot:

A **scatter plot** (also called a **scatterplot**, **scatter graph**, **scatter chart**, **scattergram**, or **scatter diagram**)is a type of plot or mathematical diagram using Cartesian coordinates to display values for typically two variables for a set of data. If the points are coded (color/shape/size), one additional variable can be displayed. The data are displayed as a collection of points, each having the value of one variable determining the position on the horizontal axis and the value of the other variable determining the position on the vertical axis.

```
In [19]: px.scatter_3d(grouped,x="Confirmed",y="Deaths",z="Recovered")
```

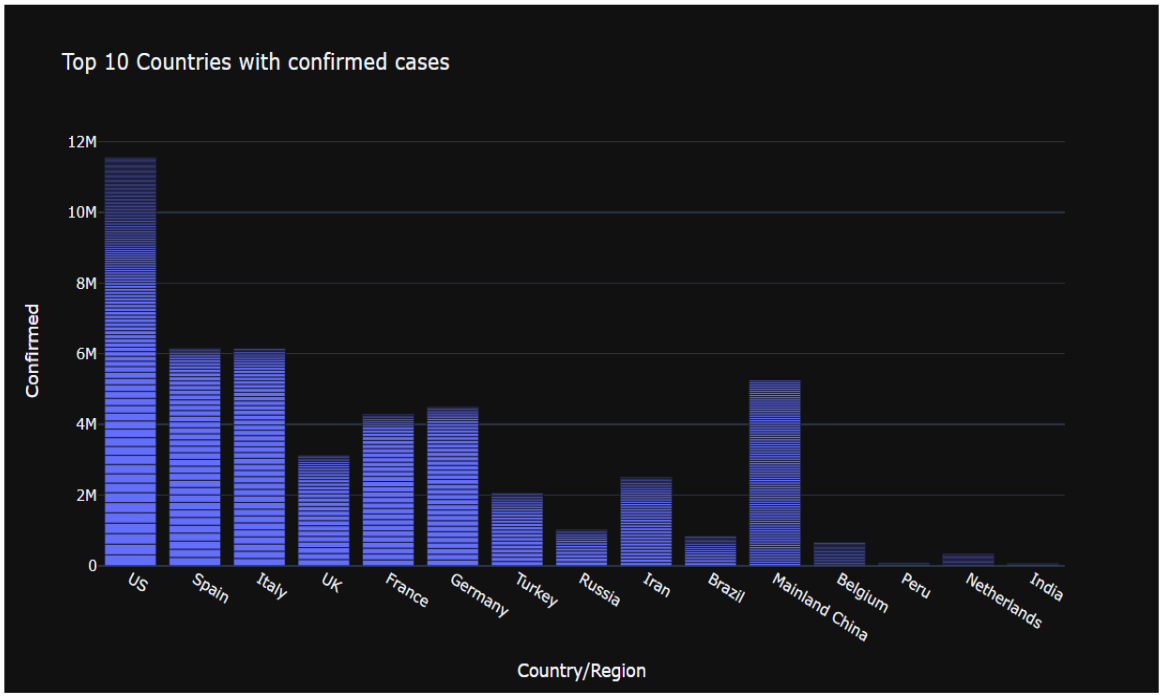


2.Bar Plot:

A bar plot or bar chart is a graph that represents the category of data with rectangular bars with lengths and heights that is proportional to the values which they represent. The bar plots can be plotted horizontally or vertically. A bar chart describes the comparisons between the discrete categories. One of the axis of the plot represents the specific categories being compared, while the other axis represents the measured values corresponding to those categories.

