LP1 Assignment DA R3

Bigmart Sales Analysis

Date - 10th September, 2020.

Assignment Number - DA R3

Title

Bigmart Sales Analysis

Problem Definition

For data consisting of transaction records of a sales store. The data has 8523 rows of 12 variables. Predict the sales of a store.

Learning Objectives

- Learn Regression algorithms
- Learn to summarize the properties in the training dataset.
- Learn to split the dataset into training and test datasets.
- Learn to develop a predictive regression model

Learning Outcomes

I will be able to develop a predictive model for sales of an item at BigMart.

Software Packages and Hardware Apparatus Used

- Operating System : 64-bit Ubuntu 18.04
- Programming Language : Python 3
- Jupyter Notebook Environment : Google Colaboratory
- Python Libraries : Kaggle, Kaggle CLI, Sklearn, Pandas, MatPlotLib, Graphviz, PyCaret

Programmer's Perspective

Let S be the system set:

S = {s; e;X; Y; Fme;DD;NDD; Fc; Sc}
where Dataset is loaded into the dataframe

s=start state

e=end state

- predicted sales

X=set of inputs

- X = {X1}
  - where X1 = BigMart Sales Dataset (8523 records, 12 columns)

Y=set of outputs

- Y = {Y1, Y2}
  - Y1 = Predicted Values
  - Y2 = Accuracy Score (Metric - RMSE)

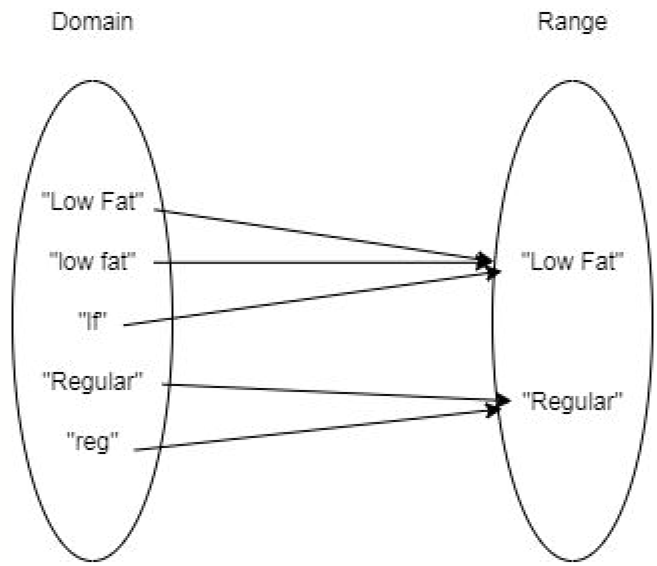
Fme is the set of main functions

- Fe = {f0}
  - f0 = Main Display Function

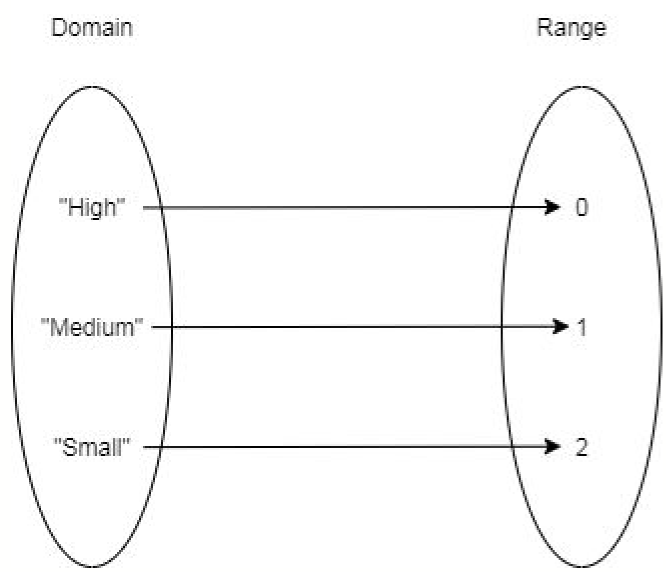
Ff is the set of friend functions

- Ff = {f1,f2,f3,f4,f5,f6} where
  - f1 = function to load dataset into dataframe
  - f2 = function to handle null values
  - f3 = function to handle redundant values
  - f4 = function to generate label encoding of string values
  - f5 = function to split dataset into test and train data
  - f6 = function to train the model

