Logic behind model and explanation of process used

The main idea I used after preprocessing and EDA is to Make and Random forest model for those three columns from the data available(Model 1) and train an other Random forest model to do time series analysis(extending those for next three years) of all other significant columns except those three. Now using the Model 1 on these extended columns and date columns(break date columns to day, month, year and take their powers also upto degree 5) predict those three columns and smoothen the result using moving average

- The first thing I observed is this data set is very unorganised
- Most of the data was missing
- And the data that is present is not continuous
- So firstly I observe the columns with almost all zeros

```
Missing values:
Date                                      0
Rainfall_Gallicano                     2859
Rainfall_Pontetetto                    2859
Rainfall_Monte_Serra                   2859
Rainfall_Orentano                      2859
Rainfall_Borgo_a_Mozzano               2859
Rainfall_Piaggione                     3224
Rainfall_Calavorno                     2859
Rainfall_Croce_Arcana                  2859
Rainfall_Tereglio_Coreglia_Antelminelli    2859
Rainfall_Fabbriche_di_Vallico          2859
Depth_to_Groundwater_LT2               2867
Depth_to_Groundwater_SAL               3480
Depth_to_Groundwater_PAG               3955
Depth_to_Groundwater_CoS               3266
Depth_to_Groundwater_DIEC              4499
Temperature_Orentano                      0
Temperature_Monte_Serra                   0
Temperature_Ponte_a_Moriano               0
Temperature_Lucca_Orto_Botanico           0
Volume_POL                             2494
Volume_CC1                             2494
Volume_CC2                             2494
Volume_CSA                             2494
Volume_CSAL                            2494
Hydrometry_Monte_S_Quirico              904
```

