**Advertising Sales**

**Channel Prediction-**

**Machine Learning**

**Regression**

**By Sudha Udayakumar**

**A picture containing graphical user interface

Description automatically generated**

Table of Contents

[Sales Channel Prediction Using Machine Learning 6](#_Toc73816913)

[1. Introduction to Machine Learning 6](#_Toc73816914)

[1.1 Types of Machine Learning: - 6](#_Toc73816915)

[2. Take Away from this Article 8](#_Toc73816916)

[3. Assumptions 8](#_Toc73816917)

[4. 7 Stages of Machine Learning 8](#_Toc73816918)

[4.1 Understanding the Problem 9](#_Toc73816919)

[4.1.1 Loading the Dataset from the source 9](#_Toc73816920)

[4.1.2 Summary of the Data source, Problem Statement 9](#_Toc73816921)

[4.1.3 Looking at the Data 9](#_Toc73816922)

[4.2 Initial Analysis of the Data 10](#_Toc73816923)

[4.2.1 Cleaning any Text 10](#_Toc73816924)

[4.2.2 Null & Unique Values 10](#_Toc73816925)

[4.2.3 Type of columns 11](#_Toc73816926)

[4.2.4 Statistical Info of the Dataframe 11](#_Toc73816927)

[4.3 Exploratory Data Analysis 12](#_Toc73816928)

[4.3.1 Distribution of Features 13](#_Toc73816929)

[4.3.2 Is there a relationship between sales and spend various advertising channels? 14](#_Toc73816930)

[4.3.2 Co-relation- Heatmap 14](#_Toc73816931)

[4.4 Data preparation for Model Building 15](#_Toc73816932)

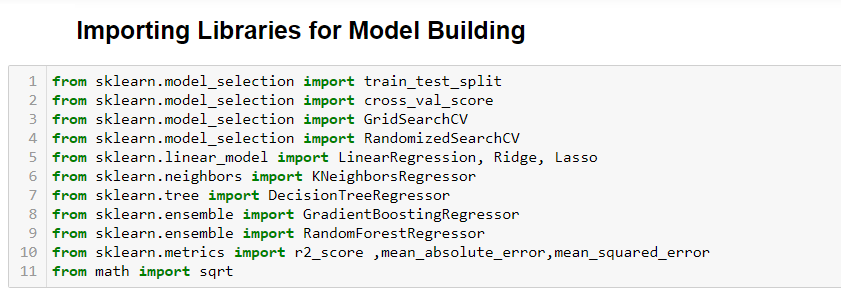
[4.4.1 Dropping unwanted columns 15](#_Toc73816933)

[4.4.2 Handling Outliers 15](#_Toc73816934)

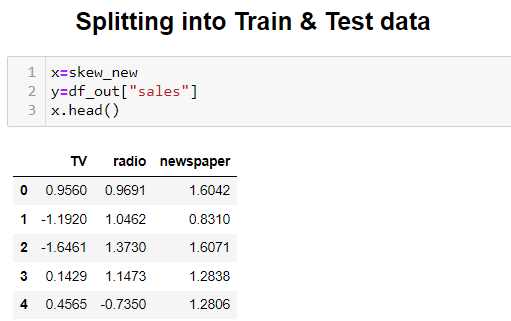
[4.4.3 Skewness Handling 18](#_Toc73816935)

[4.5 Model building 19](#_Toc73816936)

[4.5.1 Importing the required libraries for Model Building and Evaluation 19](#_Toc73816937)

[ 19](#_Toc73816938)

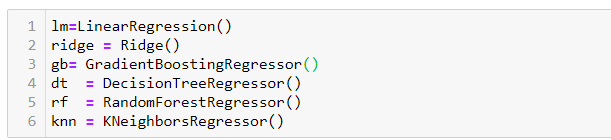
[4.5.2 Splitting our data into X & Y 19](#_Toc73816939)

[ 19](#_Toc73816940)

[4.5.3 Finding Best Random State 19](#_Toc73816941)

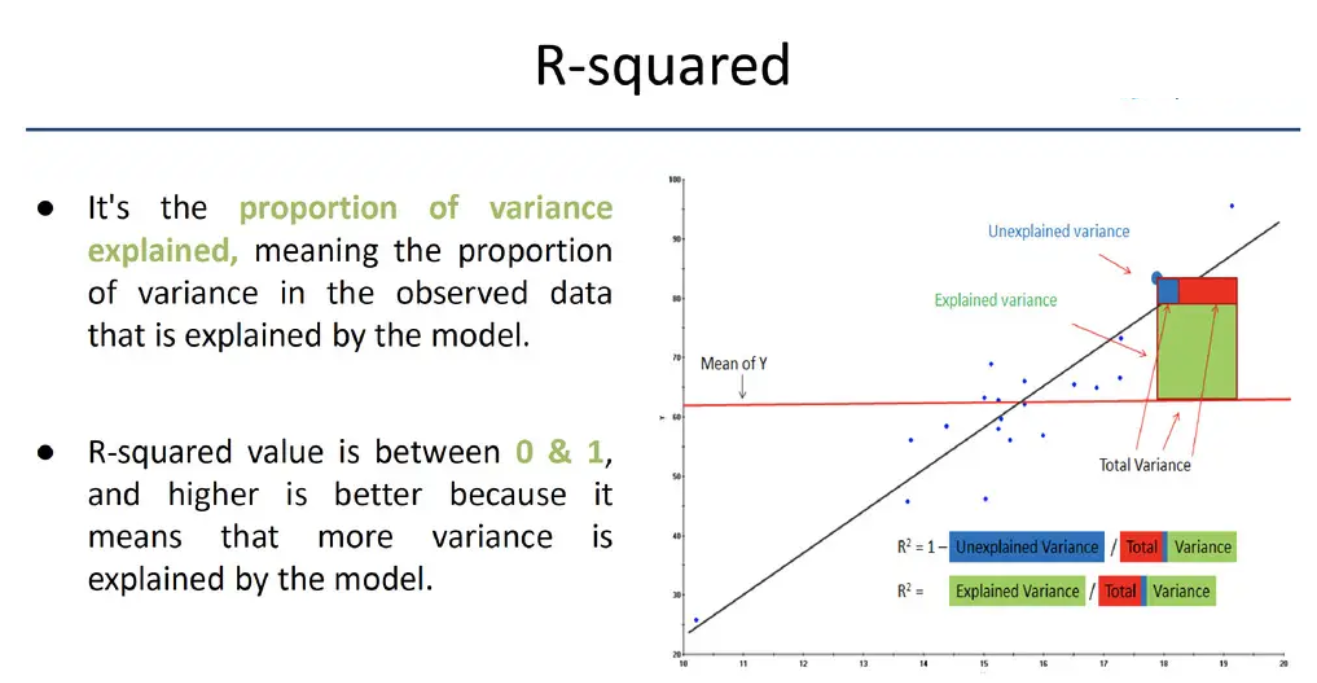
[ 20](#_Toc73816942)