Function to handle Redundant Values



Label Encoding : Outlet Size



DD = Deterministic Data

- BigMart Sales Dataset

NDD = Non-deterministic data (Eg - Null Values in Dataset)

- 1463 null values in the the feature Item_Weight
- 2410 null values in the the feature Outlet_Size

Fc = failure case

- High Value of RMSE
- Low Value of R2

Concepts related Theory

Linear Regression

In statistics, linear regression is a linear approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression.

The relationships are modeled using linear predictor functions whose unknown model parameters are estimated from the data. Such models are called linear models.

If the goal is prediction, forecasting, or error reduction, linear regression can be used to fit a predictive model to an observed data set of values of the response and explanatory variables. After developing such a model, if additional values of the explanatory variables are collected without an accompanying response value, the fitted model can be used to make a prediction of the response.

Given a dataset of n statistical units, a linear regression model assumes that the relationship between the dependent variable y and the p-vector of regressors x is linear. This relationship is modeled through a disturbance term or error variable ε — an unobserved random variable that adds "noise" to the linear relationship between the dependent variable and regressors.

Dataset = $\{y_i,\, x_{i1}, \ldots, x_{ip}\}_{i=1}^{n}$

Model Equation :

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^\mathsf{T} \boldsymbol\beta + \varepsilon_i, \qquad i = 1, \ldots, n,$$

Matrix Notation : $\mathbf{y} = X\boldsymbol\beta + \boldsymbol\varepsilon,$

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix},$$

$$X = \begin{pmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_n^\top \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix},$$

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

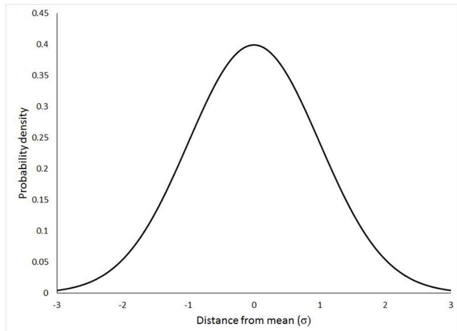
Dataset Description

The data scientists at BigMart have collected 2013 sales data for 1559 products across 10 stores in different cities. Also, certain attributes of each product and store have been defined. The aim is to build a predictive model and find out the sales of each product at a particular store.

- **Item_Identifier**: Unique product ID
- **Item_Weight**: Weight of product
- **Item_Fat_Content**: Whether the product is low fat or not
- **Item_Visibility**: The % of total display area of all products in a store allocated to the particular product
- **Item_Type**: The category to which the product belongs
- **Item_MRP**: Maximum Retail Price (list price) of the product
- **Outlet_Identifier**: Unique store ID
- **Outlet_Establishment_Year**: The year in which store was established
- **Outlet_Size**: The size of the store in terms of ground area covered
- **Outlet_Location_Type**: The type of city in which the store is located
- **Outlet_Type**: Whether the outlet is just a grocery store or some sort of supermarket
- **Item_Outlet_Sales**: Sales of the product in the particular store. This is the outcome variable to be predicted.

