Sana Afreen

Sales Predictions using Advertising Costs by using Machine Learning Model

Problem Statement – In the business world, advertising is a crucial element for any company looking to promote its products or services. However, advertising costs can be substantial, and businesses need to determine the effectiveness of their advertising campaigns. This is where sales prediction comes in – it's a critical aspect of advertising that helps companies understand how much revenue they can expect from their advertising campaigns. In this project I build a model which predicts sales based on the money spent on different advertising platforms for marketing.

Steps to be taken in the Project is sub-divided into the following sections. These are:

- Loading necessary libraries such as numpy , pandas , sklearn etc.
- Loading the dataset as CSV file and showing first 10 rows.
- Drop the unnecessary columns from dataset.
- Calculate statistical values and round them up to 3 decimal places.
- Checking for null values and return their sum of numbers of true values in
- each column.
- Handle the null by mean of all values fill into them.
- Extracting all information about data.
- Checking Shape of Data.
- Visualization of Sales by different source of Advertisement cost using Python data visualization.
- Data preprocessing or (Data cleaning) performed by the one hot encoding in this process we change categorical data into numerical data and the technique is called feature Engineering.
- Splitting the cleaned data into dependent and independent variables.
- Splitting the data into train and test sets with train_test_split using sklearn library.
- Import different kind of Regression Models and Train that model with the help of. fit().
- Predicting the trained models and then checking their accuracy of the model using accuracy score.
- Then recall the train_test_split and split the data into training and testing set with different models.

- Then predicting the trained models and checking the accuracy of model and check the accuracy difference.

Step-1 – Loading Necessary Libraries used in machine learning.

```
# Import Necessary Libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as snr
from sklearn.preprocessing import OneHotEncoder, PolynomialFeatures
from sklearn.metrics import r2_score
```

Step-2 – Loading the dataset as csv file and showing first ten rows.

```
data=pd.read_csv(r'/content/gdrive/My Drive/advertising.csv')
data.head(10)
```

	TV	Radio	Newspaper	Sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	12.0
3	151.5	41.3	58.5	16.5
4	180.8	10.8	58.4	17.9
5	8.7	48.9	75.0	7.2
6	57.5	32.8	23.5	11.8
7	120.2	19.6	11.6	13.2
8	8.6	2.1	1.0	4.8
9	199.8	2.6	21.2	15.6

Step-3 – Calculate statistical values and round them up to 3 decimal places.

```
# Calculate statistical values and round them up to 3 decimal places.
data.describe().round(3)
```

	TV	Radio	Newspaper	Sales
count	200.000	200.000	200.000	200.000
mean	147.042	23.264	30.554	15.131
std	85.854	14.847	21.779	5.284
min	0.700	0.000	0.300	1.600
25%	74.375	9.975	12.750	11.000
50%	149.750	22.900	25.750	16.000
75%	218.825	36.525	45.100	19.050
max	296.400	49.600	114.000	27.000

Step-4 – Checking for null values and return their sum of numbers of true values ineach column.

```
## Mark null values as True and returns sum of number of True values in each column
data.isnull().sum()
```

	0
TV	0
Radio	0
Newspaper	0
Sales	0

dtype: int64

Sana Afreen

Step-5 – Extracting all information about data.

```
# Extracting all information about data

data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   TV         200 non-null    float64
 1   Radio      200 non-null    float64
 2   Newspaper  200 non-null    float64
 3   Sales      200 non-null    float64
dtypes: float64(4)
memory usage: 6.4 KB
```

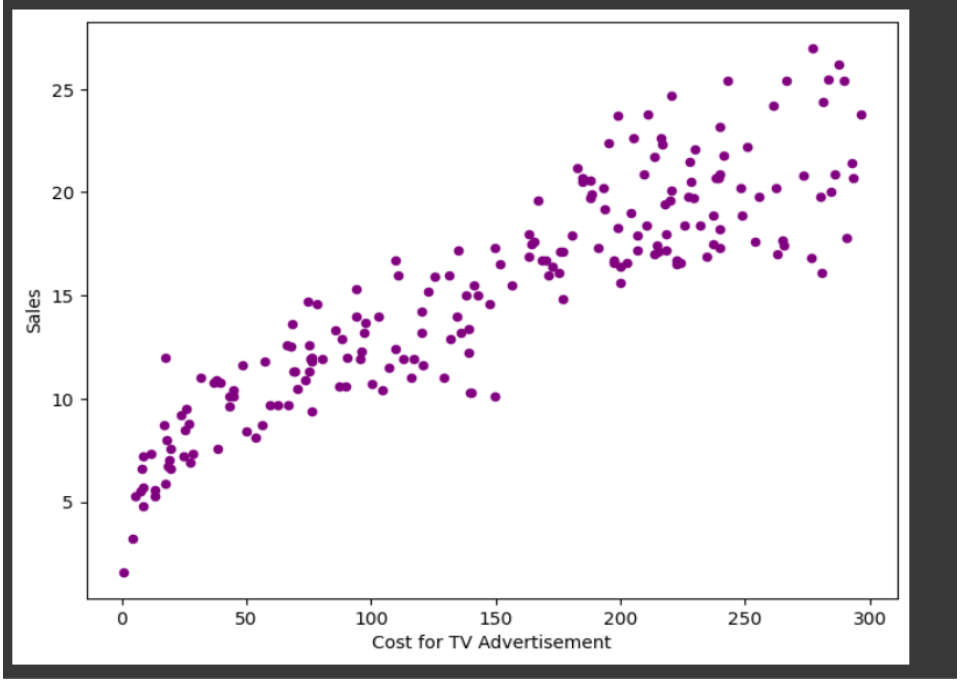
Step-6 – Checking shape of data.

```
#Shape of Data

data.shape
```

