*Predicting The Energy Output Of Wind Turbine Based On   Weather Condition*

**Project Description:**

Wind energy plays an increasing role in the supply of energy world-wide. The energy output of a wind farm is highly dependent on the wind conditions present at its site. If the output can be predicted more accurately, energy suppliers can coordinate the collaborative production of different energy sources more efficiently to avoid costly overproduction.

              Better prediction models for the upcoming supply of renewable energy are important to decrease the need of controlling energy provided by conventional power plants.

                        Hereby, we formulate the prediction task as regression problem and test different regression techniques such as linear regression, Random Forest and Long short term memory. In our experiments, we analyze predictions for individual turbines and show that a machine learning approach yields feasible results for short-term wind power prediction.

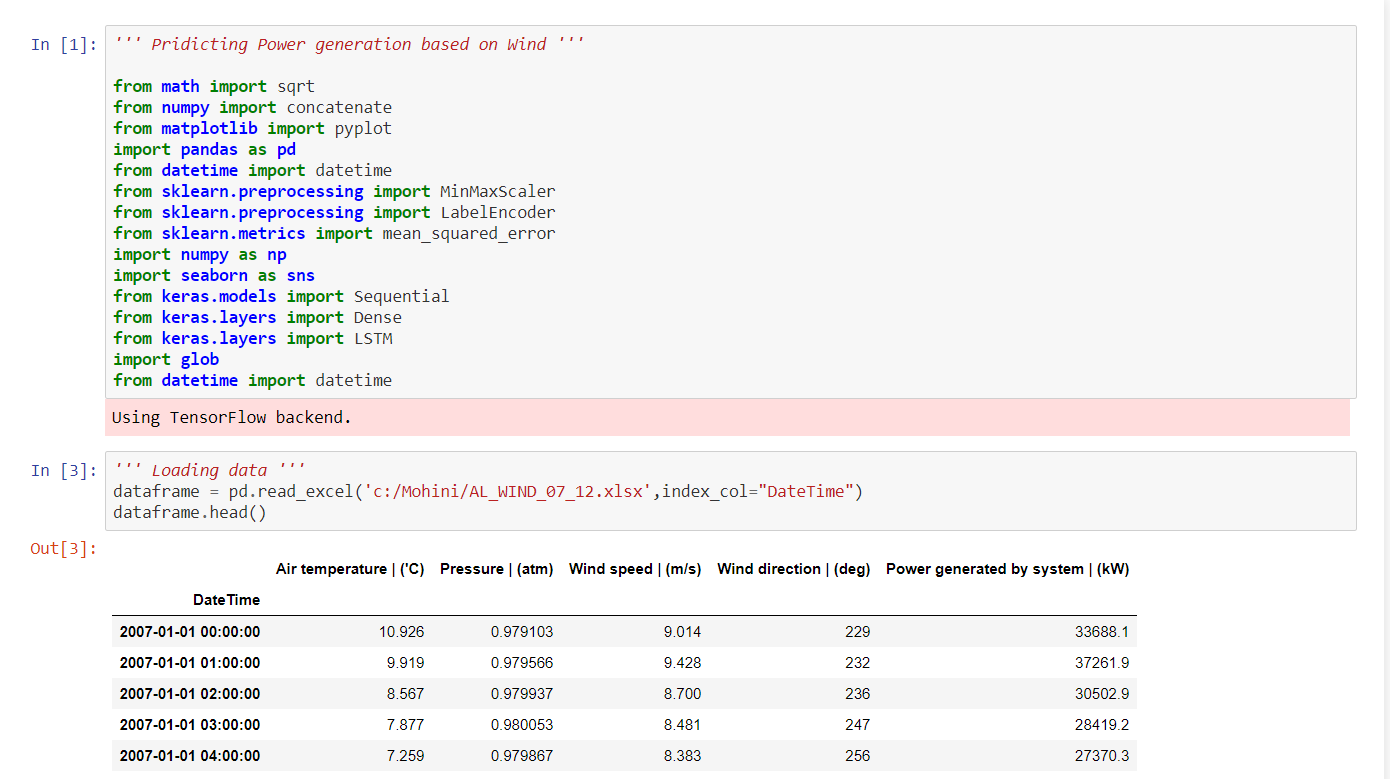
**Data Collection:**

Data collection is the process of gathering and measuring information on variables of interest, in an established systematic fashion that enables one to answer stated research questions, test hypotheses, and evaluate outcomes. The data collection component of research is common to all fields of study including physical and social sciences, humanities, business, etc. While methods vary by discipline, the emphasis on ensuring accurate and honest collection remains the same.

***Why is Data Collection Important?***

Collecting data allows you to capture a record of past events so that we can use data analysis to find recurring patterns. From those patterns, you build predictive [models](https://www.datarobot.com/wiki/model) using machine learning [algorithms](https://www.datarobot.com/wiki/algorithm) that look for trends and predict future changes.

Predictive models are only as good as the data from which they are built, so good data collection practices are crucial to developing high-performing models. The data need to be error-free (garbage in, garbage out) and contain relevant information for the task at hand. For example, a loan default model would not benefit from tiger population sizes but could benefit from gas prices over time.



**Data Pre-Processing:**

Data pre-processing in Machine Learning is a crucial step that helps enhance the quality of data to promote the extraction of meaningful insights from the data. Data pre-processing in Machine Learning refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for a building and training Machine Learning models. In simple words, data pre-processing in Machine Learning is a data mining technique that transforms raw data into an understandable and readable format.

          Data preprocessing in Machine Learning is a crucial step that helps enhance the quality of data to promote the extraction of meaningful insights from the data. Data preprocessing in Machine Learning refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for a building and training Machine Learning models. In simple words, data preprocessing in Machine Learning is data mining techniquethat transforms raw data into an understandable and readable format.

                Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.

***Need of Data Preprocessing:***

• For achieving better results from the applied model in Machine Learning projects the format of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format, for example, Random Forest algorithm does not support null values, therefore to execute random forest algorithm null values have to be managed from the original raw data set.  
• Another aspect is that data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in one data set, and best out of them is chosen.

**1. Rescale Data:**

• When our data is comprised of attributes with varying scales, many machine learning algorithms can benefit from rescaling the attributes to all have the same scale.  
• It is also useful for algorithms that weight inputs like regression and neural networks and algorithms that use distance measures like K-Nearest Neighbors.  
• We can rescale your data using scikit-learn using the MinMaxScaler class.

**2. Binarize Data (Make Binary):**

• We can transform our data using a binary threshold. All values above the threshold are marked 1 and all equal to or below are marked as 0.  
• This is called binarizing your data or threshold your data. It can be useful when you have probabilities that you want to make crisp values. It is also useful when feature engineering and you want to add new features that indicate something meaningful.  
• We can create new binary attributes in Python using scikit-learn with the [Binarizer](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.Binarizer.html) class.

**3. Standardize Data:**

• Standardization is a useful technique to transform attributes with a Gaussian distribution and differing means and standard deviations to a standard Gaussian distribution with a mean of 0 and a standard deviation of 1.  
• We can standardize data using scikit-learn with the [StandardScaler](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html) class.

➤ Acquire the data:

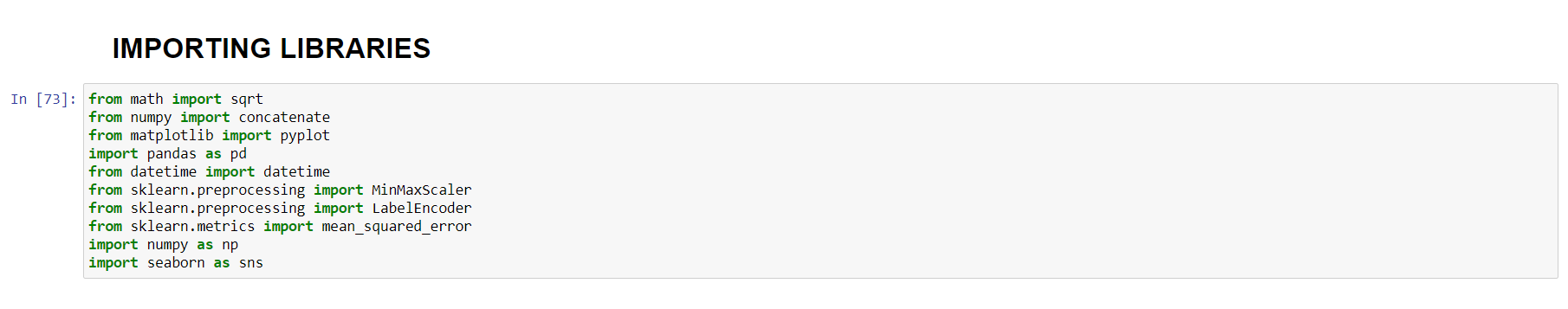
To build and develop Machine Learning models, you must first acquire the relevant dataset. This dataset will be comprised of data gathered from multiple and disparate sources which are then combined in a proper format to form a dataset. Dataset formats differ according to use cases.

