Table of Contents

[**Data cleaning** 1](#_Toc99841939)

[**Data visualization** 3](#_Toc99841940)

[**Prediction of rainfall based on the clean data** 7](#_Toc99841941)

[**Splitting data into clusters and testing the model’s accuracy** 10](#_Toc99841942)

There are four steps done to produce the output of the prediction

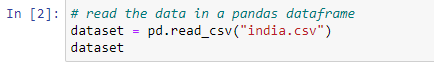
1. Data cleaning
2. Data visualization
3. Prediction of rainfall based on the clean data
4. Splitting data into clusters and testing the model’s accuracy

We begin by importing the required python libraries which include

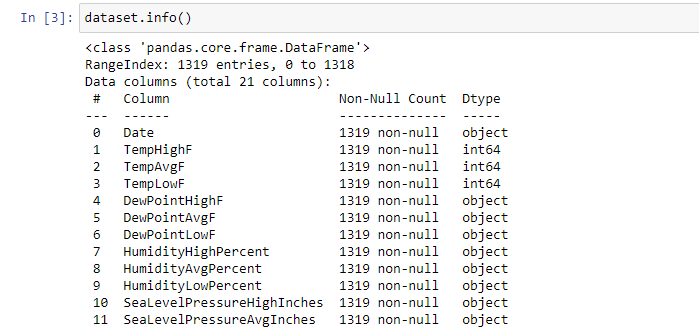
* Pandas
* Sklearn
* Matplotlib
* seaborn

# **Data cleaning**

We import our data into the jupyter notebook workspace using the panda’s library function read\_csv

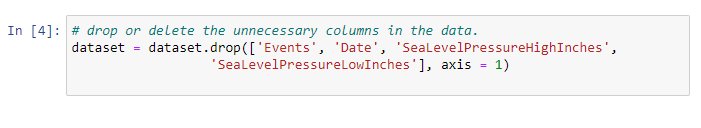


The we can look at the information contained in this dataset using the info() function

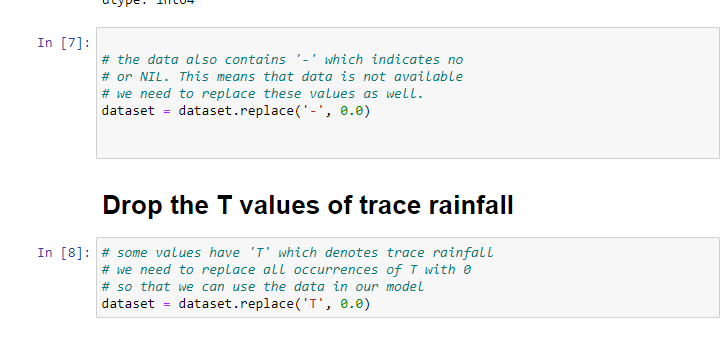


With this we will know what values we are working with.

The we clean our data by first dropping unnecessary columns which will not be used for this project.



Then we will replace the null values in the dataset since these values my alter the results of our prediction. In this case we have values like ‘- ‘and ‘T’ which are not integers therefore we need to drop them.

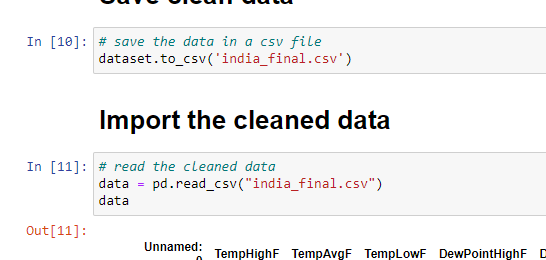


Then we will save our cleaned data in another csv file so that later we will import it and use the clean data.



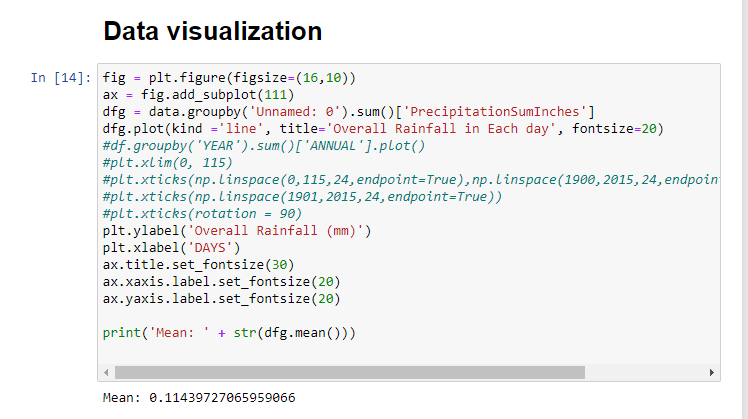
# **Data visualization**

We import the cleaned data using the pandas library like shown earlier.

