AIR QUALITY PREDICTION

A mini project report submitted to

GANDHI INSTITUTE OF TECHNOLOGY AND MANAGEMENT(GITAM)

in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

submitted by

AKULA SAI TEJA 2215316465

under the guidance of

Mrs. T. Aruna Sri(M. Tech)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF TECHNOLOGY

GANDHI INSTITUTE OF TECHNOLOGY AND MANAGEMENT(GITAM)

(Declared as Deemed-to-be-University u/s 3 of UGC Act 1956)

HYDERABAD CAMPUS

OCTOBER-2019

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF TECHNOLOGY

GANDHI INSTITUTE OF TECHNOLOGY AND MANAGEMENT(GITAM)

(Declared as Deemed-to-be-University u/s 3 of UGC Act 1956)

HYDERABAD CAMPUS

DECLARATION

I hereby declare that the mini project report **"AIR QUALITY PREDICTION"** submitted to GITAM (Deemed-to-be-University), Hyderabad in partial fulfillment of the requirements for the award of the degree of "Bachelor of Technology" in "Computer Science and Engineering" is an original work carried out by us and has not been submitted earlier to this or any other university.

AKULA SAI TEJA

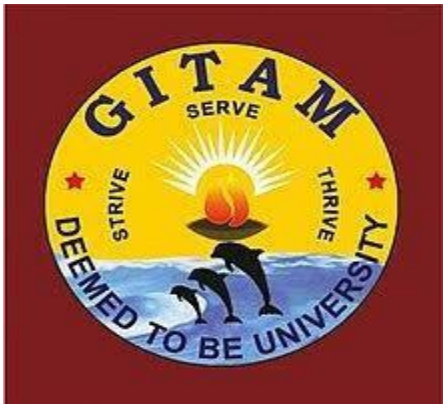
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF TECHNOLOGY

GANDHI INSTITUTE OF TECHNOLOGY AND MANAGEMENT(GITAM)

(Declared as Deemed-to-be-University u/s 3 of UGC Act 1956)

HYDERABAD CAMPUS



CERTIFICATE

This is to certify that the project report **"AIR QUALITY PREDICTION"** is submitted by **AKULA SAI TEJA (2215316465),** in partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Mrs. T. Aruna Sri **Prof. S. Phani Kumar**

(Project Guide) **(Project Coordinator)** **Head of the Department**

PANEL MEMBERS

1.

2.

3.

ACKNOWLEDGEMENT

Our project would not have been successful without the help of several people. I would like to thank the personalities who were part of our project in numerous ways, those who gave us outstanding support from the birth of the project.

I am extremely thankful to our honorable Pro-Vice Chancellor, Prof. N.Siva Prasad for providing necessary infrastructure and resources for the accomplishment of our project.

I am highly indebted to Prof. N. Seetharamaiah, Principal, School of Technology, for his support during the tenure of the project.

I am very much obliged to our beloved Prof.S.Phani Kumar, Head of the Department of Computer Science & Engineering for providing the opportunity to undertake this project and encouragement in completion of this project.

I hereby wish to express our deep sense of gratitude to Mrs. T. Aruna Sri, Department of Computer Science and Engineering, School of Technology for the esteemed guidance, moral support and invaluable advice provided by him for the success of the project.

I am also thankful to all the staff members of Computer Science and Engineering department who have co operated in making our project a success. I would like to thank all our parents and friends who extended their help, encouragement and moral support either directly or indirectly in our project work.

Sincerely,

AKULA SAI TEJA

Contents

ABSTRACT 4

List of Figures 5

List of Tables 6

INTRODUCTION 7

1.1 Background 7

1.2 Machine Learning 7

1.3 Objective and Goals of the Project 8

1.4 Limitations of the Project 8

LITERATURE SURVEY 9

2.1 Air Quality Index-PM2.5 9

2.2 Machine Learning Algorithms 10

2.2.1 Multiple Linear Regression Model 10

2.3.2 Logistic Regression Model 11

2.2.3 Naïve Bayes Classifier 12

REQUIREMENT ANALYSIS 14

3.1 Technical Requirements 14

3.2 Hardware Requirements 15

DESIGN OF THE PROJECT 16

4.1 Design 16

4.1.1 System Architecture 16

4.1.2 Machine Learning Process 16

4.1.3 User interface 18

4.2 Methodology 18

4.2.1 Data Collection 18

4.2.2 Importing Packages that are Necessary 20

4.2.3 Pre-processing and Data Cleaning 21
4.2.4 Correlation Analysis 22
4.2.5 Reducing Dimensionality 23
4.2.6 Train-Test-Split 24
4.3 Algorithm 24
4.3.1 Multiple Linear Regression 24
4.3.2 Logistic Regression 26
4.3.3 Naive Bayes classifier 27
4.4 Model evaluation 29
4.5 User-Interface Design 30
4.5.1 Electron.js 30
4.5.2 Dependencies 30
IMPLEMENTATION 31
5.1 Tools Used 31
5.1.1 Anaconda-Jupyter Notebooks 31
5.1.2 Visual Studio Code 32
5.1.3 Electron.js and Node.js 33
5.2 User-Interface concept 34
DEPLOYING THE PROJECT 35
6.1 Electron packager 35
6.2 User Interface 35
PREDICTIONS AND INSIGHTS 39
7.1 By Manually Giving Parameters 39
7.3 Conclusion 40
FUTURE WORK AND SCOPE 41
8.1 Air Quality Index 41

8.2 Real Time Predictions 41

REFERENCES 42

ABSTRACT

AIR QUALITY PREDICTION

Air Pollution is one of the most important factors that affects the environment and habitat. Many Metropolitan cities suffer from heavy pollution due to various reasons that have some serious effects on health, especially children. Air Pollution arises due to bad quality of air that is polluted by vehicular emission, Industrial effluents, CO, etc.

The objective is to analyse the quality of Air using Machine Learning Algorithms and Python. The Metric PM2.5 is used for grading the quality of air, The higher the value, the more the bad quality of air, similarly lower the value better the quality of air.

The Analysis is performed on collected datasets for extracting some insights. Furthermore, we go on to build a model that has high accuracy along with good metrics for evaluation. The Algorithms like Logistic regression and Naïve Bayes Classifier were used to predict the Air quality. Once the model is built we deploy it using electron.js and node.js. So that a user can interact with our model just by giving the parameters and will end up the PM2.5 value. This PM2.5 value tells whether the air is polluted or not.

List of Figures

Fig 1.1 : Machine Learning 7

Fig 2.1 : Multiple Linear Regression 11

Fig 2.2 : Logistic Regression 12

Fig 4.1 Machine Learning and Deployment 16

Fig 4.2 : Machine Learning Process 18

Fig 4.3 Weather dataset 19

Fig 4.4 Importing Necessary Packages and the Dataset 21

Fig 4.5 Null Values imputed! 22

Fig 4.6 Heatmap is visual representation of Correlation 23

Fig 4.7 Feature selection to reduce dimensionality 25

Fig 4.8 Model initialization and its accuracy 25

Fig 4.9 Transformed target variable 26

Fig 4.11 Transformed data 28

Fig 4.12 Cross Validation scores 29

Fig 5.1 Anaconda Jupyter notebook Homepage 31

Fig 5.2 A project in Jupyter Notebook 32

Fig 5.3 Workspace 33

Fig 6.1 Home Tab 36

Fig 6.2 About Tab 36

Fig 6.3 Predict Tab 37

Fig 6.4. Requirements Tab 37

Fig 7.1 Output results for the above parameters using MLR 39

Fig 7.2 Output results for the above parameters using LR 39

List of Tables

Table 7.1 Output results for Random Dates chosen on September 32

INTRODUCTION

1.1 Background

“Most air pollution comes from energy use and production,” says John Walke, director of the Clean Air Project, part of the Climate and Clean Air program at NRDC. “Burning fossil fuels releases gases and chemicals into the air.” And in an especially destructive feedback loop, air pollution not only contributes to climate change but is also exacerbated by it.

“Air pollution in the form of carbon dioxide and methane raises the earth’s temperature,” Walke says. “Another type of air pollution is then worsened by that increased heat: Smog forms when the weather is warmer and there’s more ultraviolet radiation.”

Climate change also increases the production of allergenic air pollutants including mold (thanks to damp conditions caused by extreme weather and increased flooding) and pollen (due to a longer pollen season and more pollen production).

1.2 Machine Learning

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

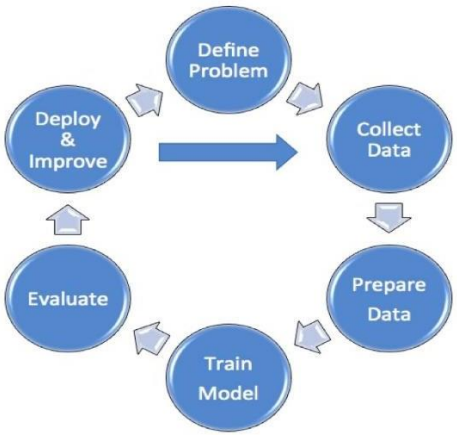


Fig 1.1 : Machine Learning

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

1.3 Objective and Goals of the Project

- To Build a Predictive model that can predict the Amount of PM2.5.
- To Build an accurate model.

1.4 Limitations of the Project

- We limit ourselves to the Location: ICRISAT, Patancheru.
- The time period is 1 year 6months, ICRISAT Station has been collecting the PM2.5 data since 2018 JAN.
- Our Predictions are limited to the Data of ICRISAT.

LITERATURE SURVEY

2.1 Air Quality Index-PM2.5

The AQI is an index for reporting daily air quality. It tells you how clean or polluted your air is, and what associated health effects might be a concern for you. The AQI focuses on health affects you may experience within a few hours or days after breathing polluted air. EPA calculates the AQI for five major air pollutants regulated by the Clean Air Act: ground-level ozone, particle pollution (also known as particulate matter), carbon monoxide, Sulphur dioxide, and nitrogen dioxide. For each of these pollutants, EPA has established national air quality standards to protect public health. Ground-level ozone and airborne particles are the two pollutants that pose the greatest threat to human health in this country.

