Air Quality Index Prediction

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering/Information Technology

By

Rishabh Goyal (161343)

Under the supervision of

Dr. Hemraj Saini
&
Dr. Geetanjali
to

Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

Candidate's Declaration

I hereby declare that the work presented in this report entitled **"Air Quality Index Prediction"** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August 2019 to May 2020 under the supervision of **Dr. Hemraj Saini & Dr. Geetanjali.**

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

R. Goyal

Rishabh Goyal, 161343

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr Hemraj Saini

Associate Professor

Department of Computer Science & Engineering and Information Technology

Geetanjali

Dr Geetanjali

Assistant Professor

Department of Computer Science & Engineering and Information Technology

Acknowledgment

I would like to express my deepest appreciation to all those who provided us the possibility to complete this report. A special gratitude we give to our final year project supervisor, Dr. Geetanjali and Dr. Hemraj Saini whose contribution in stimulating suggestions and encouragement, helped me and my partner to coordinate our project well especially in writing this report.

Furthermore we would also like to acknowledge with much appreciation the crucial role of Jaypee University of Information Technology, who gave the permission to use all required equipment and the necessary materials to complete the project A framework on automated trade system using time-series data and machine learning classifiers .A special thanks goes to my supervisor, Dr. Geetanjali and Dr. Hemraj Saini who help me to assemble the parts and gave suggestion about the project "Algorithmic Trading" he have invested his full effort in guiding us for achieving the goal. We have to appreciate the guidance given by other supervisor as well as the panels especially in our project presentation that has improved our presentation skills thanks to their comment and advices.

Thanking You,

Rishabh Goyal (161343)

Table of Content

Chapter 1- Introduction……………………………………………………..……………..……8

1.1 Introduction…………………………………………………………………….……… 8

1.2 Problem Statement……………………...……………………………………………….9

1.3 Objective………………………………………..…………………………………10

1.4 Methodology………………………………………..…………………………..………10

Chapter 2 – Literature Survey………………………..…………………………………11

Chapter 3 – System Development……………………………………………………13

3.1 Introduction to NumPy Library………………………………………………………13

3.2 Introduction to Pandas Libray……………………………………………………14

3.3 Introduction to Scikit learn………………………………………………………16

3.4 Regression Techniques…………………………………………………………...…16

3.4.1 Linear Regression………………………………………………..…………17

3.4.2 Polynomial Regression……………………………………………...………19

3.5 Classifier Algorithms………………………..…………………………………20

3.5.1 Logistic Regression…………………………………………………………20

3.5.2 Decision Tree…………………………………………………………….23

3.5.3 Naïve Bayes……………………..…….……………………………………27

3.5.4 K- Nearest Neighbour………………………………………………………30

3.5.5 Support Vector Machine (SVM)……………………………………………33

3.5.6 Neural Network…………………………...…………………………………….38

Chapter 4 – Perfomance Analysis……………………………………………………41

4.1 Bivariate Analysis…………………………………………….……………………41

4.2 Comparisions of Algorithms Output……………………………………………………43

Chapter 5 – Conclusion………………………………………………………………46

References……………………………………………………………………..……………47

List of Abbreviations

Abbreviation	**Meaning**
CSV	Comma-Separated Values
AQI	Air Quality Index
Ppb	part per billion
Conc.	Concentration
NO_x	Nitrogen Oxide Compounds
RMSE	Root Mean Squared Error
kNN	K-Nearest Neighbour

List of Figures

Fig 3.1 Array Creation Using NumPy ……………………………………………………………13

Fig 3.2 DataFrame ……………………………………………………………………………..15

Fig 3.4.1 Polynomial Curve ………………………………………………………………..19

Fig 3.4.1 Types of Curve ……………………………………………………………………19

Fig 3.5.1 Sigmoid function …………………………………………………………………20

Fig 3.5.2 Decision Tree ……………………………………………………………………..23

Fig 3.5.3 Prediction of Class ……………………………………………………………..24

Fig 3.5.4 Neural Network Layer ……………………………………………………………39

Fig 4.1.1 Analysis between Carbon Monoxide (CO) & PM2.5 Level ……………………….41

Fig 4.1.2 Analysis between Rain & PM2.5 Level ……………………………………………41

Fig 4.1.3 Analysis between O3 & PM2.5 Level ……………………………………………..42

Fig 4.1.4 Analysis between NO2 & PM2.5 Level ……………………………………………42

Fig 4.2.1 Output of Logistic Regression ……………………………………………………43

Fig 4.2.2 Output of Decision Tree ……………………………………………………………43

Fig 4.2.3 Output of Naïve Bayes ……………………………………………………………44

Fig 4.2.4 Output of SVM ……………………………………………………………………….44

Fig 4.2.5 Output of KNN ……………………………………………………………………45

Fig 4.2.6 Output of Neural network ……………………………………………………..45

Abstract

In this project, we attempt Air Quality Index prediction by using ML algorithms to estimate the h conc. of air pollutant (particulate matters (PM2.5) and SO2). ML, as very famous accepted technique, is capable to proficiently trained a models using efficient techniques. Though there some work apply ML to Air Quality forcasting, previous research are limited to several-year and simply trained standards regressions model (linear or notlinear) to estimate the air smog conc. In this project, we proposed advanced model to estimate the air pollution conc on the basis of meteorological data of earlier day by formulate the predict over as a multi-task learn question. That enable us to choose a good quality models with different algorithm technique. We proposed a useful algorithm by enforce the estimate models of successive hours to be shut to every other and evaluate it with numerous distinctive regularizations for MTL, nuclear model regularization, and `2,1-norm regularization. Our experiment have show that the projected parameter-reducing formulations and succssive-related regularization attain improved presentation than accessible average regression model and accessible regularization.

Chapter - 1
Introduction

1.1 Introduction

Air is the necessary usual resource for the continuation and survival of the whole existence on this earth. form of existence together with plant life and animal depends on Air for essential existence. all living organism needs superior quality of Air who is out of dangerous gas to keep on their existence. Accordings to the world bad poisoned places by, two of the bad contamination troubles in the world's are urban Air Quality and inside Air contamination. The growing people, its automobile, industry are pollute a Air at an shocking rates. Air contamination can reason lasting and short-range healths effect. It is find that the aged and youthful children are more exaggerated by air contamination. Short-range healths effects comprise eyes, nose, esophagus, headache, sensitive to reaction, and higher respiratory infection. lasting healthiness effect are lung tumor, mind harm, liver harm, harm, heart sickness, and respiratory illness. Also contribute to the reduction of the O3, which protect the Planet from sun UV rays. Other harmfull effects of air contamination is the configuration of acidic rainwater, which troubles tree, soil, river, and wildlifes. Some another ecological effect of air contamination are haze, eutrophicatio, and worldwide environment change. Air contamination is the most shocking concern for us today. addressing this concerns, in the past decade, many researcher have spent lot of time study and building various model and method in Air Quality study and estimation.

Air Quality estimation have been conduct by traditional approach in all year. The approach involves manual compilation, evaluation of raw data. the conventional algorithm for Air Quality forecasting use arithmetical and numerical methods. In methods, primarily a physical models is planned, figures is code with arithmetical equation. But these methods suffer from disadvantage as following:

1) These provides narrow correctness as these are not able to estimate the extremes point i.e. the pollutions utmost & smallest

2) cut-off can't found using these methods

3) It use inefficients methods for good outputs estimations

4) Existences of difficult arithmetical equations

With the expansion in technologies & researches, alternative to conventional approaches has been projected who uses ML algorithms. current time, many researcher has develope and use big-data analytic methods and ML based algorithms to predict Air Index valuation to gain good precision in valuation and estimations. The major goal is to provides a picture the gigantic researches works and usefull evaluation on present state of the art on relevant big-data methods and ML algorithms for Air Index estimation.

1.2 Problem Statement

Poor physical condition impact from revelation to outer surface air pollutant are composite func. of contaminant composition and conc. Main outside air pollutant in urban areas comprise ozone layer, particle matter (PM), (SO2), (CO), (NOx), unstable natural compound, pesticide, and metal. amplified fatality and morbidity have find in relationship with improved air pollutant (O3, PM and SO2) conc . According to the report from the meteorological department 10 parts per million amplify in the O3 addition relation may reason in excess early fatality per annum in the, as for many other megacities has struggle with air contamination as a result of industrialization and urban area. O3 precursor (such as NOx, and CO) emission have considerably diminish as the late 1970s, ozone level has not been according with principles put by the ecological fortification Agency to defend communal physical condition. element size is dangerous in determining the element evidence position in the individual respiratory arrangement. PM2.5, refer to particle with a thickness less than or equal to 2.5 _m, has been an rising distress, as these particle can be deposit into the lung gas exchange section, the alveoli . The revise the yearly ordinary of PM2.5 by falling the conc to 12 _g/m3 to offer improved shield beside health effect linked with long range and short-range experience . SO2, as an essential precursor of new element configuration and element expansion, has also been find to be linked with respiratory disease in various country. Hence, we chosen O3, PM2.5 and SO2 for test in this study.

1.3 Objective

i. Air Index estimation is an significant method to direct and manage air pollution. The distinctiveness of air provide influence its fitness for a precise uses. A a small number of air pollutant, call criterion atmosphere pollutant. These pollutant could damage physical condition, damage the atmosphere and reason goods damage. The present criteria pollutant following:

   1) CO
   2) Lead Pb
   3) NO2
   4) O3
   5) PM
   6) SO2.

ii. Learn and Understood the various ML approaches to estimate the accurate Air Index.

1.4 Methodology

I. Tools Used

i. Python IDLE

ii. NumPy

iii. Pandas

iv. Scikit-Learn / SkLearn

II. Research Involved

i. Machine Learning

III. Language Used

i. Python

Chapter – 2
Literature Survey

Many earlier period workings has been projected to apply ML approaches to Air Index estimation. Some researchs have planned to forecast objective into discredit stage. It elaborate property on Air contamination from metorological attributes such as temp, wind speed, rainfall, solar emission, and moisture and classify air contamination into unlike stage (lower, medium, higher, and dangerous level) by by means of a languid knowledge approach, the casing base calculation classification. It employed the neuro fuzzy network neurocomputing classify to forecast and classify O3 conc. into many levels (lower, medium, higher, and dangerous level) on the source of meteorological attributes and other pollutant such as SO2, NO, NO2. Kurt and Oktay model geographic relationships into a neural system approach and estimate every day conc. levels of SO2, CO2, and PM10. though, the procedure of convert worsening responsibilities to categorization tasks is not simple, as it pay no attention to the degree of the numeric information and consequently is imprecise. Other researches have work on forecast conc. of pollutant. Corani work on instruction neural system approach to predict O3 and PM10 conc. depending upon of information from the previous date. typically compare be the performance of forward neural system and prune neural system (PNNs). additional hard work have been made on practical a progressing machine and gray approach to get improved conventional FFNN approach. Jiang explore numerous algorithms (physical and chemical models, regressions models, and multiple level perceptron) on the air contaminant calculate task, and their consequences demonstrate that mathematics model are competitive with the traditional substantial and compound model. Ni, X. Y. et al. compare various geometric model on the base of PM2.5 information approximately and their consequences show that linear regressions approach can in some case be enhanced than the another approaches. MTL focus on erudition numerous tasks that have commonalities that can get better the effectiveness and accuracy of the approaches. It has achieve wonderful success in a lot of field, for ex natural language dealing out, figure detection bioinformatics market prediction etc. various of regularizations can be use to get better the commonalities of the connected work, that include the , nuclear norm, ethereal norm, Frobenius rule, and so on. But, the majority of the previous ML mechanism on air contaminant forecast did not consider the similarity among the approaches and only focus on improves the algorithms efficiency for a solo task, that is, improving forecast efficiency for every either separately or identically. As a consequence, we resolute to used meteorological and contaminant information to carry out estimation of conc. on the base of linear regression. In this project, we will focus on three dissimilar estimation algorithms formulations and use the MTL configuration with various regularizations. To the best of our considerate, this is the first work that has make use of MTL for the air contaminant forecast. We conquered logical algorithms and efficient technique to attain the best probable solution. The models assessment metric was the root mean squared error.

Chapter - 3

System Development

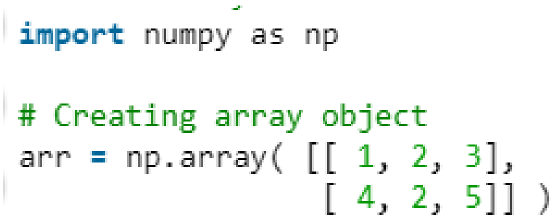
In this chapter, how the various features of NumPy, Pandas and Scikit-Learn library are to be implemented will be discussed.

3.1 Introduction to NumPy Library

NumPy is a normally used for array library. It provide a higher efficiency multi-dimensional array objects, and features for functioning the array.

It is essential library for technical compute in Python. It consist a variety of functions counting some significant are:

1. A dominant N Dimension array objects
2. complicated functions
3. Tool for integrate various language codes.
4. valuable linear arithmatical, Fourier transforms, and number ability



```
import numpy as np

# Creating array object
arr = np.array( [[ 1, 2, 3],
                 [ 4, 2, 5]] )
```

Fig 3.1 Array Creation Using NumPy

its understandable logical use, NumPy could as well be use as an well-organized multi dimension comsist of specific information.

uninformed data could be defined using Numpy library that allow us to NumPy faultlessly and quickly put together with a broad variety of database.

3.2 Introduction to Pandas Library

3.2.1 Introduction

Pandas is an open-source library that is made primarily for operational with relational or labeled data both easily and intuitively. It provides various data structure and operation for manipulate arithmetic data and time series. This collection is built on the top of the NumPy library. Pandas is fast and it has high-efficiency & output for users.

3.2.2 Key Features of Pandas

1. Speedy and efficients DataFrames objects with default and modified indexings.

2. Tool for loading data in-memory data object from dissimilar files format.

3. Data alignments and integrate handling of missing data.

4. Reshape and pivot of dateset.

5. Labelled base slice, index and subsets of large dataset.

6. Column from a data structures could be delete or insert.

7. Groups by data for aggregation & conversion.

8. High performance merge and joines the data.

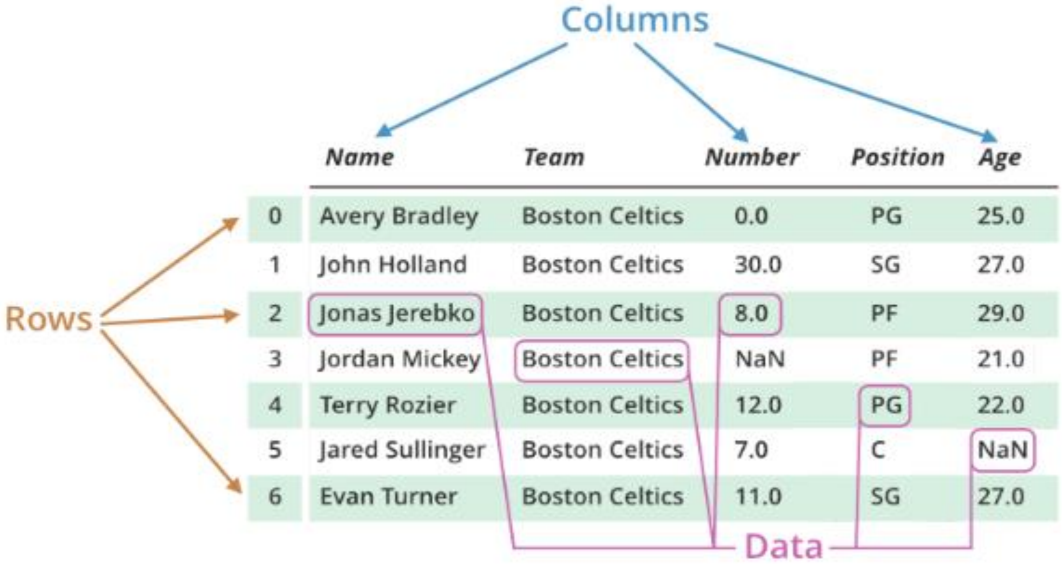
9. Time Series function.

