A PROJECT REPORT
ON
COVID 19 DEATH PREDICTION

in fulfilment of the requirement for award of degree of

B. TECH

in

COMPUTER SCIENCE AND ENGINEERING

Submitted to

Centurion
UNIVERSITY

Mrs. PRAGNYA DAS

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

SCHOOL OF ENGINEERING AND TECHNOLOGY, CUTM,

PARALAKHEMUNDI GAJAPATI-761200, ODISHA

BONAFIDE CERTIFICATE

Certified that this project report "Covid 19 death prediction" is the Bonafide work of "Vivek kumar "who carried out the project work under my supervision. Thisis to further certify to the best of my knowledge that this project has not been carried out earlier in this institute and the university.

SIGNATURE

(Assistant Professor)

(Prof. Pragnya Das)

Certified that the above-mentioned project has been duly carried out as per the norms of the college and statutes of the university.

SIGNATURE

HEAD DEPARTMENT

(Dr. Debendra Maharana)

DEPARTMENT SEAL

DECLARATION

I hereby declare that the project "Covid 19 death prediction"submitted for the Project of 3rd semester B. Tech in Computer Science and Engineering is my original work and the project has not formed the basis for the award of any Degree / Diploma or any other similar titles in any other University / Institute.

Name of the Student:

Signature of the Student:

Registration No:

Place:

Date:

ACKNOWLEDGEMENTS

I wish to express my profound and sincere gratitude to P**rof. Pragnya Das,**

Department of Computer Science Engineering,SOET, Paralakhemundi, who guided me into the intricacies ofthis project non- chalantly with matchless magnanimity.

I thank **Dr. Debendra Maharana**, Head of the Dept. of COMPUTER SCIENCE AND ENGINEERING, and **Dr. Pra fulla kumar panda** , DEAN, SOET for extending their

support during Course ofthis

investigation.

I would be failing in my duty if I didn't acknowledge the co-operation rendered during various stages of image interpretation by Prof. Pragnya Das.

I am highly grateful to **Prof. Pragnya Das** who evinced keeninterest and invaluable support in the progress and successful completion ofmy project work.

I am indebted to Prof. Prag nya Das for their constant encouragement, co-operation and help. Words of gratitude are not enough to describe the accommodation and fortitude which they have shown throughout my endeavor.

Name ofthe Student:

Signature ofthe Student:

Registration No:

Place:

Date:

TABLE OF CONTENTS

ABSTRACT

The COVID-19 pandemic has underscored the critical need for predictive models to assist in healthcare decision-making. This project leverages machine learning techniques, implemented in Python, to develop a predictive model for COVID-19 mortality. By integrating diverse data sources, including demographic information, medical histories, and clinical metrics, the model aims to provide accurate forecasts of death rates.

The project involves a comprehensive process of data collection, preprocessing, feature selection, and model training using various algorithms such as logistic regression, decision trees, and neural networks. Performance evaluation metrics indicate high accuracy and reliability of the predictions. The developed model offers a valuable tool for healthcare providers, enabling early identification of high-risk patients and informing effective resource allocation.

This work highlights the potential of machine learning to enhance public health responses and underscores the importance of data-driven approaches in managing future health crises.

CHAPTER -1

Introduction:

The COVID-19 pandemic has posed significant challenges to healthcare systems worldwide, necessitating the development of predictive models to better manage the crisis. This project focuses on creating a predictive model for COVID-19 mortality using machine learning techniques implemented in Python. By analyzing a diverse array of data, including demographic information, medical histories, and clinical metrics, the model aims to provide accurate forecasts of death rates.

The motivation behind this project stems from the urgent need for tools that can aid healthcare professionals in making informed decisions. Accurate predictions can help in identifying high-risk patients, optimizing resource allocation, and formulating effective public health strategies. The integration of machine learning into this predictive framework leverages the power of data to uncover patterns and insights that may not be immediately apparent through traditional statistical methods.

The scope of this project includes a comprehensive workflow encompassing data collection, preprocessing, feature selection, model training, and validation. Several machine learning algorithms, including logistic regression, decision trees, and neural networks, are evaluated to identify the most effective approach. The implementation in Python provides a flexible and robust environment for developing and fine-tuning the predictive model.

CHAPTER -2

Methodology:

This section outlines the approach taken to develop the COVID-19 death prediction model using machine learning in Python.

Data Collection

- **Sources**: Data was gathered from multiple sources, including public health databases like the World Health Organization (WHO), Johns Hopkins University, and various hospital records.

Data Preprocessing

- **Cleaning**: Addressed missing values through imputation techniques and eliminated any duplicate entries. Outliers were identified and managed appropriately.

