Advertising **Sales Channel Prediction**

**Project Description:-**

Advertising Sales Channel Prediction Problem Statement: Sales Channel Prediction Case Study

When a company enters a market, the distribution strategy and channel it uses are keys to its success in the market, as well as market know-how and customer knowledge and understanding. Because an effective distribution strategy under efficient supply-chain management opens doors for attaining competitive advantage and strong brand equity in the market, it is a component of the marketing mix that cannot be ignored .

The distribution strategy and the channel design have to be right the first time. The case study of Sales channel includes the detailed study of TV, radio and newspaper channel. The predict the total sales generated from all the sales channel.

**Analysis:-**

Analyzing and anticipating the sales for the given budget for TV, radio, and newspapers**.**  In this model, we need to feed the advertising budget of TV, radio, and newspapers to the model and the model will forecast the possible sales. For designing the model, the machine learning method I opted for is 5 regression model , and the programming was done in Juypter notebook.

**Dataset Description:**

The advertising dataset captures the sales revenue generated with respect to advertisement costs across numerous platforms like radio, TV, and newspapers. I found my data from this link

<https://raw.githubusercontent.com/dsrscientist/DSData/master/Advertising.csv>

**Data:**

**Features variable :**

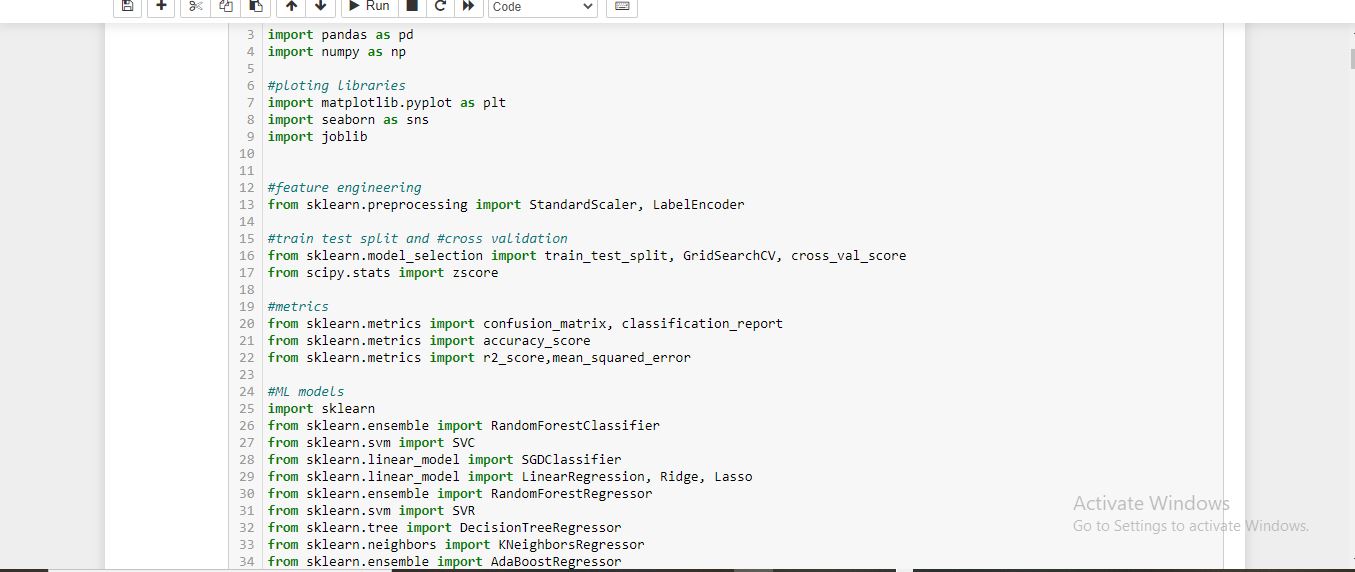
* TV: advertising dollars spent on TV.
* Radio: advertising dollars spent on Radio.
* Newspaper: advertising dollars spent on Newspaper.

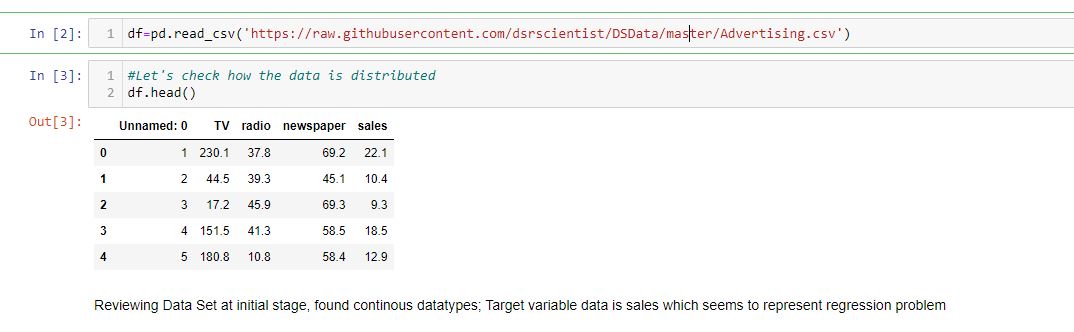
**Target variable:**

Sales budget.

