Capstone Project

Machine Learning Engineer Nanodegree

Title: Air Quality

M.Ravindra Naik

February 27th, 2019

I. Definition

Project Overview:

In early 2011, officials reported that pollution in Italy was reaching crisis levels. What's particularly troublesome is particle pollution that pervades Italy, and accounts for breathing and heart problems, causing a whopping 9% of deaths of Italians over the age of 30. New report finds that air pollution is the single biggest environmental health risk in Europe, causing hundreds of thousands of premature deaths. Particular matter, ozone, nitrogen dioxide. Europe’s air quality is significantly threatened by these pollutants, mostly in urban centers

Air quality is a significant concern for both healthy population and people suffering for different pathologies. Long or even short term exposure to significant pollution levels have been associated with the development or worsening of multiple pathologies ranging from Asthma to Lung Cancer. Air quality patterns may significantly vary in space and time due to complex fluid dynamic effects occurring in the city landscape or to the hourly, daily and seasonal variation of human activities. However, most of the national states rely on the operation of networks of certified air quality monitoring stations in order to detect and monitor air quality in cities. Unfortunately, the average low spatial density of such networks do not permit to achieve the required resolution.

Reference link:
https://www.researchgate.net/publication/319338229_Cooperative_Air_Quality_Sensing_with_
Crowdfunded_Mobile_Chemical_Multisensor_Devices

Problem Statement:

To check the quality of air using ‘Air Quality Chemical Multisensor Device’ by finding the R-Squared score(Co-Efficient) of regression using different regression models and the best model is selected based on the highest R-Squared score to evaluate the Air Quality.

For this Regression problem I am going to use three models are

1.Linear Regression

2.Lasso Regression

3.Decicion Tree Regression

And the r2 score of these models is like:

Algorithm	R2 score
Linear Regression	0.99838
Lasso Regression	0.99930
Decision Tree Regression	0.9999

Here the steps involved for this program is :

1.Loading the data by using read-csv() method

2.Visualizing the data by using matlabplot library

3.Preprocessing the data means cleaning the data (removing missing values or replacing the missing values with nan)The data set didn’t contain any missing values

4.Applying the models and finding the r2 score

5.Comparing the r2 score of the three models and finding the best model from the three models

Metrics:

R^2 Score:

R-squared is a statistical measure that’s used to assess the goodness of fit of our regression model. In R-squared we have a baseline model which is the worst model. This baseline model doesn’t make use of any independent variables to predict the value of dependent variable Y. Instead it uses the mean of the observed responses of dependent variable Y and always predicts this mean as the value of Y.

R-squared is given by

$$R^2 = 1 - \frac{SSE}{SST}$$

Where SSE is the sum of squared errors of our regression model

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

And SST is the sum of squared errors of our baseline model.

$$SST = \sum_{i=1}^{n} (y_i - \bar{y}_i)^2.$$

II. Analysis:

Data Exploration:

Dataset Link: https://archive.ics.uci.edu/ml/datasets/Air+Quality

In this project I have used 15 attributes and around 9300 trained and test data to evaluate the Rsquared score. And this coefficient is found out by using different Regression Methods. The dataset used are shown in the below.

```
In [3]: display(air_data)
```

	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
0	2004-03-10	18:00:00	2.6	1360.000000	150	11.881723	1045.500000	166.0	1056.250000	113.0	1692.000000	1267.500000	13.600000
1	2004-03-10	19:00:00	2.0	1292.250000	112	9.397165	954.750000	103.0	1173.750000	92.0	1558.750000	972.250000	13.300000
2	2004-03-10	20:00:00	2.2	1402.000000	88	8.997817	939.250000	131.0	1140.000000	114.0	1554.500000	1074.000000	11.900000
3	2004-03-10	21:00:00	2.2	1375.500000	80	9.228796	948.250000	172.0	1092.000000	122.0	1583.750000	1203.250000	11.000000
4	2004-03-10	22:00:00	1.6	1272.250000	51	6.518224	835.500000	131.0	1205.000000	116.0	1490.000000	1110.000000	11.150000
5	2004-03-10	23:00:00	1.2	1197.000000	38	4.741012	750.250000	89.0	1336.500000	96.0	1393.000000	949.250000	11.175000
6	2004-03-11	00:00:00	1.2	1185.000000	31	3.624399	689.500000	62.0	1461.750000	77.0	1332.750000	732.500000	11.325000
7	2004-03-11	01:00:00	1.0	1136.250000	31	3.326677	672.000000	62.0	1453.250000	76.0	1332.750000	729.500000	10.675000
8	2004-03-11	02:00:00	0.9	1094.000000	24	2.339416	608.500000	45.0	1579.000000	60.0	1276.000000	619.500000	10.650000
9	2004-03-11	03:00:00	0.6	1009.750000	19	1.696658	560.750000	-200.0	1705.000000	-200.0	1234.750000	501.250000	10.250000
10	2004-03-11	04:00:00	-200.0	1011.000000	14	1.293620	526.750000	21.0	1817.500000	34.0	1196.750000	445.250000	10.075000