3.2.3 Series

Pandas Library is a 1D labelled array capable of holding numbers of any type (integers - datatype, string-datatype, float-datatype, python objects, etc.). The axes label is jointly call as index. Pandas Series is a feature in excel datasheet. Label not require be distinguishing but be required to be a hash type. The entity support int and labeled base index and provide a host of approaches for executing operation connecting the indexing.

3.2.4 DataFrame

Pandas Data Frames is 2D dimension changeable, potentially mixed tabular statistics arrangement by label axis (row and attribute). A Data frames is a 2D data structure, i.e., information is combined in a tabular method in row and attributes. Pandas Data Frames contains of 3 major workings, the data, row, and attributes.



	Name	*Team*	*Number*	*Position*	*Age*
0	Avery Bradley	Boston Celtics	0.0	PG	25.0
1	John Holland	Boston Celtics	30.0	SG	27.0
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN
6	Evan Turner	Boston Celtics	11.0	SG	27.0

Fig 3.2 DataFrame

3.2.5 Why Pandas is used for Machine Learning

Pandas is usually used for data science but have you wonder why? This is for the reason that pandas is used in combination with other libraries that are used for data science. It is built on the top of the **NumPy** library which means that a lot of structure of NumPy are used or simulated in Pandas. The data produced by Pandas is often used as input for plotting functions of **Matplotlib**, arithmetical analysis in **SciPy**, machine learning algorithm in **Scikit-learn**.

Pandas program can be compile from any text editor but it is optional to use Jupyter Notebook for this as Jupyter given the capability to perform code in a particular cell rather than execute the entire file. Jupyter also provide an easy way to visualize pandas dataframes and plots.

3.3 Introduction to SciKit Learn Library

SKlearned is an open source Python api that executes a compilation ML algorithms, pre processing of data, cross validation of data and apparition approach using a combined crossing point.

3.3 .1 Important features of scikit-learn:

1. easy and proficient tool for statistics removal & statistics examination. It pooled many categorization, regressions and cluster approach together with SVM, RM, gradient method, k-means clustering etc.
2. Available to all and varied and re usable in a variety of framework.
3. build on the peak of NumPy, SciKitPy, and matplot library.
4. Open resource, commercial working.

3.4 Regression Techniques

3.4.1 Regression Analysis

Regressions study is a structure of analytical approach method that analysis the connection among a **needy** (target) and **self-determining var** This method is use for predicting, moment sequence approach and decisive the underlying cause link among the variable. For ex, connection among rash driving and amount of roads accident by a driv is bwtter considered through regressions.

3.4.2 Why do we use Regression Analysis?

Let say, you would like to predict enlargement in sale of a corporation base on accessible monetary condition. You have the current corporation information which specify the development in sale is roughly two and a half period the expansion in the financial system. with this approaching, we can estimate upcoming sale of the corporation base on present & former time information.

There are numerous benifit of using regression study. That are followings:

1. It signify the **important relations** among needy variable and self-regulating variable.
2. It signify the **power of contact** of numerous self-sufficient variables on a reliant variable.

3.4.2 Types of Regression Techniques

i. Linear Regression

Linear Regression is the mainly basic algorithm in Machine Learning. It is a regression algorithm which means that it is helpful when we are necessary to forecast continuous values, that is, the output variable 'y' is continuous in nature.

Terms to be used here:

1. Features - These are the independent variables in any dataset represented by x1 , x2 , x3, x4,... xn for 'n' features.

2. Target / Output Variable - This is the dependent variable whose value depends the independent variable by a relation (given below) and is represented by 'y'.

3. Function or Hypothesis of Linear Regression is represented by - y = m1.x1 + m2.x2 + m3.x3 + … + mn.xn + b Note: Hypothesis is a function that tries to fit the data.

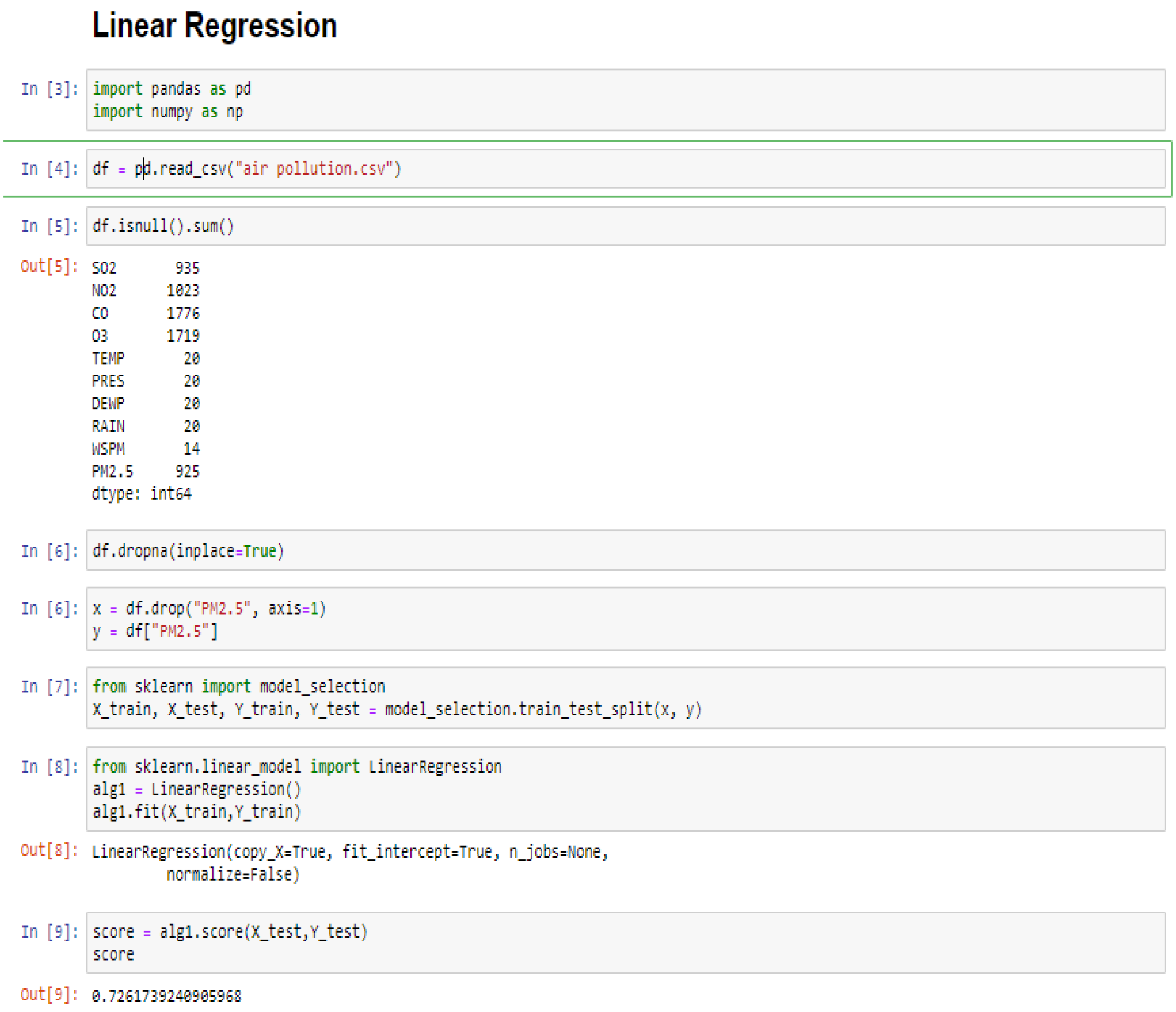
4. Intercept - Here b is the intercept of the line. We usually include this 'b' in the equation of 'm' and take 'x' values for that 'm' to be 1. So modified form of above equation is as follows: y = mx Where mx = m1.x1 + m2.x2 + m3.x3 + … + mn.xn + mn+1.xn+1 Here mn+1 is b and xn+1 = 1

5. Training Data - This data contains a set of dependent variables that is 'x' and a set of output variable, 'y'. This data is given to the machine for it to learn or get trained on some function (here the function is the equation given above) so that in future on giving some new values of 'x' (called testing data), our machine is able to predict values of 'y' based on that function.

Linear regression assumes linear relation between x and y.

The hypothesis function for linear regression is y = m1.x1 + m2.x2 + m3.x3 + … + mn.xn + b where m1 , m2 , m3 are called the parameters and b is the intercept of the line. This equation shows that the output variable y is linearly dependent on the features x1 , x2 , x3. The more you are dependent on a particular feature, more will be the value of corresponding m for that feature. We can find out which feature is more important or which feature is more affecting the result by varying the values of m one at a time and see if it is affecting the result, that is , the value of y.

So, here in order to predict the values of y for given features values (x values) we use this equation.



Linear Regression

```
In [3]: import pandas as pd
        import numpy as np

In [4]: df = pd.read_csv("air pollution.csv")

In [5]: df.isnull().sum()

Out[5]: SO2       935
        NO2      1023
        CO       1776
        O3       1719
        TEMP       20
        PRES       20
        DEWP       20
        RAIN       20
        WSPM       14
        PM2.5     925
        dtype: int64

In [6]: df.dropna(inplace=True)

In [6]: x = df.drop("PM2.5", axis=1)
        y = df["PM2.5"]

In [7]: from sklearn import model_selection
        X_train, X_test, Y_train, Y_test = model_selection.train_test_split(x, y)

In [8]: from sklearn.linear_model import LinearRegression
        alg1 = LinearRegression()
        alg1.fit(X_train,Y_train)

Out[8]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
                 normalize=False)

In [9]: score = alg1.score(X_test,Y_test)
        score

Out[9]: 0.7261739240905968
```

ii. Polynomial Regression

A regression is a polynomial regression equations if the degree of self-determining changeable variable is additional than single. The equations following stand for a polynomial equations:

Y $= a + b*x^2$

In these regression method, the greatest fit streak is not a in a directly line. It is slightly a curve that fit into the statistics point.

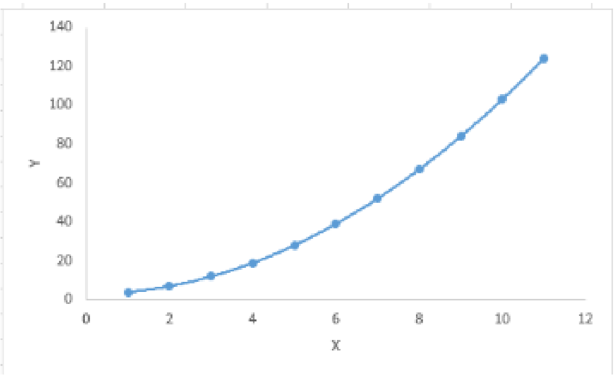


Fig 3.4.1 Polynomial Curve

While these may be a enticement to robust a superior level polynomial to got lesser fault, this could consequence in over fit. forever plot the relations to observe the fitting line and focal point on construction sure that curve fit the nature of the difficulty. Here is an ex of how plot be capable of facilitate out:

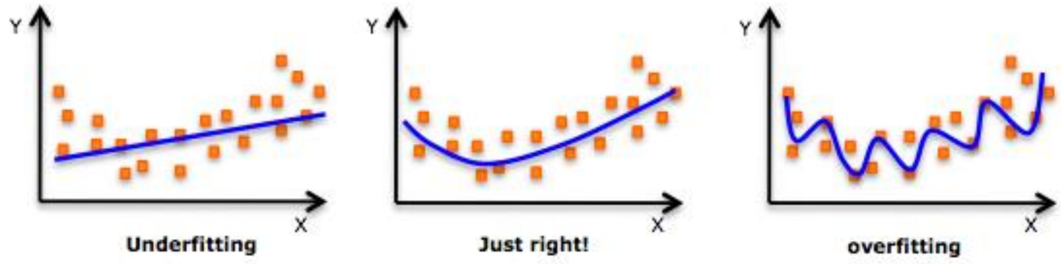


Fig 3.4.2 Type of Curve

3.5 Classifier Algorithms

3.5.1 Logistic Regression

Logistics Regression is a arithmetic way for analyzing a dataset in which there are 1 or more non dependent variable that decides the result. Logistic Regression is in fact a classification algorithm. Logistics regression is used to depict data and to clarify the relationships between one reliant dual variable and one or small interval or ratio-level non dependent variable. Prior to initial with Logistic regression we require to be familiar with about a functions called Sigmoid functions and its properties.

Sigmoid Function:

A sigmoid function is a numerical function having an "S" formed curve (sigmoid curve). Mathematically, the functions is :

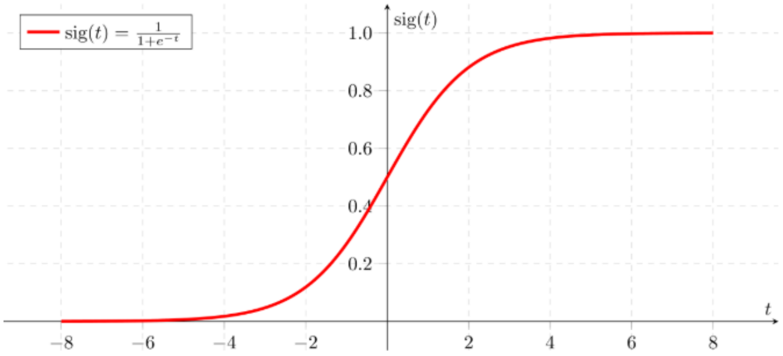


Fig 3.5.1 Sigmoid Function

With its output ranging between 0 and 1. As we can clearly see that the the curve quickly goes toward 1 when t>0 and toward 0 when t<0 and at t=0 it is equal to 0.5. Value of the above function for t=2 is 0.88 and for t=-2 is 0.119 , which shows how sharply it goes towards 0 and 1.

Because of the property of sigmoid function to give output between 0 and 1 we can use its output like probability, but not exactly as probability. For example, the property of probability that P(true)+P(false)=1 may not be true is case of sigmoid function i.e. S(true)+S(false) may not be equal to 1. As at t=0 we have S(t)=0.5 , and for t>0 we have S(t)>0.5 (sharply rising to 1 so we consider it 1) and for t<0 we have S(t)<0.5 (sharply falling to 0 so we consider it 0) , we have our decision boundary as 0.5. For ex. as our brink is 0.5 and our forecast func. return 0.7, we will categorize this study as positive(1). If our forecast was 0.2 we will categorize the study as negative(0) For logistics regressions with numerous groups we can choose the class with the uppermost estimated possibility.

