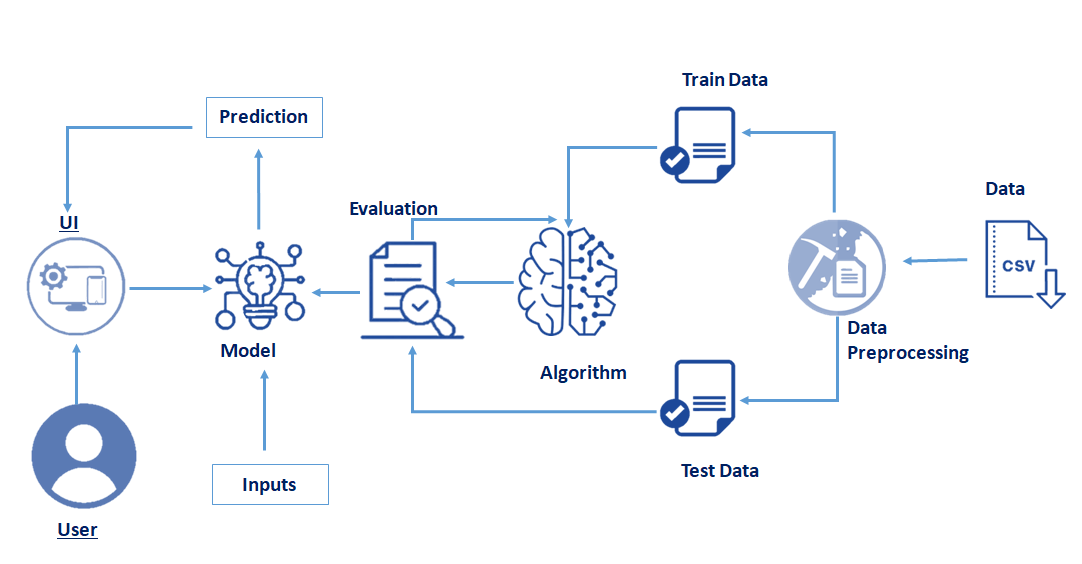
**Rainfall Prediction**

**Project Description:**

Rainfall now a days occurring irrespective of seasons, Particularly during the torrential rainfall event. Moreover, one of the major focuses of Climate change study is to understand whether there are extreme changes in the occurrence and frequency of heavy rainfall events. The accuracy level of the ML models used in predicting rainfall based on historical data has been one of the most critical concerns in hydrological studies. An accurate ML model could give early alerts of severe weather to help prevent natural disasters and destruction. Hence, there is needs to develop ML algorithms capable in predicting rainfall with acceptable level of precision and in reducing the error in the dataset of the projected rainfall from climate change model with the expected observable rainfall.

**Technical Architecture:**

****

**Pre requisites:**

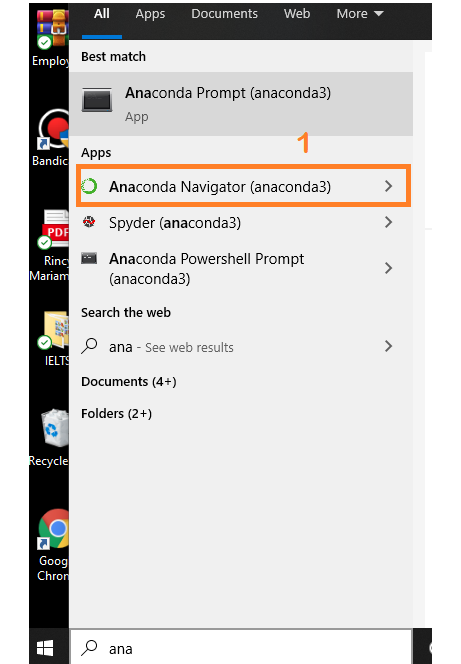
**To complete this project, you must require following software’s , concepts and packages**

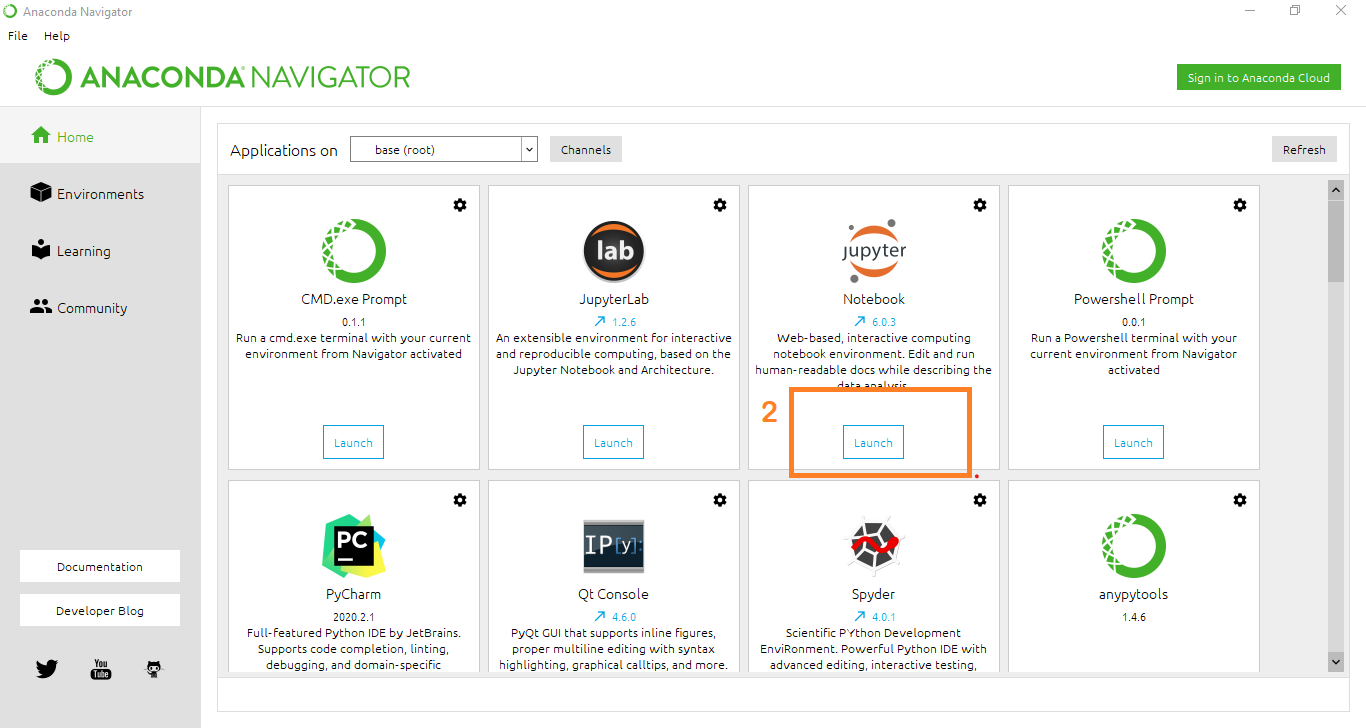
* **Anaconda navigator:**
  + Refer to the link below to download anaconda navigator
  + **Link : https://www.youtube.com/watch?v=5mDYijMfSzs**
* **Python packages:**
  + Open anaconda prompt as administrator.
  + Type “pip install numpy” and click enter.
  + Type “pip install pandas” and click enter.
  + Type “pip install matplotlib” and click enter.
  + Type “pip install scikit-learn” and click enter.
  + Type “pip install Flask” and click enter.