ne	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	AH
00	2.6	1360.000000	150	11.881723	1045.500000	166.0	1056.250000	113.0	1692.000000	1267.500000	13.600000	48.875001	0.757754
00	2.0	1292.250000	112	9.397165	954.750000	103.0	1173.750000	92.0	1558.750000	972.250000	13.300000	47.700000	0.725487
00	2.2	1402.000000	88	8.997817	939.250000	131.0	1140.000000	114.0	1554.500000	1074.000000	11.900000	53.975000	0.750239
00	2.2	1375.500000	80	9.228796	948.250000	172.0	1092.000000	122.0	1583.750000	1203.250000	11.000000	60.000000	0.786713
00	1.6	1272.250000	51	6.518224	835.500000	131.0	1205.000000	116.0	1490.000000	1110.000000	11.150000	59.575001	0.788794
00	1.2	1197.000000	38	4.741012	750.250000	89.0	1336.500000	96.0	1393.000000	949.250000	11.175000	59.175000	0.784772
00	1.2	1185.000000	31	3.624399	689.500000	62.0	1461.750000	77.0	1332.750000	732.500000	11.325000	56.775000	0.760312
00	1.0	1136.250000	31	3.326677	672.000000	62.0	1453.250000	76.0	1332.750000	729.500000	10.675000	60.000000	0.770238
00	0.9	1094.000000	24	2.339416	608.500000	45.0	1579.000000	60.0	1276.000000	619.500000	10.650000	59.674999	0.764819
00	0.6	1009.750000	19	1.696658	560.750000	-200.0	1705.000000	-200.0	1234.750000	501.250000	10.250000	60.200001	0.751657

The 15 attributes that I have used in dataset are explained below.

Attributes:

1.Date - DD/MM/YY

2.Time - HH.MM.SS

3.CO(GT) - True hourly averaged concentration CO in mg/m^3
4.PT08.S1(CO) - PT08.S1 (tin oxide) hourly averaged sensor response (nominally CO targeted)

5. NMHC(GT) - True hourly averaged overall Non Metanic Hydrocarbons concentration in microg/m^3

6. C6H6(GT) - True hourly averaged Benzene concentration in microg/m^3

7. PT08.S2(NMHC) - PT08.S2 (titania) hourly averaged sensor response (nominally NMHC targeted)

8. NOx(GT) - True hourly averaged NOx concentration in ppb

9. PT08.S3(NOx) - PT08.S3 (tungsten oxide) hourly averaged sensor response

10. NO2(GT) - True hourly averaged NO2 concentration in microg/m^3

11. PT08.S4(NO2) - PT08.S4 (tungsten oxide) hourly averaged sensor response

12. PT08.S5(O3) - PT08.S5 (indium oxide) hourly averaged

sensor response (nominally O3 targeted)

13.T - Temperature in Â°C 14.RH - Relative Humidity (%) 15.AH - Absolute Humidity

Data Set Information:

The dataset contains 9358 instances of hourly averaged responses from an array of 5 metal oxide chemical sensors embedded in an Air Quality Chemical Multisensor Device. The device was located on the field in a significantly polluted area, at road level, within an Italian city. Data were recorded from March 2004 to February 2005 (one year)representing the longest freely available recordings of on field deployed air quality chemical sensor devices responses. Ground Truth hourly averaged concentrations for CO, Non Metanic Hydrocarbons, Benzene, Total Nitrogen Oxides (NOx) and Nitrogen Dioxide (NO2) and were provided by a co-located reference certified analyzer. Evidences of cross-sensitivities as well as both concept and sensor drifts are present as described in De Vito et al., Sens. And Act. B, Vol. 129,2,2008 (citation required) eventually affecting sensors concentration estimation capabilities. Missing values are tagged with -200 values. This dataset can be used exclusively for research purposes. Commercial purposes are fully excluded.

