Deaths due to Air Pollution

Data Science With Python Lab Project Report

Bachelor

in

Computer Science

By

Team Name

s190812 : Talla Sandeep

s190771 : Nelakuri Surya Teja

Rajiv Gandhi University Of Knowledge And Technologies

S.M. Puram , Srikakulam -532410

Andhra Pradesh, India

Abstract

Air pollution refer to the harmful substances and pollutants in the Earth. Which can shows the bad impact on humans life. The pollutants may be harmful gases or others.

The pollution can be released to the atmosphere by human and nuturally. It increase day by day. from industris, burning of fuels and human wastage, agricultural activities. These activities release pollutants (carbon monoxide, sulfur dioxide, nitrogen oxides, etc)

So our project aim is to analyse the impact of air pollution on humans, specifically Deaths due to air pollution. So for this purpose we taken data set from kaggle website. Based on that data we predecting no of deaths in specific year.

Contents

Chapter 1

Introduction

1.1 Introduction to Your Project

In recent years, the effect of air pollution on human health have become very critical. The releasing of pollutants into the atmosphere from different sources, such as industries, vehicle, and burning fuels, has lead to decreasing the air quality. Therefore the project helps to predict the no of deaths due to air pollution in different years.

In this process of predicting, the the project gone through different stages.

Pre-processing

Visualization

Analysis

1.2 Application

Based on our model predictions the project applications are following:

- Predicting deaths due to air pollution in various years.
- Helps to inform public health policies to reduce Air pollution.
- Helps to increase awareness of peoples health.
- Finding which country have more deaths.
- Helps to take charges to increase Environmental Areas.

1.3 Motivation Towards Your Project

Now a days the Air Pollution is high in all countries in the world. For this reason most of the people facing many different health issues. In our country and in the world many peoples dies by affect of polluted air. So my motive for this project to analyse the past data and conclude where the most deaths are occur in worldwide. and how pollution affects different countries.

This project help to environmental professional and experts. Because our model provides the capability to predict deaths due to polluted Air.

Finally, the motivation behind this project is to address the problem number of deaths caused by air pollution, promote fairness, and find solutions to protect people's health.

1.4 Problem Statement

To provide set of techniques to preprocess the data and to analyze the data.

Chapter 2

Approach To Your Project

2.1 Explain About Your Project

Our project mainly focus on studying the deaths caused by air pollution. It aims to understand the impact of air pollution on humans health like deaths. So it helps identify predict deaths in various years. Therefore it helps to reduce the number of deaths related to pollution.

The project include graphical representation to represent deaths by polluted air. This also include machine learning model for predicting deaths.

Overall, project outcome helps to public health policies, environmental regulations for making and planning better decisions.

2.2 Data Set

We are taken dataset from kaggle website. It contains 7 columns and 6468 rows.

Explanation of columns present in Dataset :

Entity : The first column of our dataset is Entity column. It contains the different countries.

Code : The code column is the second one. It contains country codes corresponding to the country in the Entity column.

Air pollution (total) (deaths per 100,000) : It is the another column contains the total deaths.

Indoor air pollution (deaths per 100,000) : It contains deaths due to indoor air pollution.

Outdoor particulate matter (deaths per 100,000) : The column contains deaths due to outdoor pollution.

Outdoor ozone pollution (deaths per 100,000) : Deaths caused by ozone pollution.

2.3 Prediction technique

In this implementation of Project we are using predefined machine learning techniques from the scikit learn library based on the Accuracy measures we pick best Algorithms. The scikit learn library also known as sklearn. In this project we used Linear Regression Algorithm and Random Forest Algorithm.

Linear Regression : Linear Regression is a statistical model commonly used in analysis of relationship between different variables (Dependent and

Independent Variables).

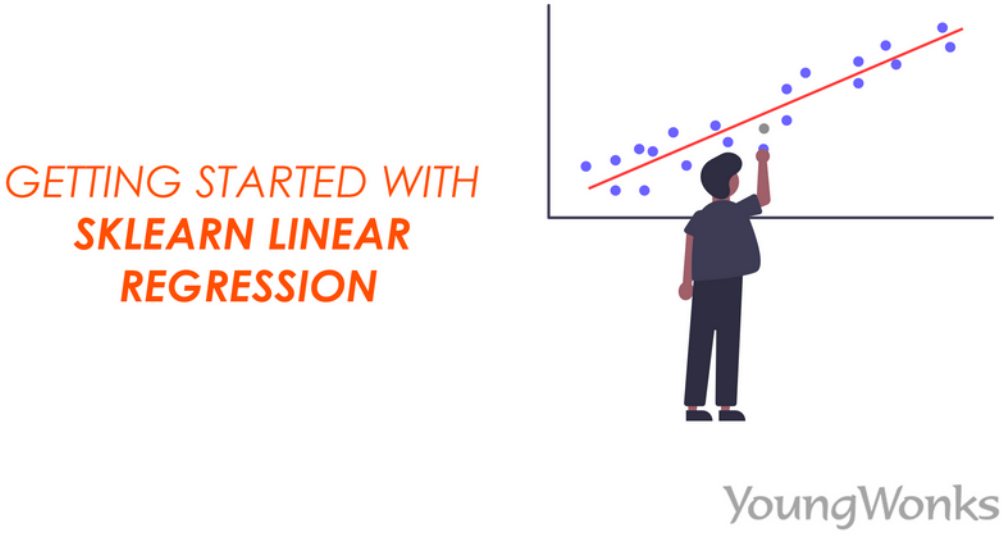

Figure 2.1: Linear Regression

Random Forest : Random Forest is a popular machine learning algorithm used for both classification and regression tasks. It is an ensemble learning method that combines multiple decision trees to make predictions.