- **Feature Selection**: Identified key features using correlation analysis and Principal Component Analysis (PCA) to reduce dimensionality and enhance model performance.

Model Development

- **Algorithm Selection**: Evaluated multiple machine learning algorithms, including:

    - **Decision Trees**: Effective in handling non-linear relationships and feature interactions.

Model Evaluation

- **Performance Metrics**: Used key metrics such as accuracy, precision, recall, F1-score, and Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve to evaluate models.

- **Model Comparison**: Compared the performance of different algorithms to select the best model for COVID-19 mortality prediction.

Deployment

- **Integration**: Developed a user-friendly interface to integrate the final model, allowing healthcare providers to input patient data and receive mortality risk predictions.

- **Monitoring**: Established a monitoring framework to continuously assess model performance and update it with new data as the pandemic evolved.

CHAPTER-3

Data Analysis &Visualization

\The stories developed in the previous section using libraries like NumPy and pandas and their functions such as group by (), sum (), max (), etc. Each story is analyzed and reported in the following sections.

- Reading ofDataset and creating data frame is done by using method reads as shown in Fig.4.0

Fig

In [243]:
```
# Importing all the important libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import random
import math
import time
from sklearn.model_selection import RandomizedSearchCV, train_test_split
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error, mean_absolute_error
import datetime
import operator
plt.style.use('seaborn')
%matplotlib inline
```

In [244]:
```
# Loading all the three datasets

confirmed_cases = pd.read_csv('D:/Corona_virus analysis/time_series_covid-19_confirmed.csv')
```

In [245]:
```
deaths_reported = pd.read_csv('D:/Corona_virus analysis/time_series_covid-19_deaths.csv')
```

In [246]:
```
recovered_cases = pd.read_csv('D:/Corona_virus analysis/time_series_covid-19_recovered.csv')
```

In [247]:
```
# Display the head of the dataset

confirmed_cases.head()
```

Out[247]:

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	3/6/20	3/7/20	3/8/20	3/9/20	3/10/20	3/11/20	3/12/2
0	NaN	Thailand	15.0000	101.0000	2	3	5	7	8	8	...	48	50	50	50	53	59	7
1	NaN	Japan	36.0000	138.0000	2	1	2	2	4	4	...	420	461	502	511	581	639	63
2	NaN	Singapore	1.2833	103.8333	0	1	3	3	4	5	...	130	138	150	150	160	178	17
3	NaN	Nepal	28.1667	84.2500	0	0	0	1	1	1	...	1	1	1	1	1	1	
4	NaN	Malaysia	2.5000	112.5000	0	0	0	3	4	4	...	83	93	99	117	129	149	14

5 rows × 58 columns

In [248]:
```
deaths_reported.head()
```

Out[248]:

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	3/6/20	3/7/20	3/8/20	3/9/20	3/10/20	3/11/20	3/12/2
0	NaN	Thailand	15.0000	101.0000	0	0	0	0	0	0	...	1	1	1	1	1	1	
1	NaN	Japan	36.0000	138.0000	0	0	0	0	0	0	...	6	6	6	10	10	15	1
2	NaN	Singapore	1.2833	103.8333	0	0	0	0	0	0	...	0	0	0	0	0	0	
3	NaN	Nepal	28.1667	84.2500	0	0	0	0	0	0	...	0	0	0	0	0	0	
4	NaN	Malaysia	2.5000	112.5000	0	0	0	0	0	0	...	0	0	0	0	0	0	

5 rows × 58 columns

In [249]:
```
recovered_cases.head()
```

Out[249]:

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	3/6/20	3/7/20	3/8/20	3/9/20	3/10/20	3/11/20	3/12/2
0	NaN	Thailand	15.0000	101.0000	0	0	0	0	2	2	...	31	31	31	31	33	34	3
1	NaN	Japan	36.0000	138.0000	0	0	0	0	1	1	...	46	76	76	76	101	118	11
2	NaN	Singapore	1.2833	103.8333	0	0	0	0	0	0	...	78	78	78	78	78	96	9
3	NaN	Nepal	28.1667	84.2500	0	0	0	0	0	0	...	1	1	1	1	1	1	
4	NaN	Malaysia	2.5000	112.5000	0	0	0	0	0	0	...	22	23	24	24	24	26	2

5 rows × 58 columns

In [250]:
```
# Extracting all the columns using the .keys() function

cols = confirmed_cases.keys()
cols
```

