NTT Code for Good

Topic: Disaster Relief & Response

Kerala Rainfall Prediction

Ashish Singh (NTT Data)
Sumit Sahoo (NTT Data)
Kotha Tejas (IIIT Bangalore)
Tanmay Jain (IIIT Bangalore)

India is the land of agriculture and being a peninsular country, it is highly susceptible to heavy rainfalls in the south. States like Maharashtra, Karnataka, Kerala, Tamil Nadu, Andhra Pradesh, Odisha and West Bengal, being along the coastline of India have high humidity and moderate temperature all year long. Monsoons play an important role in the livelihood of majority of the population thus having a greater impact on life, society and economy.

As the water in the oceans takes longer to heat than land during day, and remains heated late till night, the temperature in these states remains moderate throughout the year. This differential heating and cooling of land and water also creates low pressure on the landmass of India while the seas around it experience comparatively high pressure, causing more humid air to flow into the Indian landmass. All these factors, including other facts like the El Nino, La Nina, North Atlantic Sea Surface Temperature, Equatorial SE Indian Ocean Sea Surface Temperature, East Asia Mean Sea Level Pressure, North Atlantic Mean Sea Level Pressure and North Central Pacific wind together influence the overall rainfall in India.

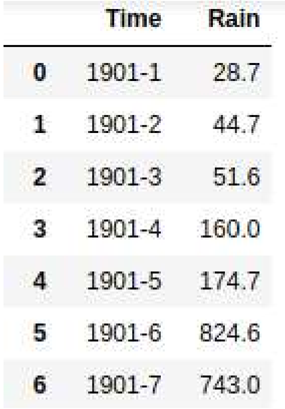
These factors generally cause cyclones and typhoons in the Indian peninsula, which eventually lead to excessive rainfall. Whenever this happens, there is a high chance of floods. The rivers, dams and other water bodies get overfilled, causing floods in the nearby areas.

One such catastrophe was the floods in Kerala in August 2018. It caused havoc in the entire state, there was a huge loss of state property and a lot of people died either due to drowning or due to lack of essential supplies like food, shelter, water, etc. People from all over the country came forward to help and life was finally restored in Kerala, but the loss was still huge.

Disasters like these have less chances of causing this intense damage if we are warned early on. Such situations call for a safety measure that can warn us before the disaster even happens. Hence, we came up with a solution for predicting rainfall in states like Kerala for any given month based on the data of past 115 years of rainfall across India (*Source: https://data.gov.in/resources/sub-divisional-monthly-rainfall-1901-2017*).