There are several online sources from where you can download datasets like kaggle etc.

➤ **Import important libraries:**

Since Python is the most extensively used and also the most preferred library by Data Scientists around the world, we’ll here use you popular Python libraries for data pre-processing in Machine Learning. The predefined Python libraries can perform specific data pre-processing jobs. There are three core Python libraries used for this data pre-processing in Machine Learning. They are:

* **NumPy** – NumPy is the fundamental package for scientific calculation in Python. Hence, it is used for inserting any type of mathematical operation in the code. Using NumPy, you can also add large multidimensional arrays and matrices in your code.
* **Pandas** – Pandas is an excellent open-source Python library for data manipulation and analysis. It is extensively used for importing and managing the datasets. It packs in high-performance, easy-to-use data structures and data analysis tools for Python .
* **Matplotlib** – Matplotlib is a Python 2D plotting library that is used to plot any type of charts in Python. It can deliver publication-quality figures in numerous hard copy formats and interactive environments across platforms .
* **Scikit-learn:** Scikit-learn (formerly scikits.learn and also known as sklearn) is a [free software](https://en.wikipedia.org/wiki/Free_software) [machine learning](https://en.wikipedia.org/wiki/Machine_learning) [library](https://en.wikipedia.org/wiki/Library_(computing)) for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) [programming language](https://en.wikipedia.org/wiki/Programming_language). It features various [classification](https://en.wikipedia.org/wiki/Statistical_classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis) and [clustering](https://en.wikipedia.org/wiki/Cluster_analysis) algorithms including [support vector machines](https://en.wikipedia.org/wiki/Support_vector_machine), [random forests](https://en.wikipedia.org/wiki/Random_forests), [gradient boosting](https://en.wikipedia.org/wiki/Gradient_boosting), [*k*-means](https://en.wikipedia.org/wiki/K-means_clustering) and [DBSCAN](https://en.wikipedia.org/wiki/DBSCAN), and is designed to interoperate with the Python numerical and scientific libraries [NumPy](https://en.wikipedia.org/wiki/NumPy) and [SciPy](https://en.wikipedia.org/wiki/SciPy).



➤ **Import dataset:**

In this step, we need to import the dataset/s that we have gathered for the ML project at hand. However, before we can import the dataset/s, we must set the current directory as the working directory.

Here, we use Jupyter Notebook as our platform.

Once you’ve set the working directory containing the relevant dataset, we import the dataset using the “read\_csv()” function of the Pandas library. This function can read a CSV file (either locally or through a URL) and also perform various operations on it. The read\_csv() is written as:

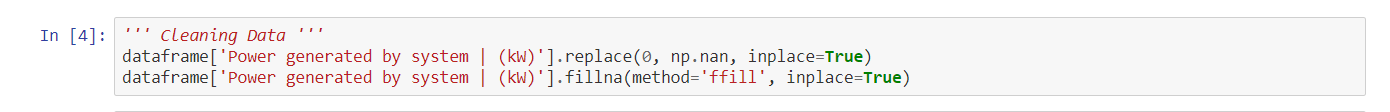
data\_set= pd.read\_csv(‘dataset.csv’)

In this line of code, “data\_set” denotes the name of the variable wherein you stored the dataset. The function contains the name of the dataset as well. Once you execute this code, the dataset will be successfully imported.

➤ **Identifying and handling missing values:**

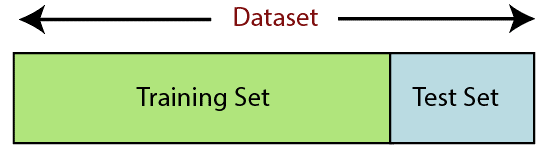
In data preprocessing, it is pivotal to identify and correctly handle the missing values, failing to do this, you might draw inaccurate and faulty conclusions and inferences from the data. Needless to say, this will hamper your ML project.Basically, there are two ways to handle missing data:

* **Deleting a particular row** – In this method, you remove a specific row that has a null value for a feature or a particular column where more than 75% of the values are missing. However, this method is not 100% efficient, and it is recommended that you use it only when the dataset has adequate samples. You must ensure that after deleting the data, there remains no addition of bias.
* **Calculating the mean** – This method is useful for features having numeric data like age, salary, year, etc. Here, you can calculate the mean, median, or mode of a particular feature or column or row that contains a missing value and replace the result for the missing value. This method can add variance to the dataset, and any loss of data can be efficiently negated. Hence, it yields better results compared to the first method (omission of rows/columns). Another way of approximation is through the deviation of neighbouring values. However, this works best for linear data.



➤ **Splitting the dataset:**

Every dataset for Machine Learning model must be split into two separate sets – training set and test set.



Training set denotes the subset of a dataset that is used for training the machine learning model. Here, you are already aware of the output. A test set, on the other hand, is the subset of the dataset that is used for testing the machine learning model. The ML model uses the test set to predict outcomes.

Usually, the dataset is split into 70:30 ratio or 80:20 ratio. This means that you either take 70% or 80% of the data for training the model while leaving out the rest 30% or 20%. The splitting process varies according to the shape and size of the dataset in question.Here, we use 60:40 ratio means we take 60% of data for training the model and rest 40% to test it.

**What is Overfitting/Underfitting a Model?**

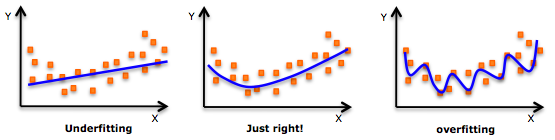
As mentioned, in statistics and machine learning we usually split our data into two subsets: training data and testing data (and sometimes to three: train, validate and test), and fit our model on the train data, in order to make predictions on the test data. When we do that, one of two thing might happen: we overfit our model or we underfit our model. We don’t want any of these things to happen, because they affect the predictability of our model — we might be using a model that has lower accuracy and/or is ungeneralized (meaning you can’t generalize your predictions on other data). Let’s see what under and overfitting actually mean:

***Overfitting***

Overfitting means that model we trained has trained “too well” and is now, well, fit too closely to the training dataset. This usually happens when the model is too complex (i.e. too many features/variables compared to the number of observations). This model will be very accurate on the training data but will probably be very not accurate on untrained or new data. It is because this model is not generalized (or not AS generalized), meaning you can generalize the results and can’t make any inferences on other data, which is, ultimately, what you are trying to do. Basically, when this happens, the model learns or describes the “noise” in the training data instead of the actual relationships between variables in the data. This noise, obviously, isn’t part in of any new dataset, and cannot be applied to it.

***Underfitting***

In contrast to overfitting, when a model is underfitted, it means that the model does not fit the training data and therefore misses the trends in the data. It also means the model cannot be generalized to new data. As you probably guessed (or figured out!), this is usually the result of a very simple model (not enough predictors/independent variables). It could also happen when, for example, we fit a linear model (like [linear regression](https://medium.com/towards-data-science/simple-and-multiple-linear-regression-in-python-c928425168f9)) to data that is not linear. It almost goes without saying that this model will have poor predictive ability (on training data and can’t be generalized to other data).



It is worth noting the underfitting is not as prevalent as overfitting. Nevertheless, we want to avoid both of those problems in data analysis. You might say we are trying to find the middle ground between under and overfitting our model. As you will see, train/test split and cross validation help to avoid overfitting more than underfitting. Let’s dive into both of them!

