Air Quality Index Prediction

Machine Learning Internship at FTS

By,
Akshata Kotti
Shubham Urmaliya
Abhishek
Shreya Basu
Neha Kumari
Lokesh

Contents

- Problem Statement
- Data Visualisation before preprocessing
- Data Visualisation on AQI
- Data preprocessing
  - ❖ Missing value treatment
  - ❖ Air Quality Index(AQI) calculation
  - ❖ Outlier treatment
- Data Visualisation after preprocessing
- Model Making
  - ❖ XGBoost
  - ❖ Stacked LSTM

Problem Statement

To create a model which will predict the Air Quality Index (AQI).

We were given two datasets:

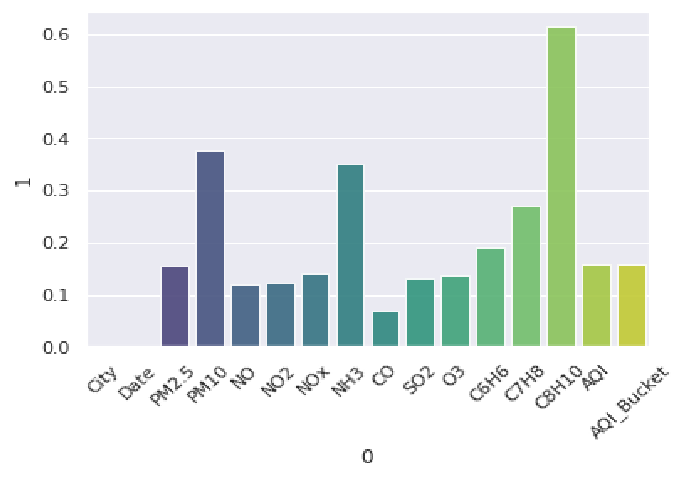
1. cities_by_day → day-wise information including the amount of various chemical substances present in different cities and the AQI information.
2. cities_by_hours → hours-wise information including the amount of various chemical substances present in different cities and the AQI information.

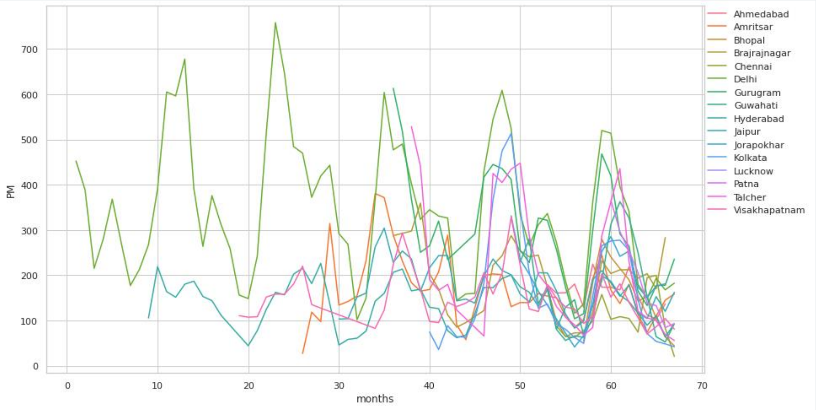
We have initially performed Exploratory Data Analysis including Data preprocessing, Outlier treatment and Data visualization to study the datasets.
We have then used certain algorithms like XGBoost and Stacked LSTM to create a model that will predict the AQI for any future reference using the input we are giving.

Data Visualisation before preprocessing

We used some visualisation techniques to understand the trends and relationships between different columns. The results are following.

- There are a lot of missing values for xylene,PM2.5 and NH3. But after looking at correlations AQI is reasonably dependent on these gases. So it is not good to drop these columns.
- The second image is a plot of PM(PM2.5 +PM10) with months. From this graph we can see that values are not missing at random they are missing for long periods of time from this we found that the imputation methods like linear interpolation will not give realistic results and we started thinking about methods like KNN imputation.