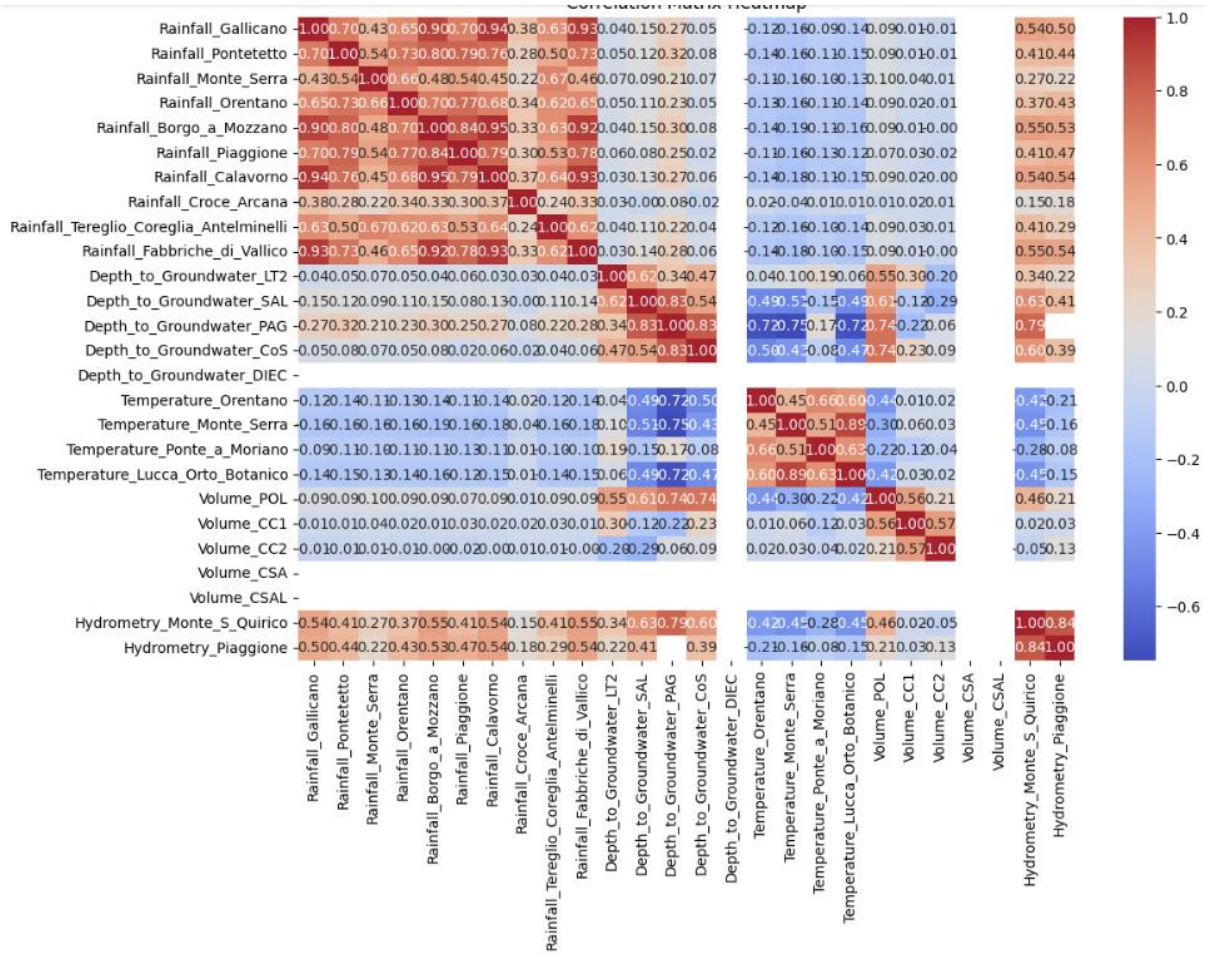
From this observation I removed

Volume_CSAL

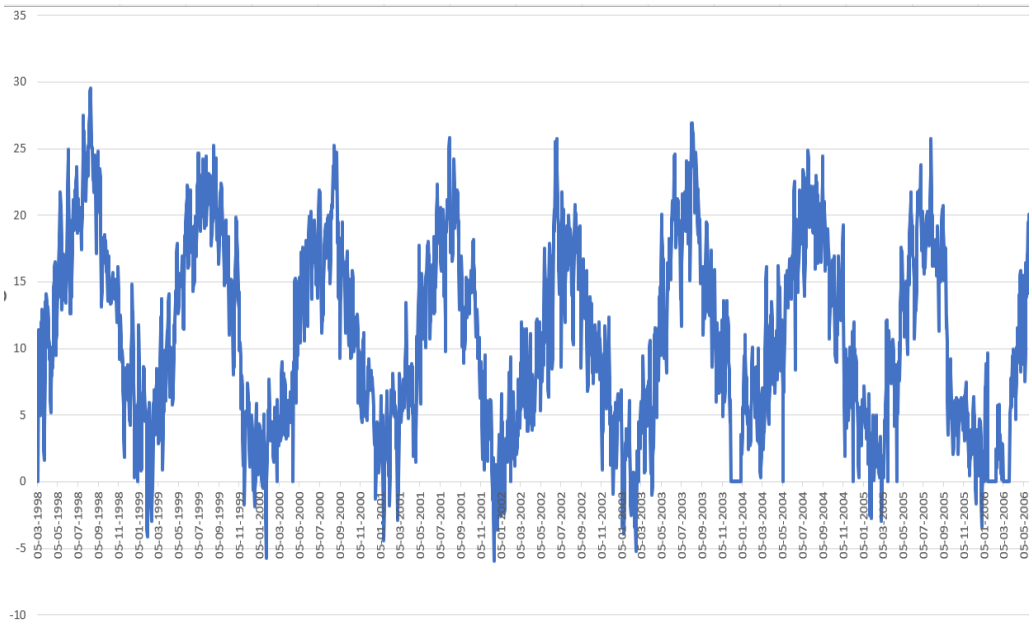
Volume_CSA

Depth_to_Groundwater_DIEC

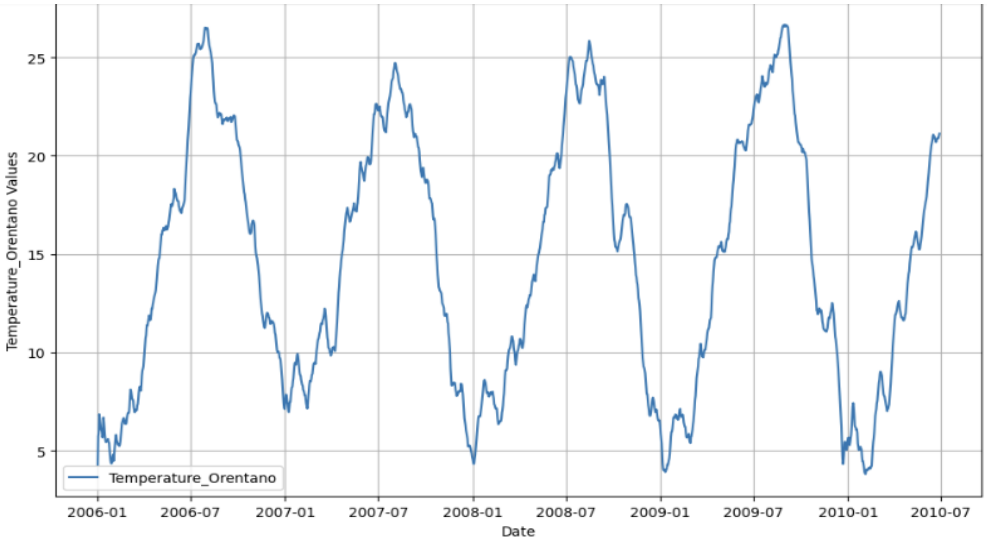
- Then I check correlation between the columns and remove columns which are highly correlated (Correlation matrix can be seen below)



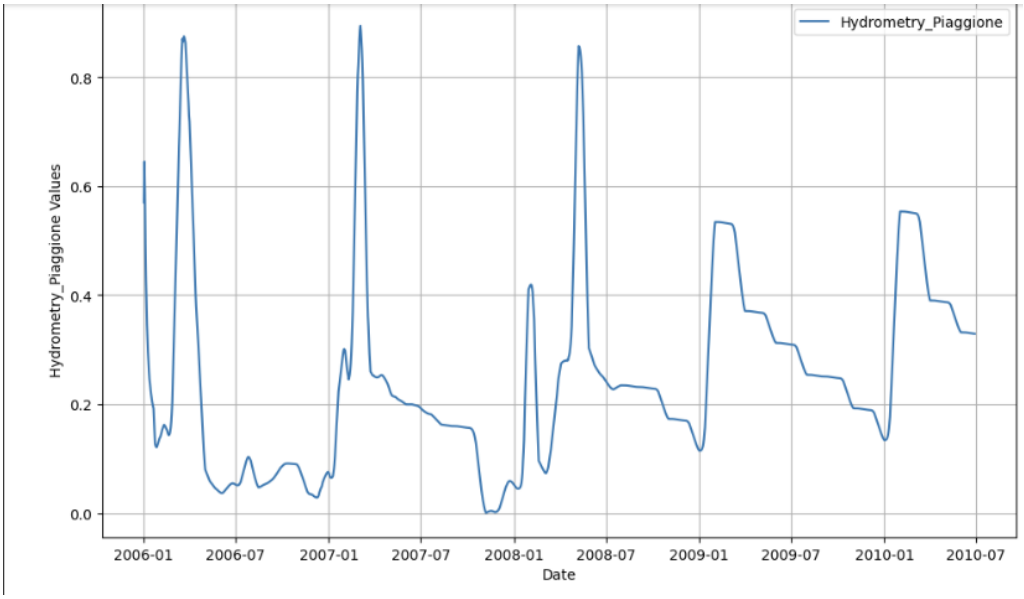
- Then I plotted all the columns and observe their behaviour
  A demo image is given below



- From the above image I observed a lot of noise in the data and in some of the columns some entries were missing also
  For noise reduction - Moving average on an window of 10 samples
  After noise reduction the above plot looks like



For filling missing values – I used Linear regression (trained on data that is available and predict those values which are missing)
Results of regression can be seen below