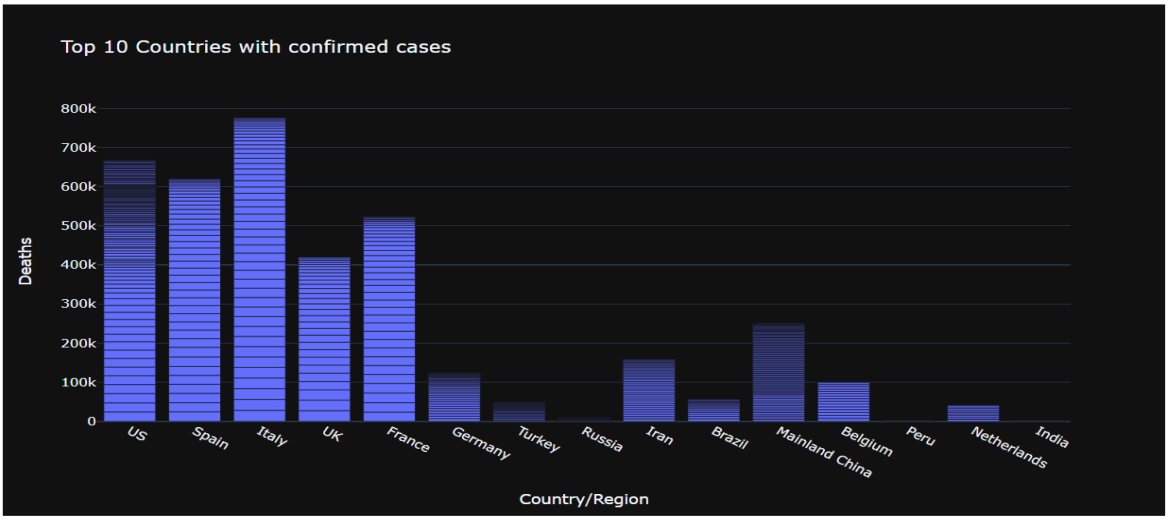
This is for confirmed cases :

```
In [20]: px.bar(grouped_df[0:500], x="Country/Region", y="Confirmed",title="Top 10 Countries with confirmed cases")
```




This is for death cases :

```
In [21]: px.bar(grouped_df[0:500], x="Country/Region", y="Deaths",title="Top 10 Countries with confirmed cases")
```

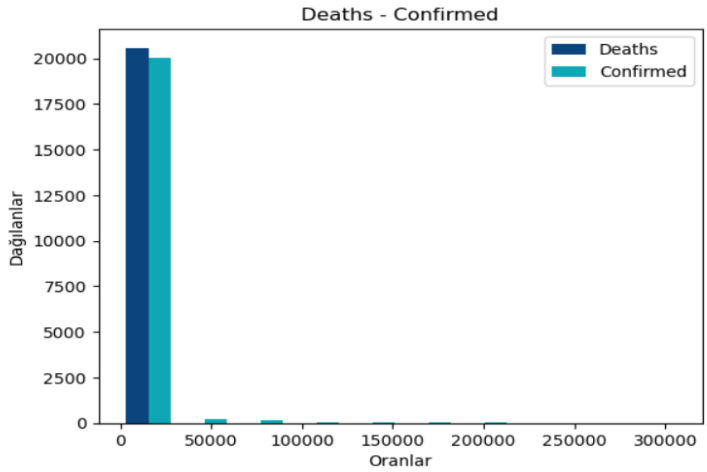


3.Histogram :

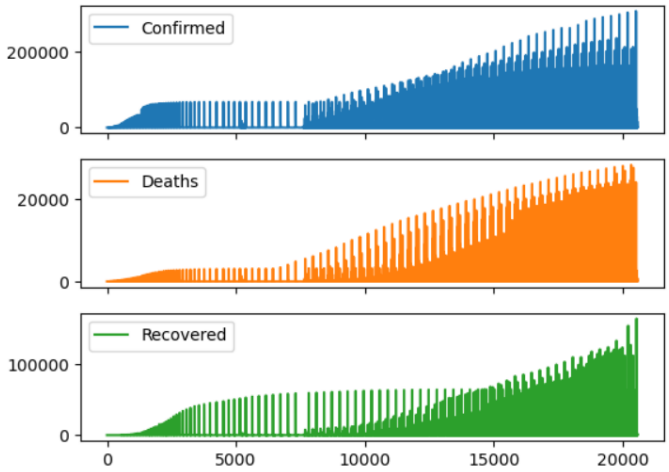
A histogram is a graphical representation of data points organized into user-specified ranges. Similar in appearance to a bar graph, the histogram condenses a data series into an easily interpreted visual by taking many data points and grouping them into logical ranges or bins.

```
plt.hist([df.Deaths,df.Confirmed],
        color=["#0c457d", "#0ea7b5"],
        label=["Deaths","Confirmed"])

plt.xlabel("Oranlar")
plt.ylabel("Dağılanlar")
plt.title("Deaths - Confirmed")
plt.legend()
plt.show()
```