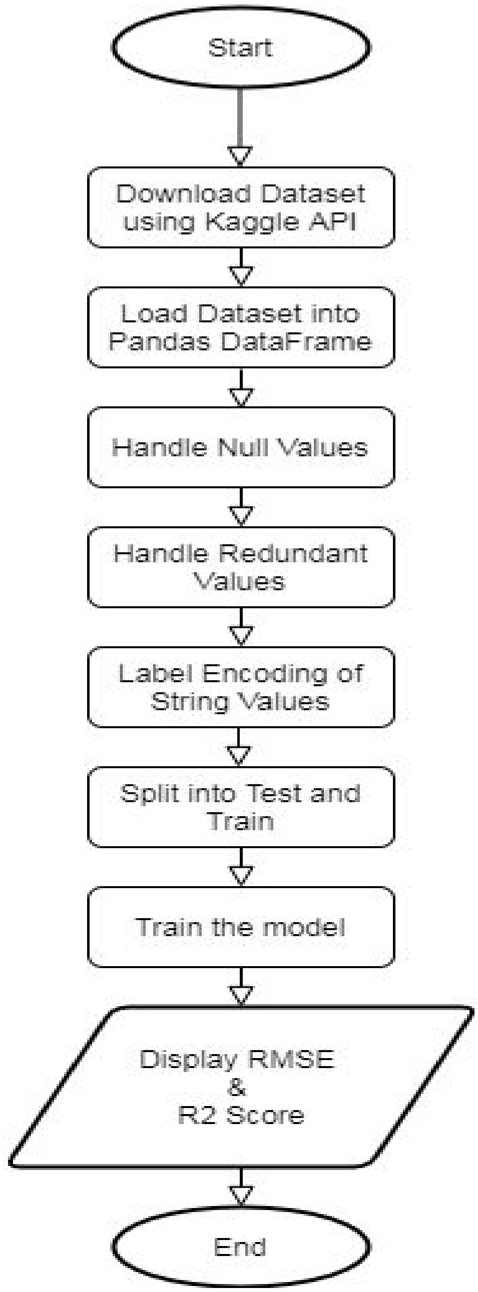
Gaussian Distribution

It is a symmetric distribution where most of the observations cluster around the central peak and the probabilities for values further away from the mean taper off equally in both directions.



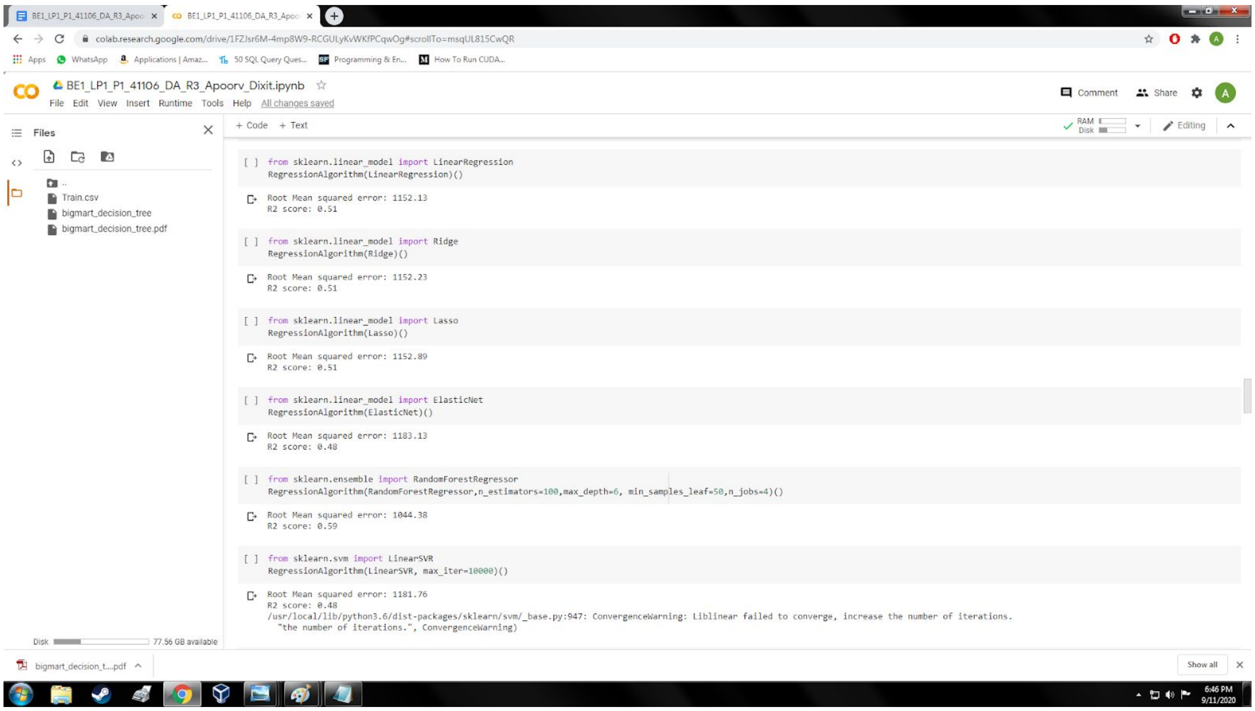
To deal with missing values in a numerical feature with gaussian distribution, we can calculate the mean of the feature and replace it with the missing values. This is an approximation which can add variance to the data set. But the loss of the data can be negated by this method which yields better results compared to removal of rows and columns. Replacing with mean, median or mode is a statistical approach of handling the missing values. This method is also called leaking the data while training.

