LP1 Assignment DA 3

Date: 7th October, 2020

Title: Big Mart 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 Big Mart.

S/W & H/W Packages:

1. Operating System: 64-bit Open source Linux or its derivative
2. Programming Language: Python3
3. Google Colaboratory(uses Tesla K80 GPU)

Related Mathematics:

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

DD = Deterministic Data

- BigMart Sales Dataset

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

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

Fc = failure case

- High Value of RMSE
- Low Value of R2

Concepts Related to Theory:

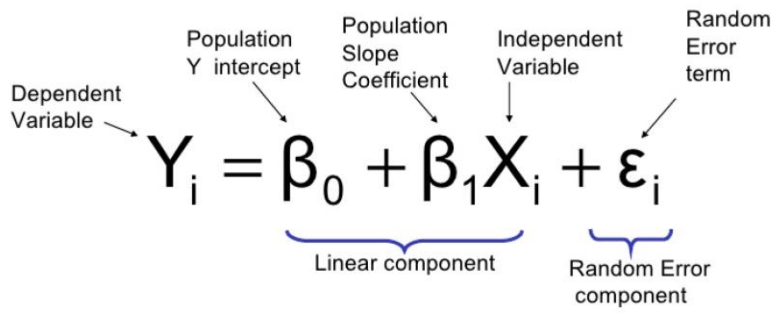
Linear Regression

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.



$$Y_i = \underbrace{\beta_0 + \beta_1 X_i}_{\text{Linear component}} + \underbrace{\varepsilon_i}_{\text{Random Error component}}$$

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

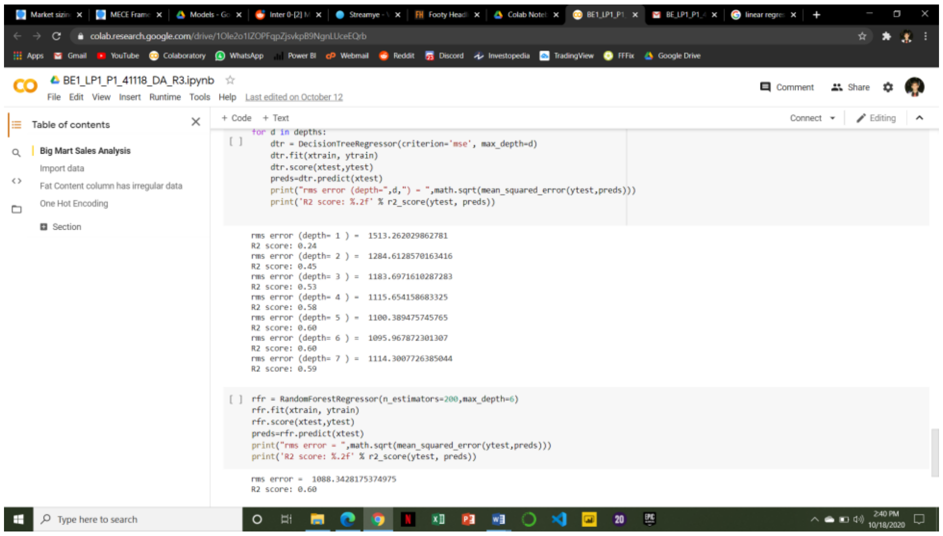
● Item_Weight: Weight of product

● 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.

Output:

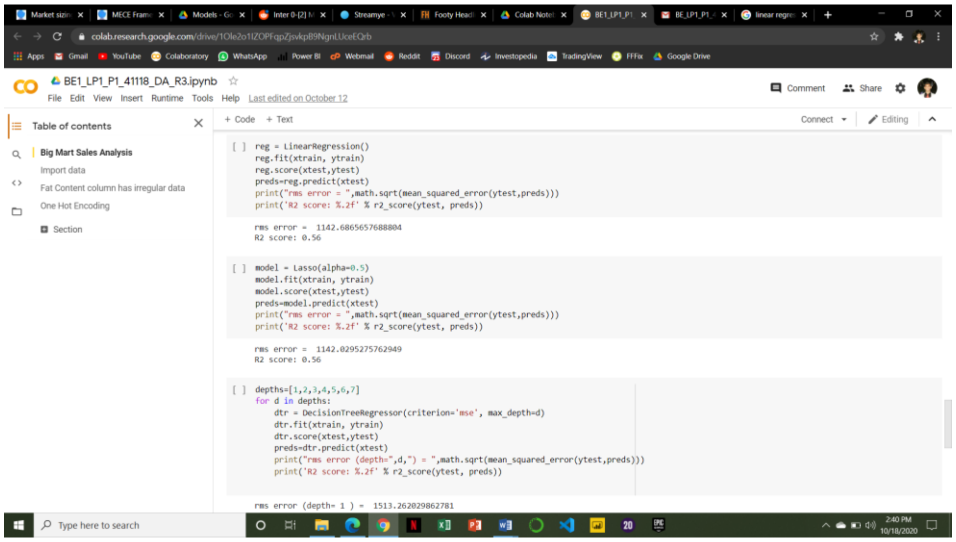


```
for d in depths:
    dtr = DecisionTreeRegressor(criterion='mse', max_depth=d)
    dtr.fit(xtrain, ytrain)
    dtr.score(xtest,ytest)
    preds=dtr.predict(xtest)
    print("rms error (depth=",d,") = ",math.sqrt(mean_squared_error(ytest,preds)))
    print('R2 score: %.2f' % r2_score(ytest, preds))
```

```
rms error (depth= 1 ) =  1513.262029862781
R2 score: 0.24
rms error (depth= 2 ) =  1284.6128570163416
R2 score: 0.45
rms error (depth= 3 ) =  1183.6971610287283
R2 score: 0.53
rms error (depth= 4 ) =  1115.654158683325
R2 score: 0.58
rms error (depth= 5 ) =  1100.389475745765
R2 score: 0.60
rms error (depth= 6 ) =  1095.967872301307
R2 score: 0.60
rms error (depth= 7 ) =  1114.3007726385044
R2 score: 0.59
```

```
rfr = RandomForestRegressor(n_estimators=200,max_depth=6)
rfr.fit(xtrain, ytrain)
rfr.score(xtest,ytest)
preds=rfr.predict(xtest)
print("rms error = ",math.sqrt(mean_squared_error(ytest,preds)))
print('R2 score: %.2f' % r2_score(ytest, preds))
```

```
rms error =  1088.3428175374975
R2 score: 0.60
```



```
reg = LinearRegression()
reg.fit(xtrain, ytrain)
reg.score(xtest,ytest)
preds=reg.predict(xtest)
print("rms error = ",math.sqrt(mean_squared_error(ytest,preds)))
print('R2 score: %.2f' % r2_score(ytest, preds))
```

```
rms error =  1142.6865657688804
R2 score: 0.56
```

```
model = Lasso(alpha=0.5)
model.fit(xtrain, ytrain)
model.score(xtest,ytest)
preds=model.predict(xtest)
print("rms error = ",math.sqrt(mean_squared_error(ytest,preds)))
print('R2 score: %.2f' % r2_score(ytest, preds))
```

```
rms error =  1142.0295275762949
R2 score: 0.56
```

```
depths=[1,2,3,4,5,6,7]
for d in depths:
    dtr = DecisionTreeRegressor(criterion='mse', max_depth=d)
    dtr.fit(xtrain, ytrain)
    dtr.score(xtest,ytest)
    preds=dtr.predict(xtest)
    print("rms error (depth=",d,") = ",math.sqrt(mean_squared_error(ytest,preds)))
    print('R2 score: %.2f' % r2_score(ytest, preds))
```

```
rms error (depth= 1 ) =  1513.262029862781
```

Results:

Algorithm	Accuracy(R2)
Linear Regression	56%
Lasso Regression	56%
Decision Tree	60%
Random Forest	60%

Notebook Link:

https://colab.research.google.com/drive/1Ole2o1IZOPFqpZjsvkpB9NgnLUceEQrb?usp=sharing

Conclusion:

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