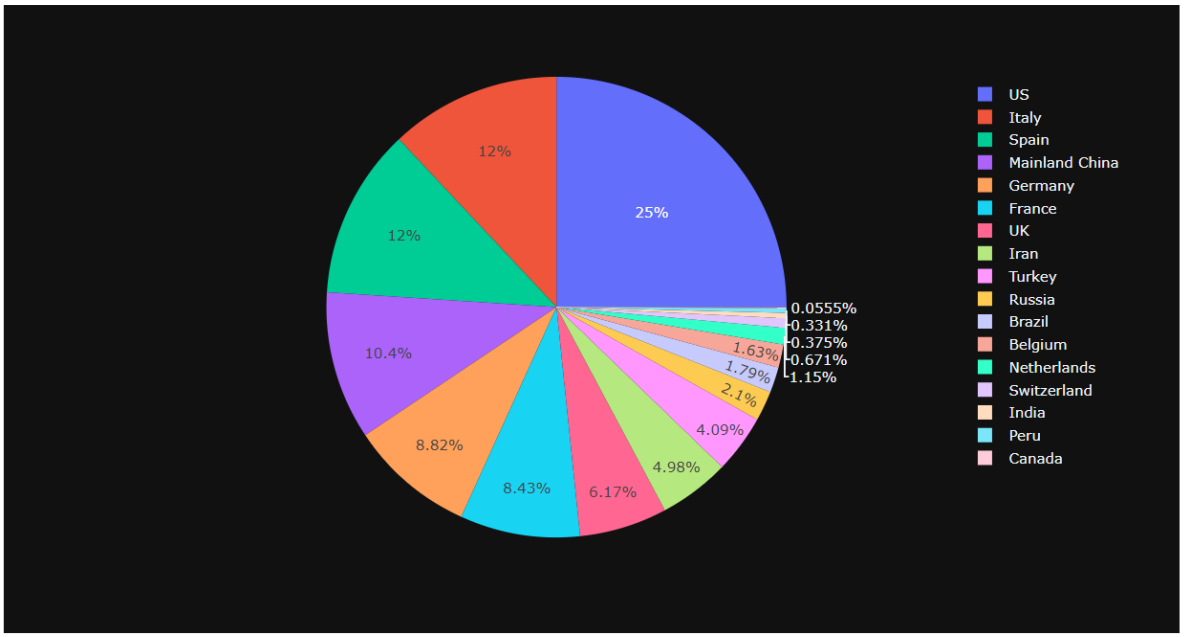

In [23]:
```
# we can draw subplots.
df = df.loc[:, ["Confirmed", "Deaths", "Recovered"]]
df.plot(subplots = True)
plt.show()
```



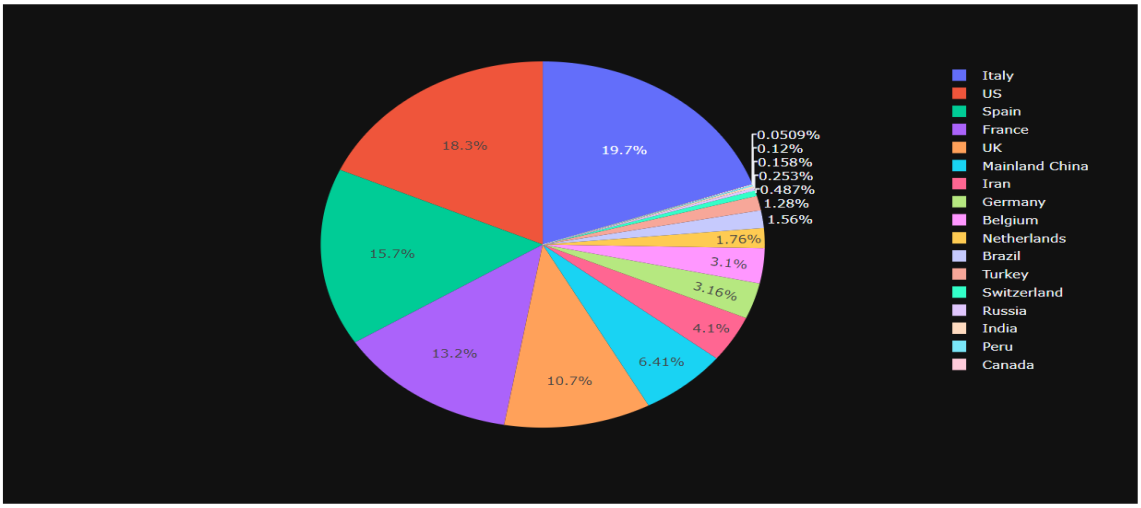
5.pie chart :