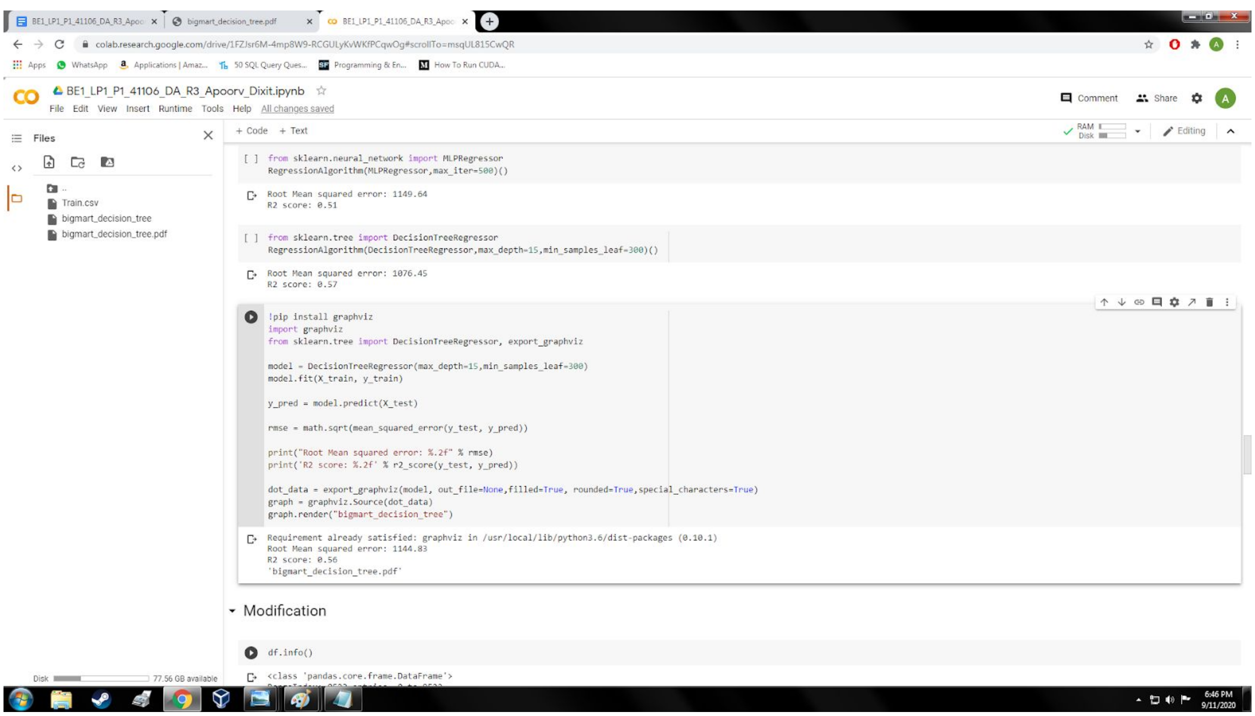
Flowchart


Start
Download Dataset using Kaggle API
Load Dataset into Pandas DataFrame
Handle Null Values
Handle Redundant Values
Label Encoding of String Values
Split into Test and Train
Train the model
Display RMSE & R2 Score
End