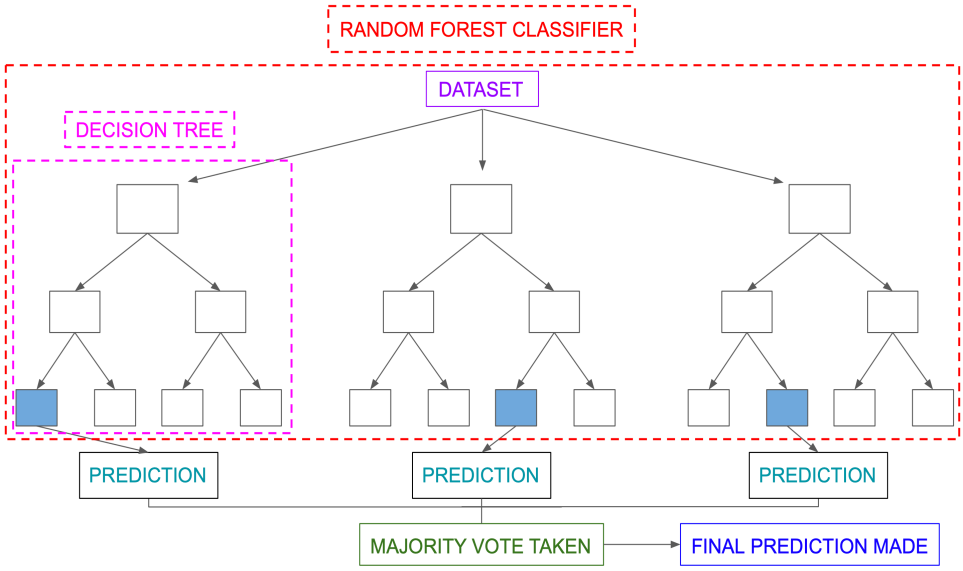

Figure 2.2: Random Forest

2.4 Graphs

2.4.1 Let us plot Bar Graphs 1

The below Bar graph was showing Top 10 most polluted countries on Average (1990-2017)

```
plt.style.use('fivethirtyeight')
#Used fivethirtyeight style
Top_countries = pivot_table.sort_values('Mean',ascending =
↪  False)['Mean'].head(10)
Top_countries.plot.bar(figsize = (20,8), title = 'Top 10 most
↪  polluting countries on average (1990-2017)', ylabel =
↪  "Levels of pollution", cmap = 'twilight_shifted')
plt.show()
```

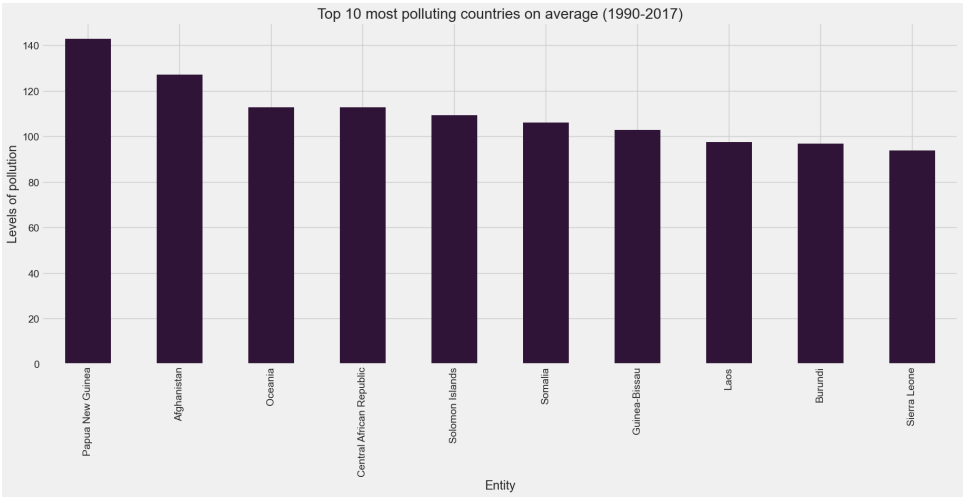

Figure 2.3: Top 10 most polluted countries

The below Bar graph was showing Top 10 least polluting countries on Average (1990-2017)

```
#First created pivot table based on that ploted this graph

plt.style.use('fivethirtyeight')
#Here used fivethirtyeight style
least_countries = pivot_table.sort_values('Mean',ascending =
↪  True)['Mean'].head(10)
least_countries.plot.bar(figsize = (20,8), title = 'Top 10
↪  least polluting countries on average (1990-2017)', ylabel
↪  = "Levels of pollution", cmap = 'twilight_shifted')
plt.show()
```

