National College of Ireland

Configuration Manual

MSc Research Project
MSc in Data Analytics

Jose Geo Vattolly
Student ID: x22139508

School of Computing
National College of Ireland

Supervisor: Taimur Hafeez

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……. ……Jose Geo Vattolly……..……

Student ID: ……..…..x22139508……..……

Programme: …… MSc..Data Analytics ……**Year:** ……2023……..

Module: ……..…… MSc Research Project ……..……

Supervisor: …… Taimur Hafeez ……..……

Submission Due Date: ……14/12/2023……..……

Project Title: .. Effects of Carbon dioxide(CO2) Emission on people Death and Global warming…

Word Count: ……1900….…… **Page Count:** ……14……..……..……

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……Jose Geo Vattolly……

Date: ……14/12/2023……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	☒
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☒
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☒

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Jose Geo Vattolly
Student ID: x22139508

1 Introduction

The research project is on “Effects of Carbon Dioxide (CO2) Emissions on People's Death and global warming”. Global warming is contributed to by the carbon dioxide levels in the atmosphere, which trap heat and contribute to rising temperatures and climate change. The effects of this warming, such as worsening heat illnesses and extreme weather events with a global impact, are threatening human health. To predict the future, of death rate due to pollution risk, increase in temperature anomalies every month and the emission of carbon dioxide(CO2) models like LSTM, ARIMA, Random Forest, XGBoost, Linear Regression and ANN were employed. This handbook deals with the major configurations that have been completed as a result of this forecasting effort. All information about the system settings and a few of the programs used for this study is contained in it. In the following section, you can see and discuss the program code.

2 System Requirement

The project was completed and carried out on a laptop that fulfilled the following specifications:

Item	Value
OS Name	Microsoft Windows 11 Home Single Language
Version	10.0.22621 Build 22621
Other OS Description	Not Available
OS Manufacturer	Microsoft Corporation
System Name	LAPTOP-ELBPGCN0
System Manufacturer	ASUSTeK COMPUTER INC.
System Model	ASUS TUF Gaming F15 FX506LI_FX506LI
System Type	x64-based PC
System SKU	
Processor	Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz, 2496 Mhz, 4 Core(s), 8 Logica...
BIOS Version/Date	American Megatrends Inc. FX506LI.310, 26-11-2021
SMBIOS Version	3.2
Embedded Controller Version	3.05
BIOS Mode	UEFI
BaseBoard Manufacturer	ASUSTeK COMPUTER INC.
BaseBoard Product	FX506LI
BaseBoard Version	1.0

Figure 1: System Configuration

3 Software's Required:

The fundamental programming language for running the models and retrieving their results is Python. The well-known Anaconda utility has provided Python. The creation of the model and visualization of this graph was done using Excel and Lucidchart.

```
In [1]: from platform import python_version

        print(python_version())

        3.9.13
```

Figure 2: Python Version

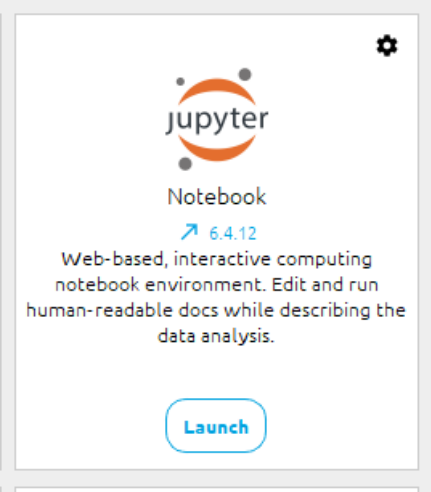

Figure 3: Jupiter Version

4 Data preparation and Feature selection

4.1 Importing the Dataset

The Carbon dioxide data was collected from the ourworldindata.org which contains the record of CO2 emissions of all continents and countries from the year 1949 to 2021. The death rate record was collected from Kaggle.com. Which has all the continents and countries records from 1990 to 2019. The global anomalies data was collected from data.world that has the temperature anomalies data from 1880 to 2020. The below figure shows the loading of three data in the Python code.

```
global_temp_anomalies_df = pd.read_csv('C:/Users/geojo/Downloads/Global Temperature Anomalies.csv')
```