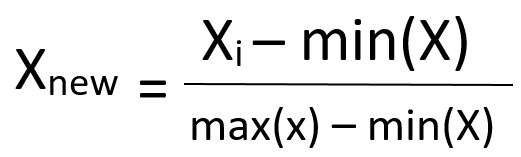
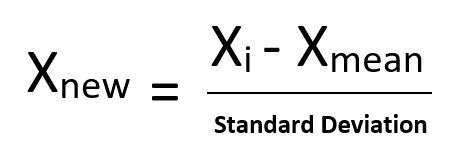
Let’s quickly go over the libraries we’ve imported:

* **Pandas** — to load the data file as a Pandas data frame and analyze the data. If you want to read more on Pandas, feel free to check out [my post](https://medium.com/@adi.bronshtein/a-quick-introduction-to-the-pandas-python-library-f1b678f34673)!
* From **Sklearn**, I’ve imported the *datasets* module, so I can load a sample dataset, and the *linear\_model*, so I can run a linear regression
* From **Sklearn,** sub-library **model\_selection**, I’ve imported the *train\_test\_split* so I can, well, split to training and test sets
* From **Matplotlib** I’ve imported *pyplot* in order to plot graphs of the data

**Feature Selection:**

Feature scaling marks the end of the **data preprocessing in Machine Learning.** It is a method to standardize the independent variables of a dataset within a specific range. In other words, feature scaling limits the range of variables so that you can compare them on common grounds.

In the dataset, you can notice that the age and salary columns do not have the same scale. In such a scenario, if you compute any two values from the age and salary columns, the salary values will dominate the age values and deliver incorrect results. Thus, you must remove this issue by performing feature scaling for Machine Learning.

* **Min-Max Normalization:** This technique re-scales a feature or observation value with distribution value between 0 and 1.  
  
* **Standardization:** It is a very effective technique which re-scales a feature value so that it has distribution with 0 mean value and variance equals to 1.  
  

**Data Visualization:**

Data Visualization is the presentation of data in graphical format. It helps people understand the significance of data by summarizing and presenting a huge amount of data in a simple and easy-to-understand format and helps communicate information clearly and effectively.

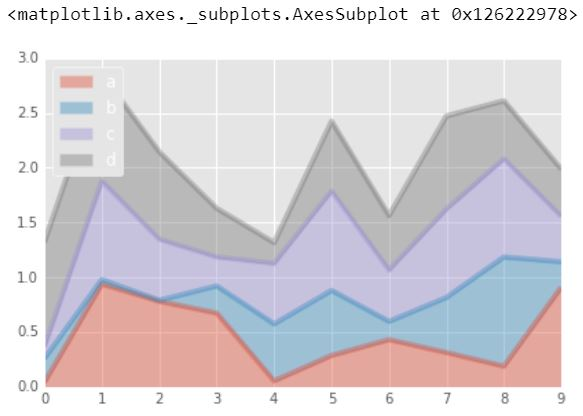
Plot Types –

There are several plot types built-in to pandas, most of them statistical plots by nature:

* df.plot.area
* df.plot.barh
* df.plot.density
* df.plot.hist
* df.plot.line
* df.plot.scatter
* df.plot.bar
* df.plot.box
* df.plot.hexbin
* df.plot.kde
* df.plot.pie

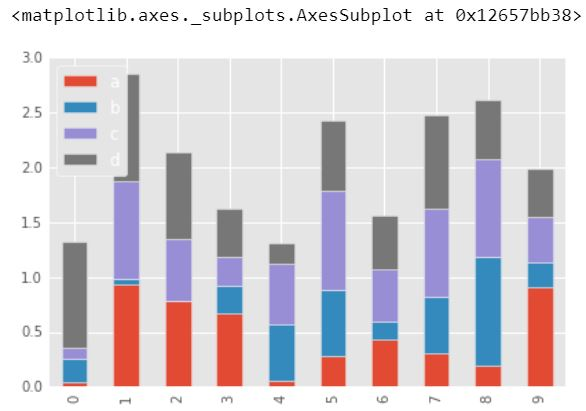
You can also just call df.plot(kind='hist') or replace that kind argument with any of the key terms shown in the list above (e.g. ‘box’, ‘barh’, etc.).

1.) **Area:** An area chart or area graph displays graphically quantitative data. It is based on the line chart. The area between axis and line are commonly emphasized with colors, textures and hatchings. Commonly one compares two or more quantities with an area chart.



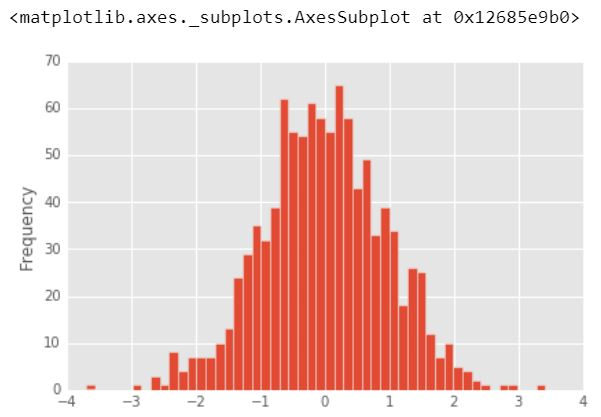
2)Barplot:

A bar chart or bar graph is a chart or graph that presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent. The bars can be plotted vertically or horizontally. A vertical bar chart is sometimes called a line graph.



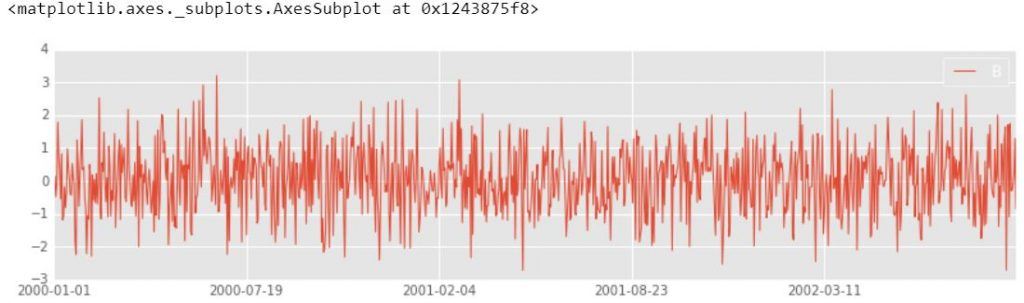
**3)Histogram:**

A histogram is a plot that lets you discover, and show, the underlying frequency distribution (shape) of a set of continuous data. This allows the inspection of the data for its underlying distribution (e.g., normal distribution), outliers, skewness, etc.



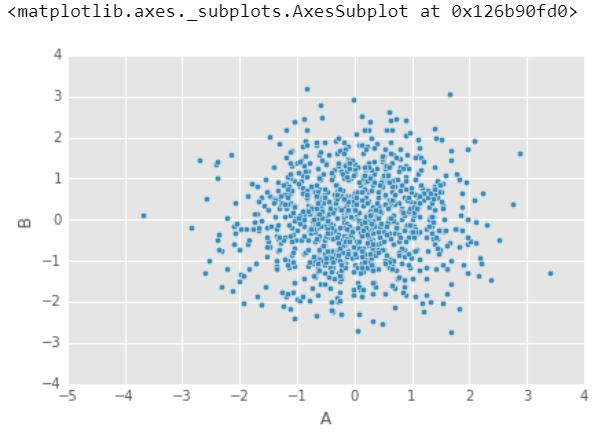
4)Line Plot:

A line plot is a graph that shows frequency of data along a number line. It is best to use a line plot when the data is time series. It is a quick, simple way to organize data.



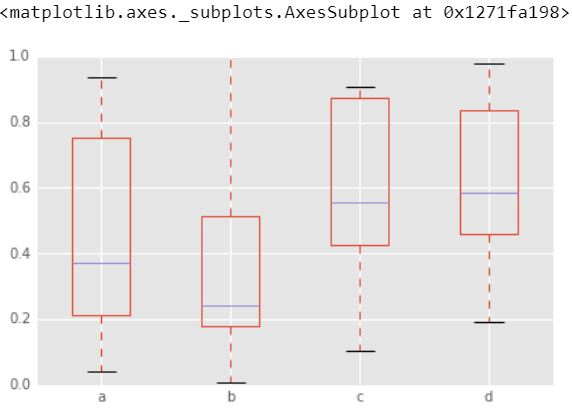
5)Scatter Plot:

Scatter plots are used when you want to show the relationship between two variables. Scatter plots are sometimes called correlation plots because they show how two variables are correlated.



6)Box Plot:

It is a plot in which a rectangle is drawn to represent the second and third quartiles, usually with a vertical line inside to indicate the median value. The lower and upper quartiles are shown as horizontal lines either side of the rectangle.  
A boxplot is a standardized way of displaying the distribution of data based on a five number summary (“minimum”, first quartile (Q1), median, third quartile (Q3), and “maximum”). It can tell you about your outliers and what their values are. It can also tell you if your data is symmetrical, how tightly your data is grouped, and if and how your data is skewed.



