PROJECT - 6

PYTHON: WALMART STORE PREDICTION – FORECASTING

ABSTRACT:

The objective is predicting store sales using historical markdown data. One challenge of modelling retail data is the need to make decision based on limited history.
If Christmas comes once in in a year, so does the chance to see how strategic decisions impacted the bottom line.

SUMMARIES OF PROBLEM, DATA, METHODS, AND TECHNOLOGIES:

❖ PROBLEM SUMMARY

The project can be sub-divided into three pieces namely –

A> DATA ANALYSIS & VISUALIZATION (PRE PROCESSING)
B> PREPRAING PREDICTION MODEL
C> DATA ANALYSIS & VISUALIZATION (POST PROCESSING)

❖ DATA SUMMARY

The input data provided is in csv data format (sas7bdat). The data need to be imported using 'read_csv' function of 'pandas' library.

Input csv data: \input\

There are total **5** data frames that need to be imported for the solution; below are the data definitions including **sampleSubmission.csv** which is the template for output data to be submitted -

- **stores.csv:** This file contains anonymized information about the 45 stores, indicating the type and size of store.
- **train.csv:** This is the historical training data, which covers to 2010-02-05 to 2012-11-01. Within this file you will find the following fields:
  - ✓ Store - the store number
  - ✓ Dept - the department number
  - ✓ Date - the week
  - ✓ Weekly_Sales - sales for the given department in the given store
  - ✓ IsHoliday - whether the week is a special holiday week
- **test.csv:** This file is identical to train.csv, except we have withheld the weekly sales. You must predict the sales for each triplet of store, department, and date in this file.
- **features.csv:** This file contains additional data related to the store, department, and regional activity for the given dates. It contains the following fields:
  - ✓ Store - the store number
  - ✓ Date - the week
  - ✓ Temperature - average temperature in the region
  - ✓ Fuel_Price - cost of fuel in the region

Email: banerjee.kaustav@outlook.com
Github: https://github.com/KBanerjee90

- ✓ MarkDown1-5 - anonymized data related to promotional markdowns that Walmart is running. MarkDown data is only available after Nov 2011, and is not available for all stores all the time. Any missing value is marked with an NA.
- ✓ CPI - the consumer price index
- ✓ Unemployment - the unemployment rate
- ✓ IsHoliday - whether the week is a special holiday week

For convenience, the four holidays fall within the following weeks in the dataset (not all holidays are in the data):

- ✓ Super Bowl: 12-Feb-10, 11-Feb-11, 10-Feb-12, 8-Feb-13
- ✓ Labor Day: 10-Sep-10, 9-Sep-11, 7-Sep-12, 6-Sep-13
- ✓ Thanksgiving: 26-Nov-10, 25-Nov-11, 23-Nov-12, 29-Nov-13
- ✓ Christmas: 31-Dec-10, 30-Dec-11, 28-Dec-12, 27-Dec-13

❖ METHODS SUMMARY

Table shows the wide variety of data pre-processing, analysis, and visualization techniques that I applied to complete the tasks as part of the project –

Task ID	Analytical Techniques	Visualization Techniques
DATA ANALYSIS & VISUALIZATION	Descriptive statistics using pandas libraries. Straightforward data manipulation	Seaborn (regplot)
PREPRAING PREDICTION MODEL & EVALUATION	Random Forest Regressor and AdaboostRegreesor	

❖ TECHNOLOGIES SUMMARY

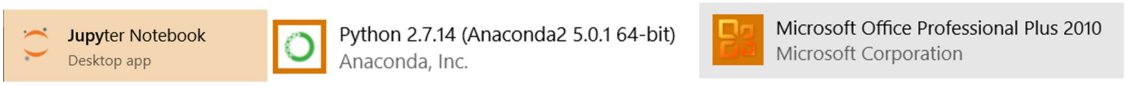
The following list summarizes the technology that I used:

Computing platforms:

Processor:	Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz 2.90 GHz
Installed memory (RAM):	8.00 GB (7.88 GB usable)
System type:	64-bit Operating System, x64-based processor

Email: banerjee.kaustav@outlook.com
Github: https://github.com/KBanerjee90

Microsoft Excel 2010 on the single-machine platform
Jupyter Notebook on the single-machine platform
Python 2.7.14 (Anaconda2 5.0.1 64bit on the single-machine platform