The above steps allow you to install the packages in the anaconda environment

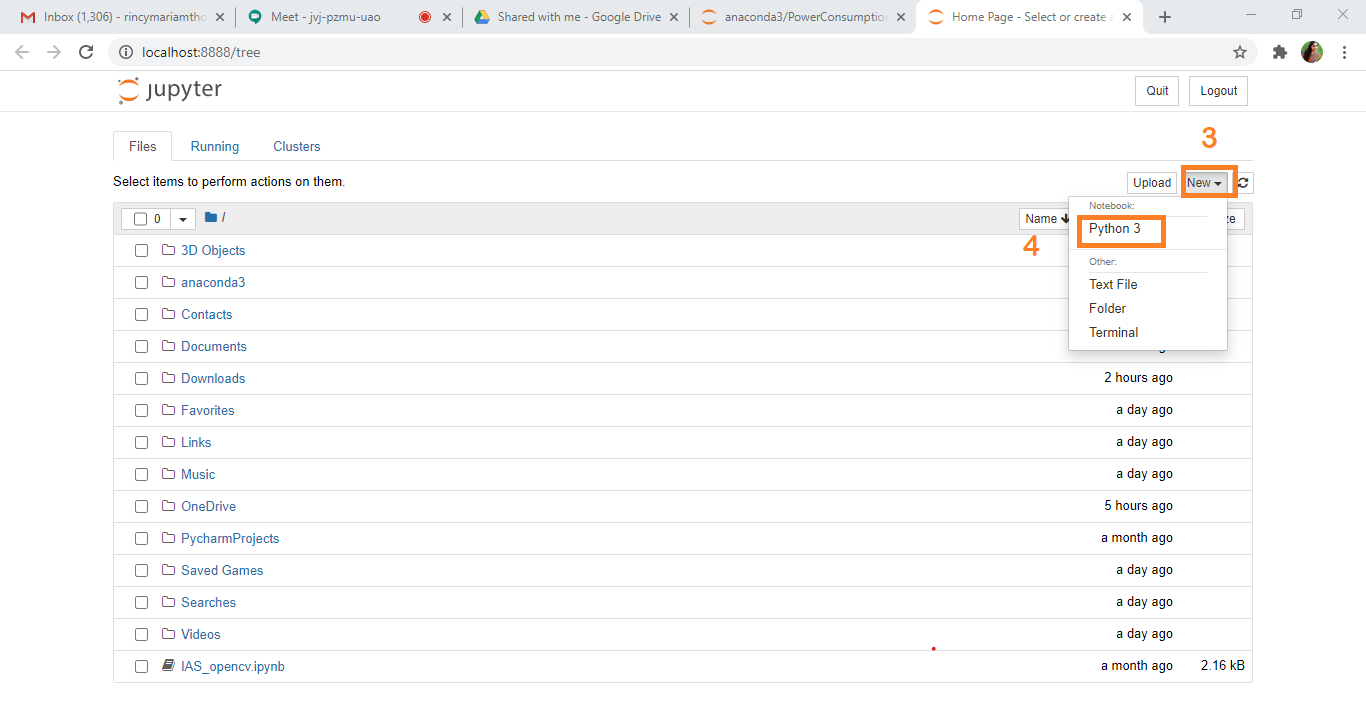
**Prior Knowledge:**

* **Machine Learning concepts**
  + **Supervised learning:** [**https://www.youtube.com/watch?v=QeKshry8pWQ**](https://www.youtube.com/watch?v=QeKshry8pWQ)
  + **Unsupervised learning:** [**https://www.youtube.com/watch?v=D6gtZrsYi6c**](https://www.youtube.com/watch?v=D6gtZrsYi6c)
  + **Metrics :** [**https://www.youtube.com/watch?v=aWAnNHXIKww**](https://www.youtube.com/watch?v=aWAnNHXIKww)
  + [**https://www.youtube.com/watch?v=YSB7FtzeicA**](https://www.youtube.com/watch?v=YSB7FtzeicA)
* **Launch Jupyter**
  + Search for Anaconda Navigator and open Launch Jupyter notebook.





* Then you will be able to see that the jupyter notebook runs on local host:8888.
* To Create a new file Go to New 🡪Python3.The file in jupyter notebook is saved with .ipynb extension.



* Flask Basics : <https://www.youtube.com/watch?v=lj4I_CvBnt0>

**Project Objectives:**

By the end of this project:

* You’ll be able to understand the problem to classify if it is a regression or a classification kind of problem.
* You will be able to know how to pre-process / clean the data using different data preprocessing techniques.
* You will able to analyse or get insights of data through visualization.
* Applying different algorithms according to dataset and based on visualization.
* You will able to know how to find accuracy of the model.
* You will be able to know how to build a web application using Flask framework.

**Project Flow:**

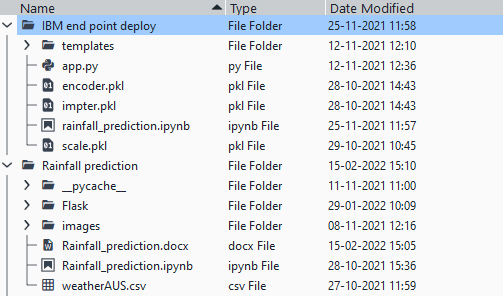
* User interacts with the UI (User Interface) to enter the input values
* Entered input values are analyzed by the model which is integrated
* Once model analyses the input the prediction is showcased on the UI

To accomplish this, we have to complete all the activities and tasks listed below

* Data Collection.
* Collect the dataset or Create the dataset
* Data Preprocessing.
  + Import the Libraries.
  + Importing the dataset.
  + Checking for Null Values.
  + Data Visualization.
  + Taking care of Missing Data.
  + Feature Scaling.
  + Splitting Data into Train and Test.
* Model Building
  + Import the model building Libraries
  + Initializing the model
  + Training and testing the model
  + Evaluation of Model
  + Save the Model
* Application Building
  + Create an HTML file
  + Build a Python Code

**Project Structure:**

Create a Project folder which contains files as shown below



* A python file called app.py for server side scipting.
* We need the model which is saved and the saved model in this content is **Rainfall.pkl**
* Templates folder which contains index.HTML file, chance.HTML file, noChance.HTML file.
* Scale.pkl for scaling,encoder.pkl file for encoding the categorical data,imputer.pkl file for filling out the missing values

**Milestone 1: Data Collection:**

ML depends heavily on data, without data, it is impossible for an “AI” to learn. It is the most crucial aspect that makes algorithm training possible. In Machine Learning projects, we need a training **data set.**It is the actual **data set** used to train the model for performing various actions.