A **pie chart** (or a **circle chart**) is a circular statistical graphic, which is divided into slices to illustrate numerical proportion. In a pie chart, the arc length of each slice (and consequently its central angle and area) is proportional to the quantity it represents. While it is named for its resemblance to a pie which has been sliced, there are variations on the way it can be presented. The earliest known pie chart is generally credited to William Playfair's *Statistical Breviary* of 1801.Pie charts are very widely used in the business world and the mass media.

```
In [24]: px.pie(grouped_df[0:600],values="Confirmed",names="Country/Region")
```




```
In [25]: px.pie(grouped_df[0:600],values="Deaths",names="Country/Region")
```



**Splitting Dataset into Train-Test data:**

The train-test split is used to estimate the performance of machine learning algorithms that are applicable for prediction-based Algorithms/Applications. This method is a fast and easy procedure to perform such that we can compare our own machine learning model results to machine results. By default, the Test set is split into 30 % of actual data and the training set is split into 70% of the actual data. Scikit-learn alias "**sklearn"** is the most useful and robust library for machine learning in Python. The **scikit-learn library** provides us with the "model_selection" module in which we have the splitter function "train_test_split ()."

**Model Fitting:**

In this project we used most of the Regression Algorithms like

1. Linear Regression
2. Kneighbours regressor
3. Decision tree
4. Adaboost
5. Gradient boosting
6. Random forest
7. Gridsearch cv

1.Linear Regression:

In statistics, linear regression is a linear approach for modelling the relationship between a scalar response and one or more explanatory variables (also known as dependent and independent variables). The case of one explanatory variable is called *simple linear regression*; for more than one, the process is called **multiple** linear regression. This term is distinct from multivariate linear regression, where multiple correlated dependent variables are predicted, rather than a single scalar variable.

In linear regression, the relationships are modelled using linear predictor functions whose unknown model parameters are estimated from the data. Such models are called linear models. Most commonly, the conditional mean of the response given the values of the explanatory variables (or predictors) is assumed to be an affine function of those values; less commonly, the conditional median or some other quantile is used. Like all forms of regression analysis, linear regression focuses on the conditional probability distribution of the response given the values of the predictors, rather than on the joint probability distribution of all of these variables, which is the domain of multivariate analysis.

Linear regression was the first type of regression analysis to be studied rigorously, and to be used extensively in practical applications. This is because models which depend linearly on their unknown parameters are easier to fit than models which are non-linearly related to their parameters and because the statistical properties of the resulting estimators are easier to determine.

Given a data set $\{y_i,\ x_{i1}, \ldots, x_{ip}\}_{i=1}^{n}$ of $n$ statistical units, a linear regression model assumes that the relationship between the dependent variable $y$ and the $p$-vector of regressors $\mathbf{x}$ is linear. This relationship is modeled through a *disturbance term* or *error variable* $\varepsilon$ — an unobserved random variable that adds "noise" to the linear relationship between the dependent variable and regressors. Thus, the model takes the form

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^{\mathsf{T}} \boldsymbol{\beta} + \varepsilon_i, \qquad i = 1, \ldots, n,$$

2.kneighbors regressor :