Figure 4: Loading of temperature anomalies data

```
In [3]: co2_data = pd.read_csv('C:/Users/geojo/Downloads/annual-co2-emissions-per-country.csv')
```

Figure 5: Loading of CO2 emission data

```
In [2]: lives_affected_df = pd.read_csv('C:/Users/geojo/Downloads/lives-affected-due-to-airpollution.csv')
```

Figure 6: Loading of death rate data

4.2 Required Libraries

```
: import pandas as pd
  from sklearn.model_selection import train_test_split
  from sklearn.linear_model import LinearRegression
  from sklearn.preprocessing import OneHotEncoder
  from sklearn.compose import ColumnTransformer
  from sklearn.pipeline import Pipeline
  from sklearn.metrics import mean_squared_error
  import matplotlib.pyplot as plt
  import numpy as np
  from sklearn.model_selection import train_test_split
  from sklearn.preprocessing import StandardScaler
  from keras.models import Sequential
  from keras.layers import Dense
  from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
  from xgboost import XGBRegressor
  from sklearn.ensemble import RandomForestRegressor
```

Figure 6: Libraries required to implement the models

The above libraries are important in implementing the models such as Random Forest, XGBoost, Linear Regression and ANN(Artifical nueral networks) to predict the death rate due to pollution and the mean_squared_error and mean_absolute_error are helpful in finding the accuracy of all the models including LSTM and ARIMA.

4.3 Data Preprocessing and Transforming the Data

Preprocessing for the prediction of Death rate:

The Carbon dioxide(CO2) and the death rate data were combined for the prediction of the death rate due to pollution.

|:

	Entity	Country Code	Year of Observation	Death Count
0	Afghanistan	AFG	1990	37231
1	Afghanistan	AFG	1991	38315
2	Afghanistan	AFG	1992	41172
3	Afghanistan	AFG	1993	44488
4	Afghanistan	AFG	1994	46634
...	...	...	...	...
6835	Zimbabwe	ZWE	2015	13246
6836	Zimbabwe	ZWE	2016	13131
6837	Zimbabwe	ZWE	2017	12926
6838	Zimbabwe	ZWE	2018	12745
6839	Zimbabwe	ZWE	2019	12667

6840 rows × 4 columns

Figure 6: Death count due to pollution risk

```
In [11]: co2_emissions_df.head()
```

Out[11]:

	Entity	Code	Year	Annual CO_2 emissions
0	Afghanistan	AFG	1949	14656.0
1	Afghanistan	AFG	1950	84272.0
2	Afghanistan	AFG	1951	91600.0
3	Afghanistan	AFG	1952	91600.0
4	Afghanistan	AFG	1953	106256.0

Figure 7: CO2 emission data

```
In [13]: merged_df.head()
```

Out[13]:

	Entity	Code_x	Year	Lives Affected	Code_y	Annual CO_2 emissions
0	Afghanistan	AFG	1990	37231	AFG	2024326.1
1	Afghanistan	AFG	1991	38315	AFG	1914301.0
2	Afghanistan	AFG	1992	41172	AFG	1482054.0
3	Afghanistan	AFG	1993	44488	AFG	1486943.0
4	Afghanistan	AFG	1994	46634	AFG	1453829.0

Figure 8: Combined data of CO2 and death rate

```
In [14]: lives_affected_pollution_df['Entity'].value_counts()

Out[14]: Afghanistan          30
         Northern Ireland     30
         Norway               30
         OECD Countries       30
         Oman                 30
                              ..
         Guam                 30
         Guatemala            30
         Guinea               30
         Guinea-Bissau        30
         Zimbabwe             30
         Name: Entity, Length: 228, dtype: int64
```

Figure 9: Each Country had 30 records when combined

Preprocessing for the prediction of CO2 emission:

From the CO2 data only the continents were filtered from the dataset and on each continent LSTM and ARIMA model was applied.

```
In [4]: continents = ['Asia', 'Africa', 'North America', 'South America', 'Antarctica', 'Europe', 'Australia']
        co2_data = co2_data[co2_data['Entity'].isin(continents)]
```

Figure 10: Separating the continent's record from the dataset

```
Missing Values:
Entity                      0
Code                     1360
Year                        0
Annual CO₂ emissions        0
dtype: int64
```

Figure 11: The null values were checked in the dataset

Preprocessing for the prediction of temperature anomalies:

For the prediction of temperature anomalies, all the month's data was set in a single row, and the year was converted to date and time format, and the seasonality, trends, and residuals were checked in the dataset.

