Flood and Drought Forecasting using ML

PROJECT REPORT

Submitted in fulfillment for the J-Component of Technical Answers for Real World Problems (TARP)-

ITE3999

B. Tech – Information Technology

By

17BIT0084 - RK Sneha

17BIT0157 - Pranav Kumar

17BIT0175 - Rajat Bothra

17BIT0224- Abhinav Hada

17BIT0252- Prithesh S

Under the guidance of

Dr. ASWANI KUMAR CHERUKURI

SITE

School of Information Technology and Engineering

Winter Semester 2019-20

METHODOLOGY:

- We have gathered a real-time dataset for chennai's weather conditions from **NASA's Global Modelling and Assimilation Office Online.**

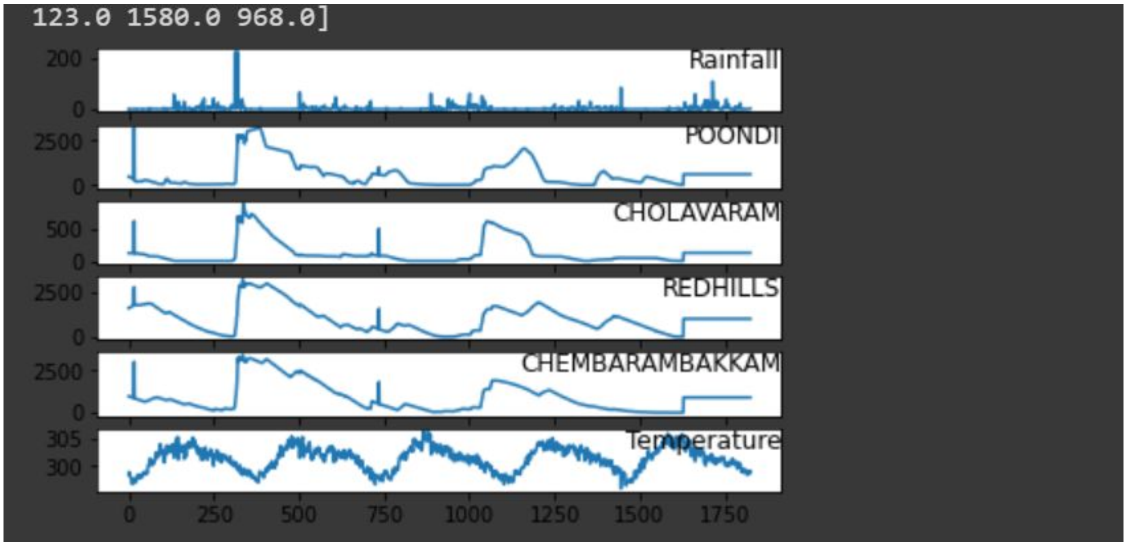
- Our dataset contains a total of **12 columns** and has about 4 years of daily conditions as data rows.

  It contains the following weather related fields- **Rainfall**, **Relative Humidity**, **Pressure**, **Wind Speed**, **Wind Direction**, **Snowfall**, **Snow-depth** and **Short wave irradiation** (Sunlight).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Date	Rainfall	Relative H	Pressure	Wind spee	Wind dire	Temperat	Snowfall	Snow dep	Short-wav	POONDI	CHOLAVAI	REDHILLS	CHEMBARAMBAKKAM
2	01-01-2015	0.251461	81.35	1006.4	2.11	177.31	298.33	0	0	5343.8440	487	123	1580	968
3	02-01-2015	1.928893	83.74	1009.09	3.5	148.75	298.83	0	0	5198.6390	481	123	1588	960
4	03-01-2015	0.826237	82.31	1009.81	2.56	104.63	298.64	0	0	5586.1750	476	123	1606	952
5	04-01-2015	0.231998	79.81	1010.02	2.92	49.43	298.53	0	0	5238.7940	470	123	1616	944
6	05-01-2015	0.107708	80.94	1009.59	3.34	35.46	298.17	0	0	5791.9620	460	123	1627	938
7	06-01-2015	0.147977	80.24	1010	3.22	29.6	297.88	0	0	5623.7990	449	123	1636	932
8	07-01-2015	0.034046	77.32	1009.56	2.88	42.92	297.88	0	0	5713.8100	438	123	1647	924
9	08-01-2015	0.097288	77.28	1010.47	2.02	61.38	297.79	0	0	5211.3670	427	123	1656	917
10	09-01-2015	0.078794	74.6	1011.57	3.4	39.02	297.2	0	0	5294.0260	415	123	1668	910
11	10-01-2015	0.013771	68.27	1011.88	4.5	28.26	296.66	0	0	5914.8890	402	123	1677	904
12	11-01-2015	0.000485	68.59	1011.66	4.78	38.62	296.78	0	0	5866.1800	389	123	1688	898
13	12-01-2015	0.002388	65.72	1011.31	4.72	40.33	296.81	0	0	6008.0660	374	123	1698	892
14	13-01-2015	0.003901	66.2	1010.29	4.15	33.13	296.82	0	0	5928.7430	360	122	1709	886
15	14-01-2015	0.020379	68.12	1010.38	3.57	43.56	296.7	0	0	5861.3990	347	122	1717	880