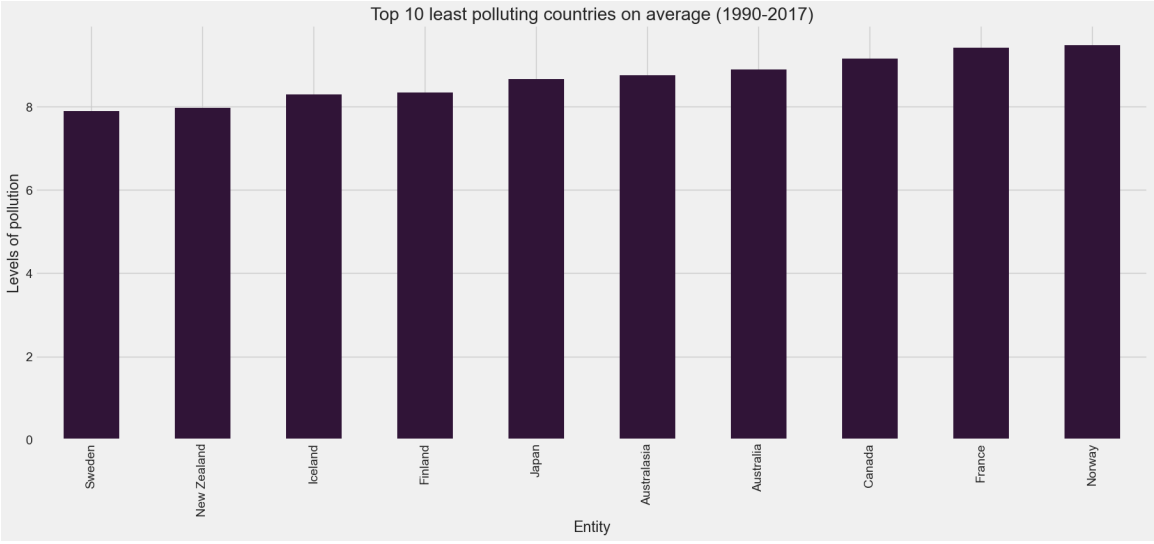

Figure 2.4: Top 10 Least polluted countries

2.4.2 Histographs

Let us plot the histographs on Indoor Air Pollution

```
sns.displot(data['Indoor air pollution (deaths per 100,000)'],
↪  kde=False, bins=10, color = '#0D0F5B', label = 'Indoor Air
↪  Pollution')
plt.style.use('fivethirtyeight')
plt.legend()
plt.show()
```

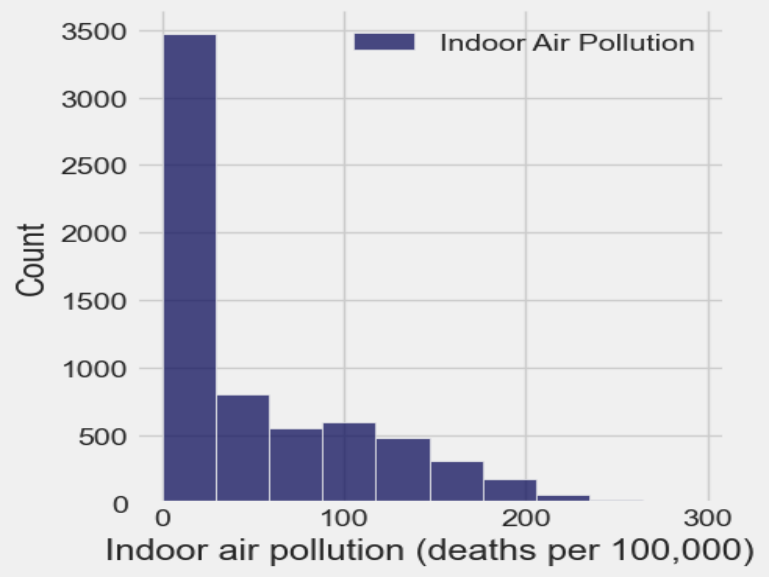

Figure 2.5: Indoor Air Pollution

Let us plot the histographs on Outdoor Air Pollution

```
sns.displot(data['Outdoor particulate matter (deaths per
↪  100,000)'], kde=False, bins=15, color = '#581845', label =
↪  'Outdoor particulate')
```

```
plt.style.use('fivethirtyeight')

plt.legend()

plt.show()
```

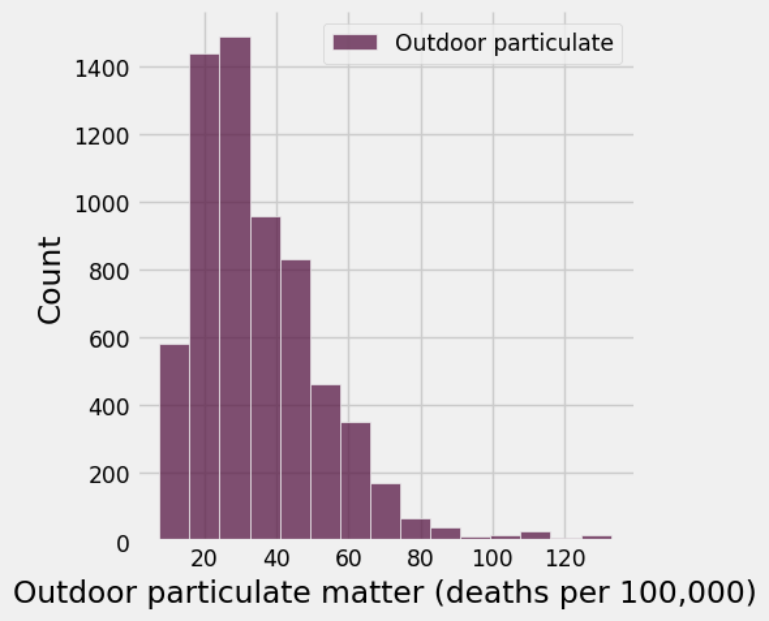

Figure 2.6: Outdoor Air Pollution

2.4.3 Heat Map

Let us also plot Heatmap between every columns using Seaborn library

```
plt.figure(figsize = (8,6))

sns.set_context('paper', font_scale = 1.4)

data_mx = data.corr()

heatmap = sns.heatmap(data_mx, annot = True, cmap = 'Blues',
↪  annot_kws={"color": "black"})

heatmap.set_xticklabels(heatmap.get_xticklabels(),
↪  rotation=45, ha='right', fontname='Arial')
```

```
heatmap.set_yticklabels(heatmap.get_yticklabels(),
 ↪  fontname='Arial')
plt.title('Heat Map on Total Dataset')
plt.show()
```