**Model building:**

**Linear Regression:**

It is used to estimate real values (cost of houses, number of calls, total sales etc.) based on continuous variable(s). Here, we establish relationship between independent and dependent variables by fitting a best line. This best fit line is known as regression line and represented by a linear equation Y= a \*X + b.

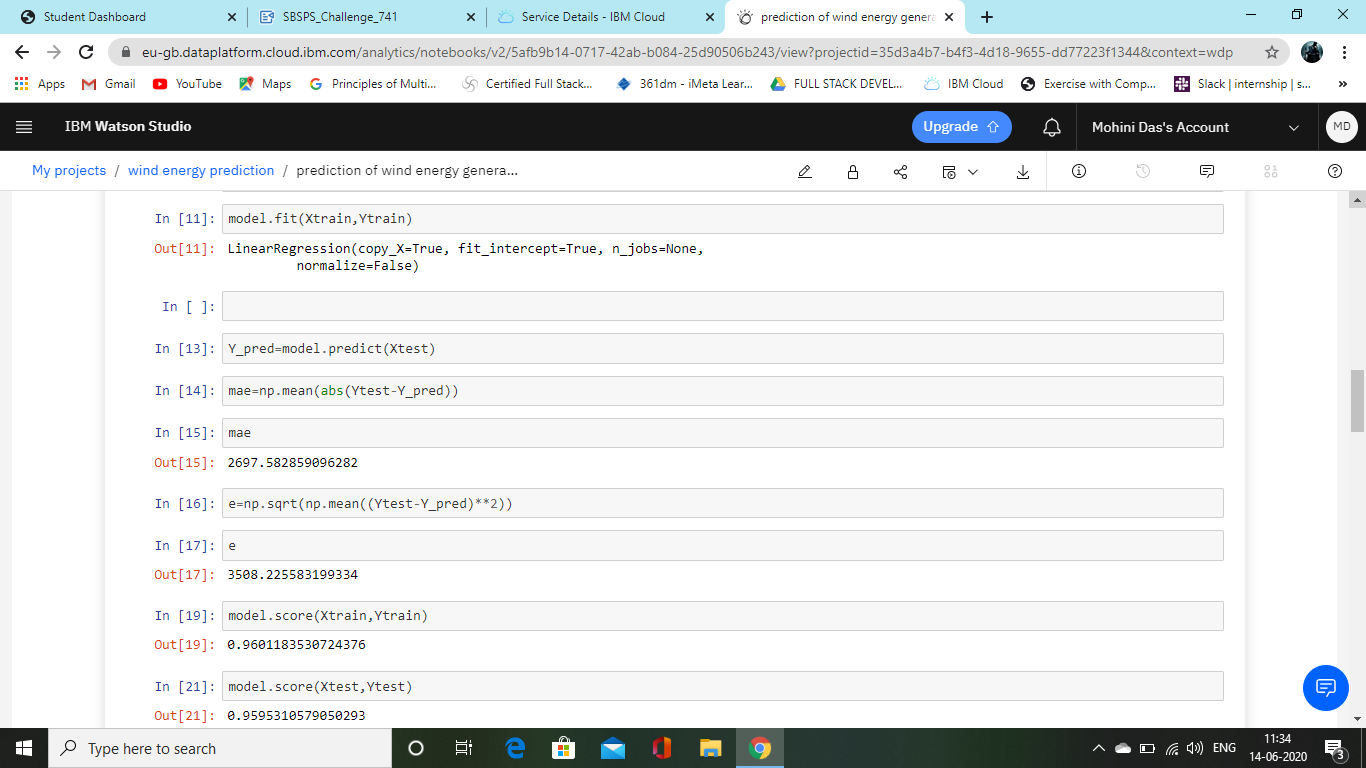
The best way to understand linear regression is to relive this experience of childhood. Let us say, you ask a child in fifth grade to arrange people in his class by increasing order of weight, without asking them their weights! What do you think the child will do? He / she would likely look (visually analyze) at the height and build of people and arrange them using a combination of these visible parameters. This is linear regression in real life! The child has actually figured out that height and build would be correlated to the weight by a relationship, which looks like the equation above.

In this equation:

* Y – Dependent Variable
* a – Slope
* X – Independent variable
* b – Intercept

These coefficients a and b are derived based on minimizing the sum of squared difference of distance between data points and regression line.

After performing Linear regresion our RMSE and accuracy value is given below:



**Logistic Regression:**

Don’t get confused by its name! It is a classification not a regression algorithm. It is used to estimate discrete values ( Binary values like 0/1, yes/no, true/false ) based on given set of independent variable(s). In simple words, it predicts the probability of occurrence of an event by fitting data to a [logit function](https://en.wikipedia.org/wiki/Logistic_function). Hence, it is also known as **logit regression**. Since, it predicts the probability, its output values lies between 0 and 1 (as expected).

Again, let us try and understand this through a simple example.

Let’s say your friend gives you a puzzle to solve. There are only 2 outcome scenarios – either you solve it or you don’t. Now imagine, that you are being given wide range of puzzles / quizzes in an attempt to understand which subjects you are good at. The outcome to this study would be something like this – if you are given a trigonometry based tenth grade problem, you are 70% likely to solve it. On the other hand, if it is grade fifth history question, the probability of getting an answer is only 30%. This is what Logistic Regression provides you.

Our problem is a regression problem so we did not perform this algorithm.

**Decision Tree Model:**

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving **regression and classification problems** too.

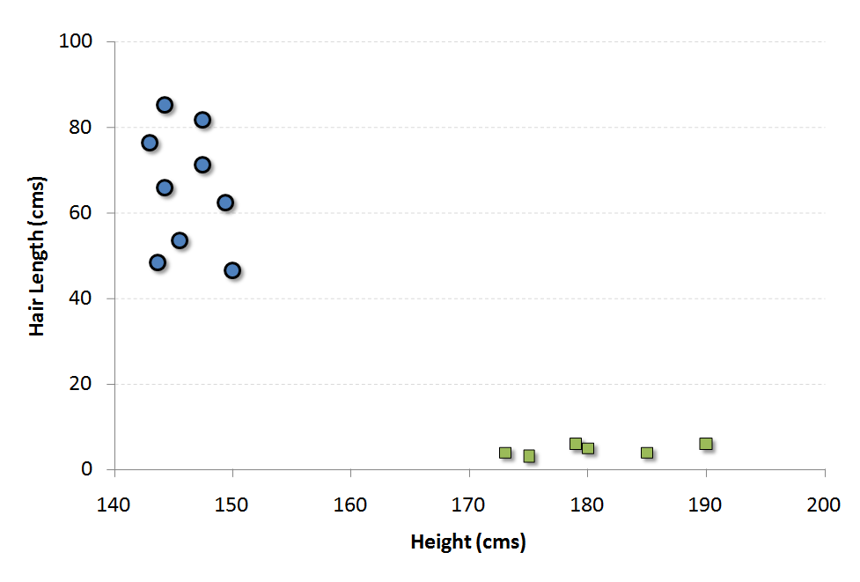
The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by **learning simple decision rules** inferred from prior data(training data).

In Decision Trees, for predicting a class label for a record we start from the **root** of the tree. We compare the values of the root attribute with the record’s attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

**SVM(Support Vector Machine):**

It is a classification method. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate.

For example, if we only had two features like Height and Hair length of an individual, we’d first plot these two variables in two dimensional space where each point has two co-ordinates (these co-ordinates are known as **Support Vectors**)



Now, we will find some *line* that splits the data between the two differently classified groups of data. This will be the line such that the distances from the closest point in each of the two groups will be farthest away.

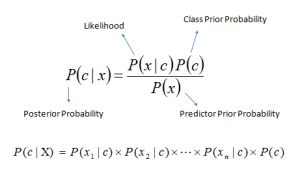
**Naive Bayes:**

It is a classification technique based on Bayes’ Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as ‘Naive’.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/09/Bayes_rule-300x172.png)

**KNN(K Nearest Neighbour):**

K-nearest neighbors (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems. However, it is mainly used for classification predictive problems in industry. The following two properties would define KNN well −

* Lazy learning algorithm − KNN is a lazy learning algorithm because it does not have a specialized training phase and uses all the data for training while classification.
* Non-parametric learning algorithm − KNN is also a non-parametric learning algorithm because it doesn’t assume anything about the underlying data.