**Activity1: Download The dataset**

You can collect datasets from different open sources like kaggle.com, data.gov, UCI machine learning repository etc.

Please refer to the link given below to download the data set and to know about the dataset

https://docs.google.com/spreadsheets/d/1RA2OO0LZTeQykI\_mvnensAjp6LM4YzWI1Tz0SUG5-Ao/edit#gid=121883362

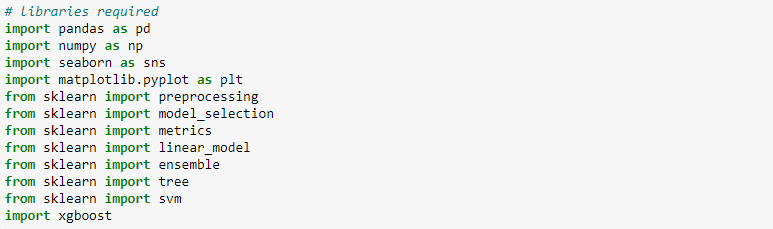
**Milestone 2: Data Preprocessing**

Data Pre-processing includes the following main tasks

* + Import the Libraries.
  + Importing the dataset.
  + Checking for Null Values.
  + Data Visualization.
  + Feature Scaling.
  + Splitting Data into Train and Test.

**Activity 1: Import Necessary Libraries**

* + It is important to import all the necessary libraries such as pandas, numpy, matplotlib.
  + **Numpy**- It is an open-source numerical Python library. It contains a multi-dimensional array and matrix data structures. It can be used to perform mathematical operations on arrays such as trigonometric, statistical, and algebraic routines.
  + **Pandas**- It is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.
  + **Seaborn**- Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.
  + **Matplotlib**- Visualisation with python. It is a comprehensive library for creating static,animated, and interactive visualizations in Python
  + **Sklearn** – which contains all the modules required for model building



**Activity 2: Importing the Dataset**

* You might have your data in .csv files, .excel files
* Let’s load a .csv data file into pandas using **read\_csv() function.**We will need to locate the directory of the CSV file at first (it’s more efficient to keep the dataset in the same directory as your program).
* If your dataset is in some other location ,Then

**Data=pd.read\_csv(r”File\_location/datasetname.csv”)**

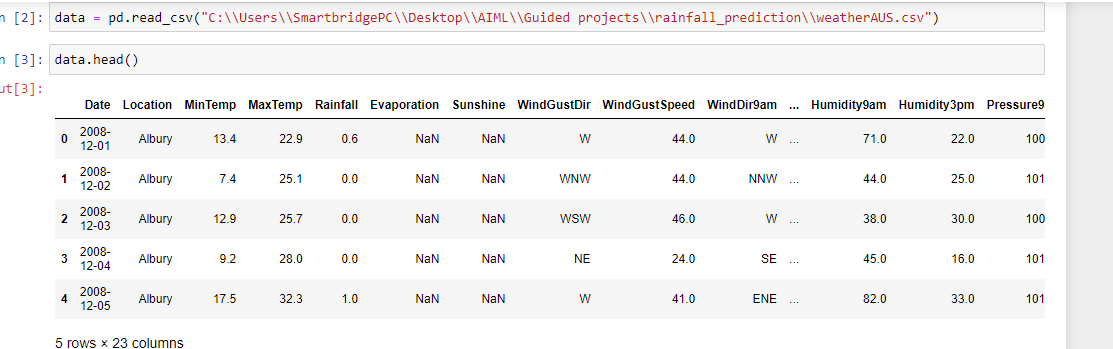
**Note:**r stands for "raw" and will cause backslashes in the string to be interpreted as actual backslashes rather than special characters.

* If the dataset in same directory of your program, you can directly read it, without giving raw as r.
* Our Dataset weatherAus.csv contains following Columns
* Location, MinTemp, MaxTemp, Rainfall, WindGustSpeed,
* WindSpeed9am, WindSpeed3pm, Humidity9am, Humidity3pm
* Pressure9am, Pressure3pm, Temp9am, Temp3pm, RainToday,
* WindGustDir, WindDir9am, WindDir3pm,date
* Raintommorrow – output column

The output column to be predicted is **RainTommorow** .Based on the input variables we predict the chance of rain. The predicted output gives them a fair idea about it will rain or not.

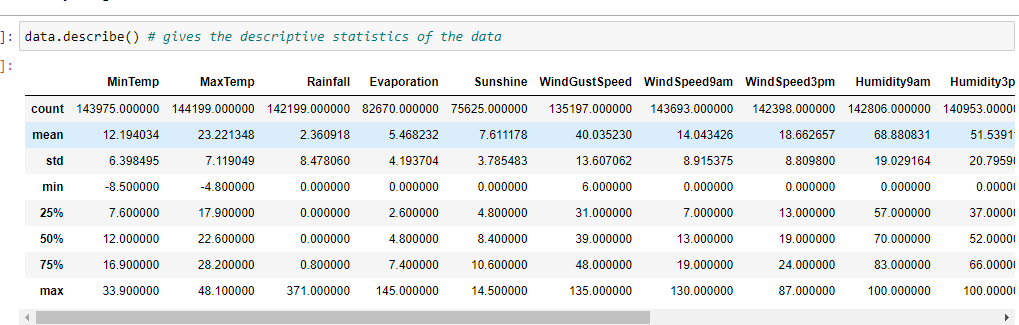
**Activity 3: Analyse the data**

* head() method is used to return top n (5 by default) rows of a DataFrame or series.



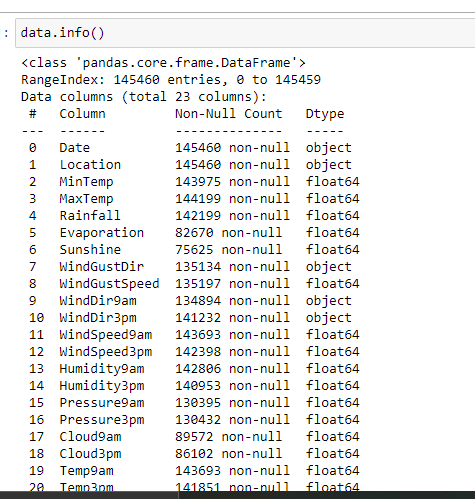
* describe() method computes a summary of statistics like count, mean, standard deviation, min, max and quartile values.