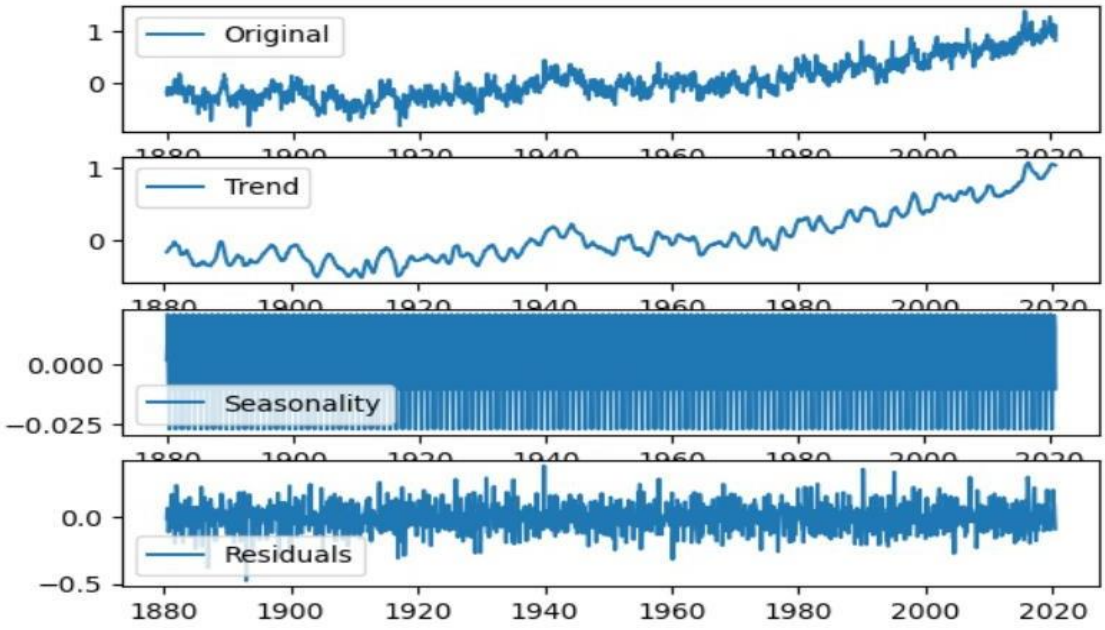

Figure 11: Seasonality, trend, and Residuals check in temperature anomalies data

4.3 Feature Selection and Splitting the Data

Splitting of Death rate data:

Splitting of death rate data into training and testing, 80 percent of data was taken as training and 20 percent for testing.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Figure 12: Splitting of Death rate data

Splitting of temperature anomalies data:

Splitting of temperature anomalies data into training and testing.

```
In [11]: def split_data(data):
             train_data = data.loc[data.index <= '2019-12-31']
             test_data = data.loc[(data.index >= '2020-01-01') & (data.index <= '2020-12-31')]
             return train_data, test_data
```

Figure 13: Splitting of temperature anomalies data

Splitting of CO2 emission data:

Splitting of CO2 emission data into training and testing for LSTM, 80 percent of data was taken as training and 20 percent for testing.

```
train_data, test_data = train_test_split(co2_data, test_size=0.2, shuffle=False)
```

Figure 14: Splitting of CO2 emission data for LSTM

Splitting of CO2 emission data into training and testing for ARIMA, 80 percent of data was taken as training and 20 percent for testing.

```
train_data, test_data = data[:train_size], data[train_size:]
```

