Predicting Monsoon Rainfall in South Asia

Aditya Thakur

December 10th 2016

I. Definition

Project Overview

Weather Prediction is a problem that is been discussed for centuries but there exists no such model which can predict a certain meteorological phenomena. Researchers have gotten very close to the part that considering enough features around a certain trend, a ‘close-enough’ prediction can be made about it. The weather is a vast concept and Monsoon prediction is just one tiny part of it. The monsoon is defined as a wind in the region of South and Southeast Asia, blowing from the southwest between May and September and bringing rain (the *wet monsoon*), or from the northeast between October and April (the *dry monsoon*) [1]. There have been a few models that predict the monsoon with more or less accurate results. There exists a model based which was launched recently is said to be able to predict monsoon intensity with 95% of accuracy and only 38 years of data.

Problem Definition

In this project it has been planned to implement a Monsoon Prediction model incorporating both the intensity and the onset of the monsoon rainfall for south Asian regions. The data that will be used to predict these results will be of 45 years from year 1960 to 2005. The Time series predictive model will be used with previous year’s weather conditions and some external factors and using the trends in the features a general solution will be obtained using a computational Intelligence approach.

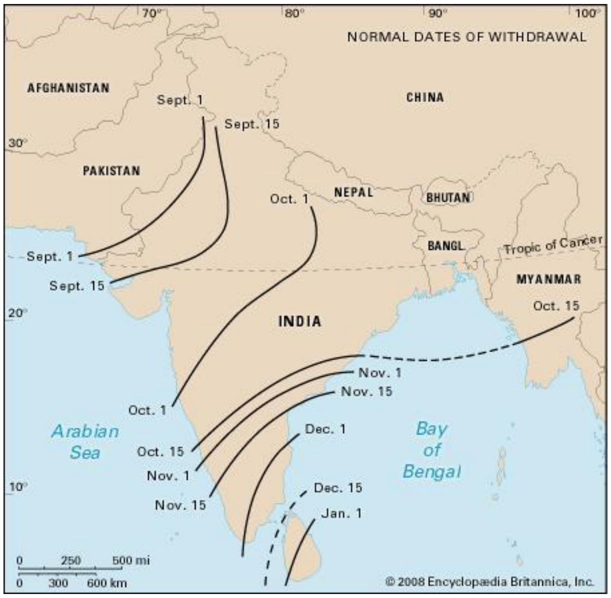

Figure 1: Monsoon Withdrawal Dates

Metrics

An R^2 Score Metric will be used to measure the accuracy of the results obtained by the model as a Regressor based approach is being used to implement the prediction model. The R^2 Score is the one minus Residual Sum of Squares divided by the total sum of squares [2].

II. Analysis

Data Exploration

The dataset that is being used as an input to this model is the data from year 1962 to 2005. The data contains all the atmospheric conditions including monthly rainfall and temperature at multiple locations in south Asia. Another feature that was added was average rainfall for the whole south Asia. A few more factors that were added as the features were El-Nino Southern Oscillation Effect [3] and the Indian Ocean Dipole Effect [4]. It has been researched that these effects are directly or indirectly affect the monsoon rainfall in the future years.