- kneighbors regressor: K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

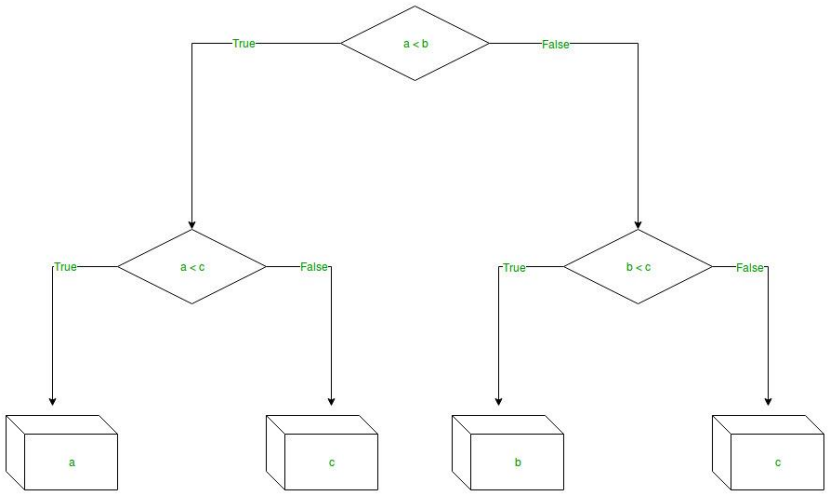
3.Decision Tree Regression:

Decision Tree is a decision-making tool that uses a flowchart-like tree structure or is a model of decisions and all of their possible results, including outcomes, input costs, and utility. Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables.

The branches/edges represent the result of the node and the nodes have either:

1. Conditions [Decision Nodes]
2. Result [End Nodes]

The branches/edges represent the truth/falsity of the statement and take makes a decision based on that in




the example below which shows a decision tree that evaluates the smallest of three numbers:

Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output. Continuous output means that the output/result is not discrete, i.e., it is not represented just by a discrete, known set of numbers or values.

4.adaboost:

**AdaBoost**, short for *Adaptive Boosting*, is a statistical classification meta-algorithm formulated by Yoav Freund and Robert Schapire in 1995, who won the 2003 Gödel Prize for their work. It can be used in conjunction with many other types of learning algorithms to improve performance. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. Usually, AdaBoost is presented for binary classification, although it can be generalized to multiple classes or bounded intervals on the real line.

AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. In some problems it can be less susceptible to the overfitting problem than other learning algorithms. The individual learners can be weak, but as long as the performance of each one is slightly better than random guessing, the final model can be proven to converge to a strong learner.

Although AdaBoost is typically used to combine weak base learners (such as decision stumps), it has been shown that it can also effectively combine strong base learners (such as deep decision trees), producing an even more accurate model.

Every learning algorithm tends to suit some problem types better than others, and typically has many different parameters and configurations to adjust before it achieves optimal performance on a dataset. AdaBoost (with decision trees as the weak learners) is often referred to as the best out-of-the-box classifier. When used with decision tree learning, information gathered at each stage of the AdaBoost algorithm about the relative 'hardness' of each training sample is fed into the tree growing algorithm such that later trees tend to focus on harder-to-classify examples.

5.gradient boosting :

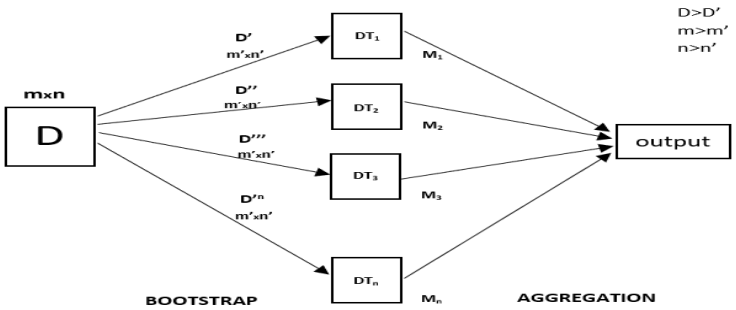
**Gradient boosting** is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees. When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest. A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.Gradient boosting is a method standing out for its prediction speed and accuracy, particularly with large and complex datasets. From Kaggle competitions to machine learning solutions for business, this algorithm has produced the best results. We already know that errors play a major role in any machine learning algorithm. There are mainly two types of error, bias error and variance error. Gradient boost algorithm *helps us minimize bias error* of the model

The main idea behind this algorithm is to build models sequentially and these subsequent models try to reduce the errors of the previous model. But how do we do that? How do we reduce the error? This is done by building a new model on the errors or residuals of the previous model.