Each category corresponds to a different level of health concern. The six levels of health concern and what they mean are:

- "Good" AQI is 0 to 50. Air quality is considered satisfactory, and air pollution poses little or no risk.
- "Moderate" AQI is 51 to 100. Air quality is acceptable; however, for some pollutants there may be a moderate health concern for a very small number of people. For example, people who are unusually sensitive to ozone may experience respiratory symptoms.
- "Unhealthy for Sensitive Groups" AQI is 101 to 150. Although general public is not likely to be affected at this AQI range, people with lung disease, older adults and children are at a greater risk from exposure to ozone, whereas persons with heart and lung disease, older adults and children are at greater risk from the presence of particles in the air.
- "Unhealthy" AQI is 151 to 200. Everyone may begin to experience some adverse health effects, and members of the sensitive groups may experience more serious effects.
- "Very Unhealthy" AQI is 201 to 300. This would trigger a health alert signifying that everyone may experience more serious health effects.
- "Hazardous" AQI greater than 300. This would trigger a health warnings of emergency conditions. The entire population is more likely to be affected.

The term fine particles, or particulate matter 2.5 (PM2.5), refers to tiny particles or droplets in the air that are two- and one-half microns or less in width. Like inches, meters and miles, a micron is a unit of measurement for distance. There are about 25,000 microns in an inch. The

widths of the larger particles in the PM2.5 size range would be about thirty times smaller than that of a human hair. The smaller particles are so small that several thousand of them could fit on the period at the end of this sentence.

2.2 Machine Learning Algorithms

The Most Important factor is that we choose a correct Model for our Predictive Model. It is possible to change the Type of data *i.e.* Data Manipulation to fit in with Regression or Classification Models.

Regression models are best suited for numerical data to target a numerical Variable. Initially our data consist of numerical data only.

2.2.1 Multiple Linear Regression Model

Multiple Linear Regression is simply a heavier version of Linear Regression.

In Linear Regression the there is only one Dependent Variable and one Independent Variable. The Equation is

$$y = a + bx \quad \text{...equation (1)}$$

Where, y is the dependant variable and x is the independent variable. ‘a’ and ‘b’ slopes and intercepts of the model respectively.

For instance, here is the equation for multiple linear regression with two independent variables:

$$Y = a0x0 + a1x1 + \cdots anxn + b \quad \text{...equation (2)}$$

This holds true for any given number of variables.

Multivariate linear regression can be thought as multiple regular linear regression models, since you are just comparing the correlations between features for the given number of features.

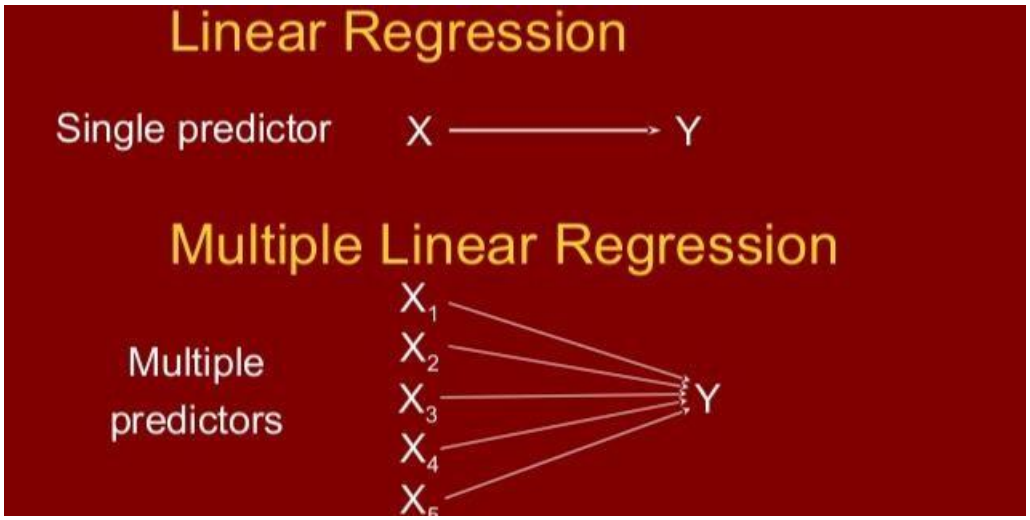


Fig 2.1 : Multiple Linear Regression

Drawback: A linear regression is not appropriate for predicting the value of a binary variable for two reasons:

A linear regression will predict values outside the acceptable range (e.g. predicting probabilities outside the range 0 to 1). Since the dichotomous experiments can only have one of two possible values for each experiment, the residuals will not be normally distributed about the predicted line.

2.3.2 Logistic Regression Model

Even though the model is named Regression it is Logistic Regression is a Classification Model. Best suited when the Target is dichotomous (binary Class Labels) *i.e. a Yes or no question.*

$$p = \frac{1}{1 + e^{-(b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p)}} \quad \textbf{....equation(3)}$$

Simply speaking, we used the sigmoid function over the MLR equation.

The logistic function, also called the sigmoid function was developed by statisticians to describe properties of population growth in ecology, rising quickly and maxing out at the

carrying capacity of the environment. It's an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.

$$P=1 / (1 + e^{-Y}) \text{ .....equation(4)}$$

Where e is the base of the natural logarithms.

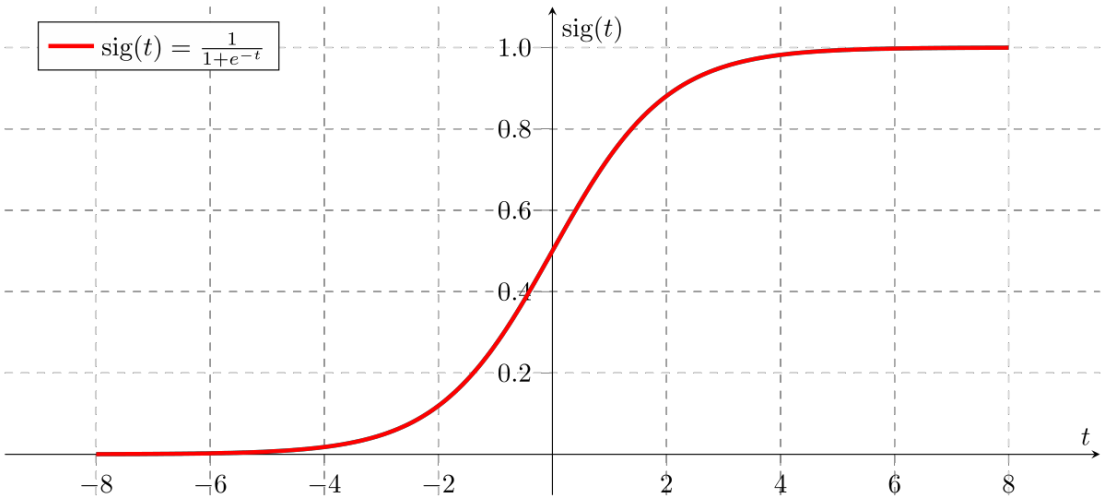


Fig 2.2 : Logistic Regression

2.2.3 Naïve Bayes Classifier

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

When to use:

Moderate or large training set available.

Attributes that describe instances are conditionally independent given classification.

Bayes' Theorem is stated as:

$$P(h|d) = (P(d|h) * P(h)) / P(d) \text{ …equation(5)}$$

Where,

P(h|d) is the probability of hypothesis h given the data d. This is called the posterior probability.

P(d|h) is the probability of data d given that the hypothesis h was true.

P(h) is the probability of hypothesis h being true (regardless of the data). This is called the prior probability of h.

P(d) is the probability of the data (regardless of the hypothesis).

REQUIREMENT ANALYSIS

3.1 Technical Requirements

DATASET-for Training Our Model, we will need the data to train and understand the nature of the data.

Microsoft Excel- is a spreadsheet manipulation tool that is used to organize and manipulate data, in our project, we use it for Data Cleaning and Pre-processing.

Machine Learning Algorithms-for building our model like Regression or classification.

Python 3.7 Packages-

NumPy: for the data handling and operations. NumPy is the fundamental package for scientific computing with Python.

Pandas: is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Pandas is a NumFOCUS sponsored project.

Sci-kit: for Machine Learning Algorithms, this package has many tools that help implementing the machine learning process like train-test split, initialization of the model.

Matplotlib: for Data Visualization, it is easy to plot graphs using matplotlib and visual representation of data helps understanding the nature of data.

Electron.js and Node.js: for the deployment and User Interface. Both these technologies are powerful and useful in creating a full-stack application.

SweetAlert: for Alerts is a new package for interactive dialogue boxes.

3.2 Hardware Requirements

To Run Python a minimum of 4GB Ram is required. Anything Below 4GB would result in huge differences in results and also Slows down our work. A Minimum Integrated GPU should be available in the systems that will help understand the visualizations better.

DESIGN OF THE PROJECT

4.1 Design

4.1.1 System Architecture

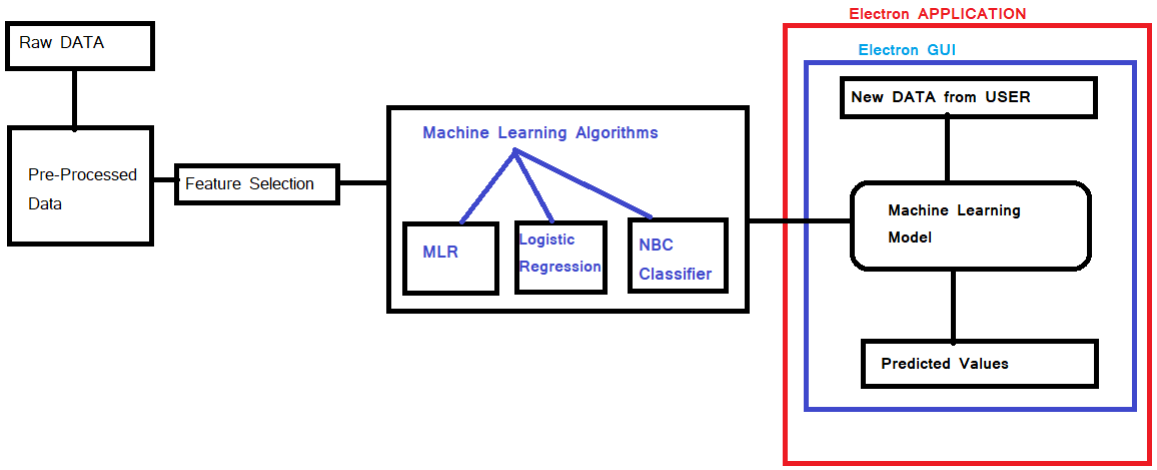


Fig 4.1 Machine Learning and Deployment

The idea is to collect data about weather and pm2.5 readings and to find the factors that affect the PM2.5 values.

Air Quality data can be collected from the nearby embassy. Weather data from the nearest Weather Station. Once we have raw data, we must clean it to our desire i.e. remove or handle null values etc. Once the Data is cleaned it is Machine Learning ready. Before we build our model, we must extract features. Apply any Machine learning algorithm to the data. Compare metrics with different train_test split ratios. Choose the best model. Predict Values from new data.