Figure 15: Splitting of CO2 emission data for ARIMA

5 Implementation and evaluation of data:

After that, the ARIMA and auto ARIMA models were trained using the training set of temperature anomalies. The best fit for (p,d,q) achieved using auto ARIMA is (2,1,1).

```
Performing stepwise search to minimize aic
 ARIMA(2,1,2)(1,0,1)[12] intercept   : AIC=-110.683, Time=0.64 sec
 ARIMA(0,1,0)(0,0,0)[12] intercept   : AIC=-95.577, Time=0.03 sec
 ARIMA(1,1,0)(1,0,0)[12] intercept   : AIC=-108.426, Time=0.12 sec
 ARIMA(0,1,1)(0,0,1)[12] intercept   : AIC=-111.922, Time=0.15 sec
 ARIMA(0,1,0)(0,0,0)[12]             : AIC=-97.333, Time=0.02 sec
 ARIMA(0,1,1)(0,0,0)[12] intercept   : AIC=-113.409, Time=0.09 sec
 ARIMA(0,1,1)(1,0,0)[12] intercept   : AIC=-111.839, Time=0.12 sec
 ARIMA(0,1,1)(1,0,1)[12] intercept   : AIC=-110.116, Time=0.27 sec
 ARIMA(1,1,1)(0,0,0)[12] intercept   : AIC=-114.915, Time=0.11 sec
 ARIMA(1,1,1)(1,0,0)[12] intercept   : AIC=-113.096, Time=0.21 sec
 ARIMA(1,1,1)(0,0,1)[12] intercept   : AIC=-113.139, Time=0.18 sec
 ARIMA(1,1,1)(1,0,1)[12] intercept   : AIC=-111.335, Time=0.47 sec
 ARIMA(1,1,0)(0,0,0)[12] intercept   : AIC=-109.975, Time=0.06 sec
 ARIMA(2,1,1)(0,0,0)[12] intercept   : AIC=-115.796, Time=0.20 sec
 ARIMA(2,1,1)(1,0,0)[12] intercept   : AIC=-113.995, Time=0.43 sec
 ARIMA(2,1,1)(0,0,1)[12] intercept   : AIC=-114.036, Time=0.34 sec
 ARIMA(2,1,1)(1,0,1)[12] intercept   : AIC=-112.186, Time=0.60 sec
 ARIMA(2,1,0)(0,0,0)[12] intercept   : AIC=-109.037, Time=0.13 sec
 ARIMA(3,1,1)(0,0,0)[12] intercept   : AIC=-114.080, Time=0.23 sec
 ARIMA(2,1,2)(0,0,0)[12] intercept   : AIC=-114.248, Time=0.24 sec
 ARIMA(1,1,2)(0,0,0)[12] intercept   : AIC=-115.414, Time=0.18 sec
 ARIMA(3,1,0)(0,0,0)[12] intercept   : AIC=-112.034, Time=0.06 sec
 ARIMA(3,1,2)(0,0,0)[12] intercept   : AIC=-111.894, Time=0.20 sec
 ARIMA(2,1,1)(0,0,0)[12]             : AIC=-115.182, Time=0.09 sec

Best model:  ARIMA(2,1,1)(0,0,0)[12] intercept
Total fit time: 5.188 seconds
```

Figure 16: Using auto ARIMA, the best fit for temperature anomalies data was found.

The training set data for pollution-related death rate prediction was trained using Random Forest, Linear Regression, XGBoost, and ANN (Artificial neural network).

```
5]: models = {
        'Linear Regression': LinearRegression(),
        'Random Forest': RandomForestRegressor(random_state=42),
        'XGboost': XGBRegressor()
    }
```

Figure 17: The model function were set in an array

```
model = Pipeline([
    ('preprocessor', preprocessor),
    ('regressor', models[model_name])
])
```

Figure 18: A pipeline was attached to the array.

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

Figure 19: Every model was trained with the training set

```
model = Sequential()
model.add(Dense(units=64, activation='relu', input_dim=X_train.shape[1]))
model.add(Dense(units=32, activation='relu'))
model.add(Dense(units=1))
model.compile(optimizer='adam', loss='mean_squared_error')

model.fit(X_train, y_train, epochs=50, batch_size=32, validation_split=0.2)
```

Figure 20: ANN (Artificial neural network) was used to train the training set data.

