TIME SERIES ANALYSIS

Main objective of the analysis that also specifies whether your model will be focused on a specific type of Time Series, Survival Analysis, or Deep Learning and the benefits that your analysis brings to the business or stakeholders of this data.

Main objective of the analysis is to perform Time Series Analysis for prediction.

Benefits:

- Data is about air pollution levels in India
- Accurate air quality predictions in future is imperative for necessary policy planning initiatives by Government
- Getting an idea of air quality predictions in future will help determine if existing policies and ongoing climate initiatives are successful or not

Brief description of the data set you chose, a summary of its attributes, and an outline of what you are trying to accomplish with this analysis.

The data I took is about pollutant levels in the air across cities and states of India from 1987 to 2015, with value entries ranging in frequency from weekly to monthly.

So, the data is made up of 435742 rows and 13 columns. So, we have 435742 recordings of pollution across time, city, state and land-use type. The features/ variables are:

Location of observatory: (Categorical Variables) stn_code, state, location, location_monitoring_station, agency type

Date of entry: (date variables) sampling_date, date

Pollution details: (Numerical Variables) so2, no2, rspm, spm, pm2_5

Outline: I am trying to build model to predict monthly median Air quality index for India in future by utilizing data from 1987 to 2015.

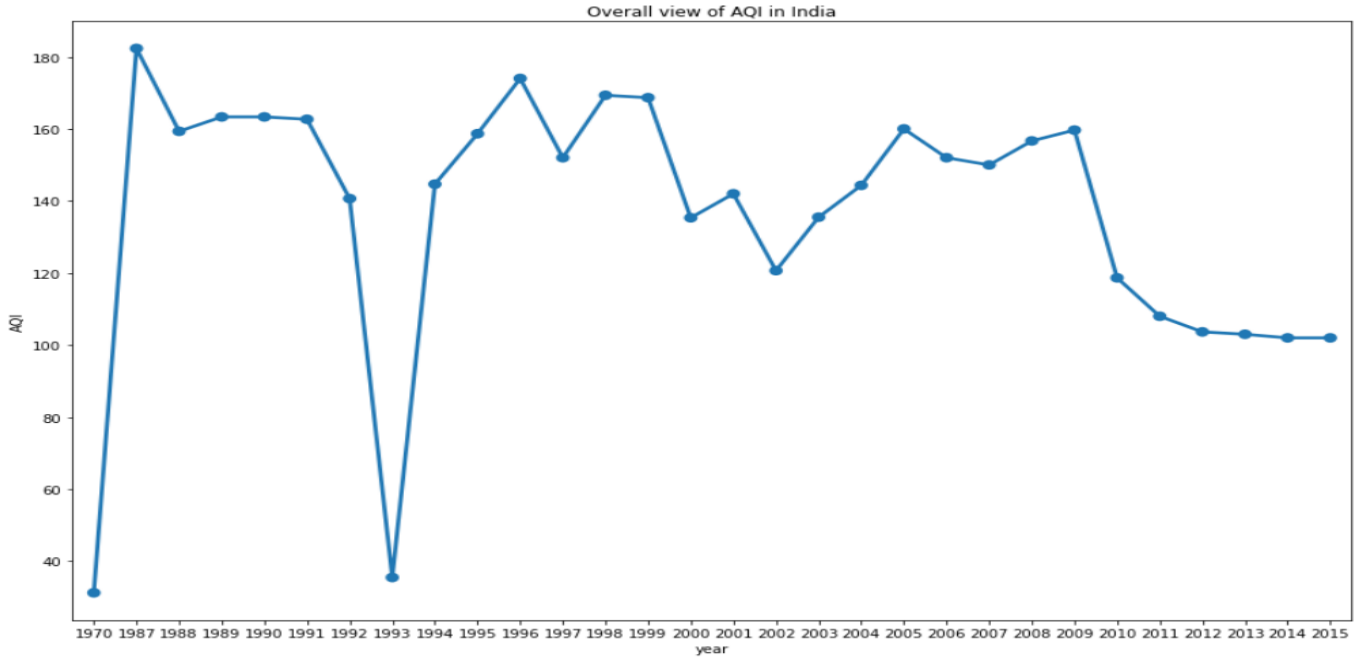
Brief summary of data exploration and actions taken for data cleaning or feature engineering.

1. Handling date variables
    a. Sampling_date contains varying date formats and that is transformed to standard timestamp in date. Hence, this variable will be dropped.
    b. Date variable has date of observation in pandas date format and hence will be used as index for series containing final AQI (Air Quality Index).