**Step 1: Import the required libraries and dataset*.***

The dataset which I chose for this exercise or program is in the form of CSV so, I used **pd.read\_csv from**thepandas module as shown in the picture below dataset contains 4 columns named TV, radio, newspaper, and sales.

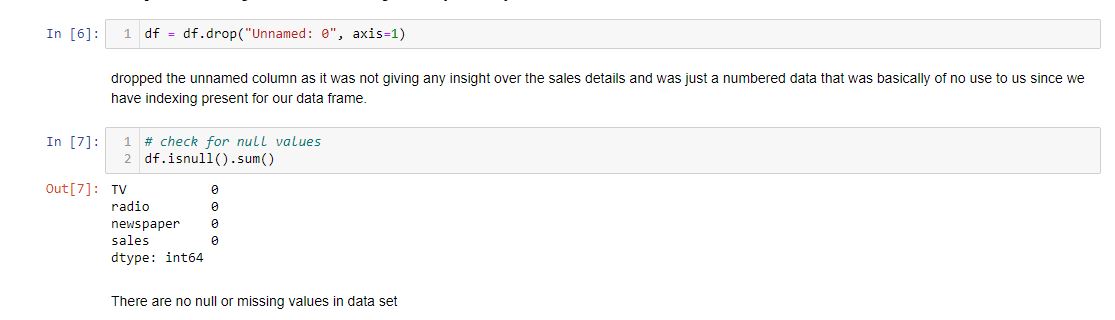




Here 4 column are present with TV, News paper, Radio, Sales. And 200 rows.

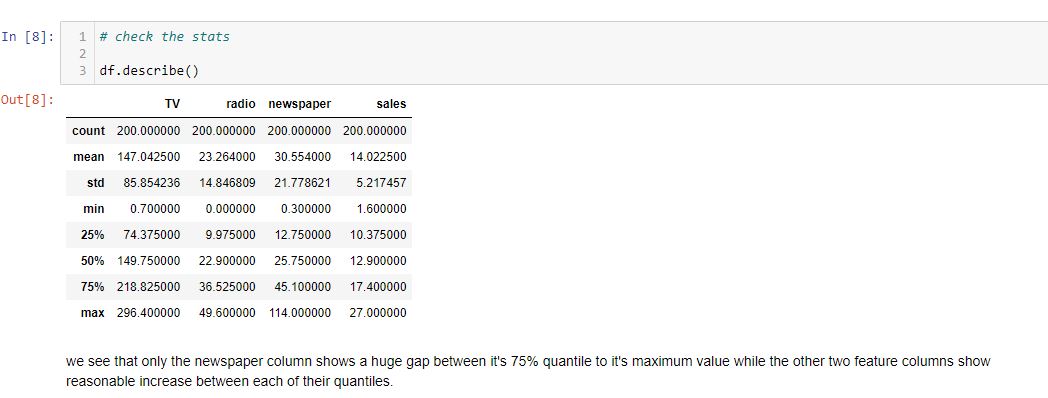
**Step 2: Check for null values in the dataset and data inspection*.***

After the extraction of data, it’s time to check the dataset for null values and duplicate values.



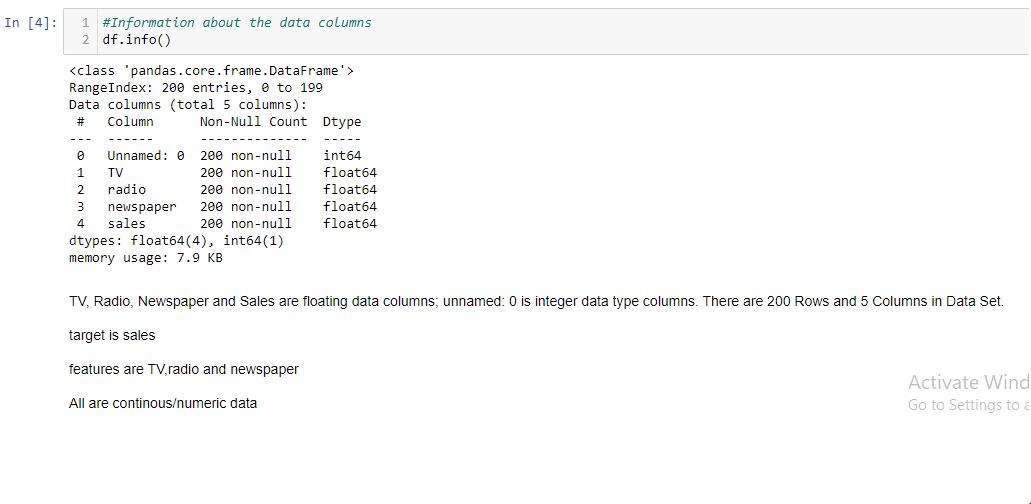
No null values are present in this dataset.it is good for me.if any null values were present then we have to apply fillna method

Data Description



Only news paper column show a huge gap b/w quantile to it’s maximum value otherwise other coloumn show resasonable increase b/w each of their quantile..