Description of Data set:

In [4]:
```
# Display a desription of total data set
display(air_data.describe())
```

	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
count	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000
mean	-34.207524	1048.869652	-159.090093	1.865576	894.475963	168.604200	794.872333	58.135898	1391.363266	974.951534	9.776600
std	77.657170	329.817015	139.789093	41.380154	342.315902	257.424561	321.977031	126.931428	467.192382	456.922728	43.203438
min	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000
25%	0.600000	921.000000	-200.000000	4.004958	711.000000	50.000000	637.000000	53.000000	1184.750000	699.750000	10.950000
50%	1.500000	1052.500000	-200.000000	7.886653	894.500000	141.000000	794.250000	96.000000	1445.500000	942.000000	17.200000
75%	2.600000	1221.250000	-200.000000	13.636091	1104.750000	284.200000	960.250000	133.000000	1662.000000	1255.250000	24.075000
max	11.900000	2039.750000	1189.000000	63.741476	2214.000000	1479.000000	2682.750000	339.700000	2775.000000	2522.750000	44.600000

Information of Dataset:

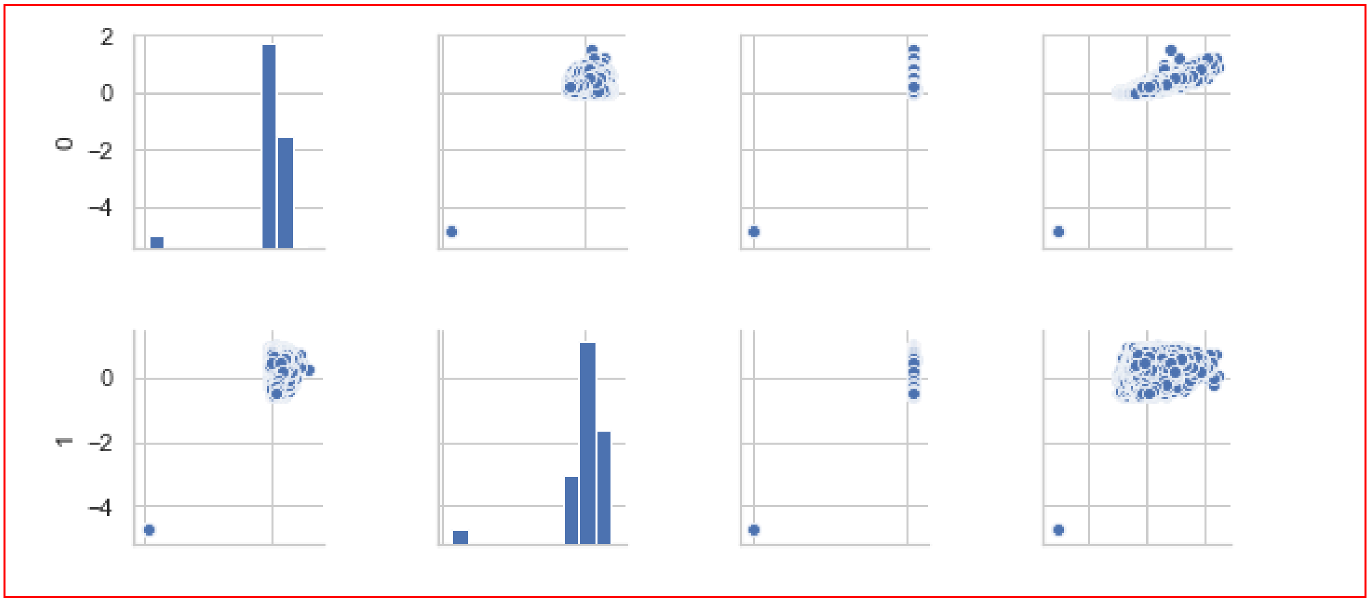
```
In [5]: #Information about the Data
        air_data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9357 entries, 0 to 9356
Data columns (total 15 columns):
Date             9357 non-null datetime64[ns]
Time             9357 non-null object
CO(GT)           9357 non-null float64
PT08.S1(CO)      9357 non-null float64
NMHC(GT)         9357 non-null int64
C6H6(GT)         9357 non-null float64
PT08.S2(NMHC)    9357 non-null float64
NOx(GT)          9357 non-null float64
PT08.S3(NOx)     9357 non-null float64
NO2(GT)          9357 non-null float64
PT08.S4(NO2)     9357 non-null float64
PT08.S5(O3)      9357 non-null float64
T                9357 non-null float64
RH               9357 non-null float64
AH               9357 non-null float64
dtypes: datetime64[ns](1), float64(12), int64(1), object(1)
memory usage: 1.1+ MB
```