Working of KNN Algorithm

K-nearest neighbors (KNN) algorithm uses ‘feature similarity’ to predict the values of new datapoints which further means that the new data point will be assigned a value based on how closely it matches the points in the training set. We can understand its working with the help of following steps −

Step 1 − For implementing any algorithm, we need dataset. So during the first step of KNN, we must load the training as well as test data.

Step 2 − Next, we need to choose the value of K i.e. the nearest data points. K can be any integer.

Step 3 − For each point in the test data do the following −

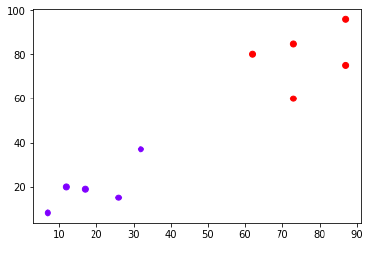
* 3.1 − Calculate the distance between test data and each row of training data with the help of any of the method namely: Euclidean, Manhattan or Hamming distance. The most commonly used method to calculate distance is Euclidean.
* 3.2 − Now, based on the distance value, sort them in ascending order.
* 3.3 − Next, it will choose the top K rows from the sorted array.
* 3.4 − Now, it will assign a class to the test point based on most frequent class of these rows.

Step 4 − End

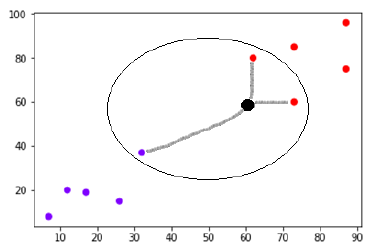
Example

The following is an example to understand the concept of K and working of KNN algorithm −

Suppose we have a dataset which can be plotted as follows −



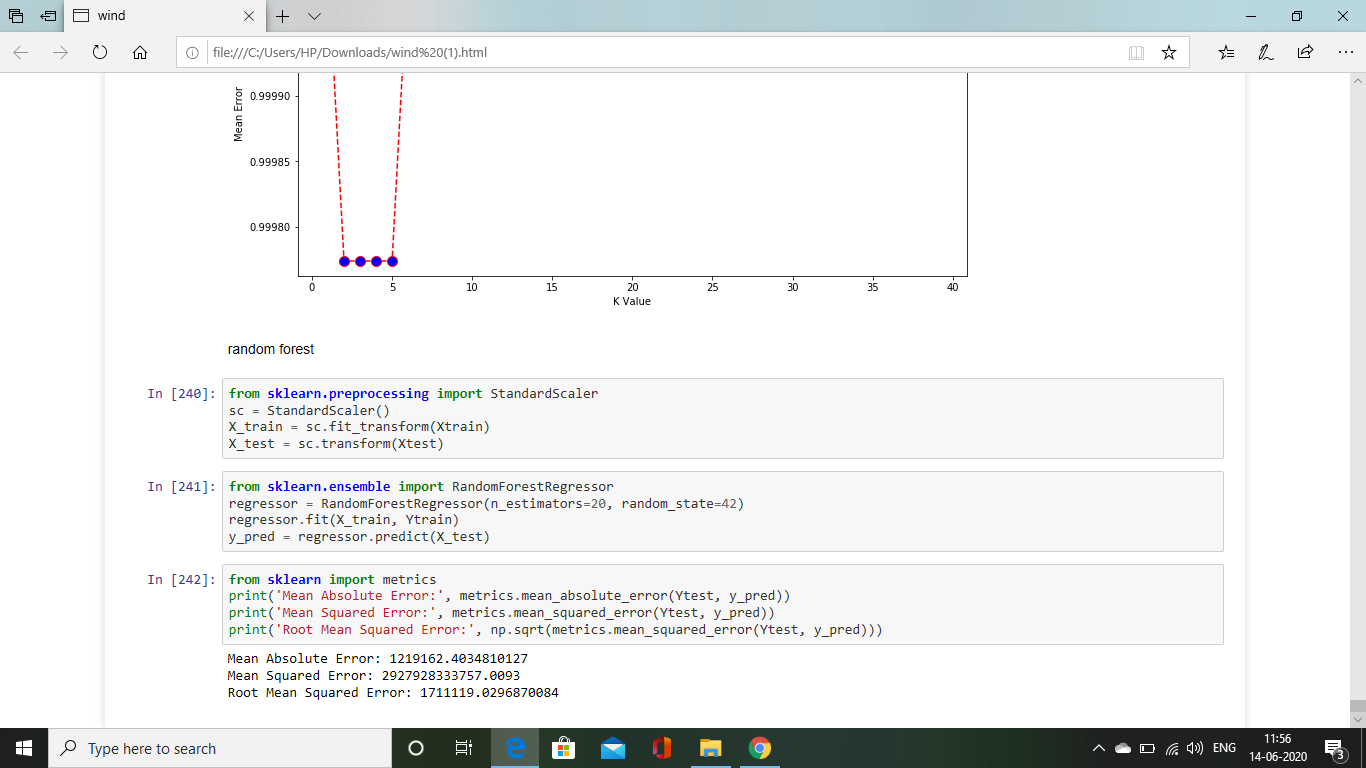
Now, we need to classify new data point with black dot (at point 60,60) into blue or red class. We are assuming K = 3 i.e. it would find three nearest data points. It is shown in the next diagram −



We can see in the above diagram the three nearest neighbors of the data point with black dot. Among those three, two of them lies in Red class hence the black dot will also be assigned in red class.