Multiclass classification:

Multiclass Classification, as the name suggest ,are the kind of problem in which using the given parameter/feature we need to classifies into more than 2 classes. Instead of y=0,1 we will expand our definition so that y=0,1...n. Basically we re-run binary classifications multiple times, once for each class. For each sub-problems, we selects one class (YES) and lump all the others into a second class (NO). Then we take the classes with the highest predicted value.

Lets understand it better with the example of Iris datasets where we have to categorised flowers in 3 categories based on four features.

Lets consider the 3 categories as A, B and C.

We run binary classification (train) on this dataset with 3 different structures and meaning of the data ->

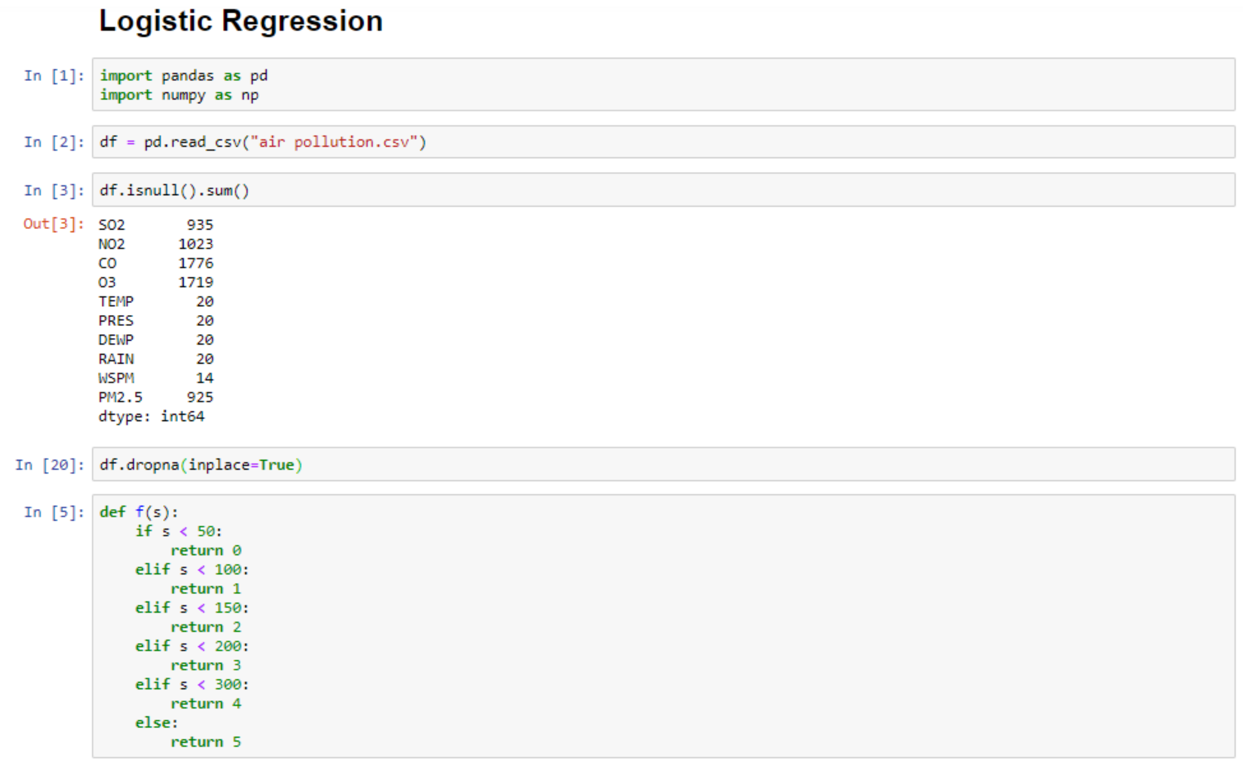
1) one time with A as true and others (B and C) as false.

2) Second with B as true and others (A and C) as false.

3) Third with C as true and others (A and B) as false.

Now, when we get a test samples we pass it into the model. Let the output from the three structures as shown above be O1, O2 and O3. We classify the test sample in the classes which has the highest value among these three output. For example, if O1=0.5 and O2=0.6 and O3=0.9 we classify the test sample as C.

Here we kind of use the output of Logistic regression as being the possibility of the test sample being in that particular class and not in other classes. But as mentioned earlier also this measure is not strictly possibility.



Logistic Regression

```
In [1]: import pandas as pd
        import numpy as np
```

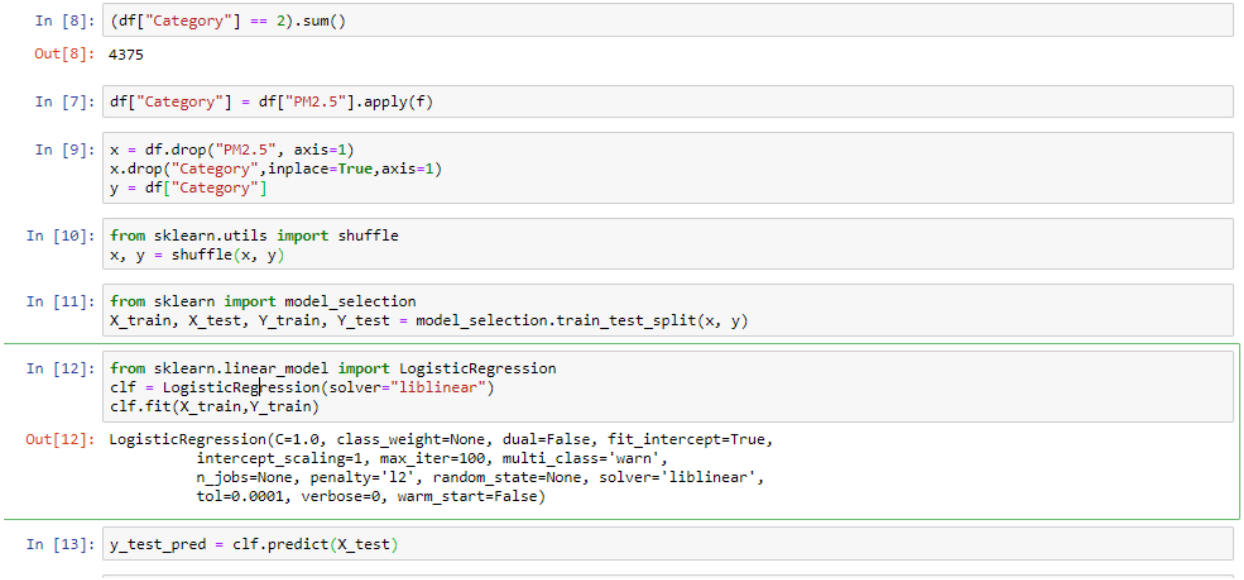
```
In [2]: df = pd.read_csv("air pollution.csv")
```

```
In [3]: df.isnull().sum()
```

```
Out[3]: SO2       935
        NO2      1023
        CO       1776
        O3       1719
        TEMP       20
        PRES       20
        DEWP       20
        RAIN       20
        WSPM       14
        PM2.5     925
        dtype: int64
```

```
In [20]: df.dropna(inplace=True)
```

```
In [5]: def f(s):
            if s < 50:
                return 0
            elif s < 100:
                return 1
            elif s < 150:
                return 2
            elif s < 200:
                return 3
            elif s < 300:
                return 4
            else:
                return 5
```



```
In [8]:  (df["Category"] == 2).sum()

Out[8]:  4375

In [7]:  df["Category"] = df["PM2.5"].apply(f)

In [9]:  x = df.drop("PM2.5", axis=1)
         x.drop("Category",inplace=True,axis=1)
         y = df["Category"]

In [10]: from sklearn.utils import shuffle
         x, y = shuffle(x, y)

In [11]: from sklearn import model_selection
         X_train, X_test, Y_train, Y_test = model_selection.train_test_split(x, y)

In [12]: from sklearn.linear_model import LogisticRegression
         clf = LogisticRegression(solver="liblinear")
         clf.fit(X_train,Y_train)

Out[12]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, max_iter=100, multi_class='warn',
                   n_jobs=None, penalty='l2', random_state=None, solver='liblinear',
                   tol=0.0001, verbose=0, warm_start=False)

In [13]: y_test_pred = clf.predict(X_test)
```

3.5.2 DecisionTree

- Decision tree approach comes in the category of supervise knowledge. It can be apply to resolve together regressions and classifications problem.
- Decision tree apply the hierarchy representations to resolve the problem in which every leaves node keep in touch to a group label and characteristic are represent on the interior node of the hierarchy.
- We can symbolize any bool func. on distinct attribute by the decision tree approach.

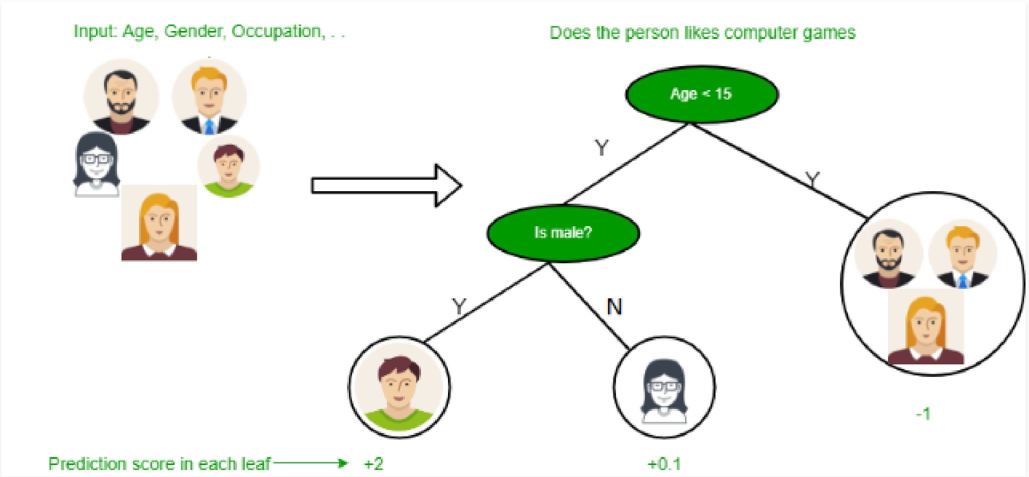

Input: Age, Gender, Occupation, . .
Does the person likes computer games
Age < 15
Y
Y
Is male?
Y
N
-1
Prediction score in each leaf
+2
+0.1

Fig 3.5.2 Decision Tree

Following are a few assumption that we make while use decision tree:

- In the starting, we considering the entire training dataset as the core nodes.
- Column value are favored to be unconditional. If the value are constant then they are discretize preceding to preparing the algorithm.
- On the base of column value proceedings are scattered repeatedly.
- Use arithmetical func. for order columns as roots or the interior nodes.

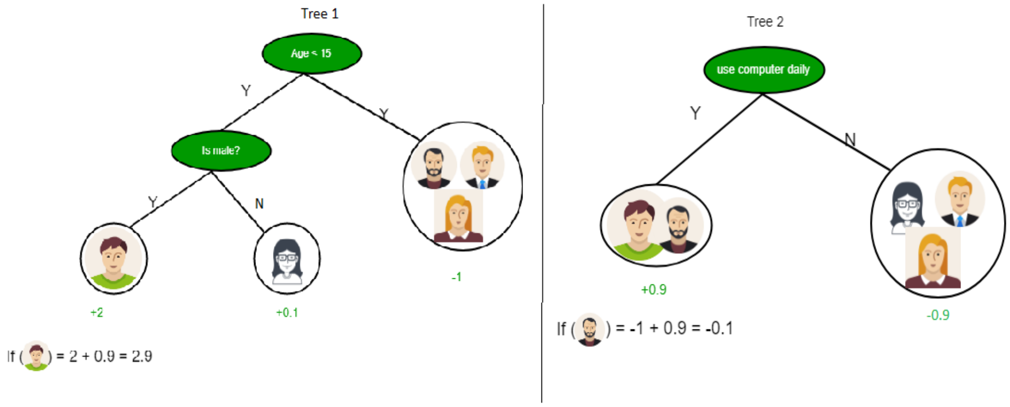


Fig 3.5.3 Prediction of Class

because you be able to observe from the upper images that Decision Tree working on the addition of Product figure which is also recognized as Disjunctive standard Form. the upper picture, we are estimating the use of PC in the every day life of the public.

In Decision Tree the main problem is to recognition of the characteristic for the roots nodes in every stage. This procedure is identified as characteristic collection. We include two accepted characteristic collection procedures:

1) Information Gain
2) Gini Index

1. **Information Gain**

   while we utilize a leaf in a decision tree to divider the preparation example into lesser division the randomness varies. Information gain is a procedures of this variation in randomness.

   Definition: Let S is a set of example, A is an column, S_v is the separation of S with A = v, and value (A) is the set of all apparent value of A

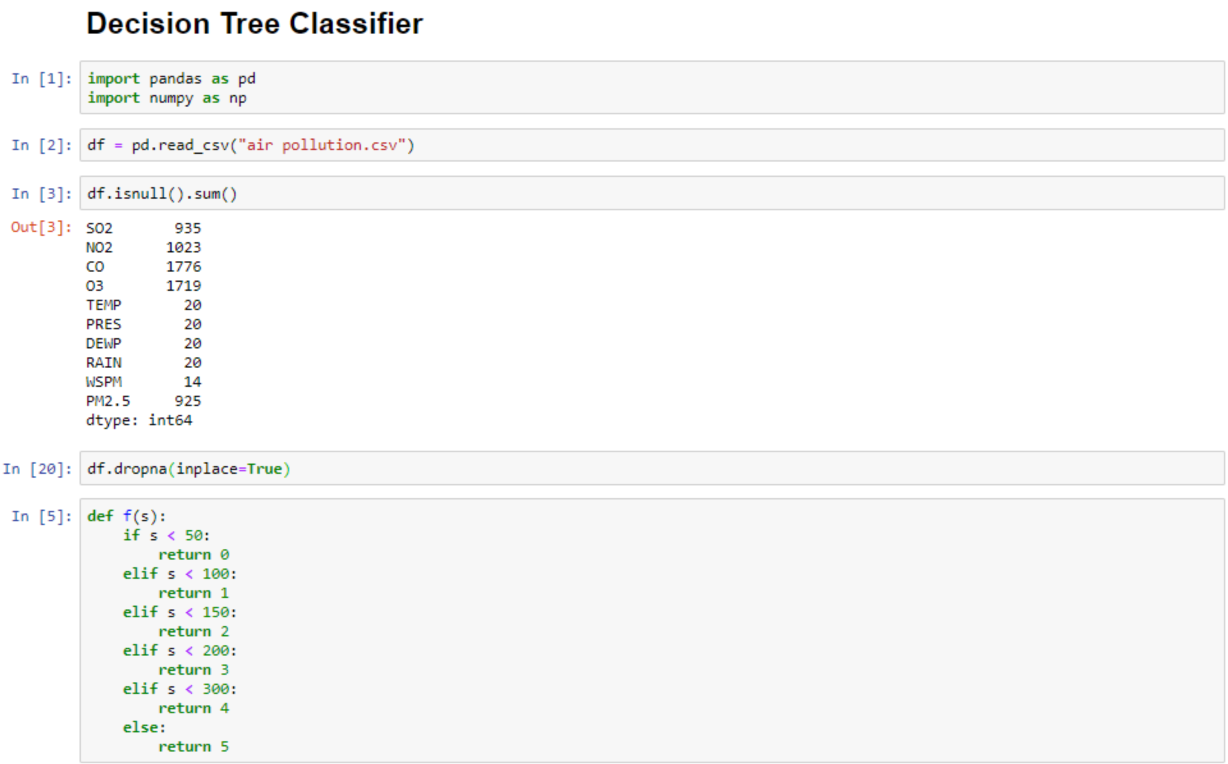
$$Gain(S, A) = Entropy(S) - \sum_{v \epsilon Values(A)} \frac{|S_v|}{|S|}.Entropy(S_v)$$

2. **Gini Index**

   Gini index or Gini impurity dealings the power or possibility of a exacting changeable being incorrectly classify what time it is erratically selected. But what is in fact destined by ‘impurity’? If all the component fit in to a only group, then it can be call clean. The degree of Gini index ranges among 0 and 1, where 0 signify that all component fit in to a definite group or if here subsist just one group, and one indicate that the component are erratically scattered crosswise a choice of module. A Gini Index of 0.5 indicates equally extend elements into some classes.

$$\text{Gini} = 1 - \sum_{i=1}^{n} (p_i)^2$$

where p_i is the likelihood of an object being classified to a particular class.



