Air Pollution Prediction using Attribute Weighted k-NN

Data Collection and Merging

The London Air website [https://www.londonair.org.uk/LondonAir/Default.aspx] provided data for five years, from 2014 to 2018. This webpage offers data on London's air quality, as measured by the UK Air Quality Monitoring Network. *Barking and Dagenham - Rush Green* is the location being studied. Readings of various pollutants - Barometric Pressure, Nitric Oxide, Nitrogen Dioxide, Oxides of Nitrogen, Rainfall, Relative Humidity, Solar Radiation, Sulphur Dioxide, Temperature, Wind Direction, Wind Speed - were recorded at 15-minute interval daily starting on January 1, 2014, and ending on December 31, 2018, in the downloaded data. The following attributes are included in the downloaded data set:

- *Site*: The location where the reading is kept track of.
- *Species*: A variety of contaminants are measured.
- *ReadingDateTime*: The date and time at which the reading was taken.
- *Value*: This is the value of the species listed in the 'Species' column.
- *Units*: In the Species column, specify the units for each species.
- *Provisional or Ratified*: Provides information about whether the reading is provisional or ratified.

	Site	Species	ReadingDateTime	Value	Units	Provisional or Ratified
0	BG1	SOLR	01/01/2014 00:00	NaN	W/m2	P
1	BG1	SOLR	01/01/2014 00:15	0.0	W/m2	P
2	BG1	SOLR	01/01/2014 00:30	0.0	W/m2	P
3	BG1	SOLR	01/01/2014 00:45	0.0	W/m2	P
4	BG1	SOLR	01/01/2014 01:00	0.0	W/m2	P
...	...	...	...	...	...	...
210235	BG1	RHUM	31/12/2018 22:45	NaN	%	P
210236	BG1	RHUM	31/12/2018 23:00	NaN	%	P
210237	BG1	RHUM	31/12/2018 23:15	NaN	%	P
210238	BG1	RHUM	31/12/2018 23:30	NaN	%	P
210239	BG1	RHUM	31/12/2018 23:45	NaN	%	P

1928256 rows × 6 columns

Figure 1. Merged dataset

This 5-year data scattered into several different data sets, are merged into a single data set since more data yields better results. Since the columns *'Site', 'Units',* and *'Provisional or Ratified'* does not enhance the analysis, it is discarded. Now this long form of dataset contains three columns – *'ReadingDateTime', 'Species'*, and *'Value'*. After converting into wide form, the values of each species are present in individual columns with the name of the column representing the name of the species. The resultant dataset has the following features:

- *ReadingDateTime*: Represents the date and time at which reading is recorded.
- *BP*: Represents barometric pressure (mBar)
- *NO*: Represents concentration of NO (µg/m^3)
- *NO2*: Represents concentration of NO_2 (µg/m^3)
- *NOX*: Represents concentration of NOX (µg/m^3)
- *RAIN*: Represents rainfall (mm)
- *RHUM*: Represents relative humidity (%)
- *SO2*: Represents concentration of SO_2 (µg/m^3)
- *SOLR*: Represents solar radiation (W/m^2)

- *TMP*: Represents temperature (ºC)
- *WDIR*: Represents wind direction (ºN)
- *WSPD*: Represents wind speed (m/s)

```
1 dataframeA = dataframe.pivot(columns = 'Species', values = 'Value')
2 dataframeA
```

Species ReadingDateTime	BP	NO	NO2	NOX	RAIN	RHUM	SO2	SOLR	TMP	WDIR	WSPD
2014-01-01 00:00:00	NaN	NaN	NaN	NaN	0.0	NaN	NaN	NaN	NaN	NaN	NaN
2014-01-01 00:15:00	997.0	NaN	NaN	NaN	0.0	NaN	2.4	0.0	NaN	187.0	4.9
2014-01-01 00:30:00	997.0	NaN	NaN	NaN	0.0	NaN	2.4	0.0	NaN	192.0	3.5
2014-01-01 00:45:00	997.0	NaN	NaN	NaN	0.0	NaN	NaN	0.0	NaN	209.0	3.5
2014-01-01 01:00:00	997.0	NaN	NaN	NaN	0.0	NaN	NaN	0.0	NaN	203.0	3.2
...	...	...	...	...	...	...	...	...	...	...	...
2018-12-31 22:45:00	1029.0	NaN	NaN	NaN	0.0	NaN	-0.6	5.0	10.0	282.0	NaN
2018-12-31 23:00:00	1029.0	NaN	NaN	NaN	0.0	NaN	2.3	5.0	10.0	282.0	NaN
2018-12-31 23:15:00	1029.0	NaN	NaN	NaN	0.0	NaN	2.3	5.0	10.0	272.0	NaN
2018-12-31 23:30:00	1029.0	NaN	NaN	NaN	0.0	NaN	5.2	5.0	10.0	277.0	NaN
2018-12-31 23:45:00	1029.0	NaN	NaN	NaN	0.0	NaN	2.3	4.0	10.0	268.0	NaN

175296 rows × 11 columns

Figure 2. Dataset prepared for pre-processing

Data Pre-processing

Out of 175296 rows and 11 columns, rows containing all missing values (NaN) and columns having most of the entries missing are dropped from the dataset. Negative value reading of a pollutant is converted, as it means nothing in the real-world scenario, into a missing value reading. All the missing values then use Linear Interpolation. The equation for Linear Interpolation function is:

$$f(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x_1 - x_0)$$

where $(x_0, f(x_0)), (x_1, f(x_1))$ and $(x, f(x))$ are the three coordinates such that x lies in the interval (x_0, x_1).