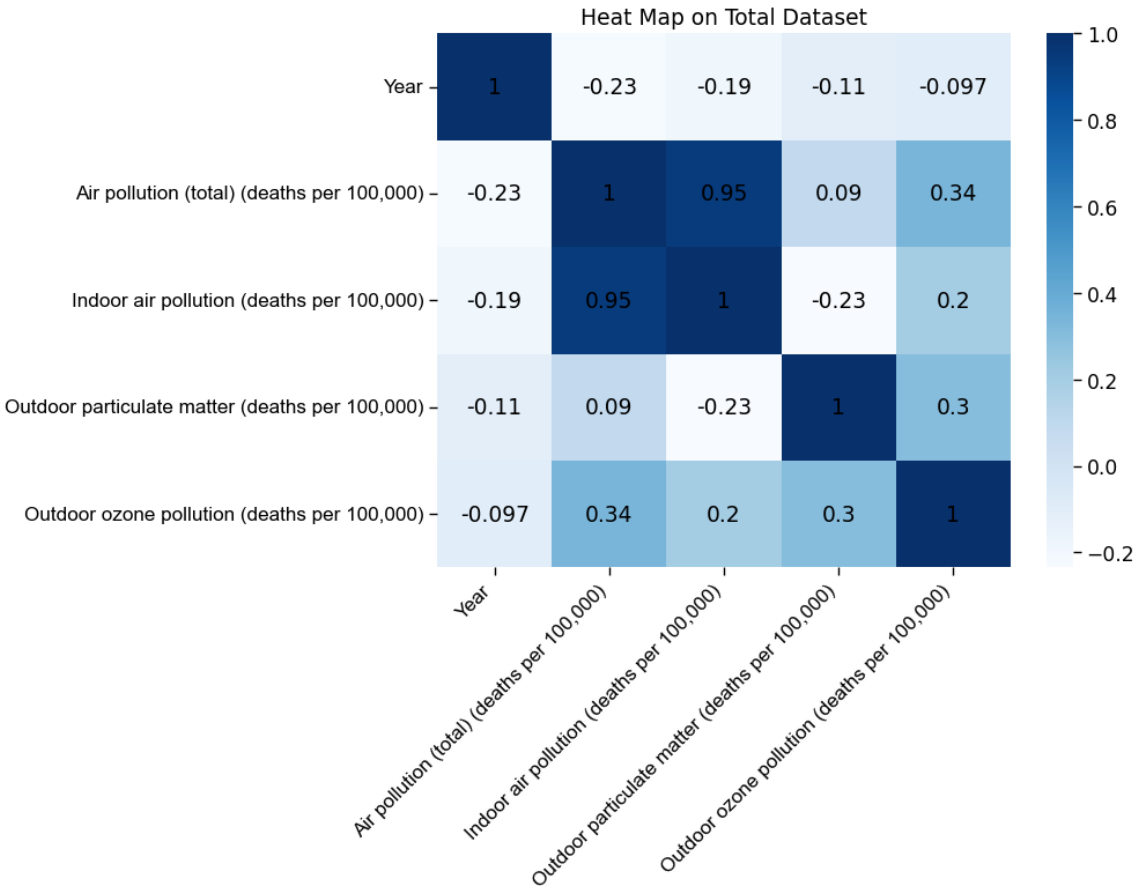

Figure 2.7: Heat Map

2.4.4 Scatter Plot

Now i want plot a Scatter Plot Between India and Pakistan on Indoor Outdoor Deaths

```
india = data[data['Entity'] == 'India']
pakistan = data[data['Entity'] == 'Pakistan']
plt.style.use('seaborn-paper')
random_sizes = np.random.randint(100, 500, size = len(india))
plt.scatter(india['Indoor air pollution (deaths per
↪  100,000)'], india['Outdoor particulate matter (deaths per
↪  100,000)'],
            alpha = 1.0, c = india['Indoor air pollution
             ↪  (deaths per 100,000)'], s = random_sizes, cmap
             ↪  = 'twilight_shifted', label = 'India')

plt.scatter(pakistan['Indoor air pollution (deaths per
↪  100,000)'], pakistan['Outdoor particulate matter (deaths
↪  per 100,000)'],
            alpha = 1.0, c = pakistan['Indoor air pollution
             ↪  (deaths per 100,000)'], s = random_sizes, cmap
             ↪  = 'Greens', label = 'Pakistan')
```

```
plt.xlabel('Indoor Air Pollution Deaths')

plt.ylabel('Outdoor Air Pollution Deaths')

plt.legend(loc = 3)

plt.show()
```