- Extract features from the Date column and normalize these newly created columns

```
future_data['year2'] = future_data['Year'] ** 2
future_data['year3'] = future_data['Year'] ** 3
future_data['year4'] = future_data['Year'] ** 4
future_data['year5'] = future_data['Year'] ** 5

future_data['month2'] = future_data['Month'] ** 2
future_data['month3'] = future_data['Month'] ** 3
future_data['month4'] = future_data['Month'] ** 4
future_data['month5'] = future_data['Month'] ** 5

min_reference_year = 1998
max_reference_year = 2013
normalize_columns = ['Year', 'Month', 'year2', 'year3', 'year4', 'year5', 'month2', 'month3', 'month4', 'month5']
for column in normalize_columns:
    future_data[column] = (future_data[column] - min_reference_year) / (max_reference_year - min_reference_year)
```

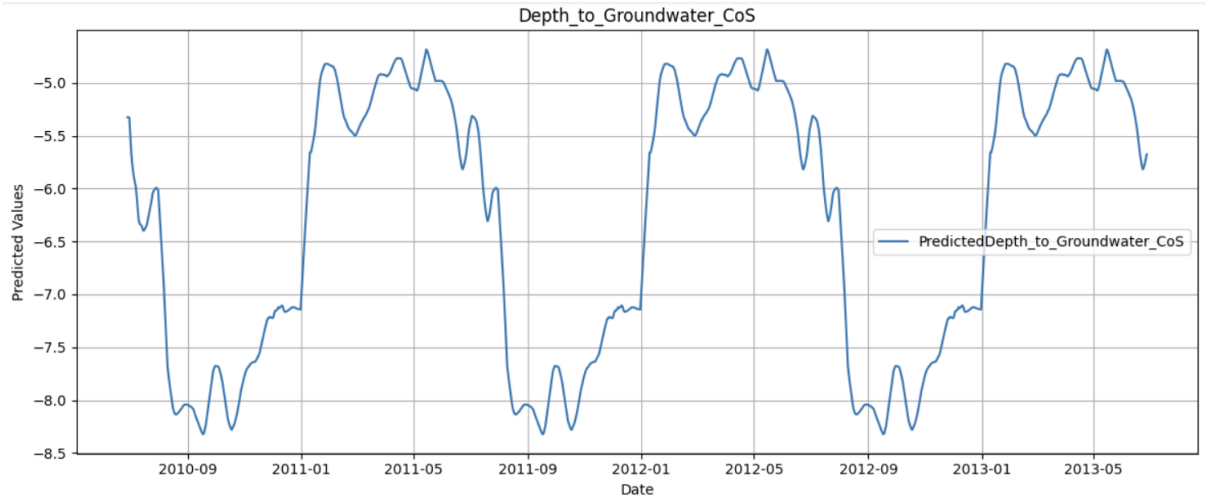
- So then I created and Random Forest Regression between the data preprocessed and those three columns to use later for next three years

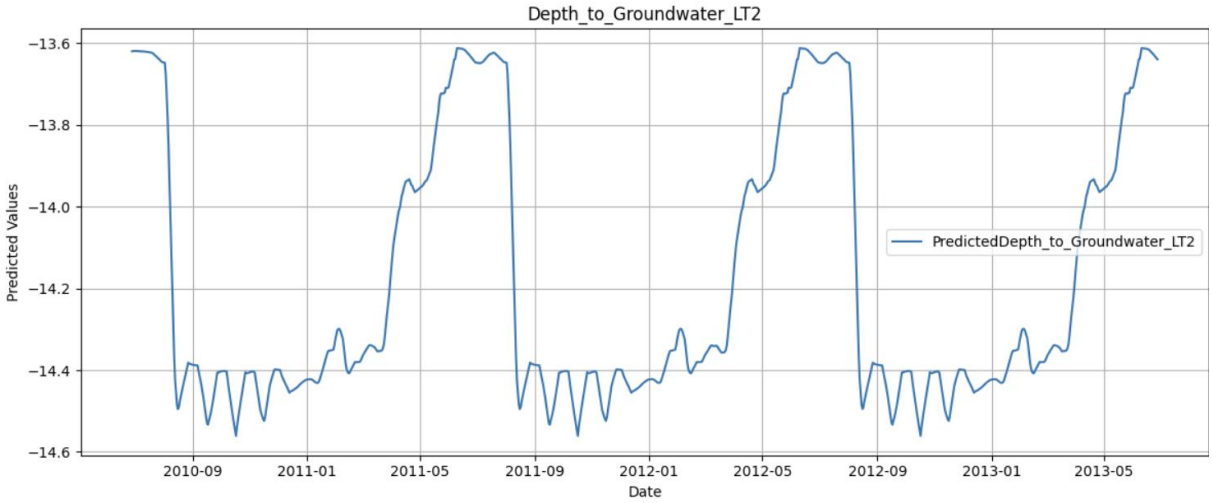
```
# Initialize the Random Forest Regressor
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)

for target_column in target_columns:
    df_filter = df.dropna(subset=target_column)
    X = df_filter[['Year', 'Month', 'year2', 'year3', 'year4', 'year5', 'month2'
    y = df_filter[target_column]
    rf_model.fit(X, y)
    # Predict the target column for the next three years
    future_data[target_column] = rf_model.predict(future_data[['Year', 'Month',
```

