**I) Data Collection and Data Study**

Let's BEGIN!!

First things first, let us study about the following -

1) What is Air Quality Index (AQI)

2) Importance of AQI in our daily life

3) Impact of air pollution (which is indicated by AQI) on human health

4) Formula of AQI and how to calculate AQI

5) Various components/sub-indices needed to be calculated in order to calculate AQI and their individual impacts

This initial understanding is required to go ahead with the project

The dataset needed for this project was got from the Government website (Central Pollution Control Board) - https://cpcb.nic.in/

The dataset is an open source file which was generated by various weather monitoring stations located in various parts of the country. There are 691 operating stations covering 304 cities/towns located in various parts of India.

After collecting the dataset, the dataset was thoroughly studied to understand about the data and to get some initial insights which would ultimately help us in planning on the next steps.

Then, the dataset was passed on to the next step where it was processed in order to feed it to our model.

------------------------------------------------------------------------------------------------------------------

**II) EDA and pre-processing**

The raw dataset obtained, had several NaN values, duplicate entries. Proceeding with the same dataset (without pre-processing) would be disastrous for the project. So, this dataset definitely needed some treatment.

Also, the dataset did not have the AQI column ; the AQI was to be calculated with the help of the individual pollutants concentration levels.

And in the end we again had to check if any NaN values had crept in, which would again hamper the entire project in the later stage.

So, we went ahead with a step-by-step treatment process as shown below -

a) Perform EDA

EDA was performed to get initial insights and to conduct preliminary examination of the data. On performing EDA, we found that our data had several NaN values, duplicate entries. So, a proper treatment of the data was very crucial.

b) Treating Data

1) Takes in "1) original\_data\_excel\_v1.xlsx"

Handling of duplicate values - count the duplicate value sets - keep only one row from each set

Gives out "2) data\_after\_duplicate\_v2.xlsx"

2) Takes in "2) data\_after\_duplicate\_v2.xlsx"

Handling of NaN or Na or NULL values - i) for categorical columns - used 'backward fill' method followed by

'forward fill' method

Gives out "3) data\_after\_fillna\_v3.xlsx"

Takes in "3) data\_after\_fillna\_v3.xlsx" - ii) for numerical columns - used 'backward fill' method followed by

'forward fill' method

Gives out "4) data\_after\_bfill\_ffill\_v4.xlsx"

X3) Takes in "4) data\_after\_bfill\_ffill\_v4.xlsx"

Scaling is done for (numerical) columns - used 'Min-Max Scaling' or 'Normalization' method

=> range of values is always between 0 and 1 (inclusive of 0 and 1)

Gives out "X5) data\_after\_minmax\_scaling\_vX5.xlsx"

(Step 3 is OPTIONAL - it was finally DEPRICATED)

c) Calculate AQI

3) Takes in "4) data\_after\_bfill\_ffill\_v4.xlsx"

Rename the file as "AQI.xlsx"

Calculated AQI and formed the AQI column

Gives out "5) data\_after\_AQI\_v5.xlsx"

Rename the file as "AQIdata.xlsx"

d) Pre processing again for final touch

Rename "AQIdata.xlsx" as "5) data\_after\_AQI\_v5.xlsx"

4) Takes in "5) data\_after\_AQI\_v5.xlsx"

Final treatment for NaN value which had again crept up - used 'backward fill' method followed by

'forward fill' method

i) Categorical - Gives out "6) data\_after\_fillna\_v6\_final.xlsx"

ii) Numerical - Gives out "data\_final.xlsx"

Now that we have the cleaned/ pre-processed dataset with us, we can now go ahead with the next step and feed this to the FB\_Prophet model.

------------------------------------------------------------------------------------------------------------------

**III) Forecasting using FB\_Prophet**

Now that we have completed the pre-processing of the data, we have the cleaned/pre-processed data file which we will feed to the FB\_Prophet model.

Let us see the steps -

1) Ask the user to input the city for which the AQI is to be forecasted

2) Drop all the other cities' data (just keep the data of the city which the user has entered in the above step)

3) A few blank values had crept in the 'Date' column - which was treated by dropping such rows (there were only 5

such rows ; which is very less compared to the total 4.3 lakh rows - so we can safely drop them!)

4) FB\_Prophet requires only 2 columns (in our case, AQI column which is renamed as "y" and the date column

which is renamed as "ds" - because that is the format which is to be given to FB\_Prophet)

Drop all the other columns

5) Split the dataset into 'training data' and 'testing data'

6) Do the Model Training

7) Test the model

8) Evaluate the performance of our model - we will use MAE (Mean Absolute Error) and MAPE (Mean Absolute

Percentage Error)

9) Time for some Interactive Prediction! Ask the user to input the time duration for which the user wants to do

the prediction - for the city which the user has entered in step 1

10) Do the prediction

11) Plot the forecast for better visualization

------------------------------------------------------------------------------------------------------------------

**IV) Forecasting using ARIMA**

Now that we have successfully predicted using FB\_Prophet, it is time for some validation. We will do so by doing the same prediction using some other model.

Let us choose ARIMA for doing the next prediction.

Here are the steps -

1) Ask the user to input the city for which the AQI is to be forecasted

2) Drop all the other cities' data (just keep the data of the city which the user has entered in the above step)

3) A few blank values had crept in the 'Date' column - which was treated by dropping such rows (there were only 5

such rows ; which is very less compared to the total 4.3 lakh rows - so we can safely drop them!)