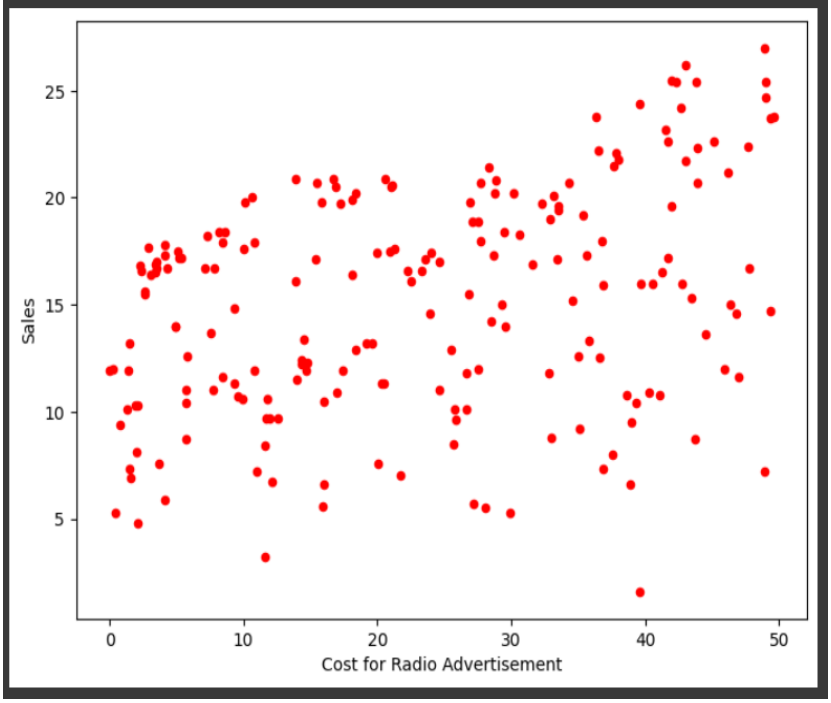
Step-7 – Visualization of Sales by different source of Advertisement cost usingPython data visualization.

```
ax=data.plot.scatter(x='TV', y='Sales', figsize=(8,6), color='purple')
ax.set_xlabel('Cost for TV Advertisement')
plt.show()
```

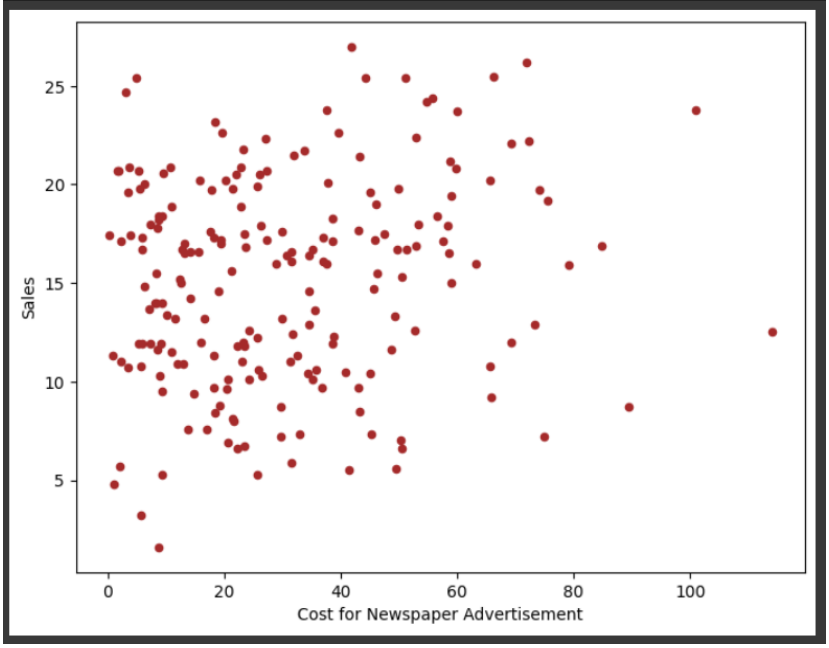


Sana Afreen

```
ax=data.plot.scatter(x='Radio', y='Sales', figsize=(8,6), color='red')
ax.set_xlabel('Cost for Radio Advertisement')
plt.show()
```




```
ax=data.plot.scatter(x='Newspaper', y='Sales', figsize=(8,6), color='brown')
ax.set_xlabel('Cost for Newspaper Advertisement')
plt.show()
```



