**Table of Contents**

**1.**Importing Necessary Dependencies

**2.**Understanding the data

**3.**Read and Load the Dataset

**4.**Exploratory Data Analysis(EDA)

**5.**Splitting of Data into Training and Testing Subset

**6.**Training the Model using Regression Algorithms

**7.** Predictions on Test Data

**8.** Evaluating the Model

**9.**Conclusion

### **1.Importing Necessary Dependencies**

In this step, we will import the necessary libraries such as:

* **For Linear Algebra:** Numpy
* **For Data Preprocessing, and CSV File I/O:**Pandas
* **For Model Building and Evaluation:** Scikit-Learn
* **For Data Visualization:**Matplotlib, and Seaborn, etc.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

**2.Understanding the data**

The given data is having different types of Data .They are Categorical and Numerical data types

Status and Item type are categorical data type and remaining all columns are Numerical data type

### **3.Read and Load the Dataset**

In this step, we will read and load the dataset using some basic function of pandas such as

* **For Load the CSV file:**pd.read\_csv( )
* **To print some initial rows of the dataset:**df.head( )
* **Statistical Details for Numerical Columns:**df.describe( )
* **Basic Information About the dataset:**df.info ( )

**Load the Dataset:**

df = pd.read\_excel("daily\_offers.xlsx")

**Shape of the data:**

df.shape

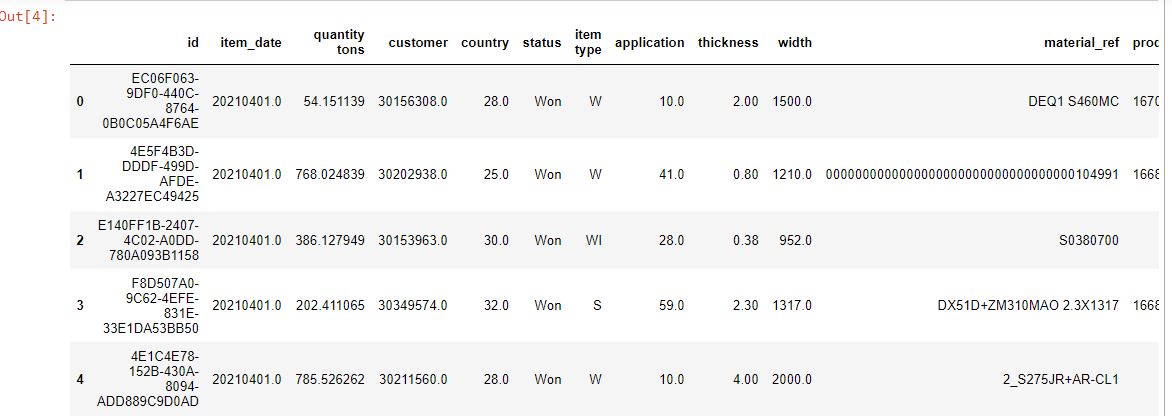
(181673, 14)

No of records – 181673

No of columns - 14

**Print some initial rows of the dataset:**

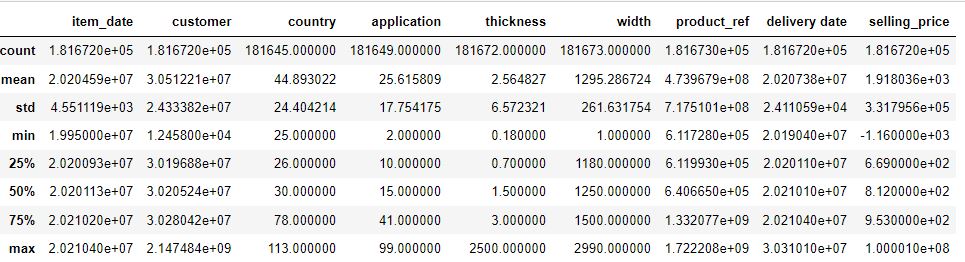
df.head(5)



**Statistical Details for Numerical Columns:**

df.describe()

**Output:**

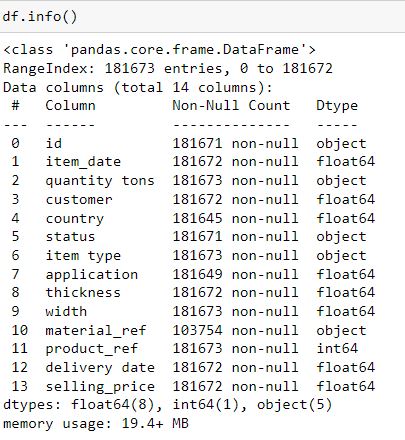


**4. Exploratory Data Analysis(EDA):**

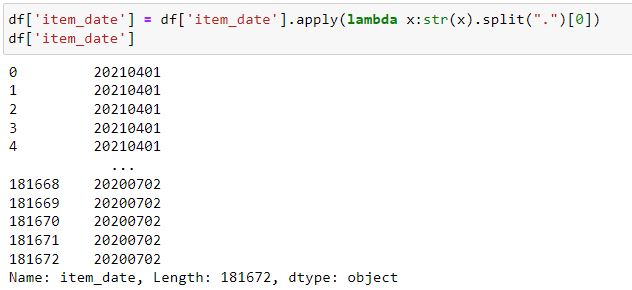
**Data Cleaning :**

I observedthat for Item\_date and delivery\_date data type is wrong and also for few columns data type is wrong. I had converted the data types into datetime format using pd.to\_datetime .