Data Visualization

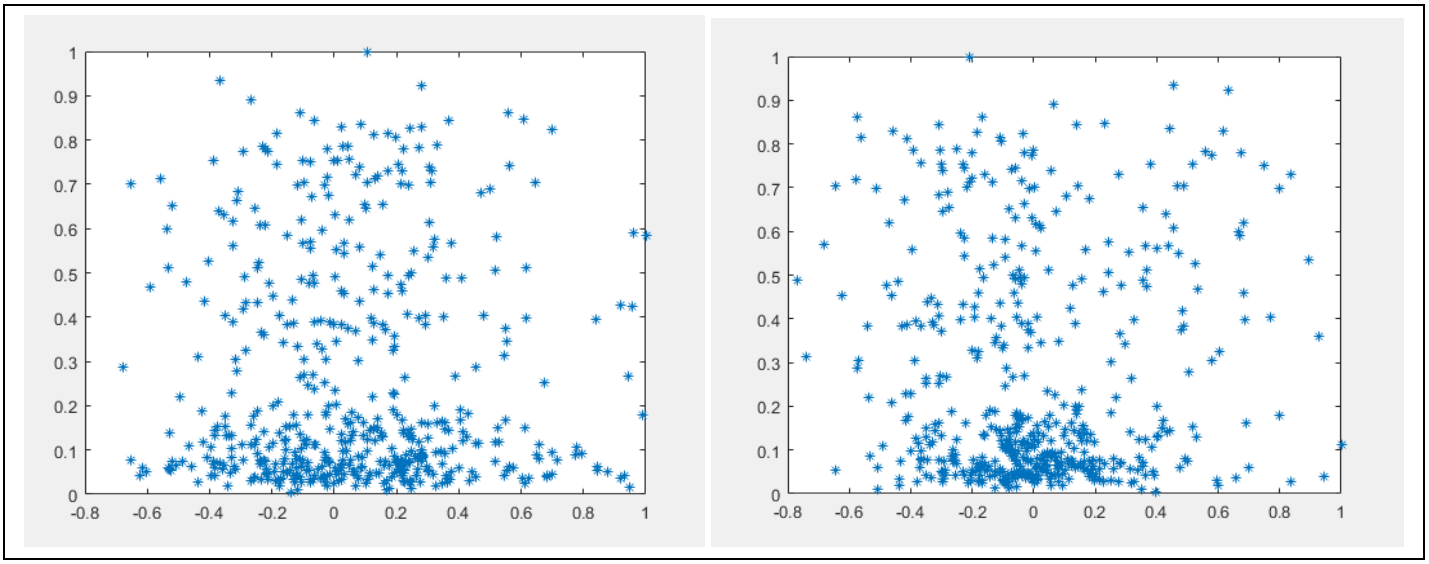

Figure 2: Correlation of Rainfall with the ENSO and Indian Dipole Index

From Figure 2, it can be seen as the Data of El Nino Southern Oscillation and Indian Ocean Dipole index are not directly correlated with the actual monsoon rainfall and we need a time series kind of input to find a relation between them.

Algorithms and Techniques

In problem statement we have decided to implement the model using a Multilayer Perceptron Neural Network Regressor. With this model we will also be testing the data with different Ensemble learning approaches, Decision Tree approach and the Gradient Descent approach. The initial assumption is that the model should work well with the neural nets as there are a lot of non-linearity with the data and neural networks are better suited to handle this sort of data without really overfitting it.

- Multi-Layer Perceptron :
  - A Multilayer perceptron is the Network of neural networks cascaded in a form where the weights are turned by the means of backpropagation algorithm. The advantage of using a neural net based algorithm is that it can model for the non-linearity in the data. The drawback is it takes a lot of time to train.
- Random Forest
  - Random Forest is an ensemble learning approach where a set of decision trees are taken together to form a forest like structure. This is similar to decision tree as it is combination of many decision trees, thus are good at handling higher dimensional datasets. Uses a bagging approach to combine multiple trees. It can handle the Categorical data really well similar to the Decision Trees.
- Gradient Boosting
  - It is another tree ensemble model. Similar to the Random Forest Classifier but uses a boosting Approach. It has more hyper parameters to tune and is prune to overfitting compared to the Random forest.

- Stochastic Gradient Descent
  - Stochastic Gradient Descent approach is a supervised incremental gradient descent approach of finding a minimum of the maximum function approximation. It is computationally faster for the larger datasets but takes a lot of time to reach an approximate solution. Also the approximation it achieves is close enough to the actual approximation it achieves through Batch Gradient Descent.
- Decision Tree
  - Decision Trees are supervised learning models which forms a certain decision rules by observing the structure of the data. Advantage of using decision tree is it can handle both numerical and categorical values at the same time. Sometimes it generates over complex trees, so it is necessary to keep an eye on overfitting

The Time series predictive model is implemented through these methods and the data will be subjected to these models in the batches of 6 months for 3 years. So 25 different data points for 6 months for 3 different years will be subjected to these models totaling 900 data points per step. The output label will be 6 months data of the next 6 months. With the error obtained through the comparison

Benchmark

We have kept [5] as our benchmark model for this project as referred earlier. The implementation of that model is really close to the one we have. Here we have tested our model with a larger dataset as there they have used 38 years of data and here we are using 45 years of data. The accuracy claimed by the referred model is 95% which is questionable as no validation/cross-validation methods were discussed in [5]. The accuracy that will be achieved in this method should be equal to the one they have suggested. The name of Metric is not mentioned in the referred report.