4.1.2 Machine Learning Process

Data Collection: The quantity & quality of your data dictate how accurate our model is.The outcome of this step is generally a representation of data (Guo simplifies to specifying a table)

which we will use for training. Using pre-collected data, by way of datasets from Kaggle, UCI, etc., still fits into this step.

Data Preparation: Wrangle data and prepare it for training. Clean that which may require it (remove duplicates, correct errors, deal with missing values, normalization, data type conversions, etc.) Randomize data, which erases the effects of the particular order in which we collected and/or otherwise prepared our data. Visualize data to help detect relevant relationships between variables or class imbalances (bias alert!), or perform other exploratory analysis. Split into training and evaluation sets.

Choose a Model: Different algorithms are for different tasks; choose the right one.

Train the Model: The goal of training is to answer a question or make a prediction correctly as often as possible. Linear regression example: algorithm would need to learn values for m (or W) and b (x is input, y is output). Each iteration of process is a training step

Evaluate the Model: Uses some metric or combination of metrics to "measure" objective performance of model. Test the model against previously unseen data. This unseen data is meant to be somewhat representative of model performance in the real world, but still helps tune the model (as opposed to test data, which does not).Good train/eval split? 80/20, 70/30, or similar, depending on domain, data availability, dataset particulars, etc.

Parameter Tuning: This step refers to *hyperparameter* tuning, which is an "artform" as opposed to a science. Tune model parameters for improved performance. Simple model hyperparameters may include: number of training steps, learning rate, initialization values and distribution, etc.

Make Predictions: Using further (test set) data which have, until this point, been withheld from the model (and for which class labels are known), are used to test the model; a better approximation of how the model will perform in the real world

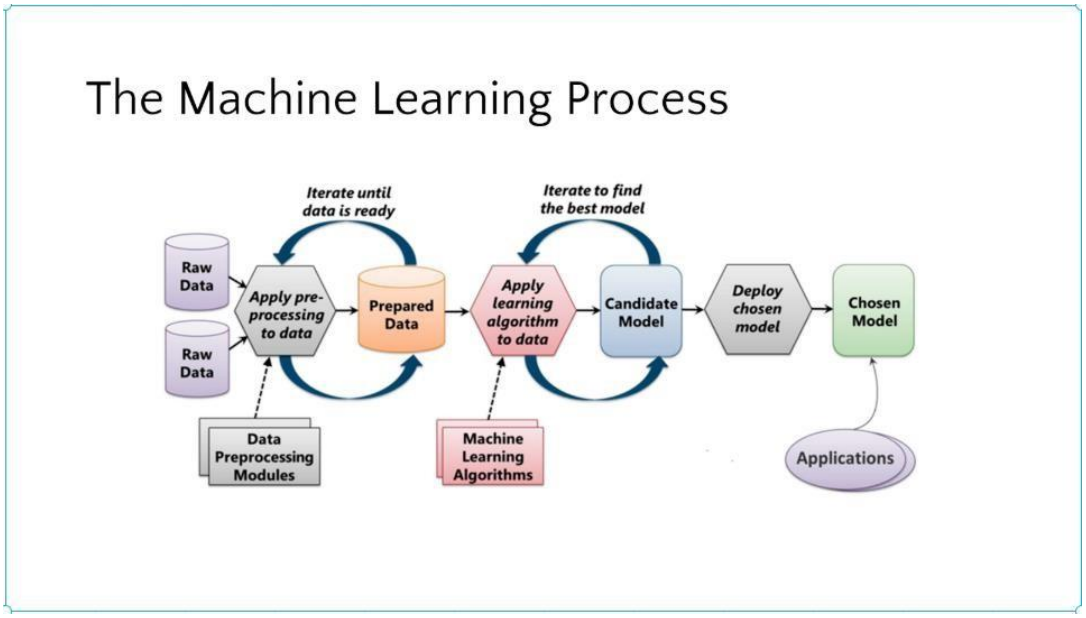


Fig 4.2 : Machine Learning Process

The Entire Machine Learning Process is iterative until a desired accuracy.

4.1.3 User interface

At this Point, we will have a model(s) now, the next step is to make it user friendly or easily interpretable by a common man. We need a front-end framework to integrate our Model.

We have several different ways to do this:

- Node-red from IBM
- AWS Sage maker
- Google cloud Platform's ML services
- Electron.js
- Flutter.js

We choose Electron.js for the User Interface as it is highly robust and has many features.

4.2 Methodology

4.2.1 Data Collection

Data collection is the process of gathering and measuring information on targeted Variables in an established system, which then enables one to answer relevant questions and evaluate

outcomes. Data collection is a component of research in all fields of study including physical and social sciences, humanities and business.

The DATA:

We Found our data and statistics at cpcb.nic.in and opeaq.org. CPCB collects and publishes data related to the Environment, Weather etc. Openaq.org is one of the largest AQI data repositories that publishes data from all over the world.We Took our Data from the ICRISAT Station at Patancheru.

Weather DATASET-

We have General Parameters like Rain, Temperature, Wind Velocity, Relative Humidity etc. These Factors are analysed Using Different Analysis Methods using Machine Learning With Python 3.

1. Date- The date of the recording
2. Stdweek- the week of the recording
3. Month- the month of the recording
4. Year- the year of the recording
5. Evap- values of that day
6. MaxTemp- Maximum temperature recorded on that day
7. MaxTemp- Minimum temperature recorded on that day
8. Wind Velocity-in kmph
9. Solar Radiation-received for that day
10. Bright Sunshine-the amount of sunlight received in the day (measured in hrs)

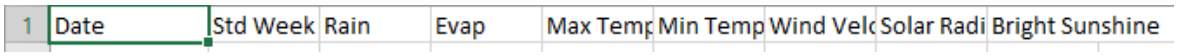


Fig 4.3 Weather dataset

PM2.5 DATASET-

In this Dataset we simply have the daily average PM2.5.

4.2.2 Importing Packages that are Necessary

Pandas: Pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Pandas is a NumFOCUS sponsored project.

NumPy: NumPy is the fundamental package for scientific computing with Python. It contains among other things:

a powerful N-dimensional array object, sophisticated (broadcasting) functions, tools for integrating C/C++ and Fortran code, useful linear algebra, Fourier transform, and random number capabilities.

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

Matplotlib: Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupiter notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible.

You can generate plots, histograms, power spectra, bar charts, error charts, scatterplots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery. For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Scikit-learn: Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python.

Seaborn: Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Stats models: stats models is a Python module that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and

statistical data exploration. An extensive list of result statistics are available for each estimator. The results are tested against existing statistical packages to ensure that they are correct.

4.2.3 Pre-processing and Data Cleaning

The very First thing to do is to load our dataset into the workspace i.e. in the Notebook.

Loading The data: To Load the data we use the function the pd. read_csv('path of the file'), since our data is of the csv format.

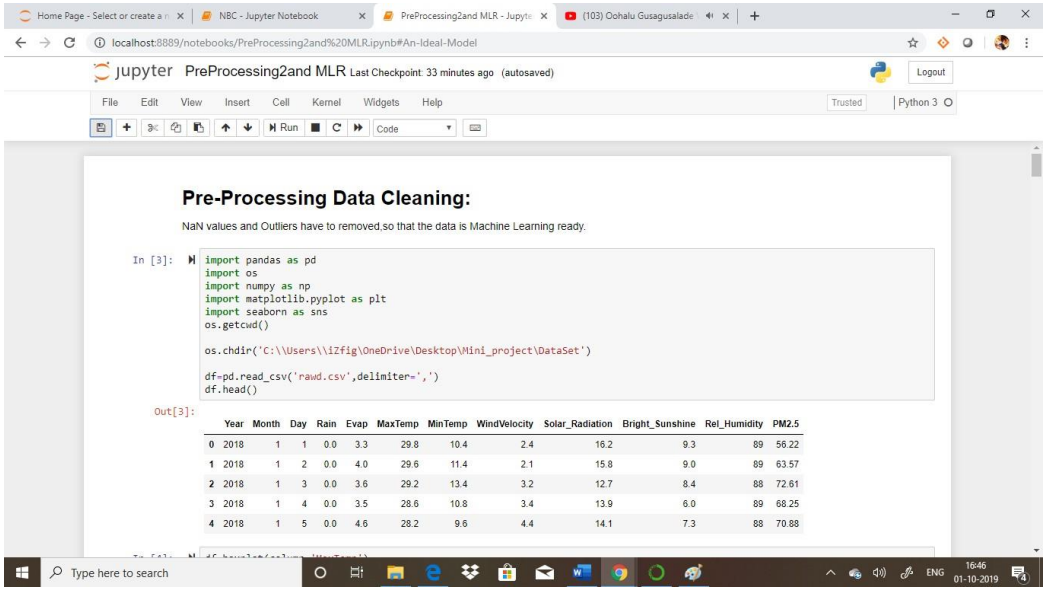


Fig 4.4 Importing Necessary Packages and the Dataset

Checking for Null values:

We have to remove the Outliers present in MinTemp and MaxTemp.-(**INTER-Quartile Method**)
Also Impute the NaN Values in PM2.5 Column.(**fillna()**).

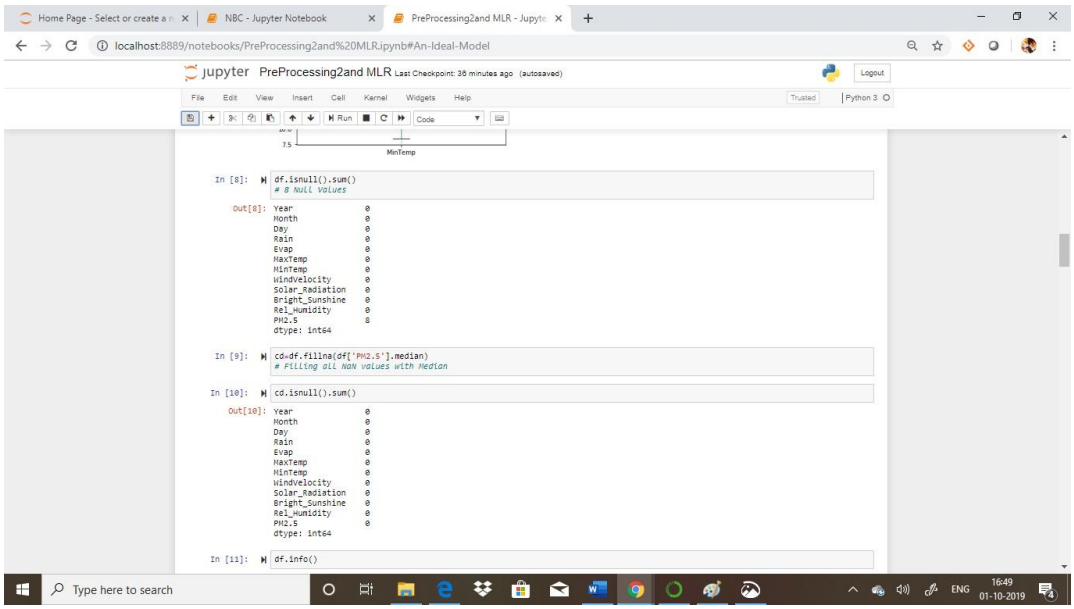


Fig 4.5 Null Values imputed!

