**Forecasting runoff using time series data and RNN**

**Introduction:**

Flooding is increasingly becoming one of the most difficult challenges to tackle. As the population is increasing, there is more urbanization resulting in more impervious developments, which results in decrease of infiltration and increase of runoff (sometimes resulting in flooding). If we can predict the runoff in advance, we might be able to predict if there is going to be flooding event in future. I will use time series of weather data (such as rainfall, temperature etc.) from NCDC and other data such as land use, soil, management to develop a multivariate recurrent neural network. This model can be used to predict future runoff using forecasted weather data.

**Objective:**

The objective is to build a RNN time series model that can predict future runoff using forecasted weather data.

**Potential Beneficiaries:**

The model can be used by people who are related to development of real state, by state and government agencies and by insurance industry.

**Dataset Availability:**

We will develop the model for Fish River Watershed in Alabama. The runoff data will be used for training the model and can be downloaded from USGS website.

Weather data set can be downloaded from:

<https://www.ncdc.noaa.gov>

**Methodology:**

All the codes for this project will be developed using python language. I will start the project with data ingestion from the csv files, cleaning the data using data wrangling techniques. Next step, would be to perform exploratory analysis by analyzing the effect of independent variables on the dependent variable (runoff). Calculate initial statistical inferences from the data samples.

Divide the data into training and testing datasets. Use the training dataset to develop a multivariate time series model using Keras. Test the dataset using the rest of the testing dataset. Score the dataset based on the model performance during training and testing phase.

**Deliverables:**

The deliverables for this project will include codes used for the exploratory analysis and model development, results in the form of jupyter notebook and a report presenting the exploratory analysis, model, results, discussion and conclusions.

**Data wrangling:**

Two datasets were imported from the website (runoff from USGS and weather data from NCDC) using their urls:

1. Fish\_river\_flow.txt - This file contains runoff data for Fish River in Alabama from USGS website.
2. Robertsdale\_weather.csv – This file consists of weather data from a weather station located at Robertsdale, Alabama.

The steps taken to clean the data are as follows:

***Generating a dataframe:*** The runoff data (text file) and weather data (csv file) was converted into a pandas dataframe (pd.read\_csv).

***Checking dataframe:*** The dataframe consisting of relevant categories was checked using commands - dataframe.head(), dataframe.info(), and dataframe.describe(). The dataset consists of 7593 rows and 1805 columns.

***Resample data:*** Runoff data was obtained every half an hour, while the weather data we obtained was of daily frequency. In order to match the frequency of both dataset, runoff data was resampled daily (runoff.resample(‘D’).mean()) .

***Merge dataset:*** Both the datasets were merged together (using pfd.merge) using an outer join. This resulted in a combined dataset for runoff, precipitation (PRCP), minimum temperature (TMIN) and maximum temperature (TMAX).

***Missing data:*** In weather dataset we had missing values for precipitation, minimum and maximum temperature. We used forward fill method for filling missing weather data,

**EDA and Inferential Statistics:**

1. *Plot time series data*

Time series plots were generated for runoff, precipitation, TMAX and TMIN using matplotlib.

1. *Check collinearity between independent variables*

A heatmap was plotted to visualize the correlation between all the variables. The plot showed that there is strong relationship between precipitation (PRCP) and runoff. In order to further substantiate the relationship, we used scatter plot (sns.lmplot).

* Correlation coefficient between the two variables was estimated using numpy package (np.corrcoef(x,y)). The estimated value was 0.645

1. *Check stationarity of the time series data*

Dickey fuller test was used to check the stationarity of the time series dataset for runoff and precipitation data.

* Null hypothesis: there is a unit root
* Alternate hypothesis: there is no unit root

P-values for both the series were less than alpha = 0.05, suggesting that we can reject the null hypothesis.

**Data Preparation for Model:**

***Scaling the data:*** All the time series data was scaled between 0 and 1 (using MinMaxScaler (feature\_range=(0,1)).

***Shifting the data:*** The time series data was shifted backwards by one day (using df.shift).

***Combining all the data:*** The time series data shifted by a day and the original time series data were combined to form a dataset. The columns that were irrelevant were removed from the dataset.

***Separation of data into training and testing datasets:*** In order to train the model and test it the data was separated in two portions. 15 years of data (365\*15) were used to train the model and rest of the data (1099 rows) were kept separately for testing the prediction power of the model.

**Model Development:**

An Long Short-Term Memory (LSTM) recurrent neural network was developed using Keras package with tensorflow as the backend. The model was defined with 1 hidden layer consisting of 30 neurons. The input data consisted of 4 variables and output was in the form of one feature, runoff.

While Mean Absolute Error was used as a loss function, and the adam gradient descent was used for training the model and reducing the error. The change in the loss function during the training and testing phase were plotted using matlotlib.

Final R-square and rmse values calculated for predicting the runoff during testing phase were 0.395 and 5.161 respectively.

**Future Developments:**

The model can be explored further by adding new variables and shifting the time series by more number of days.