The training set data for CO2 emission prediction was trained using LSTM and ARIMA for all continents.

```
model = Sequential()
model.add(LSTM(50, input_shape=(seq_length, 1)))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')
model.fit(X_train, y_train, epochs=100, batch_size=64)
```

Figure 21: All of the continent's CO2 emission training set data were trained using the LSTM in a function code

```
model = auto_arima(train_data, seasonal=True, m=12, stepwise=True, trace=True)
model.fit(train_data)
```

Figure 22: All of the continent's CO2 emission training set data were trained using the auto ARIMA in a function code

```
In [31]: auto_arima_predictions.index = test_data.index
         plot_actual_vs_predicted(test_data, auto_arima_predictions, title='Auto ARIMA Model')
```

Figure 23: Call this function to plot the anticipated graph and predict temperature anomalies.

```
forecast, conf_int = model.predict(n_periods=len(test_data), return_conf_int=True)
actual_data = test_data.values
```

Figure 24: This line of ARIMA code forecasted the CO2 emissions of all continents.

```
y_pred = model.predict(X_test)
y_pred = scaler.inverse_transform(y_pred)
```

Figure 25: This LSTM code line projected the CO2 emissions of all continents.

```
y_pred = model.predict(X_test)
```

Figure 26: The above code was used to predict the Random Forest, XGBoost, and Linear Regression in a function for predicting pollution-related death rates.

6 Results

Comparison of ARIMA and LSTM for CO2 emission prediction:

For all continents, the ARIMA model was applied in four ways: one was by using only ARIMA by taking AutoRegressive, Integration, and Moving Average as(p,d,q) (1,1,1), the second was by using autoregression technique, which helps in finding the best combination of (p,d,q), the third was by using Square root transformation and auto ARIMA, and the fourth was by using Log transformation and auto ARIMA.

The results obtained in Africa using these four methodologies is given below:

Algorithms	Mean Squared error	Mean Absolute error	Root Mean Squared error

ARIMA for Africa	290531760.331297	1.2051250316340288e+17	347149107.9686118
Auto ARIMA for Africa	339259433.9861032	1.628588964059672e+17	403557798.09832346
Auto ARIMA with Log transformation for Africa	5435814095.230929	8.273068665910785e+19	9095641080.160751
Auto ARIMA with square transformation for Africa	211195072.87397638	5.634386201854745e+16	237368620.54312792

Here, we can see that auto ARIMA with square transformation produces superior results than previous methods of implementing ARIMA, with mean squared error 211195072.87397638, mean absolute error 5.634386201854745e+16, and root mean squared error 237368620.54312792.

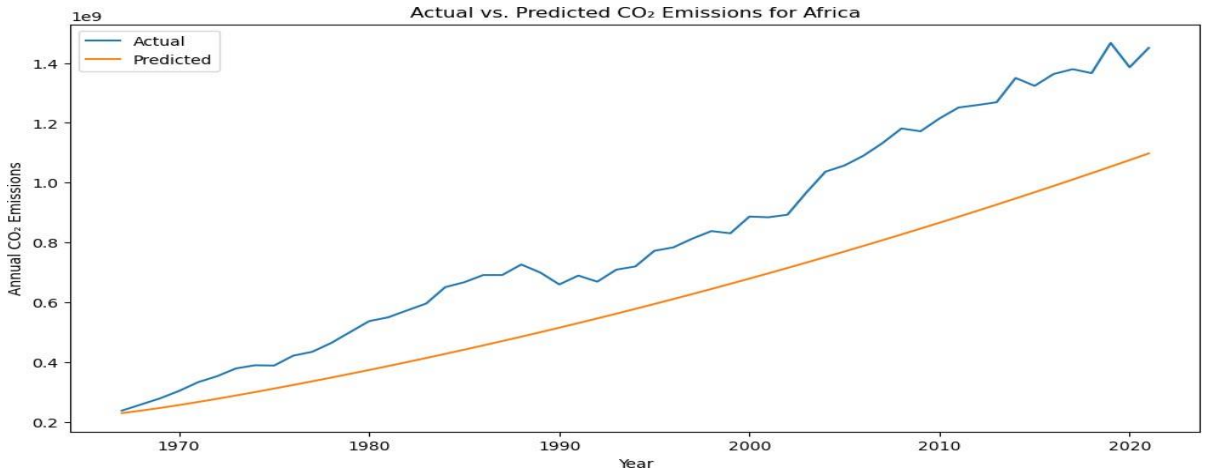

Figure 27: This graph compares anticipated CO2 emission data to real data for Africa using auto ARIMA with square transformation.References

When comparing findings in North America, it appears to be the same as in Africa, where the results were better when ARIMA employed square transformation and auto ARIMA.