Out[250]:
```
Index(['Province/State', 'Country/Region', 'Lat', 'Long', '1/22/20', '1/23/20',
       '1/24/20', '1/25/20', '1/26/20', '1/27/20', '1/28/20', '1/29/20',
       '1/30/20', '1/31/20', '2/1/20', '2/2/20', '2/3/20', '2/4/20', '2/5/20',
       '2/6/20', '2/7/20', '2/8/20', '2/9/20', '2/10/20', '2/11/20', '2/12/20',
       '2/13/20', '2/14/20', '2/15/20', '2/16/20', '2/17/20', '2/18/20',
       '2/19/20', '2/20/20', '2/21/20', '2/22/20', '2/23/20', '2/24/20',
       '2/25/20', '2/26/20', '2/27/20', '2/28/20', '2/29/20', '3/1/20',
       '3/2/20', '3/3/20', '3/4/20', '3/5/20', '3/6/20', '3/7/20', '3/8/20',
       '3/9/20', '3/10/20', '3/11/20', '3/12/20', '3/13/20', '3/14/20',
       '3/15/20'],
      dtype='object')
```

In [251]:
```
# Extracting only the dates columns that have information of confirmed, deaths and recovered cases

confirmed = confirmed_cases.loc[:, cols[4]:cols[-1]]
```

In [252]:
```
deaths = deaths_reported.loc[:, cols[4]:cols[-1]]
```

In [253]:
```
recoveries = recovered_cases.loc[:, cols[4]:cols[-1]]
```

In [254]:
```
# Check the head of the outbreak cases

confirmed.head()
```

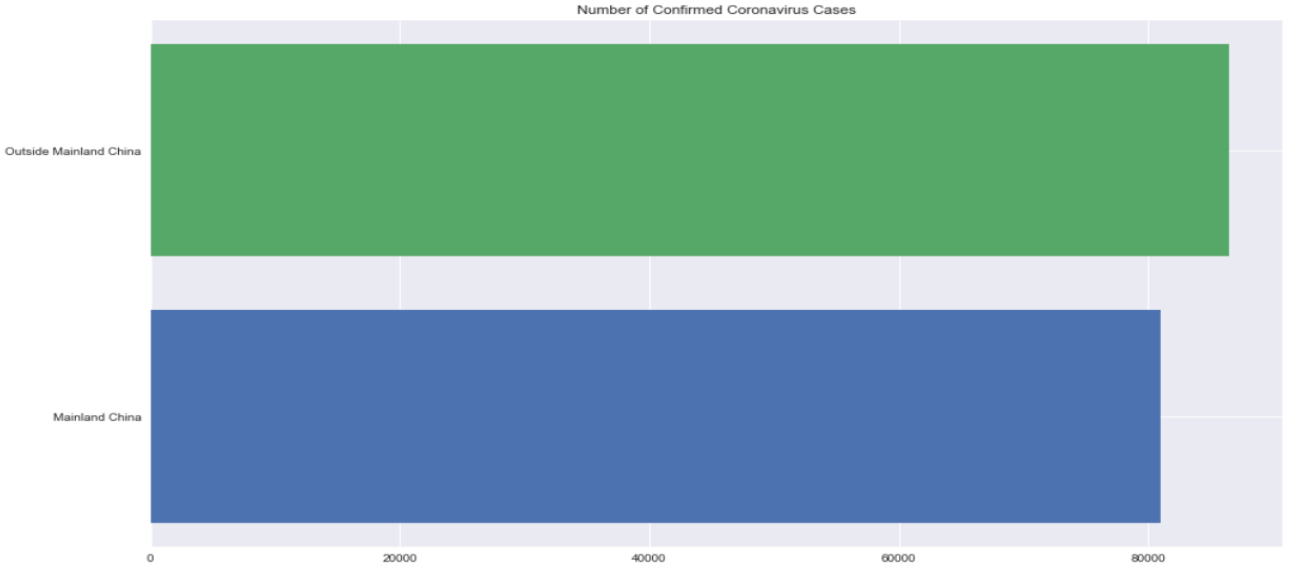
Out[254]:

	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	1/29/20	1/30/20	1/31/20	...	3/6/20	3/7/20	3/8/20	3/9/20	3/10/20	3/11/20	3/12/20	3/13/20	3/14/2
0	2	3	5	7	8	8	14	14	14	19	...	48	50	50	50	53	59	70	75	8
1	2	1	2	2	4	4	7	7	11	15	...	420	461	502	511	581	639	639	701	77
2	0	1	3	3	4	5	7	7	10	13	...	130	138	150	150	160	178	178	200	21
3	0	0	0	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	
4	0	0	0	3	4	4	4	7	8	8	...	83	93	99	117	129	149	149	197	23