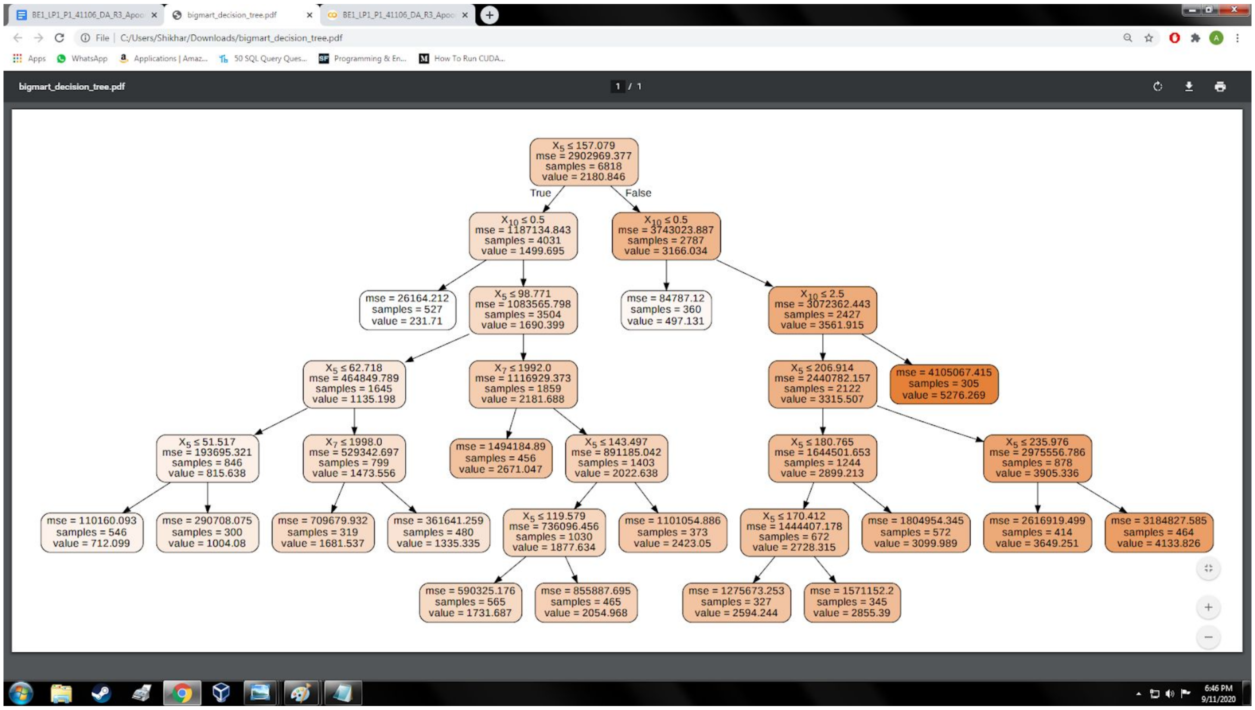
Output Screenshots

Regression Models - RMSE and R2 Score

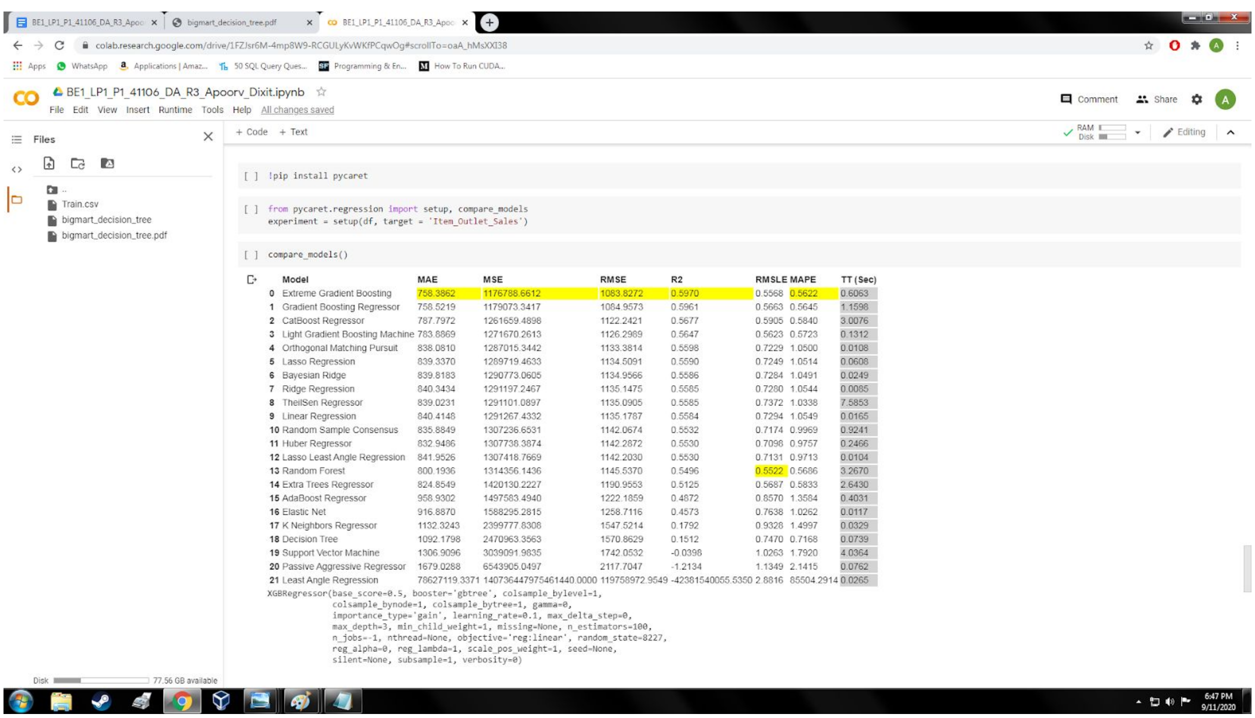




Decision Trees of Decision Tree Regressor



PyCaret Python Library