Algorithms	**Mean Squared error**	**Mean Absolute error**	**Root Mean Squared error**
ARIMA for North America	2160261185.587062	5.33167749227389e+18	2309042548.8227563
Auto ARIMA for North America	651150994.9901432	5.2449058358417197e+17	724217221.270091
Auto ARIMA with Log transformation for North America	5435814095.230929	8.273068665910785e+19	9095641080.160751
Auto ARIMA with square transformation for North America	662676064.6131425	1.016603949809434e+18	1008267796.6737974

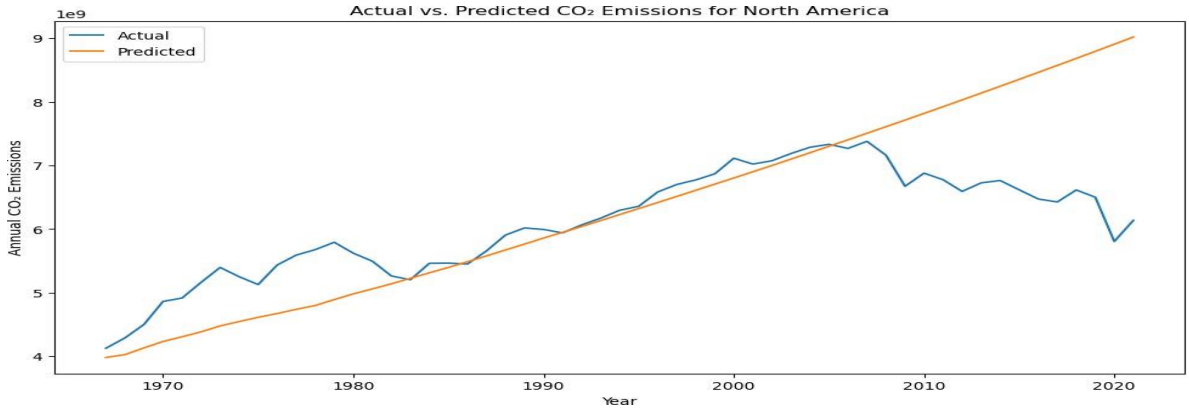

Figure 28: This graph compares anticipated CO2 emission data to actual data for North America using auto ARIMA with square transformation.

Similarly, when auto ARIMA and square transformation were used, all of the continent's CO2 emission projections performed better. Root Mean Squared errors across South America, Europe, Australia, and Asia were 84789367.91980399, 5681130173.117291, 38456109.359867394, and 6173500177.719932.

When compared to ARIMA, LSTM produced better results. The accuracy was enhanced when log transformation was applied to it in LSTM. The first technique was to apply LSTM without any transformation and put a single layer with 100 epochs, and the second method was to apply log transformation with two layers and 70 epochs. The output of LSTM with log transformation is shown below.

Algorithm	**Mean Squared error**	**Mean Absolute error**	**R-squared value**
LSTM for Asia	2.3564833517148513	0.6000155924906578	0.9746838822849571
LSTM for Europe	0.01053538793285718	0.08264950142925974	0.9978587510632275
LSTM for North America	0.6503821173554817	0.4458126905330167	0.9874727276998669
LSTM for South America	0.6075929142721977	0.2716939328707382	0.9928710565032028
LSTM for Africa	0.5921187680114269	0.2607452187040052	0.9930747222830251
LSTM for Australia	0.954941086802987	0.2783716062941217	0.9870752854723714

