# FORECASTING RUNOFF USING TIME SERIES DATA AND RNN

CAPSTONE PROJECT - 2

BY: HARSH VARDHAN SINGH

# BACKGROUND

- Flooding is increasingly becoming one of the most difficult challenges
  - more urbanization
  - increase in impervious areas
- If we can predict runoff in advance we may be able to
  - predict flooding in advance
  - prepare for future calamities

# OBJECTIVES

- Provide an exploratory data analysis on different time series variables and obtain inferential statistics

- To develop a RNN time series model that can predict future runoff using forecasted weather data.

# POTENTIAL BENEFICIARIES:

- people working in real state industries
- state and other agencies
- insurance industry

# METHODOLOGY:

- Data Wrangling
- Exploratory Data Analysis
- Inferential Statistics
- Model Development

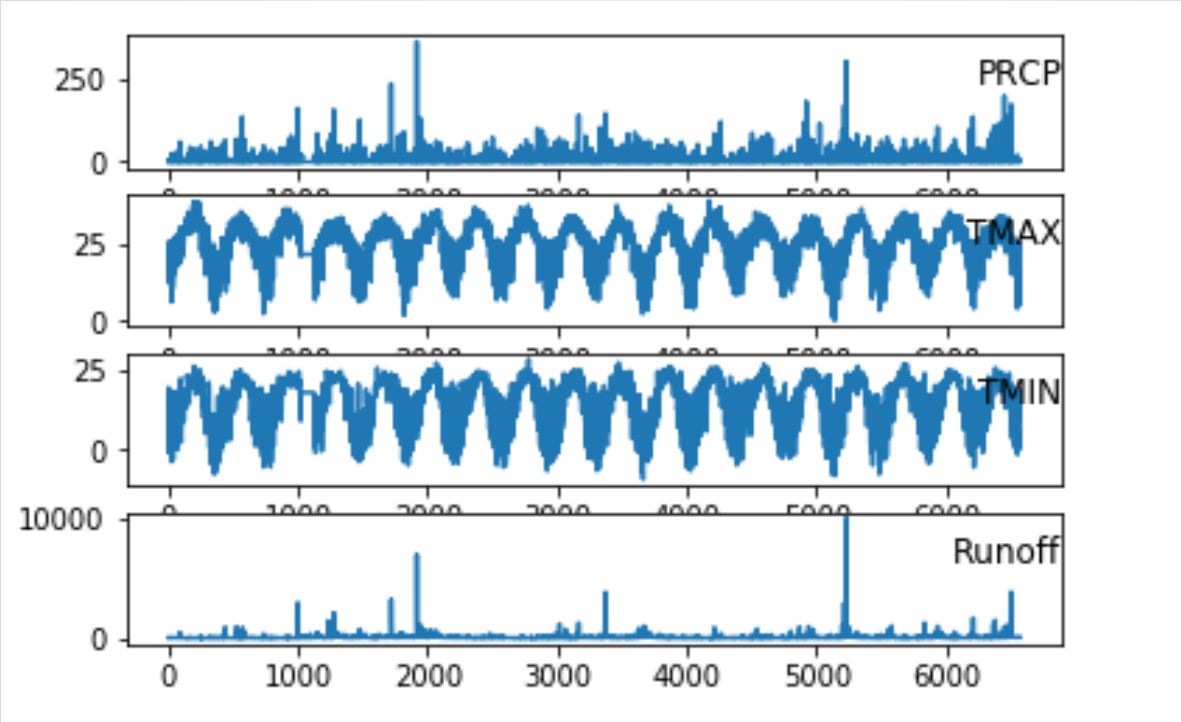
# DATA WRANGLING:

- Import dataset:
  - Fish_river_flow.txt - This file contains runoff data for Fish River in Alabama from USGS website.
  - Robertsdale_weather.csv – This file consists of weather data from a weather station located at Robertsdale, Alabama.
- Generating a dataframe

# DATA WRANGLING:

- Checking dataframe:
  - dataframe.head(), dataframe.info(), and dataframe.describe()
- Resample data into daily format
- Merge datasets: using outer join
- Missing data: using ffilll

# PLOT TIME SERIES DATA:



# INFERENTIAL STATISTICS:

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
2. *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.

# DATA PREPARATION:

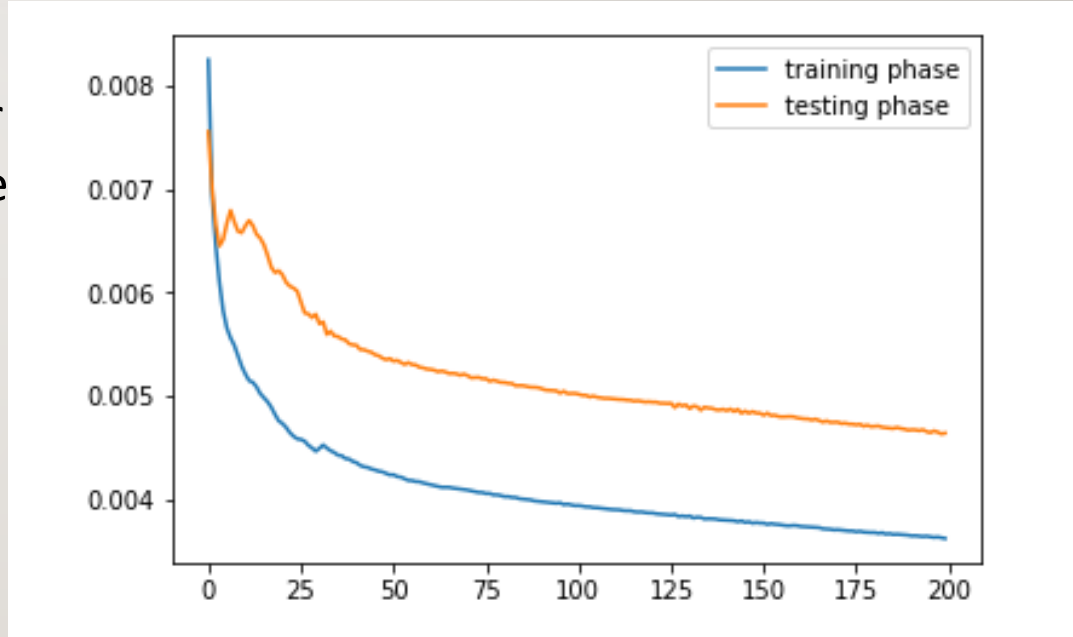
- ***Scaling the data:*** All the time series data was scaled between 0 and 1 (using MinMaxScaler (feature_range=(0,1)).
- ***Shifting the data:*** The time series data was shifted backwards by one day (using df.shift).
- ***Combining all the data:*** The time series data shifted by a day and the original 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:

- A 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.

# MODEL DEVELOPMENT:

- 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.



# THANK YOU