Boxplots were made to detect outliers. Boxplots provide five-number summary – the length of the box is an interquartile range that indicates the distance between the first and third quartiles and gives light about the range that the middle half of the data covers. The ends of a box are quartiles.

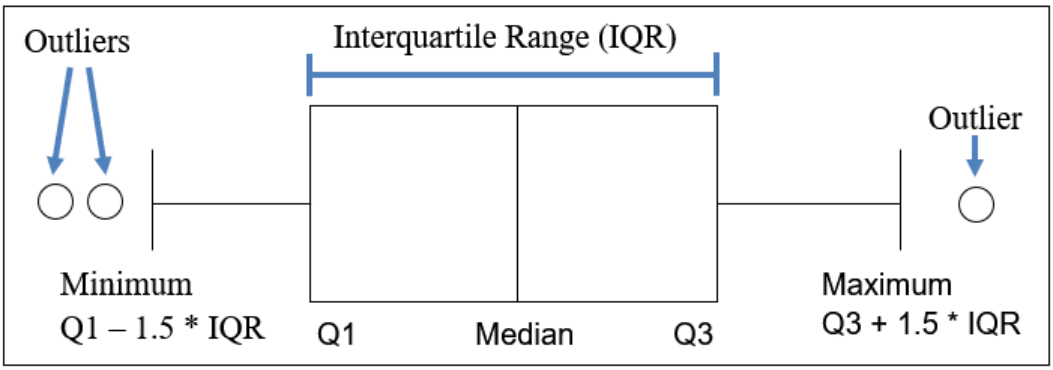

Figure 3. Boxplot explained

The interquartile range (IQR) is defined as:

$$IQR = Q_3 - Q_1,$$

where Q_3 and Q_1 are the 75th and the 25th percentiles, respectively. Values that are lying at least 1.5 * IQR above Q_3 and at least 1.5 * IQR below Q_1 are the outliers. They are converted into a missing value and again linearly interpolated to fill in missing values in the dataset.

Heatmap was plotted to find the correlation between 9 columns, and it was discovered that attribute *'RAIN'* did not correlate with any other attribute giving correlation coefficient value as NaN. The reason being all the 169284 rows of *'RAIN'* was filled with zero values. Hence, column *'RAIN'* is dropped leaving us with 8 attributes to analyse on.

Different Algorithms of k-NN Model Considered

After data pre-processing steps, the previous five readings of NO, NO2 and NOX are added as individual columns in the data set before proceeding with the model training. The value of *k* is chosen to be 11. The dataset is split into training data and testing data in the ratio of 7:3. The testing dataset is split into batches of 500 instances and likewise 102 iterations will be performed.

Simple k-NN

k-NN aims to find the k neighbors that are most like an instance, using the Euclidean distance. Each instance of the training set is checked to find the nearest k instances for a particular testing instance. Suppose $u_i \; and \; v_j$ are the training and test instances, respectively; m be the total number of predictor variables, and $u_{ip} \; and \; v_{jp}$ are the pth attribute of training and testing instances, respectively. Then the distance between $u_i \; and \; v_j$ is calculated as:

$$d(u_i, v_j) = \sqrt{\sum_{p=1}^{m}(u_{ip} - v_{jp})^2}$$

After finding the k nearest neighbors, the target variable of a testing instance is calculated by taking the mean of the target values of all the neighbors.

$$y_j = \frac{\sum_{i=1}^{k} y_i}{k}$$

where y_j is the target variable of the testing instance, k is the number of nearest neighbors and y_i is the target value of the ith nearest neighbor.

Distance Weighted k-NN

Instead of taking the simple mean of all the k nearest neighbors, the weighted mean of all is taken by giving distance-based weights to all of them. Higher weights are given to closer neighbors as they should influence the results maximum.

$$y_j = \frac{\sum_{i=1}^{k} w_i y_i}{\sum_{i=1}^{k} w_i}$$

where y_j is the target variable of the testing instance, k is the number of nearest neighbors, y_i is the target value of the i[th] nearest neighbor, and w_i is the weight given to y_i. w_i is calculated in two ways:

i. $w_i = \{\frac{1}{d(u_i, v_j)}, \quad if\ d(u_i, v_j) \neq 0\ 1$,

ii. $w_i = \frac{1.0}{d(u_i, v_j)+0.001}, if\ d(u_i, v_j) = 0$

Pearson Correlation Coefficient Based Attribute Weighted k-NN

k-NN uses all the attributes for prediction giving equal weightage. If there are too many attributes, then complexity gets increased, and all the predictors may not be appropriate for the prediction. Attributes are weighted differently while finding the distance between training and testing instances in Attribute Weighted k-NN. This eliminates the dominating effect of irrelevant attributes while measuring the distance between instances. The least relevant attributes can be eliminated by setting their weights to zero.

