**Project Mid-Review**

Rishbha Jain, ME16B162

Guide: Prof. S Balaji

Co-guide: Prof. C Balaji

Date: 16/02/2021

**Abstract**:

This project is based on the prediction of tropical cyclones over the Bay of Bengal for different Representative Concentration Pathway (RCP) scenarios that quantify the level of emissions, in the coming years.

**Objective**:

             This project work is aimed at predicting the time windows in a year when a tropical cyclone will occur. We have GCM output data (variables like rel.humidity, mean sea level pressure, etc) for upcoming years as well as ~30 years of past data. The problem here is that a for current mesoscale downscaling models, a basic downscaling of the data to 10km resolution, per year takes around 28 days occupying a space of 400GB. By finding and limiting the general time windows for the occurrence of a cyclone, we can reduce the time taken for researchers to predict cyclone occurrence.

**Progress:**

**Data analysis:**

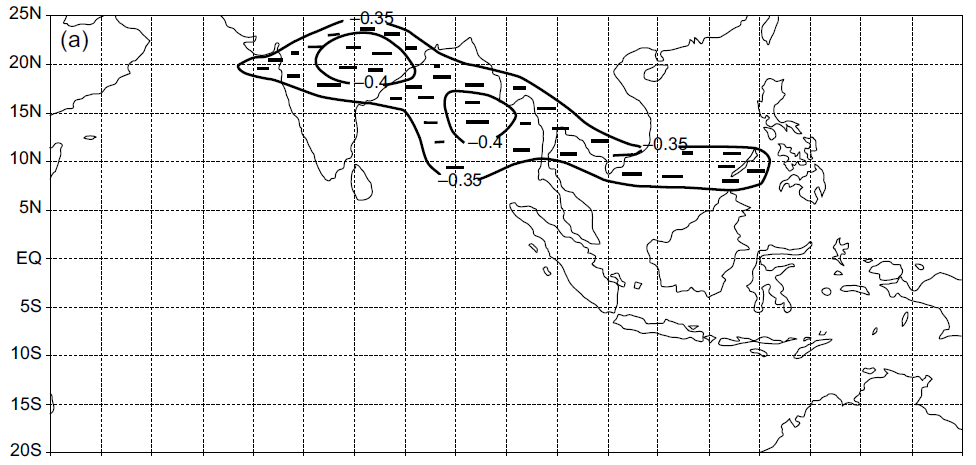
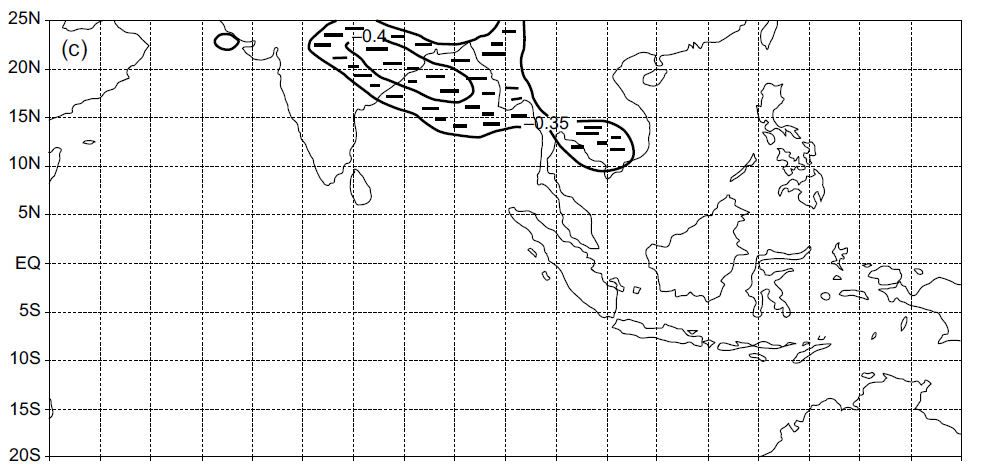
From 1982 to 2020, the month-wise cyclone occurrence distribution over the Bay of Bengal is as follows:

Although cyclones predominantly occur in 2 phases, May-July and October-December, we can’t neglect the remaining months while predicting the occurrence.