- **Poondi**, **Cholavaram**, **Red hills** and **Chembarambakkam** are the four water reservoirs in Chennai. The last 4 columns indicate **water levels** respectively**.**

- **Our target is to predict whether a place is likely to have flood or drought on the next day based on the rainfall it can expect for the next day.**



- As we can see here, there exists a **dependency** between **water levels in our reservoirs** and **rainfall measured** in the surrounding areas.

- The traditional way of predicting floods or droughts is understanding whether a place has had **more than 50% of rainfall** than the expected avg levels.

- To understand this better, we can see that **Chennai had it's last flood in November of 2015** with about **183% higher rainfall** than the place is expected to have in the month of November.

  *Chennai received 1,049 mm (41.3 in) of rainfall in November, the highest recorded since November 1918 when 1,088 mm (42.8 in) of rainfall was recorded. Registered the heaviest rainfall — **183% higher at 181.5 cm as against average rainfall of 64 cm in October–December period** and Tiruvallur district recorded 146 cm compared to average 59 cm of rain. The flooding in Chennai city was described as the worst in a century. The continued rains led to schools and colleges remaining closed across Puducherry and Chennai, Kancheepuram and Tiruvallur districts in Tamil Nadu, and fishermen were warned against sailing because of high waters and rough seas.*

  Source:https://en.wikipedia.org/wiki/2015_South_India_floods#cite_note-35

- So, it makes more sense to predict the rainfall we can expect for a place **tomorrow** based on it's conditions for a period of time and then cross checking it with how much rainfall the place was **expected to have** to understand whether the place is likely to have one of the following conditions: **Flood**, **Drought** or **Normal**.

- While it makes sense to have a **daily analysis for flood prediction**, as floods occur in a relatively short span of time, droughts do not occur with just no rainfall in one day. It is more of a monthly analysis. Where **there is little or no rainfall for an extended period of time**.

- Therefore it was more suitable to take **daily data** to predict the same.

- To predict the condition of one particular day, we need to know the **weather conditions of that place** in the past.

- The **LSTM model** is a great fit for this problem.

Why is LSTM optimal for our project?

- We are right now predicting the amount of rainfall that we can expect on a particular day based on its **past weather conditions**, and **water levels in the 4 reservoirs**.

- LSTM model is known for remembering **long term dependencies.** While we could assume that only our last day's weather conditions are just enough to predict today's rainfall or any other weather condition, **it doesn't make sense**. Because an event occurs with a series of changes for a particular time.

- LSTM can understand long term dependencies because LSTM has a **memory cell** and **reset cell** in its structure. Memory cell is known to remember it all while a reset cell is for resetting/clearing lstm's memory. After every batch of given size while training, an LSTM model's memory will be reset.

Workings of our model:

- In our model, we have given our batch size to be **50** which means that LSTM can understand long term dependencies for 50 days and after every 50 days, it will be reset.

- This offers great results to **predict rainfall at a particular day** for our place as it works as a sequential model remembering conditions and their dependencies for every 50 days.