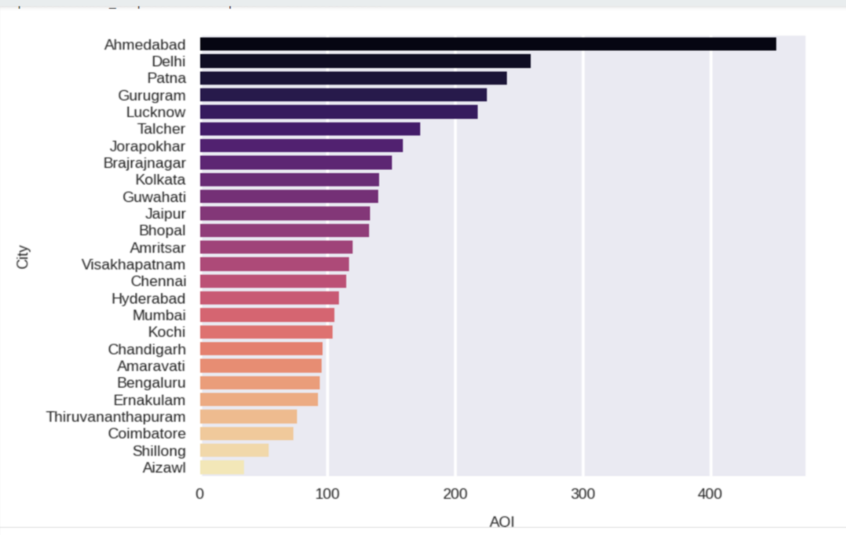
Data Visualisation on AQI

Data visualization is the graphical representation of information and data. We use different visual elements like charts, graphs, and maps, data visualization tools to provide an accessible way to see and understand trends, outliers, and patterns in data.

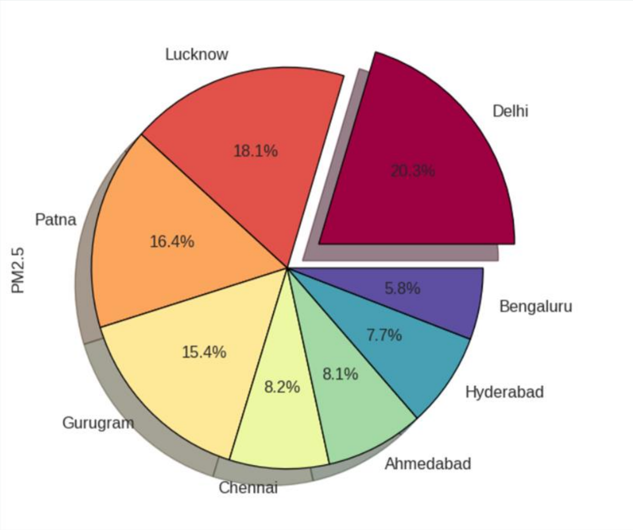
Visualization has been done on the dataset of cities_by_day to study certain trends. Some screenshots have been attached herewith. The link to the file has been given here:
https://colab.research.google.com/drive/1UIySiXXD82j0ocehY9wtBLlZj7am7gl-#scrollTo=RHBP32Q3qcLu

	Proportion
C8H10	0.613220
PM10	0.377231
NH3	0.349734
C7H8	0.272290
C6H6	0.190410
AQI	0.158511
AQI_Bucket	0.158511
PM2.5	0.155701
NOx	0.141715
O3	0.136196
SO2	0.130507
NO2	0.121398
NO	0.121296
CO	0.069723
City	0.000000
Date	0.000000