**Random Forest:**

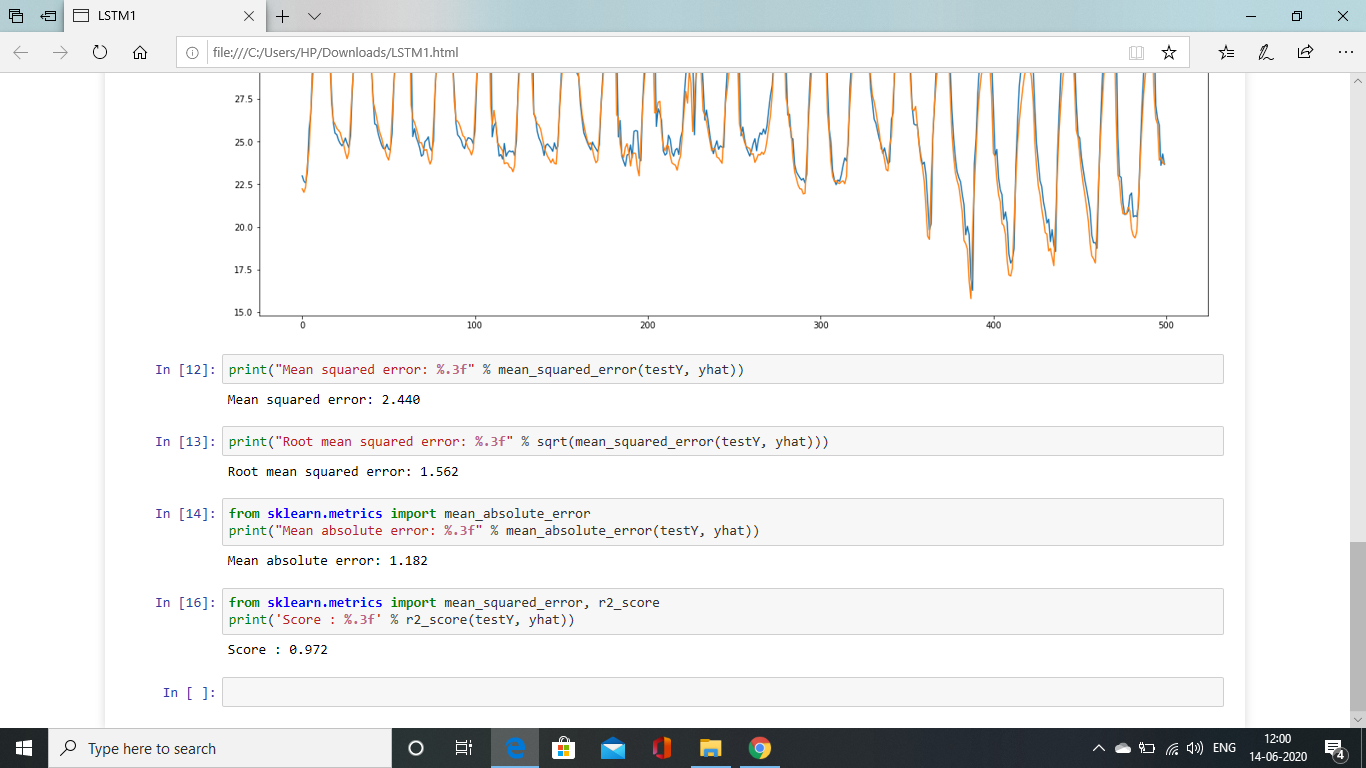
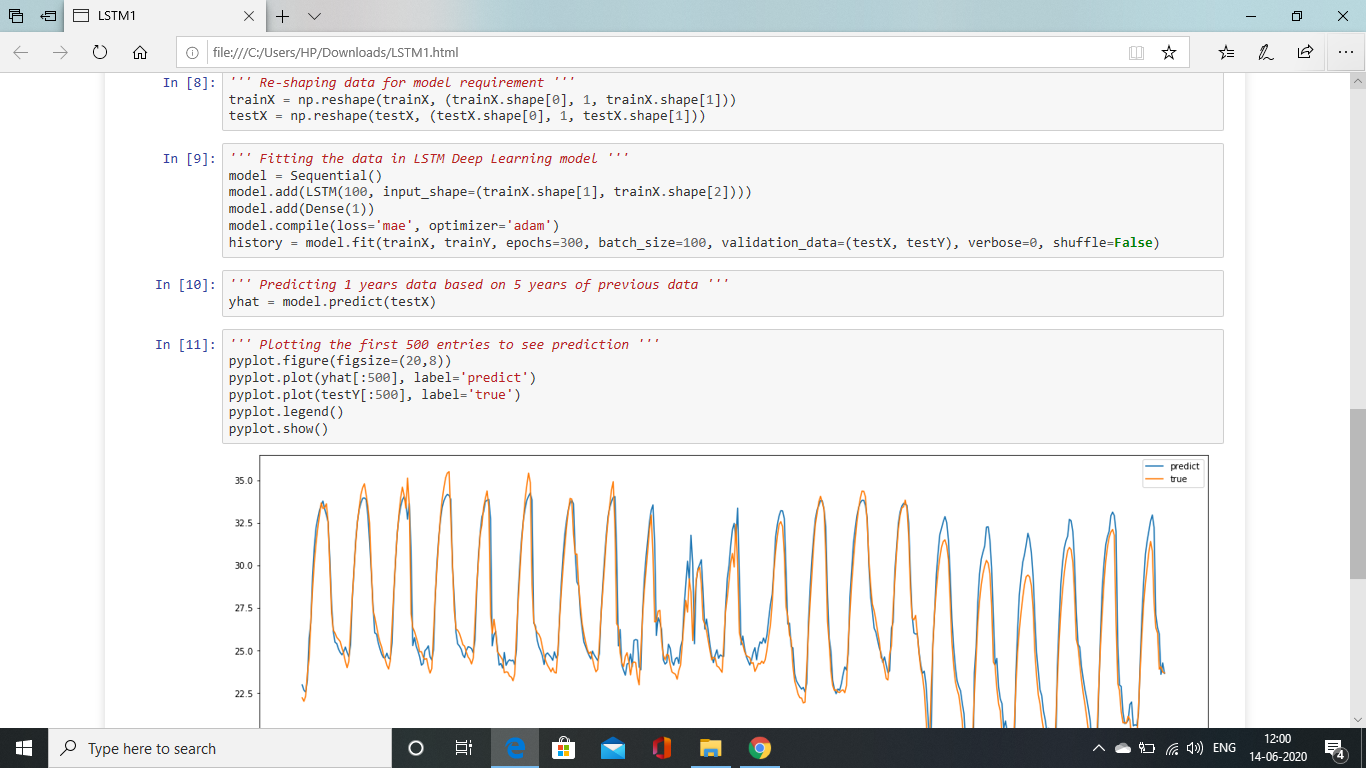
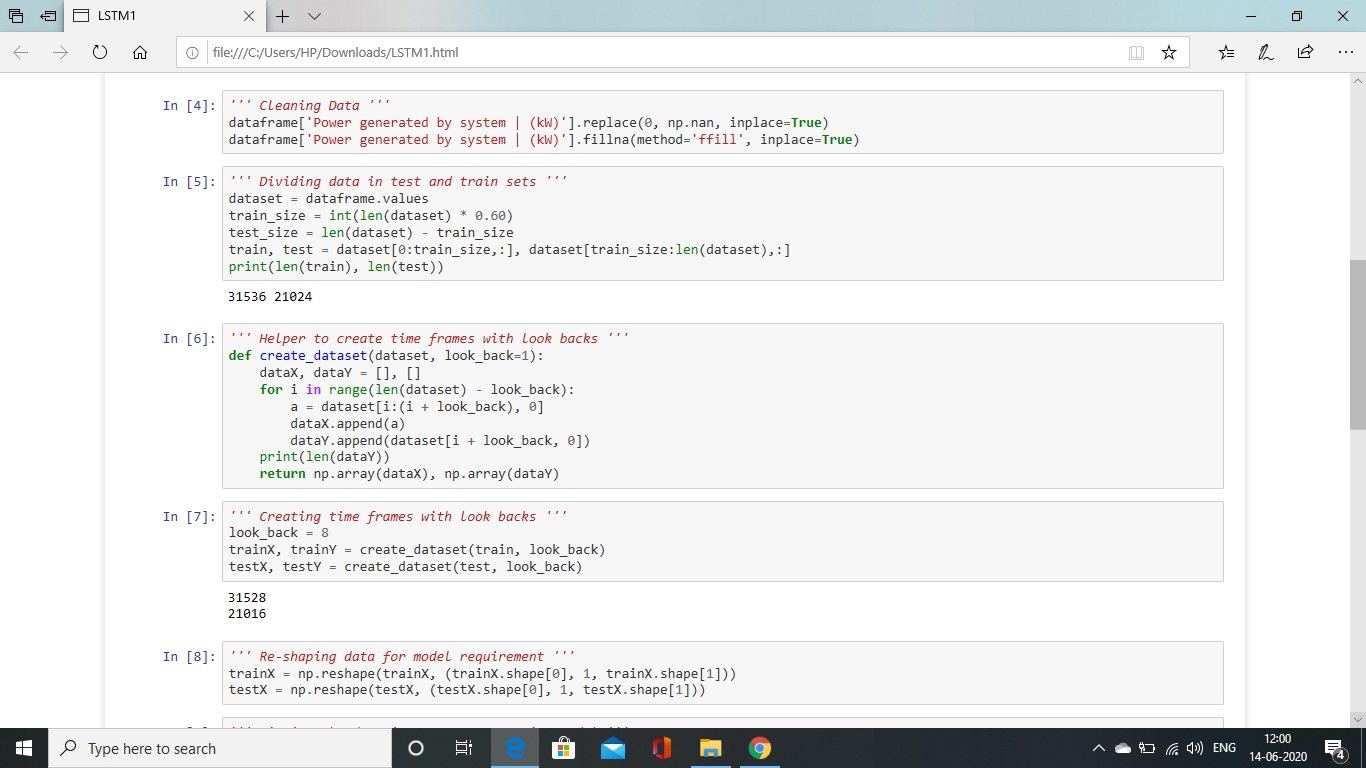
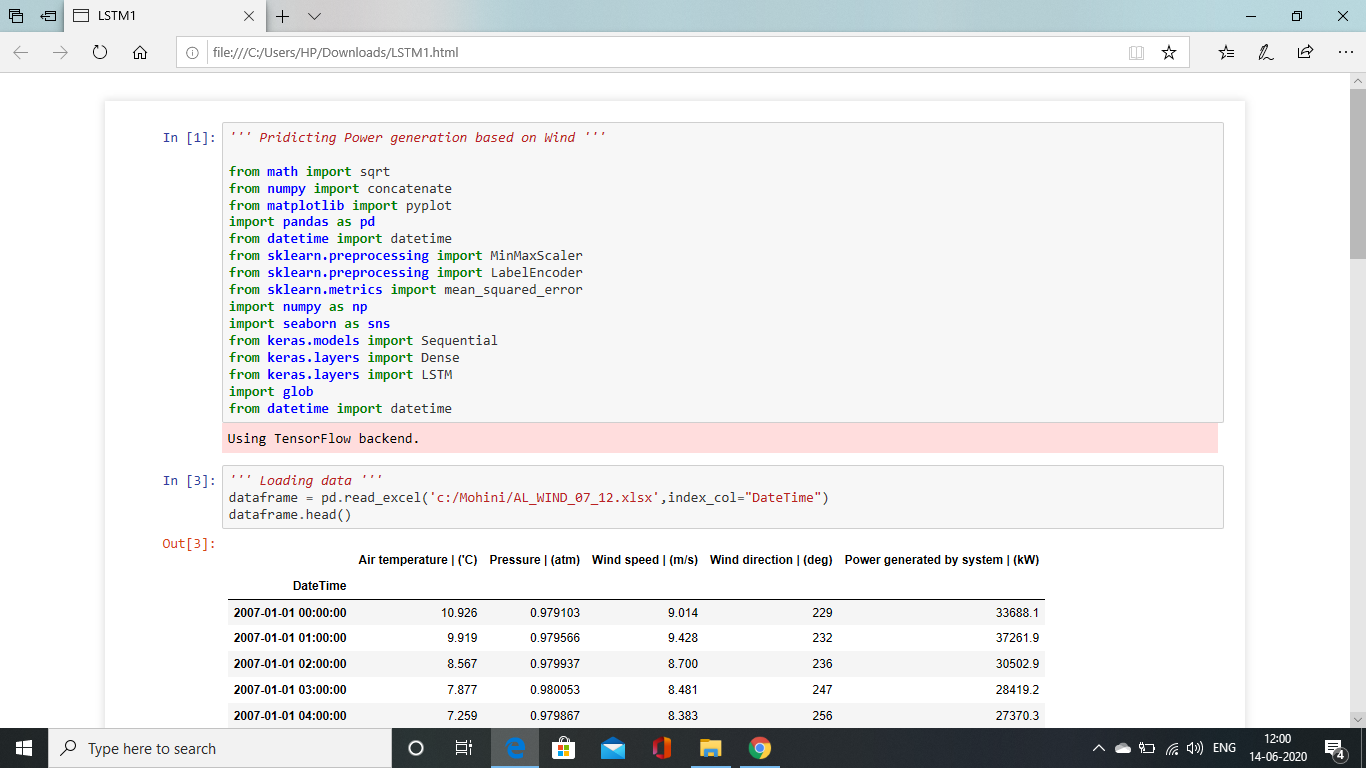
Random forest, like its name implies, consists of a large number of individual decision trees that operate as ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model’s prediction.