**Feature Selection** (Source: [Link1](https://mountainscholar.org/bitstream/handle/10217/247/0234_Bluebook.pdf;sequence=1),[Link2](https://www.intechopen.com/books/recent-developments-in-tropical-cyclone-dynamics-prediction-and-detection))

Different areas and ocean bodies observe different predominant variables that indicate cyclone occurrence. On going through various papers, I found a paper that correlated various climatic variables with TC occurrence count and settled on the following parameters for building a model:

1. Mean Sea Level Pressure: The SST increases and consequently MSLP decreases in an area of tropical cyclone genesis. A low pressure area surrounded by higher pressure is favourable for genesis. There is a negative correlation here.
2. Geopotential height at 500 HPa: There is a negative correlation between TC occurrence and geopotential height at 500 HPa. Geopotential height contours can be used to calculate the wind, which is faster where the contours are more closely spaced and tangential to the geopotential height contours.



Correlation plot between TC counts and left) Mean sea level pressure; right) geopotential height at 500HPa. [Source](https://www.sciencedirect.com/science/article/pii/S0187623617300115#:~:text=A%20neural%20network%20(NN)%20model,October%2C%20November%2C%20December).&text=This%20tropical%20cyclone%20prediction%20technique%20may%20be%20useful%20for%20operational%20prediction%20purposes.)

*Future steps:*

1. I plan on performing these correlations myself with my particular datasets in order to settle on the best features possible.
2. From the graphs, it is clear that the BOB can be further subdivided based on features relevant at different regions. I plan on identifying region specific important features.

**Feature Extraction**

This is an area that I haven’t worked on yet, but plan to. This is imperative to reducing the dimensionality of the data which is a big problem for these models.

**Dataset Creation:**

1. Dependent Variable: Here, I have taken a categorical classification problem, where the dependent variable is a ‘Yes’ or ‘No’ depending on whether a tropical cyclone has occurred in a certain month or not. The dataset was derived from [Best Track (imd.gov.in)](http://www.rsmcnewdelhi.imd.gov.in/index.php?option=com_content&view=article&id=48&Itemid=194&lang=en), and is from 1982-2020.
2. Independent variables: These were derived from [ERA5 hourly data on pressure levels from 1979 to present (copernicus.eu)](https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-pressure-levels?tab=overview), and feature a 10\*10 degree grid (from the advice of the paper) of the 2 features highlighted above over the period in question.
3. Final Data dimensions: training set=320\*112 and testing set=136\*112 for independent variables, and 320\*1, 136\*1 from categorical dependent variable.

**Model Selection:**

With the advice of my professors, I’ve started with a basic classification model logistic regression. Besides that, from the paper I have also tried to run a basic neural network across the data.

*Future steps:*

1. Try different models like Bi-LSTMs and Decision Trees for specific regions in the Bay of Bengal.

**Methodology:**

1. Training/Test data division: I’ve used stratified sampling with a ratio of 0.75 for this. This is so that there will be an equal distribution of the categorical output in the training and test dataset.

1. Logistic Regression (weighted): Logistic Regression is a classification algorithm. Logistic Regression is part of a larger class of algorithms known as Generalized Linear Model (glm)

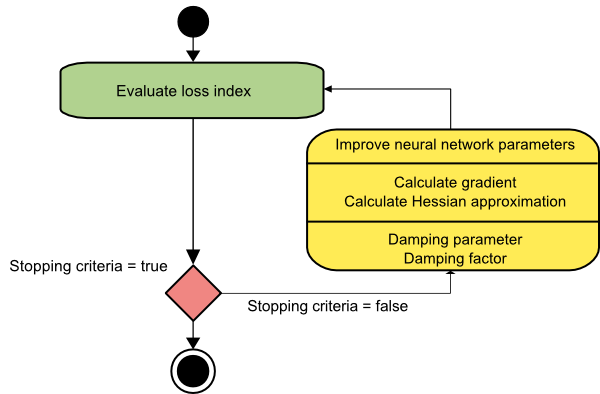
The general equation:

Where p=probability of success for each sample, and V are the independent variables.

Along with the basic linear model, I have performed a weighted logistic regression, where I give samples that a cyclone occurs in (‘1’) a higher weight. This is because: a) There are a disproportionally lower number of data samples where a cyclone has occurred, b) Due to the immensity of the natural disaster, having false negatives is far worse than false positives.

