*2.1.3 Historical Climate Data.*We used Weather API from World Weather Online Developer portal [67] to collect historical climate data of different districts of Bangladesh for each month over five year period from 2015 − 2019. Since we collected the twitter data till September 2019, the collected data also was till September 2019 (57 months). The API returns monthly climate data for each region in the form of a JSON object that contains the following  
climate information.  
• **Absolute maximum temperature:**Maximum temperature of the region in one month period measured in degree Celsius.

• **Absolute minimum temperature:**Minimum temperature of the region in one month period measured in degree Celsius.  
• **Average temperature:**Average monthly temperature of an area reported in degree Celsius.  
• **Sun hour:**This attribute measures the duration of sunshine within given period (a day/ week/ month/ year)  
for a particular region and it is reported in hour.  
• **UV Index:**It is a measurement of the strength of ultraviolet radiation at a particular place and time. The  
amount of exposure of an area to ultraviolet radiation is reported in standard UV Index unit which is equivalent  
to 25 milliWatts per square meter.  
• **Humidity:**Expressed as percentage, this value indicates the amount of water vapor present in the air relative  
to what the air can hold.  
• **Visibility:**Reported in miles, this value represents the distance at which an object or light can be clearly  
discerned.  
• **Pressure:**This is the atmospheric air pressure measured in millibars.  
• **Cloud Cover:**Average cloud cover, reported as percentage, refers to the fraction of the sky obscured by clouds  
when observed from a particular location.  
• **Heat index:**Reported in degree Celsius,it is a form of feels like temperature that is measured based on  
temperature and relative humidity.  
• **Dew point:**It is the atmospheric temperature, measured in degree Celsius, below which water droplets begin  
to condense and dew can form.  
• **Wind speed:**It is reported in miles per hour.  
• **Feels like temperature:**This is a measure of what the human body perceives the temperature to be based on  
original temperature, relative humidity, and wind speed. It is measured in degree Celsius.  
• **Rainfall:**Average all-period rainfall data in millimeters for a particular region.  
• **Rain Days:** Average rain days of all months since 2015 to 2019 september.

Now, using the API, we were able to collect climate data of five years for 53 districts in Bangladesh.

Among  
the reported values, only rainfall is reported as the average rainfall amount of all time. Since we were interested in year wise monthly climate data, two of our authors manually searched in World Weather Online platform to retrieve the number of rain days in a month and the amount of average rainfall for that month for each district in Bangladesh over our designated five year period.

After the collecting the data we started creating our dataset on which we decided to run our analysis further and predict, when there might an epidemic outbreak of dengue. Thus Labeling the whole dataset was important.

The tweets that we collected from twitter earlier we decided to use those as our labeling factor. We merged all the 2249 English tweets and 26437 Bengali tweets and labeled the locations as found in our text column. While labeling the locations we tokenized all the texts and removed stopwords, punctuations and external links. For such kind of semantic analysis we used NLTK[1] , SPACY[2] and RE[3] libraries of python and lemmatized the data in such a way that it is suitable for the continuation of our analysis.

At first we merged the years and months such that the new feature looks like *year-month*. Then we grouped the whole dataset by *year-month* feature. Now from the *location* and *year-month* columns from the twitter dataset we mapped the labels as following:

*If twitter (year-month) = weather(year-month) AND twitter(location) = weather(location) :*

*label = 1*

*Else:*

*label = 0*

This indicates if there is any post regarding dengue affected/infected/dead/awareness in the twitter dataset and if so, we labelled the corresponding month to be 1 and else to be 0.

Thus we labelled the whole dataset and the final dataset looked like the following:

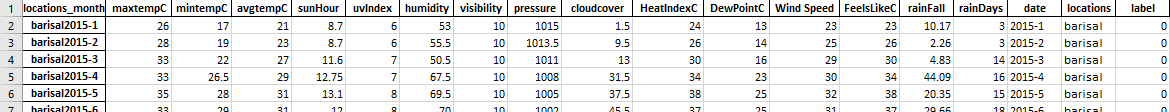
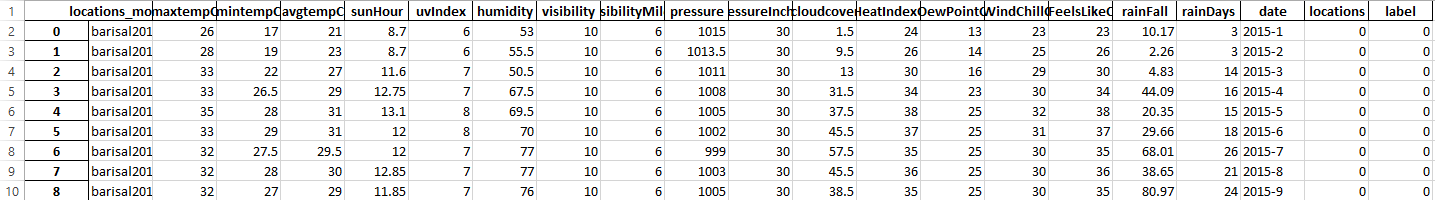


Fig: Dengue Dataset

Then we numbered the locations so that we could analyze the values and find out correlations among different factors. The locations became numeric values like Barisal district = 0, Bhola district= 1 and thus for all the locations.



The next phase was to calculate correlations between different factors and locations.

**=======================**

**Correlation**

**========================**

Then we divided our data into feature set X , often referred to as set of independent variables, and labels which is the class we will be trying to predict.

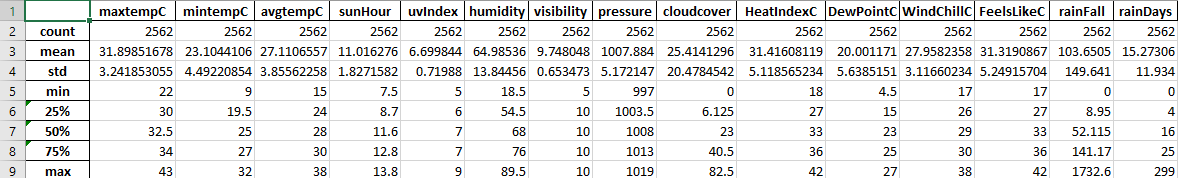
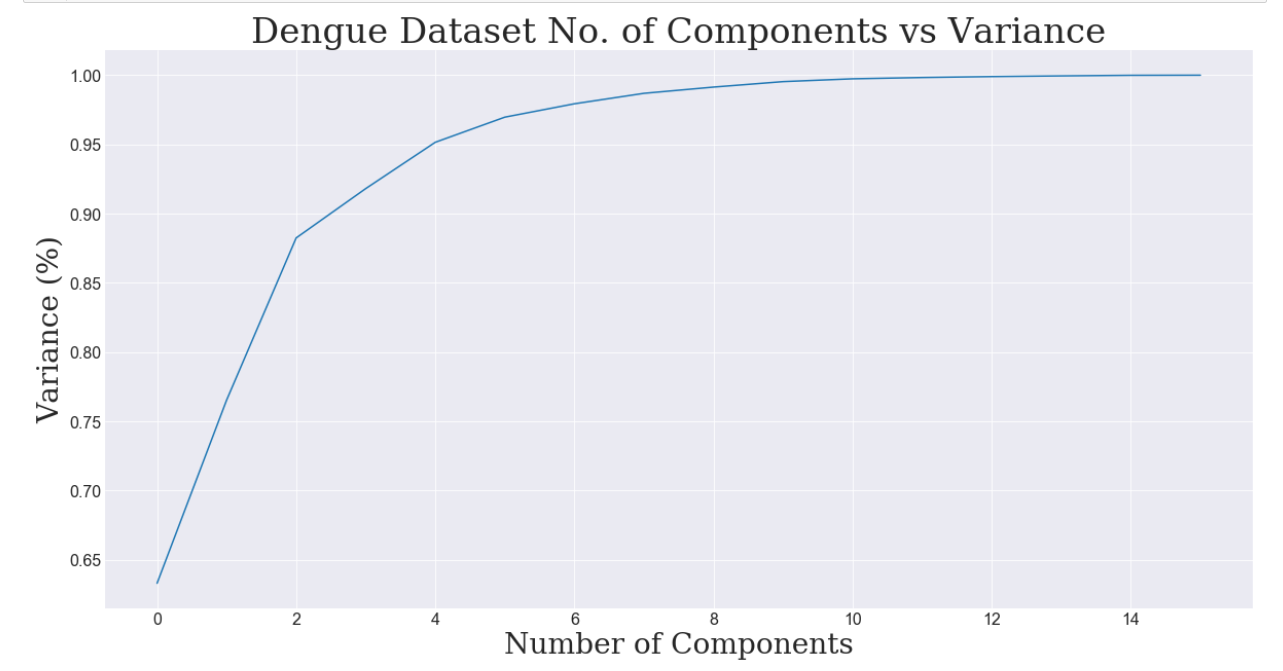