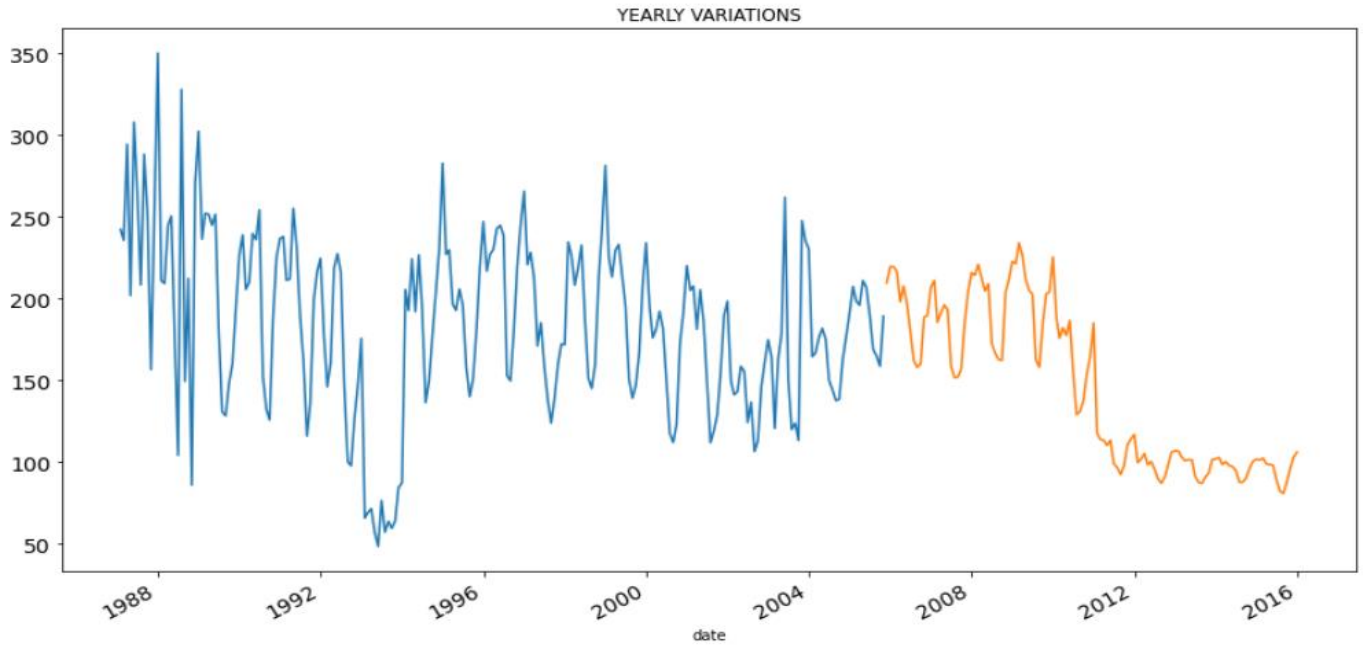
2. Handling non-numerical variables
    a. Since we are interested in calculating AQI for overall India, we need information on city and state.
    b. Hence, we drop stn_code, location_monitoring_station, agency type
    c. Now with respect to time, removing change in frequency across monitoring stations, one data point is taken for a month at a particular location. If weekly or fortnightly or tri-weekly observations are found, their mean is taken.(if missing, values other than missing are taken for mean)
    d. For multiple locations in a city, median is taken as final.
    e. For a state, mean AQI for its cities is taken as final.
    f. For overall India, median of all states is taken as final.
3. Handling missing values
    a. After above changes, missing values were present only for 2 or 3 small sates/UTs.
    b. So they are dropped from further analyses.
4. Handling outliers
    a. None encountered
5. Variable Transformation/ Feature engineering.
    a. AQI is a cumulative measure of pollutants in the air namely, so2, no2, rspm, spm, pm2_5.
    b. Based on the standard formula available in the net, these columns were transformed to form one column called AQI and individual pollutant columns are dropped.

Summary of training at least three variations of the Time Series, Survival Analysis, or Deep Learning model you selected. For example, you can use different models or different hyperparameters.

First of all I look at overall view of AQI over time:



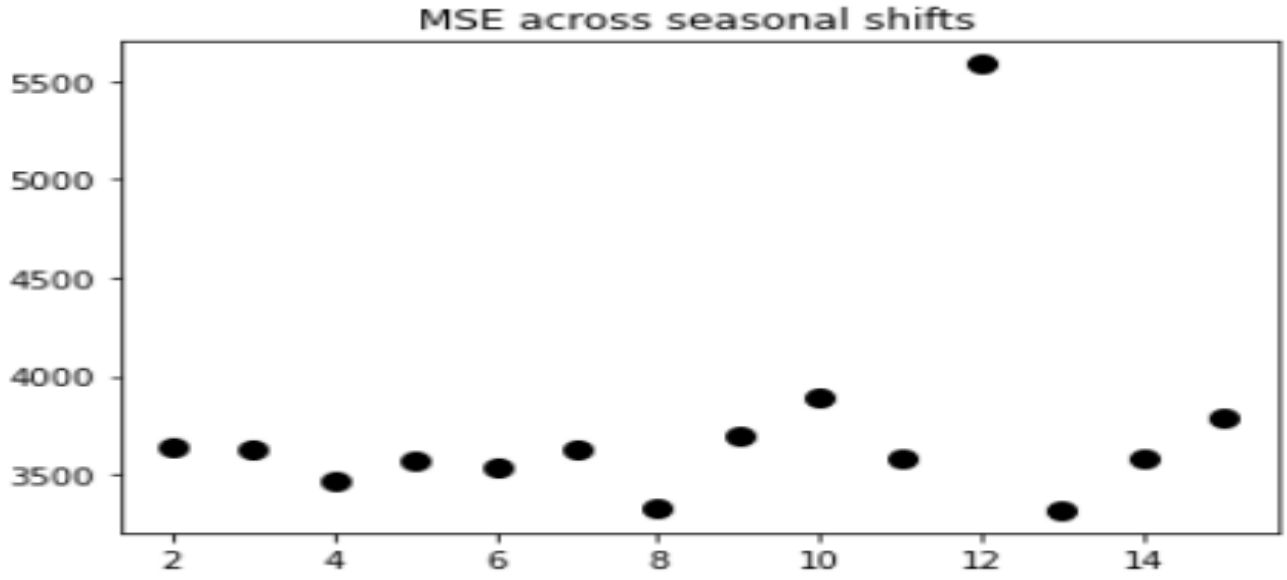
Then I am creating train and test data for comparison of predictions later. Here is the plot of that:



Now lets model!!

From the above plot we can see that there is effect of both trend and seasonality and seasonality seems independent of trend.

Hence I start with a naïve additive model with triple exponential smoothing. But its not explicit what seasonality I should take. Hence I do a grid search for it from 2 to 15.

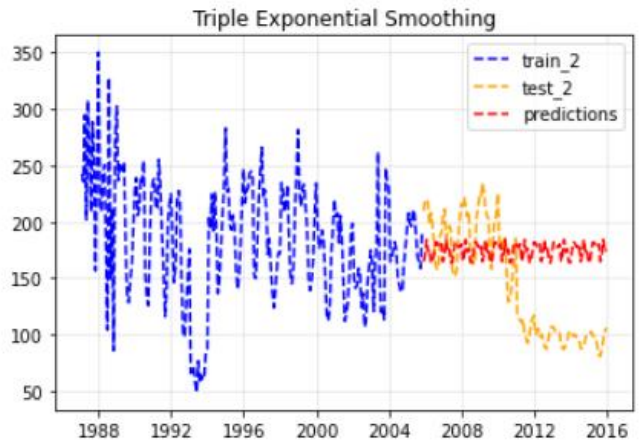


We can clearly see that both seasonality of 8 and 12 gives lowest errors. Hence these two form my first two models:

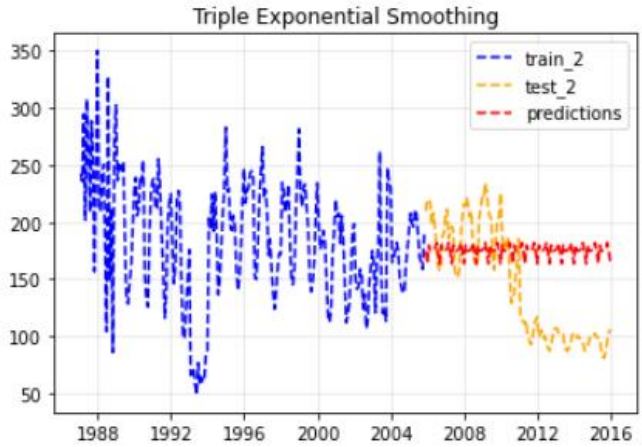
1. **Model 1**: ***ExponentialSmoothing****(train,damped=True, trend="additive", seasonal="additive", seasonal_periods=13)*
2. **Model 2**: ***ExponentialSmoothing****(train,damped=True, trend="additive", seasonal="additive", seasonal_periods=8)*

The predictions for these two models are as follows:

MSE: 3320.4353758383313

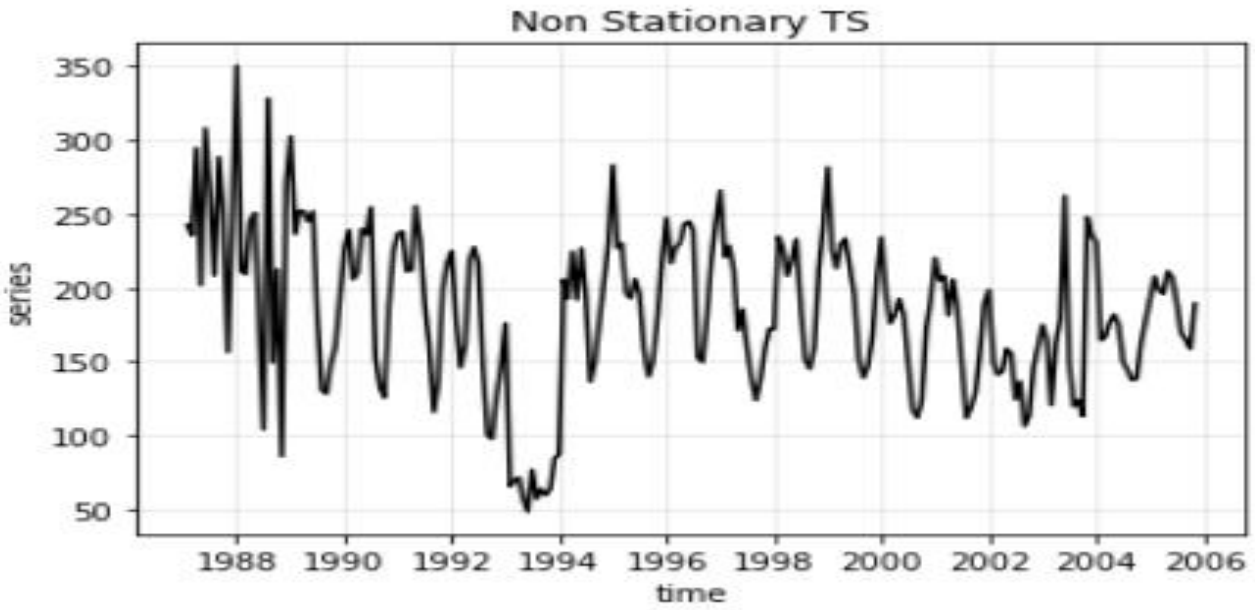


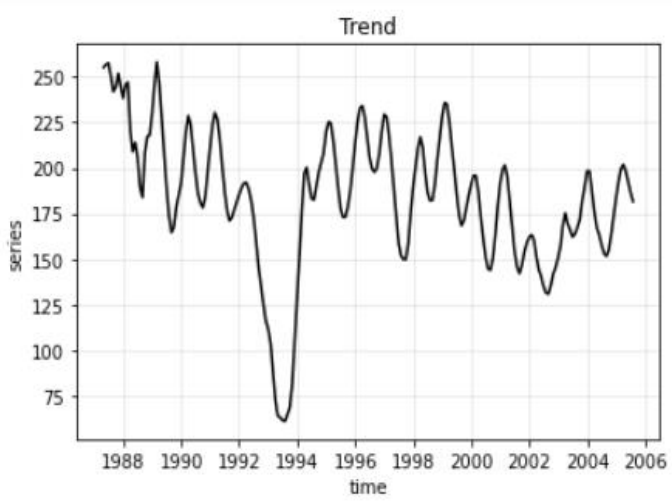
MSE: 3330.089667798365

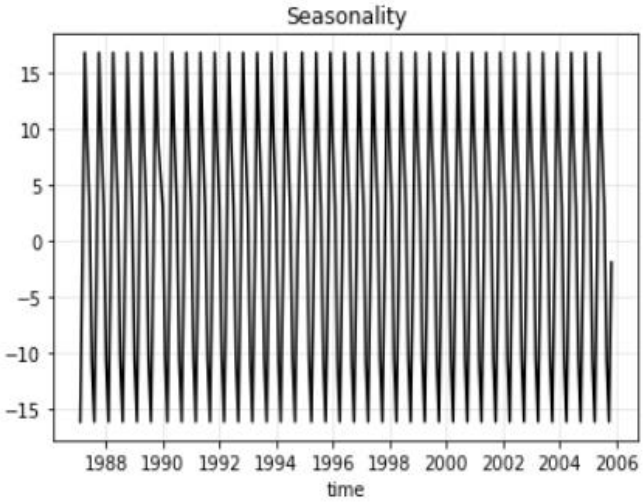


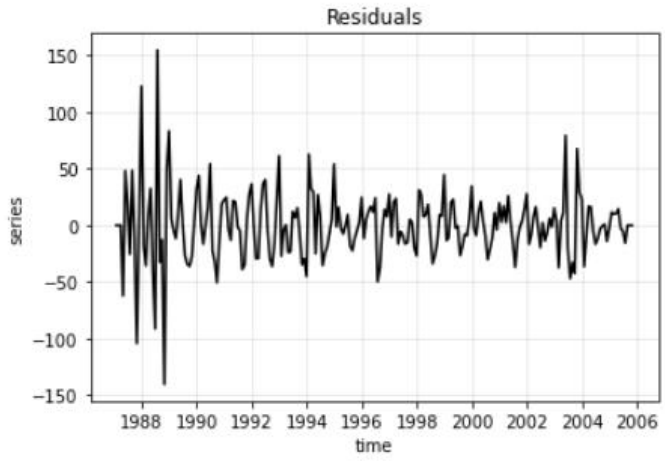
It is clear that even triple exponential smoothing gives very bad results for both models, but among them, one with seasonality of 12(plus base=13) is giving better results. Hence we assume from here on, data has seasonality 12, ie, yearly which is intuitive.