Calculating the proportion of missing values



Grouping the cities based on average AQI



Pie-chart showing distribution of pollutant in top polluted cities

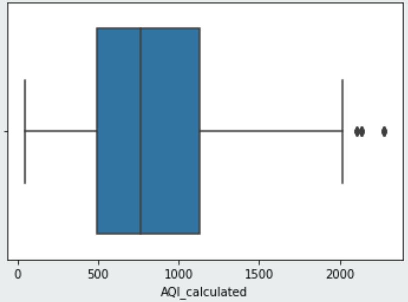
Data preprocessing

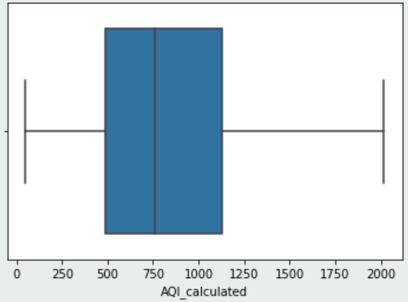
KNN Imputation

```
def fun(dframe):
  lis = []
  for i in range(0, dframe.shape[1]):
    if(dframe.iloc[:,i].dtypes == 'object'):
      dframe.iloc[:,i] = pd.Categorical(dframe.iloc[:,i])
      dframe.iloc[:,i] = dframe.iloc[:,i].cat.codes
      dframe.iloc[:,i] = dframe.iloc[:,i].astype('object')

      lis.append(dframe.columns[i])
  KNN = KNNImputer(n_neighbors=3)
  dframe = pd.DataFrame(KNN.fit_transform(dframe))
  return dframe
```

Outlier Detection Using Quantile Regression

```
Q1=df['AQI_calculated'].quantile(0.25)
Q3=df['AQI_calculated'].quantile(0.75)
IQR=Q3-Q1
print(Q1)
print(Q3)
print(IQR)
Lower_Whisker = Q1 - 1.5*IQR
Upper_Whisker = Q3 + 1.5*IQR
print(Lower_Whisker, Upper_Whisker)
df = df[df['AQI_calculated']< Upper_Whisker]
```





Data Preprocessing of Cities_by_day and Cities_by_hours dataset

1] Missing value treatment: Methods used to treat missing values are:

- Citywise Mean imputation
- Citywise Linear interpolation
- Citywise K-Nearest Neighbors(KNN) imputation

2] AQI calculation: AQI is the maximum of sub-indices calculated for individual pollutants.

3] Outlier treatment: Outliers were detected and treated using Quantile Regression.

Percentage of missing values in cities_by_day:

City	0.000000
Date	0.000000
PM2.5	15.570079
PM10	37.723071
NO	12.129626
NO2	12.139785
NOx	14.171549
NH3	34.973418
CO	6.972334
SO2	13.050692
O3	13.619586
C6H6	19.041008
C7H8	27.229014
C8H10	61.322001
AQI	15.851139
AQI_Bucket	15.851139

Percentage of missing values in cities_by_hour:

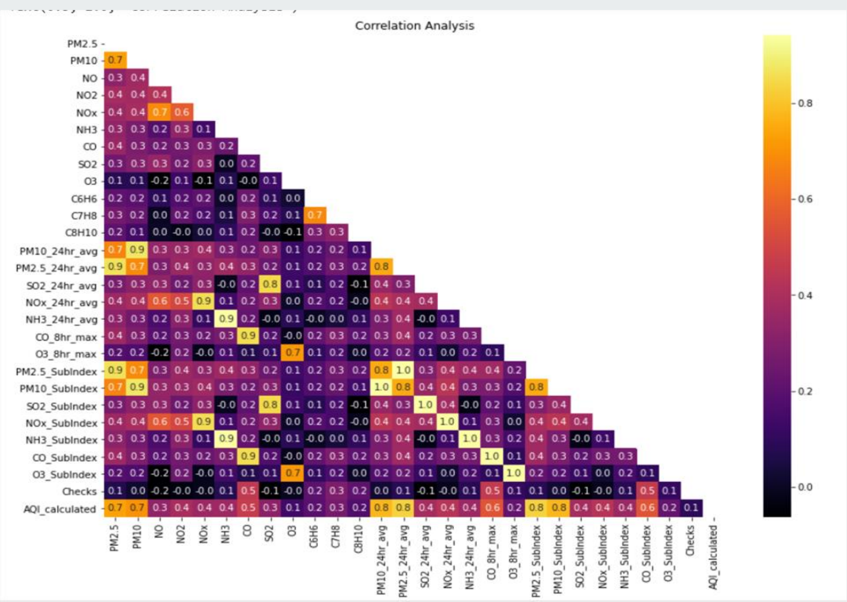
City	0.000000
Datetime	0.000000
PM2.5	20.496274
PM10	41.919407
NO	16.476355
NO2	16.545577
NOx	17.407593
NH3	38.501430
CO	12.222073
SO2	18.417517
O3	18.252940
C6H6	23.117923
C7H8	31.164683
C8H10	64.393996
AQI	18.234858
AQI_Bucket	18.234858