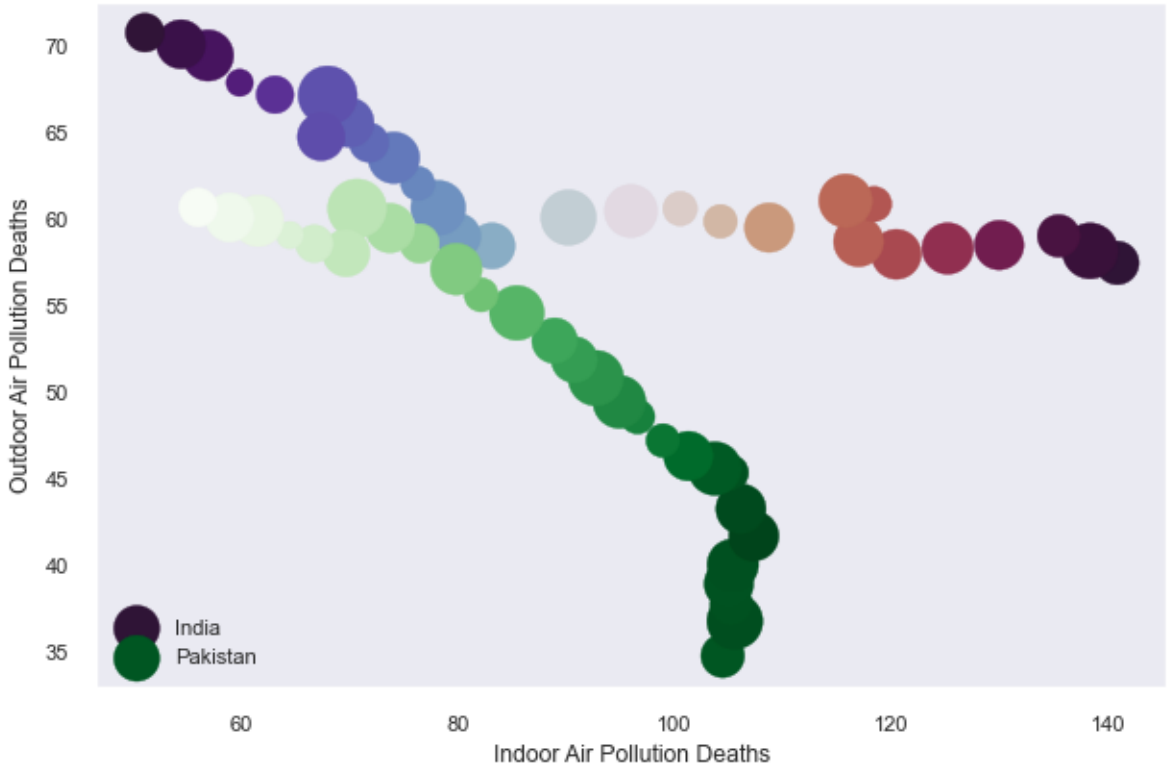

Figure 2.8: Scatter Plot

2.4.5 Joint Plot

```
sns.jointplot(x = 'Air pollution (total) (deaths per
↪  100,000)', y = 'Outdoor ozone pollution (deaths per
↪  100,000)',
        data = data.head(200), color = '#4043A8', kind='reg')
sns.set_style('white')
sns.set_context('paper', font_scale=1.4)
sns.despine(left=False, bottom=False)
plt.show()
```

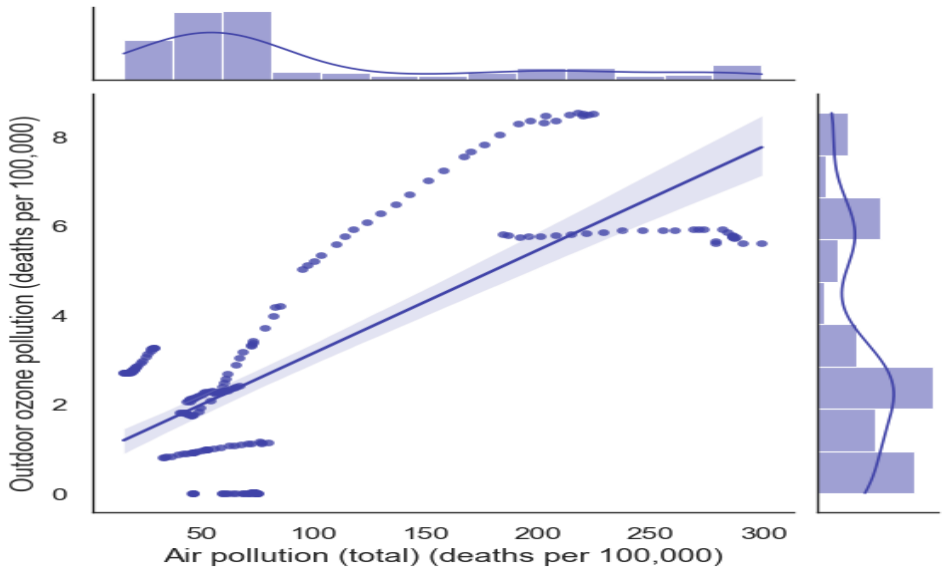

Figure 2.9: Joint Plot

2.5 Visualization

2.5.1 Displot

We decided to plot relation between numeric columns in our dataset by using Displot.

```
plt.figure(figsize = (30,5))
for i,j in enumerate(data.iloc[:,4:-1].columns):
    plt.subplot(1,5,i+1)
    sns.distplot(data[j])
    sns.set(font_scale=0.3)
    plt.tight_layout()
plt.subplots_adjust()
plt.show()
```

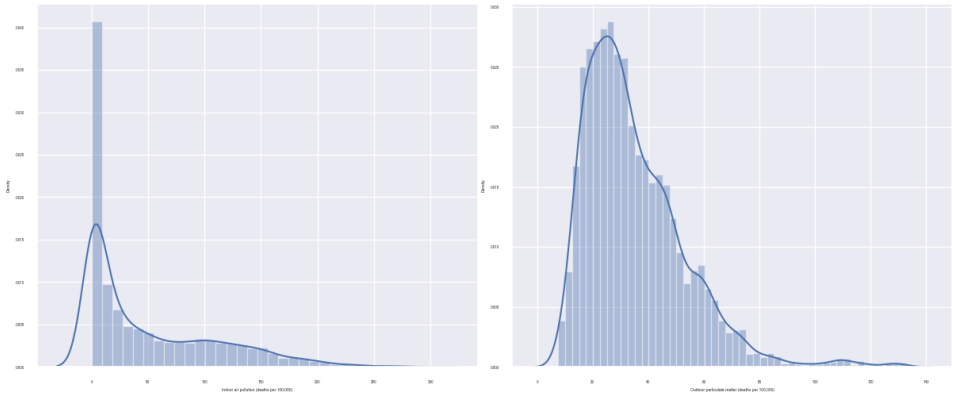

Figure 2.10: Caption

2.5.2 Histo Graph

```
plt.style.use('classic')

data.drop('Year', axis = 1).hist(bins = 20, figsize = (20,
↪  20), color = '#2E4053')

plt.show()
```