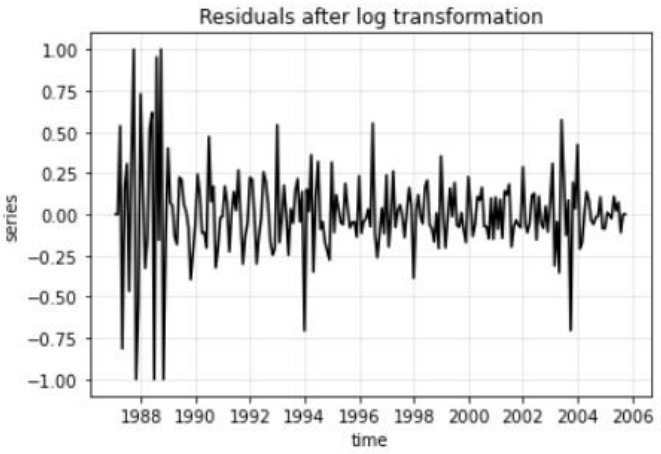
Now lets look at train portion onemore time. Clearly this is nonstationary, so we decompose based on additive model. And looking at residuals, early timeperiods have high variance than rest hence applying log transformation.







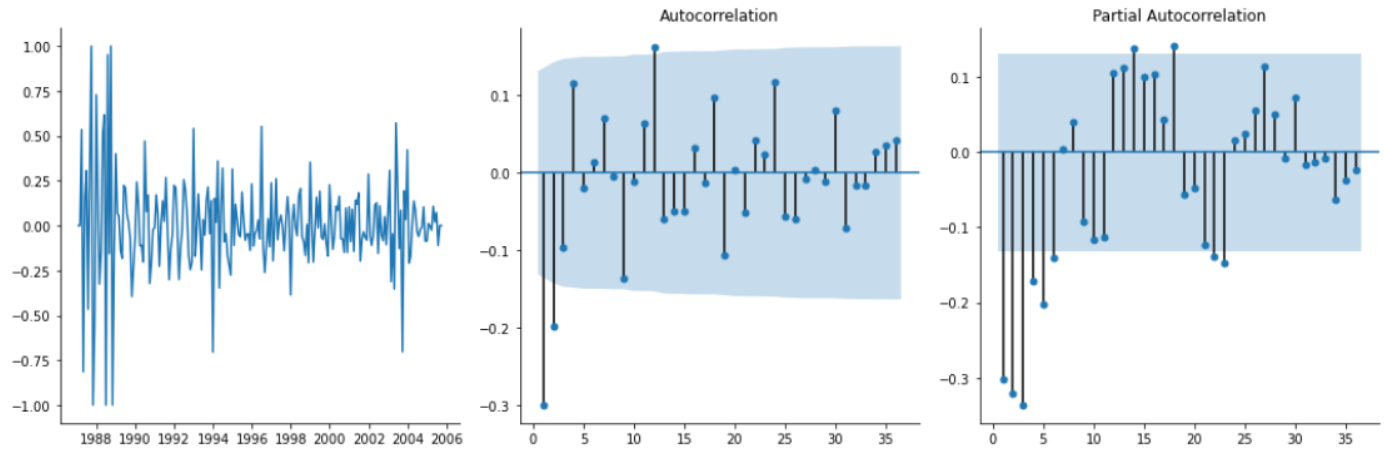




We perform DF test for testing stationarity, and we got

```
Augmented Dickey-Fuller Test
 adf: -3.7211559585342293
 pvalue: 0.0038252282011211947
 usedlag: 13
 nobs: 209
 critical_values: {'1%': -3.4620315036789666, '5%': -2.8754705024827127, '10%': -2.5741950726860647}
 icbest -88.32334215314643
```

Since p-value is less than 5%, I conclude the transformed residual is stationary. Lets study its ACF and PACF plots:



We can clearly see that:

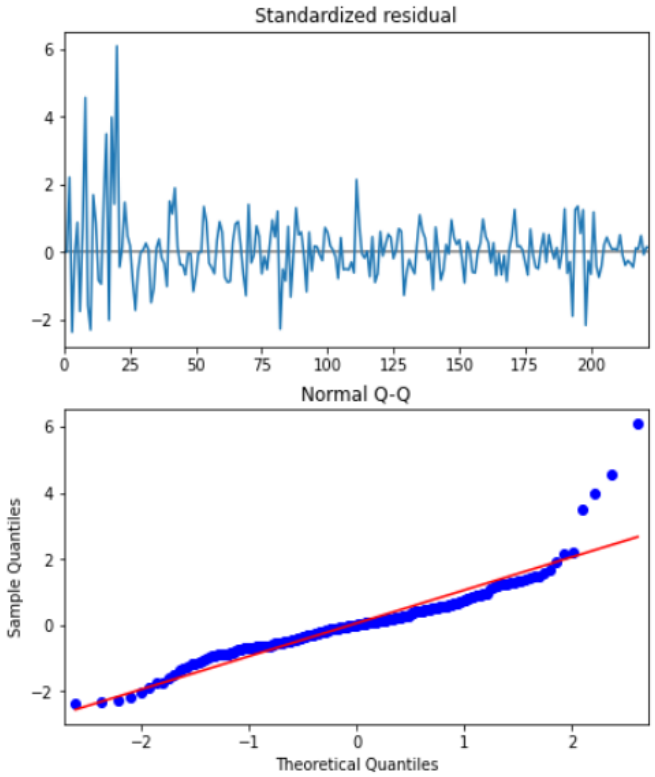
1. Series is stationary
2. No differencing is needed for ARIMA, hence a combination of AR and MA models is sufficient, ie d =0.
3. It looks from the plots that there is no seasonality effect yet we cannot be sure.
4. For an AR model, p =2 seems probable
5. For a MA model, q = 3 seems probable.

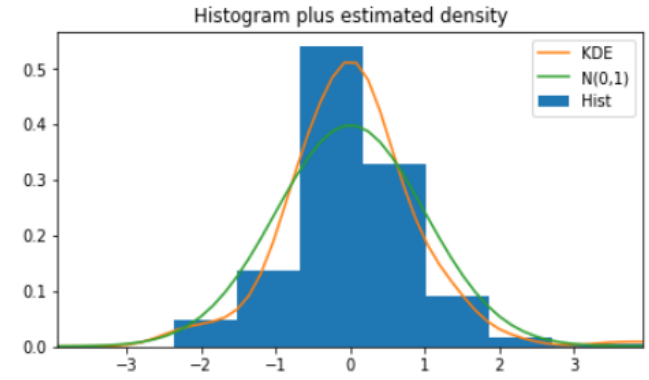
But we use the automatic ARIMA fitter to arrive at our next models (Please note that these hyperparameters have been arrived at after consideration of every one of all possible combinations through GridSearch:

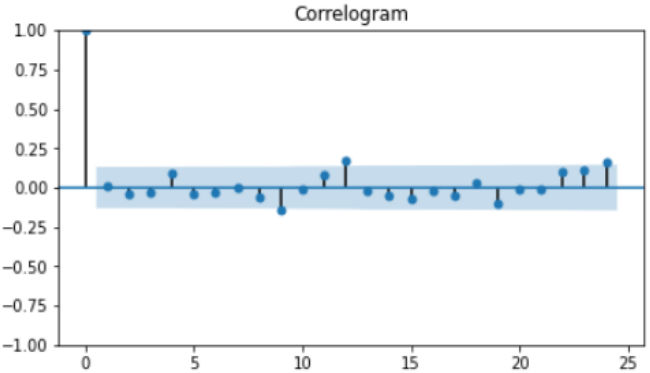
1. **Model 3: ARIMA**: ***auto_arima****(est_residual_log_diff, start_p=0, d=0, start_q=0, max_p=3, max_d=2, max_q=5, start_P=0, D=1, start_Q=0, max_P=3, max_D=2, max_Q=5, max_order=None, m=12,* ***Seasonal = False****, stationary=True, with_intercept=False, trace=True,error_action='ignore')*
2. **Model 4: SARIMA**: ***auto_arima****(est_residual_log_diff, start_p=0, d=0, start_q=0, max_p=3, max_d=2, max_q =5, start_P=0, D=1, start_Q=0, max_P=3, max_D=2, max_Q=5, max_order=None, m=12,* ***Seasonal = True****, sta tionary=True, with_intercept=False, trace=True,error_action='ignore')*

Best fit for Model 3: ARIMA(2,0,2)(0,0,0)[0]
Best fit for Model 4: ARIMA(2,0,2)(2,0,0)[12]

