Air Quality Index Prediction Model

-Bhargavi Jadhav

Problem Statement:

Predict various parameters of air quality using Python.

Approach:

1. Understanding the data.
2. Data processing:
   This is done by making relevant changes to the data frame in order to get data that is fit for modelling, like:
   i) Changing the decimal
   ii) Slicing the data frame to get rid of unwanted columns
   iii) Dropping the unwanted rows
   iv) Handling the missing values and converting them to the mean values of the column

	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O3)	T	RH
0	10/03/2004	18.00.00	2.60000	1360.0	150.0	11.9	1046.0	166.000000	1056.0	113.000000	1692.0	1268.0	13.6	48.9
1	10/03/2004	19.00.00	2.00000	1292.0	112.0	9.4	955.0	103.000000	1174.0	92.000000	1559.0	972.0	13.3	47.7
2	10/03/2004	20.00.00	2.20000	1402.0	88.0	9.0	939.0	131.000000	1140.0	114.000000	1555.0	1074.0	11.9	54.0
3	10/03/2004	21.00.00	2.20000	1376.0	80.0	9.2	948.0	172.000000	1092.0	122.000000	1584.0	1203.0	11.0	60.0
4	10/03/2004	22.00.00	1.60000	1272.0	51.0	6.5	836.0	131.000000	1205.0	116.000000	1490.0	1110.0	11.2	59.6
5	10/03/2004	23.00.00	1.20000	1197.0	38.0	4.7	750.0	89.000000	1337.0	96.000000	1393.0	949.0	11.2	59.2

3. Time series analysis is done using the FB Prophet model. The following steps were followed:
   i) Creating a data frame with respect to the Prophet model
   ii) Using an appropriate format for date and time

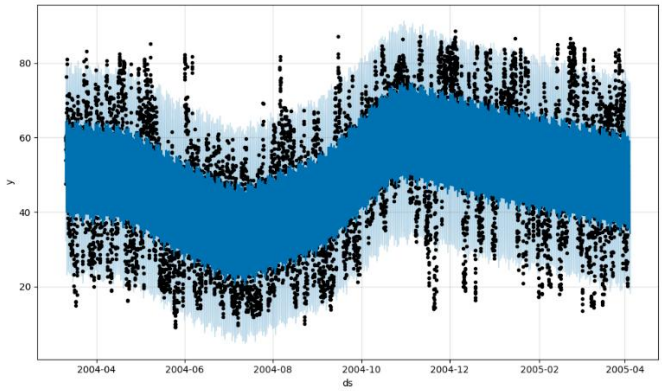
```
data
```

	ds	y	RH_lag1	AH
1	2004-03-10 19:00:00	47.7	48.9	0.7255
2	2004-03-10 20:00:00	54.0	47.7	0.7502
3	2004-03-10 21:00:00	60.0	54.0	0.7867
4	2004-03-10 22:00:00	59.6	60.0	0.7888
5	2004-03-10 23:00:00	59.2	59.6	0.7848
...	...	...	...	...
9352	2005-04-04 10:00:00	29.3	36.3	0.7568
9353	2005-04-04 11:00:00	23.7	29.3	0.7119
9354	2005-04-04 12:00:00	18.3	23.7	0.6406
9355	2005-04-04 13:00:00	13.5	18.3	0.5139
9356	2005-04-04 14:00:00	13.1	13.5	0.5028

