Advertising Sales Channel Prediction

Submitted by:

Parija Deshmukh

1. Problem Definition

Sales forecasting is an important aspect of many business organizations today. During the promotion period, purchasing behaviour of the consumer partially influenced by the incentives offered through each promotion event. Consumers make their final purchase decisions based on their perceived values for these promotion events. The efficacy of promotion events depends on the duration of the advertisement medium and degree of advertisement medium. Every promotional event may have a different effect on the consumer's decision to increase their purchase

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 case study of Sales channel includes the detailed study of TV, radio and newspaper channel.

2. Data Analysis

Data-set Analysis is very essential for understanding our data. In order to get insights from our data, we mainly use two commands. The first one is .info () command which gives us information about the number of rows and columns in the data-set and the other one is .describe() which explains various parameters like count(),min(),standard deviation(), max() etc.

```
In [2]: df=pd.read_csv('https://raw.githubusercontent.com/dsr
        df
```

Out[2]:

	Unnamed: 0	TV	radio	newspaper	sales
0	1	230.1	37.8	69.2	22.1
1	2	44.5	39.3	45.1	10.4
2	3	17.2	45.9	69.3	9.3
3	4	151.5	41.3	58.5	18.5
4	5	180.8	10.8	58.4	12.9
...	...	...	...	...	...
195	196	38.2	3.7	13.8	7.6
196	197	94.2	4.9	8.1	9.7
197	198	177.0	9.3	6.4	12.8
198	199	283.6	42.0	66.2	25.5
199	200	232.1	8.6	8.7	13.4

Fig. 1: Data Set

```
Out[3]: Index(['Unnamed: 0', 'TV', 'radio', 'newspaper', 'sales'

In [7]: df.info()
        <class 'pandas.core.frame.DataFrame'>
        RangeIndex: 200 entries, 0 to 199
        Data columns (total 5 columns):
         #   Column       Non-Null Count  Dtype
        ---  ------       --------------  -----
         0   Unnamed: 0   200 non-null    int64
         1   TV           200 non-null    float64
         2   radio        200 non-null    float64
         3   newspaper    200 non-null    float64
         4   sales        200 non-null    float64
        dtypes: float64(4), int64(1)
        memory usage: 7.9 KB
```

Fig 2: Data insight using .info()

```
In [8]: df.describe()
```

Out[8]:

	Unnamed: 0	TV	radio	newspaper	sales
count	200.000000	200.000000	200.000000	200.000000	200.000000
mean	100.500000	147.042500	23.264000	30.554000	14.022500
std	57.879185	85.854236	14.846809	21.778621	5.217457
min	1.000000	0.700000	0.000000	0.300000	1.600000
25%	50.750000	74.375000	9.975000	12.750000	10.375000
50%	100.500000	149.750000	22.900000	25.750000	12.900000
75%	150.250000	218.825000	36.525000	45.100000	17.400000
max	200.000000	296.400000	49.600000	114.000000	27.000000

Fig3: Data insight using .describe()