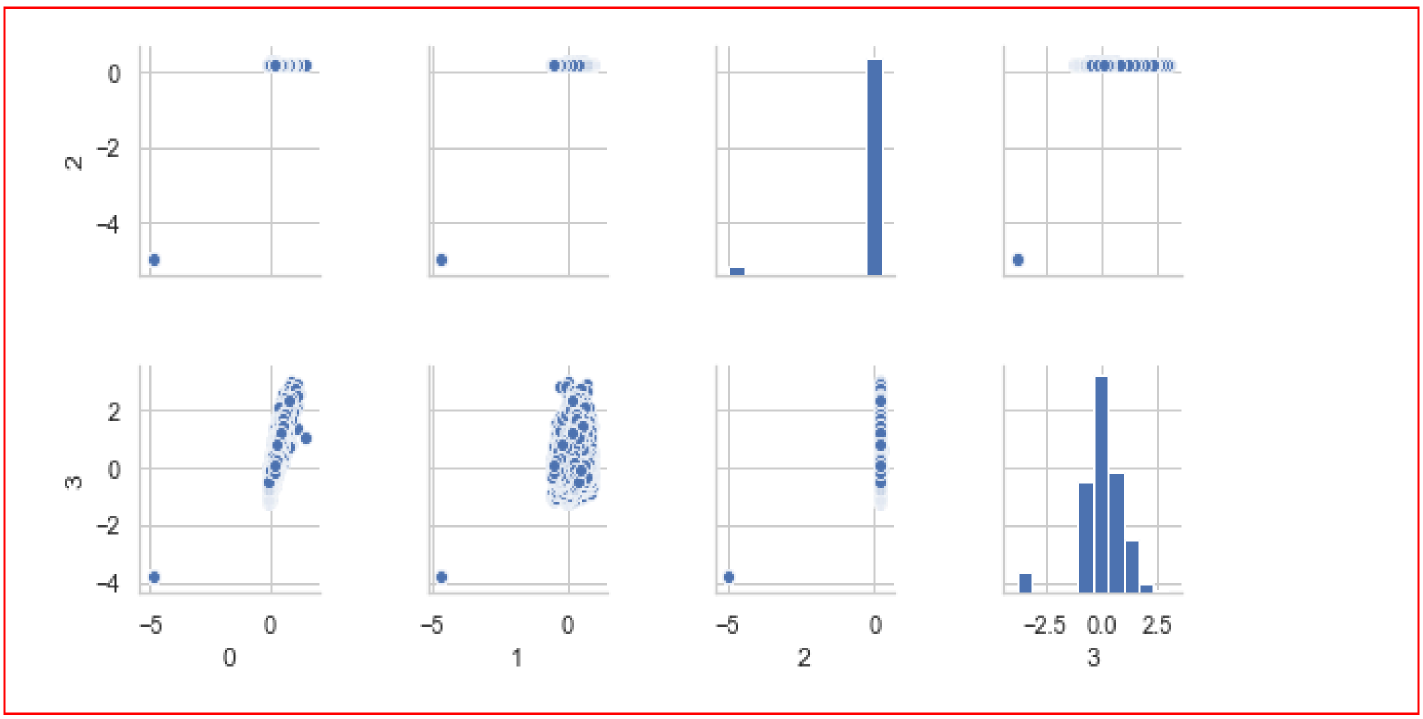
There are no missing values in this dataset, so we don't need to modify any data.

- There are no missing values existing in the data set
- so we don't need to modify any data.

Data Visualization:

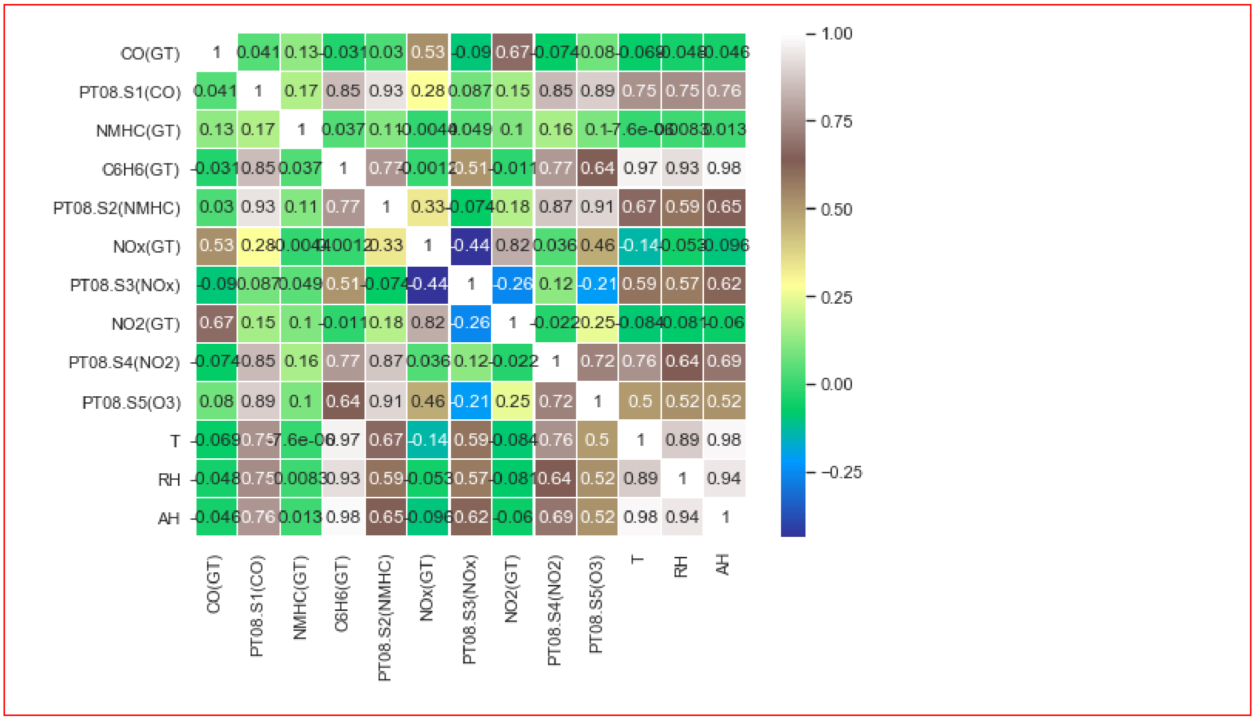
Let us visualize the absolute correlation coefficient of target variable with all the other variables. Higher absolute correlation coefficient means the variable can provide more information about how the target variable moves as shown in below figure.


0
1
2
0
−2
−4


2
3
2
0
−2
−4
−5
0
−2.5
0.0
2.5
0
1
2
3

Heat Map:

The heat map is a 2-D representation of data in which values are represented by colors. A simple heat map provides the immediate visual summary of information. More elaborate heat maps allow the user to understand complex data.



Seeing above heat map, I infer that none of displayed value pairs is having an explicitly high correlation, so there is no necessity to ditch any feature at this stage. I also notice a negative correlation between ‘PT08.S3(NOx)' and ‘NOx(GT)’, PT08.S3(NOx)’ and ‘NO2(GT)’, ‘PT08.S3(NOx)' and ‘PT08.S5(O3)’ . And there also exist some of negative correlation values relatively less than the mentioned above.

Algorithms and Techniques:

1. Linear Regression
2. Lasso Regression
3. Decision Tree Regression

Regression analysis is a form of predictive modeling technique which investigates the relationship between a dependent (target) and independent variable (s) (predictor). This

Technique is used for forecasting, time series modeling and finding the causal effect relationship between the variables. Regression analysis is an important tool for modeling and analyzing data. Here, we fit a curve / line to the data points, in such a manner that the differences between the distances of data points from the curve or line is minimized.

1. Linear Regression

It is one of the most widely known modeling technique. Linear regression is usually among the first few topics which people pick while learning predictive modeling. In this technique, the dependent variable is continuous, independent variable(s) can be continuous or discrete, and nature of regression line is linear.

R-Squared score is evaluated using this algorithm with regressor.score(X_test, y_test).

Linear Regression establishes a relationship between dependent variable (Y) and one or more independent variables (X) using a best fit straight line (also known as regression line).

It is represented by an equation Y=a+b*X + e, where a is intercept, b is slope of the line and e is error term. This equation can be used to predict the value of target variable based on given predictor variable(s).

• There must be linear relationship between independent and dependent variables • Multiple regression suffers from multicollinearity, autocorrelation, heteroskedasticity. • Linear Regression is very sensitive to Outliers. It can terribly affect the regression line and eventually the forecasted values.