[4.5.4 Instantiating the Algos 20](#_Toc73816943)

[ 20](#_Toc73816944)

[4.6 Evaluation Metrics 20](#_Toc73816945)

[4.6.1 R2 20](#_Toc73816946)

[ 21](#_Toc73816947)

[4.6.2 Cross Validation 21](#_Toc73816948)

[4.7 Improving the Model 22](#_Toc73816949)

[4.7.1. Random Forest 22](#_Toc73816950)

[4.7.2 Random Forest 2 22](#_Toc73816951)

[4.7.3 Gradient Boost Regressor 23](#_Toc73816952)

[5. Saving & Loading the Model 24](#_Toc73816953)

[5.1 Saving the Model 24](#_Toc73816954)

[5.2 Loading the Model 24](#_Toc73816955)

[6. Conclusion 24](#_Toc73816956)

# Sales Channel Prediction Using Machine Learning

## Introduction to Machine Learning

Data is growing day by day, and it is impossible to understand all of the data with higher speed and higher accuracy. More than 80% of the data is unstructured, that is audios, videos, photos, documents, graphs, etc. Finding patterns in data on planet earth is impossible for human brains. The data has been very massive, the time taken to compute would increase and this is where Machine Learning comes into action, to help people with significant data in minimum time. Machine Learning is a sub-field of AI. Applying AI, we wanted to build better and intelligent machines. It sounds similar to a new child learning from itself. So, in machine learning, a new capability for computers was developed and now machine learning is present in so many segments of technology, that we don’t even realize it while using it.

### 1.1 Types of Machine Learning: -

Machine Learning mainly divided into three categories

Diagram, PowerPoint

Description automatically generated

In this Article, lets focus on the Supervised Learning

Diagram

Description automatically generated

**A picture containing chart

Description automatically generated**

## Take Away from this Article

1. Handling a Regression Dataset
2. Exploratory Data Analysis
3. Preparing the data before Modelling
4. Evaluation Metrics used in Machine Learning -Regression Dataset
5. Several Tips, to apply for future datasets

## Assumptions

We assume that you have a basic knowledge of Python, Data Science, Machine Learning and have a familiarity with the Sckit learn and know to load the libraries like Numpy, Pandas, Seaborn, Matplotlib for data loading, pre-processing and EDA. Also have a basic understanding of the various Regression based algorithms

## 7 Stages of Machine Learning

The stages of Machine Learning are defined different by every user. We are going to follow these steps, in our article.

1. [Problem understanding](#_Understanding_the_Problem) -Understanding the problem, the various column names and type of data
2. [Initial Analysis of the Data](#_4.2_Pre-Processing_the) – Understand the type of data, check for missing data, type of data (numerical, categorical, Date) etc
3. Dataset understanding using [Exploratory Data Analysis](#_Exploratory_Data_Analysis) (EDA)
4. [Data preparation for Model Building](#_Data_preparation_for) - (We will be checking for outliers, skewness, Scaling and remove unwanted columns etc)
5. [Model building](#_4.5_Model_building) (Use various Algorithms and build)
6. [Model evaluation](#_Evaluation_Metrics) (Use different Evaluation metrics and finalize which one we are going to use)
7. [Improve Model](#_4.7_Improving_the) (There is always, scope for improvement!)

### Understanding the Problem

#### 4.1.1 Loading the Dataset from the source

Data Source: [Link](https://github.com/dsrscientist/DSData/blob/master/Advertising.csv)

#### 4.1.2 Summary of the Data source, Problem Statement

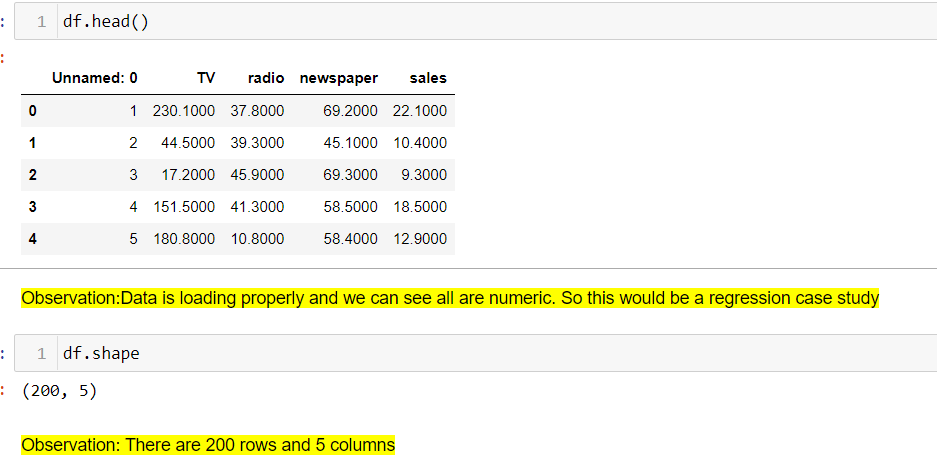
**Sales** (in thousands of units) for a particular product as a **function** of **advertising budgets** (in thousands of dollars) for *TV, radio, and newspaper media*.

* We want to find a function that given input budgets for TV, radio and newspaper **predicts the output sales**.
* Which media **contribute** to sales?
* Visualize the **relationship** between the *features* and the *response* using plots.