df.info() --🡪it will give the information of the all the columns with their data types

****

**By observing this, it is clear that item\_date and delivery date data type is wrong so I corrected using the following code. This code splits the float and take out the entire date as shown below.**

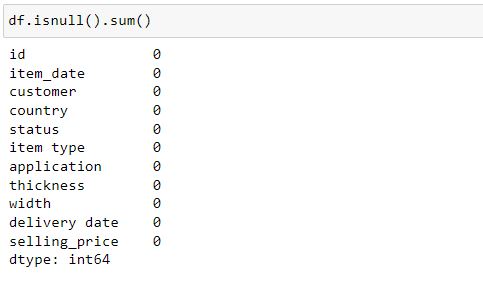
****

**Next converting the item\_date into datetime format using following code**

df['item\_date'] = pd.to\_datetime(df['item\_date'],errors='coerce')

df['item\_date']

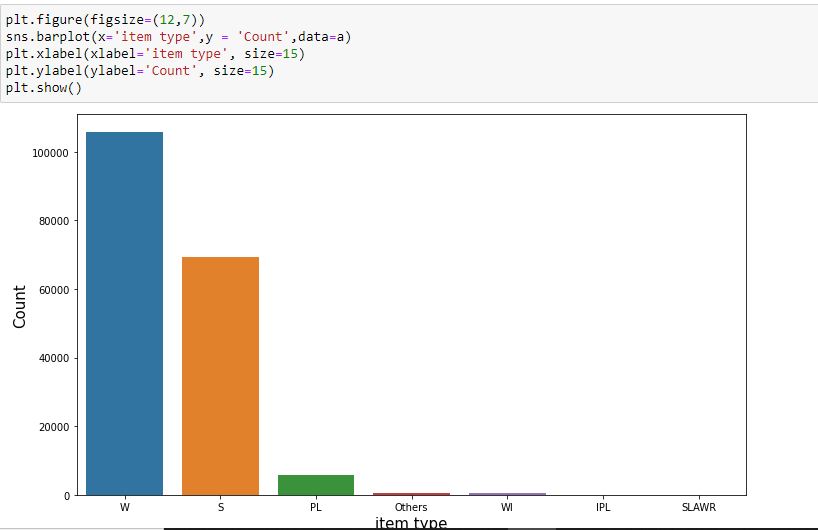
This same steps are repeated for the delivery date column and there are few null values are there in the given data columns I dropped them using dropna method as shown below



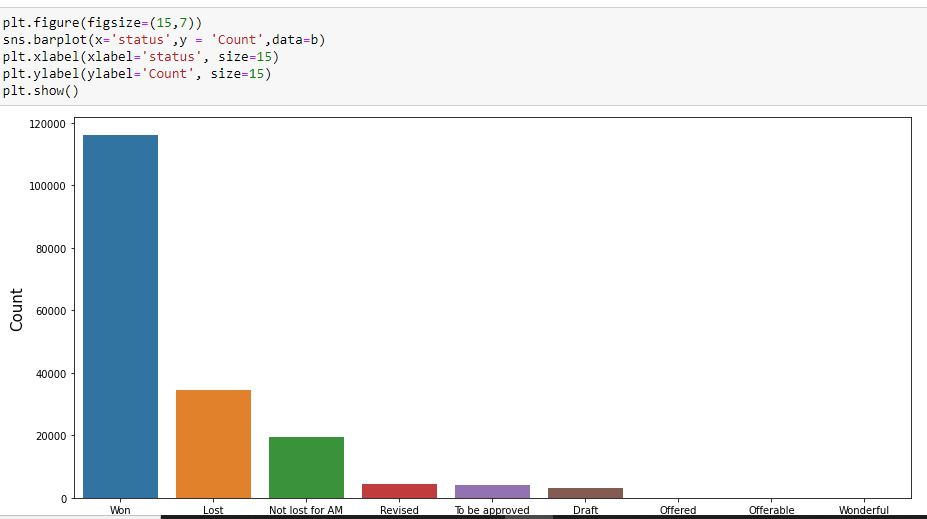
As we can see from the above image that I had removed all the null values and I checked duplicate values as well and I observed that there are no dupplicated values in the data.