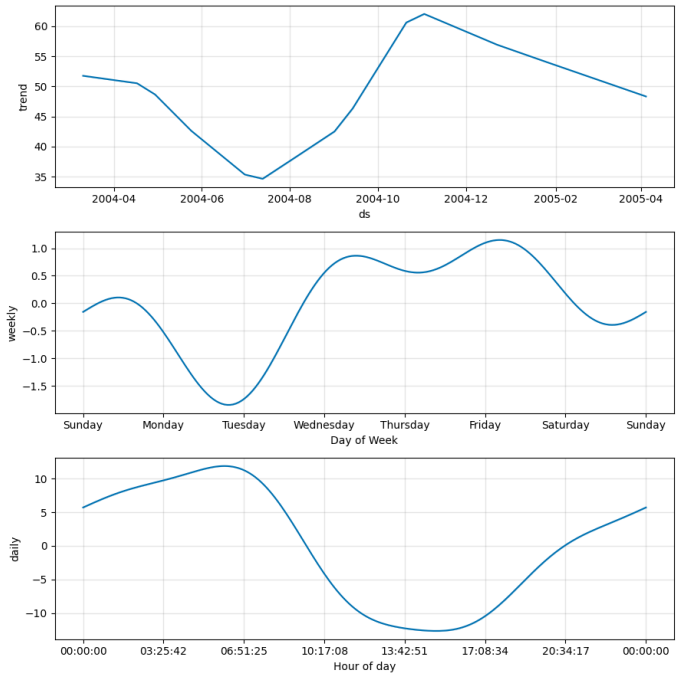
   iii) Fitting the model to the data frame
   iv) The data was split into train and test data, with the last 30 datapoints in the test data
   v) Making future predictions based on the model
   vi) XGBoostRegressor was used further to reduce the residual and improve the prediction of the model. (was chosen after a comparison between xgboost and decision tree model)

Outcomes:

	ds	yhat	yhat_lower	yhat_upper
9351	2005-04-04 10:00:00	44.267761	28.638236	60.915408
9352	2005-04-04 11:00:00	39.478999	24.005444	54.867115
9353	2005-04-04 12:00:00	36.441230	20.818122	51.561437
9354	2005-04-04 13:00:00	34.947269	18.774843	50.474574
9355	2005-04-04 14:00:00	34.261819	18.123449	49.623634



Trend and seasonality:



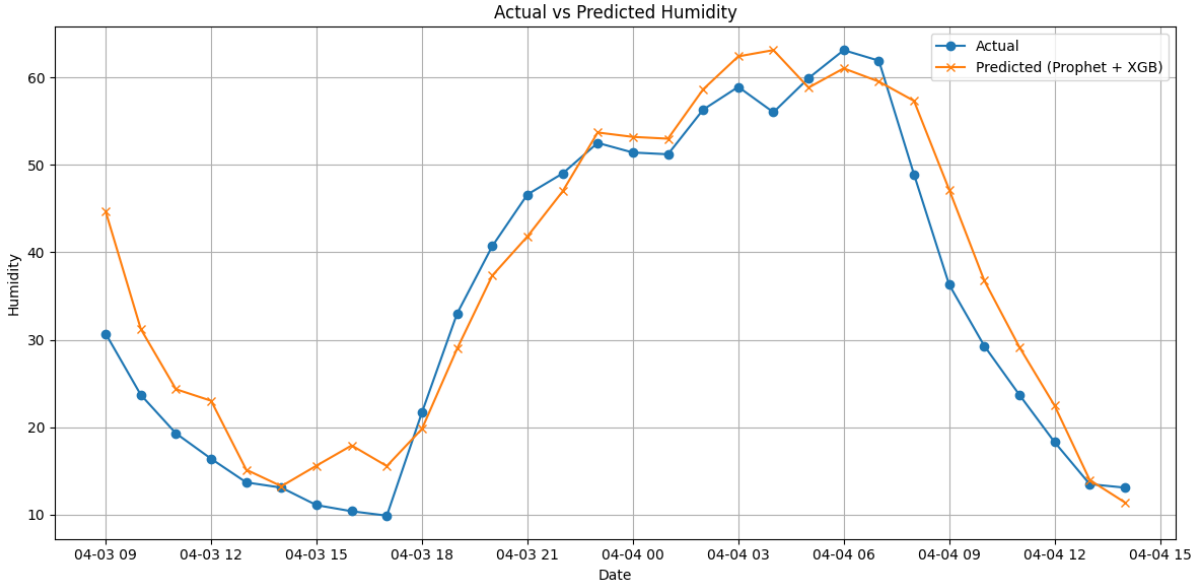
Results:

The model predicted the rh value for the last 30 datapoints with a confidence interval.

```
Prophet-only RMSE: 15.794526746007298

DT RMSE: 6.363899368280523
XGB RMSE: 5.391415487728496
```

XGB was chosen because it has less RMSE. RMSE reduced by 65.87%.



R^2 value comes out to be 0.91

The models were saved for further use.