#### 4.1.3 Looking at the Data

We have loaded the data and df.head() gives us the top 5 rows of our Dataframe.

We also get the size of the data using df.shape. We have 200 rows and 5 columns



### Initial Analysis of the Data

Before we start analysing our data, we need pre-process the data.

Pre-processing of data contains several methods; we will see few of them which are significant in our case study.

#### 4.2.1 Cleaning any Text

We see all our data is numeric, and so there is no “Cleaning” of text that is needed in the column names or column Values

#### 4.2.2 Null & Unique Values

As we see in the Observation,

1. “Unanamed: 0” looks like a serial number. So, we can drop the column. All other columns have very high values. The target Variable “Sales” is also continuous so a regression problem
2. there are no missing values.

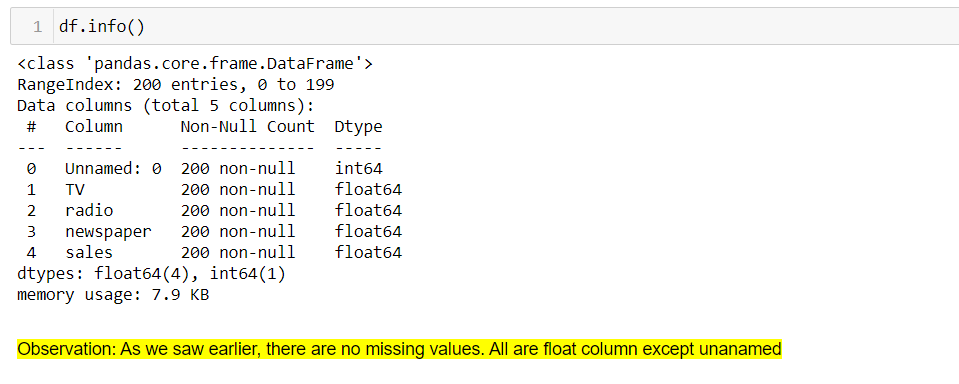
Tip 1:

After every code execution, we write an Observation. So, we need not study each line of code again. This helps when several people are working on the same dataset. Also, even for individuals when they get back to the data, they can re-collect more easily

Tip 2: df.nunique()

* If the nunique value is 1, we can drop the column. As it is not going to add any value to the data
* If the nunique value is equal to the number of rows (here 200), we can drop that column as well. As it will not add any value to our model building.
* Note: Any data that is been dropped have to be consulted with Biz. However, these are general tips given, to improve the model!

#### 4.2.3 Type of columns

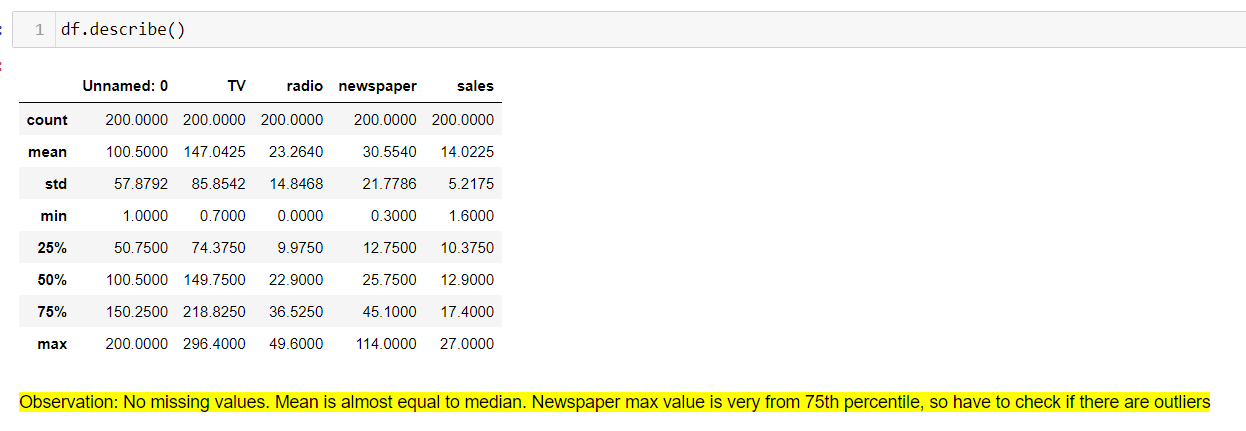


The data is numeric in nature, as we expected. It looks good and nothing else needs to be done here!

Tip 3: df.info()

* Gives you a good detail about the column type. If you saw all numbers while doing df.head(), but here if the column is listed as Object, then it means there is special character like present. It has to be handled
* If you data has data column, and it is listed as Object here, then we need to convert the column to datatime and handle it. After conversion, we can split to year/month/day or use it as is

#### 4.2.4 Statistical Info of the Dataframe



Again, we confirmed no missing values. We also made a guess on the presence of the outliers !

Tip 4: df.describe()

If you think you have a numerical column, and it is not listed here, then you are sure that there are some special characters, and they have to be handled and converted to Numerical data

We get the count, std, Mean, Median (50th Percentile) and so many statistical data from this. A very useful code