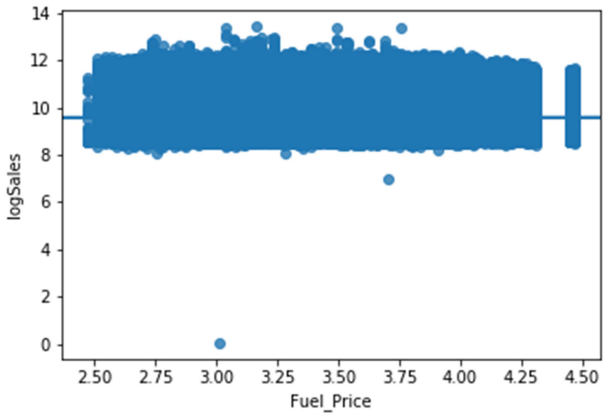
Please note:- The solution can run for 2-3 hours based on resource availability to the system.

WALMART STORE SALES PREDICTION -FORCASTING (ANALYSIS & VISUALIZATION) – PREMODELLING

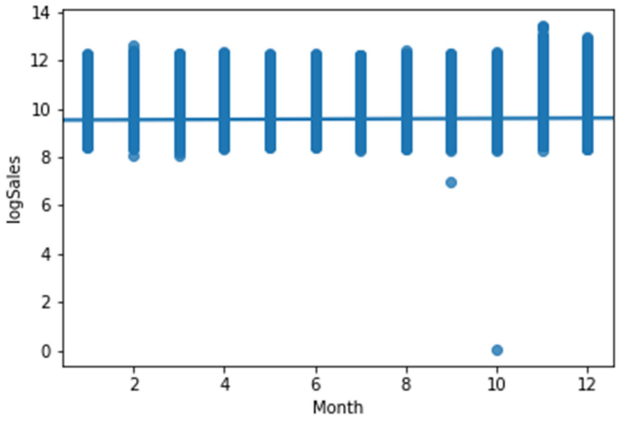
There is no data manipulation required; because the data is clean and good to go; but the volume of data is on higher side which needs consideration –

I joined features and stores to teat and train dataset and create regplot to get the analysis as –

#Plotting Fuel_Price VS logSales: there is no pattern; sales seems pretty moderate accoss Fuel price

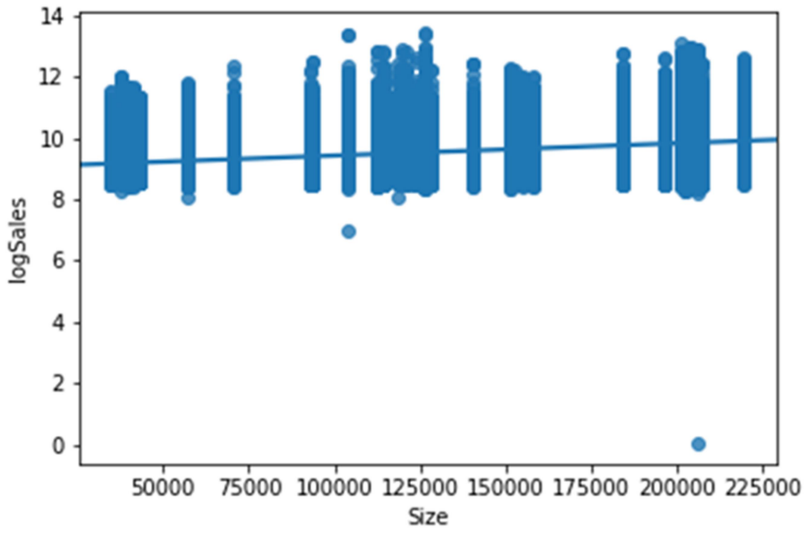


#Plotting Month VS logSales: We see a sale spark during winter/Christmas (Nov – Dec)

