Rajiv Gandhi University of Knowledge Technologies

Srikakulam

DATA SCIENCE WITH PYTHON

Mentor: **Ch. Satish kumar Sir**

Student Details:

Name of the Subject	: Data science with python
Name of the Student	: S.Pavan Kumar
Student ID Number	:S180614
Department	:CSE
Class	:CSE-2C

Department of Computer science and Engineering

PROJECT:

Future Sales Prediction Using Machine Learning.

(Simple & Multi LinearRegression)

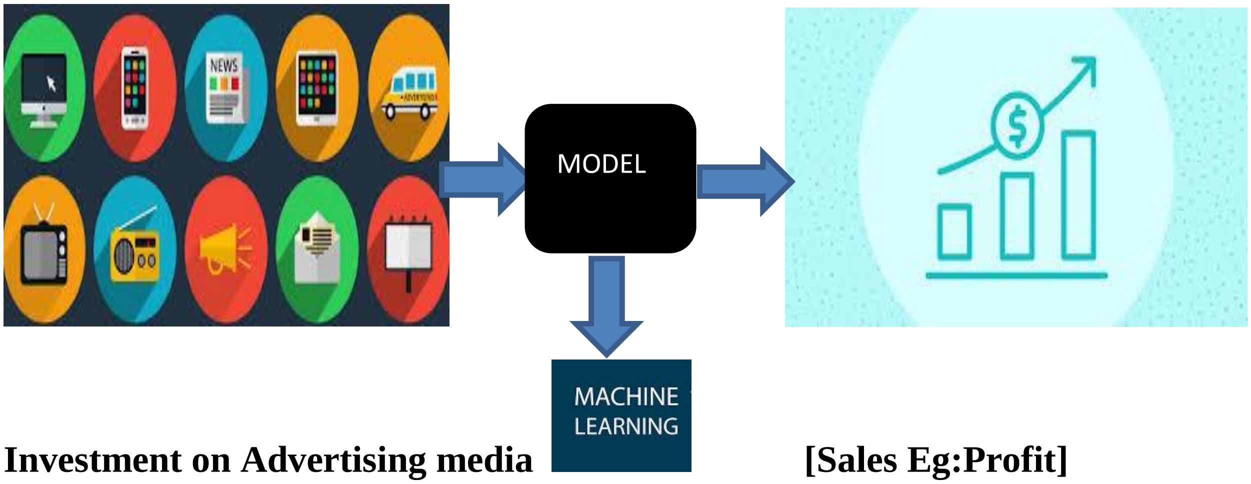
PROBLEM STATEMENT:

Build a model which predicts sales based on the money spent on different platforms for marketing.

INPUT: **Money spent on Advertising Media (Eg:TV,RADIO…)**

OUTPUT: **Sales (predicted in rupees/-)**

INPUT **OUTPUT**



Investment on Advertising media **[Sales Eg:Profit]**

[Eg:Tv]

- Investment in advertisement as input , the model will predict the future sales based on the dataset[Advertising media]. We will train and test the model with that dataset and predict the future sales.

Key Terms:

Sales Forecasting:

Sales forecasting is the process of estimating future revenue by predicting the amount of product or services a sales unit (which can be an individual salesperson, a sales team, or a company) will sell in the next week, month, quarter, or year.

Advertising Media:

Advertising media refers to a variety of mass media or alternative media channels where businesses can promote their products, services, or brand.

DATA SET:

- **USING ADVERTISING DATA SET DOWNLOADED FROM KAGGLE.**
- **Advertising.csv**

Abstract:

This work presents a framework capable of accurate analysis of real time sales data to forecast future sales, visualize sales, and draw important insights or patterns associated with products to achieve greater profits .plotting is the visualizing tool for Sales visualizations. The dataset is used through Prophet Package to train and build a model in order to predict the future sales considering the components of the database constant at the time of prediction. In other words, the dataset should be static. After building the model matplotlib comes to picture which Is used to visualize different types of trends and other useful graphs to retrieve useful insight from the refined dataset created after model building. Matplotlib is used to create a visualization environment in order to create a better user interface which has all the components which are visualized at a same place so the user can use it for knowing what are the predictions made from the dataset provided and can use it to enhancement of sales of the company.