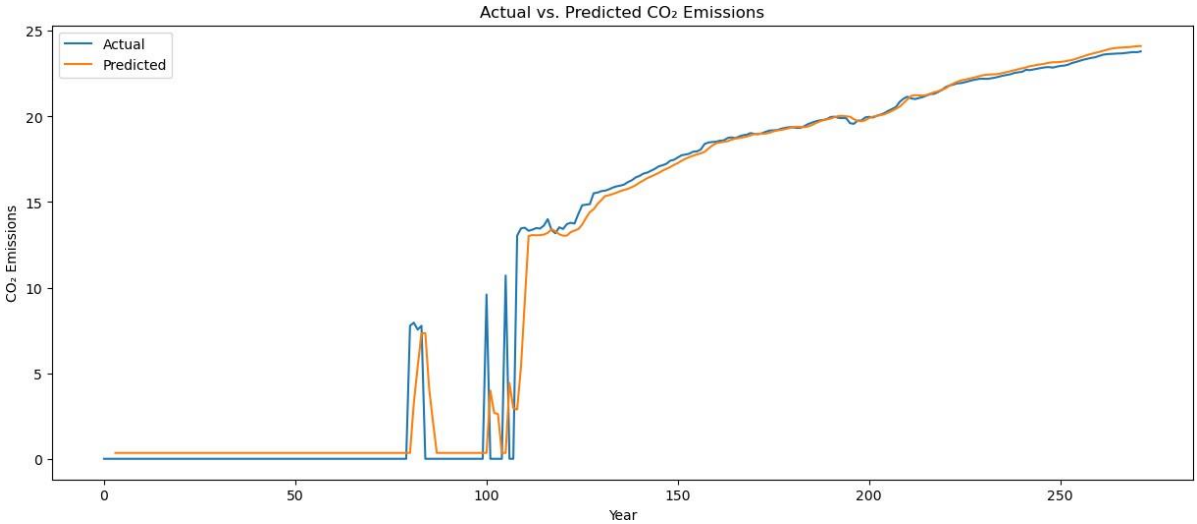

Figure 29: This graph compares predicted CO2 emission data to real data for Asia using LSTM with Log transformation.

The rest of the continents' graphs were similar to Asia's near the real CO2 emission line. When ARIMA and LSTM are compared, the LSTM model produces more accurate findings than the ARIMA model.

Comparison of Random Forest, XGBoost, Linear Regression, and ANN for the prediction of the death rate due to carbon dioxide:

When it comes to predicting death rates, Random Forest and XGBoost appear to outperform the other two strategies.

Algorithms	**Mean Squared error**	**Mean Absolute error**	**Root Mean Squared error**	**R-squared**
Random forest	21037563.003558323	1279.533198156811	4586.672323543325	0.9998861755993983
Linear Regression	28877181370.73155	45615.532187226316	169932.87313151493	0.8437590960495704
XGBoost	54954612.266509205	3149.0177540620844	7413.137815156899	0.9997026663306735
ANN	3.9506031747132114e+18	287576743.4249185	1987612430.7100747	-0.02135105080151889