Data Information:



Here no any object type data are present. All data in int and float type .

**Step 3: Exploratory Data Analysis (EDA).**

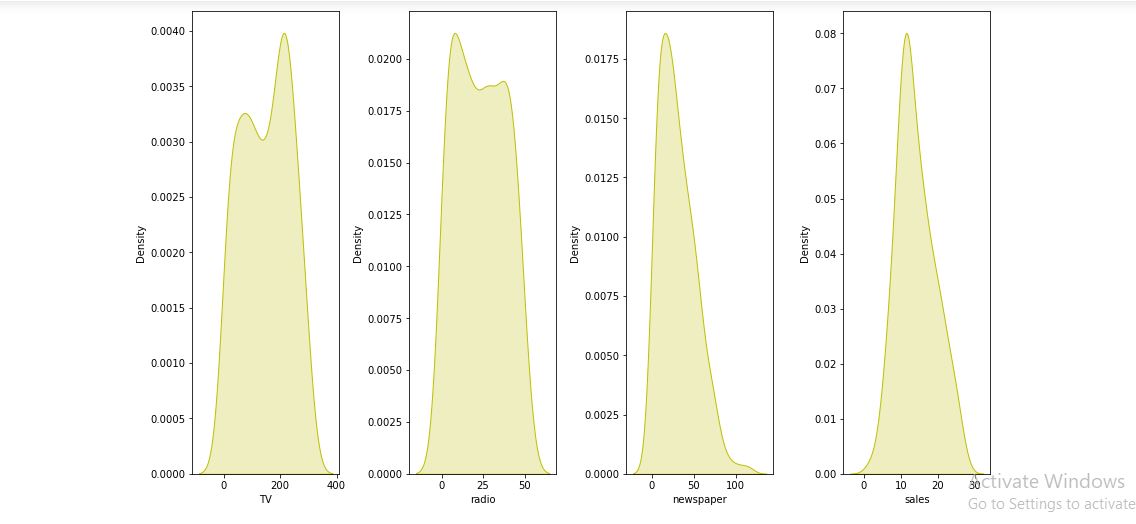
In EDA we are gonna find the relationship between features and the target variables.

Distribution plot:

The distribution plot is **suitable for comparing range and distribution for groups of numerical data**. Data is plotted as value points along an axis.

****

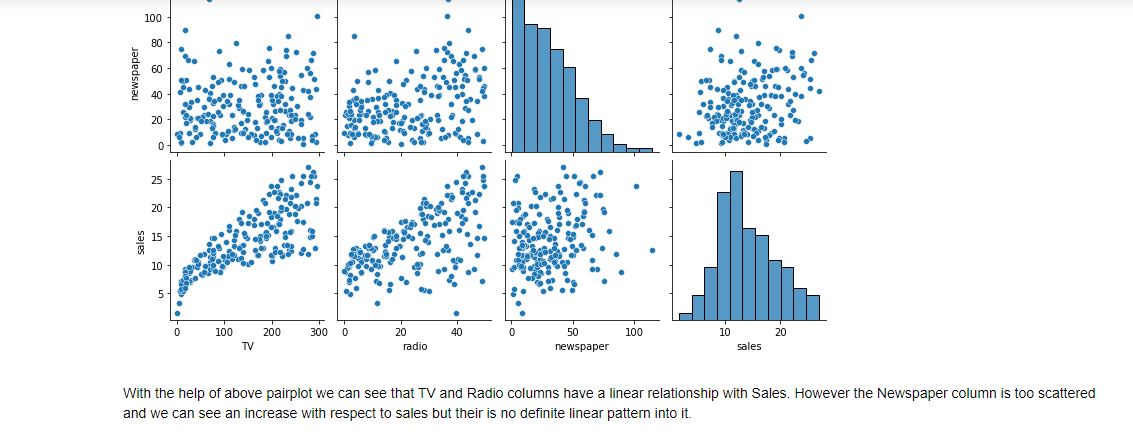
Looking at the below distribution plot we can see that almost all the columns are maintaining the bell shape curve except for the newspaper column where we can see a visible skewness.