The output is as shown below



From the data we infer that there are only decimal values and no categorical values

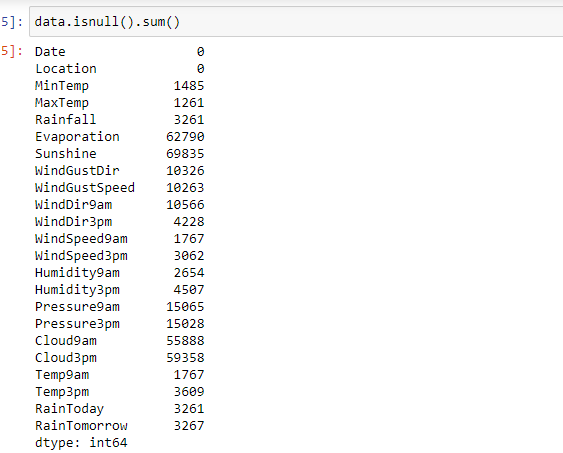
* info() gives information about the data



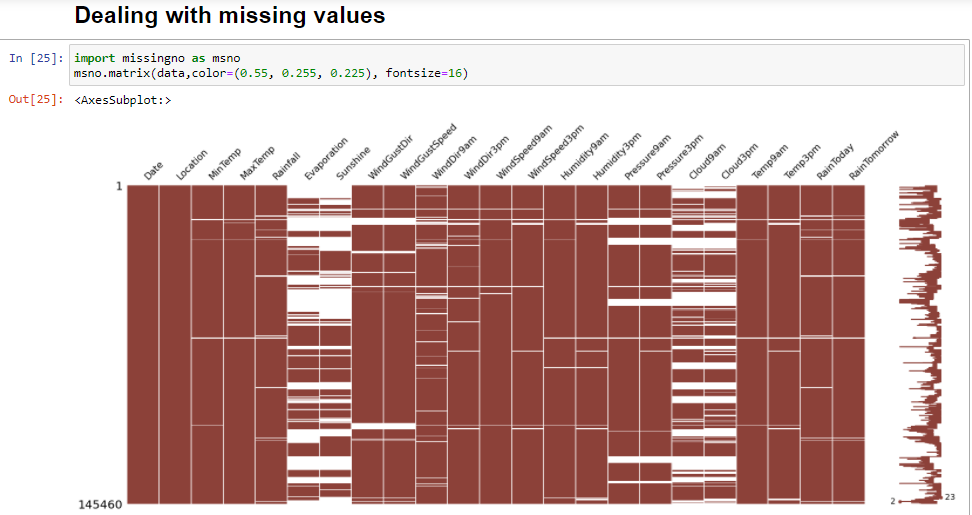
**Activity 4: Handling Missing Values**

1. After loading it is important to check the complete information of data as it can indication many of the hidden information such as null values in a column or a row

2.Check whether any null values are there or not. if it is present then following can be done,



#### 3. Missing matrix: It is way of representing the data in 2-D form. It gives coloured visual summary of the data



4.Imputing data using Imputation method in sklearn.Simpleimputer

a.Filling NaN values with mean, median and mode using fillna() method.



From the heatmap, we see that there are missing values in the dataset

**Activity 5: Data Visualisation**

* Data visualization is where a given data set is presented in a graphical format. It helps the detection of patterns, trends and correlations that might go undetected in text-based data.
* Understanding your data and the relationship present within it is just as important as any algorithm used to train your machine learning model. In fact, even the most sophisticated machine learning models will perform poorly on data that wasn’t visualized and understood properly.
* To visualize the dataset we need libraries called Matplotlib and Seaborn.
* The Matplotlib library is a Python 2D plotting library which allows you to generate plots, scatter plots, histograms, bar charts etc.

Let’s visualize our data using Matplotlib and searborn library.

Before diving into the code, let's look at some of the basic properties we will be using when plotting.

**xlabel:** Set the label for the x-axis.

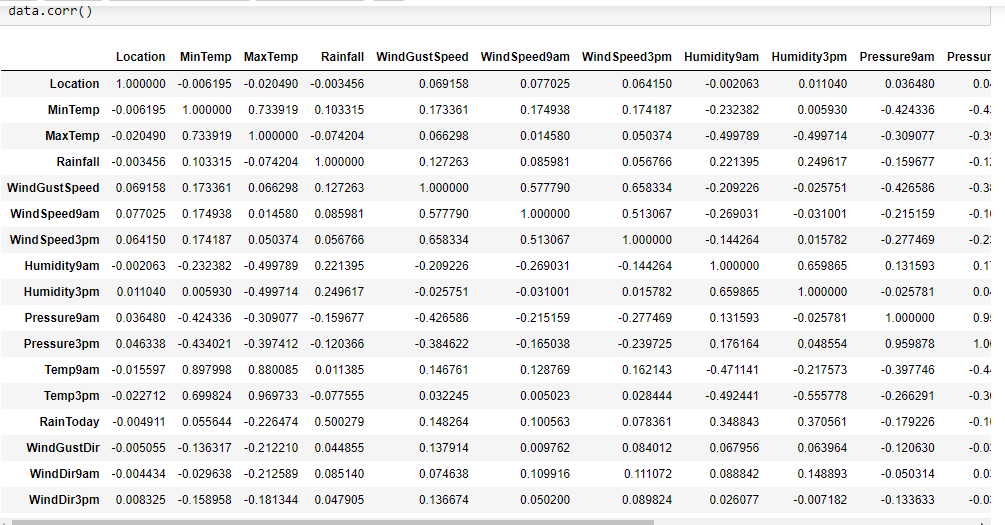
**ylabel:** Set the label for the y-axis.

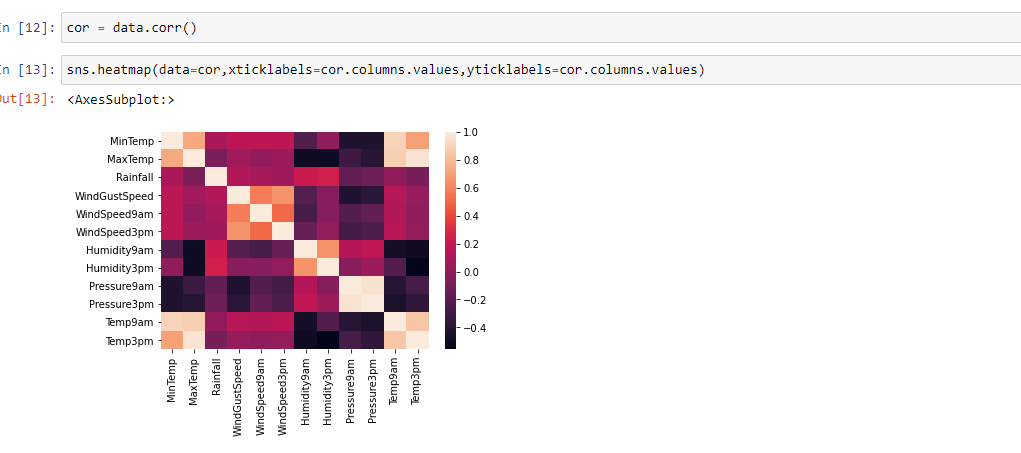
**title:** Set a title for the axes.