Fig : Statistical Description of Dengue Dataset

The shape of our dataset is now 2562, 16 and seems to have too many features. To reduce the dimensionality and find out the optimal dimension for our model, we used PCA[4] (Principle Components Analysis) to find out the number of factors/components that best represents out data with a minimal configuration. PCA mainly has two functions [5]. Firstly, it is to reduce dimension of the data and secondly, it is to find features that are important[6]. The reduced number of dimension are often referred to as principal components[7]. Another common application of PCA is to prevent data from overfitting[8] and reduce validation error. If there remains a high correlation amongst all the features PCA might be a good way for reducing dimension and lower the overfitting problem. As there no fixed rule on choosing the optimal number of components in PCA, we used 99% variance[9] standard to find out optimal number of factors. As we decrease the number of factors the following pattern is visible.

For running PCA we used *sklearn.decomposition.PCA*[10] library from python.

Before fitting the data for dimensionality reduction through PCA one rule of thumb is to normalize the data[11]. We used sklearn.preprocessing.MinMaxScaler[12] library from Python and a range of [0,1] to scale the data on a scale of 0 to 1. Then we fit and transform the data using PCA and found the following graph.



In the X axis the graph shows the no. of components and on the Y-axis the graph is showing the variance scores.

The variance percentage for different count of components can be seen in the following table:

|  |  |
| --- | --- |
| Number of Factors | Variance |
| 1 | 0.63302988 |
| 2 | 0.76535782 |
| 3 | 0.882474 |
| 4 | 0.91810669 |
| 5 | 0.95155339 |
| 6 | 0.96969941 |
| 7 | 0.97935335 |
| 8 | 0.98691043 |
| 9 | 0.99141932 |
| 10 | 0.99535534 |
| 11 | 0.9973854 |
| 12 | 0.99831264 |
| 13 | 0.99899793 |
| 14 | 0.99949722 |
| 15 | 0.99992221 |
| 16 | 1.0 |

In the table we can see that considering 8 best principle components gives us a variance of 98.69% and choosing 9 would give us 99.14% variance. Therefore, we proceeded by considering 8 components in our model which tends to give 99% of variance.

Now the shape of our data remains (2562, 8) which seems quite good enough and very convincing in terms of integrity.

But if we look at the number of classes we can something very unusual.

|  |  |  |
| --- | --- | --- |
| labels | count | percentage |
| 0 | 2365 | 92.3106948 |
| 1 | 197 | 7.6893052 |

It seems there are way too much 0 class(92.31%) than the number of class 1(7.69%) .This clearly indicates the data is not evenly distributed.

=========================

This might seem that it might be considered as an anomaly detection problem where class 1’s might be considered as anomaly. The main objective of anomaly detection algorithm is to train a parameter ꜫ[13] such that , if the validation set crosses that ꜫ mark it might be considered as an anomaly otherwise not.

==========================

Then we came across the process of classifying the data using supervised learning[14]. The main purpose of supervised learning is that, it runs a cost function[15] according to the algorithm given to it from the given training dataset and learns a weight matrix. Thus this weight matrix is multiplied across a cross validation[16] and/or validation[17] set and the final output is evaluated according to the specific algorithm. While evaluating a model we are mainly concerned about four evaluation criterions :

F1-Score :

1. <http://www.stat.columbia.edu/~fwood/Teaching/w4315/Fall2009/pca.pdf>
2. <https://www.researchgate.net/publication/280794624_Comparison_of_Five_Rules_of_Determining_the_Number_of_Components_to_Retain>
3. <https://biologydirect.biomedcentral.com/articles/10.1186/1745-6150-2-2>