**Time Series Analysis:**

**LSTM(Long Short Term Memory):**

Long short-term memory (**LSTM**) is an artificial recurrent neural network (**RNN**) architecture used in the field of deep learning.**LSTM** networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series.



**For our project the LSTM was the best model to predict the expected energy output. As the RMSE was 1.562 and Accuracy score was 0.972.**

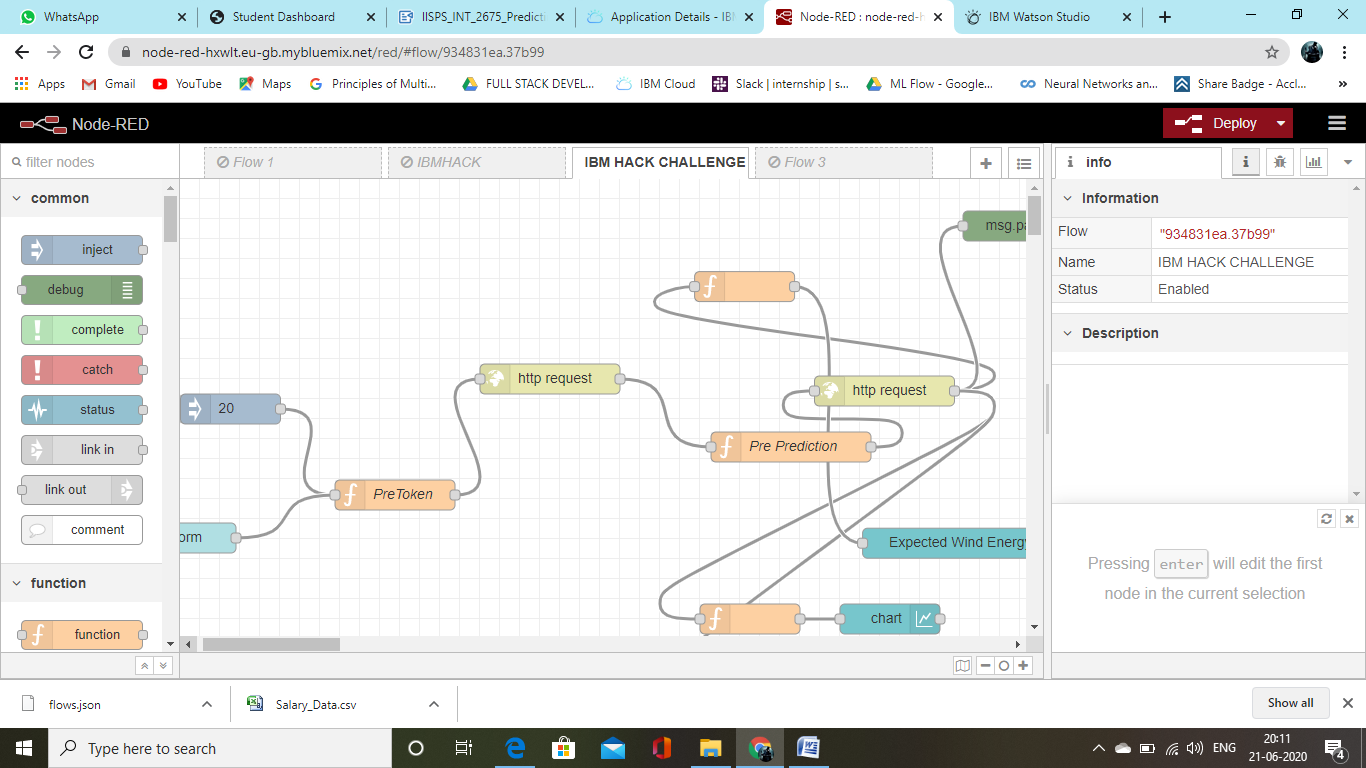
**Creating the user interface using node-red:**

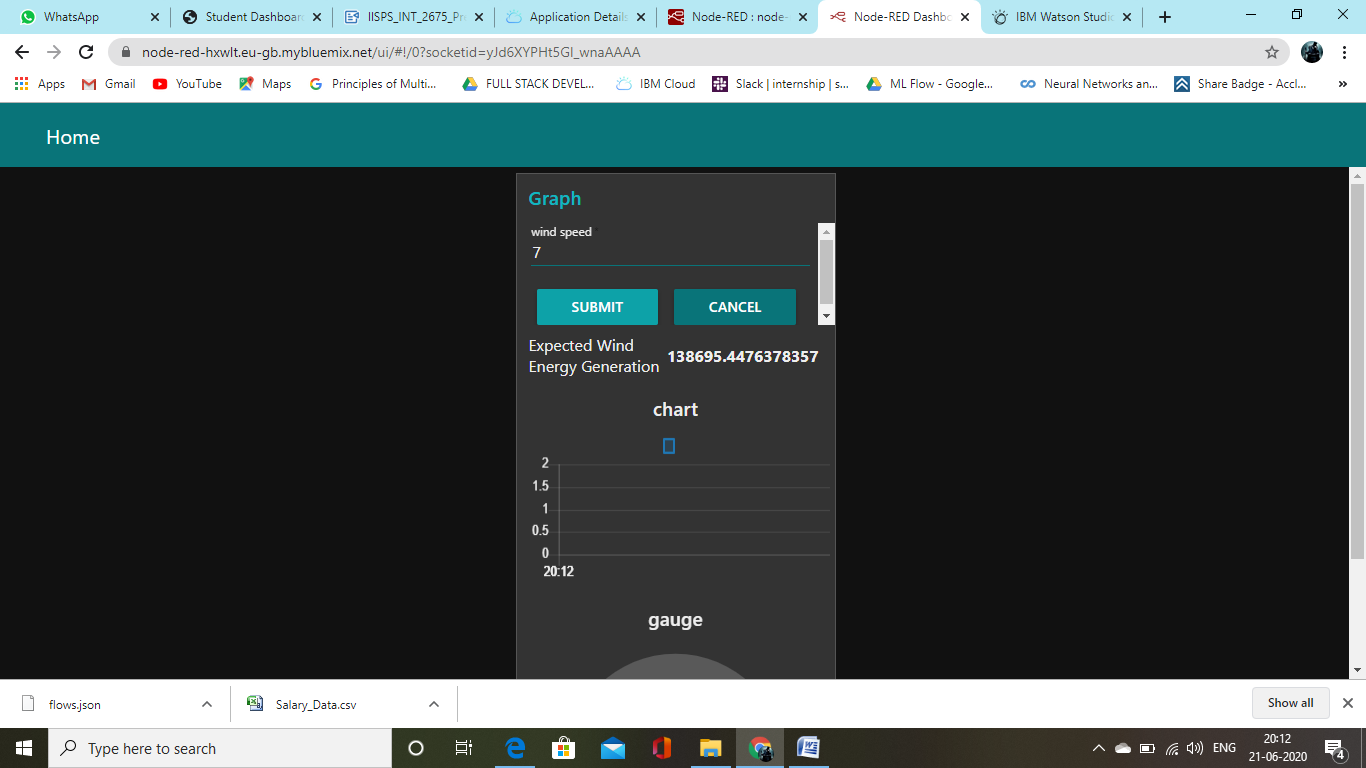
We have created the ui using node-red services from IBM cloud.