2. Lasso Regression

Lasso (Least Absolute Shrinkage and Selection Operator) also penalizes the absolute size of the regression coefficients. In addition, it is capable of reducing the variability and improving the accuracy of linear regression models. Look at the equation below:

$$= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_1}_{\text{Penalty}}$$

Lasso regression differs from ridge regression in a way that it uses absolute values in the penalty function, instead of squares. This leads to penalizing (or equivalently constraining the sum of the absolute values of the estimates) values which causes some of the parameter estimates to turn out exactly zero. Larger the penalty applied, further the estimates get shrunk towards absolute zero. This results to variable selection out of given n variables.

R-Squared score is evaluated using this algorithm with indiana_jones.score(X_test, y_test).

• The assumptions of this regression is same as least squared regression except normality is not to be assumed • It shrinks coefficients to zero (exactly zero), which certainly helps in feature selection • This is a regularization method and uses l1 regularization • If group of predictors are highly correlated, lasso picks only one of them and shrinks the others to zero

3.Decision Tree Regression :

A decision tree is a flow-chart-like structure, where each internal (non-leaf) node denotes a test on an attribute, each branch represents the outcome of a test, and each leaf (or terminal) node holds a class label. The topmost node in a tree is the root node.
Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node (e.g., Outlook) has two or more branches (e.g., Sunny, Overcast and Rainy), each representing values for the attribute tested. Leaf node (e.g., Hours Played) represents a decision on the numerical target. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical

and numerical data.   R-Squared score is evaluated using this algorithm with dtr.score(X_test, y_test)

Types of decision tree is based on the type of target variable we have. It can be of two types:

1. Categorical Variable Decision Tree: Decision Tree which has categorical target variable then it called as categorical variable decision tree. Example:- In above scenario of student problem, where the target variable was “Student will play cricket or not” i.e. YES or NO. 2. Continuous Variable Decision Tree: Decision Tree has continuous target variable then it is called as Continuous Variable Decision Tree

Benchmark Model:

Here we compare the final model with the remaining models to see if it got better or same or worse. The R^2 score is compared among the models and the best model is selected. I think Linear Regression model can be set as the benchmark model and I’m sure that the final solution would outperform the Benchmark model.

III. Methodology

Pre-processing:

In this step we will pre-process the data. Data pre-processing is considered to be the first and foremost step that is to be done before starting any process. We will read the data by using read_excel. Then we will know the shape of the data. And by using the info() we will know the information of the attributes. From that we came to know whether there exists any missing values. After that we will divide the whole data into training and testing data. We will assign 70% of the data to the training data and the remaining 30% of the data into testing data. We will do this by using train_test_split from sklearn.model_selection.

- As mentioned as in the above section there is no missing values in the data set.So no need to perform any operations to remove null values or missing values
- I am not using the any preprocessing log functions in the program because their no missing values
- So there is no preprocessing errors here so no need to handle outliers

Implementation:

Algorithm	R2 score
Linear Regression	0.99838
Lasso Regression	0.99930
Decision Tree Regression	0.9999

Out of the chosen algorithms we will start with Linear Regression model. We will take a classifier and fit the training data. After that we will predict that by using predict (X_test). Now we will predict the regression score of the testing data by using regressor.score(X_test, y_test). By doing so for, the Linear Regression will give us the R-squared score of 0.999384. We will continue the same procedure on Decision tree Regression, Lasso Regression . By following the same procedure above that is fitting, predicting and finding the R-Squared score as bellow.

Step 4.1 Linear Regression ¶

It is one of the most widely known modeling technique. Linear regression is usually among the first few topics which people pick while learning predictive modeling. In this technique, the dependent variable is continuous, independent variable(s) can be continuous or discrete, and nature of regression line is linear.

```
17]: from sklearn.linear_model import LinearRegression
```

```
18]: regressor = LinearRegression()
     regressor.fit(X_train, y_train)
```

```
18]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
              normalize=False)
```

```
19]: print("Predicted values:", regressor.predict(X_test))

     Predicted values: [  10.80465729   12.883162     20.52891883 ... -200.32873909   20.37164881
        7.20007983]
```

```
20]: print("R^2 score for liner regression: ", regressor.score(X_train, y_train))

     R^2 score for liner regression:  0.9991558434493987
```

```
21]: print("R^2 score for liner regression: ", regressor.score(X_test, y_test))

     R^2 score for liner regression:  0.9993767289506726
```

```
[26]: from sklearn.linear_model import Lasso
```

```
[27]: indiana_jones = Lasso(alpha=1.0)
      indiana_jones.fit(X_train, y_train)
```

```
t[27]: Lasso(alpha=1.0, copy_X=True, fit_intercept=True, max_iter=1000,
          normalize=False, positive=False, precompute=False, random_state=None,
          selection='cyclic', tol=0.0001, warm_start=False)
```

```
[28]: print("Coefficient of determination R^2 <-- on train set : {}".format(indiana_jones.score(X_train, y_train)))

      Coefficient of determination R^2 <-- on train set : 0.9990601619875135
```

```
[29]: print("Coefficient of determination R^2 <-- on test set: {}".format(indiana_jones.score(X_test, y_test)))

      Coefficient of determination R^2 <-- on test set: 0.9992931687782275
```