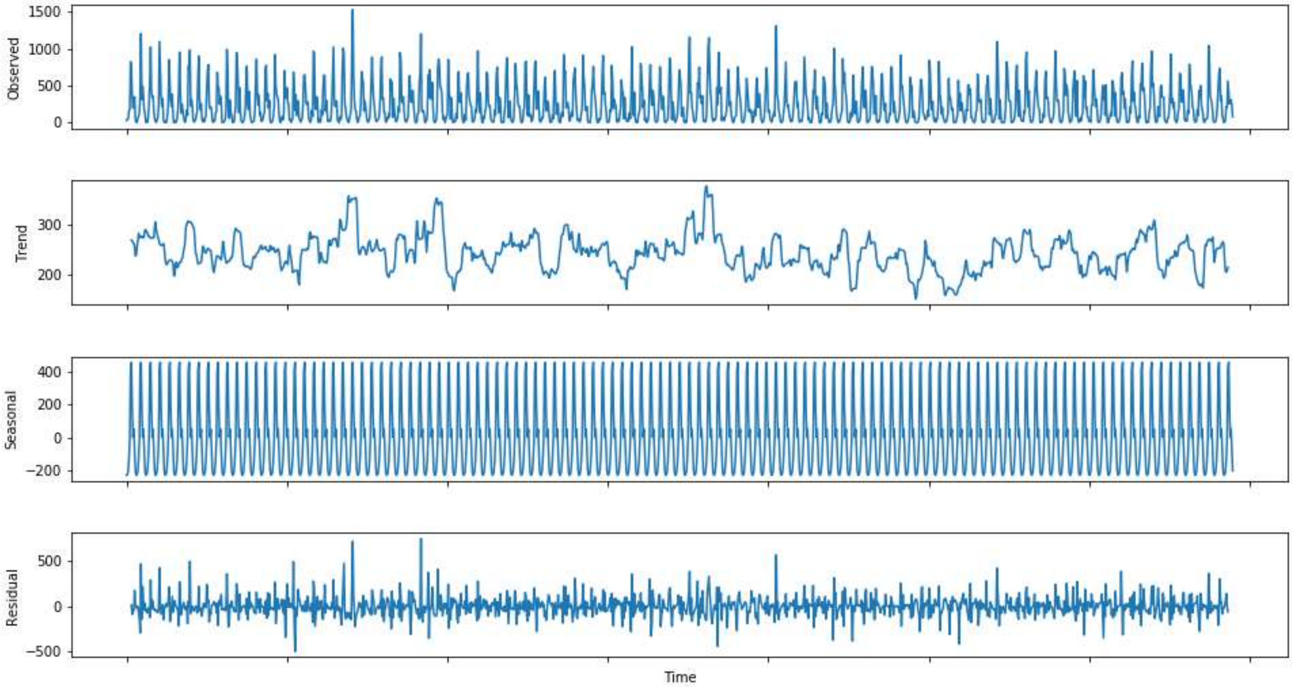
Solution Approach and Architecture

Weather patterns tend to repeat annually because of the rotation of Earth. We exploit this simple fact to create a time-series data analysis of the monthly rainfall for the 115 years. Our dataset has monthly rainfall data for all the states year-wise. After retrieving all of Kerala’s data and doing transformations our data looks like this:



	Time	Rain
0	1901-1	28.7
1	1901-2	44.7
2	1901-3	51.6
3	1901-4	160.0
4	1901-5	174.7
5	1901-6	824.6
6	1901-7	743.0

We did a small analysis to see the trend and seasonality of the data



It’s a very interesting observation that the seasonality of the rainfall pattern is almost repetitive which is a really good start in terms of time series analysis.

We used a statistical algorithm called SARIMA (Seasonal Autoregressive Integrated Moving Average) for this purpose. Seasonal ARIMA is an extension of ARIMA that explicitly supports

univariate time series data with a seasonal component. It adds three new hyperparameters as well as an additional parameter to the vanilla ARIMA algorithm.
They are as follows:

- **S**: *Seasonal*. A model that supports univariate time series data with a seasonal component.
- **AR**: *Autoregression*. A model that uses the dependent relationship between an observation and some number of lagged observations.
- **I**: *Integrated*. The use of differencing of raw observations (e.g. subtracting an observation from an observation at the previous time step) in order to make the time series stationary.
- **MA**: *Moving Average*. A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.

The parameters for ARIMA model are referred to as the order of the model and denoted by a 3-tuple (p,d,q), where

- p: Trend autoregression order.
- d: Trend difference order.
- q: Trend moving average order.