```
!pip install pycaret
```

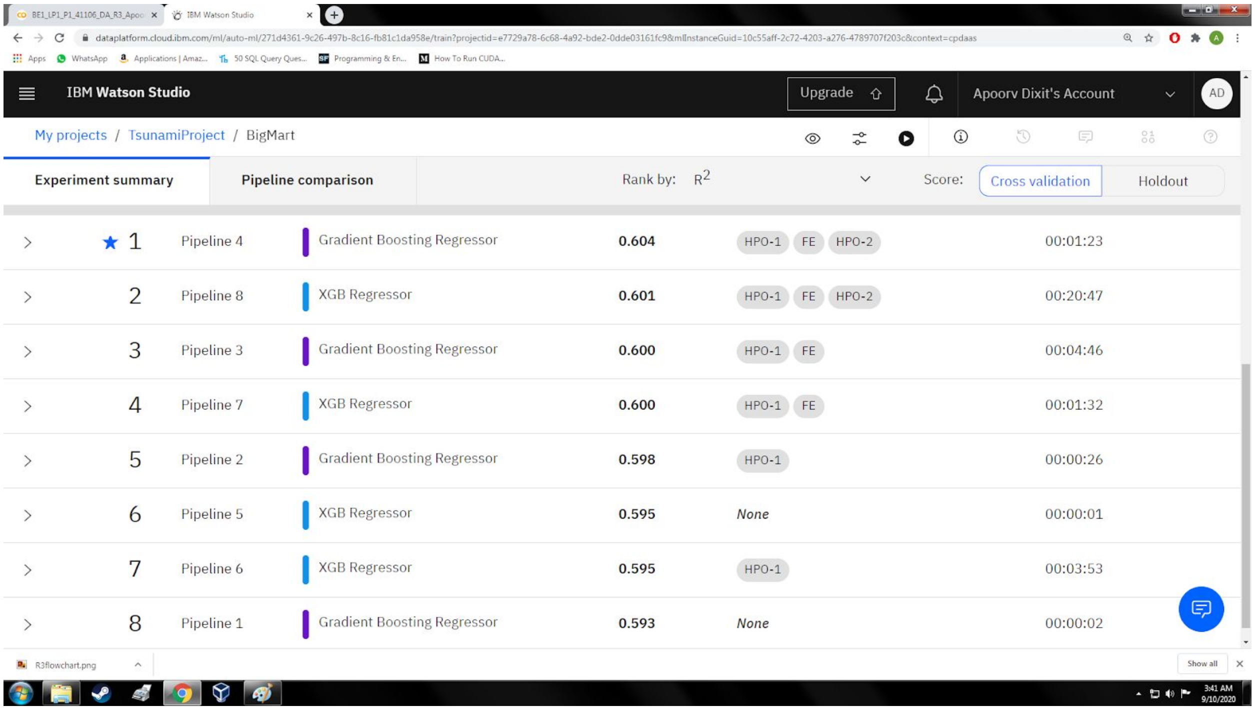
```
from pycaret.regression import setup, compare_models
experiment = setup(df, target = 'Item_Outlet_Sales')
```

```
compare_models()
```

