IC272 - Data Science III

Assignment 1 : Data Preprocessing and Visualisation

Aahan Rupal
B22183

CONTENTS:

Introduction to the problem statement

We are given a csv ” containing the landslide-related readings from various sensors installed at 10 locations around the Mandi district. These sensors give details about the factors like temperature, humidity, pressure etc of a location.
We are provided with an another csv that is a version of the above file containing some missing values.

Q1.
In **part (i)** of the question we had to compute the mean, minimum, maximum, median, and standard deviation of the first attribute (“temperature”) without using any built-in statistical function. So I used the standard formulas of every attribute and got the respective answers.

RESULTS: The statistical measures of Temperature attribute are:
Min = 7.67
Max = 31.38
Mean = 21.21
Std = 4.35
Median = 22.27

In **part (ii)** of the question we had to compute the Pearson correlation between all the attributes and print the correlation matrix in a tabular format with column names as the header and column names on the side of the table. We also had to find the redundant attribute (correlation value >0.6) with respect to ‘lightavg’.
So we defined a correlation function and used the formula given in the question. We used the correlation function to create a matrix containing the respective pearson correlation values and converted the matrix to a data frame.
The redundant attribute came out to be ‘Lightmax’ as the correlation value was equal to 0.62.
We infer that ‘Lightmax’ and ‘Lightavg’ attributes are highly positively correlated.

In the **last part** of the question we had to build a histogram of humidity for ‘Stationid=t12’. To get the t12 stationid we used the groupby().getgroup() function. Then

using pyplot from matplotlib, constructed the histogram of bin size 5. We inferred that the majority of the places have high humidity and it gradually increases from 0 to 100.

Q2.

In **part (i)** of the question we had to Drop the tuples (rows) having missing values in the target attribute ("stationid"). For this we used

```
df= df.dropna(subset='stationid')
```

Also we had to Delete(drop) the tuples (rows) having equal to or more than one-third of attributes with missing values (use in-built functions of Pandas).
For this we used

```
df= df.drop(df[df.isna().sum(axis=1)>2].index)
```

In **part (ii)** of the question we had to fill up the missing values in the attributes (independent variables) using the linear interpolation technique. We used the pd.isna() and df.loc() functions to identify the missing values and used the interpolation as

```
df.loc[j,i]=(((y2-y1)*((x-x1)/(x2-x1))))+y1
```

To fill up the missing values

We also had to find the mean, median, and standard deviation for each attribute and compare the same with that of the original file. We used the standard formulas and got the answers.

The statistical measures of temperature attribute are:
Median(org) = 22.27, Median(After IP)=22.16
Mean(org) = 21.21, Mean(After IP)=21.12
Std(org) = 4.35 , Std(After IP)=4.4

The statistical measures of humidity attribute are:
Median(org) = 91.43 Median(After IP)=91.21
Mean(org) = 83.48 Mean(After IP)=83.17
Std(org) = 18.2 Std(After IP)=18.4

The statistical measures of pressure attribute are:
Median(org) = 1014.68 Median(After IP)=1014.94
Mean(org) = 1009.01 Mean(After IP)=1010.06
Std(org) = 46.96 Std(After IP)=46.06

The statistical measures of rain attribute are:

Median(org)=19.12 Median(After IP)=15.75
Mean(org)=10701.54 Mean(After IP)=10727.96
Std(org)=24839.1 Std(After IP)=24834.77

The statistical measures of lightavg attribute are:
Median(org)=1659.07 Median(After IP)=1501.72
Mean(org)=4438.43 Mean(After IP)=4496.36
Std(org)=7569.15 Std(After IP)=7644.99

The statistical measures of lightmax attribute are:
Median(org)=6717.5 Median(After IP)=6569.0
Mean(org)=21788.62 Mean(After IP)=21473.8
Std(org)=22053.32 Std(After IP)=21933.84

The statistical measures of moisture attribute are:
Median(org)=16.76 Median(After IP)=13.91
Mean(org)=32.39 Mean(After IP)=32.52
Std(org)=33.64 Std(After IP)=33.76

We also had to find the **RMSE** values of each attribute in both the dataframes. We used the formula given in the question.
RMSE of each Attribute: [1.4868984646879404, 6.323695638280401, 2.7700459823629484, 233.41558632083678, 7307.232608509665, 0.0, 19.644433461314648]

From the graph below we can clearly see that the error in lightavg attribute is the largest. This is due to the fact that there was considerable error in the values extracted after interpolation. It also indicates that it may have considerable amount of missing values.