**Legend:** Place a legend on the axes.

1. data.corr() gives the correlation between the columns

**Correlation** is a statistical term describing the degree to which two variables move in coordination with one another. If the two variables move in the same direction, then those variables are said to have a positive correlation. If they move in opposite directions, then they have a negative correlation.





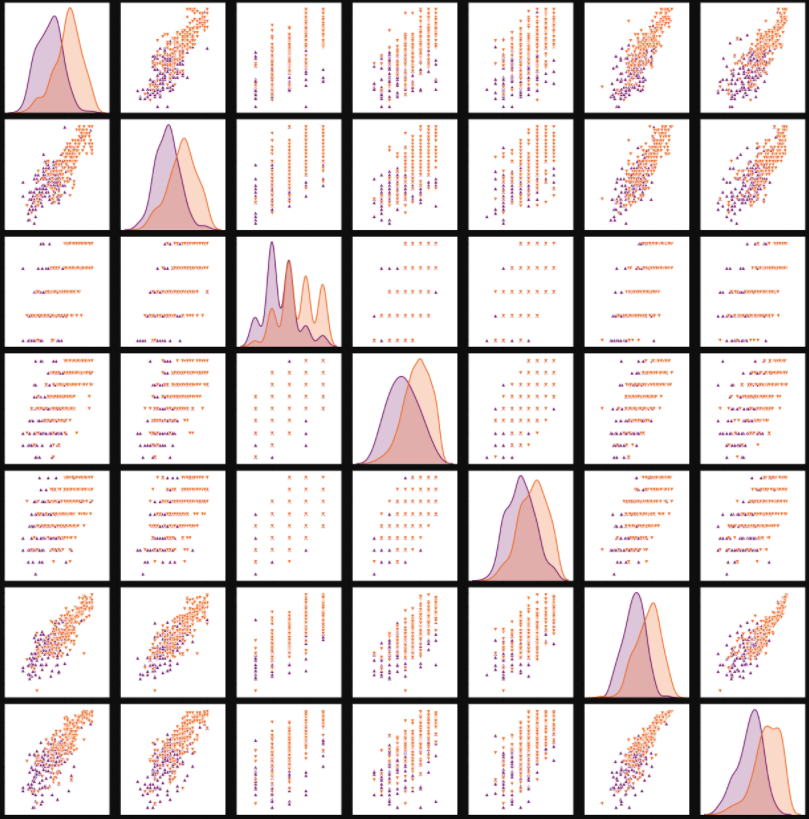
* Correlation strength varies based on colour, lighter the colour between two variables, more the strength between the variables, darker the colour displays the weaker correlation
* We can see the correlation scale values on left side of the above image

2.Pair Plot: Plot pairwise relationships in a dataset.

* By default, this function will create a grid of Axes such that each numeric variable in data will by shared across the y-axes across a single row and the x-axes across a single column. The diagonal plots are treated differently: a univariate distribution plot is drawn to show the marginal distribution of the data in each column.
* We implement this using the below code

**Code:- sns.pairplot(data)**

The output is as shown below



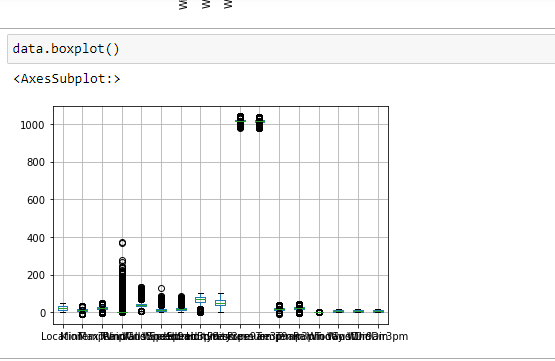
Pair plot usually gives pair wise relationships of the columns in the dataset

From the above pairplot we infer that

1.from the above plot we can draw inferences such as linearity and strength between the variables

2.how features are correlated(positive, neutral and negative)

3.Box Plot: jupyter has a built-in function to create boxplot called boxplot(). A boxplot plot is a type of plot that shows the spread of data in all the quartiles



From the above box plot we infer how the datapoints are spread and the existence of the outliers

**Activity 6: Splitting the Dataset into Dependent and Independent variable**

* In machine learning, the concept of dependent variable (y) and independent variables(x) is important to understand. Here, Dependent variable is nothing but output in dataset and independent variable is all inputs in the dataset.
* With this in mind, we need to split our dataset into the matrix of independent variables and the vector or dependent variable. Mathematically, Vector is defined as a matrix that has just one column.

To read the columns, we will use **iloc** of pandas (used to fix the indexes for selection) which takes two parameters — [row selection, column selection].

Let’s split our dataset into independent and dependent variables.

**y = data['RainTomorrow'] - independent**

**x = data.drop('RainTomorrow',axis=1)**

**Activity 7: Feature Scaling**

There is huge disparity between the x values so let us use feature scaling.

Feature scaling is a method used to normalize the range of independent variables or features of data.

* After scaling the data will be converted into array form
* Loading the feature names before scaling and converting them back to dataframe after standard scaling is applied

**Activity 8: Splitting the data into Train and Test**

* When you are working on a model and you want to train it, you obviously have a dataset. But after training, we have to test the model on some test dataset. For this, you will a dataset which is different from the training set you used earlier. But it might not always be possible to have so much data during the development phase. In such cases, the solution is to split the dataset into two sets, one for training and the other for testing.
* The train-test split is a technique for evaluating the performance of a machine learning algorithm.
* **Train Dataset**: Used to fit the machine learning model.
* **Test Dataset**: Used to evaluate the fit machine learning model.
* In general you can allocate 80% of the dataset to training set and the remaining 20% to test set.
* Now split our dataset into train set and test using train\_test\_split class from scikit learn library.

**from sklearn import model\_selection**

**x\_train,x\_test,y\_train,y\_test=model\_selection.train\_test\_split(x,y,test\_size=0.2,random\_state =0)**

**Milestone 3: Model Building:**

Model building includes the following main tasks

* + Import the model building Libraries
  + Initializing the model
  + Training and testing the model
  + Evaluation of Model
  + Save the Model

**Activity 1: Training and Testing the Model**

* Once after splitting the data into train and test, the data should be fed to an algorithm to build a model.
* There are several Machine learning algorithms to be used depending on the data you are going to process such as images, sound, text, and numerical values. The algorithms that you can choose according to the objective that you might have it may be Classification algorithms are Regression algorithms.