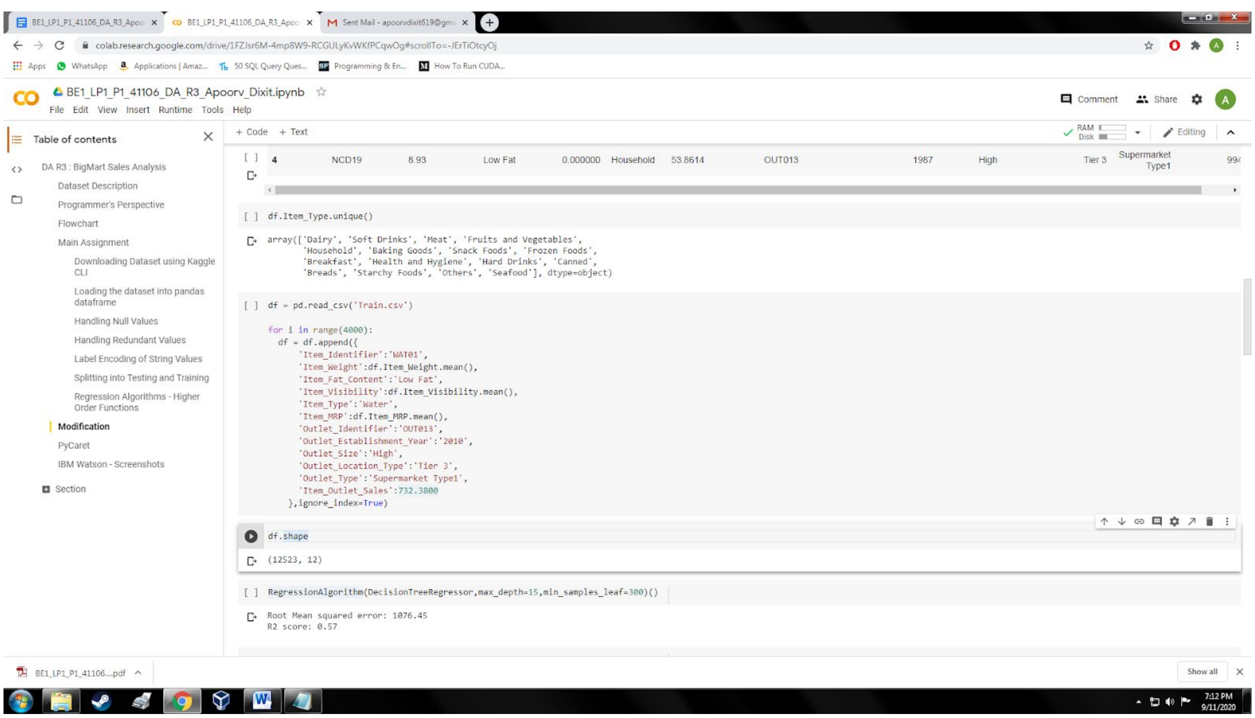
	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
0	Extreme Gradient Boosting	758.3862	1176788.6612	1083.8272	0.5970	0.5568	0.5622	0.6063
1	Gradient Boosting Regressor	758.5219	1179073.3417	1084.9573	0.5961	0.5663	0.5645	1.1598
2	CatBoost Regressor	787.7972	1261659.4898	1122.2421	0.5677	0.5905	0.5840	3.0076
3	Light Gradient Boosting Machine	783.8869	1271670.2613	1126.2989	0.5647	0.5623	0.5723	0.1312
4	Orthogonal Matching Pursuit	838.0810	1287015.3442	1133.3814	0.5598	0.7229	1.0500	0.0108
5	Lasso Regression	839.3370	1289719.4633	1134.5091	0.5590	0.7249	1.0514	0.0608
6	Bayesian Ridge	839.8183	1290773.0605	1134.9566	0.5586	0.7284	1.0491	0.0249
7	Ridge Regression	840.3434	1291197.2467	1135.1475	0.5585	0.7280	1.0544	0.0085
8	TheilSen Regressor	839.0231	1291101.0897	1135.0905	0.5585	0.7372	1.0338	7.5853
9	Linear Regression	840.4148	1291267.4332	1135.1787	0.5584	0.7294	1.0549	0.0165
10	Random Sample Consensus	835.8849	1307236.6531	1142.0674	0.5532	0.7174	0.9969	0.9241
11	Huber Regressor	832.9486	1307738.3874	1142.2872	0.5530	0.7098	0.9757	0.2466
12	Lasso Least Angle Regression	841.9526	1307418.7669	1142.2030	0.5530	0.7131	0.9713	0.0104
13	Random Forest	800.1936	1314356.1436	1145.5370	0.5496	0.5522	0.5686	3.2670
14	Extra Trees Regressor	824.8549	1420130.2227	1190.9553	0.5125	0.5687	0.5833	2.6430