Here there are no missing values so no need to preprocccessing the data

By using train_test_split method split the data for training and testing

Here by using the sklearn library importing the required libraries for the three models linear regression, Lasso Regression and decision tree

By using the score()method finding the r2 score for the three models

R-Squared Score

Linear Regression 0.9993846

Lasso Regression 0.9993083

Decision Tree Regression 0.9999983

From the above reports Decision tree Regression seems to be performing well.

Refinement:

I found out ‘Decision Tree Regression’ as the best regression model out of the chosen techniques. Decision Tree Regression’s R-Squared score is almost 100% so it indicates that the model explains all the variability of the response data around its mean.

```
In [31]: etr = ExtraTreesRegressor(n_estimators=300)
         etr.fit(X_train, y_train)

Out[31]: ExtraTreesRegressor(bootstrap=False, criterion='mse', max_depth=None,
                    max_features='auto', max_leaf_nodes=None,
                    min_impurity_decrease=0.0, min_impurity_split=None,
                    min_samples_leaf=1, min_samples_split=2,
                    min_weight_fraction_leaf=0.0, n_estimators=300, n_jobs=None,
                    oob_score=False, random_state=None, verbose=0, warm_start=False)

In [32]: print(etr.feature_importances_)
         indecis = np.argsort(etr.feature_importances_)[::-1]

         [1.70513820e-04 9.62949347e-02 4.73063539e-06 7.14764818e-02
          4.71354843e-04 5.00312568e-02 1.24781446e-04 6.68416973e-02
          2.12464007e-01 2.41419206e-01 2.60701036e-01]

In [33]: print("Coefficient of determination R^2 <-- on test set: {}".format(etr.score(X_test, y_test)))

         Coefficient of determination R^2 <-- on test set: 0.9999994318624236
```

An extra-trees regressor has been used .This class implements a meta estimator that fits a number of randomized decision trees (a.k.a. extra-trees) on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

R-Squared score based on extra trees Regressor will be more accurate as n_estimators has been used and obtained result is 0.99999691

Complications: High air pollution levels can cause immediate health problems including:

• Aggravated cardiovascular and respiratory illness • Added stress to heart and lungs, which must work harder to supply the body with oxygen • Damaged cells in the respiratory system

Long-term exposure to polluted air can have permanent health effects such as:

• Accelerated aging of the lungs

• Loss of lung capacity and decreased lung function

• Development of diseases such as asthma, bronchitis, emphysema, and possibly cancer

• Shortened life span

Other long-term complication Skin is the body's first line of defense against a foreign pathogen or infectious agent and it is the first organ that may be contaminated by a pollutant. The skin is a target organ for pollution in which the absorption of environmental pollutants from this organ is equivalent to the respiratory uptake. Research on the skin has provided evidence that traffic-related air pollutants, especially PAHs, VOCs, oxides, and PM affect skin aging and cause pigmented spots on the face.

IV. Result

Model evaluation and validation

The final model we have chosen is Decision Tree Regression which gave us more R Squared score that is 0.9999983. Here we can say that the solution is reasonable because we are getting much less R-Squared value while using other models but relatively small change in Linear Regression and Lasso Regression. So the results found from this model can be trusted.

Justification:

My final model's solution is better than the benchmark model.