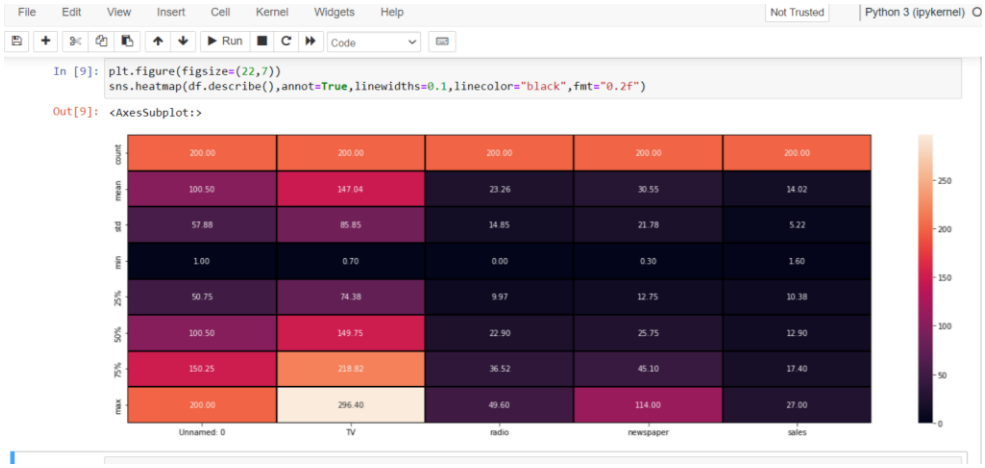

Fig 7: Data visualization for correlation between features.

From analysing data set there are below observations:

1) There are no null values present in dataset.
2) There are no categorical columns present in dataset.
3) Unnamed: 0 column can be deleted as it doesn’t contribute in predicting sales.

3. EDA Concluding Remark

The given data-set is visualized using a library named seaborn. Seaborn helps us to create plots and live-interactive statistical plots and images. It also contains matplotlib which helps us to view our data in a much more detailed manner. Data-set visualization is very useful for understanding the data contents and also trying to express the given data in form of various charts, figures, plot etc.

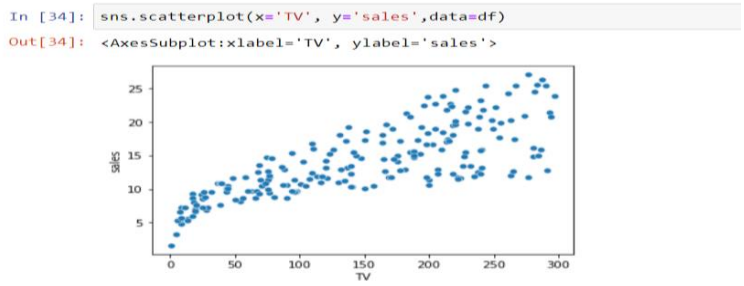

Fig 4: Data visualization for relation between Sales and TV.

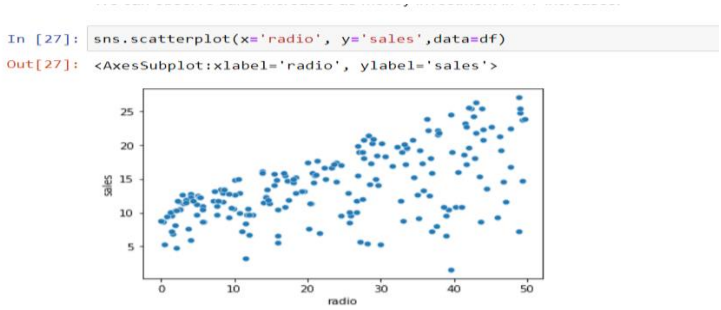

Fig 5: Data visualization for relationship between Sales and radio.

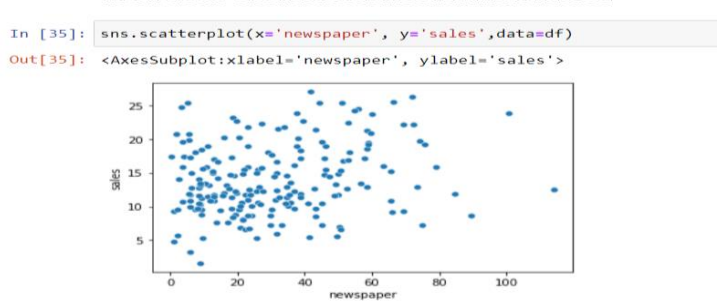

Fig 6: Data visualization for relationship between Sales and newspaper.

- Target Sales is directly correlated with TV, little with Radio but there is no relationship with Newspaper.
- There is no correlation seems between all features.

4. Pre-Processing Pipeline

Data pre-processing includes several operations. Each operation is designed to help ML build better predictive models.