1. Cross Validation: Since we have a limited dataset, with <500 rows, it makes sense to try out k-fold cross validation. The procedure for this, is to shuffle the dataset randomly, split the dataset into k groups. For each unique group, take the group as a hold out or test data set. take the remaining groups as a training data set. Fit a model on the training set and evaluate it on the test set. Keep doing this for each fold. I’ve used 10 fold here, on checking various other fold accuracies.
2. Neural Network (Levenberg Marquadt training):

I’ve built a neural network with 1 hidden layer and 7 hidden neurons. The expression for each neuron in a sigmoid neural network:

[**Source**](https://www.neuraldesigner.com/blog/5_algorithms_to_train_a_neural_network#Levenberg-Marquardt)

**Validation Study:**

1. Logistic Regression: The parameters we had to optimise here were train/test split of the data and where we should place the cut off probability for predicting a ‘1’. For this, I’ve run a grid search algorithm and iterated through both the variables separately.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Data split (across)  Probability (down) | 0.6 | 0.65 | 0.7 | 0.75 | 0.8 | 0.85 |
| 0.5 | 0.604396 | 0.666667 | 0.647059 | 0.701754 | 0.692308 | 0.588235 |
| 0.55 | 0.648352 | 0.641509 | 0.625 | 0.736842 | 0.67033 | 0.617647 |
| 0.6 | 0.686813 | 0.666667 | 0.632353 | 0.622807 | 0.615385 | 0.691176 |
| 0.65 | 0.664835 | 0.647799 | 0.691176 | 0.614035 | 0.593407 | 0.602941 |
| 0.7 | 0.648352 | 0.654088 | 0.654412 | 0.605263 | 0.615385 | 0.632353 |
| 0.75 | 0.60989 | 0.641509 | 0.647059 | 0.666667 | 0.582418 | 0.676471 |
| 0.8 | 0.653846 | 0.666667 | 0.610294 | 0.526316 | 0.593407 | 0.573529 |
| 0.85 | 0.620879 | 0.641509 | 0.654412 | 0.631579 | 0.582418 | 0.647059 |
| 0.9 | 0.620879 | 0.641509 | 0.625 | 0.622807 | 0.615385 | 0.529412 |
| 0.95 | 0.60989 | 0.63522 | 0.573529 | 0.684211 | 0.615385 | 0.617647 |

The optimum data split is a 0.75 split between train and test, and the optimum cut-off probability is 0.55. In order to further validate this, I tabulated the probabilities for data predicted by the model and separated it according to whether the real value is ‘0’ or ‘1’.

(Left) Predicted probabilities by the model when the real output is ‘0’.

(Right) Predicted probabilities by the model when the real output is ‘1’

Neural Network:

I only used a single hidden layer, on the advice of a reference paper (due to our small dataset). In order to find the number of neurons, I ran a validation study:

|  |  |
| --- | --- |
| 3 | 37.7 |
| 4 | 49.7 |
| 5 | 49.1 |
| 6 | 54.8 |
| 7 | 62.7 |
| 8 | 0.49 |
| 9 | 0.47 |
| 10 | 0.5 |
| 11 | 0.52 |
| 12 | 0.5 |

It is clear that 7 neurons is the best model.