**Pair Plot:-**

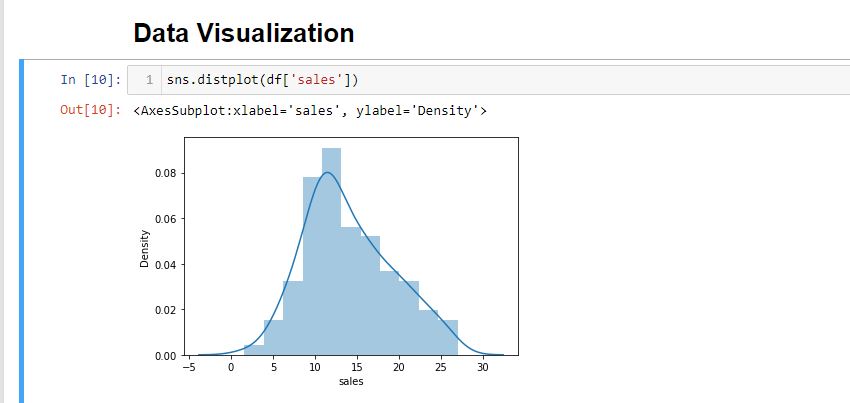
**Plot pairwise relationships in a dataset**. By default, this function will create a grid of Axes such that each numeric variable in data will by shared across the y-axes across a single row and the x-axes across a single column.

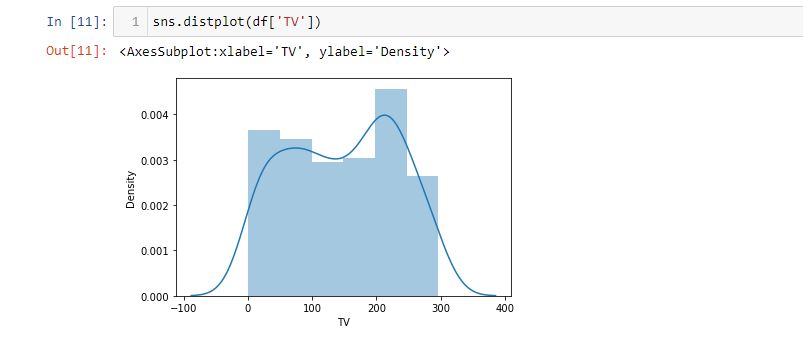


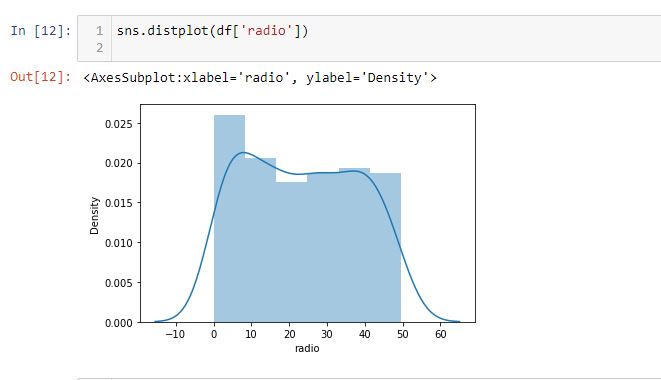


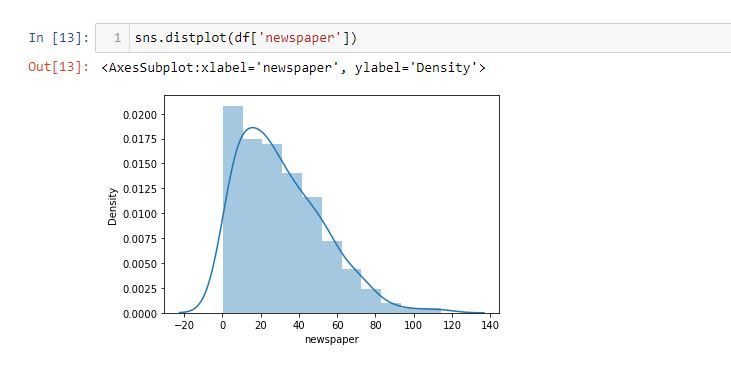
**Distplot:**

Displot is used to represent the univariate distribution of data(involving one variate or variable quantity) against the density.