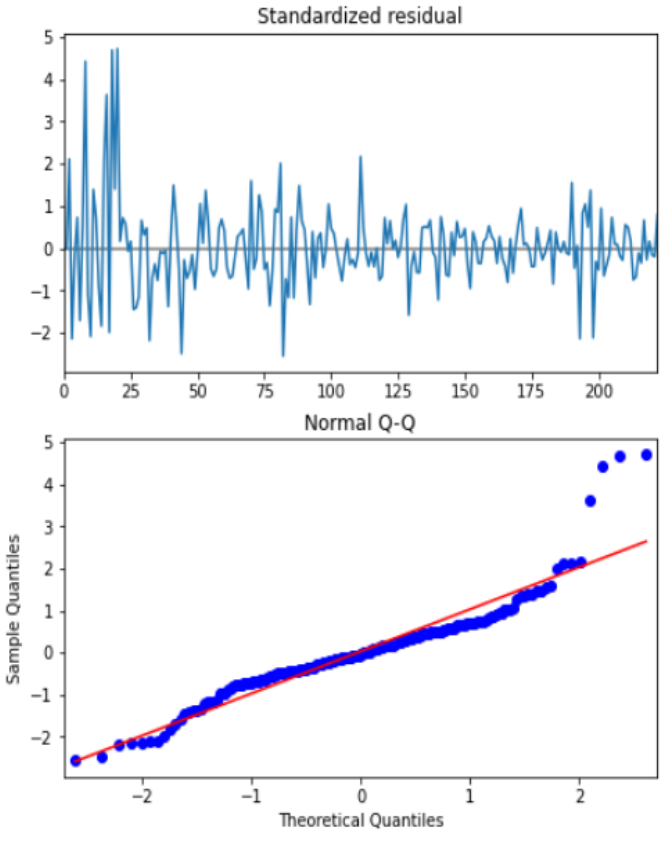
Results and goodness of fit in Model 3:

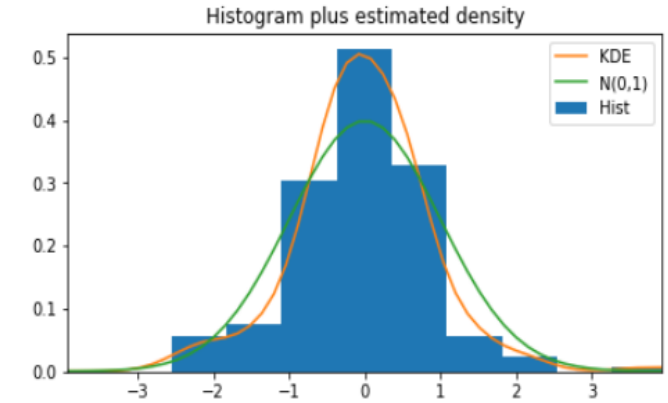


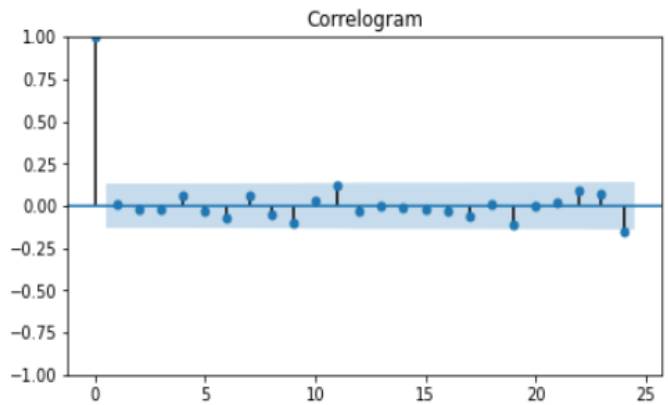




Results and goodness of fit in Model 4:







Summary of both Model 3 (left) and 4 (right) along with AIC and BIC criterion:

SARIMAX Results

Dep. Variable:	resid	**No. Observations:**	223
Model:	SARIMAX(2, 0, 2)	**Log Likelihood**	22.369
Date:	Sun, 06 Dec 2020	**AIC**	-34.739
Time:	13:13:31	**BIC**	-17.703
Sample:	0	**HQIC**	-27.862
	- 223		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	1.1104	0.081	13.714	0.000	0.952	1.269
ar.L2	-0.3793	0.059	-6.434	0.000	-0.495	-0.264
ma.L1	-1.7583	0.072	-24.282	0.000	-1.900	-1.616
ma.L2	0.8598	0.065	13.219	0.000	0.732	0.987
sigma2	0.0476	0.003	18.431	0.000	0.042	0.053

Ljung-Box (Q):	46.18	**Jarque-Bera (JB):**	700.85
Prob(Q):	0.23	**Prob(JB):**	0.00
Heteroskedasticity (H):	0.19	**Skew:**	1.62
Prob(H) (two-sided):	0.00	**Kurtosis:**	11.06

Dep. Variable:	resid	**No. Observations:**	223
Model:	SARIMAX(2, 0, 2)x(2, 0, [], 12)	**Log Likelihood**	32.748
Date:	Sun, 06 Dec 2020	**AIC**	-51.495
Time:	13:13:31	**BIC**	-27.645
Sample:	0	**HQIC**	-41.867
	- 223		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.9328	0.103	9.058	0.000	0.731	1.135
ar.L2	-0.3493	0.055	-6.305	0.000	-0.458	-0.241
ma.L1	-1.6357	0.098	-16.658	0.000	-1.828	-1.443
ma.L2	0.7832	0.086	9.112	0.000	0.615	0.952
ar.S.L12	0.2318	0.037	6.255	0.000	0.159	0.304
ar.S.L24	0.2894	0.049	5.945	0.000	0.194	0.385
sigma2	0.0427	0.002	17.276	0.000	0.038	0.048