When the target column is continuous, we use **Gradient Boosting Regressor** whereas when it is a classification problem, we use **Gradient Boosting Classifier**. The only difference between the two is the *"Loss function"*. The objective here is to minimize this loss function by adding weak learners using gradient descent. Since it is based on loss function hence for regression problems, we'll have different loss functions like Mean squared error (**MSE**) and for classification, we will have different for e.g. **log-likelihood**

6. Random Forest:

Every decision tree has high variance, but when we combine all of them together in parallel then the resultant variance is low as each decision tree gets perfectly trained on that particular sample data, and hence the output doesn't depend on one decision tree but on multiple decision trees. In the case of a classification problem, the final output is taken by using the majority voting classifier. In the case of a regression problem,




the final output is the mean of all the outputs. This part is called **Aggregation**. Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as **bagging**.

7.Gridsearch cv :

GridSearchCV is the process of performing hyperparameter tuning in order to determine the optimal values for a given model. As mentioned above, the performance of a model significantly depends on the value of hyperparameters. Note that there is no way to know in advance the best values for hyperparameters so ideally, we need to try all possible values to know the optimal values. Doing this manually could take a considerable amount of time and resources and thus we use GridSearchCV to automate the tuning of hyperparameters.

GridSearchCV is a function that comes in Scikit-learn's(or SK-learn) model_selection package. So an important point here to note is that we need to have the Scikit learn library installed on the computer. This function helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, we can select the best parameters from the listed hyperparameters.Grid search is a method for performing hyper-parameter optimisation, that is, with a given model (e.g. a CNN) and test dataset, it is a method for finding the optimal combination of hyper-parameters (an example of a hyper-parameter is the learning rate of the optimiser). You have numerous models in this case, each with a different set of hyper-parameters. Each of these parameter combinations that correspond to a single model is said to lie on a "grid" point. The purpose is to train and evaluate each of these models using cross-validation, for example. Then you choose the one that performed the best.

CODING:

## 1.LINEAR REGRESSION :

```
In [27]: from sklearn.linear_model import LinearRegression

In [28]: x=df.drop(['Deaths'],axis=1)
         y=df[['Deaths']]
```

**Splitting the data set for trainig and testing**

```
In [29]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=1)

In [30]: reg=LinearRegression()
         reg.fit(x_train,y_train)

Out[30]: LinearRegression()

In [31]: reg_pred=reg.predict(x_test)
         print(reg_pred)

         [[ -80.05382569]
          [ -82.72056341]
          [ 233.27432316]
          ...
          [1646.20895498]
          [ 101.31752114]
          [ -82.13002395]]

In [32]: reg.score(x_train,y_train)

Out[32]: 0.8248674138111862

In [33]: reg.score(x_test,y_test)

Out[33]: 0.7916365540382582
```

## 2.KNEIGHBORS REGRESSOR :

```
In [34]: from sklearn.neighbors import KNeighborsRegressor
         from scipy.stats import zscore

In [35]: xscaled=x.apply(zscore) # convert all attributes
         xscaled.describe()
```

Out[35]:

| | Confirmed | Recovered |
|---|---|---|
| count | 2.057400e+04 | 2.057400e+04 |
| mean | 9.764960e-16 | 1.203251e-14 |
| std | 1.000024e+00 | 1.000024e+00 |
| min | -2.119369e-01 | -1.417177e-01 |
| 25% | -2.113168e-01 | -1.417177e-01 |
| 50% | -2.048905e-01 | -1.414341e-01 |
| 75% | -1.637396e-01 | -1.285274e-01 |
| max | 1.716806e+01 | 2.312085e+01 |

```
In [36]: NNH =KNeighborsRegressor(n_neighbors=14)
         NNH.fit(x_train,y_train)

Out[36]: KNeighborsRegressor(n_neighbors=14)

In [37]: predicted_label=NNH.predict(x_test)
         NNH.score(x_train,y_train)

Out[37]: 0.936211803808958

In [38]: NNH.score(x_test,y_test)

Out[38]: 0.9211121033553978
```