Data Imputation: is done to replace any missing values or 'Nan' values with mean, median or mode or some constant which would not affect the meaning of our data.

• An Imputer function can be used to replace the values.

• fillna() is also used to fill 'NaN' values.

• replace() can be used replace the categorical values with numerical values for analysing the data more accurately.

4.2.4 Correlation Analysis

Correlation is a statistical technique that can show whether and how strongly pairs of variables are related. For example, height and weight are related; taller people tend to be heavier than shorter people. The relationship isn't perfect.

People of the same height vary in weight, and you can easily think of two people you know where the shorter one is heavier than the taller one. Nonetheless, the average weight of people 5'5" is less than the average weight of people 5'6", and their average weight is less than that of people 5'7", etc. Correlation can tell you just how much of the variation in peoples' weights is related to their heights.

Although this correlation is fairly obvious your data may contain unsuspected correlations. You may also suspect there are correlations, but don't know which are the strongest. An intelligent correlation analysis can lead to a greater understanding of our data.

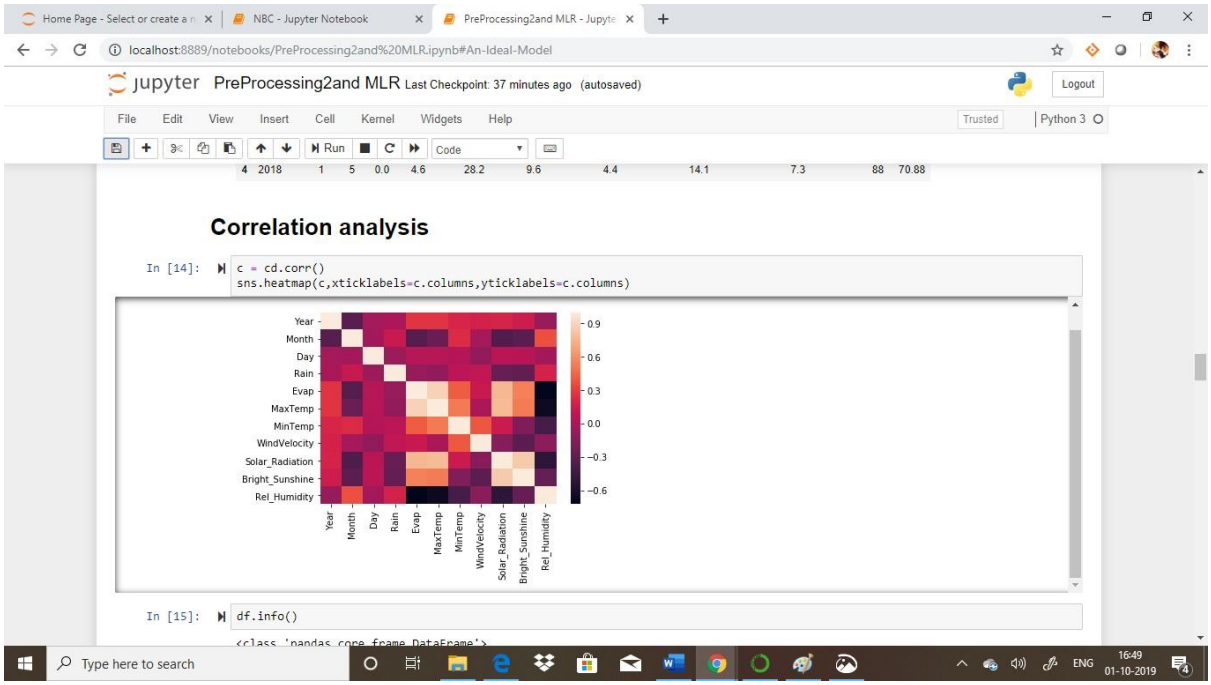


Fig 4.6 Heatmap is visual representation of Correlation

In our data from correlation analysis we found that the,

Independent Variables: MinTemp,WindVelocity, Month,Year,Rain,BrightSunshine

Dependent Variable: PM2.5

4.2.5 Reducing Dimensionality

In machine learning classification problems, there are often too many factors on the basis of which the final classification is done. These factors are basically variables called features. The higher the number of features, the harder it gets to visualize the training set and then work on it. Sometimes, most of these features are correlated, and hence redundant. This is where dimensionality reduction algorithms come into play. Dimensionality reduction is the process of reducing the number of random variables under consideration, by obtaining a set of principal variables. It can be divided into feature selection and feature extraction.

Reasons to Reduce Dimensionality

An intuitive example of dimensionality reduction can be discussed through a simple e-mail classification problem, where we need to classify whether the e-mail is spam or not. This can involve a large number of features, such as whether or not the e-mail has a generic title, the content of the e-mail, whether the e-mail uses a template, etc. However, some of these features may overlap.

In another condition, a classification problem that relies on both humidity and rainfall can be collapsed into just one underlying feature, since both of the aforementioned are correlated to a high degree. Hence, we can reduce the number of features in such problems.

A 3-D classification problem can be hard to visualize, whereas a 2-D one can be mapped to a simple 2-dimensional space, and a 1-D problem to a simple line. Now we can drop all the features except *MinTemp, WindVelocity, Month, Year, Rain, BrightSunshine.*

4.2.6 Train-Test-Split

The data we use is usually split into training data and test data. The training set contains a known output and the model learns on this data in order to be generalized to other data later on. We have the test dataset (or subset) in order to test our model's prediction on this subset.

4.3 Algorithm

4.3.1 Multiple Linear Regression

Multiple linear regression (MLR), also known simply as multiple regression, is a statistical technique that uses several explanatory variables to predict the outcome of a response variable. The goal of multiple linear regression (MLR) is to model the linear relationship between the explanatory (independent) variables and response (dependent) variable.

In essence, multiple regression is the extension of ordinary least-squares (OLS) regression that involves more than one explanatory variable.

The Formula for Multiple Linear Regression Is:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + \epsilon \quad \text{....equation (6)}$$

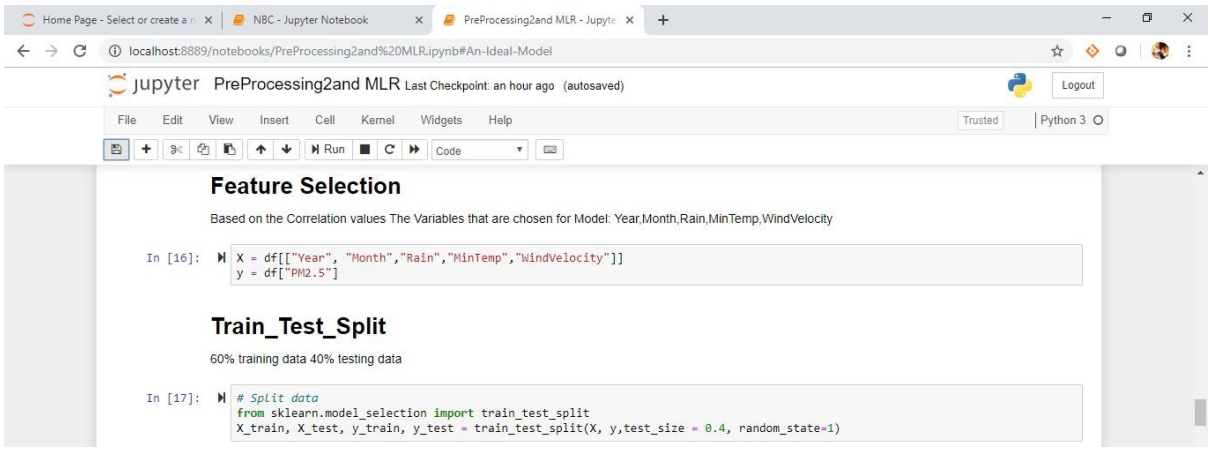


Fig 4.7 Feature selection to reduce dimensionality

Multiple linear regression (MLR) is used to determine a mathematical relationship among a number of random variables. In other terms, MLR examines how multiple independent variables are related to one dependent variable.

Once each of the independent factors has been determined to predict the dependent variable, the information on the multiple variables can be used to create an accurate prediction on the level of effect they have on the outcome variable. The model creates a relationship in the form of a straight line (linear) that best approximates all the individual data points.

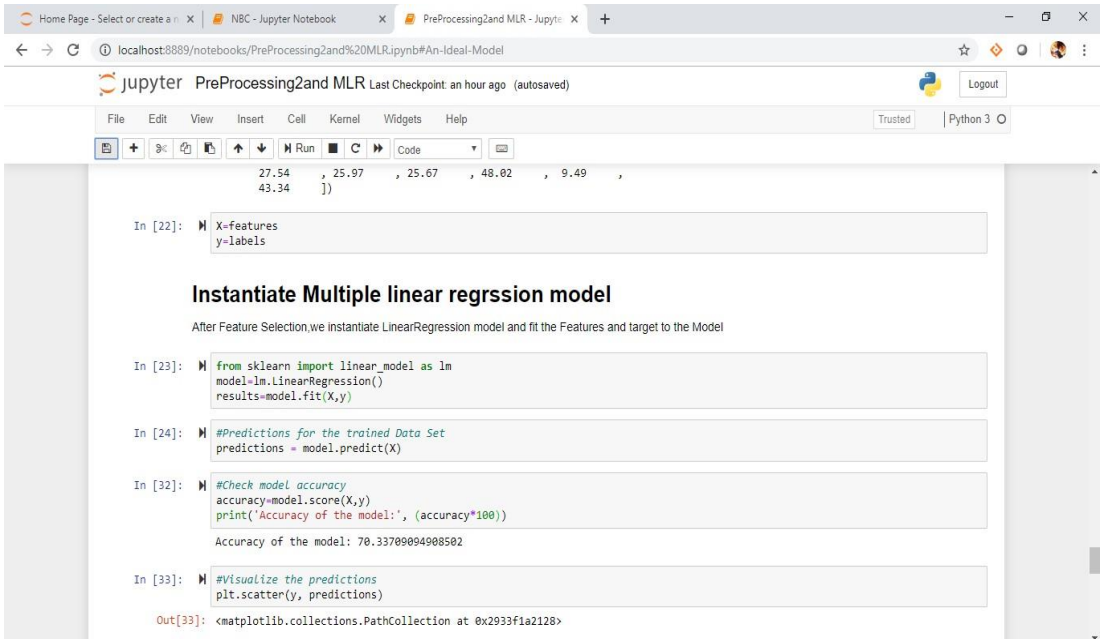


Fig 4.8 Model initialization and its accuracy

4.3.2 Logistic Regression

Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

Sometimes logistic regressions are difficult to interpret; the Intellectus Statistics tool easily allows you to conduct the analysis, then in plain English interprets the output.