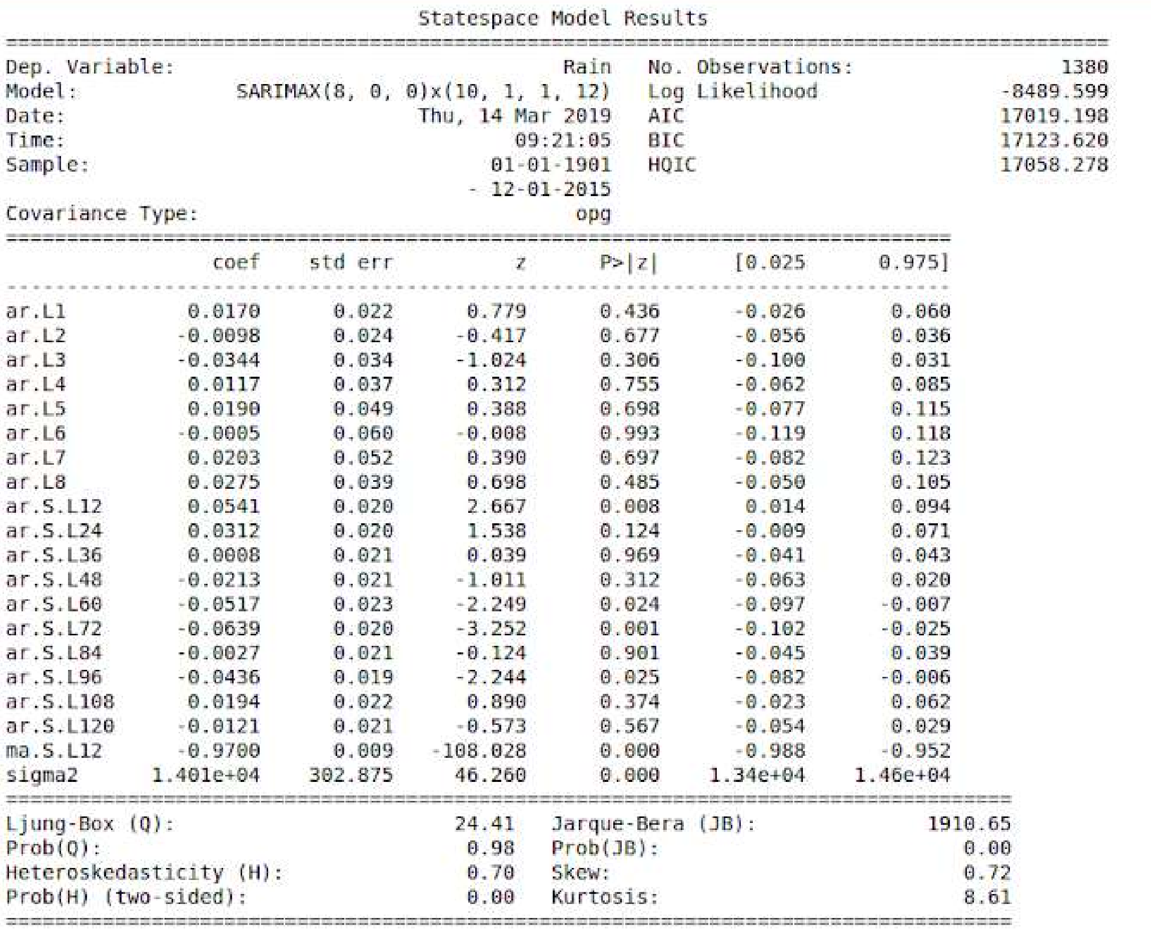
Similarly, in the SARIMA model, we use a set of 2 parameters, one the usual order like the ARIMA model, and another called the seasonal_order for the model. The seasonal_order is a 4-tuple parameter represented as (P,D,Q,S), where

- P: Seasonal autoregressive order.
- D: Seasonal difference order.
- Q: Seasonal moving average order.
- S: The number of time steps for a single seasonal period.

We performed a Grid-Search on the order and seasonal_order parameters for the model and tested the various values on their AIC (Akaike Information Criteria) to judge them. The best set of values were found to be as follows:

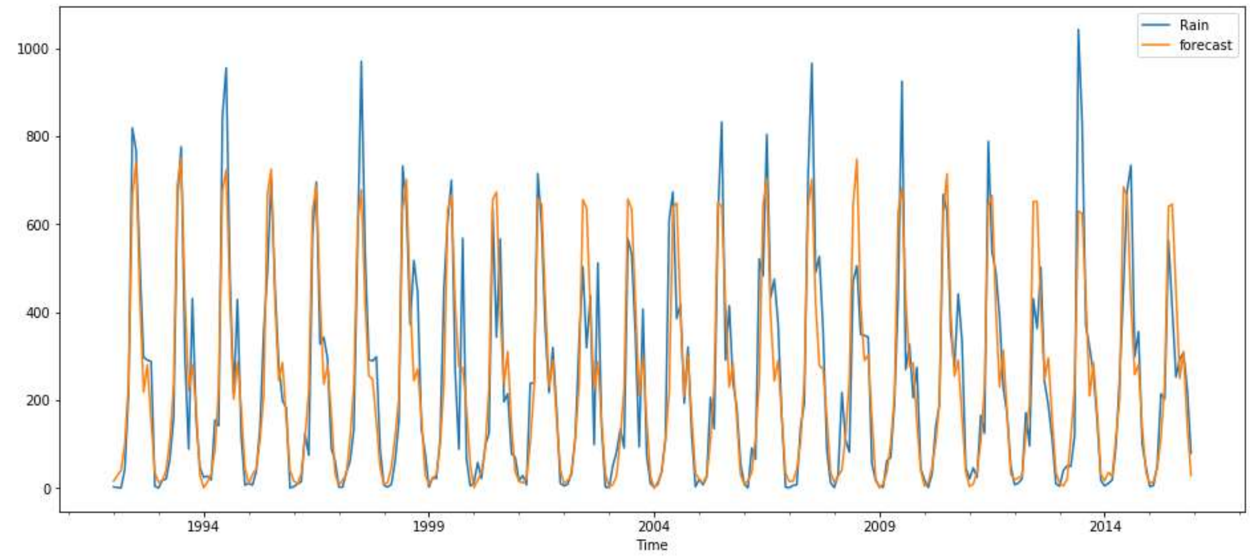
order : (8,0,0)
seasonal_order : (10,1,1,12)

