Analysis and Forecasting of Precipitation ERA5 land data of India

Omkar Mehta (omehta2)

Milestone#3 (April 2022)

1 Problem Statement

I have chosen ERA5-land hourly data [1] for my project. The objective is to find useful information about the weather in the Konkan area of Maharashtra, which receives the highest rainfall, leading to the floods. If we are able to find useful patterns in the data that could help us forecast the precipitation in the future, even by an hour, it could help the people to better prepare for the flood. The data is in ***netCDF*** file format. Hence, we can use ***xarray*** to read the data.

We discussed how Savitri river got flooded every time there is more rainfall in the short period of time, repeated for three to four days. Let's subset the dataset to get rainfall in Mahabaleshwar, where Savitri river originates.

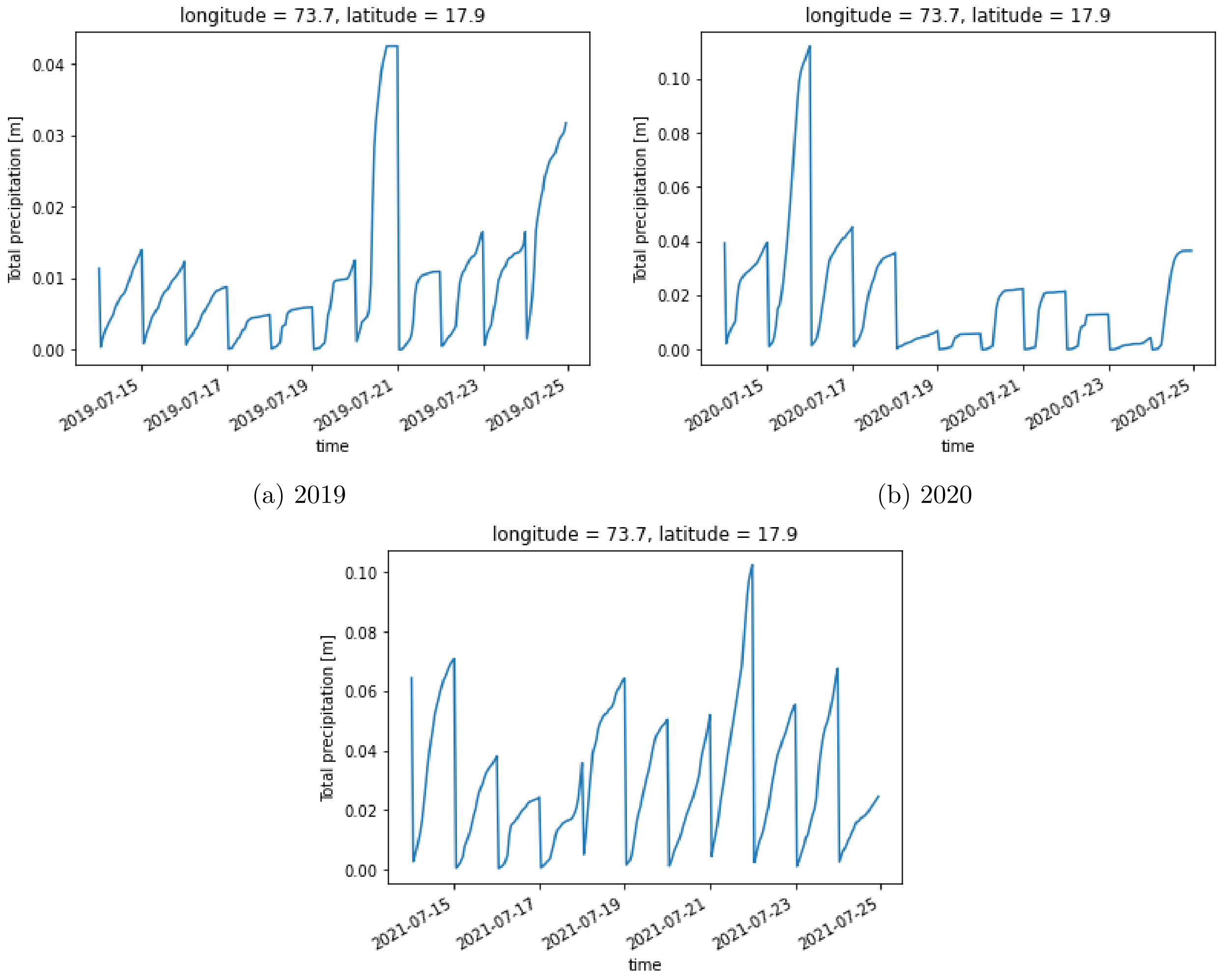
1.1 Rainfall in Mahabaleshwar [2]

- Mahabaleshwar is a hillstation in the Konkan district of Maharashtra, India. (**lat**: $17.9^o N$, **lon**: $73.65^o E$).
- On July 21, July 22 and July 23 2021, Mahabaleshwar had seen a record over 1,500 mm rainfall in the last three days.
- Mahabaleshwar is also the origin of four more rivers Koyna, Venna, Savitri and Gayatri.
- Other areas of Maharashtra also were flooded because of this.