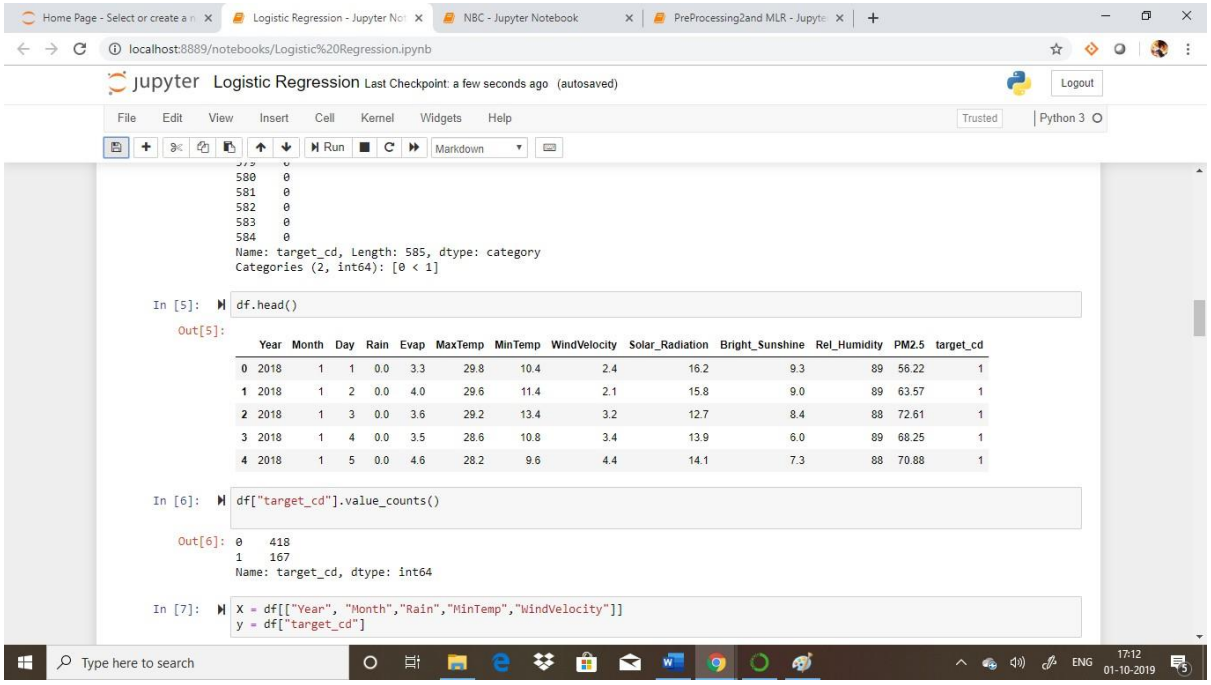


Fig 4.9 Transformed target variable

Logistic regression uses an equation as the representation, very much like linear regression.

Input values (x) are combined linearly using weights or coefficient values (referred to as the Greek capital letter Beta) to predict an output value (y). A key difference from linear regression is that the output value being modeled is a binary values (0 or 1) rather than a numeric value.

Below is an example logistic regression equation:

$$y = e^{(b0 + b1*x)} / (1 + e^{(b0 + b1*x)}) \ldots \text{equation(7)}$$

Where y is the predicted output, b0 is the bias or intercept term and b1 is the coefficient for the single input value (x). Each column in your input data has an associated b coefficient (a constant real value) that must be learned from your training data.

The actual representation of the model that you would store in memory or in a file are the coefficients in the equation (the beta value or b's).

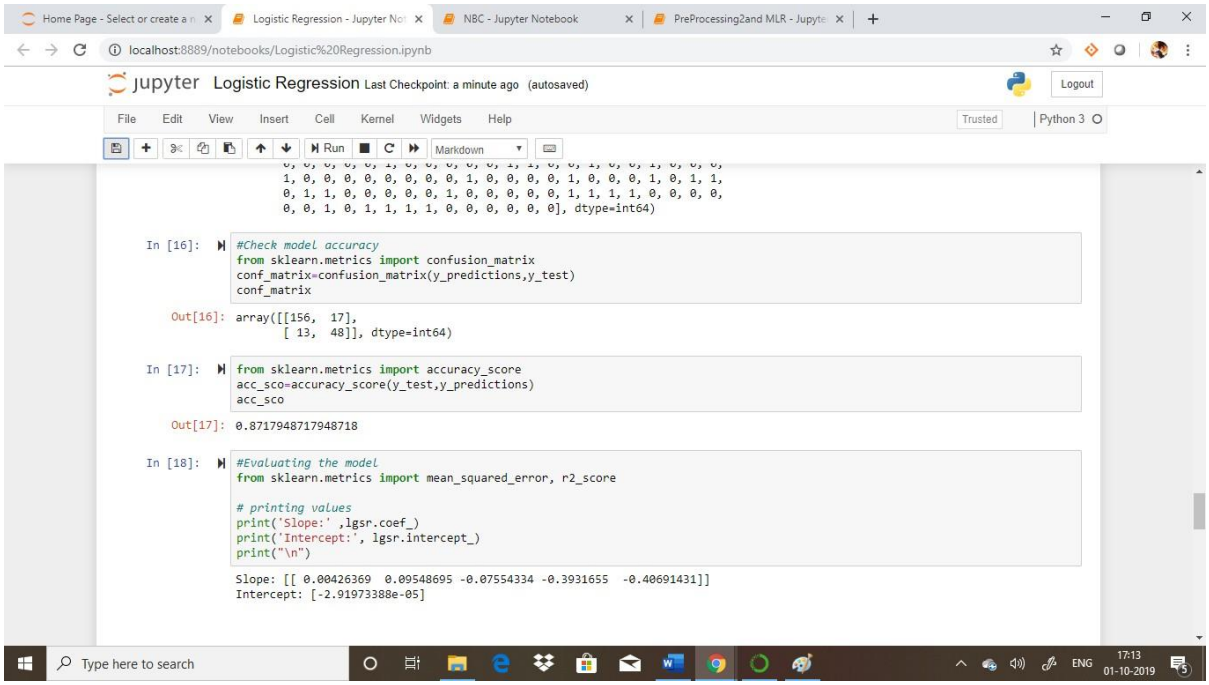


Fig 4.10 Initializing the model

After the predictions the model shows an accuracy of 87.18%.

4.3.3 Naive Bayes classifier

Naïve Bayes classifier focuses on the categorical type of data, our first objective was to convert all of our numerical data to categorical type. We used the labelling method to categorize out data to classes.

The assumptions made by Naive Bayes are not generally correct in real-world situations. In-fact, the independence assumption is never correct but often works well in practice.

In Gaussian Naive Bayes, continuous values associated with each feature are assumed to be distributed according to a Gaussian distribution. A Gaussian distribution is also called Normal distribution.

Assumptions:

In spite of their apparently over-simplified assumptions, naive Bayes classifiers have worked quite well in many real-world situations, famously document classification and spam filtering. They require a small amount of training data to estimate the necessary parameters.

Naive Bayes learners and classifiers can be extremely fast compared to more sophisticated methods. The decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one dimensional distribution. This in turn helps to alleviate problems stemming from the curse of dimensionality.

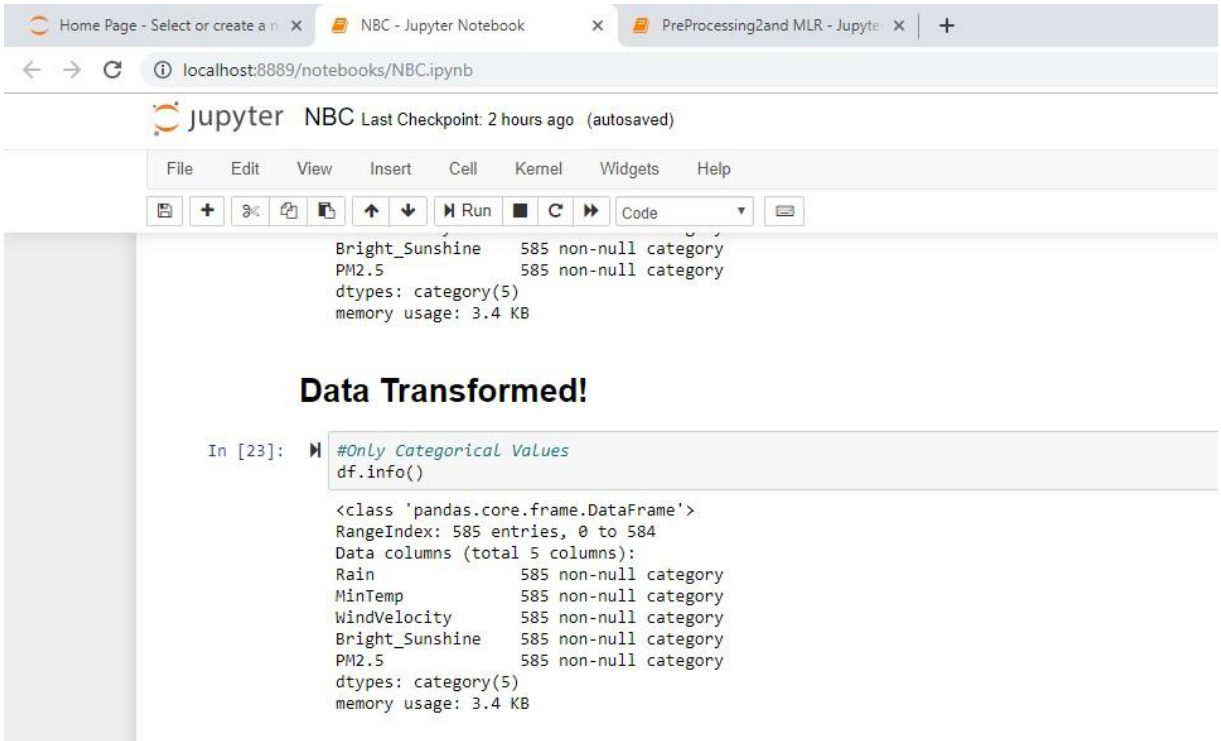


Fig 4.11 Transformed data

Now we have our Transformed Dataset which is machine learning ready. The next thing to do is to initialize a model for our dataset. After the model is built we need to verify the accuracy with the predictions. To do this we are going with cross-validation metrics.