### Exploratory Data Analysis

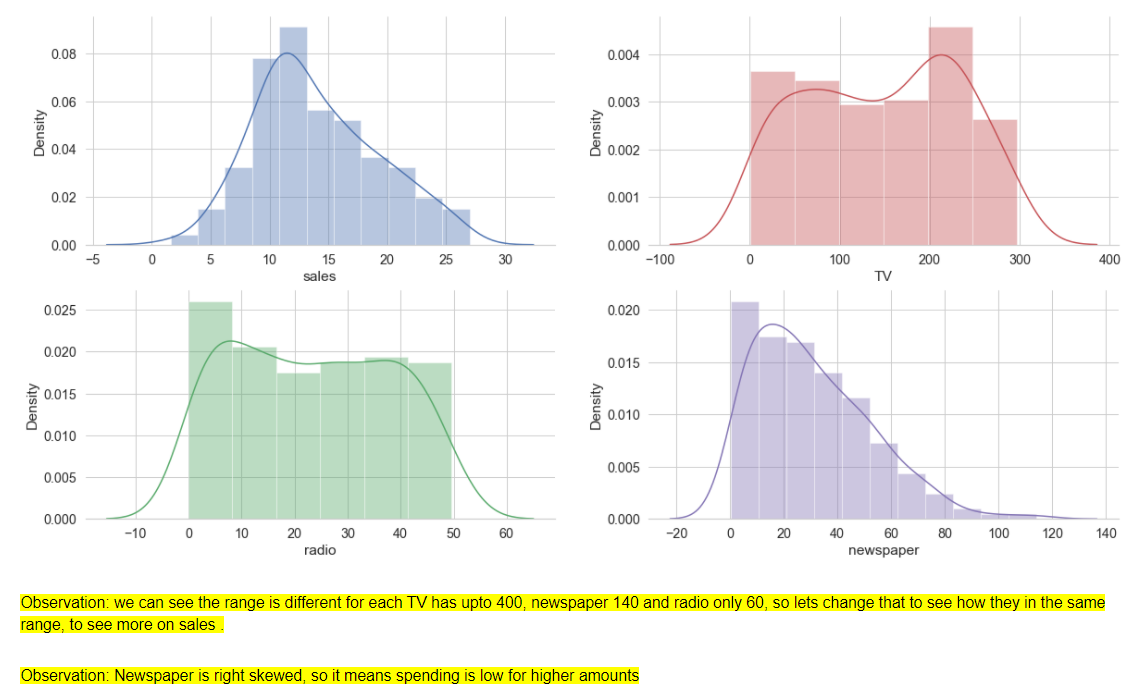
What are the **features**?

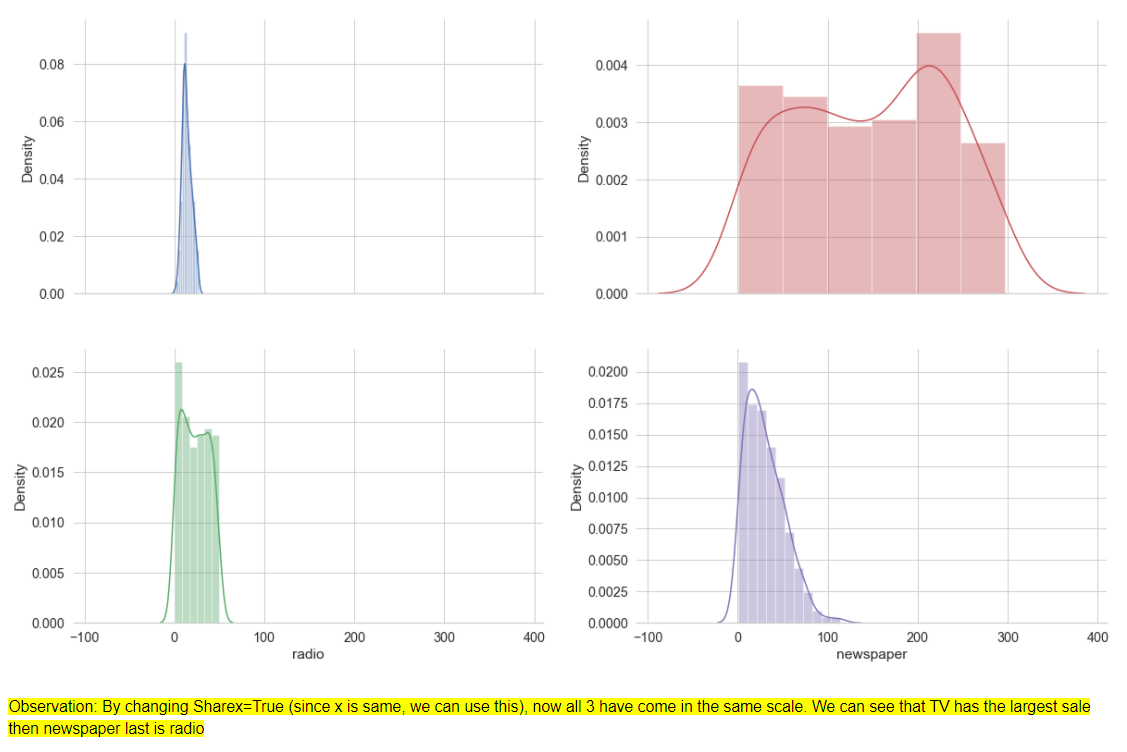
* TV: advertising dollars spent on TV for a single product in a given market (in thousands of dollars)
* Radio: advertising dollars spent on Radio
* Newspaper: advertising dollars spent on Newspaper

What is the **response**?

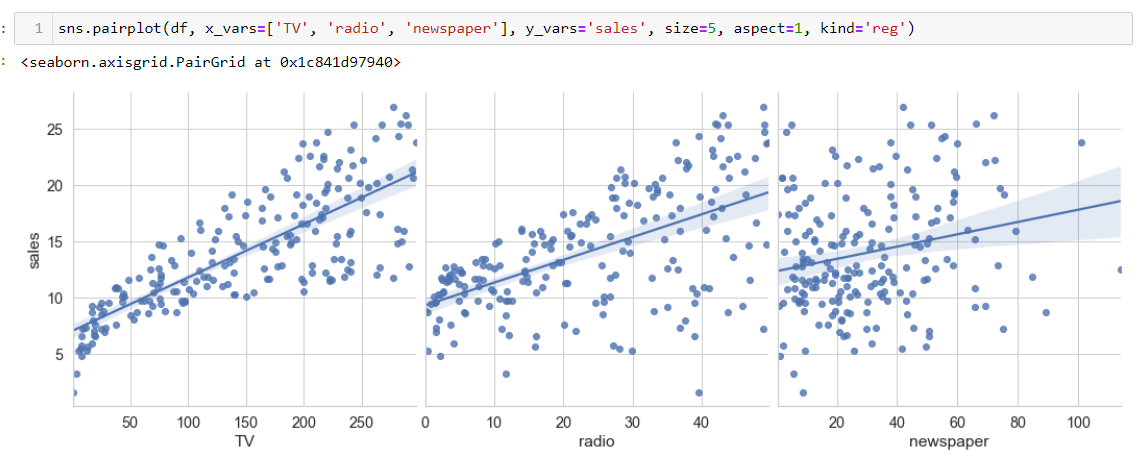
* Sales: sales of a single product in a given market (in thousands of widgets)