The locations of Mahabaleshwar and Mahad are not far away. Even though ERA5 data is collected over 0.5 degree variation, the expected rainfall in Mahabaleshwar would be similar to Mahad. This can be seen in Figure 1. We can see that Mahabaleshwar too received the same amount of heavy rainfall consecutively for many days, which leads to overflow of the Savitri river.

I have not calculated the statistics of the rainfall in Mahabaleshwar, as I am sure that it would be similar to Mahad. I am also constrained by the fact that ERA5 data collects data over 0.5 degree by 0.5 degree.

As we can see from the Figure 1, in 2021, the rainfall was consistently more than what we could see in 2019 and 2020. If the rainfall is more, the river gets flooded pretty sooner. So, once we know that the rainfall has been consistently more for a few days, we can warn the people to start taking precautions. Also, the precipitation values in Mahabaleshwar and Mahad are similar. There is not much difference.



(a) 2019

(b) 2020

(c) 2021

Figure 1: Line Plot of Precipitation in Mahabaleshwar in 2019, 2020, 2021

2 The parameters that I have chosen for Linear Regression

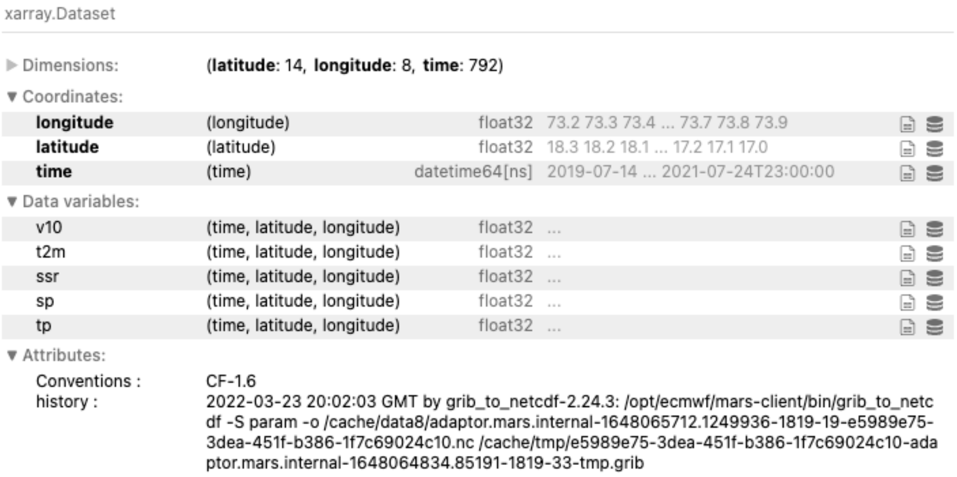
1. **10m_v_component_of_wind or v10** : northward component of the $10m$ wind, in m/s units, moving at a height of $10m$ above the surface.
2. **2m_temperature or t2m**: the temperature of air at $2m$ above the surface, in *Kelvin* units.
3. **surface_net_solar_radiation or ssr**: the amount of solar radiation that reaches a horizontal plane at the surface minus the amount reflected by the Earth's surface, in W/m^2 units.
4. **surface_pressure or sp**: pressure of the atmosphere on the surface of land, in Pa units.
5. **total_precipitation or tp**: the accumulated liquid and frozen water, comprising rain and snow, that falls to the surface in m units.

3 Subset data

Since there is a lot of data and many of the locations have null values (sea, ocean). I need to consider the region surrounding Mahad and Mahabaleshwar.

So, I have decided to consider the latitude from $17.4^o N$ to $18.33^o N$ and longitude from $73.24^o E$ to $73.88^o E$.

In the sub-data, we have 14 latitude points, 8 longitude points and 792 time points. This can be seen in the following Figure 3.