Data Visualisation after preprocessing

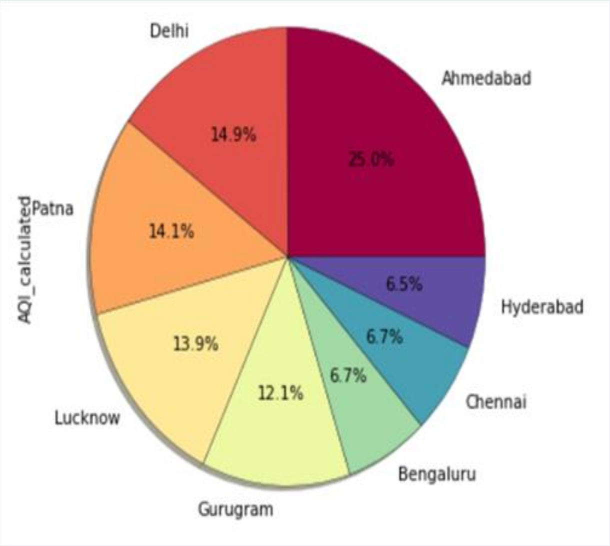
Visualization has also been performed after preprocessing the dataset cities_by_hours i.e., removing the missing values in the dataset.

	0
City	0
Datetime	0
PM2.5	0
PM10	0
NO	0
NO2	0
NOx	0
NH3	0
CO	0
SO2	0
O3	0
C6H6	0
C7H8	0
C8H10	0
AQI	129080
AQI_Bucket	129080

Proportion of missing values has been reduced to zero



Correlation analysis



Pie-chart showing imputed AQI values for top polluted cities

Model Making - (i) XGBoost Regressor

```
def fun(Ahm):
  Ahm.drop(['City'],axis=1,inplace = True)
  Ahm.set_index('Date', inplace = True)
  Ahm=Ahm.astype('float64')
  Ahm=Ahm.resample(rule='MS').mean()

ax=Ahm[['AQI_calculated']].plot(figsize=(16,12),grid=True,lw=2,color='Red')
  ax.autoscale(enable=True, axis='both', tight=True)
  X = Ahm.iloc[:, :-1]
  y = Ahm.iloc[:, -1]

  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=43)

  xgb = XGBRegressor()
  xgb.fit(X_train, y_train)
  f'Coefficient of determination R^2 on train set {xgb.score(X_train, y_train)}'
  f'Coefficient of determination R^2 on test set {xgb.score(X_test, y_test)}'

  score = cross_val_score(xgb, X, y, cv = 3)
  score.mean()
  pred = xgb.predict(X_test)

  sns.distplot(y_test - pred)
```