##### 4.3.1 Distribution of Features





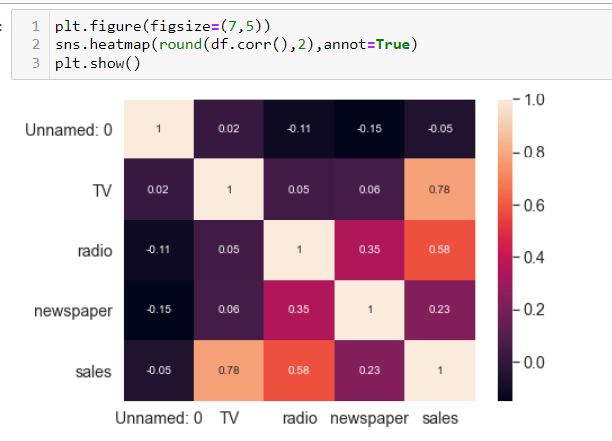
#### 4.3.2 Is there a relationship between sales and spend various advertising channels?



**Observation**

* Strong relationship between TV ads and sales
* Weak relationship between Radio ads and sales
* Very weak to no relationship between Newspaper ads and sales

#### 4.3.2 Co-relation- Heatmap



Tip 5:

Using annot=True, the numbers will also be displayed on the heatmap, which makes it more easier to read.

**Observation**

* The diagonal of the above matirx shows the auto-correlation of the variables. It is always 1. You can observe that the correlation between **TV and Sales is highest i.e. 0.78** and then between **sales and radio i.e. 0.58**.
* correlations can vary from -1 to +1. Closer to +1 means strong positive correlation and close -1 means strong negative correlation. Closer to 0 means not very strongly correlated. variables with **strong correlations** are mostly probably candidates for **model building**.

### Data preparation for Model Building

We will be doing the following

* 1. Dropping unwanted columns
  2. Check and Remove Outliers if any
  3. Check for Skewness and handle them
  4. Standard Scalar of Data

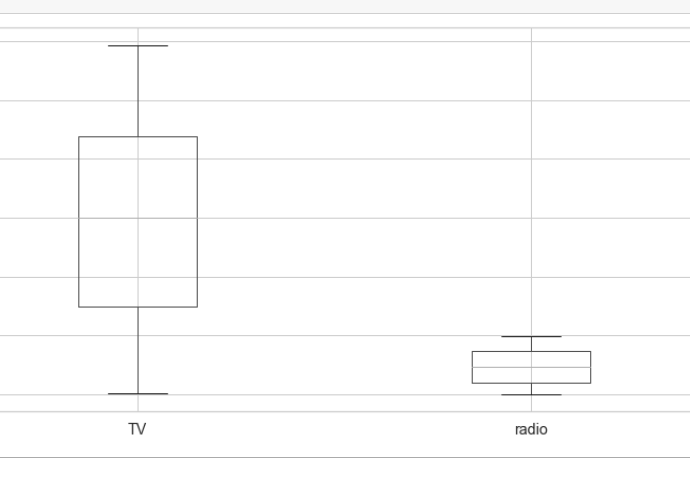
#### Dropping unwanted columns

Tip 6:

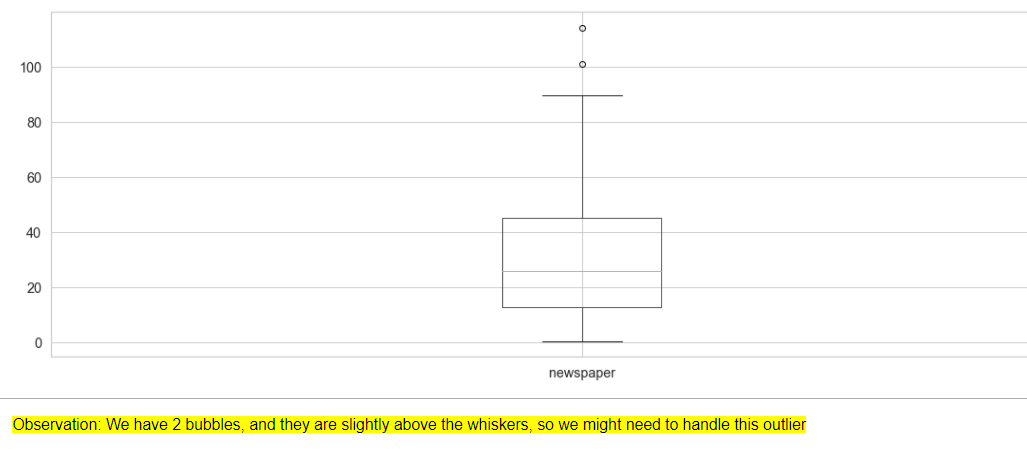
Anytime we drop a column, it’s a good practice to check if the column has been dropped. We can use any code to check like df.columns, df.info() etc. Here we have used df.head()