**Plotted Visualization:**

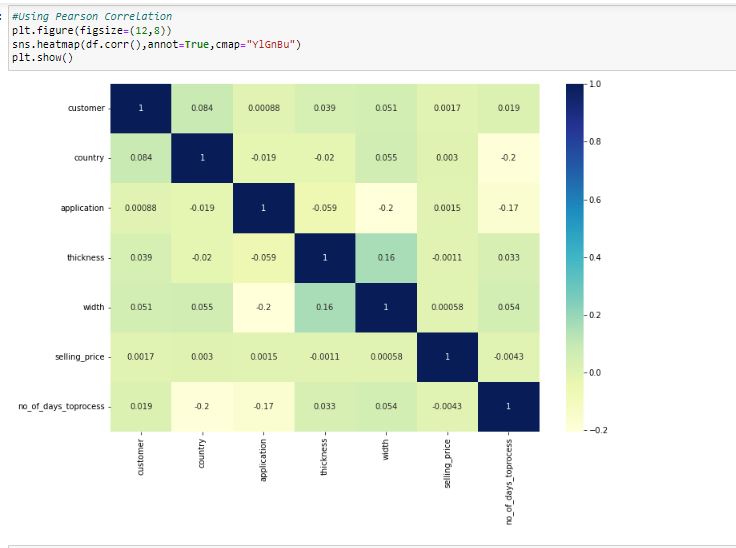
**Bar plot shows the count of each Item Type**

****

**Bar plot shows the count of each status**

****

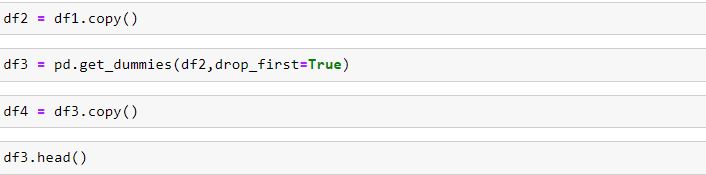
**I have plotted heatmap to check correlation between numerical variables**

****

**5. Splitting of Data into Training and Testing Subset:**

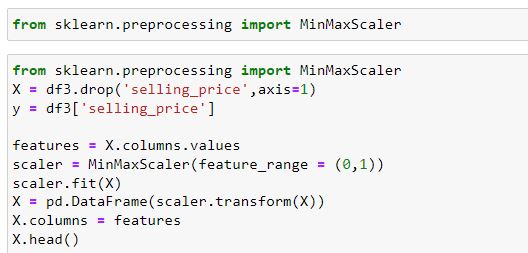
**Feature Engineering:**

Using get dummies converted all categorical data to binary 0 and 1



**Scaling the data:**

I did scaling using minmax scalar and separated dependent and independent variables



**Use model\_selection.train\_test\_split from sklearn to split the data into training and testing sets. Set test\_size=0.10 and random\_state=42:**

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.10, random\_state=42)

**6. Training the Model using DecisionTree Regression Algorithm:**

Now, at this step I am able to train the model on our training data using DecisionTree Regression.

**Import LinearRegression from sklearn.linear\_model**

from sklearn.tree import DecisionTreeRegressor

**Create an instance of a DecisionTreeRegressor() model named dtr.**

dtr= DecisionTreeRegressor()

**Train/fit dtr on the training data.**

dtr.fit(X\_train, y\_train)

**Output:**

DecisionTreeRegressor DecisionTreeRegressor(random\_state=0)

**7. Predictions on Test Data:**

Now that we have train our model, let’s evaluate its performance by doing the predictions on the unseen data.

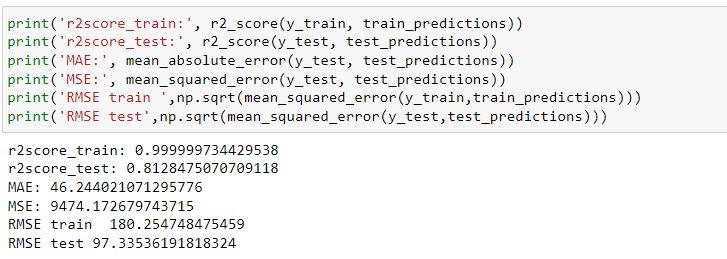
**Use dtr.predict() to predict off the X\_test set of the data.**

test\_predictions = dtr.predict(X\_test)

print("Predicted test value:", test\_predictions)

**8. Evaluating the Model:**

**Determine the metrics such as Mean Absolute Error, Mean Squared Error, and the Root Mean Squared Error.**

****

**9. Conclusion:**

I have done data cleaning ,data exploration and data analysis and find out null values and finding out corelation and scaled the data and performed the decision tree which is given me the r2 score 81% with less errors.