3. DECISION TREE:

## Decision tree model

**Note: Decision tree model got overfitted due to, as the data set is too large, so decision tree will not work well with large data sets.**

```
In [39]: from sklearn.tree import DecisionTreeRegressor

In [40]: a = DecisionTreeRegressor()

In [41]: a.fit(x_train,y_train)
Out[41]: DecisionTreeRegressor()

In [42]: a.score(x_train,y_train)
Out[42]: 0.9999976554753546

In [43]: a.score(x_test,y_test)
Out[43]: 0.8294739605766732
```

4.ADAB0OST:

## AdaBoostRegressor Model.

```
In [44]: from sklearn.ensemble import AdaBoostRegressor

In [45]: cs=AdaBoostRegressor()

In [46]: cs.fit(x_train,y_train)
Out[46]: AdaBoostRegressor()

In [47]: cs.score(x_train,y_train)
Out[47]: 0.908202268845154

In [48]: cs.score(x_test,y_test)
Out[48]: 0.8596842543684886
```

**5.**GRADIENTBOOSTING:

## GradientBoostingRegressor Model.

```
In [49]: from sklearn.ensemble import GradientBoostingRegressor

In [50]: d = GradientBoostingRegressor()

In [51]: d.fit(x_train,y_train)
Out[51]: GradientBoostingRegressor()

In [52]: d.score(x_train,y_train)
Out[52]: 0.9655329015269836

In [53]: d.score(x_test,y_test)
Out[53]: 0.885946568147981
```

## 6.RANDOMFOREST:

**RandomForestRegressor Model.**

```
In [54]: from sklearn.ensemble import  RandomForestRegressor

In [55]: c = RandomForestRegressor()

In [56]: c.fit(x_train,y_train)
Out[56]: RandomForestRegressor()

In [57]: c.score(x_train,y_train)
Out[57]: 0.986998306529837

In [58]: c.score(x_test,y_test)
Out[58]: 0.8903098288475444
```

## 7. GRIDSEARCH CV:

```
In [64]: grid_search.best_params_
Out[64]: {'bootstrap': True,
          'max_depth': 6,
          'max_features': 2,
          'min_samples_leaf': 3,
          'min_samples_split': 5,
          'n_estimators': 6}

In [65]: best_grid = grid_search.best_estimator_
         best_grid.score(x_test,y_test)
Out[65]: 0.8948816950008839

In [66]: best_grid.score(x_train,y_train)
Out[66]: 0.9551516573568353

In [67]: rf.fit(x_train,y_train)
Out[67]: RandomForestRegressor(random_state=1)

In [68]: rf.score(x_train,y_train)
Out[68]: 0.9870954297208457

In [69]: rf.score(x_test,y_test)
Out[69]: 0.8894145627239382
```

# CHAPTER8
# RESULTS
# AND
# CHALLENGES FACED

## RESULT :

- Out of all the Machine Learning Algorithms we have fitted the models with Linear Regression, decision tree, knn, adaboost, random forest Model .
- In that we have got good score in linear regression,adaboost models.
- Rest of the models have we have got overfitting.
- Random forest and decision tree algorithms model has got over fitting.
- Decision tree will not work well with large datasets.
- According to this covid dataset there are more death cases than confirmed cases
- So, by taking all the accuracies into note, we selected Linear Regression algorithm as the ideal Machine Learning Regression Algorithm for prediction of deaths.

## CHALLENGES FACED :

When we first collected our dataset from Kaggle database we found that there are null values in it.and it contained datesin the data which has to be converted into numerical values. So, we faced many challenges during the conversion of Numerical values.

# CHAPTER9
# CONCLUSION
# REFERENCES
# APPENDIX

## CONCLUSION:

This project helps doctors and common people who wants to know about death cases. We have plotted many Visualizations from the covid-19 dataset. From those visualizations we have to select useful insights and make use of them in order to increase healthcare facilities to the affected people. We have applied most of the Regression models and from those models, Linear Regression Model gives predictions with the highest accuracy.

## REFERENCES:


- Kaggle.com
- GeeksforGeeks
- Tutorials Point
- Wikipedia
- Javatpoint
- Analytics Vidya
- Towards Data science