Attribute weighted k-NN with attribute weights equal to Pearson Correlation Coefficient values and weights are computed using the proposed formula are applied to achieve better performance than simple k-NN and Distance Weighted k-NN. The distance formula used for weighing attributes in Attribute Weighted k-NN is:

I. $$d(u_i, v_j) = \sqrt{\sum_{p=1}^{m} \left| w_p (u_{ip} - v_{jp}) \right|^2}$$

II. $$d(u_i, v_j) = \sqrt{\sum_{p=1}^{m} (w_p u_{ip} - w_p v_{jp})^2}$$

where w_p is the weight given to p[th] attribute, u_i and v_j are the training and test instances, m is the total number of predictors, and u_{ip} and v_{jp} are the p[th] attribute of training and testing instances, respectively.

In (II), the attributes are first multiplied with the corresponding attribute weights and then the difference between them is computed instead of taking the difference between the attributes first and then multiplying them with the corresponding attribute weights as in (I).

Weights to the attributes are given in two ways:

i. Using Pearson Correlation Coefficient values:

$$\boxed{w_p = \left| r_{x_p,y} \right| = \left| \frac{\sum_{i=1}^{n} (x_{ip} - \underline{x_p})(y_i - \underline{y})}{n \sigma_x \sigma_y} \right|}$$

where $r_{x_p,y}$ is the correlation between attributes x_p and y and its value ranges from -1 to 1, n is the tuple count, x_{ip} and y_i are the respective values of x_p and y for the tuple i and $\underline{x_p}$ and $\underline{y}$ are the respective mean values of x_p and y, σ_x and σ_y are the respective standard deviations of x_p and y.

ii. Using the following proposed formula for weights computation:

$$\boxed{w_p = \frac{\left| r_{x_p,y} \right|}{1 - \left| r_{x_p,y} \right|}}$$

If $\left| r_{x_p,y} \right|$ is close to 1, the 1 - $\left| r_{x_p,y} \right|$ will be close to zero and when $\left| r_{x_p,y} \right|$ is divided by this term, w_p increases. In this way, the weights of highly

correlated attributes are increased even further and will lead to the right selection of the neighbors.

Information Gain Based Attribute Weighted k-NN

Information Gain measures the quality of a feature through the difference of entropy given the value of this feature. The bigger the difference of entropy that the feature brings, the more useful this feature to the discrimination among classes is, since entropy represents the degree of mix of data from different classes.

In a dataset X, let $\{C_i\}_{i=1}^{C}$ denote the set of categories in target space, and $\{A_j\}_{j=1}^{C_A}$ denote the set of values in feature A. The information gain IG(A) of feature A is computed as,

$$\boxed{\text{IG(A)} = \text{E(X)} - \text{E(X|A)}}$$

where E(X) denoted the entropy of dataset X while E(X|A) denotes the entropy of dataset X given the value of feature A. The computation of E(X) and E(X|A) is defined as:

$$E(X) = -\sum_{i=1}^{c} p(C_i) * (p(C_i))$$

$$E(X|A) = -\sum_{j=1}^{C_A}\left(p(A_j) * \sum_{i=1}^{c}\left(p(C_i|A_j) * (p(C_i|A_j))\right)\right)$$

where p(C_i) denoted probability of category C_i in the dataset X, p(A_j) shows the probability of feature A with value V_j in the dataset X, and p($C_i|A_j$) depicts the probability of category C_i given the feature A with value v_j. The computation of p(C_i) is

$$p(C_i) = \frac{num(I(C_i))}{N}$$

among which, $\{C_i\}_{i=1}^{C}$ denotes the classes in dataset X = $\{X_1, X_2, X_3, \ldots, X_N\}$, I($C_i$) denotes the instances which belong to the class C_i, num(.) counts the number of instances in (.).

To compute information gain of each attribute in our dataset, the target columns {'NO', 'NO2', 'NOX'} are first processed using z-score normalization. Z-score normalization refers to the process of normalizing every value in a dataset such that the mean of all the values is 0 and the standard deviation is 1. Z-score values fall in the range of [-3, 3].

$$z - score = \frac{x - \mu}{\sigma}$$

The target column is then discretized using equal-width binning method. In binning, the original data values are divided into small intervals known as bins and then they are replaced by a general value calculated for that bin. This has a smoothing effect on the input data. In equal-width binning, bins have equal width with a range of each bin are defined as [min + w], [min + 2w] …. [min + nw] where

$$w = (max - min) / (no\ of\ bin)$$

The width (w) is set to the value of one. After discretization, values in the target column are:

i. Left unchanged.
ii. Substituted with median value of that bin.

Information Gain acts as the weight for the model, but the distance formula is as used in (3).