

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of Bachelor of
Engineering Degree in Computer Science and Engineering

By

E. YOGESHWARAN
(Reg.No: 40733032)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU

April 2023



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by
NAAC



(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI
CHENNAI- 600119
www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **E.YOGESHWARAN (40733032)** carried out the project entitled "**SALES PREDICTION USING LINEAR REGRESSION**" under my supervision from Feb to Apr 2023.

Internal Guide

Ms.D.GEETHANJALI, M.E.,(Ph.D.)

Head of the Department

Dr. S. Vigneshwari, M.E., Ph. D

Submitted for Viva-voce Examination held on **17-04-2023**

Internal Examiner

External Examiner

DECLARATION

I **E.YOGESHWARAN** hereby declare that the Project Report entitled **SALES PREDICTION USING LINEAR REGRESSION** done by me under the guidance of **Ms.D.GEETHANJALI, M.E.,(Ph.D.)** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE :17/04/2023

PLACE: Chennai



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

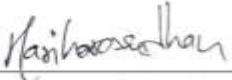
I am pleased to acknowledge my sincere thanks to the **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D**, Dean, School of Computing, **Dr. S. Vigneshwari, M.E., Ph.D** Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide, **Ms.D.GEETHANJALI, M.E.,(Ph.D.)** for her valuable guidance, suggestions, and constant encouragement that paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE

	
	COGNIBOT <i>AI meets Industry</i>
	Certificate of Training
	is hereby granted to
	E. YOGESHWARAN (40733032)
from Sathyabama Institute of Science and Technology for successfully completing the 45 hours professional training program on Machine Learning conducted by Cognibot between 10 th Feb, 2023 and 16 th Apr, 2023	
	16 th April, 2023
HARITHARASUDHAN INSTRUCTOR, COGNIBOT	DATE

Chapter No	Table of contents	Page Number
	Abstract	I
	List of Figures	II
	List of Tables	III
	List of Abbreviations	IV
1	Introduction	1
2	Aim and Scope	2
3	Proposed system & Algorithms used	3
3.1	Dataset of Sales	3
3.2	Reading and Understanding the Data	3
3.3	Data Inspection	4
3.4	Data Cleaning	4
3.5	Data Analysis	5
3.6	Model Building	5
3.7	Model Fitting	6
3.8	Model Evaluation	6
3.9	Predicting on a Test Set	6
3.10	Visualizing model fit on the test set	7
4	Hardware & Software requirements	8
5	Results	9
5.1	Performance analysis	14
5.2	Conclusion	15
6	Appendix	16
6.1	Screenshots	16

7	Source code	26
7.2	References	28

ABSTRACT

Machine Learning is transforming every walk of life and has become a major contributor in real world scenarios. The revolutionary applications of ML can be seen in every field including education, healthcare, engineering, sales, entertainment, transport and several more; the list is never ending. In this mini project we are going to predict the sales of a company with an old advertising data. The prediction includes data parameters such as Sales made by the company using various advertising methods/mediums such as TV, Radio, and Newspaper. We are going to understand the data followed by data inspection and data cleaning. Next proceed with data analysis process and the build, fit & evaluate the model. And then predict and visualize the test set. In this mini-project, a comprehensive study of sales prediction is done using the ML model called Linear Regression. The aim of this mini-project is to propose a trained model for predicting the future sales of Big Companies keeping in view the sales of previous years.

LIST OF FIGURES

Index	Figure Name	Page Number	Figure Number
3.1	Dataset of Sales	9	1
3.2	Reading and Understanding the Data	10	2
3.3	Data Inspection	10	3
3.4	Data Outliers Box Plot	11	4
3.5.1	Scatter Plot	11	5
3.5.2	Heat Map	12	6
3.6	OLS Regression Summary	12	7
3.7	Model fitting-Scatter Plot	13	8
3.8	Histogram of Error terms	13	9
3.9	Model fit on test set	14	10

LIST OF TABLES

Index	Figure Title	Page Number
3.2	Visualized Data Table	3
3.3	Description Table of Dataset	4

LIST OF ABBREVIATIONS

Index	Abbreviation	Stands for
1	ML	Machine learning
2	TV	Television
3	RMSE	Root Mean Squared Error
4	RS	Root Squared
5	CPU	Central Processing Unit
6	RAM	Random Access Memory

CHAPTER 1

INTRODUCTION

Current status of Sales in the Industry:-

The retail market of sales business is one of the gravest and major business provinces of the statistical mining in data science. The accomplishment of the statistical mining due to its numerous optimization problems and prolific data such as most appropriate prices, discounts, tips and inventory stage can be solved with the aid of analyzing the existing data.

Sales Prediction for dynamic business environment in today's competitive business market remains a challenge and fails to satisfy the demands of the customer from the sales market. To make the minor improvements of the diversified retailers in terms of higher demand for prediction is to lower the costs. It should be done while making the improvements of sales and customer satisfaction.

The Algorithm used in Sales Prediction:-

A training model trained using Linear Regression can predict development based solely on the statistics set that is used to create the model.

While making a prediction, any new information delivered is integrated into the model and regressed until an accurate model is formed. The most used prediction model is a Linear Regression model. One constituent of Linear Regression is the auto regressive model, meaning it depends upon the past errors while training to achieve an high accuracy.