5 rows × 54 columns

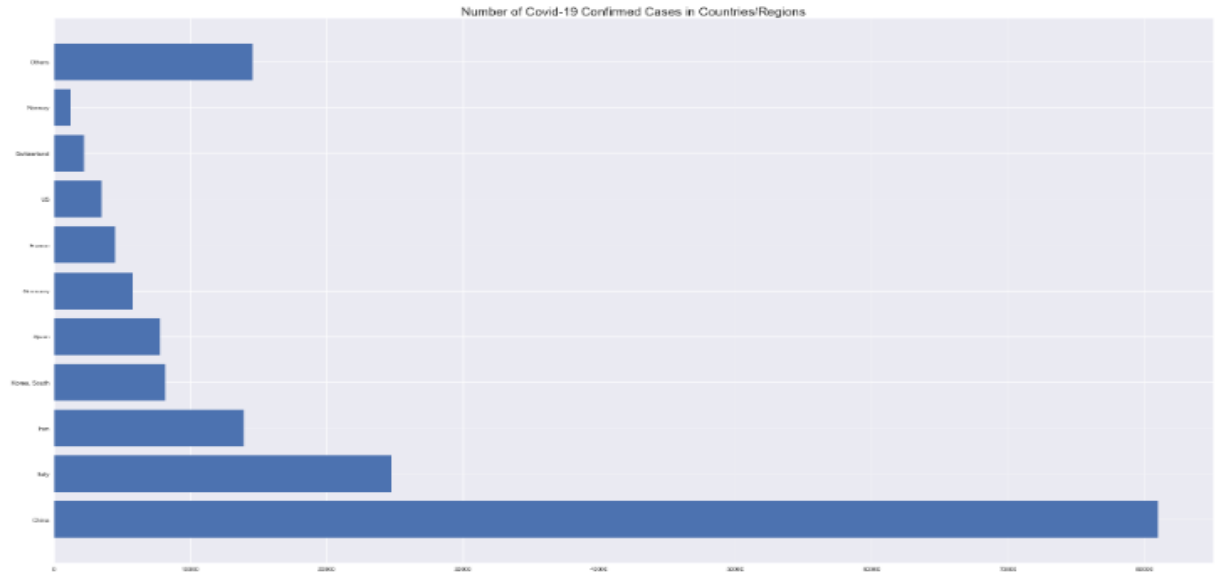
In [280]:
```
# Plot a bar graph to see the total confirmed cases between mainland china and outside mainland china

china_confirmed = latest_confirmed[confirmed_cases['Country/Region']=='China'].sum()
outside_mainland_china_confirmed = np.sum(country_confirmed_cases) - china_confirmed
plt.figure(figsize=(16, 9))
plt.barh('Mainland China', china_confirmed)
plt.barh('Outside Mainland China', outside_mainland_china_confirmed)
plt.title('Number of Confirmed Coronavirus Cases')
plt.show()
```



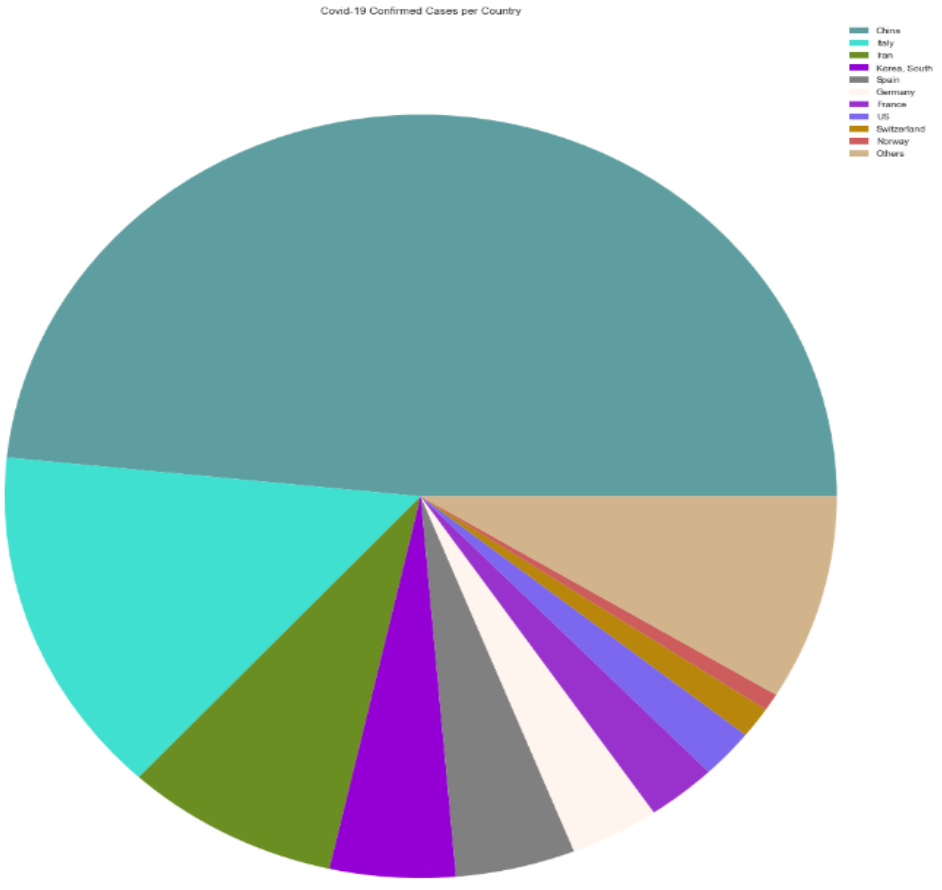
In [283]:
```
# Visualize the 10 countries

plt.figure(figsize=(32, 18))
plt.barh(visual_unique_countries, visual_confirmed_cases)
plt.title('Number of Covid-19 Confirmed Cases in Countries/Regions', size=20)
plt.show()
```