III. Methodology

Data Preprocessing

The data taken for this model was of multiple types. Monthly temperature data which ranged from minus few degrees to about 50 degree Celsius. The Monthly rainfall data ranged from 0 to few hundreds of centimeters. And the ENSO and Indian Ocean Dipole data was in a single digit range. These data points if not scaled properly could produce some unreasonable weights in the models and thus a preprocessing was done on all the data samples. The model is supposed to be a Time series model and thus the data is to be framed repeatedly in a time series format as shown in the figure below. The data was arranged in a series for 3 years with totalling 900 data points as an input at any time t. This data was subjected to the model as a series and then the output was retrieved from the model. The 3 year time series was chosen as it was tested with the sets of 1,2,3,4,5 years of data. 3 Years of data was giving the better results than the other 4 different types of data sets. The problem with the 4 or 5 years of data was that the data became too complex and the model was failing to generalise it thus it became a high variance problem. A similar thing can be said about 1 or 2 years of data as the data was too limited thus the model failed to obtain a substential trend in the features.

The Data was also split into 2 sets that were Training Set and the test set. The split was about 75% and 25%, respectively. The Split wasn’t done randomly as the data we need here was time series and randomisation could ruin the property of the data being in the series thuse first 75% of the months were taken to train the model and rest of the months were used to test the model that is to subject the model to the never seen before data.

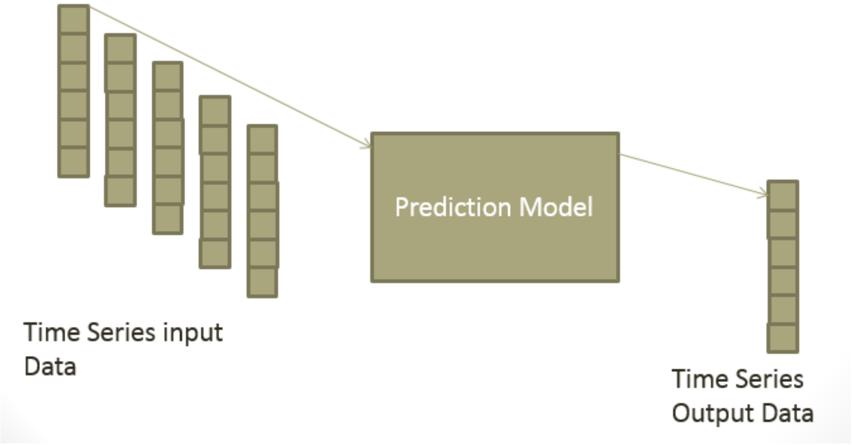

Figure 3: Time series model representation

Implementation

Project design overall plan

- Data Collection
    - Data has been collected in an excel sheet and can be retrieved into the Python workspace. The data_process.py code also arranges it in the time series format and splits it in the train and test set.
- Data Visualization
    - Data was plotted against the El Nino Index and the Indian Ocean Dipole effect and the initial assumption was verified that it is correlated to both of these factors.
- Multi-Layer Perceptron based Regressor:
    - The classifier will be subjected to the raw data first with the default parameters.
    - The grid search will be used to see the improvement after varying the parameters to obtain the best parameters for the predictor model

- Validation
    - To check if the model is functioning correctly the Prediction model will be subjected to the test set. The error and the R^2 score will be calculated.

The Multi-Layer Perceptron based Prediction model was also compared to the other few previously discussed classifiers (all with default parameters). Some Ensemble learning algorithm were producing some comparable results. But Decision Tree and the Stochastic Gradient Descent Regressors were comparatively poor in predicting these results. The Gradient Boost Classifier and Ada Boost classifier were also based on Decision tree classifier [6] as it was set as their base classifier which makes it a variation of normal Decision tree classifier that is an optimized Decision Tree. The result obtained by applying grid search to the Multilayer Perceptron was better than the result observed with Gradient Boost, Random Forest and Ada-boost which was about 0.93. The comparison table is as follows:

Prediction Model	Test Set Score
Multi Layer Perceptron	0.935794653908
Random Forest	0.92065632929
Gradient Boosting	0.922586707298
Stochastic Gradient Descent	0.743984834208
Decision Tree	0.774869205221