- Firstly, we gathered the dataset and we have **converted it** into a time-series data.

- **Time-series data is continuous data** that enables a supervised learning for LSTM and has a basic structure like:

  Input: 0th day rainfall, pressure, ...water levels
  Output:1st day rainfall.

- That is, with today's and a sequence of conditions, our lstm has to **predict tomorrow's rainfall.**

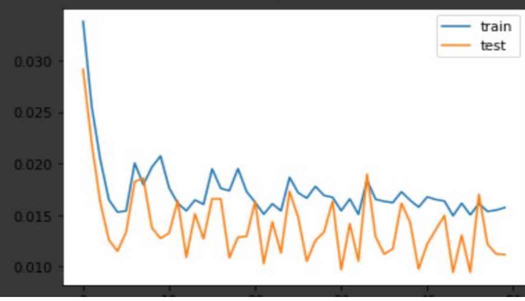
```
             Rainfall  Relative Humidity  ...  REDHILLS  CHEMBARAMBAKKAM
Date                                      ...
01-01-2015  0.251461              81.35  ...    1580.0            968.0
02-01-2015  1.928893              83.74  ...    1588.0            960.0
03-01-2015  0.826237              82.31  ...    1606.0            952.0
04-01-2015  0.231998              79.81  ...    1616.0            944.0
05-01-2015  0.107708              80.94  ...    1627.0            938.0

[5 rows x 11 columns]
Index(['var1(t-1)', 'var2(t-1)', 'var3(t-1)', 'var4(t-1)', 'var5(t-1)',
       'var6(t-1)', 'var7(t-1)', 'var8(t-1)', 'var9(t-1)', 'var10(t-1)',
       'var11(t-1)', 'var1(t)'],
      dtype='object')
   var1(t-1)  var2(t-1)  var3(t-1)  ...  var10(t-1)  var11(t-1)   var1(t)
1   0.001128   0.763252   0.558475  ...    0.499052    0.284831  0.008655
2   0.008655   0.828893   0.705467  ...    0.501579    0.282474  0.003707
3   0.003707   0.789618   0.744812  ...    0.507265    0.280118  0.001041
4   0.001041   0.720956   0.756287  ...    0.510423    0.277761  0.000483
5   0.000483   0.751991   0.732792  ...    0.513898    0.275994  0.000664

[5 rows x 12 columns]
```

  We can observe here that our **var1(t) is the rainfall for the next day** as output based on today's conditions.

- After converting our dataset into time-series data, we **train our model with 2 years of sequential data.** We have set our **shuffle=false** because we don't want to lose this sequence so that our model could learn inter dependencies better.

- We then **test our model** for the next 2 years of data.

```
Epoch 44/50
 - 0s - loss: 0.0149 - val_loss: 0.0095
Epoch 45/50
 - 0s - loss: 0.0161 - val_loss: 0.0130
Epoch 46/50
 - 0s - loss: 0.0150 - val_loss: 0.0095
Epoch 47/50
 - 0s - loss: 0.0161 - val_loss: 0.0170
Epoch 48/50
 - 0s - loss: 0.0154 - val_loss: 0.0122
Epoch 49/50
 - 0s - loss: 0.0155 - val_loss: 0.0112
Epoch 50/50
 - 0s - loss: 0.0157 - val_loss: 0.0112
```



- We defined our LSTM model to be trained over 2 years of data, i.e., 731 rows with a batch size of 50 with no. of epochs as 50. We chose our loss functions as **mean absolute error** (MAE) with **Adam optimiser.** MAE seemed to be perfect with a flexible optimiser, Adam that lets the model learn with understanding outliers importance too, as in our case, outliers could be a sudden spike in rainfall that could result in flood. We can observe from the graph and val_loss that our model has learnt.