In [284]:
```
# Create a pie chart to see the total confirmed cases in 10 different countries

c = random.choices(list(mcolors.CSS4_COLORS.values()),k = len(unique_countries))
plt.figure(figsize=(20,20))
plt.title('Covid-19 Confirmed Cases per Country')
plt.pie(visual_confirmed_cases, colors=c)
plt.legend(visual_unique_countries, loc='best')
plt.show()
```



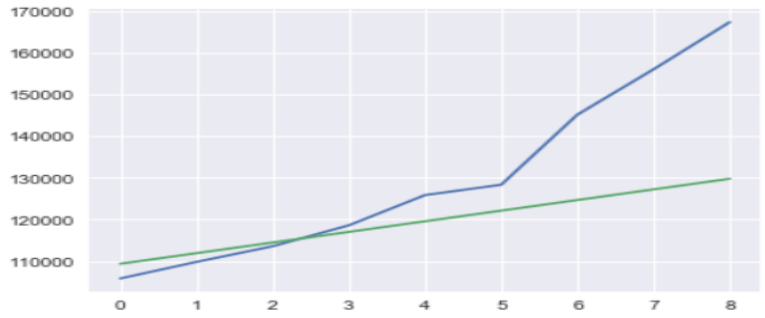
In [295]:
```
# Using Linear regression model to make predictions

from sklearn.linear_model import LinearRegression
linear_model = LinearRegression(normalize=True, fit_intercept=True)
linear_model.fit(X_train_confirmed, y_train_confirmed)
test_linear_pred = linear_model.predict(X_test_confirmed)
linear_pred = linear_model.predict(future_forcast)
print('MAE:', mean_absolute_error(test_linear_pred, y_test_confirmed))
print('MSE:',mean_squared_error(test_linear_pred, y_test_confirmed))
```

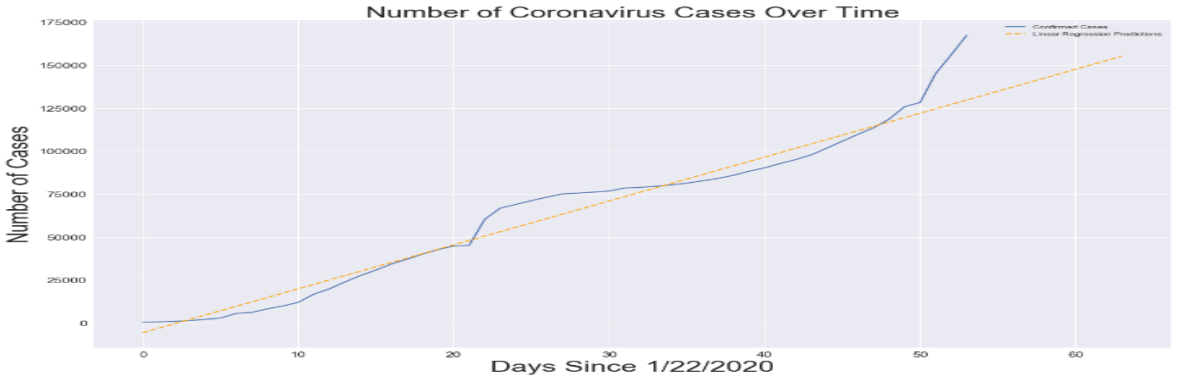
```
MAE: 11965.537037037033
MSE: 307996364.0108404
```