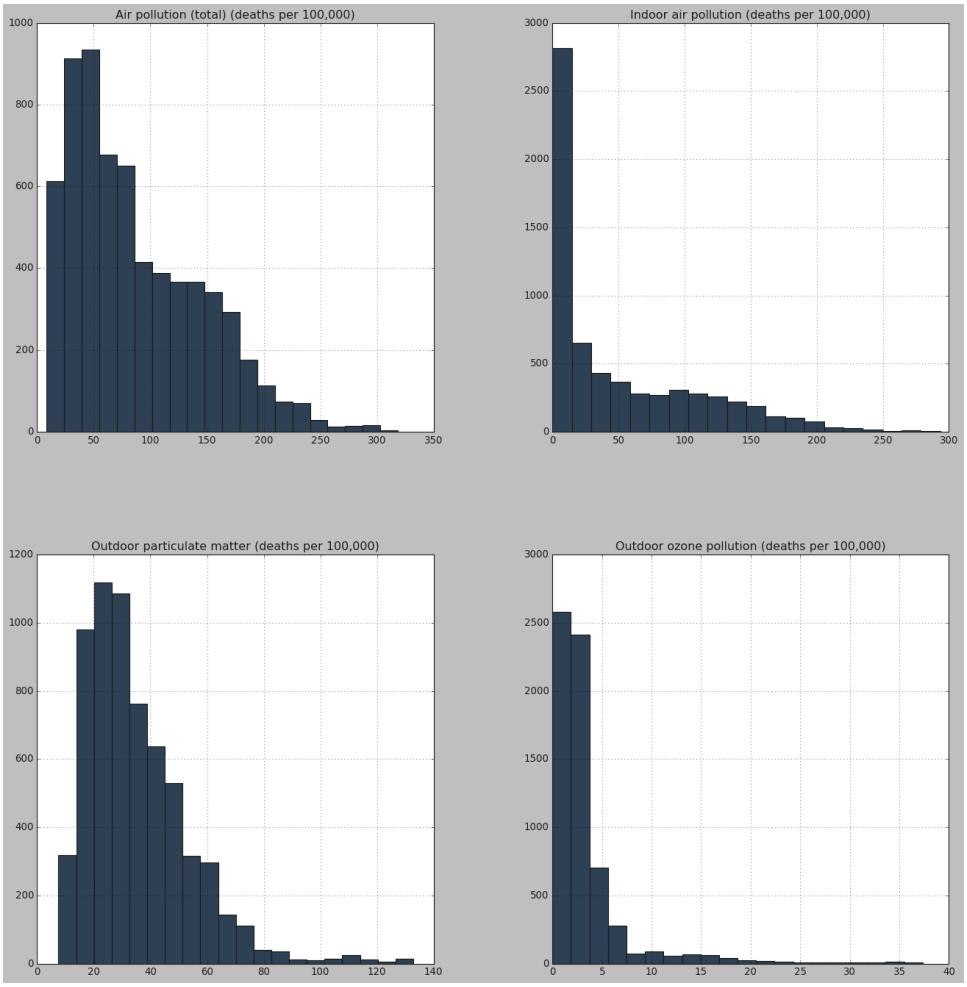

Figure 2.11: Caption

2.5.3 Plot of Different Countries on all deaths columns

The below all graphs explain how many deaths are have in different types.

```
countries = data["Entity"].unique().tolist()
def creategraph(countryname):
    Diff_countres = data[data["Entity"]==countryname]
    plt.plot(Diff_countres["Year"],Diff_countres["Indoor air
     ↪ pollution (deaths per 100,000)"],label="Indoor")
    plt.plot(Diff_countres["Year"],Diff_countres["Outdoor
     ↪ ozone pollution (deaths per 100,000)"],label="Outdoor
     ↪ Ozone")
    plt.plot(Diff_countres["Year"],Diff_countres["Outdoor
     ↪ particulate matter (deaths per
     ↪ 100,000)"],label="Outdoor Particular")
    plt.plot(Diff_countres["Year"],Diff_countres["Air
     ↪ pollution (total) (deaths per 100,000)"],label="Total
     ↪ Air Pollution")
    plt.legend(loc='center left',bbox_to_anchor=(1, 0.5))
    plt.title(countryname)
    plt.xlabel("Year")
    plt.ylabel("Deaths Per 100K")
    plt.show()
for i in range(15):
```

```
creategraph(countries[i])

plt.style.use('fivethirtyeight')
```