4) ARIMA requires only 2 columns (in our case, AQI column which is renamed as "y" and the date column

which is renamed as "ds" - because that is the format which is to be given to ARIMA)

Drop all the other columns

5) ARIMA needs 3 parameters to be defined - p: The number of lag observations included in the model, also called

the lag order.

d: The number of times that the raw observations are differenced, also

called the degree of difference.

q: The size of the moving average window, also called the order of

moving average.

6) Define 'd' first - Perform Augmented Dickey-Fuller test - if p<0.05, then d is set to 0

else, take the difference of data - now perform Augmented Dickey-Fuller test - if p<0.05, then d is set to 1

else, take the difference once again - and repeat until p<0.05 - d is set to 'the number of times difference

was taken until p<0.05'

In our case, we find that d = 0

7) Define 'p' and 'q' - use 'auto\_arima' inbuilt function to automatically select the best set of p,d,q (However

in our case we already found that d = 0 ; so choose only p and q using 'auto\_arima'

Using auto\_arima, p = 1 and q = 0

8) Also, Manually verify the result got from auto\_arima by plotting Autocorrelation function (ACF) and Partial

Autocorrelation Function (PACF, also called Partial ACF) plots

ACF and PACF plots help identify the potential values for p and q

9) From ACF and PACF plots, we find that p = 2 and q = 0

Relying on the observations on ACF and PACF plots rather than that of auto\_arima, finally for our model, p = 2,

d = 0 and q = 0

10) Split the dataset into 'training data' and 'testing data'

11) Do the Model Training

12) Test the model

13) Evaluate the performance of our model - we will use MAE (Mean Absolute Error) and MAPE (Mean Absolute

Percentage Error)

14) Time for some Interactive Prediction! Ask the user to input the time duration for which the user wants to do

the prediction - for the city which the user has entered in step 1

15) Do the prediction

16) Plot the forecast for better visualization

----------------------------------------------------------------------------------------------------------------

**V) Conclusion**

Now that we have completed the prediction using both FB\_Prophet and ARIMA (for the same city and same time duration), we can compare the results of both the models.

By looking at MAE and MAPE values for multiple predictions for multiple cities and multiple time durations, we could safely conclude that FB\_Prophet MODEL PERFORMS BETTER AS COMPARED TO ARIMA MODEL ; which actually validates whatever was seen through literature study (After studying multiple articles, blogs and research papers, it was found that FB\_Prophet performed better than ARIMA in general ; which tallies with our findings too).

This is because, FB\_Prophet works best with time series that have strong seasonal effects and several seasons of historical data. Also, it is robust to missing data and shifts in the trend, and typically handles outliers well.

----------------------------------------------------------------------------------------------------------------

VI) Taking it one level HIGHER!!

After everything is done, it is time to take it one level higher. All the above activities [from step I) till step VI)] was converted into a much more interactive GUI (Graphical User Interface) - so that the user can play around a bit! In the GUI, there is are multiple options for the user - like the option to input his/her own dataset file, varying the ratio of (training data : testing data), varying the values of 'p','d','q' in the case of ARIMA model etc. The changes can be seen on-the-spot.

Also, dashboards have been included in the 'prediction' section - making it even more interactive!

"streamlit" tool was used to create the interactive GUI. streamlit is used to transform Python scripts into interactive web apps in minutes. Also, it can be used to build dashboards, generate reports, or create chat apps. In short, streamlit is one awesome tool as in - streamlit is meant to help data scientists or machine learning engineers who are not web developers but still want to build web apps.

To run the Web-Based Interactive GUI, open the "AQIFinal.py" file in an Python IDE (preferably VS Code)

Run the "AQIFinal.py" code

And in the terminal, type "streamlit run Your\_File\_Name.py" (in our case, it is "streamlit run Final.py") and run it.

Please feel free to play around a bit :)

HOPE YOU LIKED IT!!

---------------------------------------------------------------------------------------------------------------

Last, but not the least - thanks to all the various resources and references

**REFERENCES**

1) https://cpcb.nic.in/

2) https://moef.gov.in/moef/environment/pollution/index.html#:~:text=National%20Ambient%20Air%20Quality%20Monitoring%20Programme%3A&text=The%20ambient%20air%20quality%20monitoring,States%20and%206%20Union%20Territories

3) https://www.pranaair.com/blog/what-is-air-quality-index-aqi-and-its-calculation/

4) https://peerj.com/preprints/3190/

5) https://facebook.github.io/prophet/

6) https://www.analyticsvidhya.com/blog/2022/04/an-end-to-end-guide-on-time-series-forecasting-using-fbprophet/

7) https://www.diva-portal.org/smash/get/diva2:1266336/FULLTEXT01.pdf

8) https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp

9) https://www.projectpro.io/article/how-to-build-arima-model-in-python/544

10) https://medium.com/analytics-vidhya/time-series-forecasting-arima-vs-prophet-5015928e402a

11) https://www.kaggle.com/code/sadeghjalalian/time-series-forecasting-using-arima-prophet

12) https://docs.streamlit.io/

13) https://www.datacamp.com/tutorial/streamlit

---------------------------------------------------------------------------------------------------------------