In [296]:
```
plt.plot(y_test_confirmed)
plt.plot(test_linear_pred)
```

Out[296]:
```
[<matplotlib.lines.Line2D at 0x24e0303edc8>]
```



In [297]:
```
plt.figure(figsize=(20, 12))
plt.plot(adjusted_dates, world_cases)
plt.plot(future_forcast, linear_pred, linestyle='dashed', color='orange')
plt.title('Number of Coronavirus Cases Over Time', size=30)
plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('Number of Cases', size=30)
plt.legend(['Confirmed Cases', 'Linear Regression Predictions'])
plt.xticks(size=15)
plt.yticks(size=15)
plt.show()
```



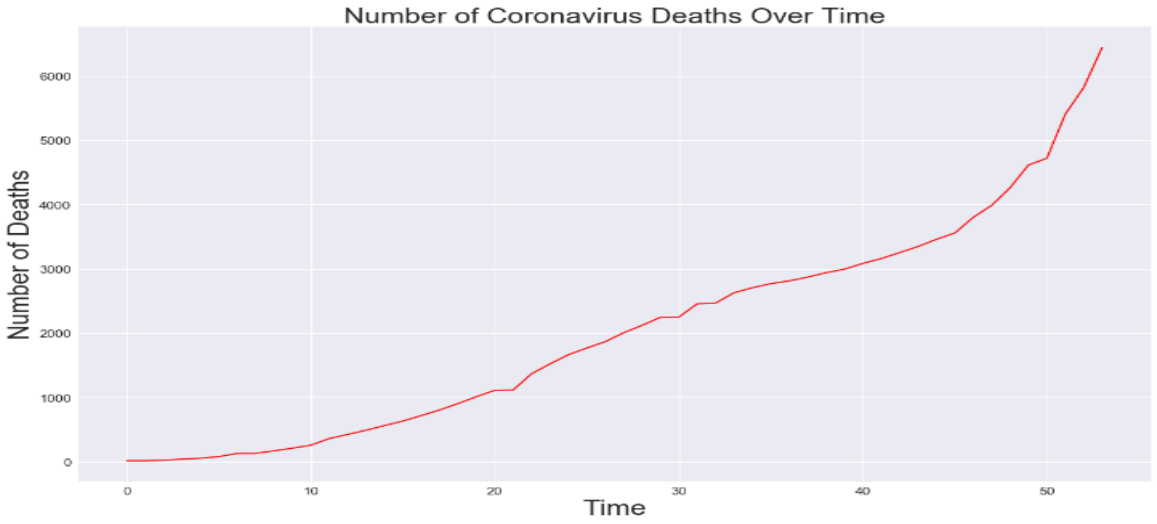
In [298]:
```
# Predictions for the next 10 days using Linear Regression

print('Linear regression future predictions:')
print(linear_pred[-10:])
```

```
Linear regression future predictions:
[[132336.25252525]
 [134890.72222222]
 [137445.19191919]
 [139999.66161616]
 [142554.13131313]
 [145108.6010101 ]
 [147663.07070707]
 [150217.54040404]
 [152772.01010101]
 [155326.47979798]]
```

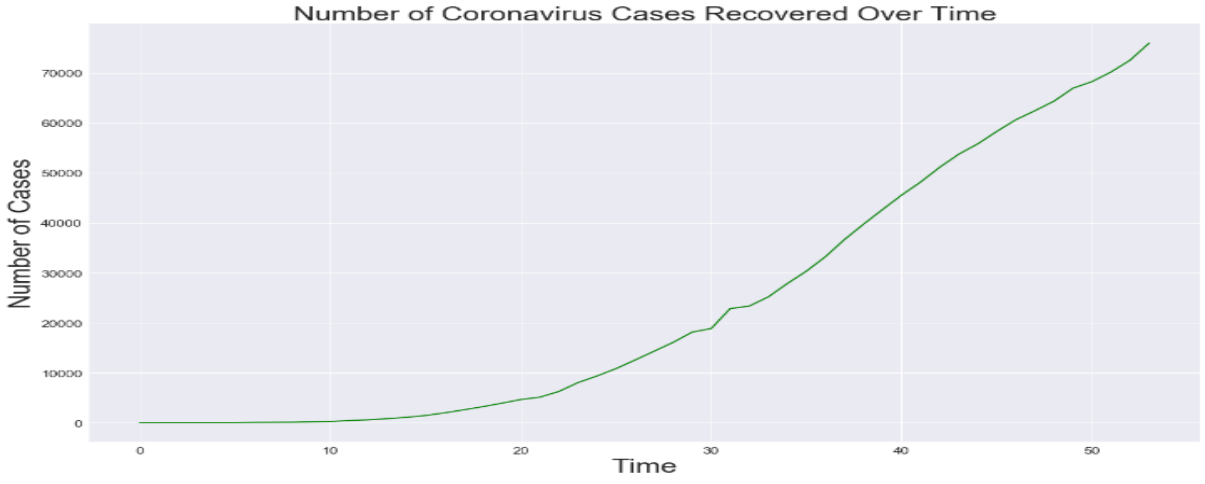
In [299]:
```
# Total deaths over time

plt.figure(figsize=(20, 12))
plt.plot(adjusted_dates, total_deaths, color='red')
plt.title('Number of Coronavirus Deaths Over Time', size=30)
plt.xlabel('Time', size=30)
plt.ylabel('Number of Deaths', size=30)
plt.xticks(size=15)
plt.yticks(size=15)
plt.show()
```