Table 1: Comparison of Multiple Prediction Algorithms

Refining

A ‘grid search’ was done on all of these models to get an optimal solution by passing the range of parameters to the grid search function. As discussed earlier, the Multilayer Perceptron based predictor has the highest test set score and thus is a good solution for the problem statement. The Test set R^2 score for ‘Gradient Boosting’ Regressor was obtained to be ‘0.9225’ which is seen to be similar to the Multi-Layer Perceptron Regressor. Same can be said about the Random Forest Regressor with their R^2 scores as 0.9206. Which makes sense as these classifiers are some variations of decisions tree classifiers thus similar results. The results are improved by the factor of 0.1 so there was an improvement of over 10% in this case.

For Multi-Layer Perceptron Regressor, the best estimator obtained was with following hyper-parameters

- Hidden Layer Neuron:- 500
- Activation Function :- Log-sigmoid
- Max Iterations :- 200
- SGD solver for weight optimization
- Momentum :- 0.9
- Regularization :- 0.0001
- Learning rate:- 0.01 constant

200 hidden layer neurons for a 900 input neural net seems to be just enough for this model as once the number of hidden layer went below 150 the model was getting more prune to under fitting. Figure 4 shows a multilayer perceptron architecture.

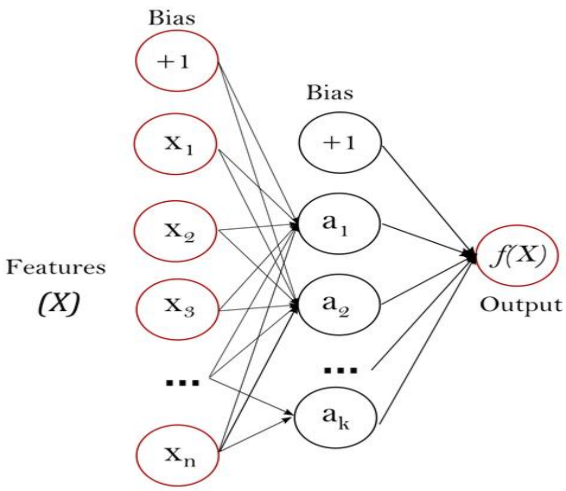

Figure 4: A Multilayer perceptron representation

IV. Results

Model Evaluation and Validation

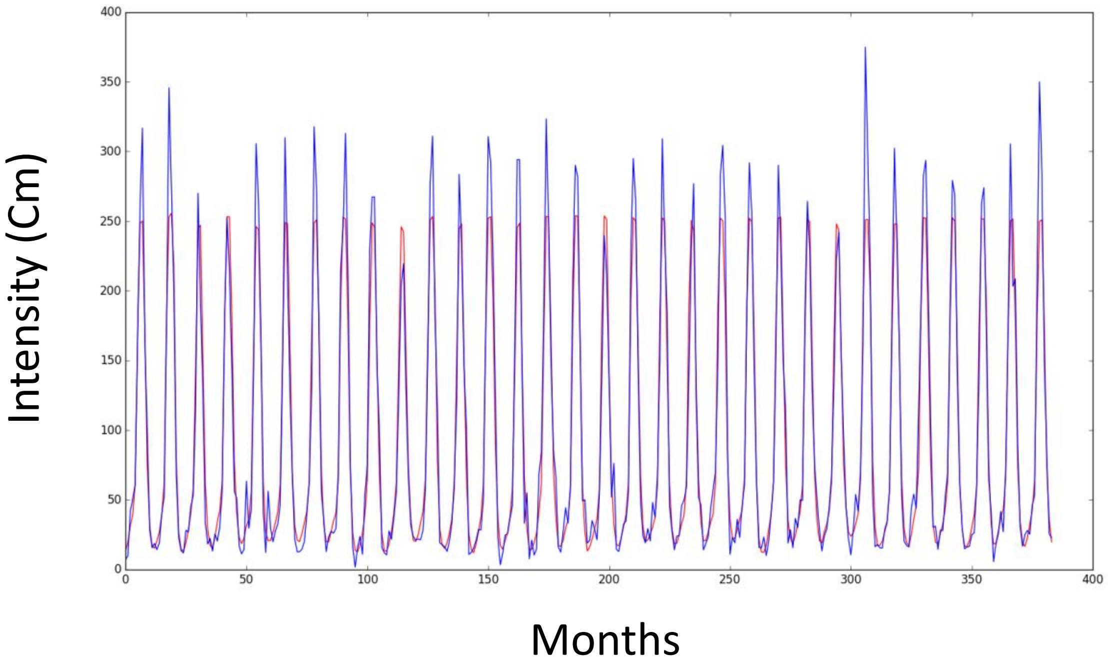

Figure 5: Model Response for the Training Set

The Figure 5 shows the model response for the training set that was used for this project which was about 75% of actual data. The Model fits the Training set pretty well without really overfitting it as a result the accuracy is not 1 but the trend was also perfectly captured which made its R^2 score about 0.98. In Figure 6 we can see the Test set subjected to the model, the data which was never seen by the model itself. Still the model was able to predict the rainfall pretty well as a result it can be seen that the red line following the blue line perfectly with some small error. From which we can conclude that the model predicts the rainfall accurately with the test set score of over 93%.