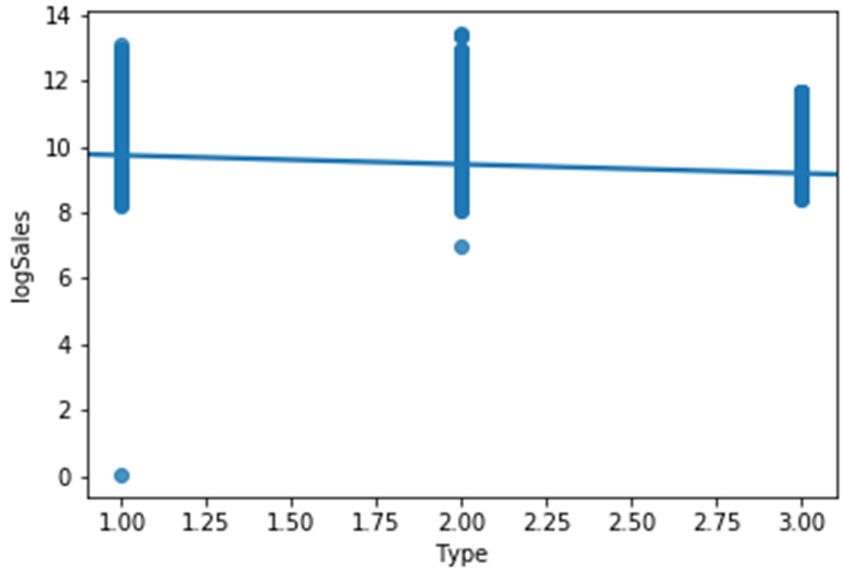


Email: banerjee.kaustav@outlook.com
Github: https://github.com/KBanerjee90

#Plotting Size VS logSales: Size of the store seems to have an increasing pattern with sales(but it saturates at some point)

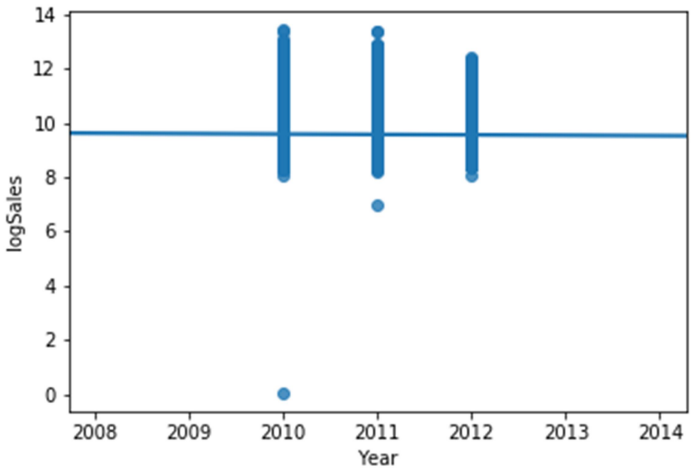


#Plotting Type VS logSales: Store type 3 have significantly lower sales than 1 and 2; an observation

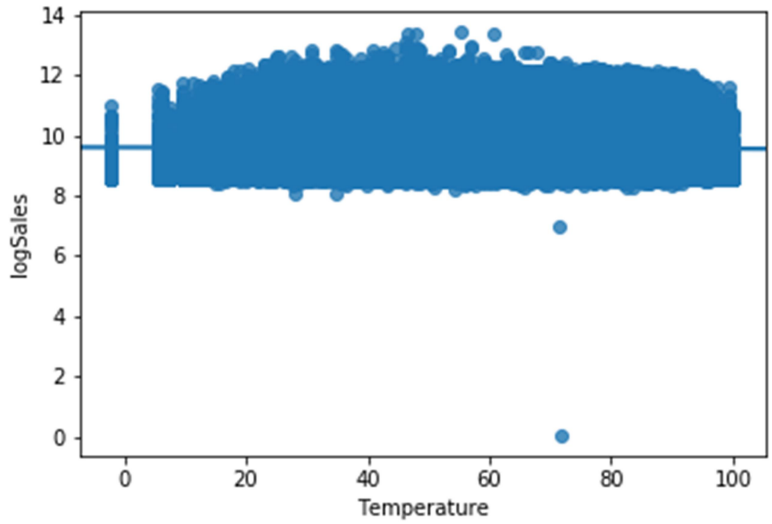


Email: banerjee.kaustav@outlook.com
Github: https://github.com/KBanerjee90

#Plotting Year VS logSales: There seems sharp decline in sales in the year 2012 as compared to 2010.