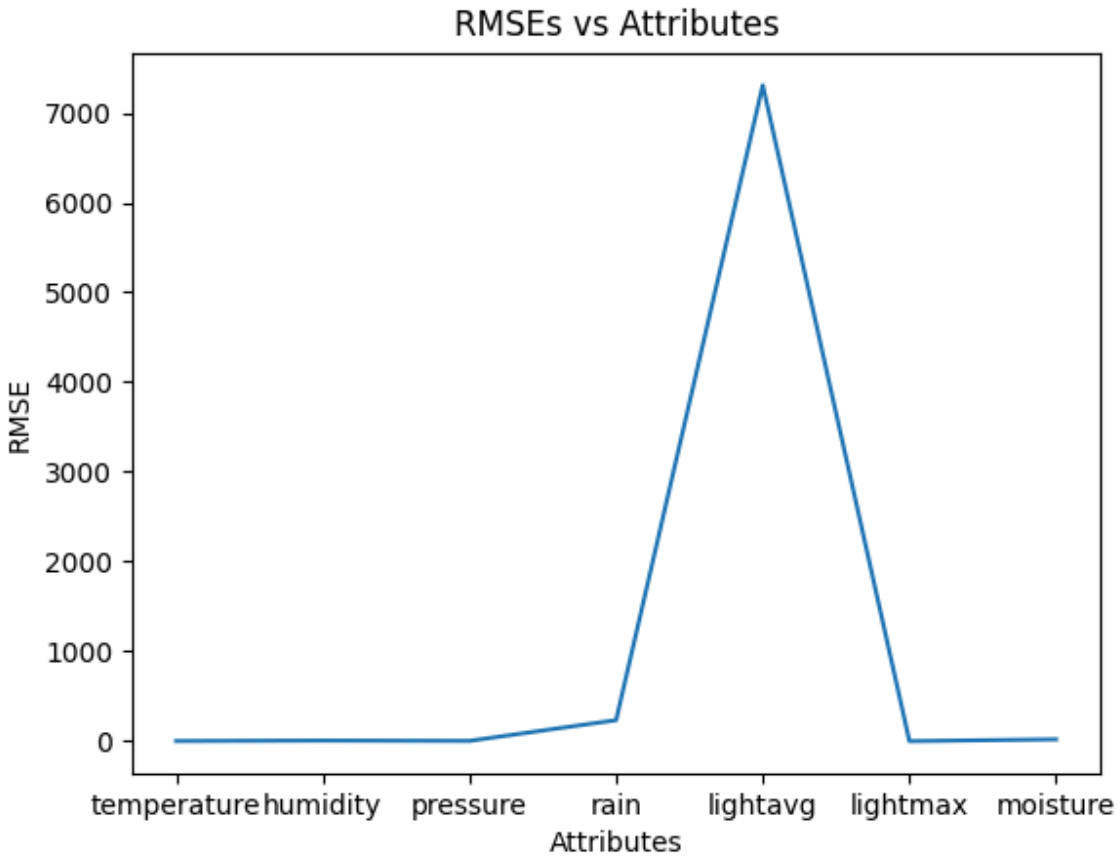

Figure-1: RMSE for each Attribute

Q3 In the question we have to detect the outliers from the data obtained after using the linear interpolation technique done in PartII - Q2. We detected the outliers by using the formula of IQR and plotted the boxplot using the plt.boxplot() command matplotlib. We saw the outliers on the boxplot as circular dots above the max point and below the min point.

In the ii part of the question we had to replace the outliers with the median of each attribute. And after the replacement we see that the number of outliers have decreased along with the number of circular dots. The outliers still remain as with the replacement there is new IQR and new conditions to be an outlier. So maybe some of the values close to maximum and minimum points can become an outlier.

Q4.

Here we had to normalize the outlier corrected dataset by implementing min-max normalization in range 5 to 12, and compare the min and max values for the original and the normalized dataset.

I have used the built in min and max function to find the minimum and maximum values for each attribute of the dataset.

The following are the minimum and maximum values of each attribute before and after the normalization:

```
Normalized:
temperature Minimum=5.0, Maximum=12.0
humidity Minimum=5.0, Maximum=12.0
pressure Minimum=5.0, Maximum=12.0
rain Minimum=5.0, Maximum=12.0
lightavg Minimum=5.0, Maximum=12.0
lightmax Minimum=5.0, Maximum=12.0
moisture Minimum=5.0, Maximum=12.0

original:
temperature Minimum=10.09, Maximum=31.38
humidity Minimum=34.21, Maximum=99.72
pressure Minimum=992.65, Maximum=1037.89
rain Minimum=0.0, Maximum=2470.5
lightavg Minimum=0.0, Maximum=10565.35
lightmax Minimum=2259.0, Maximum=54612.0
moisture Minimum=0.0, Maximum=100.0
```

We can observe that after the normalization process the min and max values changed to 5 and 12 for each attribute.

In **part (ii)** we had to standardize the dataset and compare the mean and standard deviation of each attribute before and after the standardization. To find the values of mean and standard deviation I used the inbuilt mean and standard deviation function of numpy

The output was as follows:

```
original:
temperature Mean=21.28, Std=4.16
humidity Mean=83.71, Std=17.73
pressure Mean=1014.92, Std=6.27
rain Mean=165.03, Std=396.11
lightavg Mean=2208.13, Std=2214.27
lightmax Mean=21473.8, Std=21933.84
moisture Mean=32.52, Std=33.76

Standardized:
temperature Mean=0.0, Std=1.0
humidity Mean=0.0, Std=1.0
pressure Mean=-0.0, Std=1.0
rain Mean=0.0, Std=1.0
lightavg Mean=0.0, Std=1.0
lightmax Mean=0.0, Std=1.0
moisture Mean=0.0, Std=1.0
```

We can observe that the mean changed to 0 and the standard deviation changed to 1 for each attribute of the standardized data.

List of Figures