ALGORITHMS USED: REGRESSIONS[Simple & multiple linear Regressions]

Performing Simple Linear Regression

Equation of linear regression

$y=c+m1x1+m2x2+...+mnxn$

- y is the response
- c is the intercept
- m1 is the coefficient for the first feature
- mn is the coefficient for the nth feature

In our case:

$Y=mx+c$

$y=c+m1\times TV$

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

In [1]:

```
#importing Libraries
import pandas as pd
import numpy as np
#importing Data Visualisations
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
#importing dataset
data=pd.read_csv("Advertising_media.csv")
data.head()
```

Out[2]:

	TV	Radio	Newspaper	SocialMedia	Sales
0	127.2	70.6	61.5	46.7	632.9
1	148.3	126.4	27.3	47.5	682.7
2	146.7	105.6	16.0	14.1	508.4
3	153.0	142.3	39.5	21.3	672.1
4	159.7	164.9	74.5	47.5	758.3

In [3]:

```
data.describe()
```

Out[3]:

	TV	Radio	Newspaper	SocialMedia	Sales
count	156.000000	156.000000	156.000000	156.000000	156.000000
mean	122.565385	96.405769	46.237821	36.123077	583.967949
std	32.550274	44.927655	23.809175	13.185133	120.215890
min	0.000000	0.000000	0.000000	0.000000	308.100000
25%	106.675000	62.675000	27.225000	28.325000	490.825000
50%	130.050000	104.350000	44.400000	38.400000	581.000000
75%	148.225000	133.100000	62.100000	46.700000	667.950000
max	167.000000	185.200000	141.500000	58.100000	839.100000

In [4]:

```
#Data Inspection
data.shape
```

Out[4]:

```
(156, 5)
```

In [5]:

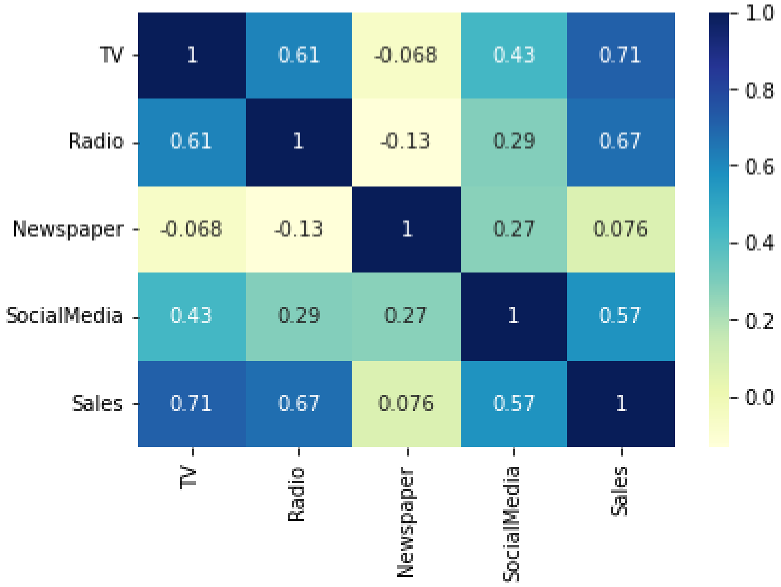
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 156 entries, 0 to 155
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   TV           156 non-null    float64
 1   Radio        156 non-null    float64
 2   Newspaper    156 non-null    float64
 3   SocialMedia  156 non-null    float64
 4   Sales        156 non-null    float64
dtypes: float64(5)
memory usage: 6.2 KB
```

In [6]:
```
#Data Cleaning
data.isnull().sum()*100/data.shape[0]
#There are no NULL values in the dataset, hence it is clean.
```

Out[6]:
```
TV             0.0
Radio          0.0
Newspaper      0.0
SocialMedia    0.0
Sales          0.0
dtype: float64
```

In [7]:
```
# Let’s see the correlation between different variables.
sns.heatmap(data.corr(), cmap="YlGnBu", annot = True)
plt.show()
```



In [8]:
```
#As is visible from the pairplot and the heatmap, the variable TV seems to be
```

Model Building

Performing Simple Linear Regression Equation of linear regression

Simple linear regression

HERE WE WILL ESTIMATE THE SALES WITH RESPECT TO THE ADVERTISEMENT ON TV

Generic Steps in model building

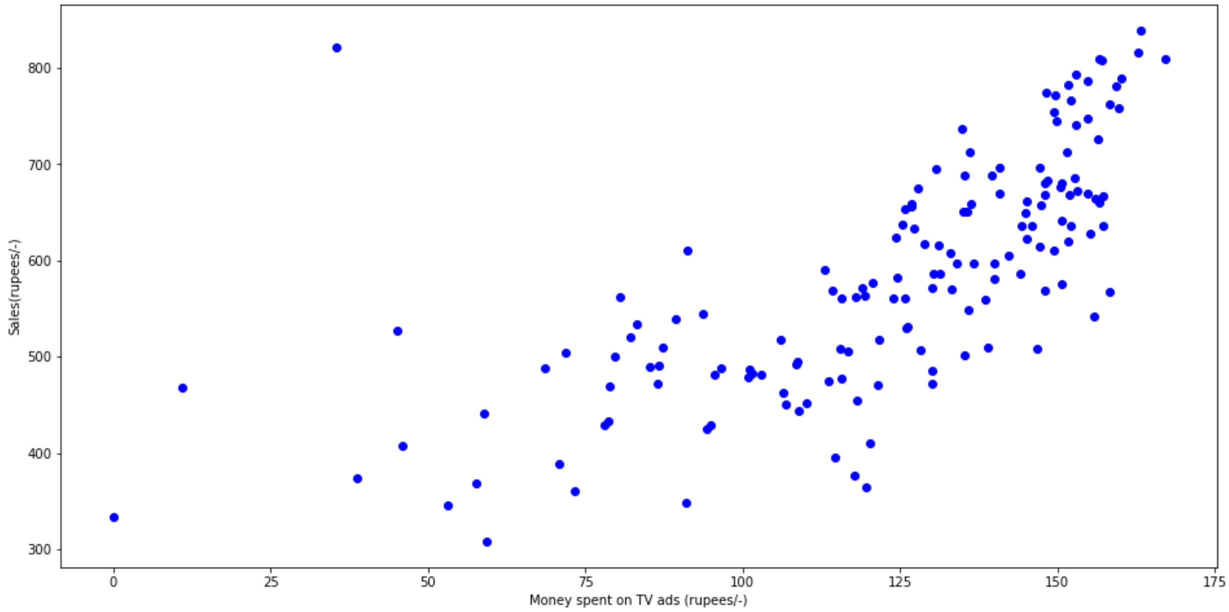
We first assign the feature variable, TV, in this case, to the variable x and the response variable, Sales, to the variable y.

In [9]:

```
#INTIALISING THE VARIABLES
x=data['TV'].values.reshape(-1,1)
y=data['Sales'].values.reshape(-1,1)
```

In [10]:

```
#PLOTING A GRAPH TO SEE THE POINTS
plt.figure(figsize=(16,8))
plt.scatter(x,y,c="b")
plt.xlabel("Money spent on TV ads (rupees/-)")
plt.ylabel("Sales(rupees/-)")
plt.show()
```



Train-Test Split

You now need to split our variable into training and testing sets. You'll perform this by importing train_test_split from the sklearn.model_selection library. It is usually a good practice to keep 80% of the data in your train dataset and the rest 20% in your test dataset

In [11]:

```
#SPLITTING OUR DATASET TO TRAINING AND TESTING DATASET
from sklearn.model_selection import train_test_split
x_train, X_test, y_train, y_test = train_test_split(x,y, test_size=0.2, random_
```

In [12]:

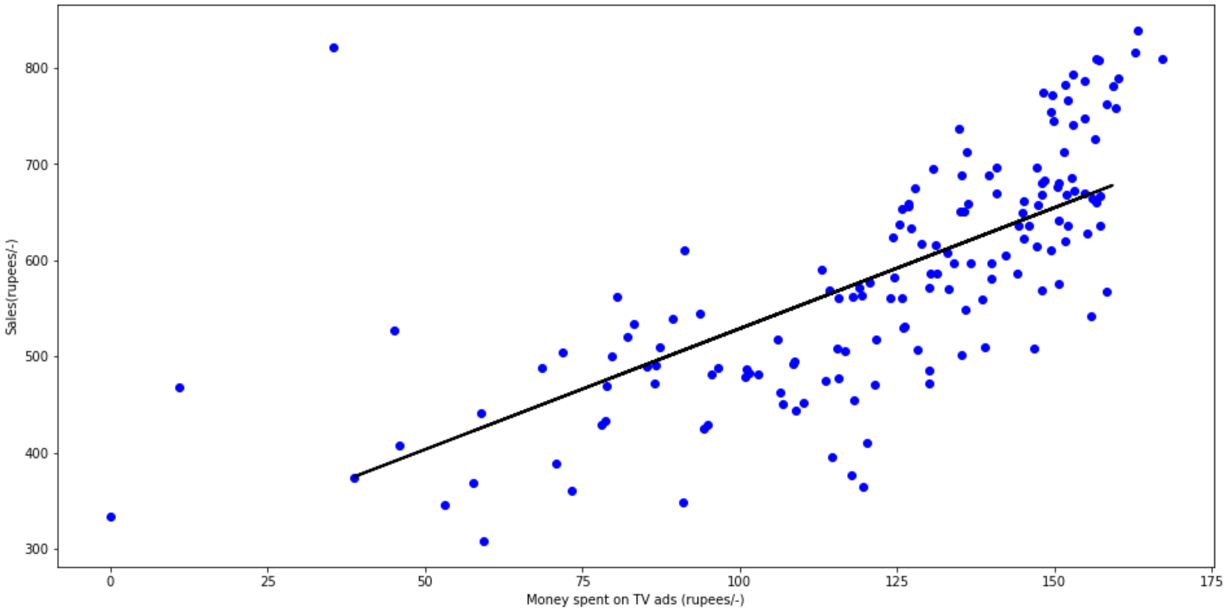
```
#FITTING LINEAR REGRESSION TO THE TRAINING SET
from sklearn.linear_model import LinearRegression
reg=LinearRegression()
reg.fit(x_train, y_train)
```

Out[12]:

```
LinearRegression()
```

In [13]:

```
#PREDICTING THE TEST SET RESULT
y_pred=reg.predict(X_test)
plt.figure(figsize=(16,8))
plt.scatter(x,y,c="b")
plt.plot(X_test,y_pred,c="black",linewidth=2)
plt.xlabel("Money spent on TV ads (rupees/-)")
plt.ylabel("Sales(rupees/-)")
plt.show()
```



In [14]:

```
#CALCULATING THE COEFFICIENTS
reg.coef_
```

Out[14]:

```
array([[2.51290983]])
```

In [15]:

```
#CALCULATING THE INTERCEPT
reg.intercept_
```

Out[15]:

```
array([277.64669569])
```

In [16]:

```
#CALCULATING THE R SQUARED VALUE(RESIDUALS)
from sklearn.metrics import r2_score
r2_score(y_test, y_pred)
```

Out[16]:

```
0.6075092528330941
```

In [17]:

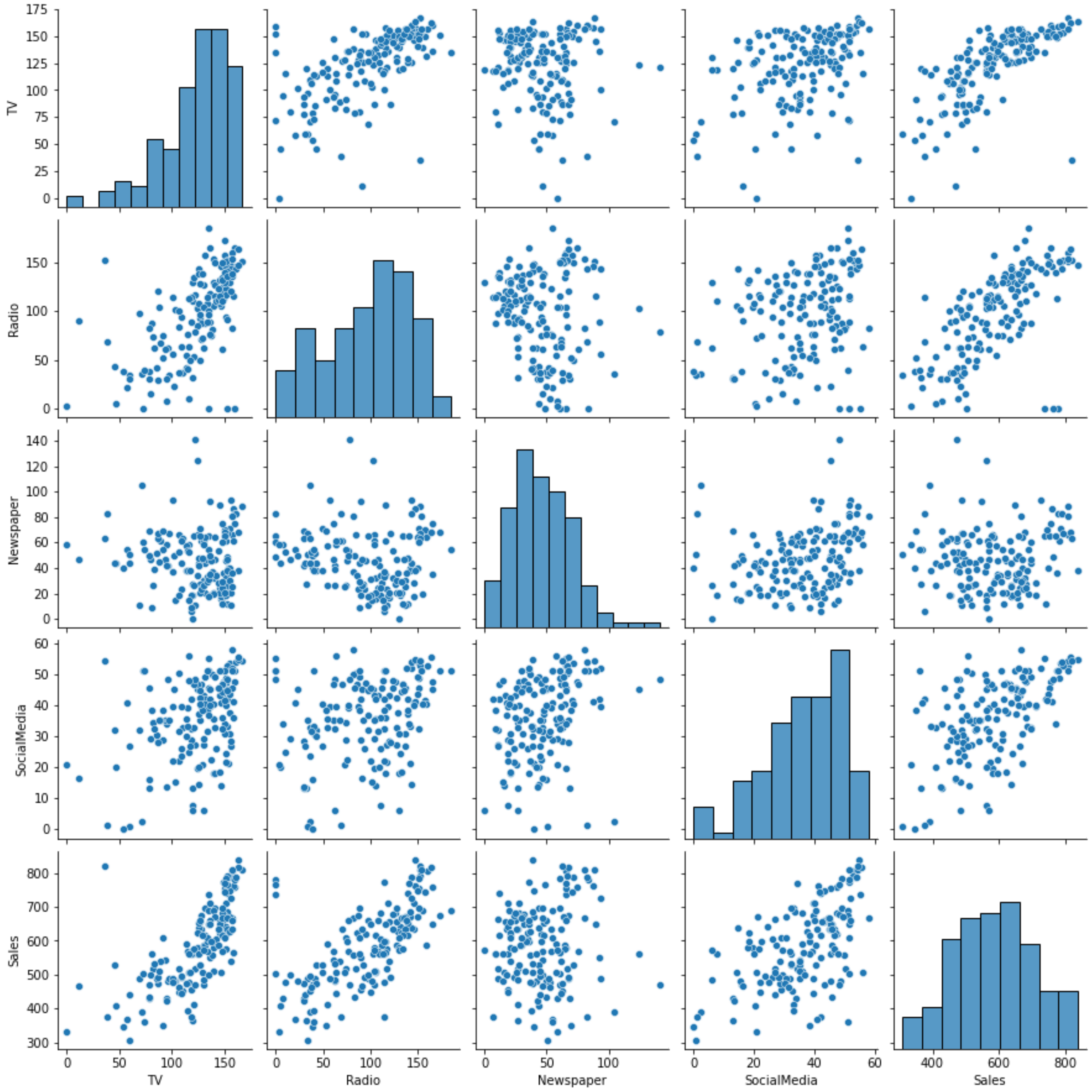
```
output=reg.predict([[127.2]])
output
```

Out[17]:

```
array([[597.28882667]])
```

In [26]:

```
sns.pairplot(data, kind='scatter')
plt.show()
```



Multiple Linear Regression

In [18]:

```
#INTIALISING THE VARIABLES
x=data.drop(['Sales'],axis=1)
y=data['Sales'].values.reshape(-1,1)
```

In [19]:

```
#SPLITTING OUR DATASET TO TRAINING AND TESTING DATASET
from sklearn.model_selection import train_test_split
x_train, X_test, y_train, y_test = train_test_split(x,y, test_size=0.2, random_
```

In [20]:

```
#FITTING LINEAR REGRESSION TO THE TRAINING SET
from sklearn.linear_model import LinearRegression
multiple_reg=LinearRegression()
multiple_reg.fit(x_train, y_train)
```

Out[20]:

```
LinearRegression()
```

In [21]:

```
#PREDICTING THE TEST SET RESULT
y_pred=multiple_reg.predict(X_test)
```

In [22]:

```
#CALCULATING THE COEFFICIENTS
multiple_reg.coef_
```

Out[22]:

```
array([[1.12782401, 1.14090308, 0.22398765, 2.52805394]])
```

In [23]:

```
#CALCULATING THE INTERCEPT
multiple_reg.intercept_
```

Out[23]:

```
array([232.84315978])
```

In [24]:

```
#CALCULATING THE R SQUARED VALUE(RESIDUALS)
from sklearn.metrics import r2_score
r2_score(y_test, y_pred)
```

Out[24]:

```
0.717562922944331
```

In [25]:

```
TV=float(input("tv:"))
Radio=float(input("Radio:"))
Newspaper=float(input("News paper"))
SocialMedia=float(input("social media:"))
y=multiple_reg.predict([[TV,Radio,Newspaper,SocialMedia]])
print("sales:  "+str(y[0][0])+" /-")
```

```
tv:127.2
Radio:70.6
News paper61.5
social media:46.7
sales:  588.6854907682148 /-
```

```
C:\Users\pavan sairigapu\anaconda3\lib\site-packages\sklearn\base.py:450: User
Warning: X does not have valid feature names, but LinearRegression was fitted
with feature names
  warnings.warn(
```

CONCLUSION:

- **This Model allows companies to efficiently allocate resources for future growth and manage their cash flow.**
- **Step 1: Identifying target and independent features.**
- **Step 2: Cleaning the data set. First, we check for null values by running input.**
- **Step 3: Exploratory Data Analysis. Descriptive Statistics.**
- **Step 4: Building a model.**
- **Step 5: Check model accuracy.**
- **Step 6: Save the model.**
- **In this way we predict the future sales by Machine Learning.**

Thank you…