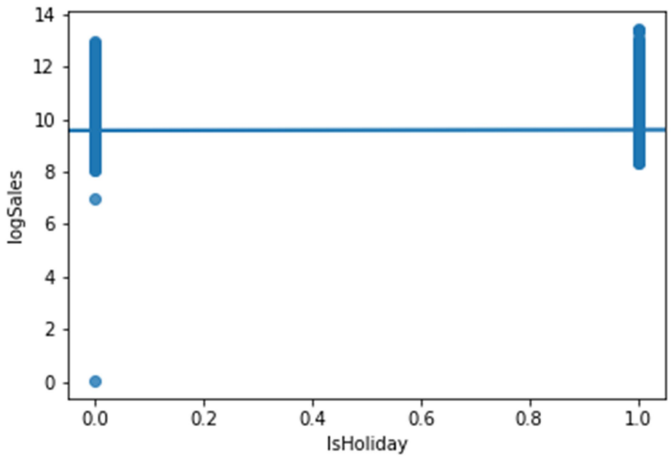


 # Plotting Temperature VS logSales: We see a sale rise where temperature is moderate (40-60 degree)



Email: banerjee.kaustav@outlook.com
Github: https://github.com/KBanerjee90

#Plotting IsHoliday VS logSales: There is no distinct/major difference in mean sale difference between a holiday and non-holiday. An important observation out of pre-processing analysis.



WALMART STORE SALES PREDICTION -FORCASTING (PREDICTION MODEL)

Firstly, I defined a function prosData which will perform the following -

#Step 1 - Importing train, test, stores, features and sampleSubmission template.

#Step 2 - Merging store and feature information train and test data frame.

#Step 3 - Split the Date Field as year, month and Day and also counts the number of days.

#Step 4 - Type conversion categorical to numeric for column 'Type'.

```
train['Type']        = train['Type'].replace('A',1)
train['Type']        = train['Type'].replace('B',2)
train['Type']        = train['Type'].replace('C',3)
```

#Step 5 - Counting the days to next Holiday and log of sales + 4990.

#Step 6 - Dropping MarkDown column since it is available only for 1 year.

Secondly, I used Random Forest Regressor and Ada Boost to get the sales prediction-

#STEP 1 - Defining the input and output file for writing prediction Results.

#STEP 2 - Calling **prosData** to return the train and test dataset.

#STEP 3 - Formatting train and test dataset adding count of department, stores and holiday.

Email: banerjee.kaustav@outlook.com
Github: https://github.com/KBanerjee90

#STEP 4 - **Random Forest Regressor and AdaboostRegreesor** to get the sales prediction.

(Code Snippet)

```
X_train, X_test, y_train, y_test = train_test_split(dataF2.drop(['logSales'],axis=1),np.asarray(dataF2['logSales'],
tmpModel_RF_trabalho = RFreg.fit(X_train,np.asarray(y_train,dtype=float))
tmpModel_RF_Submiss  = RFreg.fit(dataF2.drop(['logSales'],axis=1),
                     np.asarray(dataF2['logSales'],dtype=float))
tmpModel_AB_trabalho = ABreg.fit(X_train,np.asarray(y_train,dtype=float))
tmpModel_AB_Submiss  = ABreg.fit(dataF2.drop(['logSales'],axis=1),
                     np.asarray(dataF2['logSales'],dtype=float))
```

#STEP 5 - Writing error along with accuracy score.

#STEP 6 - Writing in submission file with prediction of sales.

(Output Files snippet -)

```
#Output files
f_Submission_RF       = open('resultRF.csv','w')          #File Submission for RF
f_Submission_AB       = open('resultAB.csv','w')          #File Submission for AB
fmetrics_RF           = open('resultRFmetrics.csv','w')   #File with the metrics for RF
fmetrics_AB           = open('resultABmetrics.csv','w')   #File with the metrics for AB
```

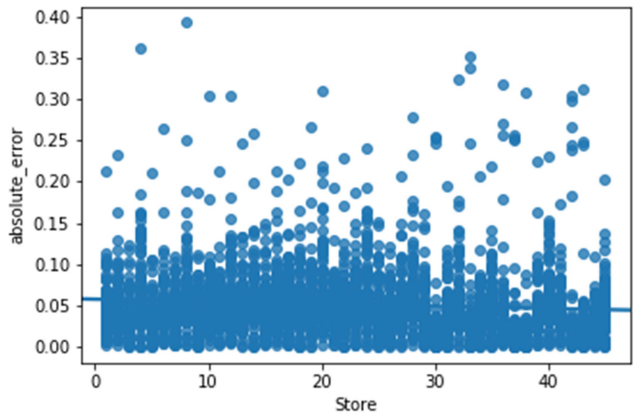
WALMART STORE SALES PREDICTION -FORCASTING (ANALYSIS & VISUALIZATION) – POSTMODELLING