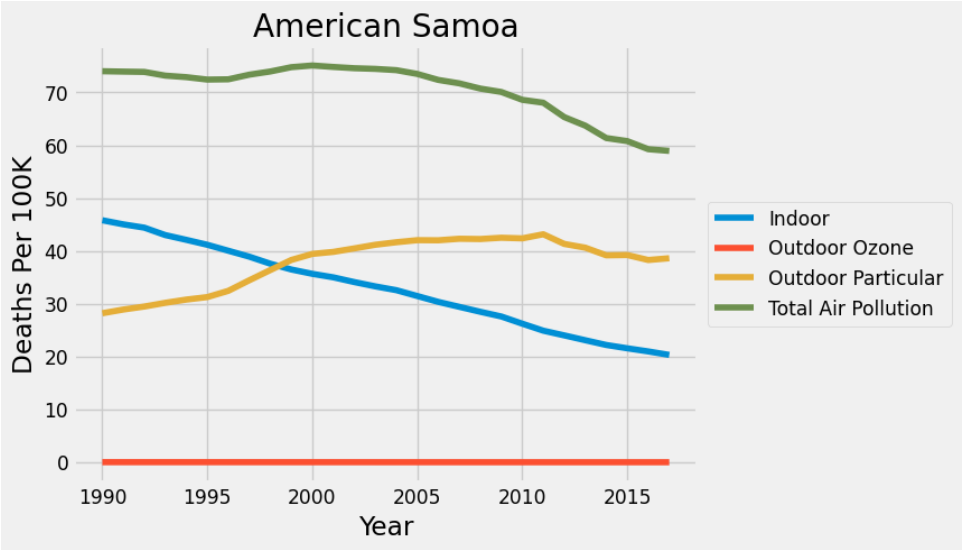

Figure 2.12: Australasia Deaths

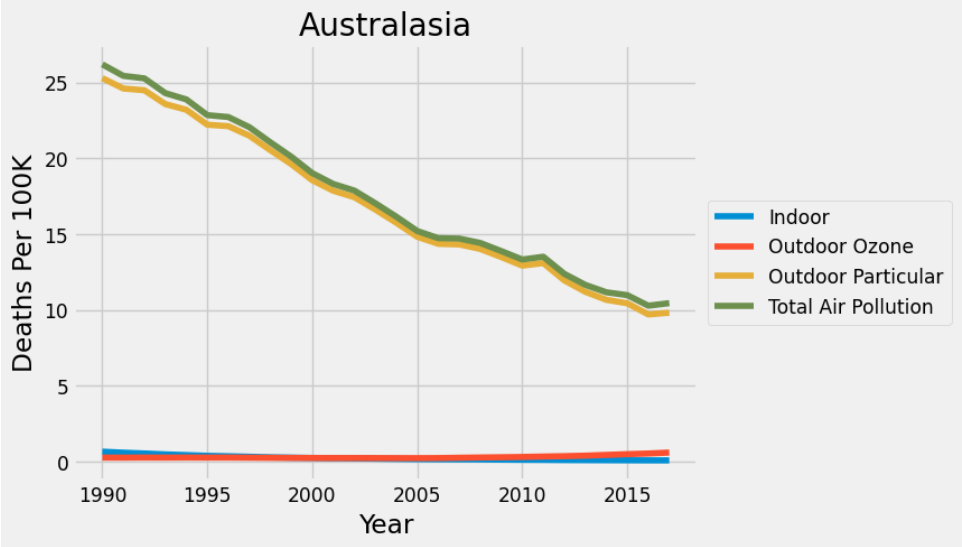

Figure 2.13: Caption

Chapter 3

Code

3.1 Explain Your Code With Outputs

In this project we used two Machine Learning Algorithms for finding the best Predictions.

One is Linear Regression Algorithm and Another one is Random Forest Algorithm. The Random Forest is given best Accuracy. So, we taken that model...

3.1.1 Importing Required Modules

For this project we are importing Pandas for data manuplation, Numpy for calculations, Matplotlib for Graphs, Seaborn for advanced plots.

```
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
```

```
from sklearn.metrics import mean_squared_error

from sklearn.metrics import mean_absolute_error

from sklearn.metrics import r2_score
```

3.1.2 Reading / Importing Dataset

```
[In]  data = pd.read_csv('Death-rates-from-air-pollution.csv')

      data
```

	Entity	Code	Year	Air pollution (total) (deaths per 100,000)	Indoor air pollution (deaths per 100,000)	Outdoor particulate matter (deaths per 100,000)	Outdoor ozone pollution (deaths per 100,000)
0	Afghanistan	AFG	1990	299.477309	250.362910	46.446589	5.616442
1	Afghanistan	AFG	1991	291.277967	242.575125	46.033841	5.603960
2	Afghanistan	AFG	1992	278.963056	232.043878	44.243766	5.611822
3	Afghanistan	AFG	1993	278.790815	231.648134	44.440148	5.655266
4	Afghanistan	AFG	1994	287.162923	238.837177	45.594328	5.718922
...	...	...	...	...	...	...	...
6463	Zimbabwe	ZWE	2013	143.850145	113.456097	27.589603	4.426291
6464	Zimbabwe	ZWE	2014	138.200536	108.703566	26.760618	4.296971
6465	Zimbabwe	ZWE	2015	132.752553	104.340506	25.715415	4.200907
6466	Zimbabwe	ZWE	2016	128.692138	100.392287	25.643570	4.117173
6467	Zimbabwe	ZWE	2017	125.028843	96.235996	26.166182	4.052495

Figure 3.1: Dataset

3.1.3 Finding Maximum Year and Minimum Year

```
# Input Code

print('Min Year :',data['Year'].min(),'<==> Max Year :
↪  ',data['Year'].max())


#Output

Min Year : 1990 <==> Max Year : 2017
```

3.1.4 Shape of the Dataset

```
#Input

   shape = data.shape

   print('Shape of the Dataset is :',shape)


#Output

  Shape of the Dataset is : (6468, 7)
```

3.1.5 Information about Dataset

```
#Input

  data.info()


#Output

    <class 'pandas.core.frame.DataFrame'>

    RangeIndex: 6468 entries, 0 to 6467

    Data columns (total 7 columns):

     #   Column

      ↪  Non-Null Count  Dtype

    ---  ------

     ↪  --------------  -----

     0   Entity                                                6468

      ↪  non-null   object
```

```
1   Code                                                5488
 ↪  non-null   object
2   Year                                                6468
 ↪  non-null   int64
3   Air pollution (total) (deaths per 100,000)          6468
 ↪  non-null   float64
4   Indoor air pollution (deaths per 100,000)           6468
 ↪  non-null   float64
5   Outdoor particulate matter (deaths per 100,000)     6468
 ↪  non-null   float64
6   Outdoor ozone pollution (deaths per 100,000)        6468
 ↪  non-null   float64
dtypes: float64(4), int64(1), object(2)
memory usage: 353.8+ KB
```

3.1.6 Reading top 5 Rows in the Dataset

```
#Input
    data.head()
#Output
```

	Entity	Code	Year	Air pollution (total) (deaths per 100,000)	Indoor air pollution (deaths per 100,000)	Outdoor particulate matter (deaths per 100,000)	Outdoor ozone pollution (deaths per 100,000)
0	Afghanistan	AFG	1990	299.477309	250.362910	46.446589	5.616442
1	Afghanistan	AFG	1991	291.277967	242.575125	46.033841	5.603960
2	Afghanistan	AFG	1992	278.963056	232.043878	44.243766	5.611822
3	Afghanistan	AFG	1993	278.790815	231.648134	44.440148	5.655266
4	Afghanistan	AFG	1994	287.162923	238.837177	45.594328	5.718922