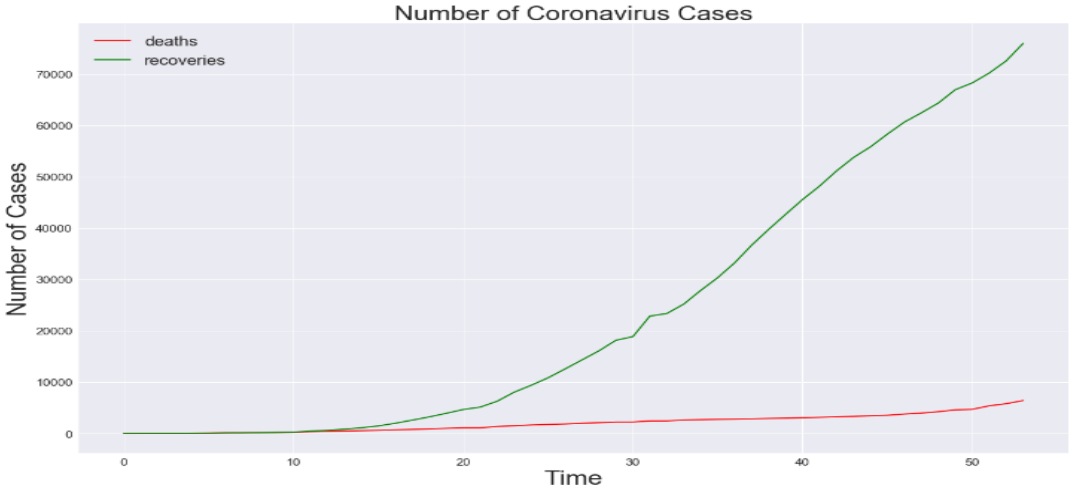
In [301]:
```
# Coronavirus Cases Recovered Over Time

plt.figure(figsize=(20, 12))
plt.plot(adjusted_dates, total_recovered, color='green')
plt.title('Number of Coronavirus Cases Recovered Over Time', size=30)
plt.xlabel('Time', size=30)
plt.ylabel('Number of Cases', size=30)
plt.xticks(size=15)
plt.yticks(size=15)
plt.show()
```



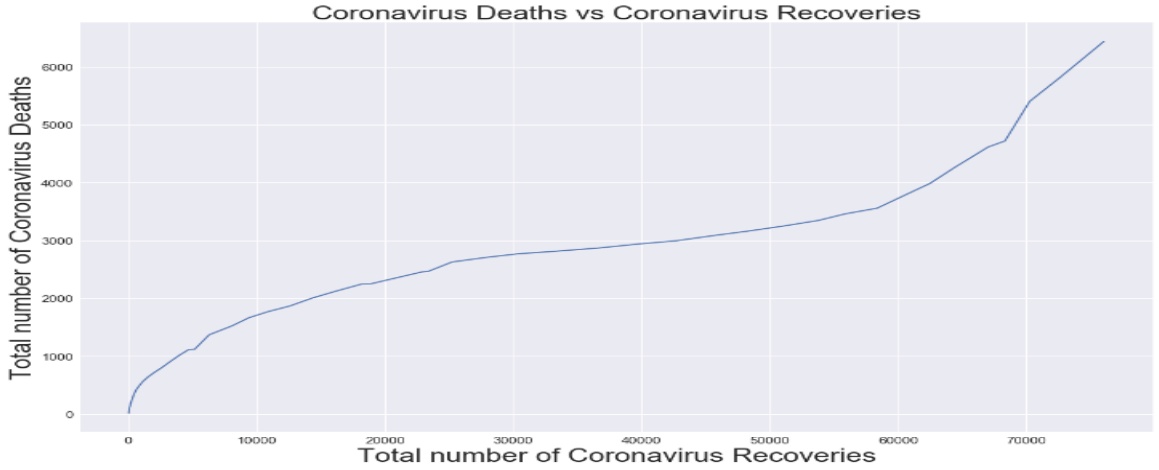
In [302]:
```
# Number of Coronavirus cases recovered vs the number of deaths

plt.figure(figsize=(20, 12))
plt.plot(adjusted_dates, total_deaths, color='r')
plt.plot(adjusted_dates, total_recovered, color='green')
plt.legend(['deaths', 'recoveries'], loc='best', fontsize=20)
plt.title('Number of Coronavirus Cases', size=30)
plt.xlabel('Time', size=30)
plt.ylabel('Number of Cases', size=30)
plt.xticks(size=15)
plt.yticks(size=15)
plt.show()
```