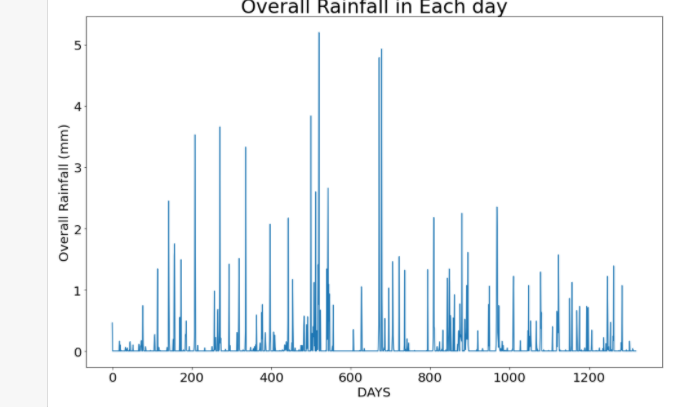


For visualization, we use matplotlib python library where we plot the overall rainfall against days of the year.

Then we will calculate its mean

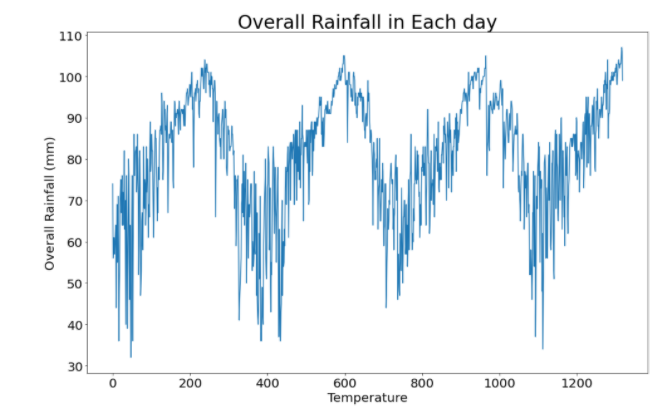


Output of this code is



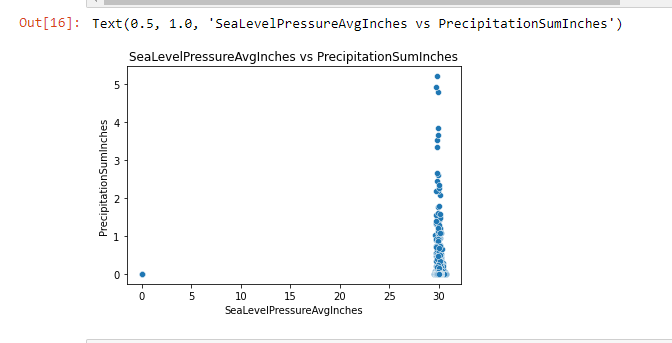
We can also plot a relationship between overall rainfall against temperature to understand the data well.



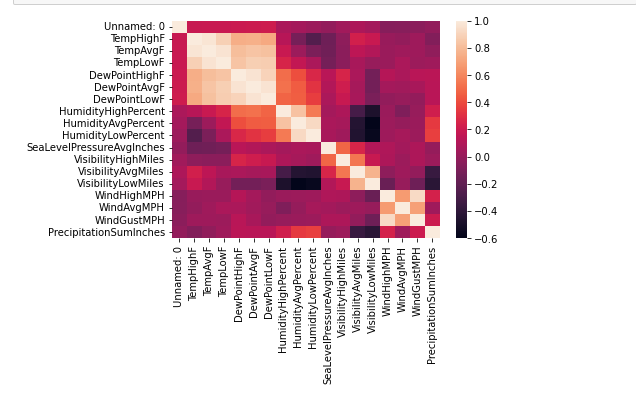


These tools are just for data visualization and others include scatter plots and heat map as implemented in the code.

**Scatter plot**



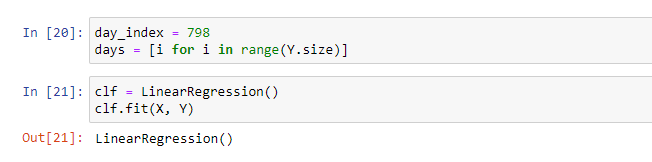
**Heat map**

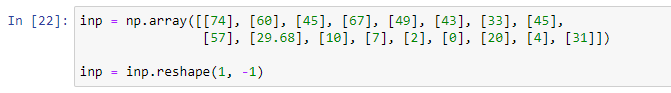


# **Prediction of rainfall based on the clean data**

We use linear regression for this.