1.Logistic Regression

2.Decision Tree Classifier

3.Random Forest Classifier

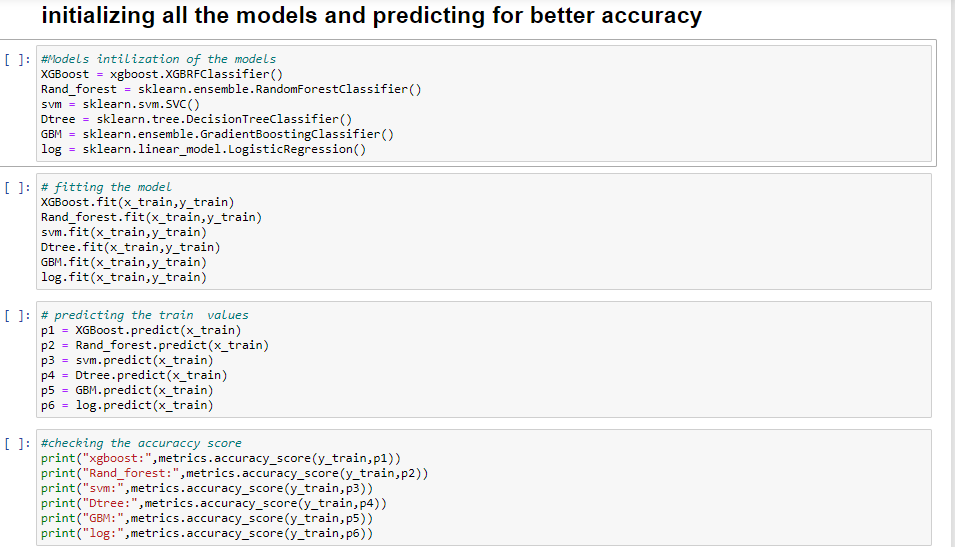
4.KNN

5.svm

5.xgboost

**Steps in Building the model:-**

* **Initialize the model**
* **Fit the models with x\_train and y\_train**
* **Predict the y\_train values and calculate the accuracy**
* **Predict the y\_test values and calculate the accuracy**



We’re going to use x\_train and y\_train obtained above in train\_test\_split section to train our decision tree regression model. We’re using the fit method and passing the parameters as shown below.

We are using the algorithm from Scikit learn library to build the model as shown below,

Once the model is trained, it’s ready to make predictions. We can use the **predict** method on the model and pass **x\_test** as a parameter to get the output as **y\_pred.**

Notice that the prediction output is an array of real numbers corresponding to the input array.

**Activity 2: Model Evaluation**

After training the model, the model should be tested by using the test data which is been separated while splitting the data for checking the functionality of the model.

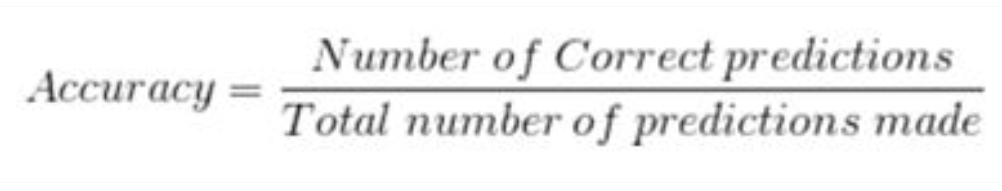
**Regression Evaluation Metrics:**

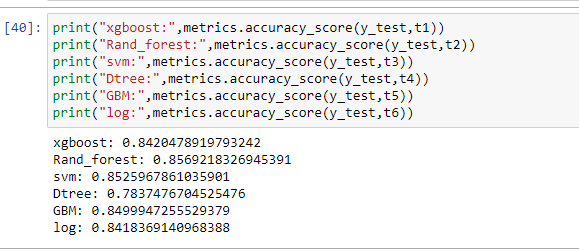
These model evaluation techniques are used to find out the accuracy of models built in classification type of machine learning models. We have three types of evaluation methods.

* Accuracy\_score
* Confusion matrix
* Roc- Auc Curve

1. Accuracy\_score

It is the ratio of number of correct predictions to the total number of input samples.





Select the model,which gives the best accuracy of all,and generate predictions and find the accuracy with training and testing data