I have generated 16 metric result plots to help me understand the best model, some of them are listed below:-

Metrics plot - output\plots\

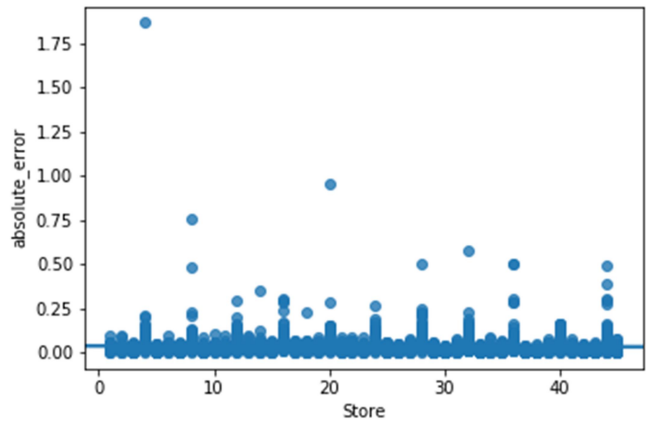
METRICS OF STORES:-

#Plotting absolute_error VS Store – RandomForest

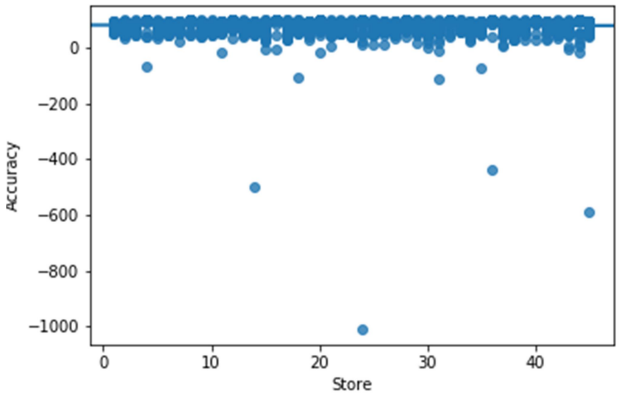


Email: banerjee.kaustav@outlook.com
Github: https://github.com/KBanerjee90

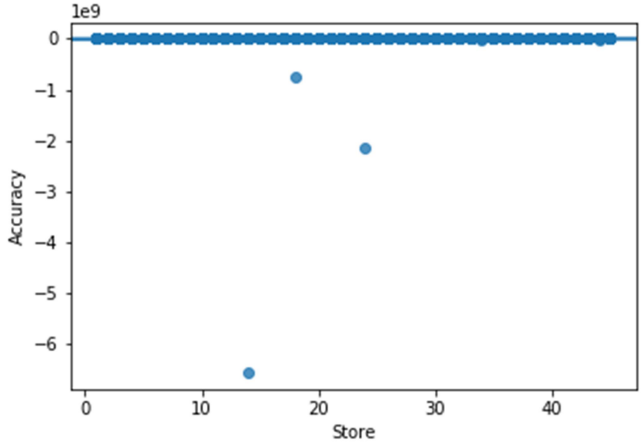
#Plotting absolute_error VS Store – Adaboost



#Plotting Accuracy VS Store – Adaboost



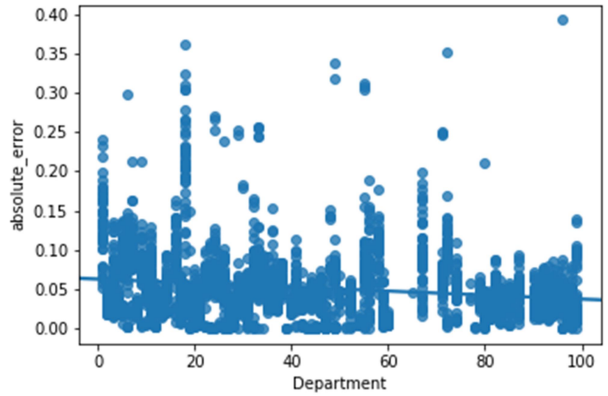
#Plotting Accuracy VS Store – RandomForest