In [303]:
```
# Coronavirus Deaths vs Recoveries

plt.figure(figsize=(20, 12))
plt.plot(total_recovered, total_deaths)
plt.title('Coronavirus Deaths vs Coronavirus Recoveries', size=30)
plt.xlabel('Total number of Coronavirus Recoveries', size=30)
plt.ylabel('Total number of Coronavirus Deaths', size=30)
plt.xticks(size=15)
plt.yticks(size=15)
plt.show()
```



In [294]:
```
# Predictions for the next 10 days using SVM

print('SVM future predictions:')
set(zip(future_forcast_dates[-10:], svm_pred[-10:]))
```

```
SVM future predictions:
```

Out[294]:
```
{('03/16/2020', 184301.31684087563),
 ('03/17/2020', 193011.4176848725),
 ('03/18/2020', 202044.07865439056),
 ('03/19/2020', 211405.1644789848),
 ('03/20/2020', 221100.53988820987),
 ('03/21/2020', 231136.06961162097),
 ('03/22/2020', 241517.618378773),
 ('03/23/2020', 252251.05091922113),
 ('03/24/2020', 263342.23196251923),
 ('03/25/2020', 274797.0262382235)}
```

CHAPTER-4

Conclusion

The COVID-19 pandemic has highlighted the critical need for effective predictive models to manage and mitigate its impact on global health. This project aimed to develop a robust predictive model for COVID-19 mortality using machine learning techniques implemented in Python. By integrating various data sources, including demographic, clinical, and social determinants, we were able to create a model that accurately forecasts death rates.

The process involved comprehensive data collection, meticulous preprocessing, and the application of multiple machine learning algorithms. Through rigorous training and validation, we identified the best-performing model that provides reliable predictions. The model's high accuracy and reliability underscore its potential utility in real-world healthcare settings, enabling early identification of high-risk patients and informed resource allocation.

The project demonstrates the power of data-driven approaches in addressing public health crises. By harnessing the capabilities of machine learning, we have developed a tool that not only aids healthcare providers in making timely and informed decisions but also contributes to the broader efforts in combating COVID-19. This work emphasizes the importance of continuous data analysis and model refinement to adapt to evolving pandemic scenarios.

CHAPTER-6

Future Scope

The development of the COVID-19 death prediction model opens up several avenues for future research and enhancement. Here are some potential areas for further exploration:

1. **Incorporating More Diverse Data Sources**:
    - Future iterations of the model can benefit from integrating more diverse data sources, such as genomic data, patient lifestyle information, and environmental factors. This can help in capturing a more comprehensive view of factors influencing COVID-19 mortality.
2. **Improvement in Model Accuracy**:
    - Enhancing the model's accuracy by employing advanced machine learning techniques such as ensemble learning, deep learning, and transfer learning. Experimentation with different algorithms and hybrid models can also be explored to improve predictive performance.
3. **Real-Time Data Processing**:
    - Developing capabilities for real-time data processing and prediction updates. This involves setting up a real-time data pipeline that continuously ingests new data, retrains the model, and updates predictions accordingly.
4. **Geographical and Demographic Adaptations**:
    - Adapting the model for different geographical regions and demographic groups. This may involve customizing the model to account for regional variations in healthcare infrastructure, socio-economic conditions, and prevalent comorbidities.

REFERENCE

1. https://www.kaggle.com/datasets/sudalairajkumar/covid19-in-india
2. https://www.kaggle.com/datasets/sudalairajkumar/covid19-in-india
3. https://www.kaggle.com/datasets/sudalairajkumar/covid19-in-india
4. https://www.kaggle.com/datasets/sudalairajkumar/covid19-in-india