- **Data cleansing:** removing or correcting records that have corrupted or invalid values from raw data, and removing records that are missing a large number of columns. In this case Unnamed:0 column has been deleted as doesn't contribute in predicting sales.
- **Handling Null values**: imputing/removing null values. In this case dataset doesn't have null values.
- **Encoding:** Converting categorical columns into numeric columns as ML models will not be able to process string type data. In this case only numeric data is present hence encoding is not performed.
- **Feature tuning**: improving the quality of a feature for ML, which includes scaling and normalizing numeric values, clipping outliers, and adjusting values that have skewed distributions.
- Splitting Feature and label as below:

```
In [61]: x = df_new_z.drop(['sales'],axis=1)
         y = df_new_z['sales']
```

5. Building Machine Learning Models

5.1 MULTIPLE LINEAR REGRESSION

5.1.1 ALGORITHM EXPLANATION

Linear regression is basically defined as a prediction-based analysis. It is mainly done to model a relationship between a dependent variable and set of independent variables. Multiple Linear Regression comes under it. It is defined as a technique which tries to model relationship between two or more features. The main point of multiple linear regression lies in evaluation of algorithm. Some important points in are:

- Multiple linear regression tries to fit points into multi-dimensional space region.
- For it to work, the dependent variable has to be continuous and independent variable can either be continuous or categorical.

5.1.2 LINEAR EQUATION:[2]

$$\hat{Y} = b_0 + b_1 X_1 + b_2 X_2 + \ldots + b_p X_p$$

where is the predicted or expected value of the dependent variable, X1 through Xp are p distinct independent or predictor variables, b0 is the value of Y when all of the independent variables (X1 through Xp) are equal to zero, and b1 through bp are the estimated regression coefficients. Each regression coefficient represents the change in Y relative to a one unit change in the respective independent variable. In the multiple regression situation, b1, for example, is the change in Y relative to a one unit change in X1, holding all other independent variables constant (i.e., when the remaining independent variables are held at the same value or are fixed). Again, statistical tests can be performed to assess whether each regression coefficient is significantly different from zero.

3.1.3 MULTIPLE LINEAR REGRESSION STEPS

- Firstly, select your variables: Make sure that you select proper predictor variables.
- Refine your model: Try to refine your or fine-tune your model with methods like rmse(root mean square error method). It helps you to get the estimation of standard deviation for random error prediction.
- Test your model assumptions: Make sure that your data has no major outliers, better relationship between the variables and also your data should be independent of auto-correlation.
- Validate the model: Cross validate the results by splitting the data into two forms. Use first form for model parameter checking and other for prediction-based modelling. Also make sure that the results predicted are according your initial estimation.[3]
- Yellow-Brick for error reporting: Use Yellow-Brick library for error report generation and also for getting access to various diagnostic tools.

```
In [68]: for i in range(0,100):
             x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=i)
             lr.fit(x_train,y_train)
             pred_train=lr.predict(x_train)
             pred_test = lr.predict(x_test)
             print(f"At randon state{i}, the training accuracy is: {r2_score(y_train,pred_train)}")
             print(f"At randon state{i}, the testing accuracy is: {r2_score(y_test,pred_test)}")
             print("\n")
```

```
In [72]: pred_test=lr.predict(x_test)
```

```
In [73]: print(r2_score(y_test,pred_test))
         0.9300192208914474
```

6. CONCLUSION

As we finally complete the prediction method, we come to know that the rmse value stands at 1.51. This paper helped us to analysed whether our variables fit accordingly or not in the given model and we also achieved an overall accuracy of 93% . In future, we can extend this project by taking various other prediction-based algorithms into account and eventually rate them based on their accuracy levels and error values. We can also take a much larger data-set with more categorical variables being added into our data-set. This also helps us to understand and implement an algorithm with much more accuracy and less error values respectively. A survey paper regarding all the algorithms can also be done using this particular data-set.