2. Confusion Matrix

It is a matrix representation of the results of any binary testing

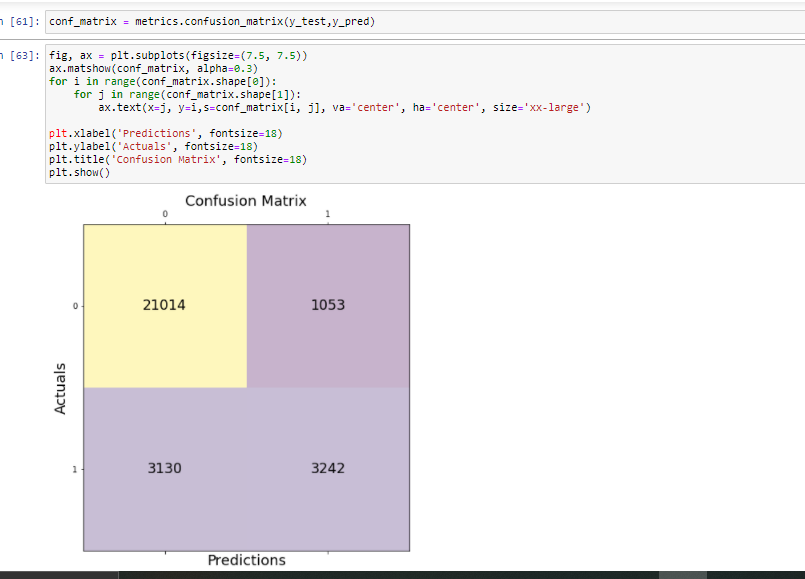


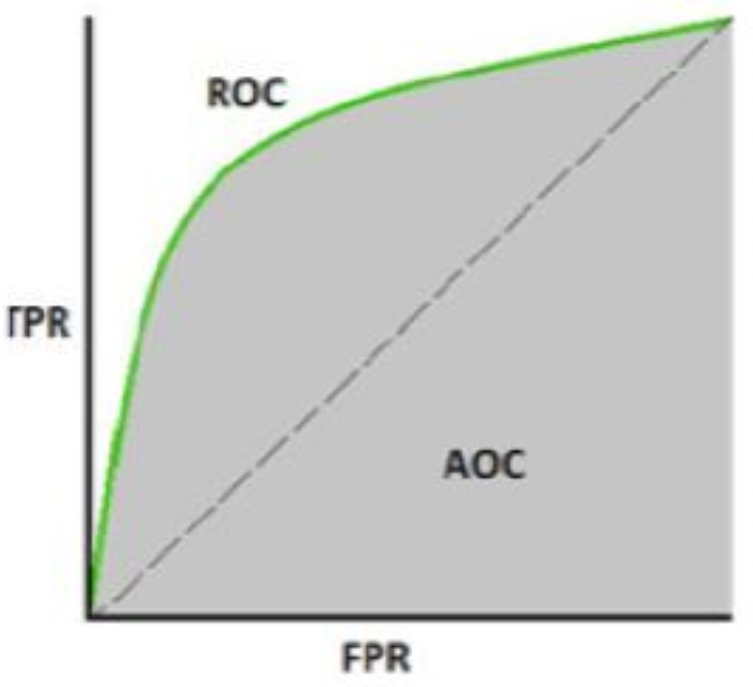
      Fig: Confusion Matrix of prediction of rainfall

1. True Positive: 3242 (You have predicted the positive case correctly!)
2. True Negative: 21014 (You have predicted negative case correctly!)
3. False Positive: 1053 (You have predicted it will rain, but in actual it will not rain)
4. False Negative: 3130 (Wrong predictions )

3. Roc-Auc Curve

* AUC is the area under the ROC curve. AUC ROC indicates how well the probabilities from the positive classes are separated from the negative classes.
* AUC - ROC curve is a performance measurement for classification problem at various thresholds settings
* ROC is a probability curve and AUC represents degree or measure of separability.
* Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1sThe ROC curve is plotted with TPR against the FPR where TPR is on y-axis and FPR is on the x-axis.
* Code for Roc and performance metrics





For testing the model we use the below method,

**Activity 3: Save the Model**

After building the model we have to save the model.

**Pickle** in **Python** is primarily **used** in serializing and deserializing a **Python** object structure. In other words, it's the process of converting a **Python** object into a byte stream to store it in a file/database, maintain program state across sessions, or transport data over the network. wb indicates write method and rd indicates read method.

This is done by the below code

****

**Milestone 4 : Application Building**

In this section, we will be building a web application that is integrated to the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.

This section has the following tasks

* Building HTML Pages
* Building server side script

**Activity 1: Build HTML Code**

* + In this HTML page, we will create the front end part of the web page. In this page we will accept input from the user and Predict the values.

For more information regarding HTML

[**https://www.w3schools.com/html/**](https://www.w3schools.com/html/)

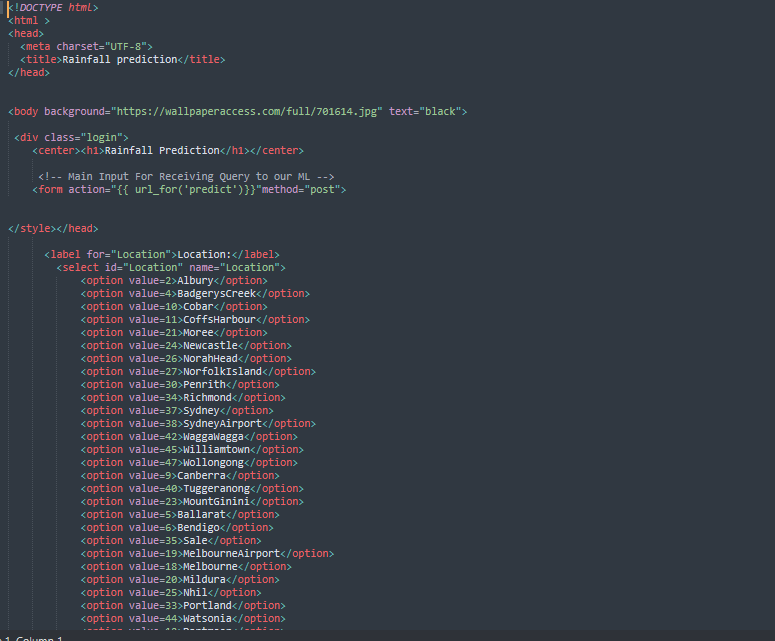
In our project we have 3 HTML files ,they are