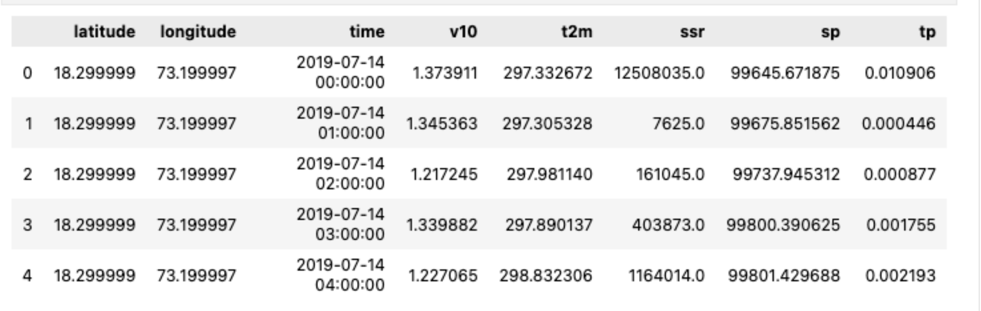
```
xarray.Dataset

▶ Dimensions:       (latitude: 14, longitude: 8, time: 792)
▼ Coordinates:
    longitude       (longitude)                         float32  73.2 73.3 73.4 ... 73.7 73.8 73.9
    latitude        (latitude)                          float32  18.3 18.2 18.1 ... 17.2 17.1 17.0
    time            (time)                      datetime64[ns]  2019-07-14 ... 2021-07-24T23:00:00
▼ Data variables:
    v10             (time, latitude, longitude)         float32  ...
    t2m             (time, latitude, longitude)         float32  ...
    ssr             (time, latitude, longitude)         float32  ...
    sp              (time, latitude, longitude)         float32  ...
    tp              (time, latitude, longitude)         float32  ...
▼ Attributes:
    Conventions :   CF-1.6
    history :       2022-03-23 20:02:03 GMT by grib_to_netcdf-2.24.3: /opt/ecmwf/mars-client/bin/grib_to_netc
                    df -S param -o /cache/data8/adaptor.mars.internal-1648065712.1249936-1819-19-e5989e75-
                    3dea-451f-b386-1f7c69024c10.nc /cache/tmp/e5989e75-3dea-451f-b386-1f7c69024c10-ada
                    ptor.mars.internal-1648064834.85191-1819-33-tmp.grib
```

Figure 2: Data subsetted for Modeling

4 Converting data to Pandas DataFrame

Since we will now move our attention to the modeling like Linear Regression, Neural Networks and more, I have converted ‘xarray‘ data to ‘pandas’s dataframe‘. After resetting index, the first five entries are shown in the following Figure **??**.



	latitude	longitude	time	v10	t2m	ssr	sp	tp
0	18.299999	73.199997	2019-07-14 00:00:00	1.373911	297.332672	12508035.0	99645.671875	0.010906
1	18.299999	73.199997	2019-07-14 01:00:00	1.345363	297.305328	7625.0	99675.851562	0.000446
2	18.299999	73.199997	2019-07-14 02:00:00	1.217245	297.981140	161045.0	99737.945312	0.000877
3	18.299999	73.199997	2019-07-14 03:00:00	1.339882	297.890137	403873.0	99800.390625	0.001755
4	18.299999	73.199997	2019-07-14 04:00:00	1.227065	298.832306	1164014.0	99801.429688	0.002193

Figure 3: First Five Entries of Pandas DataFrame

4.1 Handling Missing Values

There are some missing values in the dataframe. Since I have chosen Era5 land data, it doesn't collect data over water bodies. But, since we know from our analysis that the nearby regions receive similar amount of rainfall, we can handle missing values by interpolation's nearest neighbor method. I am using **ffill** method [3] to propagate last valid observation nearby to fill the missing value. There are a total of **4752** missing values, which were interpolated with the chosen method.

5 Linear Regression Model

5.1 Preparing data for LR Model

I have dropped the **time** variable from the data dataframe. I have also used sklearn's LabelEncoder [4] to transform **latitude and longitude** columns, making them categorical variables. This would be easier for Linear Regression model. We could drop these columns, if they are not statistically relevant to the model.

The dependent variable for the model is **tp**, and the rest of the columns in the data are independent variables (hopefully, independent).

5.2 OLS Linear Regression

I have used statsmodels's OLS method (Ordinary Least Squares) [5] for Linear Regression. The results are shown in the following Figure 4.