The json code for our node-red integration:

[{"id":"934831ea.37b99","type":"tab","label":"IBM HACK CHALLENGE","disabled":false,"info":""},{"id":"9cd92039.f8e1d","type":"function","z":"934831ea.37b99","name":"PreToken","func":"global.set(\"ws\",msg.payload.ws)\nvar apikey=\"XGic4OmC62qEgmkhYI5U9zVPY\_95WNNpHI7xUM6wX\_cm\";\nmsg.headers={\"content-type\":\"application/x-www-form-urlencoded\"}\nmsg.payload={\"grant\_type\":\"urn:ibm:params:oauth:grant-type:apikey\",\"apikey\":apikey}\nreturn msg;","outputs":1,"noerr":0,"x":255,"y":316.0000047683716,"wires":[["ddd539cd.b8afb8"]]},{"id":"ddd539cd.b8afb8","type":"http request","z":"934831ea.37b99","name":"","method":"POST","ret":"obj","paytoqs":false,"url":"https://iam.cloud.ibm.com/identity/token","tls":"","persist":false,"proxy":"","authType":"","x":410,"y":200,"wires":[["379a9b23.cba6e4"]]},{"id":"53945025.dd3e7","type":"inject","z":"934831ea.37b99","name":"","topic":"","payload":"20","payloadType":"num","repeat":"","crontab":"","once":false,"onceDelay":0.1,"x":90.5,"y":230.00000286102295,"wires":[["9cd92039.f8e1d"]]},{"id":"f58f38f7.a69d18","type":"debug","z":"934831ea.37b99","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType":"msg","x":893.0000114440918,"y":47.00000190734863,"wires":[]},{"id":"379a9b23.cba6e4","type":"function","z":"934831ea.37b99","name":"Pre Prediction","func":"var ws = global.get('ws')\nvar token=msg.payload.access\_token\nvar instance\_id=\"eef502e8-4064-4411-aaae-4bd5fa66b15f\"\nmsg.headers={'Content-Type': 'application/json',\"Authorization\":\"Bearer \"+token,\"ML-Instance-ID\":instance\_id}\nmsg.payload={\"fields\": [\"wind speed\"], \"values\": [[ws]]}\nreturn msg;","outputs":1,"noerr":0,"x":651.0000076293945,"y":268.00000381469727,"wires":[["1ff113bc.a9b1bc"]]},{"id":"1ff113bc.a9b1bc","type":"http request","z":"934831ea.37b99","name":"","method":"POST","ret":"obj","paytoqs":false,"url":"https://eu-gb.ml.cloud.ibm.com/v3/wml\_instances/eef502e8-4064-4411-aaae-4bd5fa66b15f/deployments/c5374fd0-6139-4118-91d8-f7917260dc18/online","tls":"","persist":false,"proxy":"","authType":"","x":744.5000076293945,"y":212.00000381469727,"wires":[["e9d8be70.96691","f58f38f7.a69d18","fd11fbea.542788","ca0e7ea2.3ac15"]]},{"id":"cd3565ca.f0bd58","type":"ui\_form","z":"934831ea.37b99","name":"","label":"","group":"27d4fb47.a691d4","order":1,"width":0,"height":0,"options":[{"label":"wind speed","value":"ws","type":"number","required":true,"rows":null}],"formValue":{"ws":""},"payload":"","submit":"submit","cancel":"cancel","topic":"","x":46,"y":359,"wires":[["9cd92039.f8e1d"]]},{"id":"1cad939d.b66bdc","type":"ui\_text","z":"934831ea.37b99","group":"27d4fb47.a691d4","order":2,"width":0,"height":0,"name":"","label":"Expected Wind Energy Generation","format":"{{msg.payload}}","layout":"row-spread","x":862.6000366210938,"y":364.20001220703125,"wires":[]},{"id":"e9d8be70.96691","type":"function","z":"934831ea.37b99","name":"","func":"msg.payload=msg.payload.values[0][0]\nreturn msg;","outputs":1,"noerr":0,"x":604.6000366210938,"y":107.60000610351562,"wires":[["1cad939d.b66bdc"]]},{"id":"fd11fbea.542788","type":"function","z":"934831ea.37b99","name":"","func":"msg.payload=msg.payload.values.length\nreturn msg;\n","outputs":1,"noerr":0,"x":610,"y":440,"wires":[["c56c6fae.6cce4"]]},{"id":"c56c6fae.6cce4","type":"ui\_chart","z":"934831ea.37b99","name":"","group":"27d4fb47.a691d4","order":3,"width":0,"height":0,"label":"chart","chartType":"line","legend":"true","xformat":"HH:mm","interpolate":"linear","nodata":"","dot":false,"ymin":"","ymax":"","removeOlder":1,"removeOlderPoints":"","removeOlderUnit":"3600","cutout":0,"useOneColor":false,"useUTC":false,"colors":["#1f77b4","#aec7e8","#ff7f0e","#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"useOldStyle":false,"outputs":1,"x":750,"y":440,"wires":[[]]},{"id":"8ae16b0b.8c0f58","type":"ui\_gauge","z":"934831ea.37b99","name":"","group":"27d4fb47.a691d4","order":3,"width":0,"height":0,"gtype":"gage","title":"gauge","label":"units","format":"{{value}}","min":0,"max":"1000000","colors":["#00b500","#e6e600","#ca3838"],"seg1":"","seg2":"","x":770,"y":520,"wires":[]},{"id":"ca0e7ea2.3ac15","type":"function","z":"934831ea.37b99","name":"","func":"msg.payload=msg.payload.values;\nreturn msg;\n","outputs":1,"noerr":0,"x":610,"y":520,"wires":[["8ae16b0b.8c0f58"]]},{"id":"27d4fb47.a691d4","type":"ui\_group","z":"","name":"Graph","tab":"a7c74627.0fe648","order":1,"disp":true,"width":"6","collapse":false},{"id":"a7c74627.0fe648","type":"ui\_tab","z":"","name":"Home","icon":"dashboard","disabled":false,"hidden":false}]

The flow of node-red:

The final output:



**Future Scope:**

**Models for Wind Power Ramp Forecasting**:  The variability in wind power can present a substantial challenge to the grid, when the penetration of wind power is high. The ramp event is a very critical issue and is characterized by a sudden large change (increment or decrement) in wind power. Accurate models for ramp event detection and forecasting is important for maintaining the stability of electrical grid.

**Estimation of Forecast Uncertainty Uncertainty:** Analysis of the wind forecasts made plays a key role in grid integration and other power system operations.

**Use of Wind Power Forecasting in Power System Operations**: Accurate models for wind power forecasting aids in several power system operations like operating reserve requirements, unit commitment, dispatch formulations etc.

**Condition Monitoring of Wind Farms**: Good performing forecasting models for wind power and wind turbine power curve models can also be used as performance indicator for the health of a wind turbine.

**Wind Forecasting and Offshore Wind Farms:** Due to limited land area in several regions, many countries are already into offshore wind farms. These offshore wind farms have opened up huge prospects of research in operation, maintenance, control of wind farms and wind resource assessment to identify potential sites etc.