Figure 3.2: Top 5 Rows

3.1.7 Data Cleaning

Finding Null Values

```
#Input

    print('Counting Null values for each colomns
     ↪  \n',data.isnull().sum())


#Output

Counting Null values for each colomns

Entity                                                0

Code                                                980

Year                                                  0

Air pollution (total) (deaths per 100,000)            0

Indoor air pollution (deaths per 100,000)             0

Outdoor particulate matter (deaths per 100,000)       0

Outdoor ozone pollution (deaths per 100,000)          0

dtype: int64
```

3.1.8 Machine Learning

DROPING UNWANTATED COLUMNS

```
data_rf.drop(['Entity', 'Code'], axis = 1, inplace = True)
```

#Output

	Year	Air pollution (total) (deaths per 100,000)	Indoor air pollution (deaths per 100,000)	Outdoor particulate matter (deaths per 100,000)	Outdoor ozone pollution (deaths per 100,000)
0	1990	299.477309	250.362910	46.446589	5.616442
1	1991	291.277967	242.575125	46.033841	5.603960
2	1992	278.963056	232.043878	44.243766	5.611822
3	1993	278.790815	231.648134	44.440148	5.655266
4	1994	287.162923	238.837177	45.594328	5.718922

Figure 3.3: DataSet

CHECKING NULL VALUES

#Input

```
data_rf.isna().sum()
```

#Output

```
Year                                                0
Air pollution (total) (deaths per 100,000)          0
Indoor air pollution (deaths per 100,000)           0
Outdoor particulate matter (deaths per 100,000)     0
Outdoor ozone pollution (deaths per 100,000)        0
dtype: int64
```

SELECTING FEATURES

```
#Code
    x = data_rf.drop('Air pollution (total) (deaths per
     ↪  100,000)', axis = 1)
    y = data_rf['Air pollution (total) (deaths per 100,000)']
```

TRAINING THE DATA

```
#Code
    x_train, x_test, y_train, y_test = train_test_split(x, y,
     ↪  test_size = 0.2, random_state = 42)
```

CALLING MODEL

```
#Code
    model = RandomForestRegressor()

#Output
    RandomForestRegressor()
```

FITTING MODEL

```
# Code
    model.fit(x_train, y_train)
```

PREDICTING VALUES

```
# Input

    y_pred = model.predict(x_test)

    data_pred = pd.DataFrame(y_pred, columns = ['Prediction'])

    data_pred.head(10)


# Ouput
```

	Prediction
0	79.291180
1	56.902063
2	68.791344
3	25.533878
4	29.681421
5	114.658701
6	24.900912
7	66.335895
8	181.924569
9	23.789964

Figure 3.4: TOP 10 PREDICTIONS

PREDICTING SPECIFIC YEAR

```
# Input

    new_data = pd.DataFrame({'Year' : [2022],'Indoor air
    ↪ pollution (deaths per 100,000)': [2822.7177], 'Outdoor
    ↪ particulate matter (deaths per 100,000)' : [12.346],
    ↪ 'Outdoor ozone pollution (deaths per 100,000)':
    ↪ [1.993388] })
```

```
    future = model.predict(new_data)
    print(future)
# Output
    [298.52103073]
```

PREDICTING YEARS BETWEEN RANGE

```
# Input
    year = [2022, 2023, 2024]
    indoor = [100, 150, 200]
    outdoor = [250, 305, 351]
    ozone_outdoor = [51, 8, 1]
    new_data = pd.DataFrame({
       'Year': year,
       'Indoor air pollution (deaths per 100,000)': indoor,
      'Outdoor particulate matter (deaths per 100,000)':
       ↪ outdoor,
      'Outdoor ozone pollution (deaths per 100,000)':
       ↪ ozone_outdoor
    })
    predictions = model.predict(new_data)
    data_frame = pd.DataFrame({'Year': year,
```

```
                                'Indoor air pollution (deaths per
                                ↪  100,000)':indoor,
                               'Outdoor particulate matter (deaths
                               ↪  per 100,000)': outdoor,
                                'Outdoor ozone pollution (deaths
                                ↪  per 100,000)' : ozone_outdoor,
                               'Predictions':predictions})
    data_frame
# Output
```

	Year	Indoor air pollution (deaths per 100,000)	Outdoor particulate matter (deaths per 100,000)	Outdoor ozone pollution (deaths per 100,000)	Predictions
0	2022	100	250	51	172.149857
1	2023	150	305	8	202.851751
2	2024	200	351	1	236.577282

Figure 3.5: Predictions

Mean Squared Error

```
# Input
    mse = mean_squared_error(y_test, y_pred)
    print("Mean Squared Error:", mse)
# Output
    Mean Squared Error: 1.509076498438048
```

Mean Absolute Error

```
# Input

    mae = mean_absolute_error(y_test, y_pred)

    print("Mean Absolute Error:", mae)

# Output

    Mean Absolute Error: 0.7508258992742878
```

R Squared

```
# Input

    r2 = r2_score(y_test, y_pred)

    print("R-squared:", r2)

#Output

    R-squared: 0.9995454146576859
```

Accuracy

```
# Input

    model.score(x_train, y_train)

# Output

    Accuracy : 0.9999253732410457
```

Chapter 4

Conclusion and Future Work

4.0.1 Conclusion

In this project, our aim is to develop a linear regression model to predict the number of deaths in different years based on available data.

Finally, this model predicts the number of Deaths caused by Polluted Air in different years.

	Year	Indoor air pollution (deaths per 100,000)	Outdoor particulate matter (deaths per 100,000)	Outdoor ozone pollution (deaths per 100,000)	Predictions
0	2022	100	250	51	172.149857
1	2023	150	305	8	202.851751
2	2024	200	351	1	236.577282

Figure 4.1: Result

4.0.2 Future Work

This actually helpful for finding number of deaths by polluted air. By prediction of deaths people can.

If any want to modify to increase performance then better use Feature Engineering and Selection, Cross-Validation and etc Techniques.

Thank You