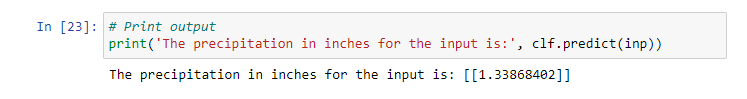
We introduce a random set of rainfall inputs that will be matched with the data set and predict whether it will rain or not in that particular region.



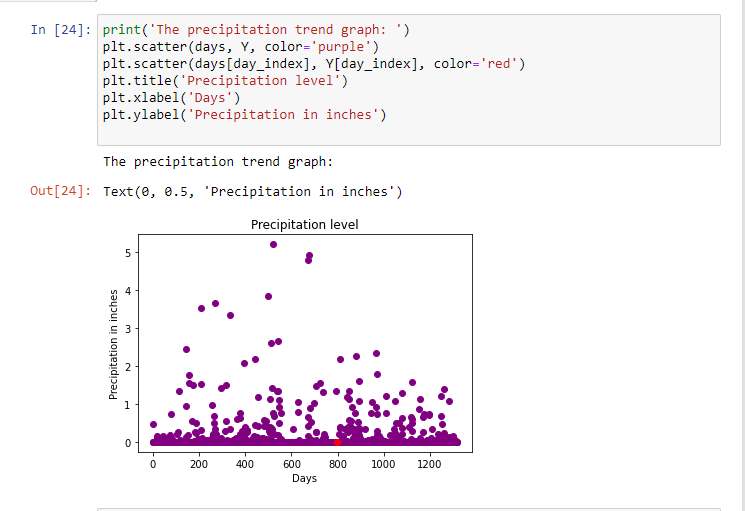


We reshape() shapes an array without changing the data of the array.

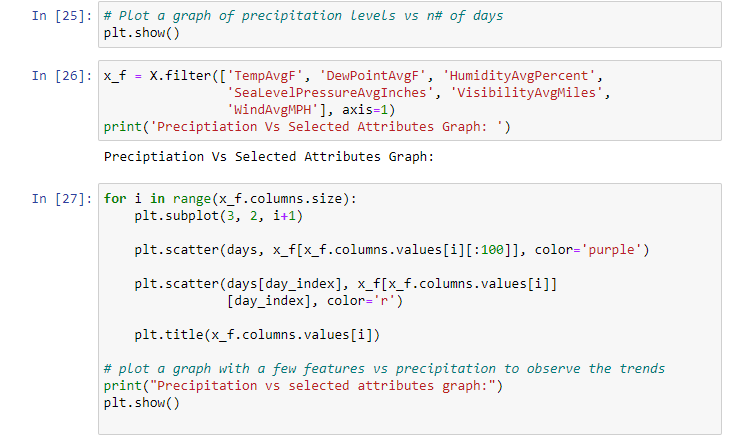
Then we can get the precipitation using the predict() function according to that rainfall input as

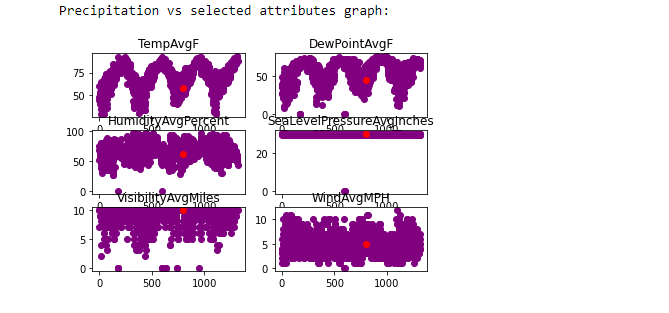


Then we plot a precipitation trend graph to visualize our prediction.



We can also plot the relationship between precipitation vs other values in the data such as temperature and dewpoint like this.



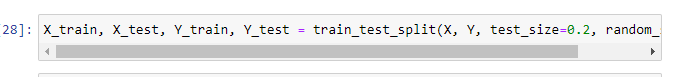


From the graph, a day (in red) having precipitation of about 2 inches is tracked across multiple parameters (the same day is tracker across multiple features such as temperature, pressure, etc.). The x-axis denotes the days and the y-axis denotes the magnitude of the feature such as temperature, pressure, etc. From the graph, it can be observed that rainfall can be expected to be high when the temperature is high and humidity is high.

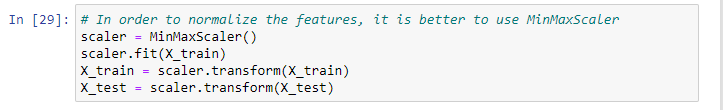
# **Splitting data into clusters and testing the model’s accuracy**

We use sklearn to train and test our data model which in this case is the linear regression model.

We split the data like this



Then we normalize the features of the data using minmaxScaler, a sklearn inbuilt function.



Then we test the accuracy of our model using the code below.