Thus the model we used is - SARIMAX(8,0,0)x(10,1,1,12)



```
                                  Statespace Model Results
============================================================================================
Dep. Variable:                                 Rain   No. Observations:                 1380
Model:              SARIMAX(8, 0, 0)x(10, 1, 1, 12)   Log Likelihood               -8489.599
Date:                              Thu, 14 Mar 2019   AIC                          17019.198
Time:                                      09:21:05   BIC                          17123.620
Sample:                                  01-01-1901   HQIC                         17058.278
                                       - 12-01-2015
Covariance Type:                                opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.0170      0.022      0.779      0.436      -0.026       0.060
ar.L2         -0.0098      0.024     -0.417      0.677      -0.056       0.036
ar.L3         -0.0344      0.034     -1.024      0.306      -0.100       0.031
ar.L4          0.0117      0.037      0.312      0.755      -0.062       0.085
ar.L5          0.0190      0.049      0.388      0.698      -0.077       0.115
ar.L6         -0.0005      0.060     -0.008      0.993      -0.119       0.118
ar.L7          0.0203      0.052      0.390      0.697      -0.082       0.123
ar.L8          0.0275      0.039      0.698      0.485      -0.050       0.105
ar.S.L12       0.0541      0.020      2.667      0.008       0.014       0.094
ar.S.L24       0.0312      0.020      1.538      0.124      -0.009       0.071
ar.S.L36       0.0008      0.021      0.039      0.969      -0.041       0.043
ar.S.L48      -0.0213      0.021     -1.011      0.312      -0.063       0.020
ar.S.L60      -0.0517      0.023     -2.249      0.024      -0.097      -0.007
ar.S.L72      -0.0639      0.020     -3.252      0.001      -0.102      -0.025
ar.S.L84      -0.0027      0.021     -0.124      0.901      -0.045       0.039
ar.S.L96      -0.0436      0.019     -2.244      0.025      -0.082      -0.006
ar.S.L108      0.0194      0.022      0.890      0.374      -0.023       0.062
ar.S.L120     -0.0121      0.021     -0.573      0.567      -0.054       0.029
ma.S.L12      -0.9700      0.009   -108.028      0.000      -0.988      -0.952
sigma2      1.401e+04    302.875     46.260      0.000    1.34e+04    1.46e+04
====================================================================================
Ljung-Box (Q):                       24.41    Jarque-Bera (JB):              1910.65
Prob(Q):                              0.98    Prob(JB):                         0.00
Heteroskedasticity (H):               0.70    Skew:                             0.72
Prob(H) (two-sided):                  0.00    Kurtosis:                         8.61
====================================================================================
```