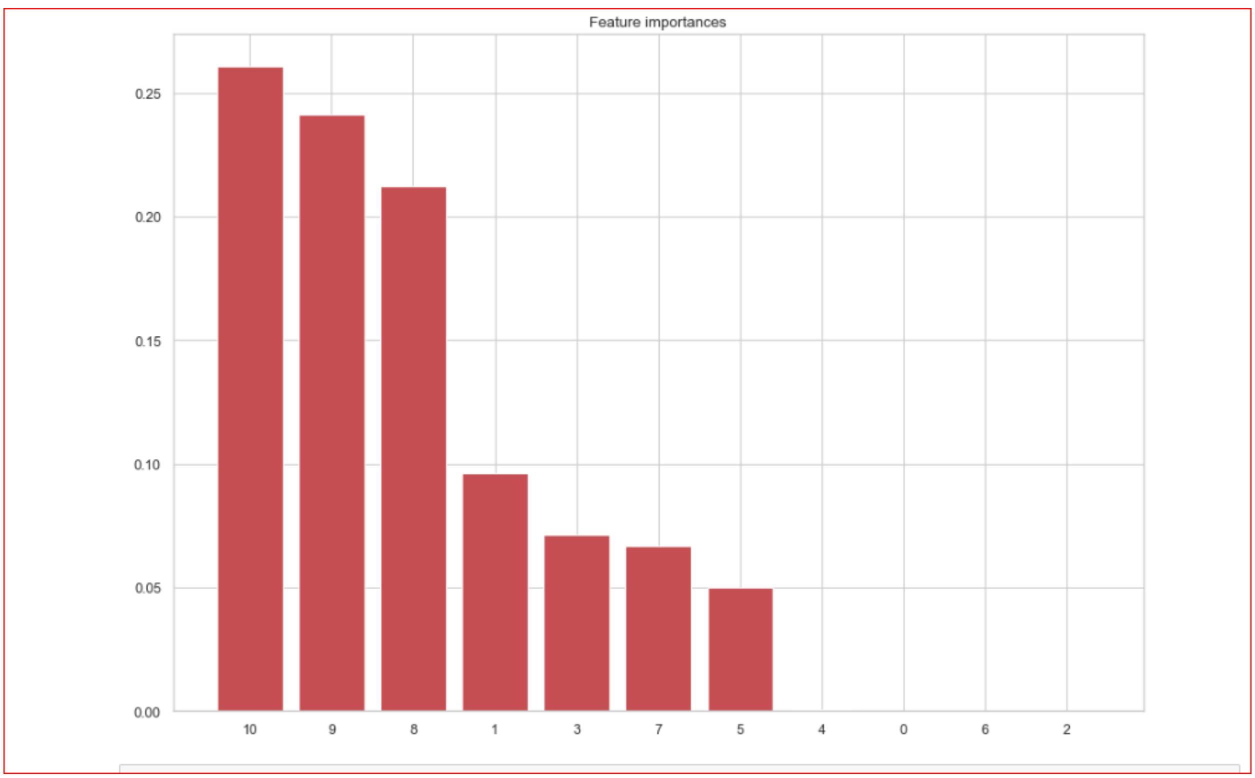
	Decision Tree	Bench mark model
R2 score	0.99999691	0.999384

From the above table we can clearly see that R-Squared score of decision tree regression is greater than R-Squared score of Linear

Regression (Benchmark Model), so we can conclude that the results for the final model are stronger than the benchmark model. Hence we can say that the decision tree regression provides the significant to solve the problem of predicting Air Quality.

V. Conclusion:

The goal of the project was to compare different machine learning algorithms and predict whether the gases present in air is responsible for human health issues and if any harmful gases exists and what measures should be taken. Here are the final results.



Reflection:

1. I have learnt how to visualize and understand the data.

2. I have learnt that the data cleaning place a very vital role in data analytics.

3. Removing the data features which are not necessary in evaluating model is very important.

4. I got to know how to use the best technique for the data using appropriate ways

5. I got to know how to tune the parameters in order to achieve the best score.

6. On a whole I learnt how to graph a dataset and applying cleaning techniques on it and to fit the best techniques to get best score.

Improvement:

The process which I have followed can be improved to describe a cooperative air quality sensor architecture based on crowd funded, mobile, electrochemical sensor based, and monitoring systems. The platform aims to produce enhanced information on personal pollutant exposure and enable cooperative reconstruction of high resolution pictures of air pollution in the urban landscape. The calibrated devices are connected to smart phones that provide georeferenced visualization of personal exposure and session based log capabilities. A cloud based interface provides a sensor fusion based mapping capability exploiting Google maps APIs. An in-lab calibration by linear regression with temperature correction has been computed and preliminary results have been reported. A small set of calibrated devices will be shipped to crowd funders for extended field tests in different Italian cities.