Final Result

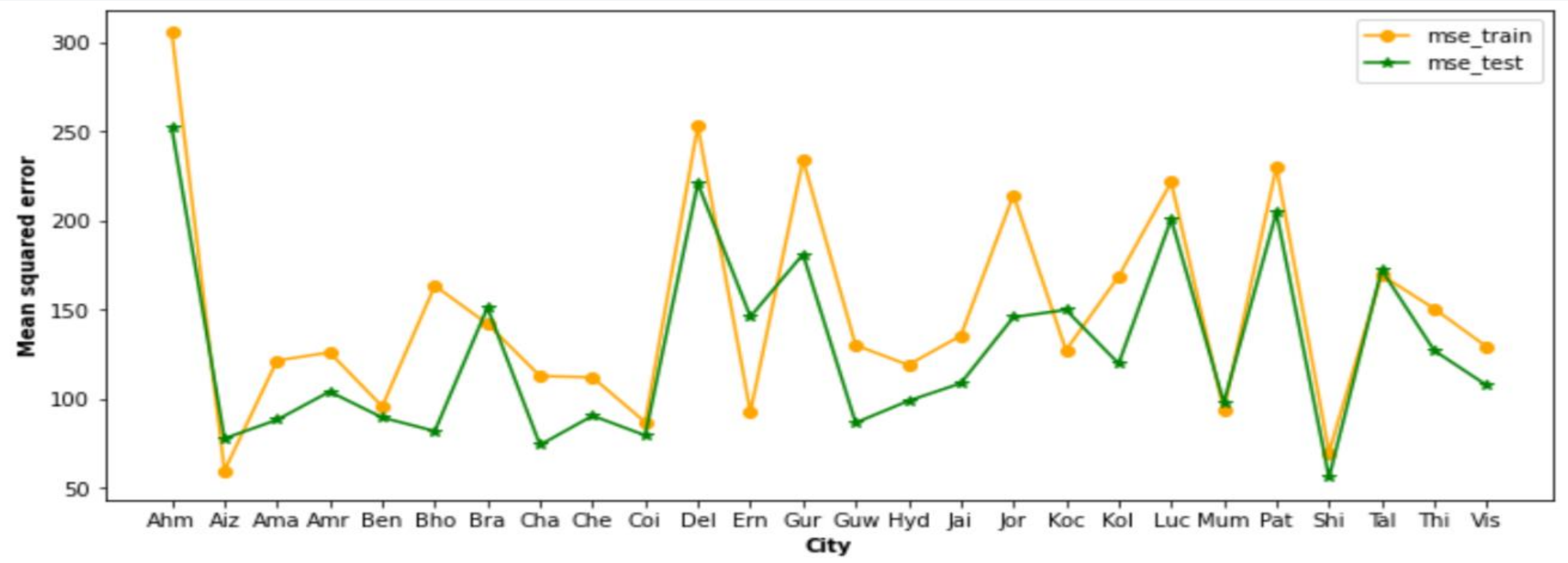
Mean Abs Error: 0.0033662200716981887
Mean Sq Error: 0.00011384331947930463
Root Mean Error: 0.010669738491608153

```
 n_estimators = [int(x) for x in np.linspace(start=100, stop=1200, num=12)]
 learning_rate = [0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6]
 max_depth = [int(x) for x in np.linspace(5, 30, num=6)]
 subsample = [0.7, 0.6, 0.8]
 min_child_weight = list(range(3, 8))
 objective = ['reg:squarederror']
 params = {
    'n_estimators': n_estimators,
    'learning_rate': learning_rate,
    'max_depth': max_depth,
    'subsample': subsample,
    'min_child_weight': min_child_weight,
    'objective': objective
 }

 search = RandomizedSearchCV(xgb, params,
scoring='neg_mean_squared_error',
                    cv=5, n_iter=100, random_state=43, n_jobs=-1,
verbose=True)
 search.fit(X,y)
 search.best_params_
 search.best_score_
 pred = search.predict(X_test)
 sns.distplot(y_test-pred)

 pred = search.predict(X_test)
 print(f"Mean Abs Error: {metrics.mean_absolute_error(y_test, pred)}")
 print(f"Mean Sq Error: {metrics.mean_squared_error(y_test, pred)}")
 print(f"Root Mean Error: {np.sqrt(metrics.mean_squared_error(y_test, pred))}")
```

(ii) Stacked LSTM

LSTMs are widely used for sequence prediction problem. The stacked LSTM model was capable of forecasting future days AQI for different cities on basis of past AQI information available.

Citywise Mean Squared error



Thank You !

[Github link for our project]

https://github.com/Haaabs/FTS-Air-Quality-Index-Prediction

[Drive link for our project]

https://drive.google.com/drive/folders/1F2tTiHf2wsl7PRcYZBMs1Qb6jrForROg

[References]

https://www.researchgate.net/publication/341990700_AIR_QUALITY_INDEX_FORECASTING_USING_HYBRID_NEURAL_NETWORK_MODEL_WITH_LSTM_ON_AQI