Decision Tree Classifier

```
In [1]: import pandas as pd
        import numpy as np
```

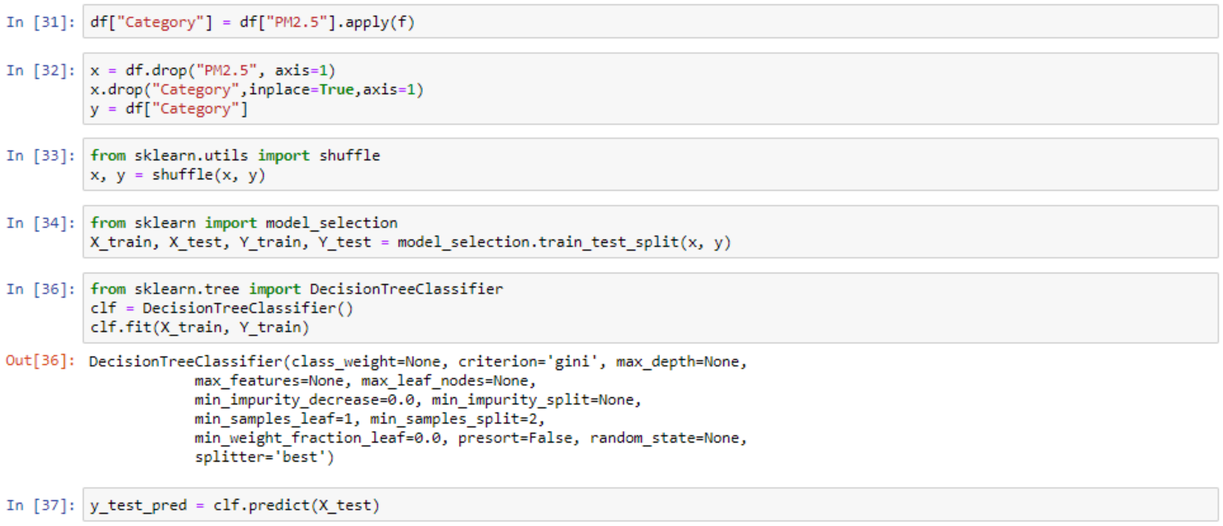
```
In [2]: df = pd.read_csv("air pollution.csv")
```

```
In [3]: df.isnull().sum()
```

```
Out[3]: SO2       935
        NO2      1023
        CO       1776
        O3       1719
        TEMP       20
        PRES       20
        DEWP       20
        RAIN       20
        WSPM       14
        PM2.5     925
        dtype: int64
```

```
In [20]: df.dropna(inplace=True)
```

```
In [5]: def f(s):
            if s < 50:
                return 0
            elif s < 100:
                return 1
            elif s < 150:
                return 2
            elif s < 200:
                return 3
            elif s < 300:
                return 4
            else:
                return 5
```



```
In [31]: df["Category"] = df["PM2.5"].apply(f)
```

```
In [32]: x = df.drop("PM2.5", axis=1)
         x.drop("Category",inplace=True,axis=1)
         y = df["Category"]
```

```
In [33]: from sklearn.utils import shuffle
         x, y = shuffle(x, y)
```

```
In [34]: from sklearn import model_selection
         X_train, X_test, Y_train, Y_test = model_selection.train_test_split(x, y)
```

```
In [36]: from sklearn.tree import DecisionTreeClassifier
         clf = DecisionTreeClassifier()
         clf.fit(X_train, Y_train)
```

```
Out[36]: DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
                     max_features=None, 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, presort=False, random_state=None,
                     splitter='best')
```

```
In [37]: y_test_pred = clf.predict(X_test)
```

3.5. 3 Naïve Bayes

Bayes Theorem defines possibility of an event based on the prior knowledge of factors that might be related to an event.

Mathematical Statements of Bayes Theorem is as follows :

Now, basically for a data point xi, we have to forecast the class that the current output Y belongs to. Suppose, there are total 'j' number of class for output.
Then,
P(y=c1|x=xi) ---> tells us that for given input xi what is the possibility that y is c1.
P(y=c2|x=xi) ---> tell us that for given input xi what is the possibility that y is c2 and so on till cj.

Out of all these probability calculation, y belongs to that particular class which has highest possibility. We will use Bayes theorem to doing these probability calculation.This gives us the possibility that the output belong to jth class for the current value of data point(xi). Since for all the classes 1,2,..., the denominator will have the same value, so we can ignore this while doing comparison. Hence, we obtain the given formula to calculate possibility.

NAIVE ASSUMPTION

The estimate for possibility P(y=cj), can be done straight from the number of training points.
Suppose there are 100 training points and 3 target classes, 10 belong to class cls1, 30 belong to class Cls2 and remaining 60 belong to class Cls3.
The estimate value of class possibility will be :
P(y = C1) = 10/100 = 0.1
P(y = C2) = 30/100 = 0.3
P(y = C3) = 60/100 = 0.6
To make the possibility estimate for P(x=xi|y=cj), naive bayes categorization algorithm assume all the features to be independent. So, we can calculate this by individually

multiplying the probability obtained for all these features (assuming features to be independent), for the output of jth class.

$P(x=xi|y=cj) = P(x=xi^1|y=cj)\ P(x=xi^2|y=cj)\ ....\ P(x=xi^n|y=cj)$

here, xi^1 denotes the value of 1st feature of ith data point and $x=xi^n$ denotes the value nth feature of the ith data point.After taking up the naive assumption, we can easily calculate the individual probabilites and then by simply multiplying the result calculate the final possibility P'.

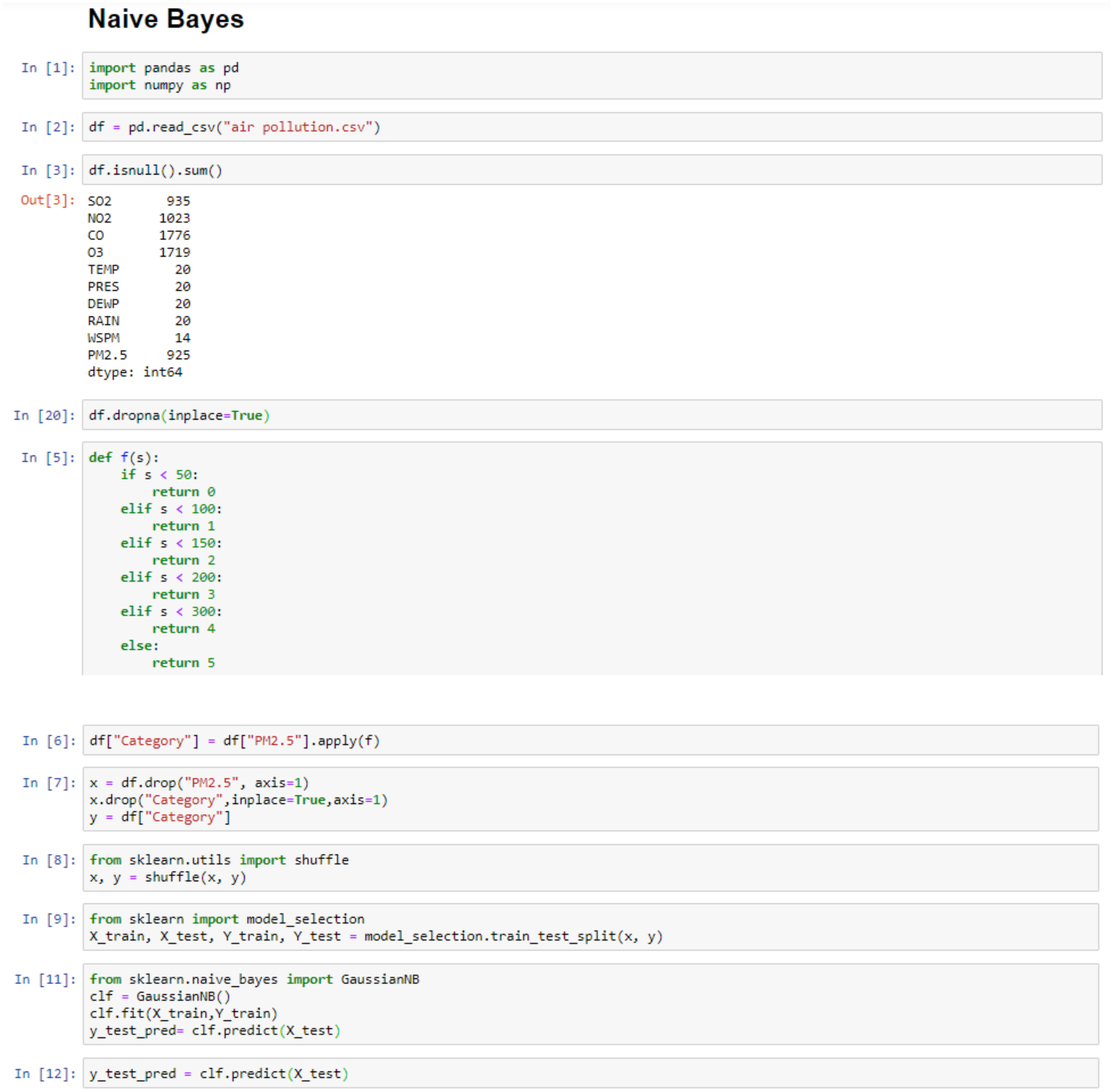
Using the above formula, we can calculate the possibility that the output y belongs to jth class, for the given ith data point. Class possibility [P(y = cj)] will be calculated from the data given and and individual possibility [$P(x=xi^k|y = cj)$] will be calculated by diving the data class wise can calculating these for the jth group.

Handling zeroes using Laplace correction

Let’s consider the following condition: you’ve trained a Naive Bayes algorithm to distinguish between spam and not spam mails. What happen if the word “Casino” doesn’t show up in your training data set, but appears in a test example?

Well, your algorithm has never seen it before, so it sets the probability that "Casino" appear in a spam document to **0**; So every time this word appear in the check data , you will try hard (it has P = 0) to mark it as not spam just since you have not seen that word in the spam part of training data.This will make the model very less proficient and therefore we want to minimize it. We want to keep in mind the possibility of any word we have not see (or for that topic see in the not-spam part of training data), may have a possibility of being a word used in spam mails greater than 0. The equal is correct for each word to be a part of not-spam mail.

To evade such issue with unobserved values for feature, as well as to combat overfiting to the data set, we imagine as if we’ve seen each word 1 (or k, if you’re smoothing by k) time more than we have actually see it, and adjust the denominator of our occurrency divisions by the size of the overall dictionary to account for the “pretence”, which really works well in perform. If you take smooth factor k equal to 1 , it becomes Laplace modification. The equation below show Laplace modification for the example taken.



Naive Bayes

```
In [1]: import pandas as pd
        import numpy as np
```

```
In [2]: df = pd.read_csv("air pollution.csv")
```

```
In [3]: df.isnull().sum()
```

```
Out[3]: SO2        935
        NO2       1023
        CO        1776
        O3        1719
        TEMP        20
        PRES        20
        DEWP        20
        RAIN        20
        WSPM        14
        PM2.5      925
        dtype: int64
```

```
In [20]: df.dropna(inplace=True)
```

```
In [5]: def f(s):
            if s < 50:
                return 0
            elif s < 100:
                return 1
            elif s < 150:
                return 2
            elif s < 200:
                return 3
            elif s < 300:
                return 4
            else:
                return 5
```

```
In [6]: df["Category"] = df["PM2.5"].apply(f)
```

```
In [7]: x = df.drop("PM2.5", axis=1)
        x.drop("Category",inplace=True,axis=1)
        y = df["Category"]
```

```
In [8]: from sklearn.utils import shuffle
        x, y = shuffle(x, y)
```

```
In [9]: from sklearn import model_selection
        X_train, X_test, Y_train, Y_test = model_selection.train_test_split(x, y)
```

```
In [11]: from sklearn.naive_bayes import GaussianNB
         clf = GaussianNB()
         clf.fit(X_train,Y_train)
         y_test_pred= clf.predict(X_test)
```

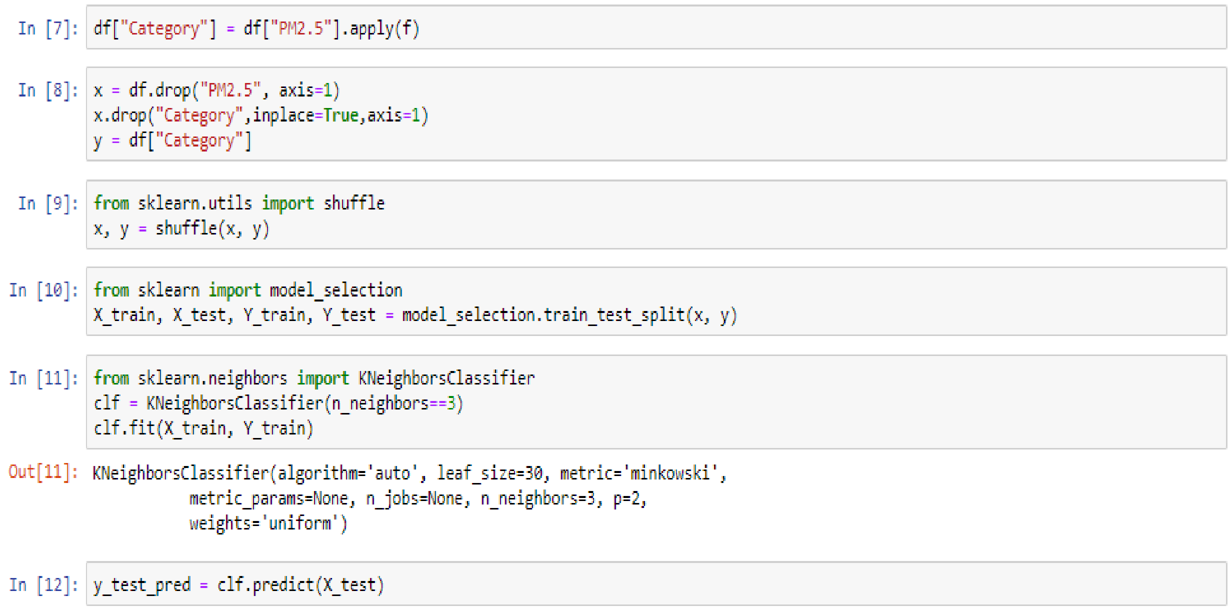
```
In [12]: y_test_pred = clf.predict(X_test)
```

3.5.4 K Nearest Neighbors

The k Nearest Neighbors computation also called k-NN is a non-parametric approach or method utilize for classification and reversion. In both the instance of categorization and deterioration the comprise of the k nearest prepare model that are accessible in the elements space. The yield it considerably rely upon whether the k-NN is utilize for group or it is be utilized for relapse.

