**Approach for Predicting the Pallet Issued for next 6 months using Machine learning**

1. Code is in python 3
2. For data pre-processing imported library are: pandas, numpy, scipy, datetime
3. Imported library for ML:

from sklearn.ensemble import AdaBoostRegressor

from sklearn.metrics import mean\_absolute\_error, r2\_score

from sklearn.model\_selection import RandomizedSearchCV

**Feature engineering/Feature creation**

Framing a time series as a supervised learning dataset using function series\_to\_supervised created by using pandas.DataFrame.shift method of python.

Function Arguments:

1. data: Sequence of observations as a list or NumPy array.
2. n\_in: Number of lag observations as input (X).
3. n\_out: Number of observations as output (y).
4. dropnan: Boolean whether or not to drop rows with NaN values

Returns:

Pandas DataFrame of series framed for supervised learning.

Features considered for machine learning models:

1. BusinessDaysInMonth
2. Features related to varaiable TransfersInMonth (var1) are:

var1(t-6),var1(t-5),var1(t-4),var1(t-3),var1(t-2),var1(t-1) and var1(t)

1. Features related to varaiable IssuesInMonth (var2)are:

var2(t-6),var2(t-5),var2(t-4),var2(t-3),var2(t-2),var2(t-1)

Target Variable: var2(t) (value of IssuesInMonth at time t)

**Model formation using ML method:**

I have used a AdaBoostRegressor (ADA) to train the model after converting the data to supervised learning dataset but also can try with other machine learning models to find the best model. I tried with a linear model as like Ridge but ADA model is better.

Finding optimal parameter for AdaBoostRegressor model using 10 fold cross validation:

I have used a grid of parameter settings method for parameter optimization

#parameter tuning code for AdaBoostRegressor

param\_dist = {

'n\_estimators': [50, 100],

'learning\_rate' : [0.01,0.03,0.04,0.05,0.1,0.3,1],

'loss' : ['linear', 'square', 'exponential'],

'random\_state': [10,20,30,40,50,60]

}

#parameter tuning using grid search method

pre\_gs\_inst = RandomizedSearchCV(AdaBoostRegressor(),

param\_distributions = param\_dist,

cv=10,

n\_iter = 10,

n\_jobs=-1)

pre\_gs\_inst.fit(x\_train\_second, y\_train\_second)

#optimised parameter

pre\_gs\_inst.best\_params\_

{'learning\_rate': 1, 'loss': 'square', 'n\_estimators': 100, 'random\_state': 20}

After finding the optimal parameter I have fit the model using AdaBoostRegressor model with optimal parameters.

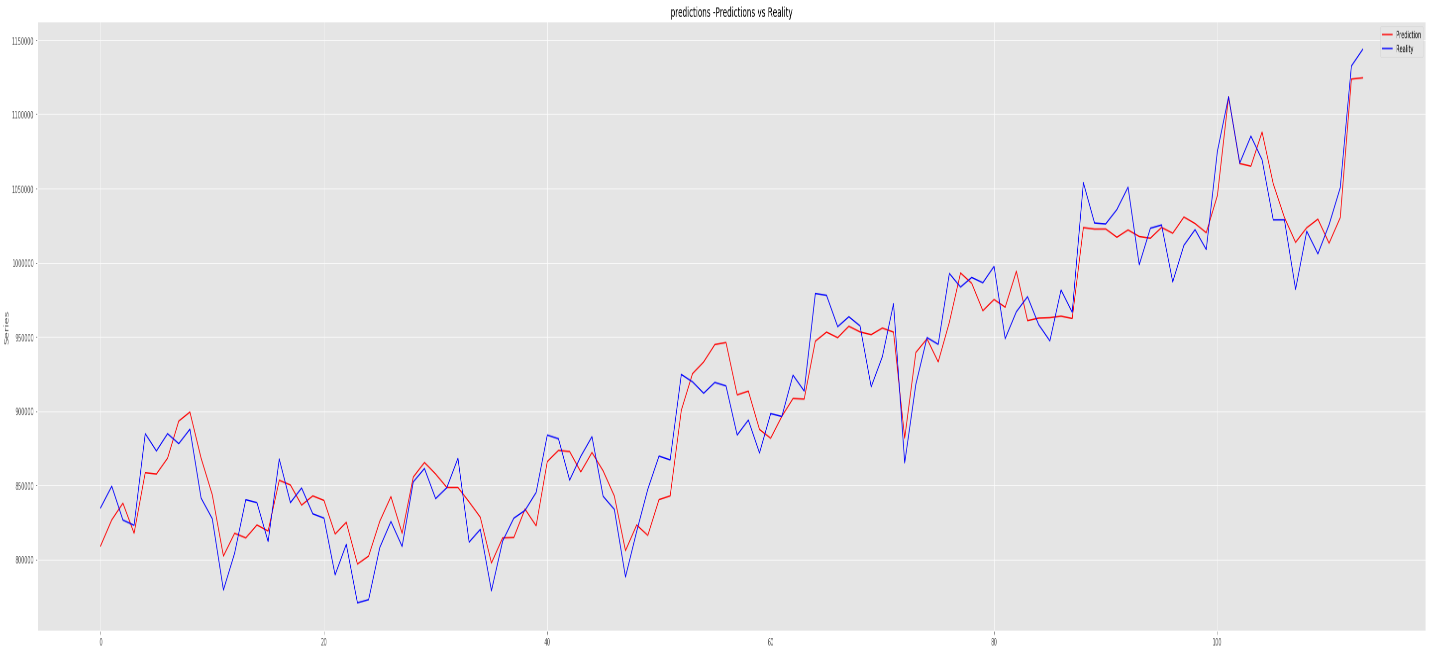
Results:

Mean absolute error when IssuesInMonth is predicted using training data: 1.7497135674796767

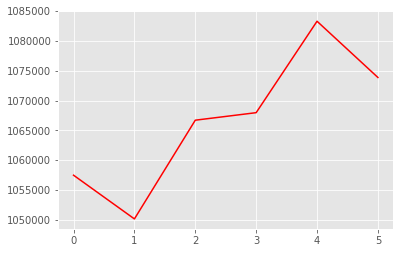
**Actual value vs predicted value (IssuesInMonth) of train data:**

Red: Prediction

Blue: Actual Value



Prediction of IssuesInMonth using test data:



|  | **Month** | **IssuesInMonth** |
| --- | --- | --- |
| **0** | 7/31/2017 | 1.057495e+06 |
| **1** | 8/31/2017 | 1.050167e+06 |
| **2** | 9/30/2017 | 1.066717e+06 |
| **3** | 10/31/2017 | 1.067978e+06 |
| **4** | 11/30/2017 | 1.083304e+06 |
| **5** | 12/31/2017 | 1.073878e+06 |