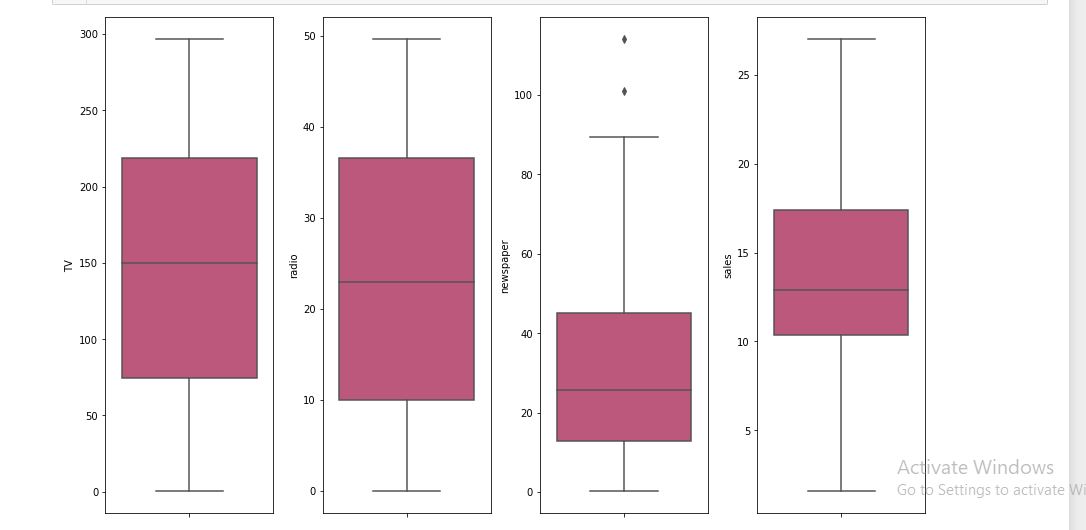




Box Plot:

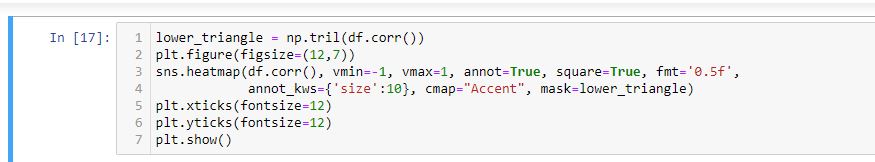
A Box Plot is also known as Whisker plot is created to display thesummary of the set of data values having properties like minimum, first quartile, median, third quartile and maximum.

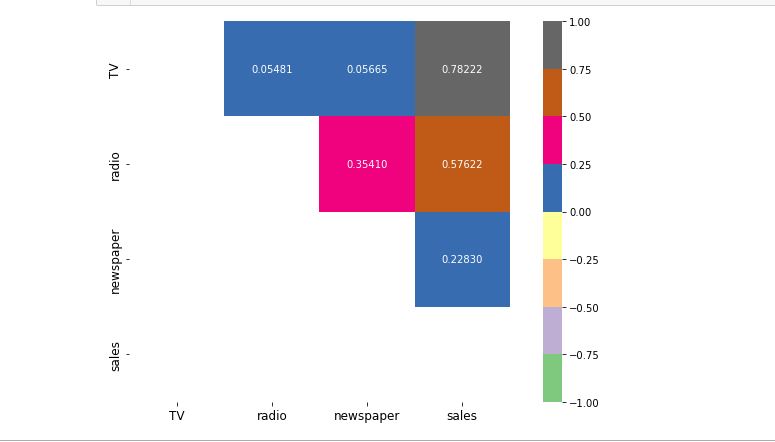




Correlation using Heat map:

Positive correlation - A correlation of +1 indicates a perfect positive correlation, meaning that both variables move in the same direction together. Negative correlation - A correlation of –1 indicates a perfect negative correlation, meaning that as one variable goes up, the other goes down.