When tested

```
  2.25337529e+00  5.98028779e-01  2.27674365e+00  2.83149219e+00
  1.73082387e+00  2.10487247e+00  2.93015742e+00  3.13838577e+00
  3.23072648e+00  2.79164577e+00  1.59653461e+00  2.01174784e+00
  3.37098694e+00  3.18348122e+00  3.14057064e+00  2.79873967e+00
  2.95293689e+00  2.96427584e+00  8.31035554e-01  1.33425367e+00
  1.51753116e+00  9.91503417e-01  1.98262882e+00  4.42557991e-01
  5.00095963e-01  2.24975729e+00]
(898,)
(898,)
Test RMSE: 6.137
```

We have an actual sequence of 898 days to be predicted and a predicted sequence with 898 days of next day's rainfall.

Our root mean square error is around 6.1 which is pretty acceptable for a sequential model.

- Later, we **cross checked our predicted results with the expected conditions** for a month. We are supposed to understand if this month could face any **drought conditions** or not. Whereas, the daily predicted sequence with expected conditions of a month to understand if this month has any **flood chance** on any day.

- **Actual sequence results:.**

```
0.49604303  normal condition
54] Date:  20-05-2019
0.411601  normal condition
Date:  21-05-2019
0.849777  normal condition
Date:  22-05-2019
0.592554  normal condition
Date:  23-05-2019
0.810335  normal condition
Date:  24-05-2019
1.386325  normal condition
Date:  25-05-2019
1.9490249  normal condition
Date:  26-05-2019
1.201545  normal condition
Date:  27-05-2019
1.350476  normal condition
Date:  28-05-2019
2.726293  normal condition
Date:  29-05-2019
3.7267342  normal condition
Date:  30-05-2019
5.687946  normal condition
Date:  31-05-2019
2.656858  normal condition
AVERAGE OF THIS MONTH  1.8099861173860488
Drought: yes
```

Our model's predicted sequence results:

```
[55] 0.98370713  normal condition
Date:  21-05-2019
1.4968262  normal condition
Date:  22-05-2019
2.410138  normal condition
Date:  23-05-2019
2.2533753  normal condition
Date:  24-05-2019
0.5980288  normal condition
Date:  25-05-2019
2.2767437  normal condition
Date:  26-05-2019
2.8314922  normal condition
Date:  27-05-2019
1.7308239  normal condition
Date:  28-05-2019
2.1048725  normal condition
Date:  29-05-2019
2.9301574  normal condition
Date:  30-05-2019
3.1383858  normal condition
Date:  31-05-2019
3.2307265  normal condition
AVERAGE OF THIS MONTH  1.881817136560717
Drought: yes
```

Conclusions from our results:

- If we remember, **Chennai had a very critical water crisis around May-June in 2019**.

- We have given conditions in our model to declare it as **flood prone** if a place had about **100% higher rainfall than the expected average conditions.**

- And to understand, if a place is likely to have **drought**, we take a month of predicted values and consider its average and **cross check it with the expected rainfall** for a place in that month, if actual average **hasn't even received 5% of expected average rainfall for that month**, we declare that this place is likely to face drought for this month.

- Our model, thus, has **predicted a flood condition only once,** and it is in mid March, 2018. While checking around the news, for March, 2018 weather conditions, we observed that Chennai had a severe rainfall. Although it predicted as flood, our **model seems to have learnt to understand weather conditions** better.

```
0  normal condition
Date:  14-03-2018
0  normal condition
Date:  15-03-2018
0  normal condition
Date:  16-03-2018
0  normal condition
Date:  17-03-2018
0  normal condition
Date:  18-03-2018
1.2504756  Flood : yes
Date:  19-03-2018
0  normal condition
Date:  20-03-2018
0  normal condition
Date:  21-03-2018
0  normal condition
Date:  22-03-2018
0  normal condition
Date:  23-03-2018
0  normal condition
Date:  24-03-2018
0.15871982  normal condition
Date:  25-03-2018
0  normal condition
Date:  26-03-2018
```

THIS STORY IS FROM MARCH 14, 2018

Rain lashes south Tamil Nadu, board exams to go ahead as scheduled

TIMESOFINDIA.COM | Updated: Mar 14, 2018, 08:26 IST

A- A+

Heavy rain is likely to occur at isolated places over south ..
Read more at:
http://timesofindia.indiatimes.com/articleshow/63293936.cms?utm_source=contentofinterest&utm_medium=text&utm_campaign=cppst