5.3 Observations

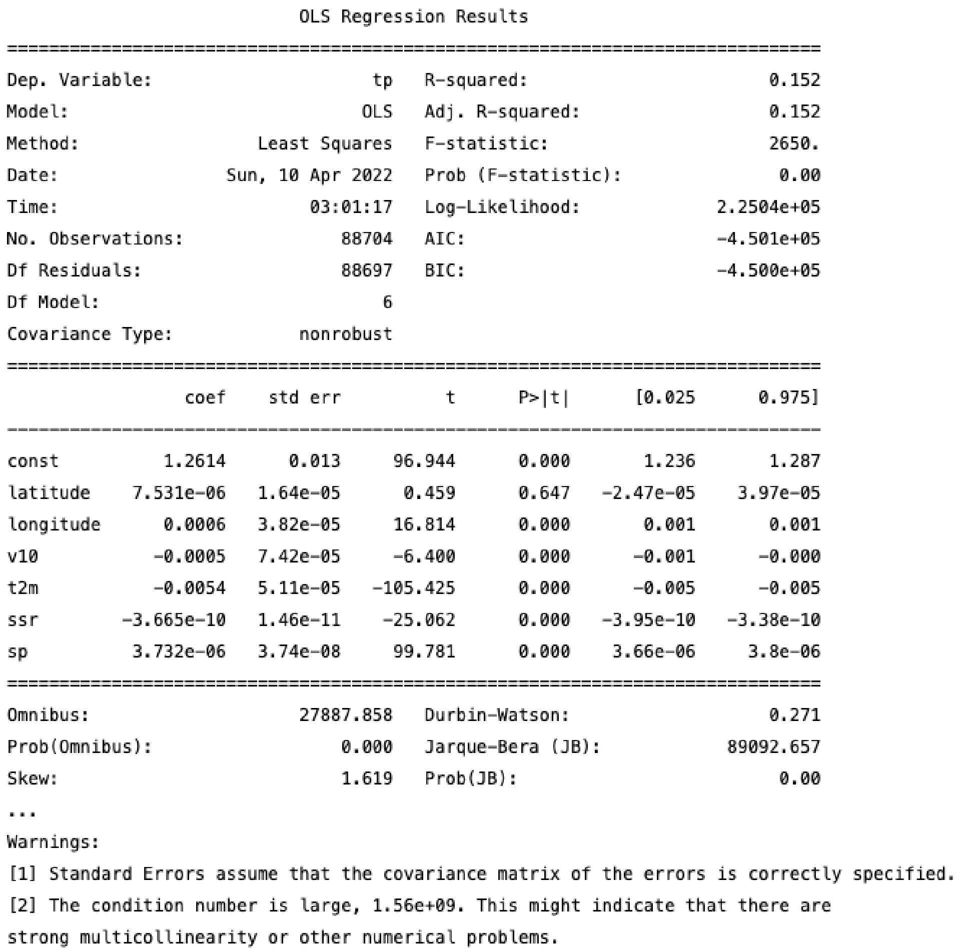
- R^2, a percentage of variation of dependent variable that is described the model. The value is **0.152**. It means that only 15.2% of the variance in the dependent variable is explained by the independent variables. This is very low. We won't get precise predictions at inference.

- The p-values for the t-test are 0 for all variables, except **latitude**. This indicates that there is sufficient evidence to reject the null hypothesis that the independent vairables has no relationship with the **tp**. The coefficients and the relationship they represent are statistically significant.

- There is a strong multi-collinearity, as shown in the warnings. This is obvious, since the data has timestamps. The data is sequential data. We need to use recurrent networks like RNN or LSTM for predictions.

6 Challenges/Concerns/Future Work

The data was not normalised. But, since we know that simple models like Linear Regression won't be too powerful for the prediction of the future values due to multicollinearity, we will be resorting to the recurrent networks like RNN or LSTM. I am expecting that LSTM models would be better than RNN, since it stores the latent space of the past values.



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                     tp   R-squared:                       0.152
Model:                            OLS   Adj. R-squared:                  0.152
Method:                 Least Squares   F-statistic:                     2650.
Date:                Sun, 10 Apr 2022   Prob (F-statistic):               0.00
Time:                        03:01:17   Log-Likelihood:             2.2504e+05
No. Observations:               88704   AIC:                        -4.501e+05
Df Residuals:                   88697   BIC:                        -4.500e+05
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          1.2614      0.013     96.944      0.000       1.236       1.287
latitude    7.531e-06   1.64e-05      0.459      0.647   -2.47e-05    3.97e-05
longitude      0.0006   3.82e-05     16.814      0.000       0.001       0.001
v10           -0.0005   7.42e-05     -6.400      0.000      -0.001      -0.000
t2m           -0.0054   5.11e-05   -105.425      0.000      -0.005      -0.005
ssr        -3.665e-10   1.46e-11    -25.062      0.000   -3.95e-10   -3.38e-10
sp          3.732e-06   3.74e-08     99.781      0.000    3.66e-06     3.8e-06
==============================================================================
Omnibus:                    27887.858   Durbin-Watson:                   0.271
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            89092.657
Skew:                           1.619   Prob(JB):                         0.00
...
Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.56e+09. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Figure 4: OLS Results

References

[1] “Reanalysis era5 land hourly data,” https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-land?tab=form.

[2] “Mahabaleshwar receives record over 1,500-mm rainfall in 3 days,” https://www.deccanherald.com/national/west/mahabaleshwar-receives-record-over-1500-mm-rainfall-in-3-days-1011863.html.

[3] “pandas.series.fillna,” https://pandas.pydata.org/pandas-docs/version/0.23.3/generated/pandas.Series.fillna.html.

[4] “sklearn.preprocessing.labelencoder,” https://pandas.pydata.org/pandas-docs/version/0.23.3/generated/pandas.Series.fillna.html.

[5] “Linear regression,” https://www.statsmodels.org/devel/regression.html.