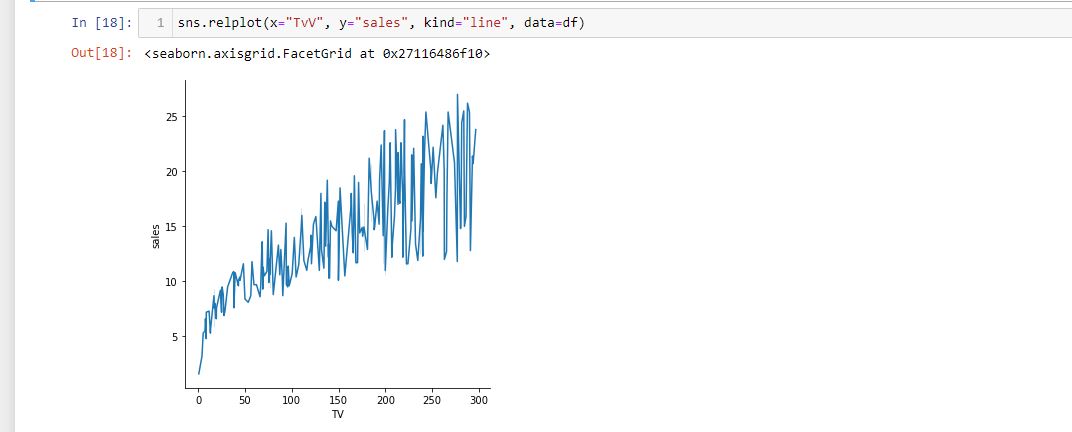




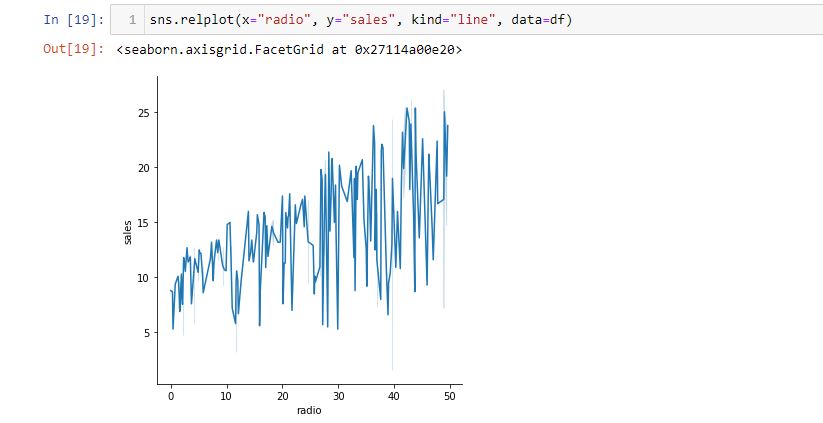
In the above heatmap we can see that there are no negative correlation present in between our features and label or between the feature columns either. Noticing the sales part we see that it has a high correlation with TV and a low correlation with Newspaper.

# Line Plots:

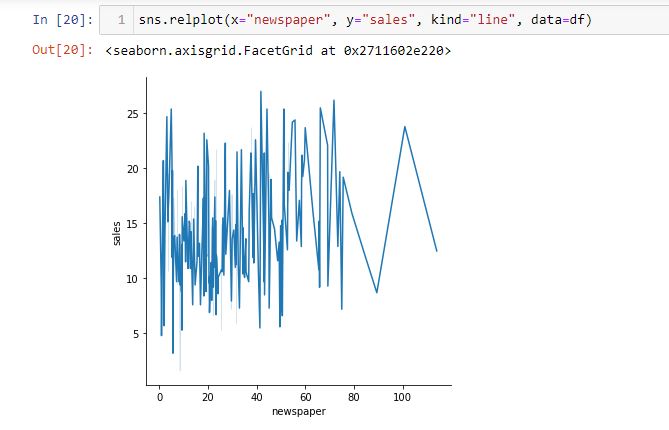
From simple to complex visualizations, it's the go-to library for most. In this tutorial, we'll take a look at how to plot a line plot in Matplotlib - one of the most basic types of plots. Line Plots display numerical values on one axis, and categorical values on the other.



 line plot for TV vs Sales shows a proper increasing pattern wherein the TV advertisements are playing a major role in the marketing field thus inreasing the sales.



The line plot for Radio vs Sales do not show a dominant relationship when it comes to sales like the TV option but it is definitely contributing towards the sales parameter.



Lastly, looking at the line plot for Newspaper vs Sales we see a very random trend where in the begin the contribution is way better than towards the end showing that the outliers reflecting in our data are mere flukes where in it contributed towards the sales from that mode of marketing.

## Model Building

### Performing Simple Linear Regression

Equation of linear regression  
y=c+m1x1+m2x2+...+mnxny=c+m1x1+m2x2+...+mnxn

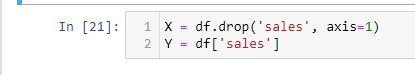
* yy is the response
* cc is the intercept
* m1m1 is the coefficient for the first feature
* mnmn is the coefficient for the nth feature

In our case:

y=c+m1×TVy=c+m1×TV

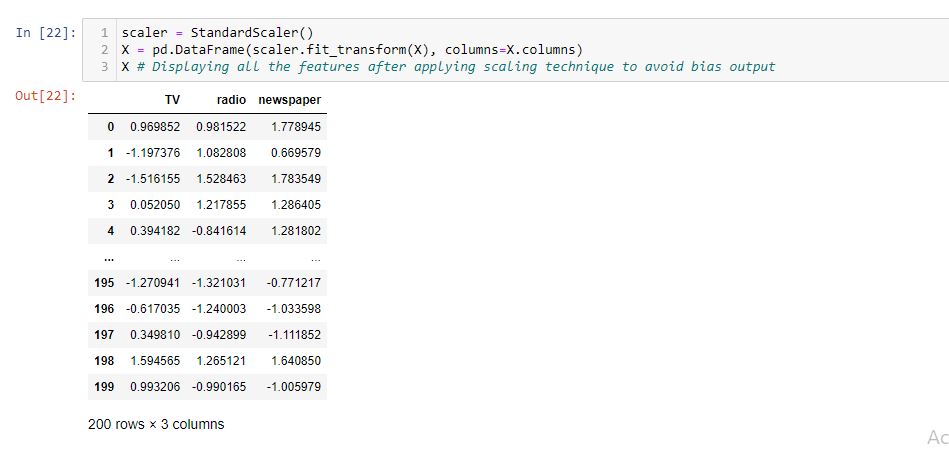
The mm values are called the model **coefficients** or **model parameters**.