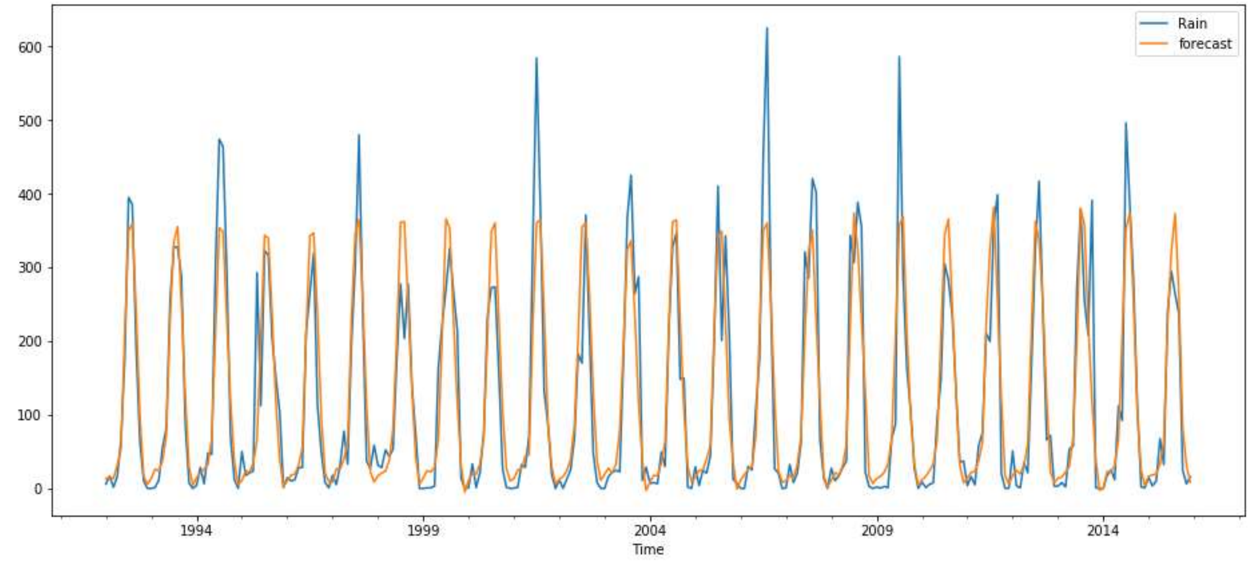
We tested our model on the data from Jan 1992 to Dec 2015, which was a total of 288 data points. We achieved an **RMSE of 105.275 mm** on the Kerala rainfall dataset. Here’s the graph that shows the relation between the actual and the predicted values:



In the graph, the blue line represents the actual rainfall in Kerala and the yellow line represents the predicted rainfall. It is clear that the model has almost accurately predicted the rainfall in all the cases. It however fails to predict the spikes in rainfall.

The current model works perfectly in cases of no flood situation but doesn't predict anomalies perfectly but with the help of wind and humidity patterns as well as temperature records we will be able to cover the gap in the performance of the model.

We used the dataset for Odisha as well and achieved a **RMSE of 59.641 mm.** Odisha is hit by cyclone almost every year hence, there are regular spikes in seasons of August. Here's the graph that shows the relation between the actual and the predicted values:



This graph also confirms that the model is indeed robust and accurately predicts rainfall in any state given a time-series dataset.

Technology/Tool/Cloud Stack

We used Python as programming language. We used the following packages in order to build our pipeline :
Matplotlib for plotting
Plotly for the demo plot
Pandas
NumPy
Jupyter Notebook
Statsmodel API for SARIMA algorithm

Hardware Specifications

We used an i7 6th generation laptop with a RAM of 8GB to perform our analysis. It takes around for 4 minutes to run the pipeline.

Demo - Video/Prototype
URL: https://frenzytejask98.github.io/

Importance:

The idea is to keep in track the rainfall patterns in the months of historically heavy rainfalls and be prepared for emergency evacuation plans. The Indian Meteorological Department can only warn about a cyclone just before 2 days of the event in which case this solution will help the Government agencies to prepare for emergency conditions.

The Kerala Floods in August 2018 claims lives of 483 people and around 14 went missing. The total loss in property and crops was estimated to be Rs 8000 crore. This huge loss could have been either eliminated or atleast reduced to a minimum possible count. This can be achieved by implementing the model proposed and predicting any upcoming floods and taking necessary precautions or preventive measures.

Challenges Faced

The dataset for Kerala was very skewed as the rainfall pattern in Kerala has been rather unstable. The dataset originally had only 115 data points which made it really hard for us to train a good model. We first converted the dataset into a 3-month dependency rainfall pattern, i.e., for any given month, we used the past 3 months of rainfall as data to predict the rainfall in the given month. This gave us a decent 1400(approx) data points to work with. But, as it is quite intuitive, the rainfall pattern in Kerala doesn’t follow this kind of dependency, thus we got a rather bad model with RMSE error around 200 mm using MLP (Multi-Layer Perceptron) Regressor.

We then built a time-series from the data and got 1380 data points. This gave a decent model as the rainfall pattern is seasonal and repeats almost similarly each year.

Preparing the dataset was a challenge for us as we had to juggle between solving the problem as a Regression problem or as a Time series problem. Both these approaches have different needs in terms of data so, figuring out on how to prepare the dataset was a challenge.