In K-NN when it is utilize for understanding, the yield is normally a class contribution. An editorial is prearranged on the off possibility that there is a better part of vote from its neighbor and, at that points the items are doled out to the classes that is more regular between its entire k close neighbor where k is a positive no. normally little. In the even that k = 1, at that points the items are basically relegated to the classes of its single close neighbors.

KNN what that stand for is one of the simple supervise ML approach most use for categorization so wants to recognize is this a dog or it is not a dog or is it a cat or not a cat it classify a data points base on how it's neighbor are classify K Nearest in store all existing case and classify new case base on a parallel measures and here we gone from cat and dog right into another favorites of mine K Nearest in stores all existing case and classify new case base on a parallel measures and here you see we have a measurements of SO2 versus the Cl levels and then the differents wines they have test and where they falled on such graph base on how much SO2 and how much cl K Nearest Neighbor is a perimeter that refer to the numbers of nearests neighbor to includes in the majority of the vote processes and so if we added a new glass of wines there red and white we want to know what the neighbor is in this cases we are going to put k equal 5 and we will talk about K in just a minute a data points is classify by the greater part of vote from its five nearest neighbor here the unknown points will be classify as red since 4 out of 5 neighbour are red so why do we decide K how do we know K equals 5. I mean that is what is the value we put in there and so we can talk about it how do we choose the factor K knn algorithms is base on features connection choose the right values of K is a process called parameters alteration and is essential for best precision so at K equal three we can classified we have a questions marks in the middle as either a as a squares or not is it a squares or is it in this cases a triangles and so if we set K equal to three we are going to look at the 3 nearest neighbor we are going to say that is a squares and if we put K equal to 7 we classified as a triangles depends on what's the others data is arounds and you could see is K changed pending on wherever that points is that considerably modify your respond and we jump where we go how is do decide the factors of K you will found this in all machine learnings choose such factors that's the features you got he is likes oh my gosh you had say decide the correct K did it correct my value in whatever machine learnings tools you are look at so that you do not has a vast bias in one direction or the other and in terms of K n n the number of K if you choose it too low the partiality is base on it is just too noise it is correct next to a couple thing & it is going to choose those things and you may get ask to answer if you K is also huge then it's going to took ceaselessly to procedure so you are going to lope into processing issue sand reserve issue so i does the most ordinary exercise and there is other options choose K is used the square root of N so sum number of principles you has you took the square root of it the majority case too if it's level number if you are using like in these cases square and triangles if it is even you wants to create your K value odd help it choose enhanced so in additional terms you are not leaving to have a stability among 2 dissimilar factors that be equivalent so typically took the square roots of N and if it is level you put in 1 to it or minus one as of it and that is what you got the k value from that is the nearly all ordinary use and it's good so lid it workings extremely fit while i use KNN we can use k n when data is label so you require a labelled in it we know we has a group of images with dog and cat data is sound free and so can observe at this time while we has a group so as to we have similar to underfed 140 23 Kitty normal that is quite confusing to have a variety of value approaching in so it's very noisy and that will reason an concern Dana said is little so we are typically functioning with lesser data-sets wherever might get into show of information if it is really clean it does not has a lots of sound because k , k is a idle apprentice i.e. it does not be trained a discriminate function from the trained sets so it is extremely lazy so if you has incredibly complex statistics and you has a big quantity of it you are not leaving to utilize the knn but it is really great to obtain a put to create still with large statistics you be able to class a small example and obtain an thought of whats that looks similar to by means of the knn and too just using for lesser information sets works actually good quality how do a knn algorithm effort believe a information set having 2 varible altitude and centimeter and load in kilogram and each tip is secret as usual or under weight so we be able to observe correct at this time we have 2 variables you be familiar with either usual or they are not they are underfed on the base of the known information we has to classify the under sets as ordinary or under weight using knn so if we has new statistics impending in this says 57 kilogram and 156 centimeter so as to departing to be normal or underfed to found the nearest neighbor we will compute the Euclidean distance according to the Euclidean distances formulas the distance between 2 point in the planes with the coordinates xy and a b is given by instance d equal the squares roots of x minus a square plus y minus B squared and you could keep in mind that as of the two edges of a triangles we are calculating the third edges since we know the X sides and the Y side lets compute it to understand clearly so has our unidentified points and we placed it here in red and we have our other point where the data is scattered around the distances D1is a squared root of 180 - 157square + 47 - 61 square which is regarding six point 7 and distance two is with reference to 13 and distances three is about 13 point four similarly we will compute the Euclidean distances of unidentified data point from all the point in the dataset and because we are deal with small amount of data that is not more that hard to do it is in reality pretty fast for a computer.

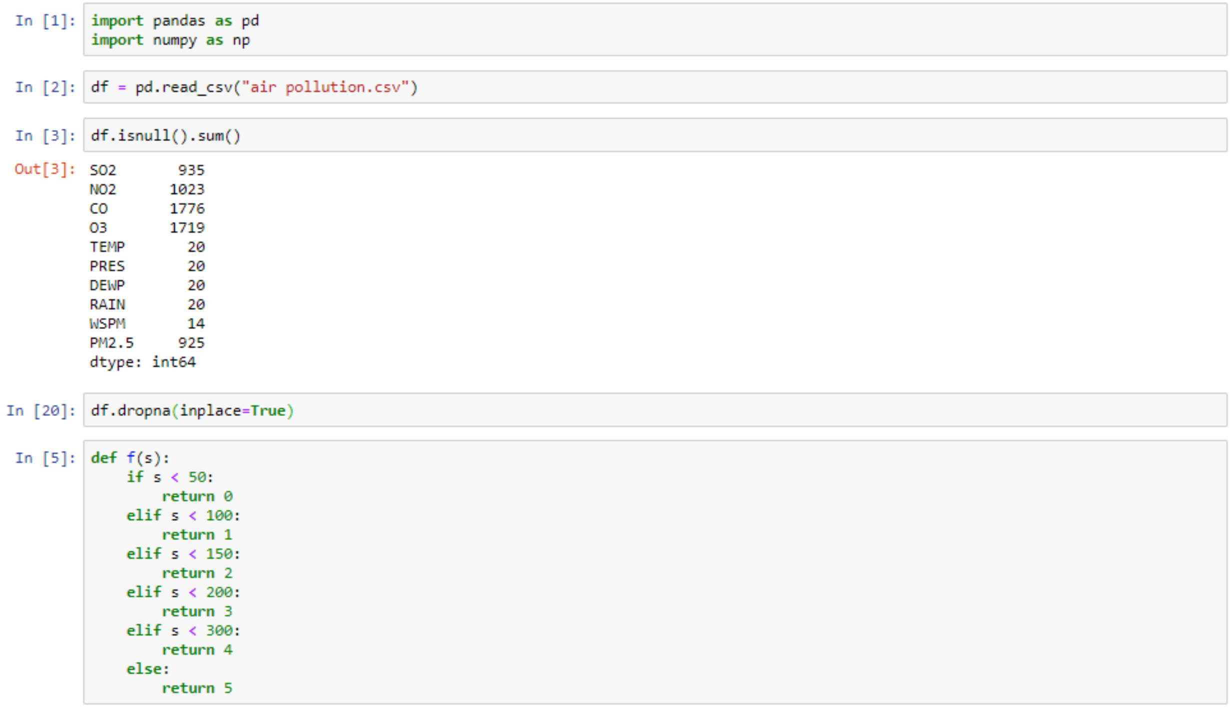


```
In [7]: df["Category"] = df["PM2.5"].apply(f)

In [8]: x = df.drop("PM2.5", axis=1)
        x.drop("Category",inplace=True,axis=1)
        y = df["Category"]

In [9]: from sklearn.utils import shuffle
        x, y = shuffle(x, y)

In [10]: from sklearn import model_selection
         X_train, X_test, Y_train, Y_test = model_selection.train_test_split(x, y)

In [11]: from sklearn.neighbors import KNeighborsClassifier
         clf = KNeighborsClassifier(n_neighbors==3)
         clf.fit(X_train, Y_train)

Out[11]: KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                    metric_params=None, n_jobs=None, n_neighbors=3, p=2,
                    weights='uniform')

In [12]: y_test_pred = clf.predict(X_test)
```

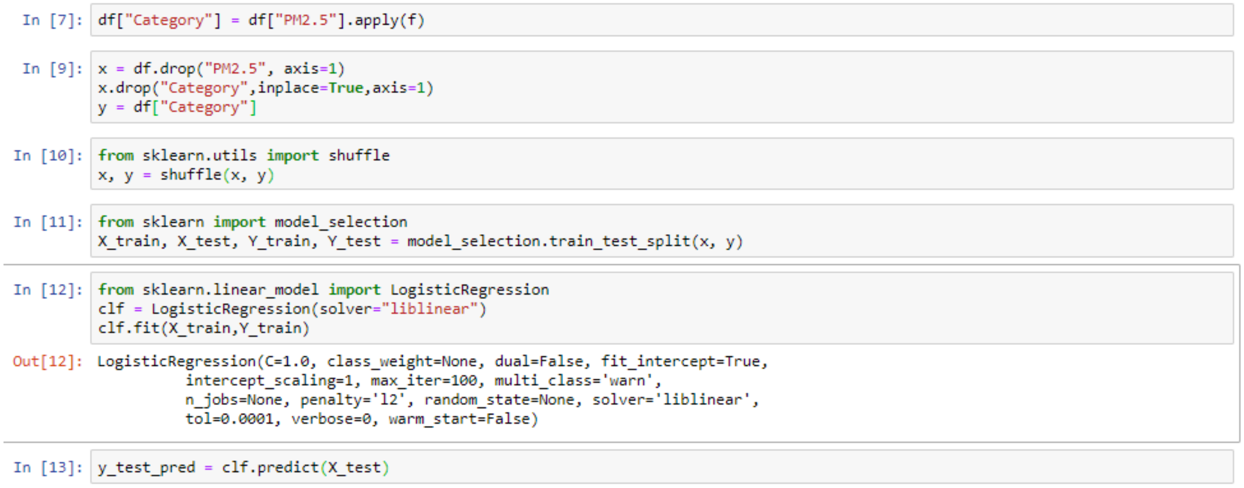
3.5.5 Support Vector Machine (SVM)

SVM is specific to supervise ML model learn beginning the precedent enter data and make upcoming estimation as output so we instruct the approach we instruct it a strawberry is and one time the algorithm is trained it could recognize up straw berry that's what is signify by supervised learning in the bigger representation of the machine knowledge algorithm and under supervised knowledge you could watch the support vectors fixed in under classification deciding what true and flase is and here is as well a regression version but it’s mainly use for classification lets took a detour an see that we could attach us to the person knowledge and found out why support vector classifier so in this ex last week my friend and I visit a shop is that an apples or a strawberries so the query come up whats do they just choose up from the stand after a pair of second we could find out that it was strawberry so lets took this approach a step additional and lets whynot construct a algorithm which could estimate an un knowing data and in this you are available to be seem in gat a few sugary strawberry or crispy apples you wants it to be clever to labelled that two and make a decision whatever the fruits is and you do so as to by have information already put in so we by now contain a group of strawberries you know strawberry and they are by now label as that we already has a group of apple we know our fruits and are label as such then one time we trained our algorithm that models then can be known the new information the new data is this picture in this case we can see a query marks on it and it come during and go it is a strawberry in this casing we are by the support vector machine approach is a supervise knowledge technique that look at data and sort it into one of 2 category and in these casing we resort the strawberries into the strawberry place at this end we be ask the query how do the prediction works earlier than we excavate into an example with information let relate these to our fruits situation we has our support vector we have took it we taken label example of information strawberries and apple and we drained a line downward the focus connecting the two groups this opening now allow to take new information in these casing an apples and a strawberry and leave them in the suitable grouping base on whichever sides of the row they go down in and these ways we can estimate the unknown as colors full and delicious as a fruits let's look the benefit of support vector machine we will create with higher dimension input space or occasionally refer to the curse of dimension we look at earlier one dimensions to another dimensions three dimension while you find to a thousand dimensions a group of troubles start occurring with the majority approach that have to used to the SVM automatically does it in high dimensionality one of the high dimensional room one higher dimensional space thats works on is spare text vector these is wherever we tokenize the terms in file so we can run our ML Approaches over though I have watch once got as higher as 3million dissimilar token that is a lots of vectors to seem at and to finish we have regularization stricture the realization parameter or labda is a stricture that helps form out where we are disappearing to have a unfairness or more proper of the data whether it's going to be above fixed to a precise example or is going to be biased to a advanced low down charge with the SVM it logically avoid the over fitting and unfairness troubles that we observe in many other approach 3 compensation of the SVM formulate it a very controlling tool to append to your repertoire of ML tools now we do assure you a uses cases study we are really going to leap to some Python language and so we are going to go into a trouble report and found off with the we have relations member leaving to the zoo we have the little kid leaving dead is that a collection of crocodile or alligator well that's rigid to distinguish and zoos are a vast position to create look at science and sympathetic how things work particularly as a little kid and so we could see the parent in thinking well what is the distinction among a crocodiles and an alligators fit one crocodile are superior in size alligator are slighter in extent snout size the crocodile have a slim muzzle and alligator have a bigger muzzle and the path in the current day and age the father's is idea how can It twist this into a session and goes let a SVM separate the two group that would be hilarious now in this case we're not leaving to used definite capacity and information we are just using that for images and hats very ordinary and a lot of ML approaches and location them up by let turn round up our sleeve and we will talked about that extra in just a minute as we split into our Python writing so here land in our definite code and I am leaving to move this into a Python editor in just a instant but let talked a small fragment about what we are going to wrap first we are leaving to wrap in the system the complex how to in fact create our SVM and we are going away to locate that there is only two appearance of rules that actually make it and the relax of it is complete so rapid and quick that it is all at this time in the primary and we will show you what that look similar to as distant as our information so we are going gone to make a few data I talk about create information just a little ago and so we will get into the create information here and you will see this polite modification of our 2 blob and we will go through that in just a moment and after that the next part is we are available to obtain this and we are going to knock it up a nick we are going away to explain you what it look like in the rear the sight but let found with in reality create our system I like to use the Anaconda Jupiter notebooks since it's extremely easy to use but you can use preferred Python editor or setup and go in there but let go ahead and button over there see what that look similar to so here we are in the anaconda Python notebooks or anaconda Jupiter notebooka with Python we are using Python 3 I think this is3.5 but it should be work in any of your3x version and you had have to gaze at the SKlearn and make sure if you are with a 2x account or an previous account let go in front and put our system in there and one of the belongings I similar to concerning the Jupiter notebook is I go up to sight and I'm leaving to go in advance and closure the line information on to create it a small bit easy to converse about and we could still add to the dimension since this is that I did in in this case I am using Google Chrome and that is however it release up for the editors though anybody any similar to I said any editors will occupation at the present the first step is going away to be our import and we are going to introduce four dissimilar part the initial two I wish for you to seem at our line 1 and line 2 our as NP and matplot documentation pieplot as p now these are incredibly identical import whenever you are liability effort the initial one is the statistics python we need that since parts of the raised area we are using uses out for the mp array and I will speak about that in a little so you could recognize what we want to used a np array against a normal python arrays and usually it is normal system to use NP for numpy the drawing plot records is how we are departure to vision our information so this have do you require the NP for the SKlearn component but the mapplot records is only for our use for mental picture and so you in reality do not require that for the SVM but we are going away to put it at hand so you include a pleasant illustration assist and we could demonstrate you what it look similar to that is actually significant at the finish what time you come to an end all so you include a kind present for everyone to appear at and then in conclusion we are going to I am going to bound one in front to row numeral four that is the SKlearn datasets sample creator introduce and create blob and tell that we be going away to construct up information and this is a implement that is in the SKlearned to create up information I in my opinion do not wish for to go to the zoo get in difficulty for jump more the hedge and perhaps flattened by the crocodile and alligator as I effort on measure their snout and by means of them length in its place we are just going away to construct up a few information and that is what that formulate blob is it is a magnificent too you are ready to experiment your system and you are not confident in relation to what information are available to set in there you can generate this blobs and it make it real simple to utilize and to finish we have our actual SVM the SKlearned importation SVM online so that cover all our import going away to produce keep in mind I use the build blob to make information and we are leaving to generate a capital X and a lower y equal to create blob in sample equal to 40 so we are going away to make 40 lines of statistics it is going away to have two center with the unsystematic state.