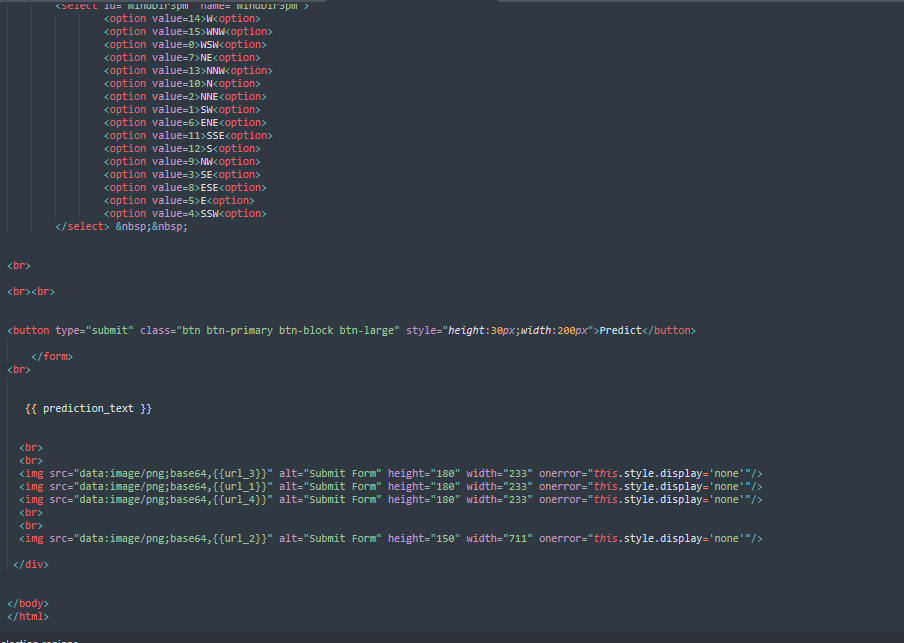
1.inex.html

2.chance.html

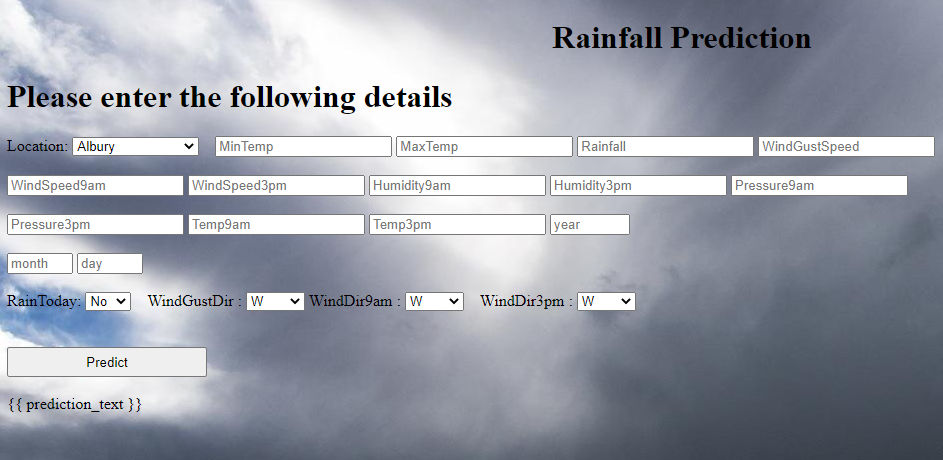
3.noChance.html

**index.html**

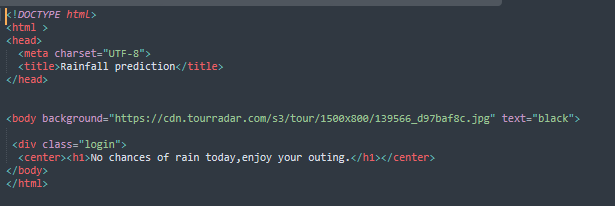
****

****

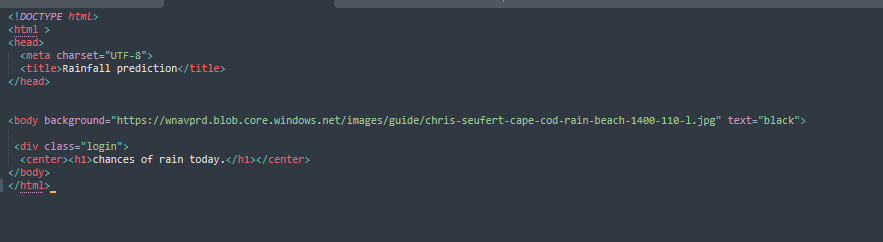
**The html page looks like**

****

**noChance.html**

****

**chance.html**

****

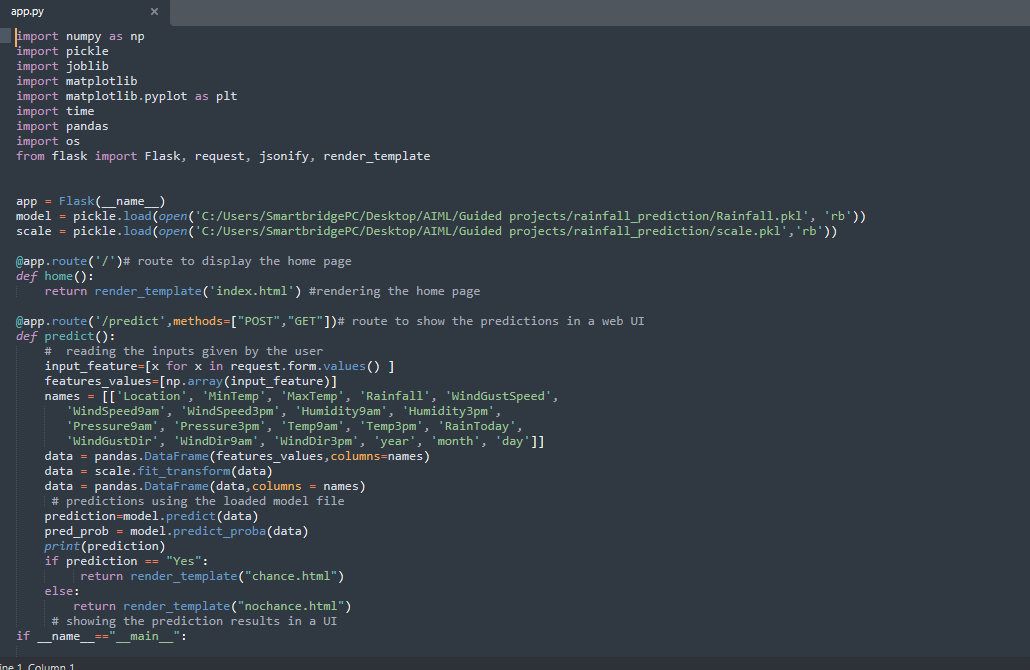
**Activity 2: Main Python Script**

Let us build app.py flask file which is a web framework written in python for server-side scripting. Let’s see step by step procedure for building the backend application.

In order to develop web api with respect to our model, we basically use Flask framework which is written in python.

* Importing the necessary libraries for building a flask application and to integrate model and html pages
* Initializing the flask app
* Calling the pkl models and saving into a variable
* Routing and rending to the html page
* Calling the inputs from the html page and saving into the variable
* Creating the data labels
* Forming the data frame with labels and the data
* Scaling the data
* Predicting the values ,by passing the data into the model
* Rendering the results on to the html pages based on the output
* If the output is class-0,it means a page which displays non potential customer will be rendered, if the output is 1 , a page with potential customer will be displayed and the output is 2 a page with highly potential customer will be rendered.

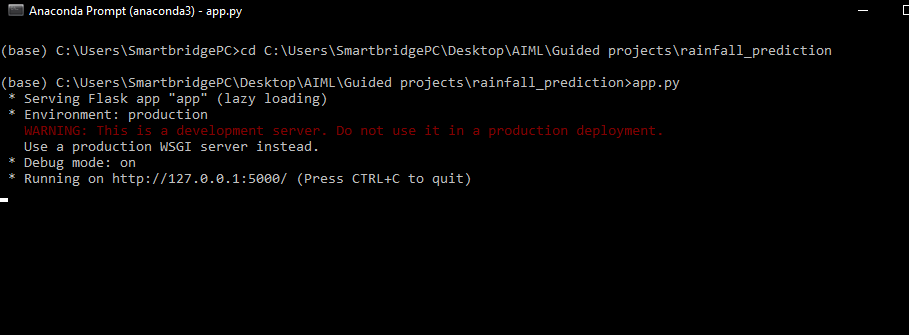
The value of \_\_name\_\_ is set to \_\_main\_\_ when module run as main program other wise it is set to name of the module

****

**Activity 3: Run the App**

* + Open anaconda prompt from the start menu
  + Navigate to the folder where your python script is.
  + Now type “python app.py” command

Navigate to the localhost where you can view your web page,Then it will run on local host:5000

****

**Activity 4:**

* Copy the http link and paste it in google link tab,it will display the form page
* Enter the values as per the form and click on predict buttion
* It will redirect to the page based on prediction output
* If the output predicted, rain will not fall, this page will be displayed on the user interface



* If the prediction is occurance of rain ,this page will be get displayed on the user interface

****