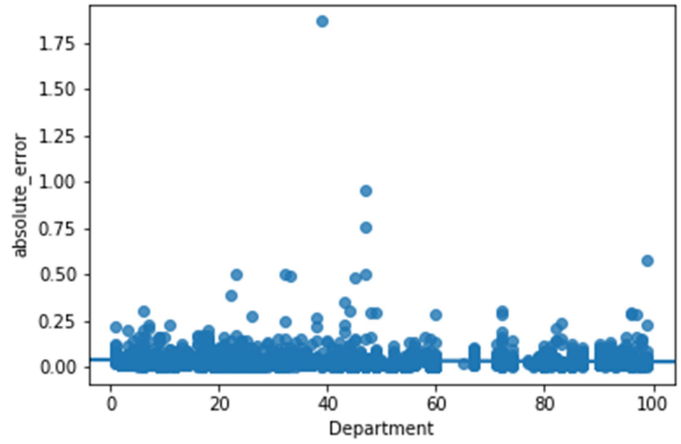
Email: banerjee.kaustav@outlook.com
Github: https://github.com/KBanerjee90

METRICS OF DEPARTMENT:-

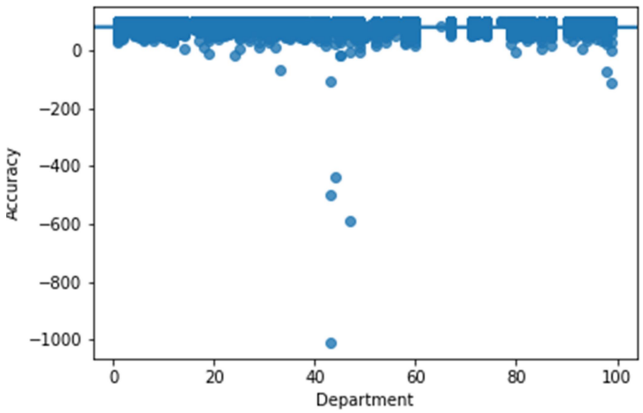
#Plotting Departament VS absolute_error – AdaBoost



#Plotting Departament VS absolute_error – RandomForest

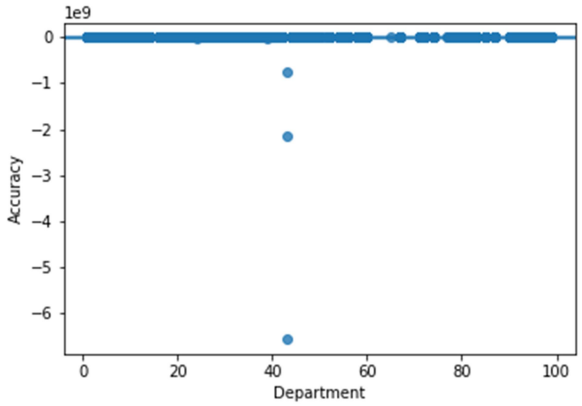


#Plotting Departament VS Accuracy – Adaboost



Email: banerjee.kaustav@outlook.com
Github: https://github.com/KBanerjee90

Plotting Departament VS Accuracy – RandomForest



- **RESULTS**

Result metrics results and plots are available at – output\result\

METRCS AND PREDICTION RESULTS:
resultAB.csv – Prediction Result using AdaBoost **resultABmetrics.csv – Error Metrics using AdaBoost** **resultRF.csv - Prediction Result using Random Forest** **resultRFmetrics.csv - Error Metrics using Random Forest**

Based on the metrics and error; by mean **Random Forest metrics** have **more accuracy** and **less absolute error**.

So we can conclude **RANDOM FOREST REGRESSOR** is a better model**.**

MODEL TYPE	MEAN ACCURACY	MEAN ABSOLUTE ERROR
RANDOM FOREST	93.4	0.03
ADA BOOST	81.75	0.05

Code – solution\ **Walmart_Store_Sales_Prediction_Solution.ipynb**

REFERENCES

https://github.com/

https://www.kaggle.com/

Email: banerjee.kaustav@outlook.com
Github: https://github.com/KBanerjee90