15	AdaBoost Regressor	958.9302	1497583.4940	1222.1859	0.4872	0.8570	1.3584	0.4031
16	Elastic Net	916.8870	1588295.2815	1258.7116	0.4573	0.7638	1.0262	0.0117
17	K Neighbors Regressor	1132.3243	2399777.8308	1547.5214	0.1792	0.9328	1.4997	0.0329
18	Decision Tree	1092.1798	2470963.3563	1570.8629	0.1512	0.7470	0.7168	0.0739
19	Support Vector Machine	1306.9096	3039091.9835	1742.0532	-0.0398	1.0263	1.7920	4.0364
20	Passive Aggressive Regressor	1679.0288	6543905.0497	2117.7047	-1.2134	1.1349	2.1415	0.0762
21	Least Angle Regression	78627119.3371	140736447975461440.0000	119758972.9549	-42381540055.5350	2.8816	85504.2914	0.0265

```
XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,
             colsample_bynode=1, colsample_bytree=1, gamma=0,
             importance_type='gain', learning_rate=0.1, max_delta_step=0,
             max_depth=3, min_child_weight=1, missing=None, n_estimators=100,
             n_jobs=-1, nthread=None, objective='reg:linear', random_state=8227,
             reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
             silent=None, subsample=1, verbosity=0)
```

IBM Watson Machine Learning Auto AI Experiment



Adding Item “Water”



Conclusion

I have successfully developed a predictive model for sales of an item at BigMart.