Cross-validation is used to generalize the results of a statistical analysis of an independent dataset.

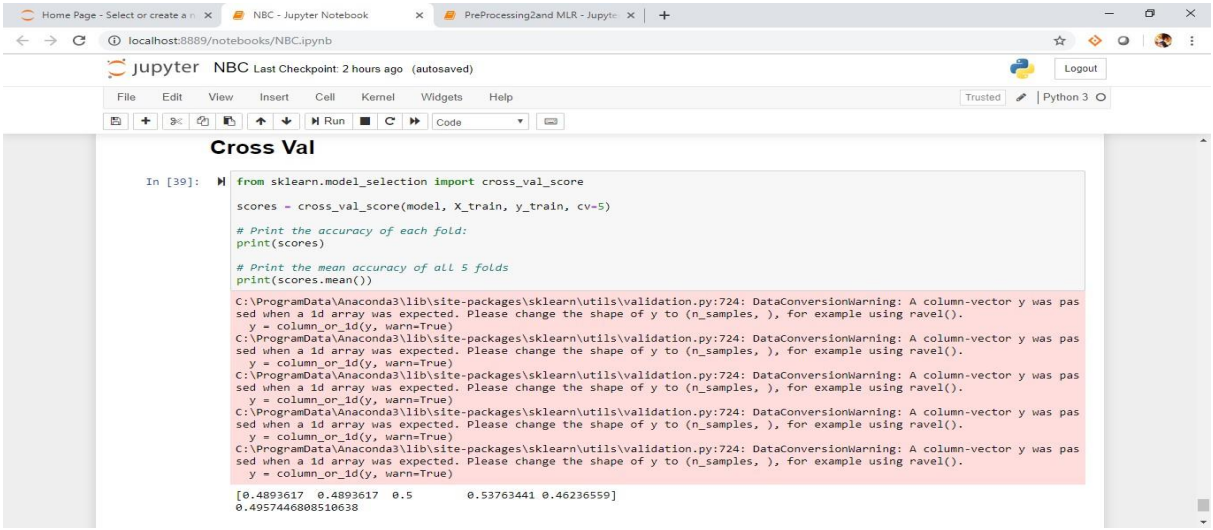


Fig 4.12 Cross Validation scores

After performing cross validations the model shows less accuracy with an average accuracy of 50 therefore, we decided not to deploy the Naive Bayes classifier.

4.4 Model evaluation

RMSE- Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line datapoints are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit. Root mean square error is commonly used in climatology, forecasting, and regression analysis to verify experimental results.

R2 Score- In statistics, the coefficient of determination, denoted R2 or r2 and pronounced "R squared", is the proportion of the variance in the dependent variable that is predictable from the independent variable(s).

4.5 User-Interface Design

4.5.1 Electron.js

Electron is an open source library developed by GitHub for building cross-platform desktop applications with HTML, CSS, and JavaScript. Electron accomplishes this by combining Chromium and Node.js into a single runtime and apps can be packaged for Mac, Windows, and Linux.

Electron began in 2013 as the framework on which Atom, GitHub's hackable text editor, would be built. The two were open sourced in the Spring of 2014.It has since become a popular tool used by open source developers, start-ups, and established companies.

In order to keep Electron small (file size) and sustainable (the spread of dependencies and APIs) the project limits the scope of the core project. For instance, Electron uses Chromium's rendering library rather than all of Chromium. This makes it easier to upgrade Chromium but also means some browser features found in Google Chrome do not exist in Electron.

4.5.2 Dependencies

Electron's version of Chromium is usually updated within one or two weeks after a new stable Chromium version is released, depending on the effort involved in the upgrade. When a new version of Node.js is released, Electron usually waits about a month before upgrading in order to bring in a more stable version. In Electron, Node.js and Chromium share a single V8 instance—usually the version that Chromium is using. Most of the time this just works but sometimes it means patching Node.js.

IMPLEMENTATION

5.1 Tools Used

5.1.1 Anaconda-Jupyter Notebooks

Directly from the platform and without involving DevOps, data scientists can develop and deploy AI and machine learning models rapidly into production. Anaconda provides the tools needed to easily:

- Collect data from files, databases, and data lakes.
- Manage environments with Conda (all package dependencies are taken care of at the time of download)
- Share, collaborate on, and reproduce projects
- Deploy projects into production with the single click of a button

Data scientists that work in silos struggle to add value for their organization. That's why Anaconda created an integrated, end-to-end data experience.

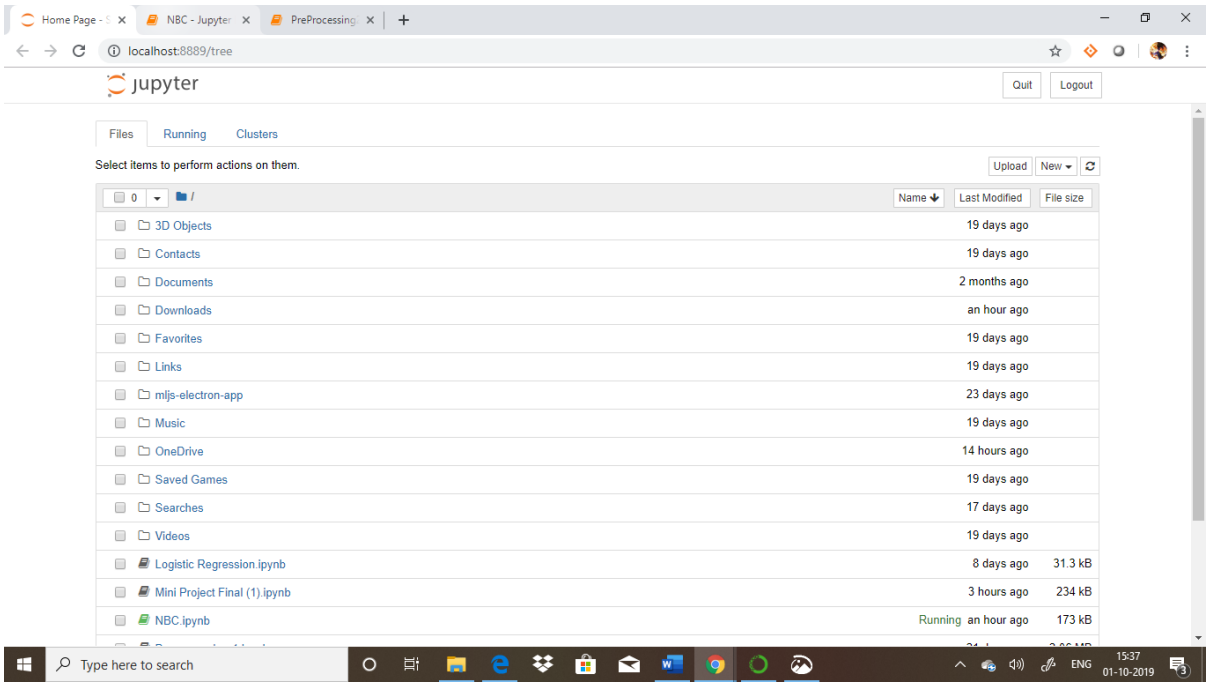


Fig 5.1 Anaconda Jupyter notebook Homepage

The open-source Anaconda Distribution is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 15 million users worldwide, it is the industry standard for developing, testing, and training on a single machine, enabling individual data scientists to:

Quickly download 1,500+ Python/R data science packages, manage libraries, dependencies, and environments with Conda, Develop and train machine learning and deep learning models with Scikit-learn, TensorFlow, and Theano

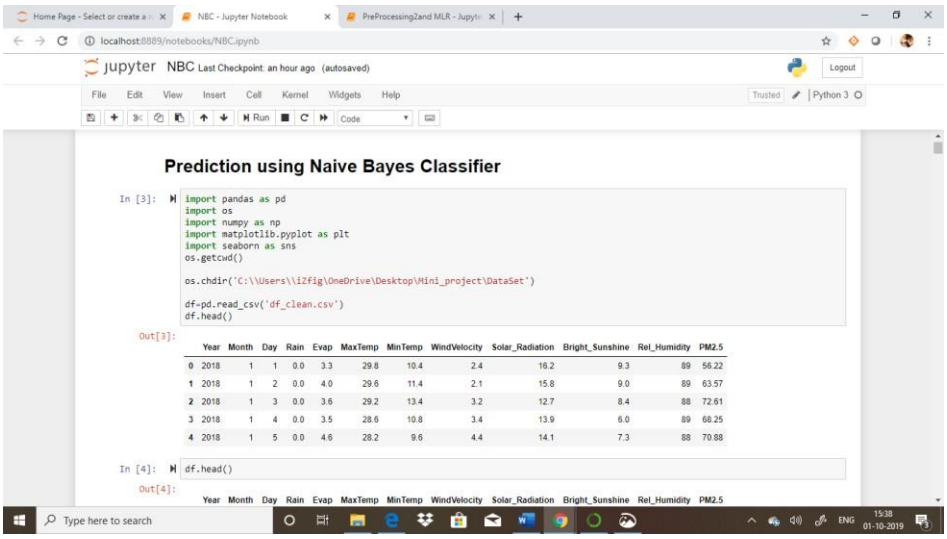


Fig 5.2 A project in Jupyter Notebook

5.1.2 Visual Studio Code

Getting up and running with Visual Studio Code is quick and easy. It is a small download so you can install in a matter of minutes and give VS Code a try.

Cross platform: VS Code is a free code editor which runs on the macOS, Linux and Windows operating systems. VS Code is lightweight and should run on most available hardware and platform versions.

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for

JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).

Visual Studio Code is an editor first and foremost, and includes the features you need for highly productive source code editing. This topic takes you through the basics of the editor and helps you get moving with your code.