Support Vector Machine



```
In [1]: import pandas as pd
        import numpy as np

In [2]: df = pd.read_csv("air pollution.csv")

In [3]: df.isnull().sum()

Out[3]: SO2       935
        NO2      1023
        CO       1776
        O3       1719
        TEMP       20
        PRES       20
        DEWP       20
        RAIN       20
        WSPM       14
        PM2.5     925
        dtype: int64

In [20]: df.dropna(inplace=True)

In [5]: def f(s):
            if s < 50:
                return 0
            elif s < 100:
                return 1
            elif s < 150:
                return 2
            elif s < 200:
                return 3
            elif s < 300:
                return 4
            else:
                return 5
```



```
In [7]: df["Category"] = df["PM2.5"].apply(f)

In [9]: x = df.drop("PM2.5", axis=1)
        x.drop("Category",inplace=True,axis=1)
        y = df["Category"]

In [10]: from sklearn.utils import shuffle
         x, y = shuffle(x, y)

In [11]: from sklearn import model_selection
         X_train, X_test, Y_train, Y_test = model_selection.train_test_split(x, y)

In [12]: from sklearn.linear_model import LogisticRegression
         clf = LogisticRegression(solver="liblinear")
         clf.fit(X_train,Y_train)

Out[12]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, max_iter=100, multi_class='warn',
                   n_jobs=None, penalty='l2', random_state=None, solver='liblinear',
                   tol=0.0001, verbose=0, warm_start=False)

In [13]: y_test_pred = clf.predict(X_test)
```

3.5.6 Neural Network

Neural system are a set of appraoches, model slackly after the person mind, that are planned to differentiate model. They understand sensory statistics throughout a kind of method approaching, classification or cluster unrefined input. The prototype they differentiate are mathematical, consist of vector, into which all real world statistics, be it imagery, noise, wording or moment in time sequence, must be interpret.

Neural system assist us gather and classify. You could believe of them as a cluster and categorization level on peak of the statistics you accumulate and organize. They assist to cluster unlabel information according to likeness amongst the case input, and they classify information what time they have a labelled datasets on trained on. (Neural system can also remove column that are fed to other approaches for cluster and classification; so you can imagine of deep neural system as workings of larger ML approaches linking algorithm for reinforcement knowledge, classification and regressions).

Deep knowledge is the name we use for “stacked neural system”; that is, system compilation of numerous level.

The level are prepared of *node*. A node is now a position wherever computation happen, slackly ornate on a neuron in the person mind, which fire when it encounter sufficient stimulus. A node combine key from the information with a set of coefficients, or weights, that also intensify or damp that key, thus transmission consequence to input with consider to the assignment the approaches are difficult to learn; e.g. which key is the majority supportive is categorize statistics with no mistake? These key influence goods are summed and then the amount is accepted throughout a node so called commencing purpose, to decide whether and to what degree that indication be supposed to development additional during the system to influence the last result, say, an act of categorization. If the signal pass throughout, the neuron has been “activated.”

Here’s a diagram of what one node might look like.

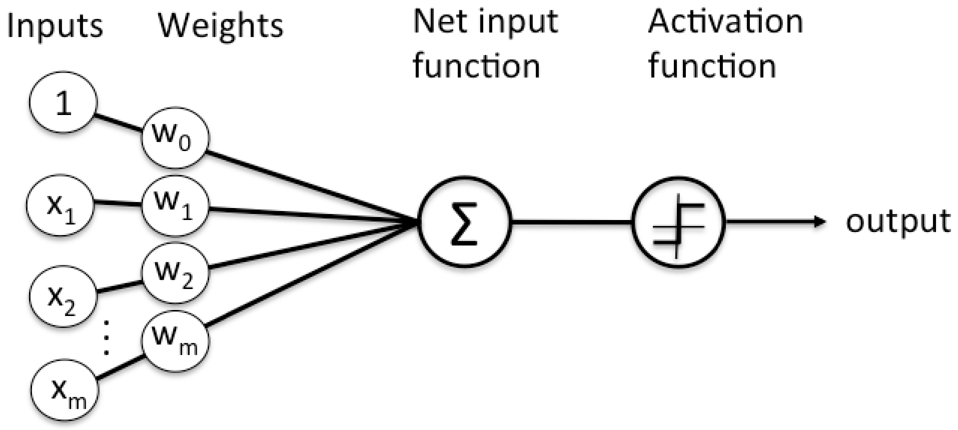
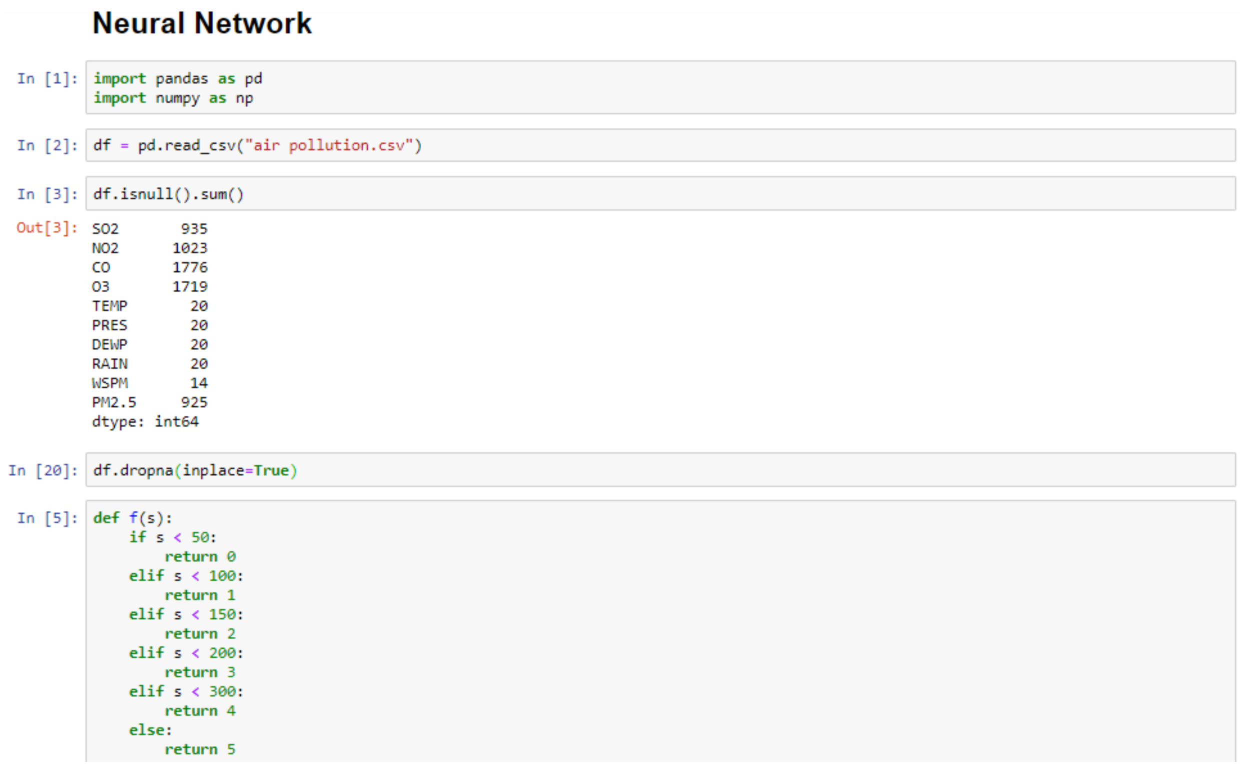


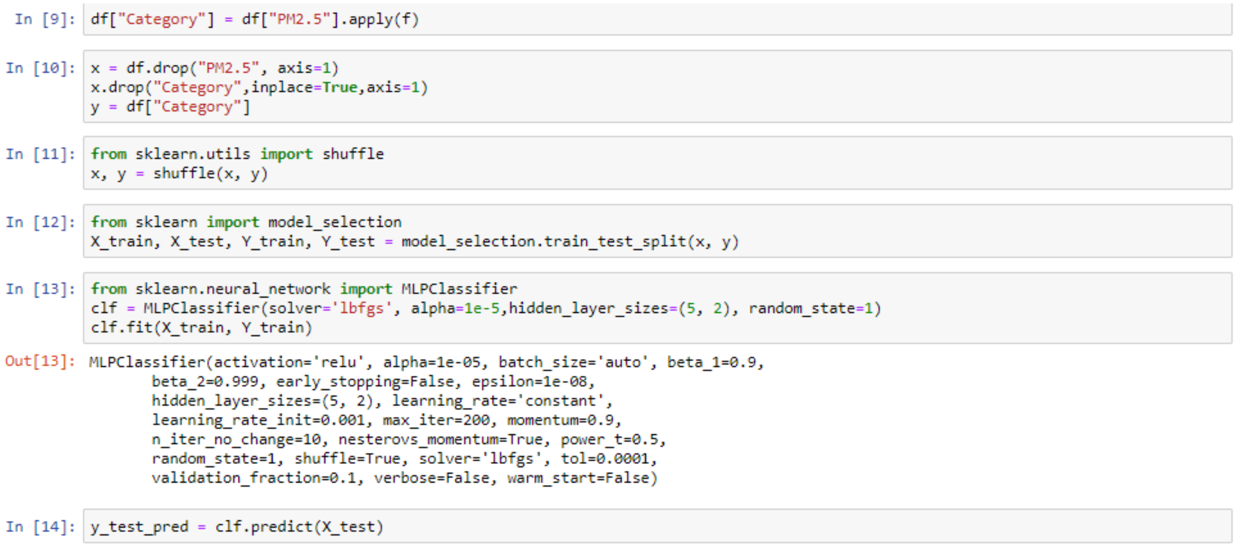
Fig 3.5.4 Neural Network Layer

key level consist of participation columns that are essentially derivative from the datasets of AQI. The key level consist of lots of feature of the AQ. The key level will be related to some unseen layer. The unseen level will set off by on to a little more unseen level and in the end level. The last level will essentially guess the likelihood. The possibility will be resolute by sigmoid func. or slightly by a softmax function. So, the key tensor resolve be the figure of enter and in among, there are lots of unseen level. It is major to present the outline of effort to the tensor or as well it will not be capable to guess the amswer. The key level is basically judge by the recurring neural system. The recurring neural system consist of long short-range recollection cell. The information is controlled within the CSV folder which is Air Quality prediction. information level comprise in order contain that are mostly gotten starting the old capitulate of securities substitute opinion. The information level encompass of several old places of interest of a securities switch. The in rank level will be connected with some unknown layer. The cloaked level will bypass on to a little more unseen level and at last the yield layer. The yield level will mostly foresee the possibility. The probability will be dictate by sigmoid beginning work or rather by softmax work. In this way, the information tensor will be the situation of in turn and in the focus of there are many unseen layers. It is necessary to give the situation of input to the tensor else it would not have the capability to await the yield. The info level is essentially made a result by the alternating neural system. The alternating neural system encompass a long temporary memory cell. The info is controlled inside the CSV file which is older securities substitute.



Neural Network

```
In [1]:  import pandas as pd
         import numpy as np

In [2]:  df = pd.read_csv("air pollution.csv")

In [3]:  df.isnull().sum()

Out[3]:  SO2       935
         NO2      1023
         CO       1776
         O3       1719
         TEMP       20
         PRES       20
         DEWP       20
         RAIN       20
         WSPM       14
         PM2.5     925
         dtype: int64

In [20]: df.dropna(inplace=True)

In [5]:  def f(s):
             if s < 50:
                 return 0
             elif s < 100:
                 return 1
             elif s < 150:
                 return 2
             elif s < 200:
                 return 3
             elif s < 300:
                 return 4
             else:
                 return 5
```



```
In [9]:  df["Category"] = df["PM2.5"].apply(f)

In [10]: x = df.drop("PM2.5", axis=1)
         x.drop("Category",inplace=True,axis=1)
         y = df["Category"]

In [11]: from sklearn.utils import shuffle
         x, y = shuffle(x, y)

In [12]: from sklearn import model_selection
         X_train, X_test, Y_train, Y_test = model_selection.train_test_split(x, y)

In [13]: from sklearn.neural_network import MLPClassifier
         clf = MLPClassifier(solver='lbfgs', alpha=1e-5,hidden_layer_sizes=(5, 2), random_state=1)
         clf.fit(X_train, Y_train)

Out[13]: MLPClassifier(activation='relu', alpha=1e-05, batch_size='auto', beta_1=0.9,
                beta_2=0.999, early_stopping=False, epsilon=1e-08,
                hidden_layer_sizes=(5, 2), learning_rate='constant',
                learning_rate_init=0.001, max_iter=200, momentum=0.9,
                n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5,
                random_state=1, shuffle=True, solver='lbfgs', tol=0.0001,
                validation_fraction=0.1, verbose=False, warm_start=False)

In [14]: y_test_pred = clf.predict(X_test)
```

