How ML model development Process

PART − 1: Analysis

- Generally we have a data, the data has both input and output columns
- Input columns denoted with X, output column denoted wit y
- We divide data into two parts
  - Train data
  - Test data
- For example we have a data with 1000 observations
- we divide data into 80: 20 ratio or 75: 25 ratio or 90: 10 ratio or 70: 30 ratio
- 80: 20 means 80% Train data 20% Test data
- By default python code divide data into 75: 25
- For 80: 20 out of 1000 observations
  - 800 observations = Train data
  - 200 observations = Test data
- Train data: Data which is used to train the model or develop the model
- Test data : Data which is used to test the model
- Train data also has both input data and output column data
  - Input data = X_{train}
  - Output column data = Y_{train}
- Test data also has both input data and output column data
  - Input data = X_{test}
  - Output column data = Y_{test}
- The model is Developed by using X_{train} and y_{train}
- Once Model is developed we will pass the only X_{test} (test input), this will give predictions and that predictions we called as $y_{predictions}$

- This $y_{predictions}$ compare with y_{test} then we calculate accuracy of the model

PART − 2: Analysis

- Data is divided into two parts, train data and test data
- train data : X_{train} and y_{train}
- Test data: X_{test} and y_{test}
- Model developed by using X_{train} and y_{train}

After model develop we want to test the mode in two ways

- *we will pass the* X_{train} *only*
- *Model will take* X_{train} *only, model will give some outpt*
- *That output we always called as* $y_{predictions}$ *only*
- *Now this* $y_{predictions}$ *will compare with* y_{train}
- *Generally accuracy should be high*
- *Here the error is called as : Train error*
- *Generally Train error should be low*
- *If Train error is high it is called as : UNDERFIT Model*

x y

X_train Y_train

1 1

2 4

3 9

4 16 ========== Y= x*x

x=3 y_prediction=19 vs y_train : train error

X_test y_test

5 25

6 36 x=6 y_pred=34 vs y_test=36 : test error

After model develop we want to test the mode in two ways

- *we will pass the* X_{test} *only*
- *Model will take* X_{test} *only, model will give some outpt*
- *That output we always called as* $y_{predictions}$ *only*
- *Now this* $y_{predictions}$ *will compare with* y_{test}
- *Generally accuracy should be high*
- *Here the error is called as : Test error*

- *Generally Test error should be low*
- *If Test error is high it is called as : Overfit Model*

Data : X and y

tarin data : X_{train} and y_{train}

Test data : X_{test} and y_{test}

Model developed by: X_{train} and Y_{train}

1) you are passing only X_{train} : $y_{prediction}$ vs y_{train} : Train error

2) you are passing only X_{test} : $y_{prediction}$ vs y_{test} : Test error

we have three types of fittings

- *Under fit : train error is very high (No need of test)*
- *Over fit: Train error is low and Test error is Very huge*

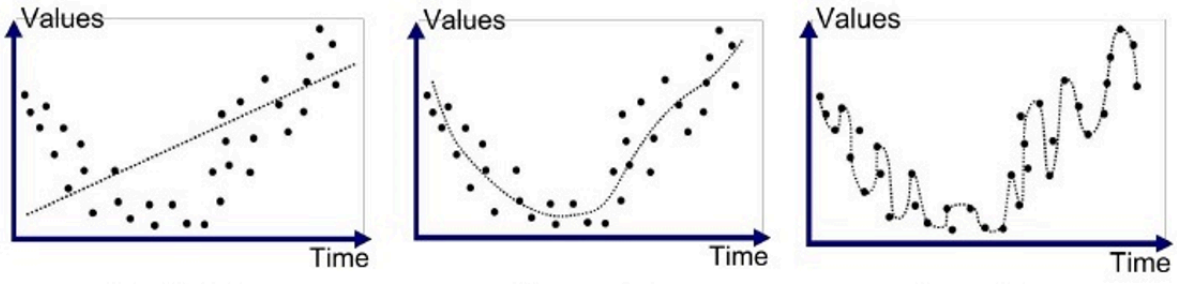
The model is just mugup

The model is Not understanding the patterns

Im giving 5qns and 5 answrs : Mugup

Im giving the 6thqns ===== you are not giving the correct answr: Test error is high

Normal fit: Train error low ==== Test error also low



Underfitted Good Fit/Robust Overfitted