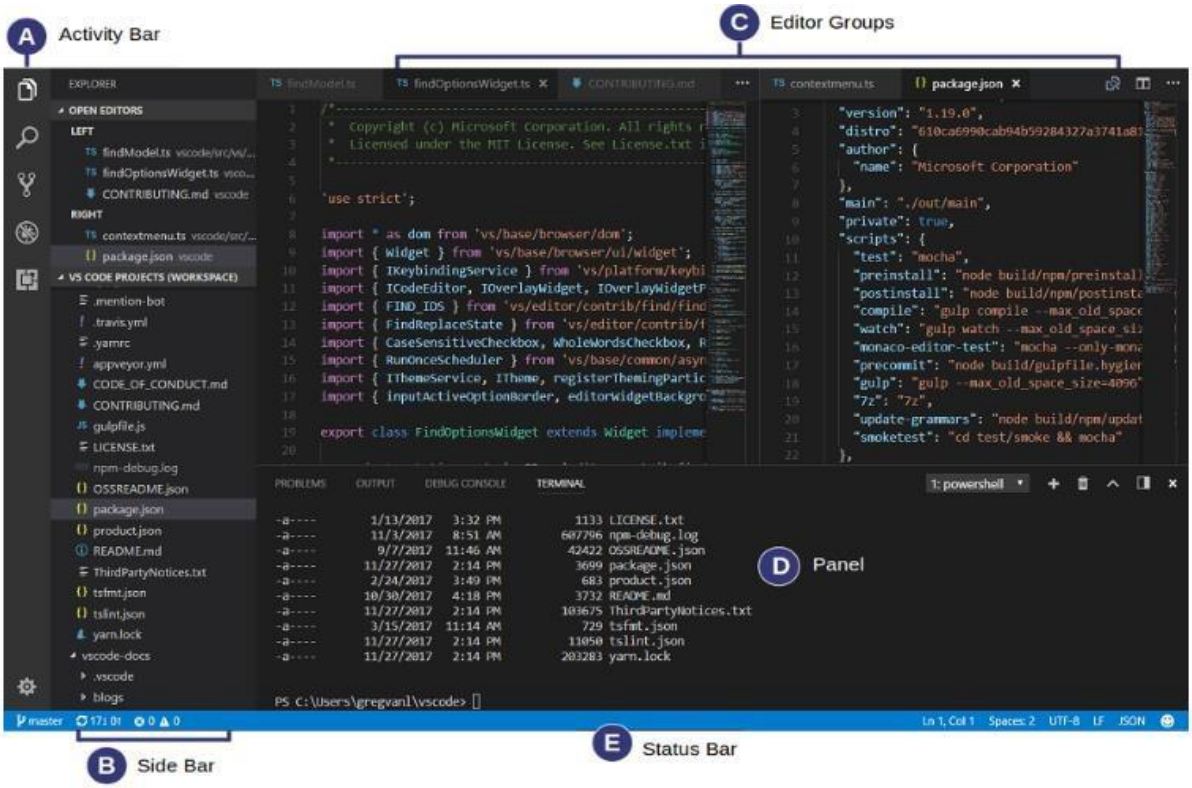


Fig 5.3 Workspace

5.1.3 Electron.js and Node.js

Node.js: First, install a recent version of Node.js. We recommend that you install either the latest LTS or Current version available. Visit the Node.js download page and select the Windows Installer. Once downloaded, execute the installer and let the installation wizard guide you through the installation.

On the screen that allows you to configure the installation, make sure to select the Node.js runtime, npm package manager, and Add to PATH options.

Once installed, confirm that everything works as expected. Find the Windows PowerShell by opening the Start Menu and typing PowerShell. Open up PowerShell or another command line client of your choice and confirm that both node and npm are available:

```
# This command should print the version of Node.js
node -v

# This command should print the version of npm
npm -v
```

Electron.js: At this point, you'll need to install electron itself. The recommended way of doing so is to install it as a development dependency in your app, which allows you to work on multiple apps with different Electron versions. To do so, run the following command from your app's directory:

```
npm install --save-dev electron
```

5.2 User-Interface concept

The Application contains 2 models that have high accuracy namely, Logistic Regression and Multiple Linear Regression. Each of them taken in Inputs from the form and the data is sent and handled by the backend Java-Script. The JavaScript then uses model coefficients and intercept along with the input variables to predict a value.The Predicted Value is then shown in a Dialogue box.

DEPLOYING THE PROJECT

6.1 Electron packager

Electron Packager is a command line tool and Node.js library that bundles Electron-based application source code with a renamed Electron executable and supporting files into folders ready for distribution.

For creating distributable like installers and Linux packages, consider using either Electron Forge (which uses Electron Packager internally), or one of the related Electron tools, which utilizes Electron Packager-created folders as a basis.

For use in npm scripts (recommended)

npm install electron-packager --save-dev

This will create a service that will pack the Developed App

```
npm run package-win
```

Now a new folder will be created “builds” which will contain our Application

6.2 User Interface

Electron.js and Node.js provides very powerful tools and packages to create GUI for projects at the same time we can create our applications with HTML and CSS.

Similar to the development of our Websites and our Web Services, the learning curve is pretty straightforward with these tools. We Deployed our Application with 4 tabs for the user to interact with. If there were no interface the user has to understand the concepts of Machine Learning and python to be able to get a prediction.

Also the Presentation of the Project is a factor.

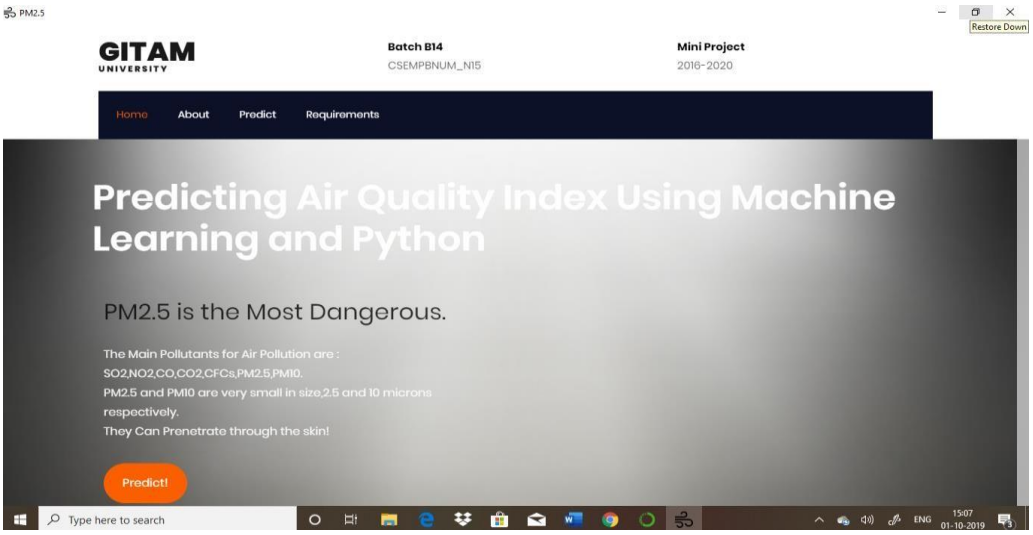


Fig 6.1 Home Tab

Home: this tab contains information about the AQI and Pollutants, And also the predict button.

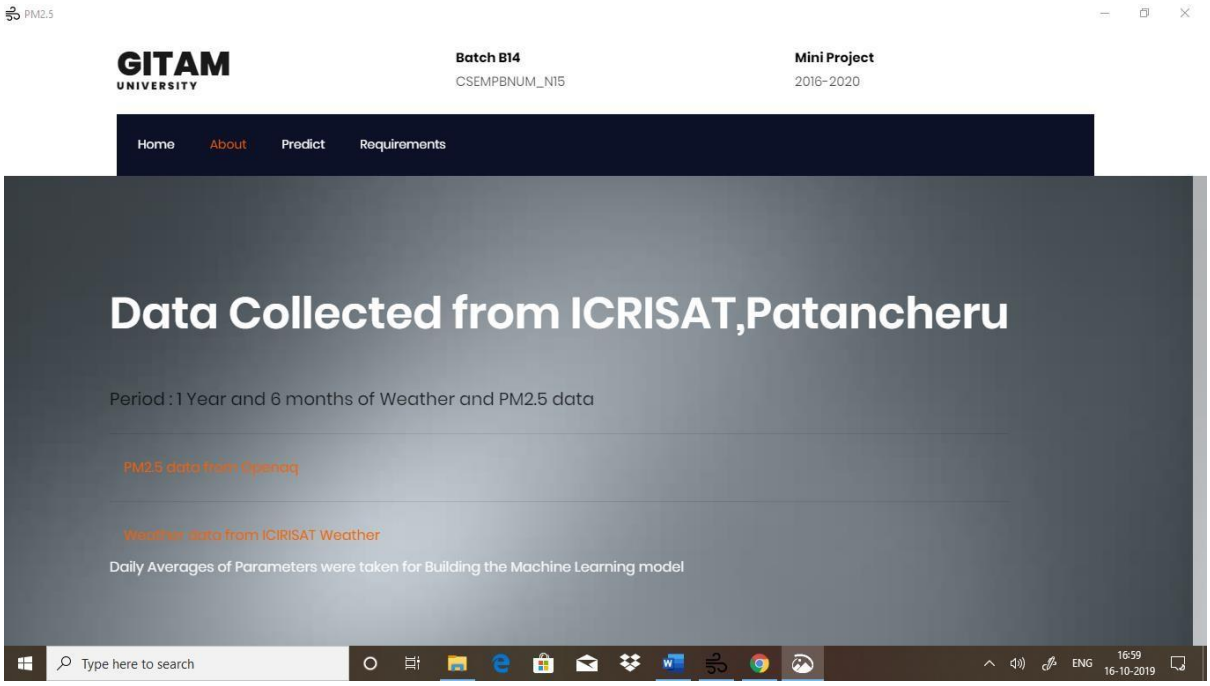


Fig 6.2 About Tab

About: this tab contains information about where we collected out DATA from.