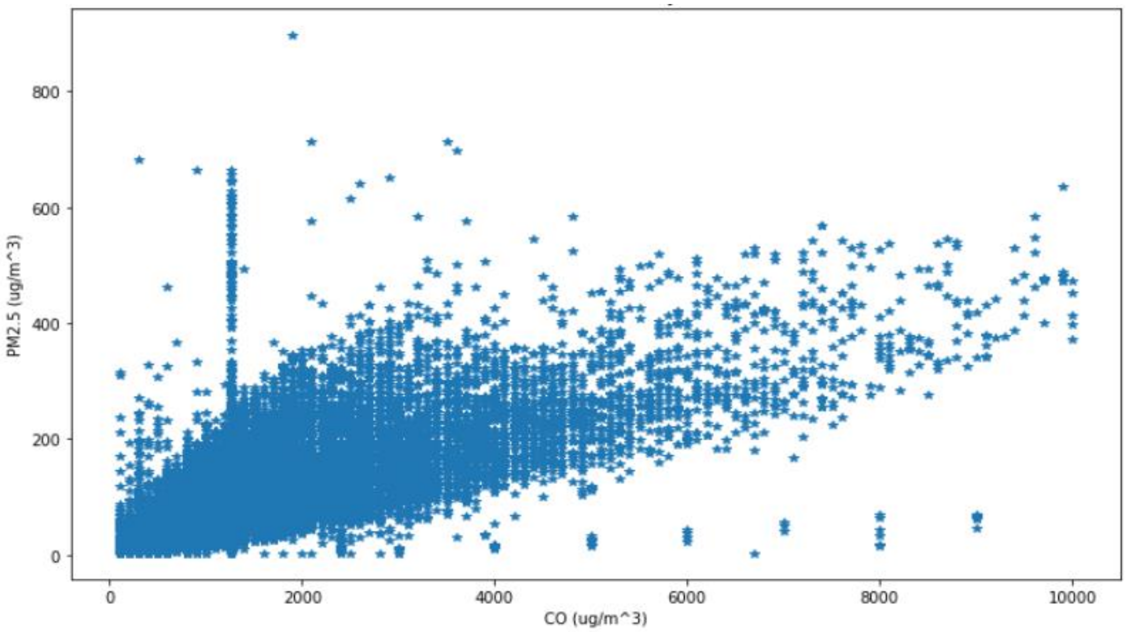
Chapter – 4

Perfomance Analysis

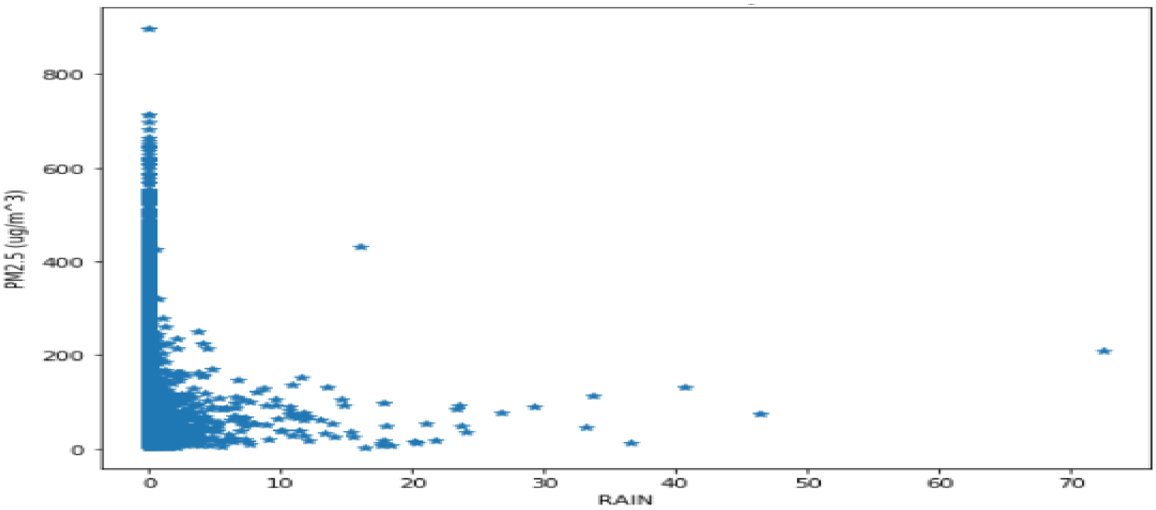
In this chapter we will do the analysis between various features of our dataset with target column and also compare the result of machine learning algorithms.

4.1 Bivariate Analysis

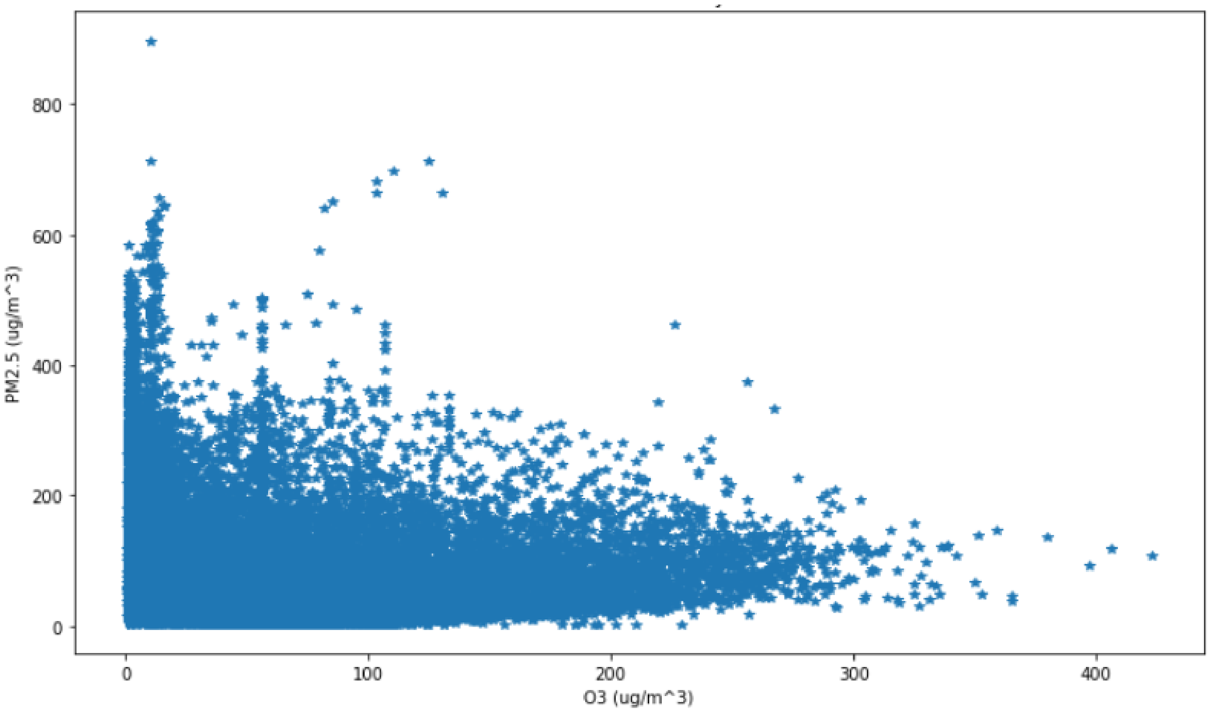
4.1.1 Analysis between Carbon Monoxide (CO) & PM2.5 Level



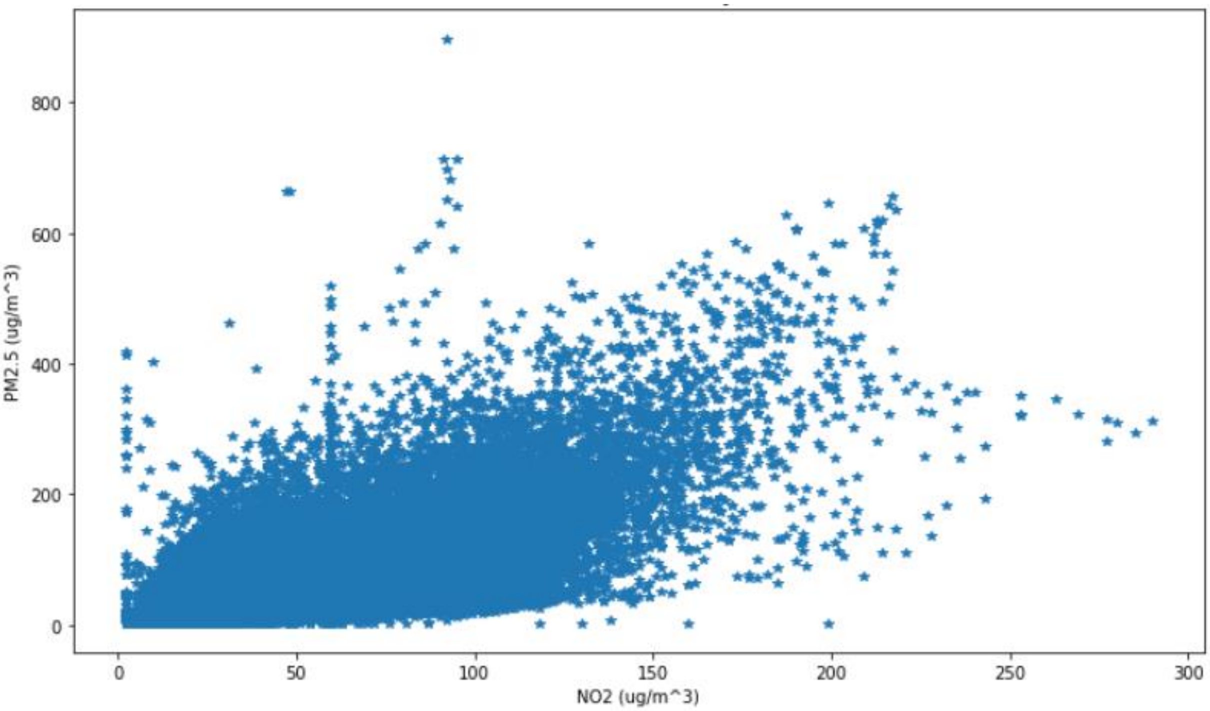
4.1.2 Analysis between Rain & PM2.5 Level



4.1.3 Analysis between O3 & PM2.5 Level

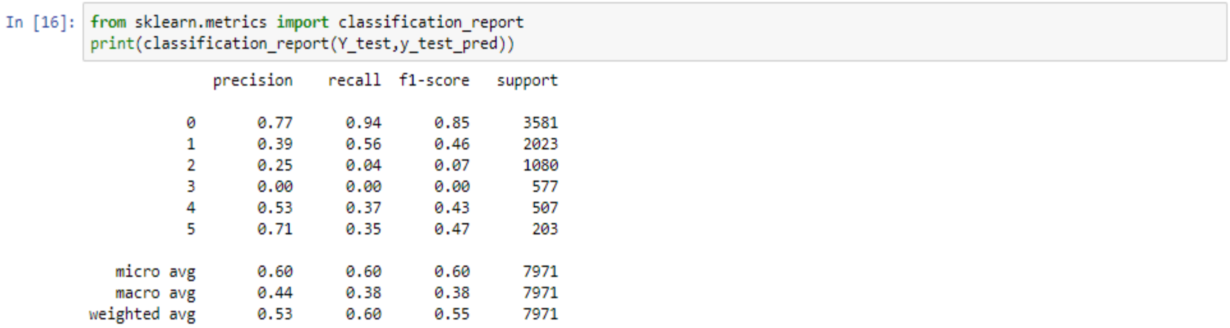


4.1.4 Analysis between NO2 & PM2.5 Level

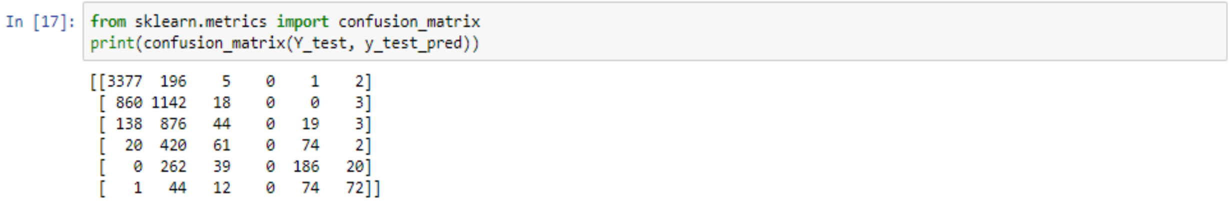


4.2 Comparision of Results

4.2.1 Output of Logistic Regression

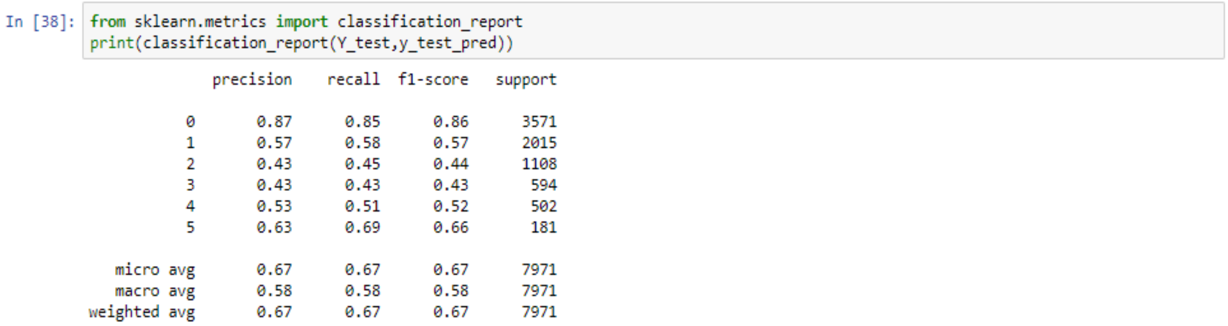


```
In [16]: from sklearn.metrics import classification_report
         print(classification_report(Y_test,y_test_pred))

              precision    recall  f1-score   support

           0       0.77      0.94      0.85      3581
           1       0.39      0.56      0.46      2023
           2       0.25      0.04      0.07      1080
           3       0.00      0.00      0.00       577
           4       0.53      0.37      0.43       507
           5       0.71      0.35      0.47       203

   micro avg       0.60      0.60      0.60      7971
   macro avg       0.44      0.38      0.38      7971
weighted avg       0.53      0.60      0.55      7971
```

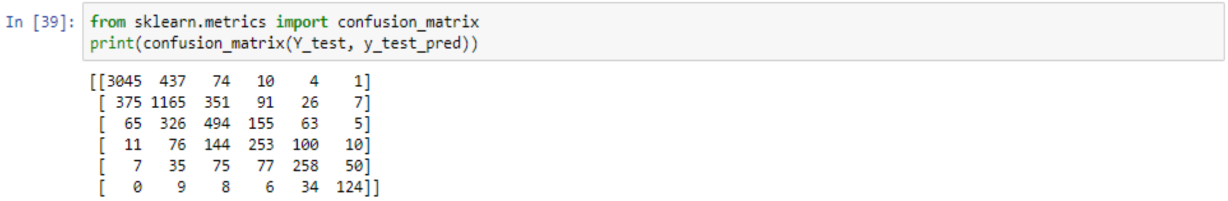


```
In [17]: from sklearn.metrics import confusion_matrix
         print(confusion_matrix(Y_test, y_test_pred))

[[3377  196    5    0    1    2]
 [ 860 1142   18    0    0    3]
 [ 138  876   44    0   19    3]
 [  20  420   61    0   74    2]
 [   0  262   39    0  186   20]
 [   1   44   12    0   74   72]]
```