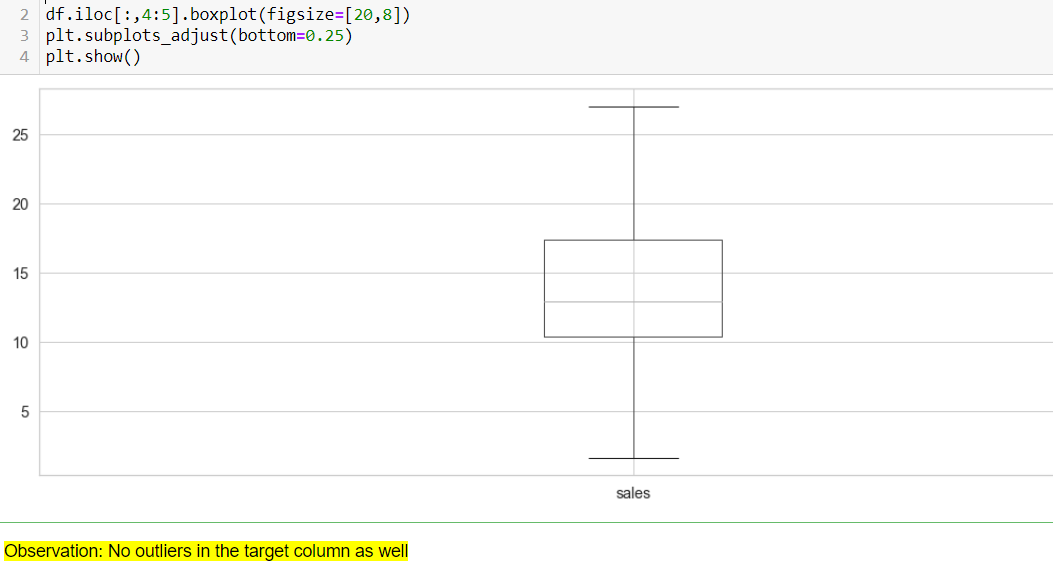
#### Handling Outliers

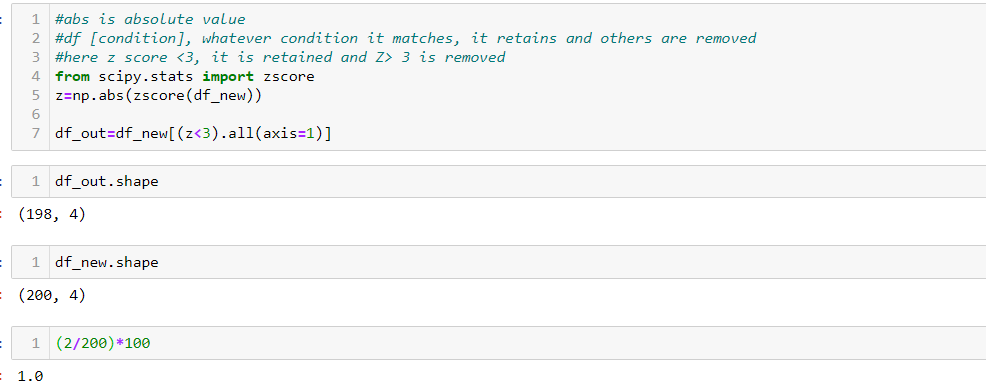
Notes: Since we are expecting to see outliers for newspaper, lets draw a separate boxplot for newspaper and others we can group and display. same way our target column, we will draw a separate box plot for "Sales"











Here we have removed the outliers, which have a Zscore above 3. When checked, we have 2 rows which have outliers. These are the same 2 bubbles we saw when we checked for Box plot. Removing 2 rows is 1% of data, so we can go ahead and remove it

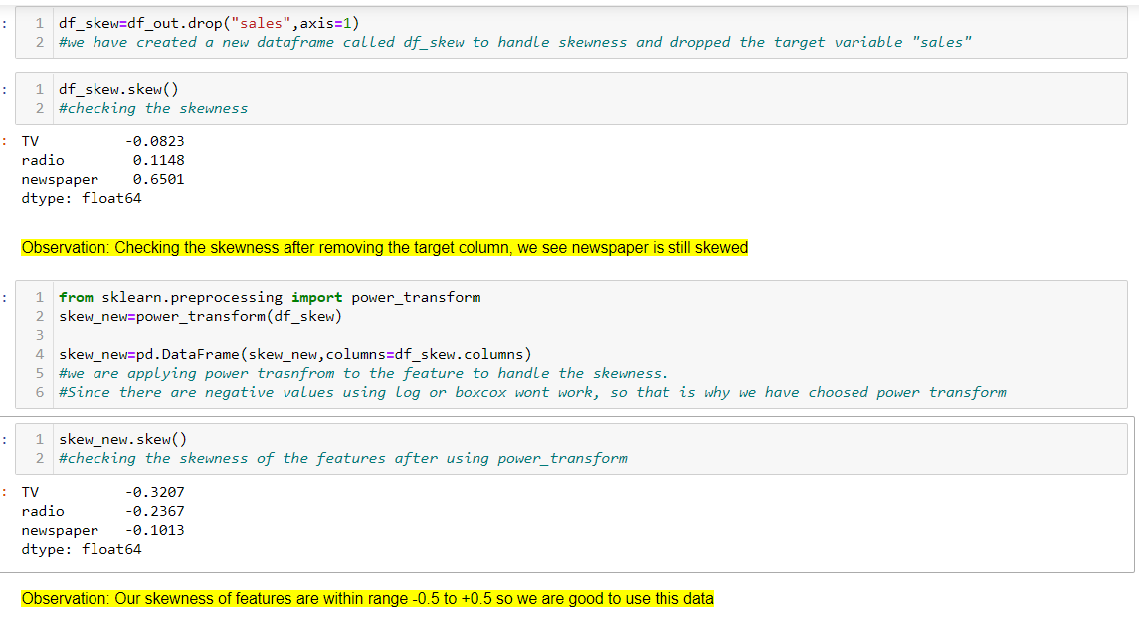
Tip 6:

1. We can check for outliers using boxplot as well as Z score.
2. We can also remove outliers using IQR (Inter Quartile Range)
3. Any removal more than 5-7% we have to check with Biz

#### 4.4.3 Skewness Handling

Tip 7:

Now we see that the skewness score is better after handling Outliers. So generally, it’s a good practice to check for outliers, if present remove them, before handling Skewness



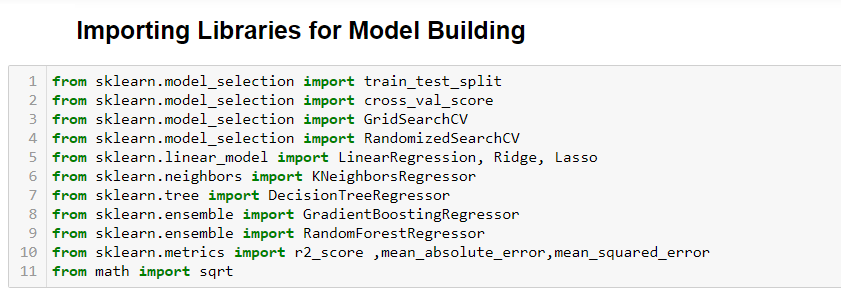
Here our Skewness is handled, and we are all set to go for Model Building

Tip 8: We can remove Skewness using multiple methods like log transform, Box Cox, Square/Cube etc.