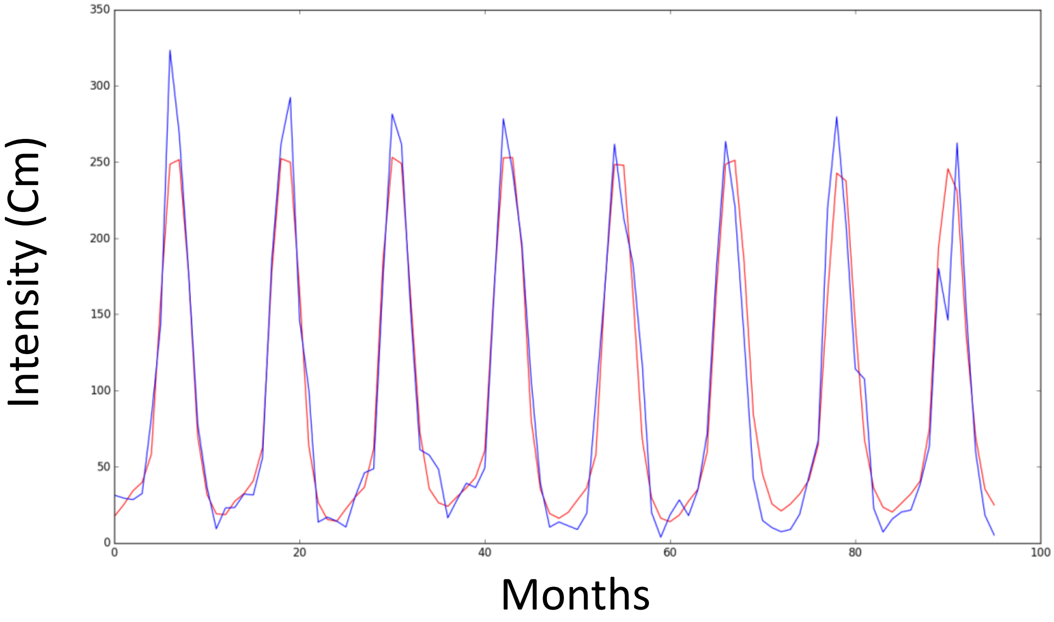

Figure 6: Model Response for the Testing Set

Failed Predictions

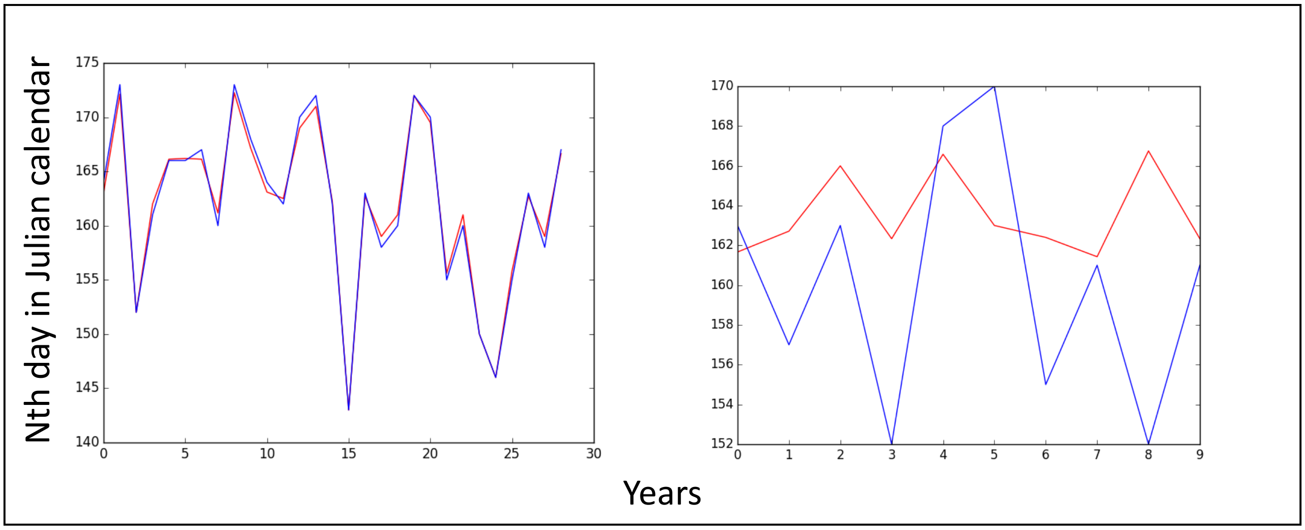

Figure 7: Model Response for the Training and Testing Set of Onset

As seen from Figure 7, the model was also subjected to the Onset data to predict on which day the monsoon will arrive in 1 location of South Asia. As seen from the left plot with the training data, the model fits it perfectly but on the right side it can be seen that there is an error as big as 5-6 days for predicting the onset for the test set. Even though it isn't bad, it can be concluded that the data is insufficient and the results will improve with more data for onset.

Justification

The benchmark model was claimed to be giving result of about 95% which was very close to the results obtained by our predictor which was about 93%. The Metric used here was an R^2 score which is much better and strict than the normal mean square error. The Metric used by the referred paper [5] was not mentioned and thus the comparison could not be done. Even though the model could not achieve results better than the earlier suggested 95% the 93% accurate results are still close enough.

The Prediction has a lot of factors affecting and not all factors were considered for this prediction. Adding more features to the data will probably improve the predictions even further. The data available has only the basic attributes and trends of about 45 years which is good, but not enough for better results. With more data, algorithms like K-NN can be used and could produce even better results than this one [7].

Conclusion