Sana Afreen

Step-8 – Splitting the cleaned data into dependent and independent variables.

```
[ ]  # Dividing data into dependent and independent variables.
     x=data.drop(['Sales'], axis=1)
     y=data['Sales']
     y.head()
```

	Sales
0	22.1
1	10.4
2	12.0
3	16.5
4	17.9

dtype: float64

```
     x.head()
```

	TV	Radio	Newspaper
0	230.1	37.8	69.2
1	44.5	39.3	45.1
2	17.2	45.9	69.3
3	151.5	41.3	58.5
4	180.8	10.8	58.4

Step-9 – Splitting the data into train and test sets with train_test_split usingsklearn library.

```
[ ]  # Dividing the cleaned data into training and testing sets.
     from sklearn.model_selection import train_test_split
     x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.8)
```

Step-10 – Import first regression model ‘Linear Regression’.

```
[ ]  # creating first Machine Learning Model 'Linear Regression'.
     from sklearn.linear_model import LinearRegression
     linear=LinearRegression()
```

Sana Afreen

Step-11 – Train the model using. fit () function.

```
# Train the model
linear.fit(x_train,y_train)
```

```
LinearRegression
LinearRegression()
```

Step-12 – Make predictions on model.

```
# Make Predictions on the model
predictions=linear.predict(x_test)
print(predictions)
```

```
[10.90414046 15.78830685 16.79048569 13.74378714  5.50507865 16.92621438
  8.63660352 10.3539861  13.31270377 10.61054609 19.13861409  6.99646804
 19.31804704 24.35287089 23.21803049 12.12329175 12.51031339  7.22563848
 20.52741761 15.81699273 17.80108454 10.67637785  8.00786539  6.40475218
 21.63847203 18.17108142 15.62665143 17.71631506  9.14020531 20.63602783
 14.86735339  9.79340474 19.32132021 10.34332984 21.24785247 22.87931303
 16.46197051 11.98159922 21.1889412  11.15650128]
```

Step-13 – Check the accuracy of model by r2 score.

```
[ ]   # Check accuracy score

      print(r2_score(y_test, predictions))
```

Sana Afreen

Step-14 – Import the Second Machine Learning Model Polynomial Regressionand train model and then make prediction

```
[ ]  # Import the second machine learning model 'Polynomial regression'.

     from sklearn.preprocessing import PolynomialFeatures
     poly = PolynomialFeatures(degree=2, include_bias=True)
     x_train_trans=poly.fit_transform(x_train)
     x_test_trans=poly.transform(x_test)
     linear=LinearRegression()
```

```
     # Train the model
     linear.fit(x_train_trans,y_train)
```

```
     LinearRegression
     LinearRegression()
```

```
[ ]  # Make predictions on model
     y_predictions=linear.predict(x_test_trans)
```

Step-15 – Check the accuracy of model by r2 score.

```
# Check acuracy score
print(r2_score(y_test,y_predictions))
```

```
0.9521990800366866
```

Sana Afreen

Step-16 – Import the Third Machine Learning Model Support Vector Regressorand train model and then make prediction.

```
[ ]  # Import the third Machine Learning Model 'SVM regressor'.
     from sklearn.svm import SVR
     svr=SVR()

     # Train the Model
     svr.fit(x_train,y_train)
```

```
SVR
SVR()
```

```
[ ]  # Make Predictions on model
     pred_svr=svr.predict(x_test)
     print(pred_svr)
```

```
[ 9.91172433 16.68803313 17.56640063 14.44721048  7.47973589 16.43424604
  9.30084377  9.32864659 13.34030516 11.4853459  18.82213348  7.88522625
 18.97964506 21.17223345 21.17073199 12.42107585 13.80133949  8.04944027
 19.61469921 16.85130059 17.99328181  9.86299292  7.95802249  7.9904025
 19.98822034 18.32969498 16.64474408 17.80450152  8.44050932 19.95228317
 14.50441414  9.54189146 18.75408087 11.61702664 20.1888989  21.10751982
 16.34696273 12.79086172 20.13322396 10.1068724 ]
```

Step-17 – Check the accuracy score.

```
[ ]  # Check the acuracy score of model
     print(r2_score(y_test, pred_svr))
```

```
0.8751319729566627
```

Sana Afreen

Conclusion – In this project, we have demonstrated in detail how to apply linear regression, polynomial regression and support vector regressor model for predicting sales from data of spend cost for advertising. By carefully selecting the right variables, preparing and cleaning the data, and selecting an appropriate regression models, businesses can accurately predict sales from advertising cost ads.

project analysis resulted in a good R-squared value of 0.87100, 0.93144 and 0.89639 respectively, which indicates that the linear regression model has a decent fit for the data and gave 87%, 93% and 89% accuracy respectively. This level of accuracy can provide businesses with valuable insights into the effectiveness of their advertising campaigns and enable them to make informed decisions about how to allocate their resources.

Sana Afreen

Sana Afreen