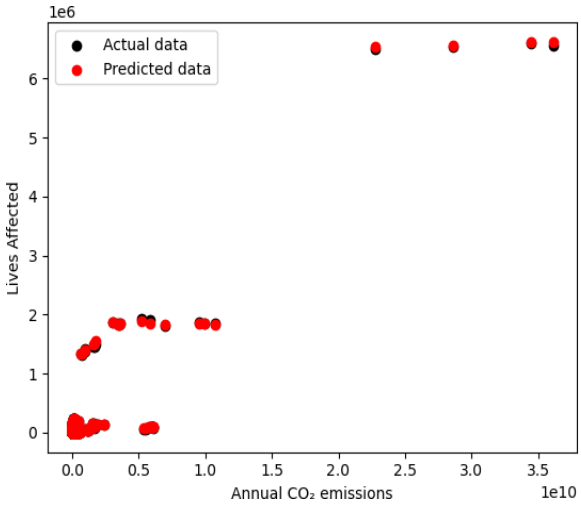

Figure 30: Prediction graph of Random forest

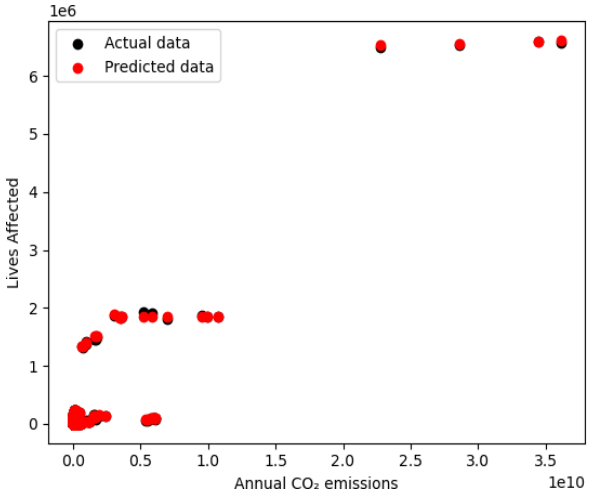

Figure31: Predicted graph of XGBoost

Results obtained when ARIMA was applied to predict the temperature anomalies:

ARIMA was used to predict temperature anomalies in two ways: without auto ARIMA and with auto ARIMA. ARIMA (1,1,1) was used for AutoRegressive, Integration, and Moving Average in ARIMA. When using auto ARIMA, the optimum fit was found to be (2,1,1) for AutoRegressive, Integration, and Moving Average.

Algorithm	Mean Squared error	Mean Absolute error	Root Mean Squared error
ARIMA	0.13033416212658233	0.02503972779973708	0.15823946347146492
Auto ARIMA	0.11554501319511012	0.0194171273999903	0.13934535299029638

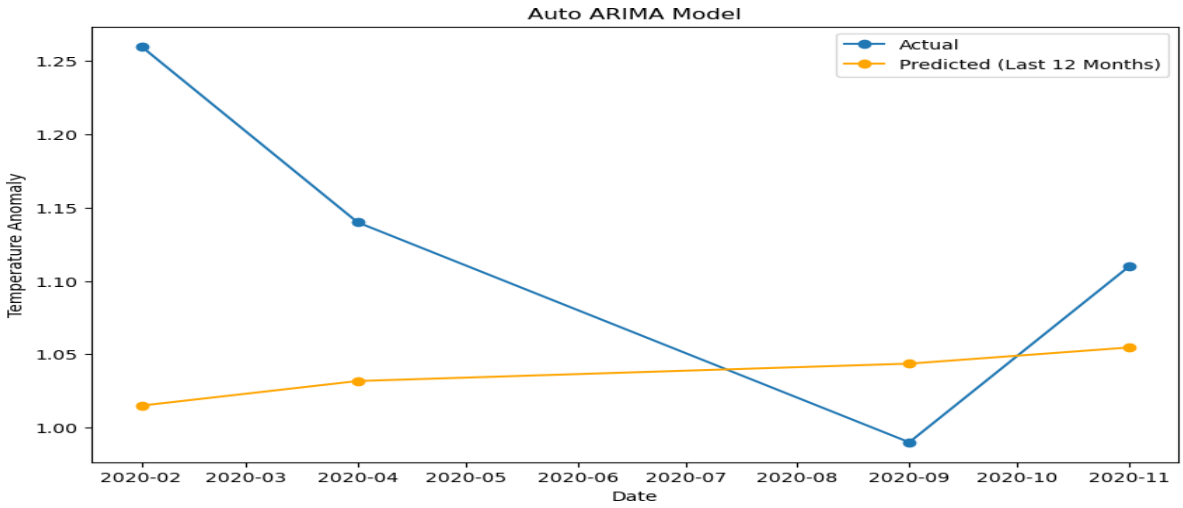

Figure32: The above graph shows the prediction of temperature anomalies using Auto ARIMA

The outcome is better when auto ARIMA is used by identifying the best fit as (2,1,1), as shown in the above table with mean Squared error 0.11554501319511012, mean Absolute error 0.0194171273999903, and root mean Squared error 0.13934535299029638.

Reference:

https://ourworldindata.org/co2-emissions

https://www.kaggle.com/code/abmsayem/impact-of-air-pollution-on-human-health/input

https://data.world/makeovermonday/2021w3