The initial assumption of the Monsoon Rainfall being dependent on previous year's weather conditions seems to be correct as the model returns results with accuracy of about 93% for the monsoon rainfall. The Onset Prediction failed as the data was limited in that case and the model failed to generalize the data. With data of few more years it is possible to create a predictor for better onset prediction. The model which was based on Multi-Layer Perceptron found to be giving the best results as it had better results for the test set from all the algorithms that were in the comparison.

Improvement

As discussed earlier, even though the data was enough for the prediction of intensity it was not enough for the prediction of onset. The data of few more years may improve results for both of these factors. Another improvement that can be done is increasing the granularity of the data. Right now the data used is a monthly data. With availability of daily data, it is possible to implement a deeper neural network based predictor which will be able to handle all the non-linearity in the data.

References

[1] Raman, C. R. V. "Monsoon definitions." *Indian J. Met, and Geoph* 15.2 (1964): 235-238.

[2] Nagelkerke, Nico JD. "A note on a general definition of the coefficient of determination." *Biometrika* 78.3 (1991): 691-692.

[3] Kripalani, R. H., and Ashwini Kulkarni. "Climatic impact of El Nino/La Nina on the Indian monsoon: A new perspective." *Weather* 52.2 (1997): 39-46.

[4] Ashok, Karumuri, Zhaoyong Guan, and Toshio Yamagata. "Impact of the Indian Ocean dipole on the relationship between the Indian monsoon rainfall and ENSO." *Geophysical Research Letters* 28.23 (2001): 4499-4502.

[5] http://www.scidev.net/global/gm/news/india-unveils-new-monsoon-forecast-model.html

[6] Dietterich, Thomas G. "Ensemble learning." *The handbook of brain theory and neural networks* 2 (2002): 110-125.

[7] Beyer, Kevin, et al. "When is “nearest neighbor” meaningful?." *International conference on database theory*. Springer Berlin Heidelberg, 1999.