# Splitting the dataset into 2 variables namely 'X' and 'Y' for feature and label



Feature Scaling:

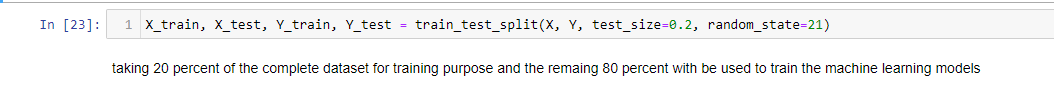
Feature Scaling or Standardization: It is a step of Data Pre Processing that is applied to independent variables or features of data. It basically helps to normalize the data within a particular range. Sometimes, it also helps in speeding up the calculations in an algorithm.



Even though all our feature columns were of float data type I was unhappy with the decimal place differences and was worried that it might make my model biased. Therefore I am using the Standard Scaler method to ensure all my feature columns have been standardized.

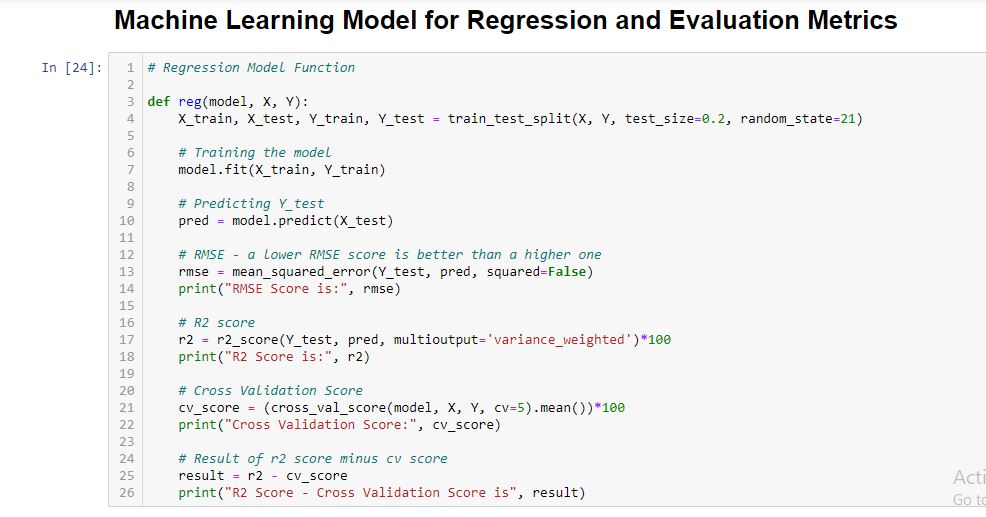
Note: Trying to retain all the original data therefore not treating the outliers to ensure I do not have any data loss for my model creation.