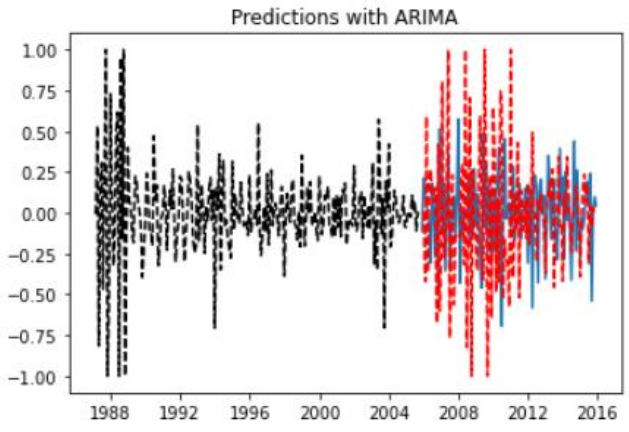
Ljung-Box (Q):	30.00	**Jarque-Bera (JB):**	334.40
Prob(Q):	0.88	**Prob(JB):**	0.00
Heteroskedasticity (H):	0.18	**Skew:**	1.19
Prob(H) (two-sided):	0.00	**Kurtosis:**	8.50

Testing for different tests:

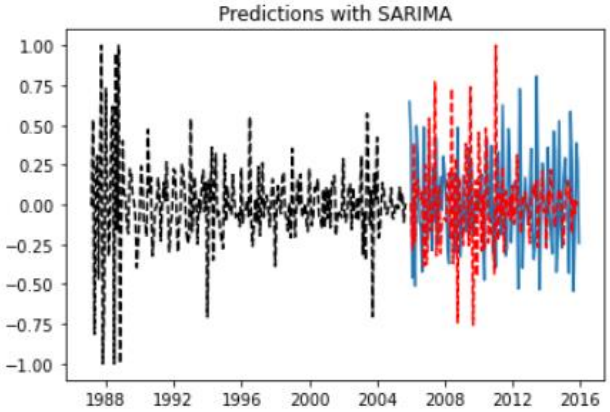
```
Test:                ARIMA                          SARIMA
Normality:           val=700.852, p=0.000,          val=334.396, p=0.000
Ljung-Box:           val=46.184, p=0.232,           val=30.001, p=0.875
Heteroskedasticity:  val=0.186, p=0.000,            val=0.184, p=0.000
Durbin-Watson:       d=1.98,                        d=1.98
```

The predictions for these two models are as follows:

MSE: 0.24293058129575476



MSE: 0.1656268877470046



A paragraph explaining which of your models you recommend as a final model that best fits your needs in terms of accuracy or explainability.

It is clear that the first 2 models were too basic and performed very poorly. Hence, we have to look at next two models.

Even though we expected q = 3 for ARMA models based on PACF plots, the automatic model finder based on AIC information criterion chose ARMA models with p and q =2. Model 3 had no seasonality component whereas model 4 had. The effect of seasonality was not visible directly through run plot or ACF or PACF. Both models performed similarly in train but model 4 tend to perform better in predictions of future.

Based on MSE and goodness of fit and tests values, I will recommend to use SARIMA model, i.e., model 4 with hyperparameters as mentioned above.

Summary Key Findings and Insights, which walks your reader through the main findings of your modeling exercise.

Findings:

1. Even our final model only gives quite decent performance, not great.
2. Residuals having high values tend to perform bad as we can see from Q-Q plot.
3. From the tests we can infer that residual have all desirable properties, viz, follows normality, serially correlated, non-heteroskedastic and passes Durbin Watson test implying that the pre-processing and decomposition is excellent.
4. Having AR order 2 implies the pollution level in current month is affected by both previous month and previous month to that which is non intuitive since the general consensus is that it only depends on previous month.
5. Having MA order 2 implies error residuals also depend on past 2 months.

Suggestions for next steps in analyzing this data, which may include suggesting revisiting this model or adding specific data features to achieve a better model.

The next steps would be to perform more advanced time series model fitting and forecasting models like Facebook's Prophet or deep learning models to improve performance of predictions.

Another flaw of this model is presence of cyclicity I guess because seasonality was not visible in plots but improved performance. A main reason for this to happen is presence of another term in additive model called cyclicity which is seasonality at irregular periods. Advanced fitting methods can be used to detect and subtract it from residuals.

If extra data is available with fixed frequency and smaller interval say daily or every alternate day, it would give much better predictions for future.