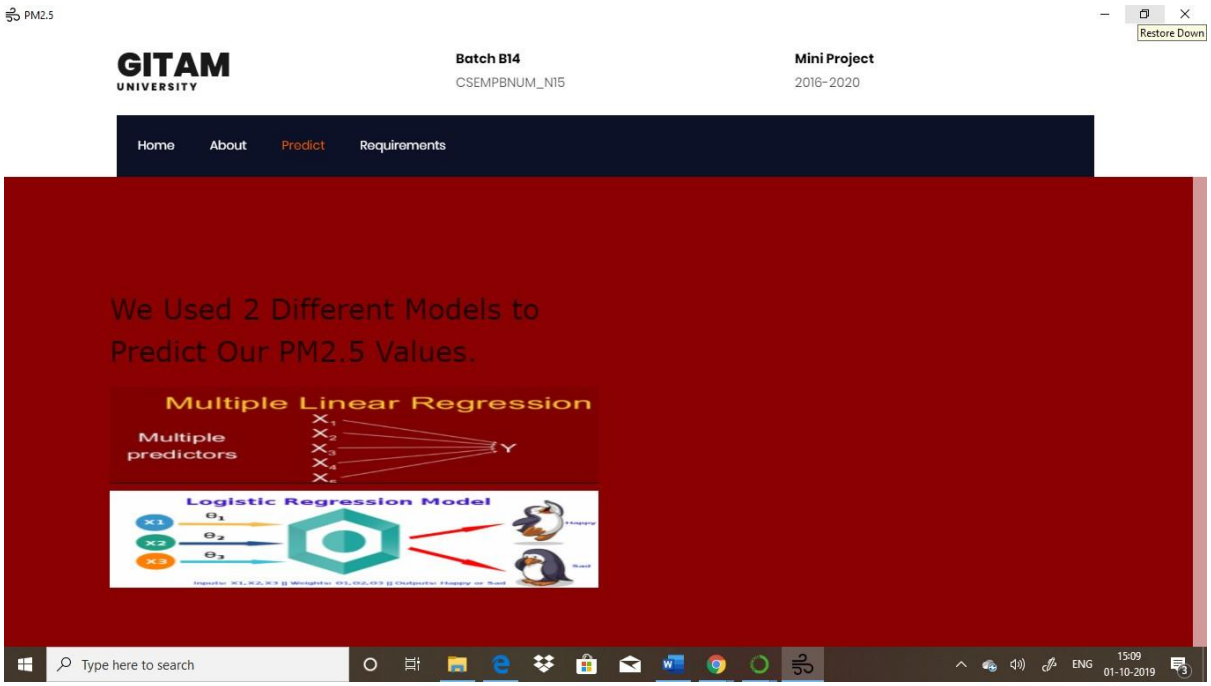


Fig 6.3 Predict Tab

Predict: this tab contains Models that we deployed. We deployed MLR model and Logistic Regression model. We choose not to deploy Naïve Bayes Classifier as it resulted in less accuracy and is not useful for predictions.

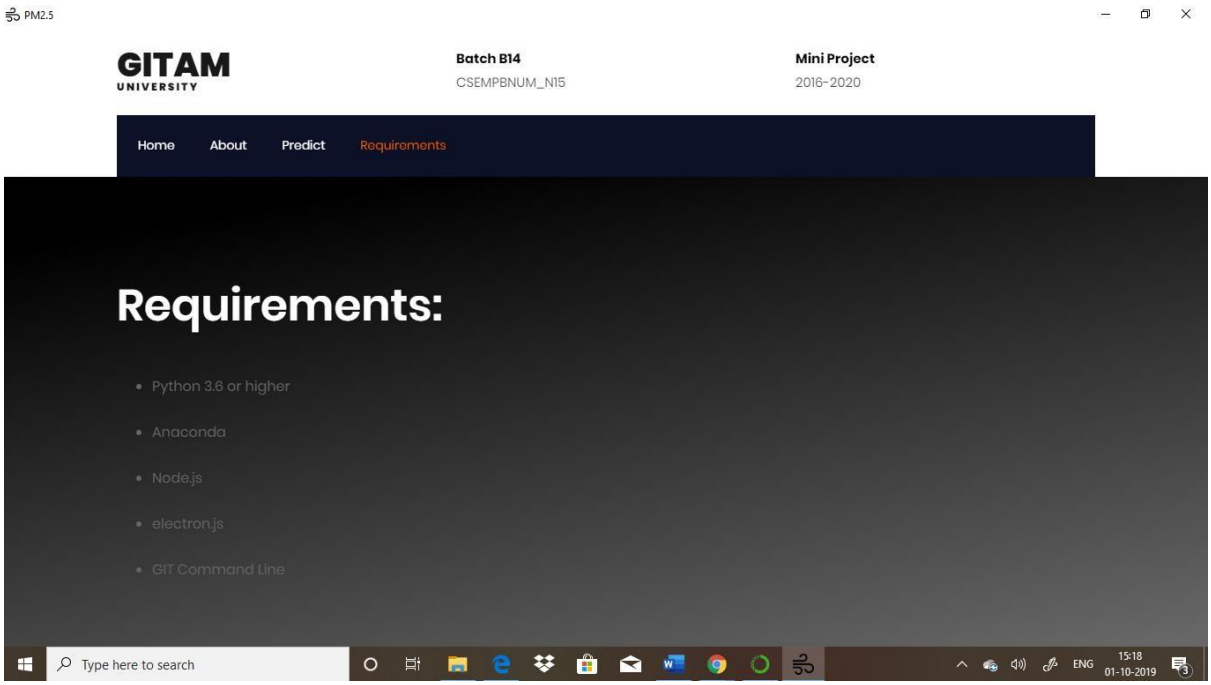


Fig 6.4. Requirements Tab

Requirements: this tab contains information about the necessary tools we used to build the application.

PREDICTIONS AND INSIGHTS

7.1 By Manually Giving Parameters

We gave a random input for the date 1/10/2019 with 2mm rain,24 minTemp,4kmph wind speed.

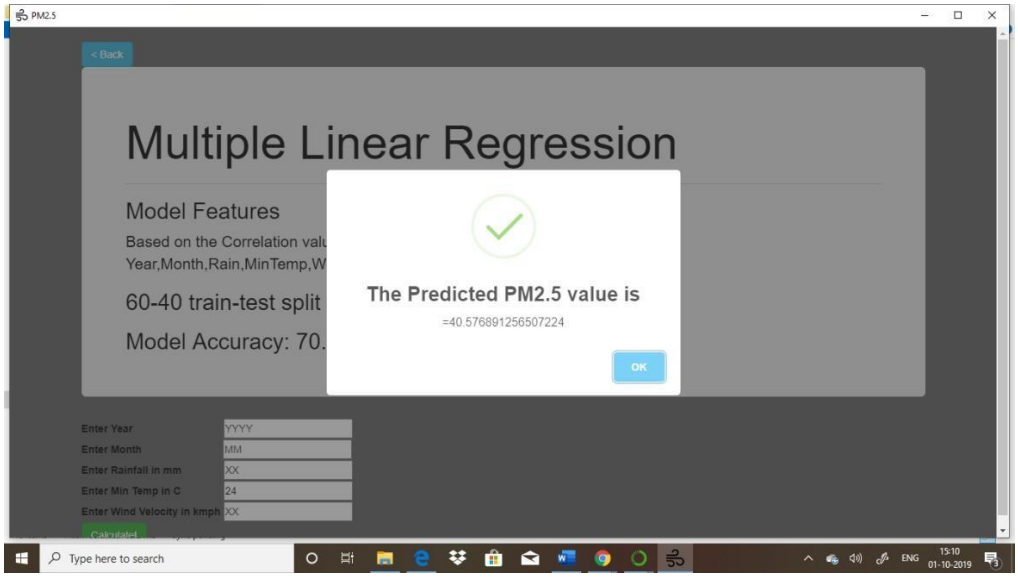


Fig 7.1 Output results for the above parameters using MLR

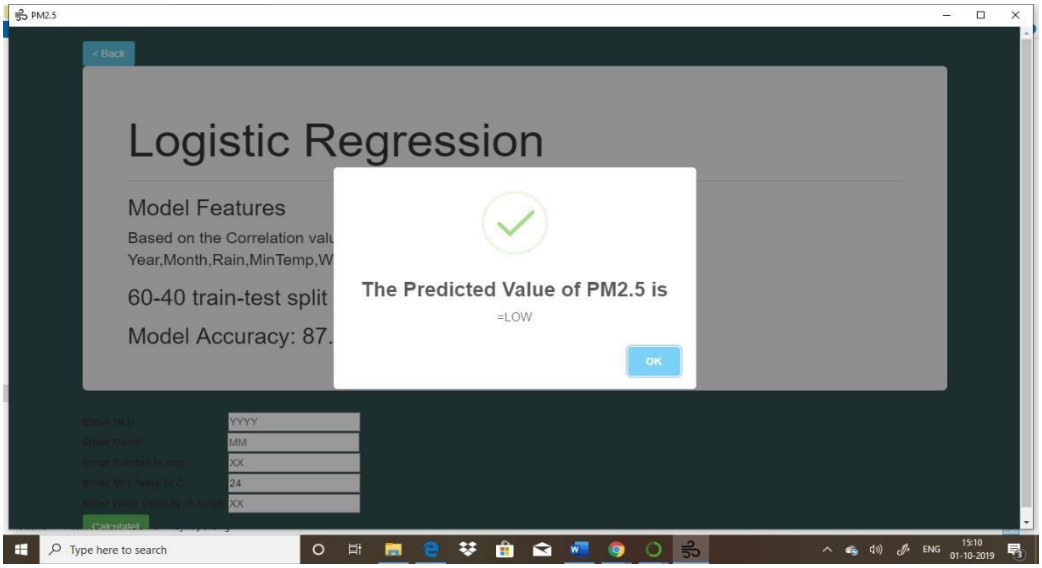


Fig 7.2 Output results for the above parameters using LR

Both the models we deployed were showing results in the range -30% to +30% of the accuracy.

7.3 Conclusion

Only 2 out of 3 Models built on the collected data showed reasonable accuracy. We choose not to deploy the Naïve Bayes Classifier as it showed very less accuracy metrics and is not capable of predicting further.

Our Model Performance for Multiple Linear Regression Model is 70.33% and for Logistic Regression is 87.18%.

FUTURE WORK AND SCOPE

8.1 Air Quality Index

OUR primary goal for PM25 is to give those living in ICRISAT information that will allow them to make good decisions for their families. It’s great to know what the air pollution level is right now, but how many people check levels multiple times per day?

Just like the weather forecast tells you whether to bring a coat or umbrella, it makes sense for a pollution forecast to tell you whether to take your mask or alter your plans.

Our Model is still not efficient.

8.2 Real Time Predictions

Real-time predictive analytics is the process of extracting useful information from data sets in real time. This is done to determine and predict future outcomes. Real-time predictive analytics does not precisely predict what will happen in the future; instead, it forecasts what might happen on the basis of certain "if" scenarios.

Real-time predictive analytics is based on a predictive model which is deployed to run in order to enable decision-making processes in real time. A predictive model is built on the basis of large amounts of data. The predictive model could be built in two ways: by a data scientist or through a streaming operational data analytics platform.

The whole process involves rigorous experimentation, historical data and other similar processes. This whole process is iterative by nature. This model is then made to predict run time when a continuous stream of data is fed to it. Therefore, real-time predictive analytics enhances the customer experience and in turn helps businesses to increase earnings.

REFERENCES

https://electronjs.org/

https://nodejs.org/en/

https://stackoverflow.com/

https://www.kaggle.com/

https://app.cpcbccr.com/AQI_India/