**Results:**

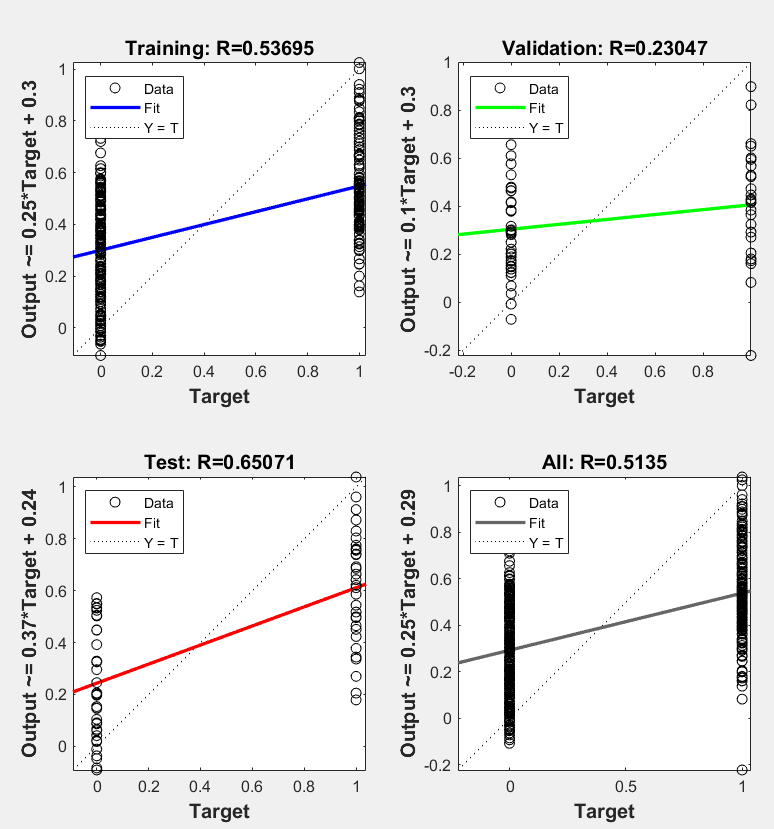
1. **Weighted Logistic Regression:**

|  |  |
| --- | --- |
| **Quantity** | **Value** |
| **Accuracy** | 0.7368 on test data  0.7339 on training data |
| **Confusion Matrix (test data)** | **Prediction**    **Real** |
| **ROC curve** |  |
| **Deviance** | Null deviance: 466.18 on 341 degrees of freedom  Residual deviance: 355.49 on 301 degrees of freedom |
| **Observations** | 1. The accuracy isn’t very good, however there has been a reduction in residuals so the model is definitely significant. 2. The testing accuracy is almost the same as the training accuracy. 3. From the confusion matrix, we see that the model has more false negatives. This may just be because our output data has more ‘0’s than ‘1’s and the model is biased to predicting 0. 4. On running multiple iterations on the same conditions and data, each iteration yielded a different value of accuracy. This implies that we don’t have enough data. 5. One reason why this could be faulty is because we have sequential data, and one assumption of logistic regression is independence of observations and no multicollinearity. |

On cross validating, the accuracy dropped to 0.6171 with SD 0.132. This probably implies that the specific split of test and train data could be the reason for the accuracy above.

1. Neural Network

|  |  |
| --- | --- |
| Quantity | Value |
| Accuracy | 62.7% on test data, with 7 hidden neurons. |
| Observations | Not enough data, so the accuracy will be low. |



Along with this, I’ve tried out various other algorithms, however I couldn’t get the accuracy to go above 70%.

**Atmospheric analysis:**

I tried to understand the threshold that the logistic regression model was predicting for the parameters geopotential height, and the mean sea level pressure. I did this by plotting the predictions from the model and the corresponding parameter, along with whether or not the model prediction is in line with the actual data. The results I got were inconclusive:

(Left) Geopotential height on the x-axis, classification output on the y-axis and colour code based on algorithm accuracy.

(Right) Mean sea level pressure on the x-axis, classification output on the y-axis and colour code based on algorithm accuracy.

Observations:

1. Both the parameters are uniformly distributed along the x axis for both ‘1’ prediction and ‘0’ prediction.
2. There is no clear threshold for both geopotential height and mean sea level pressure that can allow the model to predict the cyclone occurrence.

**Future steps:**

1. Work on dataset creation:
   1. Increase our dataset from before 1982
   2. Perform correlation analysis for my particular use case and find the most relevant parameters for different areas in the Bay of Bengal.
   3. Extract relevant features from theoretical evidence in papers, such as thermal potential mentioned earlier.
   4. Remove multicollinearity from the input data/incorporate time series elements in the model.
2. Models: Try Bi-LSTMs
3. Results visualisation:
   1. Learn the best way to visualise correlations as well as results.
4. Predict and record the results for different future scenarios

**Timeline:**

|  |  |
| --- | --- |
| What to do | Time period |
| Perform Correlations, learn how to visualise geographical correlations | Feb 15th-28th |
| Dataset preparation | March 1st-10th |
| Code and run simulations | March 10th-25th |
| Run future simulations | March 25th - |

           Other models tried:

|  |  |
| --- | --- |
| Model | Accuracy |
| Naïve Bayes with cross validation | 59.42% |
| Scaled conjugate gradient with back-propogation | 69.4% |
| Bayesian Regularization | 68.8% |
| Adaboost (trees: 9 models, depth=1-3) | 64.13% |