# Creating the training and testing data sets

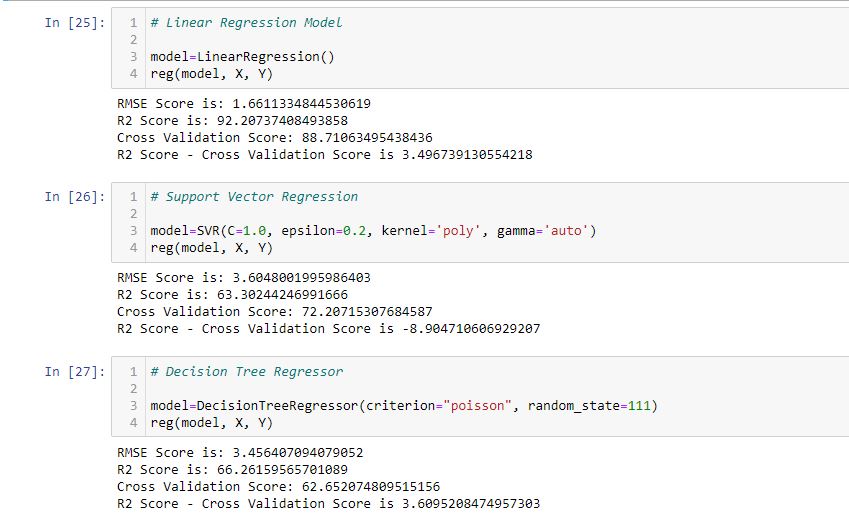


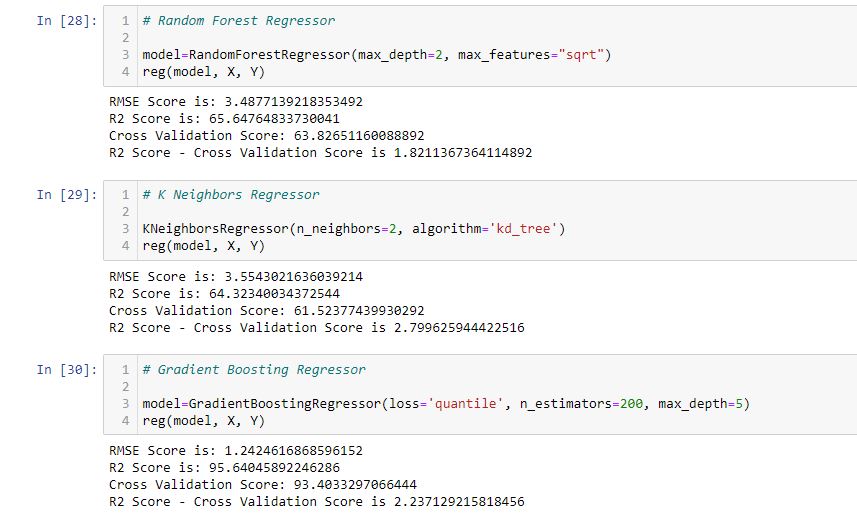
**X** is the independent variable and **Y** is the target variable (output). In **train\_test\_split**, **test\_size** indicates the size of the dataset for testing. The value of 0.2 means that the data is divided into 80% for training and 20% for testing. **random\_state**to randomize the training and test data.

The **train\_test\_split()** function returns four values in an array of **X** for training and testing and **y** for training and testing.



I have built a function that splits the training and testing features and labels, then trains the model, predicts the label, calculates the RMSE score, generates the R2 score, calculates the Cross Validation score and finally finds the difference between the R2 score and Cross Validation score.

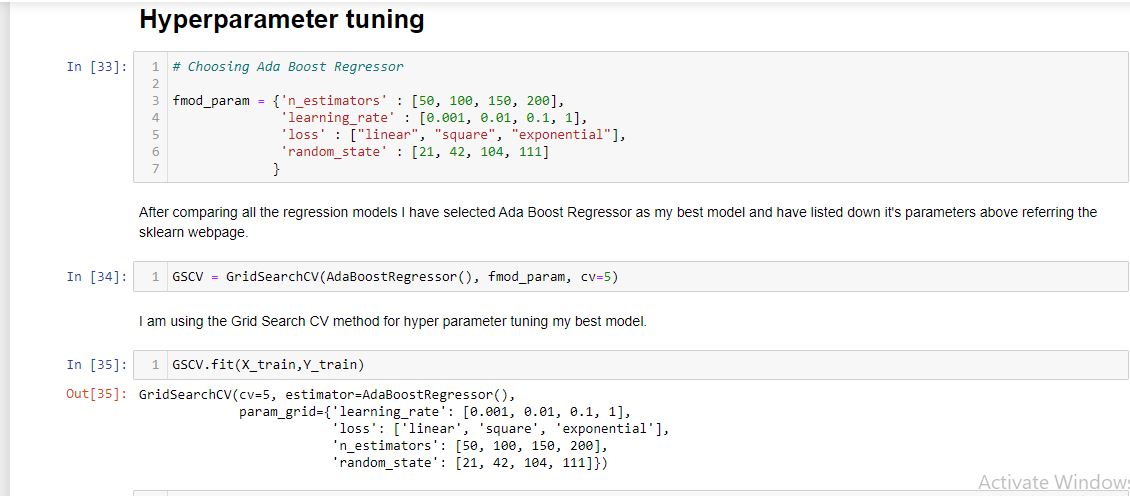






Hyperparameter Tuning

In machine learning, hyperparameter optimization or tuning is the problem ofchoosing a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a parameter whose value is used to control the learning process. By contrast, the values of other parameters (typically node weights) are learned.



After comparing all the regression models I have selected Ada Boost Regressor as my best model and have listed down it's parameters above referring the sklearn webpage.



# Saving and Loading Model Files

When we work with machine learning, it is important to save the models we have trained previously into a file. Model files can be reused to make predictions instead of having to train a new model.

To save the model, we can use the package joblib which is already available in Python.

For example, here we will save the multiple linear regression model that we have created in a file called **linear\_ads.sav** (we can save it in the models folder):



# ****Conclusion:****

# If you are doing business and are not into tv advertising, you are losing business. It is the most dependable and traditional way of advertising. TV advertising on channels appeals to the masses at large.

In a nutshell, TV advertising is the best for sales prediction. It’s a good starting point, especially when attempting to understand the relevance of python as well as statistics.