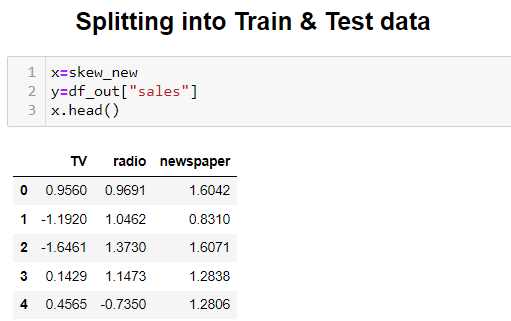
### Model building

For Model building we have the following steps. Each I have given a screen shot below

#### 4.5.1 Importing the required libraries for Model Building and Evaluation



#### 4.5.2 Splitting our data into X & Y

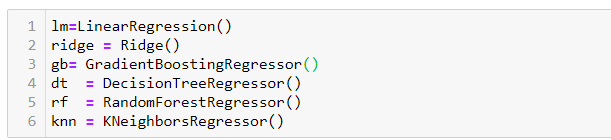


#### 4.5.3 Finding Best Random State

Below we will find the Best Random State and use the same in our Train /Test Split



4.5.4 Instantiating the Algos



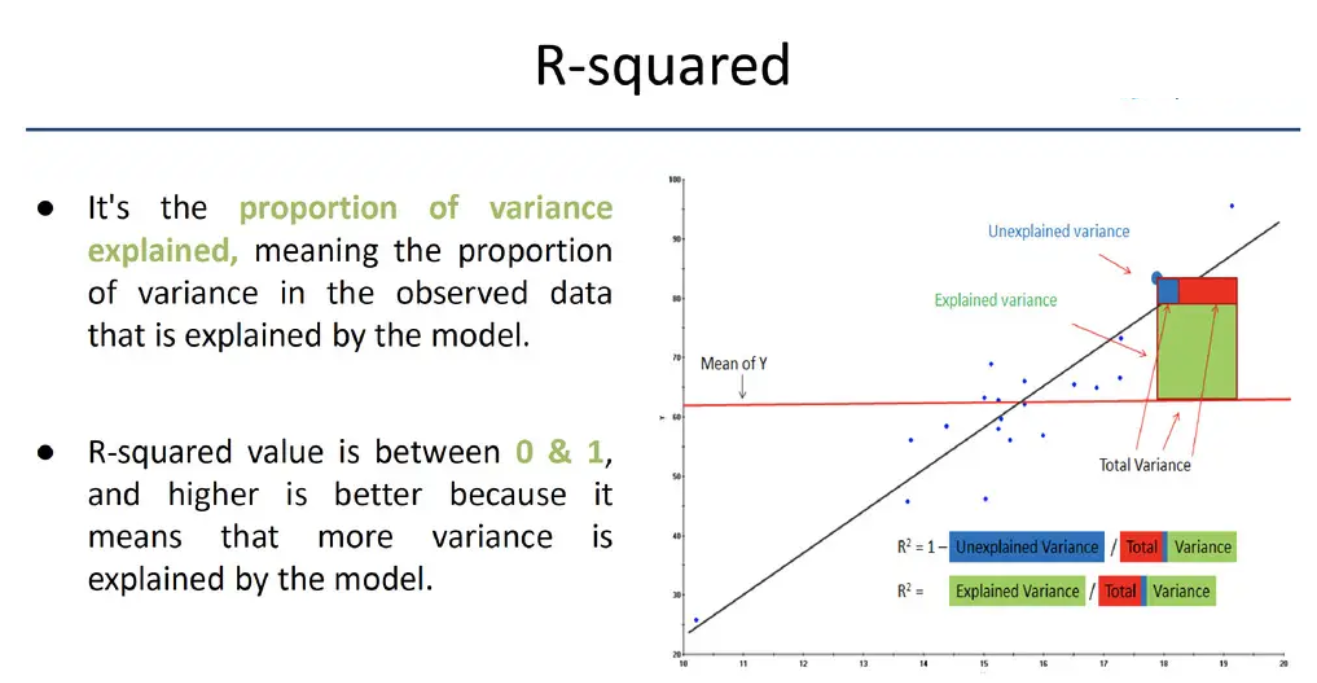
We have used a loop to run all our Algorithms & Evaluation Metrics.

### Evaluation Metrics

Though we have used to find the Error, R2, MAE, MSE, RMSE: our final evaluation is going to be based on R2.

#### 4.6.1 R2

R2 or R-Squared is the preferred metric for evaluation.

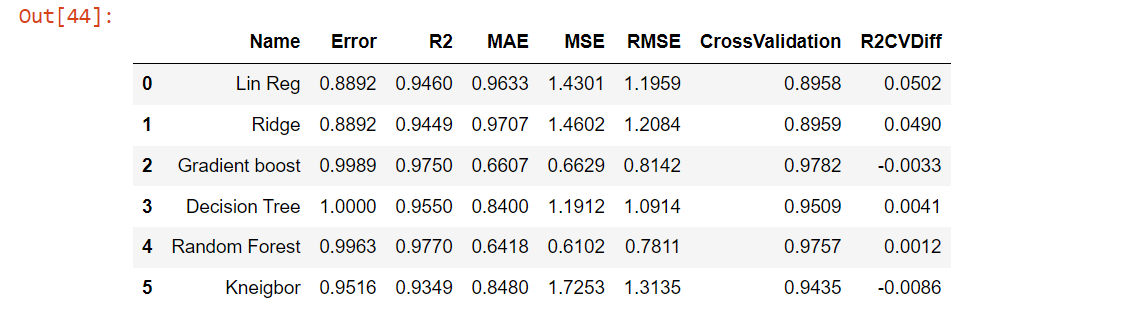


#### 4.6.2 Cross Validation

Cross Validation

* To avoid Over-fitting and Under-fitting issues we can do Cross Validation.
* We will be using the default fold size =5
* Also since R2 is our preferred metric, we will be using our scoring method as “r2” in cross validation.
* Code snippet: of Cross validation using R2
* cross\_val\_score(modelInstance,x,y,cv=5,scoring='r2').mean()

Now the loop we have run for the 6 Algos and the Evaluation Metrics, we will be comparing them using a table. Since we have finalized R2 as the metric that we will be using, we will also do a diff between the Cross Validation and R2 and find the minimum value (last column)



Observation

* 1. Highest R2 score is for Random Forest and Gradient Boost
  2. Hight Cross Validation score is for Random Forest and Gradient Boost
  3. Lowest Diff between R2 and Cross validation is for Random Forest and Gradient Boost.