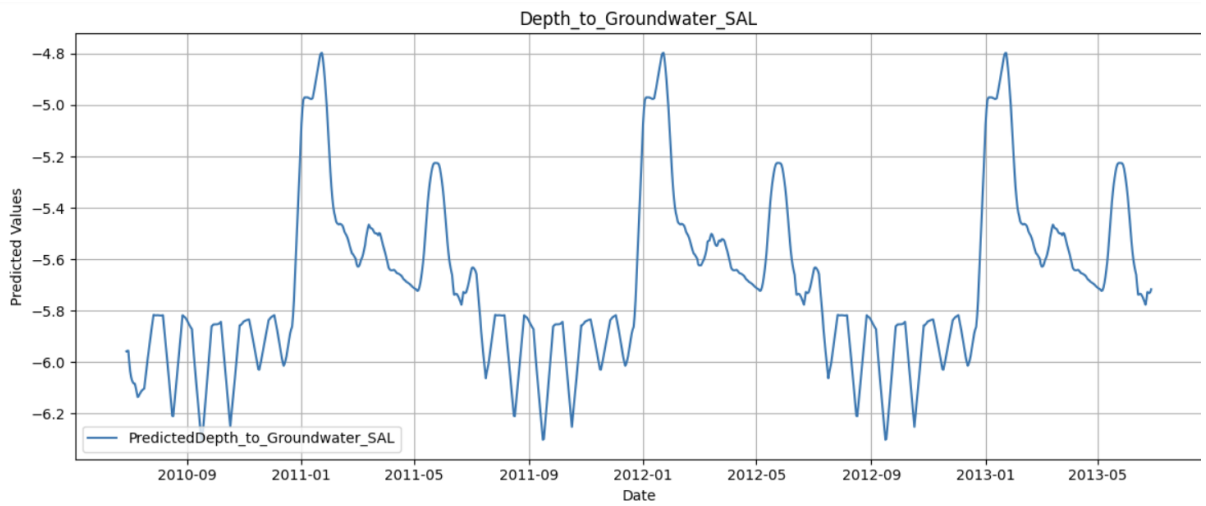
- After that I extended all the significant features for next three years using random forest code is given above
- Then I made another random forest model to predict those three columns based on these time extended features.

```
for target_column in target_columns:

    X = df_filtered[feature_columns]
    y = df_filtered[target_column]

    # Initialize the Random Forest Regressor
    randomf_model = RandomForestRegressor(n_estimators=100, random_state=42)
    randomf_model.fit(X, y)

    # Predict the target column
    future_data[target_column] = randomf_model.predict(predicted_df)
```

- The final results for all three columns are (after smoothening using Moving average over a window of 10 elements)






Depth_to_Groundwater_SAL
Predicted Values
−4.8
−5.0
−5.2
−5.4
−5.6
−5.8
−6.0
−6.2
PredictedDepth_to_Groundwater_SAL
2010-09
2011-01
2011-05
2011-09
2012-01
2012-05
2012-09
2013-01
2013-05
Date