4.2.2 Output of Decision Tree

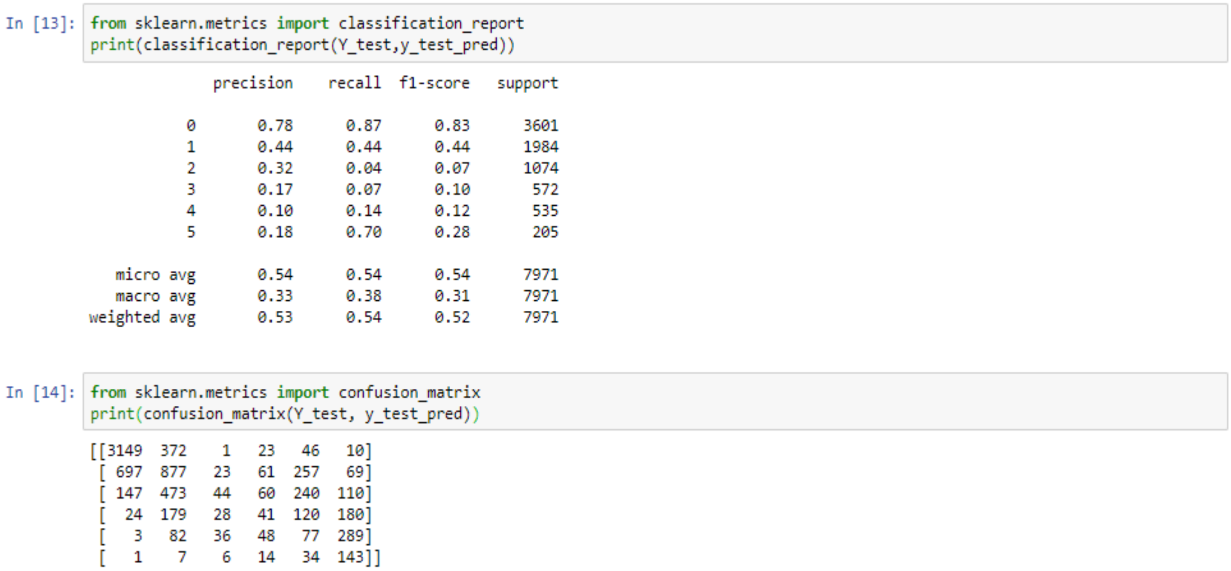


```
In [38]: from sklearn.metrics import classification_report
         print(classification_report(Y_test,y_test_pred))

              precision    recall  f1-score   support

           0       0.87      0.85      0.86      3571
           1       0.57      0.58      0.57      2015
           2       0.43      0.45      0.44      1108
           3       0.43      0.43      0.43       594
           4       0.53      0.51      0.52       502
           5       0.63      0.69      0.66       181

   micro avg       0.67      0.67      0.67      7971
   macro avg       0.58      0.58      0.58      7971
weighted avg       0.67      0.67      0.67      7971
```



```
In [39]: from sklearn.metrics import confusion_matrix
         print(confusion_matrix(Y_test, y_test_pred))

[[3045  437   74   10    4    1]
 [ 375 1165  351   91   26    7]
 [  65  326  494  155   63    5]
 [  11   76  144  253  100   10]
 [   7   35   75   77  258   50]
 [   0    9    8    6   34  124]]
```

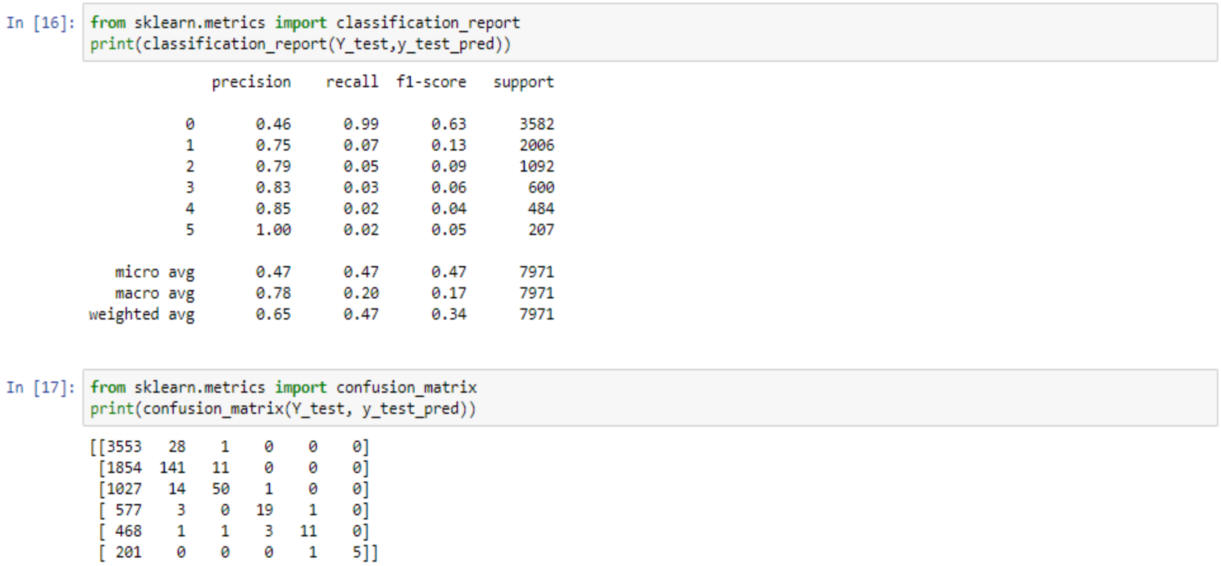
4.2.3 Output of Naïve Bayes



```
In [13]: from sklearn.metrics import classification_report
         print(classification_report(Y_test,y_test_pred))

                      precision    recall  f1-score   support

                   0       0.78      0.87      0.83      3601
                   1       0.44      0.44      0.44      1984
                   2       0.32      0.04      0.07      1074
                   3       0.17      0.07      0.10       572
                   4       0.10      0.14      0.12       535
                   5       0.18      0.70      0.28       205

           micro avg       0.54      0.54      0.54      7971
           macro avg       0.33      0.38      0.31      7971
        weighted avg       0.53      0.54      0.52      7971
```

```
In [14]: from sklearn.metrics import confusion_matrix
         print(confusion_matrix(Y_test, y_test_pred))

         [[3149  372    1   23   46   10]
          [ 697  877   23   61  257   69]
          [ 147  473   44   60  240  110]
          [  24  179   28   41  120  180]
          [   3   82   36   48   77  289]
          [   1    7    6   14   34  143]]
```

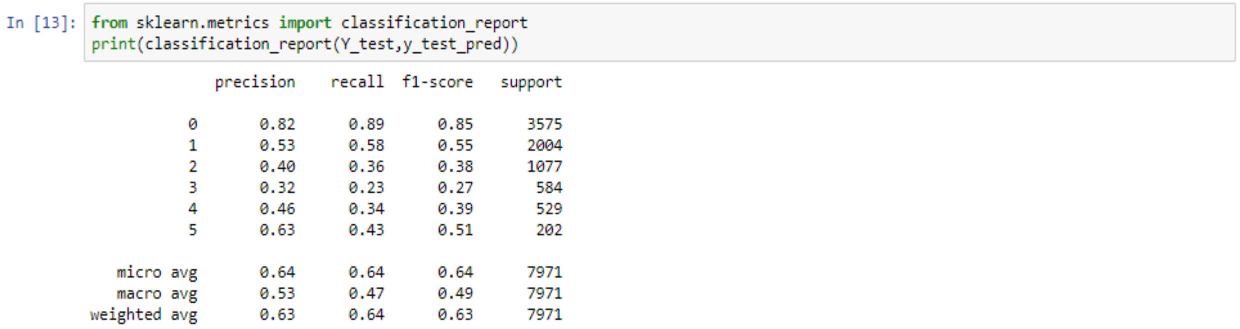
4.2.4 Output of SVM



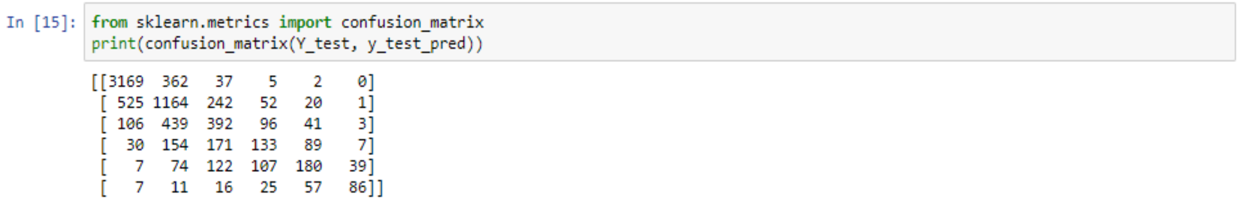
```
In [16]: from sklearn.metrics import classification_report
         print(classification_report(Y_test,y_test_pred))

                      precision    recall  f1-score   support

                   0       0.46      0.99      0.63      3582
                   1       0.75      0.07      0.13      2006
                   2       0.79      0.05      0.09      1092
                   3       0.83      0.03      0.06       600
                   4       0.85      0.02      0.04       484
                   5       1.00      0.02      0.05       207

           micro avg       0.47      0.47      0.47      7971
           macro avg       0.78      0.20      0.17      7971
        weighted avg       0.65      0.47      0.34      7971
```

```
In [17]: from sklearn.metrics import confusion_matrix
         print(confusion_matrix(Y_test, y_test_pred))

         [[3553   28    1    0    0    0]
          [1854  141   11    0    0    0]
          [1027   14   50    1    0    0]
          [ 577    3    0   19    1    0]
          [ 468    1    1    3   11    0]
          [ 201    0    0    0    1    5]]
```

4.2.5 Output of KNN

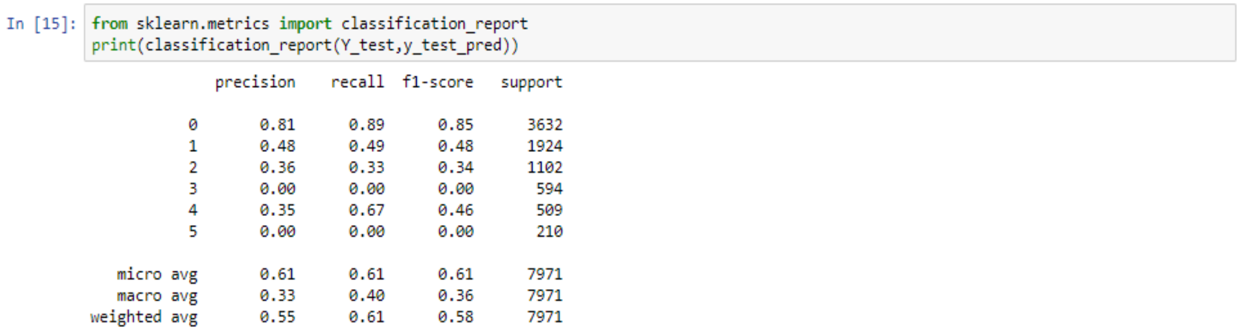


```
In [13]: from sklearn.metrics import classification_report
         print(classification_report(Y_test,y_test_pred))

                      precision    recall  f1-score   support

                   0       0.82      0.89      0.85      3575
                   1       0.53      0.58      0.55      2004
                   2       0.40      0.36      0.38      1077
                   3       0.32      0.23      0.27       584
                   4       0.46      0.34      0.39       529
                   5       0.63      0.43      0.51       202

           micro avg       0.64      0.64      0.64      7971
           macro avg       0.53      0.47      0.49      7971
        weighted avg       0.63      0.64      0.63      7971
```

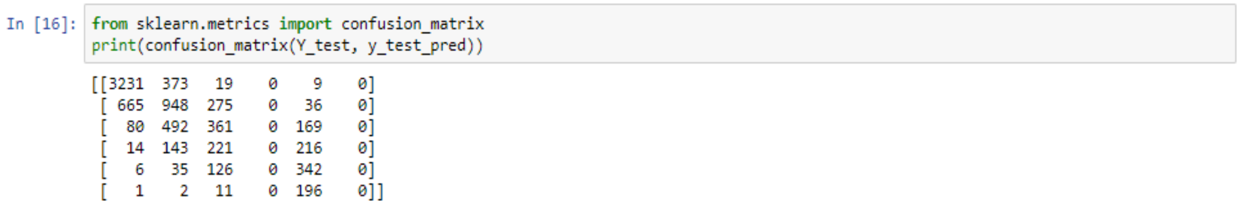


```
In [15]: from sklearn.metrics import confusion_matrix
         print(confusion_matrix(Y_test, y_test_pred))

         [[3169  362   37    5    2    0]
          [ 525 1164  242   52   20    1]
          [ 106  439  392   96   41    3]
          [  30  154  171  133   89    7]
          [   7   74  122  107  180   39]
          [   7   11   16   25   57   86]]
```

4.2.6 Output of Neural Network



```
In [15]: from sklearn.metrics import classification_report
         print(classification_report(Y_test,y_test_pred))

                      precision    recall  f1-score   support

                   0       0.81      0.89      0.85      3632
                   1       0.48      0.49      0.48      1924
                   2       0.36      0.33      0.34      1102
                   3       0.00      0.00      0.00       594
                   4       0.35      0.67      0.46       509
                   5       0.00      0.00      0.00       210

           micro avg       0.61      0.61      0.61      7971
           macro avg       0.33      0.40      0.36      7971
        weighted avg       0.55      0.61      0.58      7971
```



```
In [16]: from sklearn.metrics import confusion_matrix
         print(confusion_matrix(Y_test, y_test_pred))

         [[3231  373   19    0    9    0]
          [ 665  948  275    0   36    0]
          [  80  492  361    0  169    0]
          [  14  143  221    0  216    0]
          [   6   35  126    0  342    0]
          [   1    2   11    0  196    0]]
```

Chapter – 5

Conclusions

All the ML algorithms linear regressions, polynomial linear regressions, logistic regressions, random forests regressions and Artificially neural system were compressed by proficient environment decisive apparatus, inspite of the information that the fault in their implementation compact considerably for later on day, representative that over long timeframe, our model may compress brilliance specialized ones. Linear regressions confirmed to be a low predilection, higher variation models though polynomial regressions confirmed to be a higher predilection, lower dissimilarity models. Linear regressions is usually a higher dissimilarity model as it is unsteady to outliers, so one method to get better the linear regressions models is by assembelling of more knowledge. Experimental regression, however, was high disposition, representative that the conclusion of model was poor, and that its prediction can not be better by additional gathering of knowledge. This disposition can be predictable to the understanding choice to predict environment reliant on the environment of the preceding 2 day, which may be too diminutive to even believe about capture slant in environment that practical regressions require. On the off chance that the figures were rather found on the atmosphere of the past or five days, the disposition of the experimental regressions models can perhaps be decrease. In case, this require considerably more computation time next to retraining of the influence vector w, so this will be conceded to future works. Governments can take the following measures to reduce the amount of pm2.5

- road traffic ration,
- rising green grass along side road,
- spray water on tree nearby the road,
- vacuum clean of roads,
- cleaning of rivers and river bodies,
- firm instruction to construction company concerning pollution,
- decreasing carbon emission by focusing on renewable energy generation,
- ban on sale of firecracker,
- crackdown on burning of solid wastes and agricultural wastes,
- influencing businesses to invest on solution relating to environmental sustainability,
- issuing alert and closing down school and college,
- improving public transport systems

References

- https://www.researchgate.net/publication/
- http://cs229.stanford.edu/proj2017/final-reports/5234854.pdf
- https://pythonprogramming.net
- https://pypi.org/project/pandas/
- https://matplotlib.org
- https://www.google.co.in/amp/s/www.geeksforgeeks.org/numpy-in-python-set-1- introduction/amp/
- https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/
- https://towardsdatascience.com/machine-learning/home