So we will be fine tuning both Random Forest and Gradient Boost. The Better model will be saved and shared to Biz

### 4.7 Improving the Model

We can fine tune our Model using

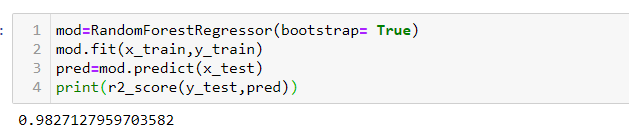
1. Random Search (used if the dataset is large and Algo are complex)
2. Grid Search (used if the dataset is small)

Since our dataset is small, lets use Grid Search to improve our model

#### 4.7.1. Random Forest

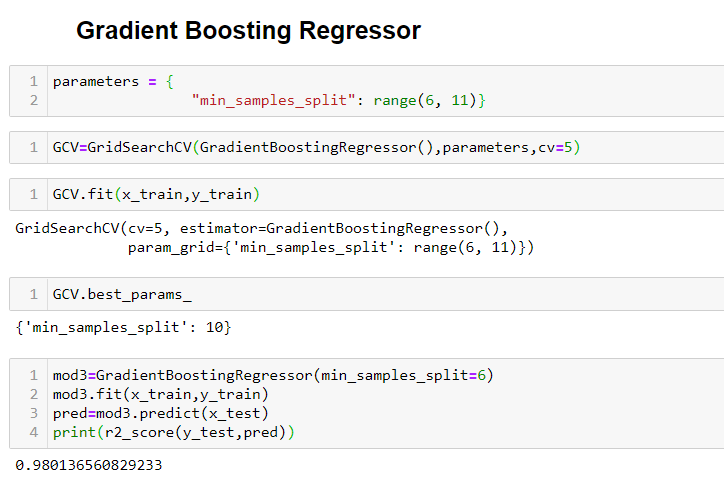
#### 4.7.2 Random Forest 2

Since our model score has slightly improved, we will reduce the parameters and apply the same code only for 1 parameter



Our model has improved now to .9827

#### 4.7.3 Gradient Boost Regressor



Gradient Boost score is also improved. Now we can either use Random Forest or Gradient Boost Hyper Tuned model. Since Random Forest is slightly higher, we will be saving the model and sharing with Biz

Tip 9:

Always try to fine tune the top 2 or 3 models

Tip 10:

Add a Table of contents, its easier to understand the flow and navigate through the document.

## Saving & Loading the Model

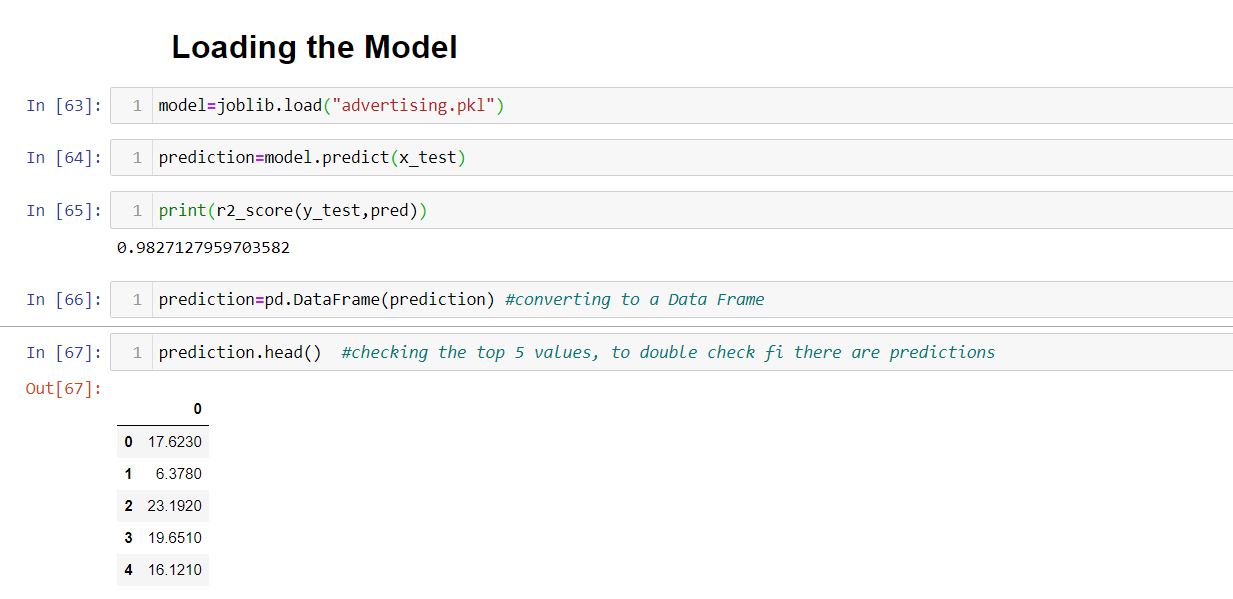
### 5.1 Saving the Model

We will see both the methods of Saving the model, though we need to do only 1 method, we will do both methods for academic purposes



### 5.2 Loading the Model

We will load the model to do a sanity check and see we are getting the same score of .9827 of the finalised model !



## Conclusion

We have got a good prediction of 98.27% with Hyper Parameter Tuning the Random Forest Model.

We have got a good score:-

1. Since we did a good Data Cleaning
2. We Handled the Outliers
3. We handled the Skewness
4. Choose the Right Algorithms
5. Choose the Correct Evaluation Metrics based on our Data

Graphical user interface, application

Description automatically generated