The inputs used for Sales Prediction:-

The prediction includes data parameters such as Sales made by the company using various advertising methods/mediums such as:

- TV.
- Radio.
- Newspaper.

CHAPTER 2

AIM AND SCOPE

The Aim of this project is to use Model Training techniques such as Linear Regression to predict the Sales of a Company/Organization using old advertising data from various platforms such as Newspaper, TV and Radio.

The ultimate goal of this project is to provide businesses with a reliable tool for predicting sales and making data-driven decisions regarding pricing, marketing, and resource allocation.

With satisfactory results and model accuracy, this project will assist shops, companies and organizations to enhance productiveness and expand income by taking benefit of future information analysis.

Sales prediction performs a necessary function in growing the effectiveness with which shops can function as it presents the relationship between the Sales data and the source of advertising through which the sales is achieved.

CHAPTER 3

PROPOSED SYSTEM AND ALGORITHMS USED

The proposed system has two steps, this project consists of two parts, the Dataset that contains the Sales data and The Prediction algorithm that is based on Linear Regression. This chapter covers the working of this Predictive Algorithm in detail.



3.1 Dataset of Sales:

The dataset contains the general sales data and the Profit earned through sales using advertising methods such as TV, Radio and Newspaper.

3.2 Reading and Understanding the Data:

- Warnings are suppressed.
- The libraries used: Numpy and Pandas are imported.
- Data is Visualized using matplotlib and seaborn.

	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

Table.1: Visualized Data Table

3.3 Data Inspection:

Data inspection meets this need: it is the act of viewing data for verification and debugging purposes, before, during, or after a translation

- The size and shape of the dataset is defined.
- The information of the dataset is described.

	TV	Radio	Newspaper	Sales
count	200.000000	200.000000	200.000000	200.000000
mean	147.042500	23.264000	30.554000	15.130500
std	85.854236	14.846809	21.778621	5.283892
min	0.700000	0.000000	0.300000	1.600000
25%	74.375000	9.975000	12.750000	11.000000
50%	149.750000	22.900000	25.750000	16.000000
75%	218.825000	36.525000	45.100000	19.050000
max	296.400000	49.600000	114.000000	27.000000

Table.2: Description Table of Dataset

3.4 Data Cleaning:

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled.

- The data is checked for null values.
- The data is checked for any outliers present.
- Since there are no considerable outliers present in the data, it is good to move to the next step.

3.5 Data Analysis:

The data is analyzed using many plotting techniques such as:

- Scatter plot- Shows the relationship between Sales and the other three data.
- Heat Map- Shows the correlation between different variables.

3.6 Model Building:

Performing Simple Linear Regression

Equation of linear regression

$$y = c + m_1x_1 + m_2x_2 + \dots + m_nx_n$$

Here,

- '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:

$$y = c + m_1 \times TV$$

The 'm' values are called the model **coefficients** or **model parameters**.

After we build and test the model, we perform a summary operation to list out all the different parameters of the regression like fitted.

1. The coefficient for TV is 0.054, with a very low p value

The coefficient is statistically significant. So the association is not purely by chance.

2. R - squared is 0.816

Meaning that 81.6% of the variance in "Sales" is explained by "TV"

This is a decent R-squared value.

3. F statistic has a very low p value (practically low)

Meaning that the model fit is statistically significant, and the explained variance isn't purely by chance.

3.7: Model fitting:

The fit is significant. Now let's visualize how well the model fits the data.

From the parameters that we get, our linear regression equation becomes:

$$Sales = 6.948 + 0.054 \times TV$$

Therefore, using a scatter plot, we visualize the model fitting as:

3.8 Model Evaluation:

Residual analysis

We need to validate assumptions of the model and the reliability for inference.

Distribution of the error terms

We need to check if the error terms are also normally distributed (which is in fact, one of the major assumptions of linear regression), let us plot the histogram of the error terms and see what it looks like.

The residuals are following the normally distributed, with a mean of 0. This means that it is alright to move onto the next step.

3.9 Predicting on a Test Set:

Now that we have fitted the model and inferred that it has decent predictive power, we can now use it to make some predictions on a test data.

The prediction on the test set returns us an RMSE:

```
In [27]: #Returns the mean squared error; we'll take a square root  
np.sqrt(mean_squared_error(y_test, y_pred))
```

```
Out[27]: 2.019296008966233
```

RMSE Value

And it returns an RS score:

```
In [28]: r_squared = r2_score(y_test, y_pred)
         r_squared
```

```
Out[28]: 0.7921031601245658
```

RS Score Value

Hence our prediction accuracy is approximately 79%.

3.10 Visualizing model fit on the test set:

Visualizing the model fit is an important step in understanding how well a machine learning model is performing.

Hence this shows us that the model fit isn't by chance, and that it works fine even in a test set.

CHAPTER 4

HARDWARE AND SOFTWARE REQUIREMENTS

4. REQUIREMENTS:

Minimum Hardware Requirements:

1. CPU: 4 cores
2. RAM: 8 GB
3. Storage: Minimum 5 GB available space for Jupyter Notebook.

Software Requirements:

1. 5 mbps uninterrupted Internet Connection.
2. Anaconda Navigator.
3. Jupyter Notebook as IDLE.
4. Python 3 (ipykernel)
5. 64-bit operating system, x64-based processor

CHAPTER 5

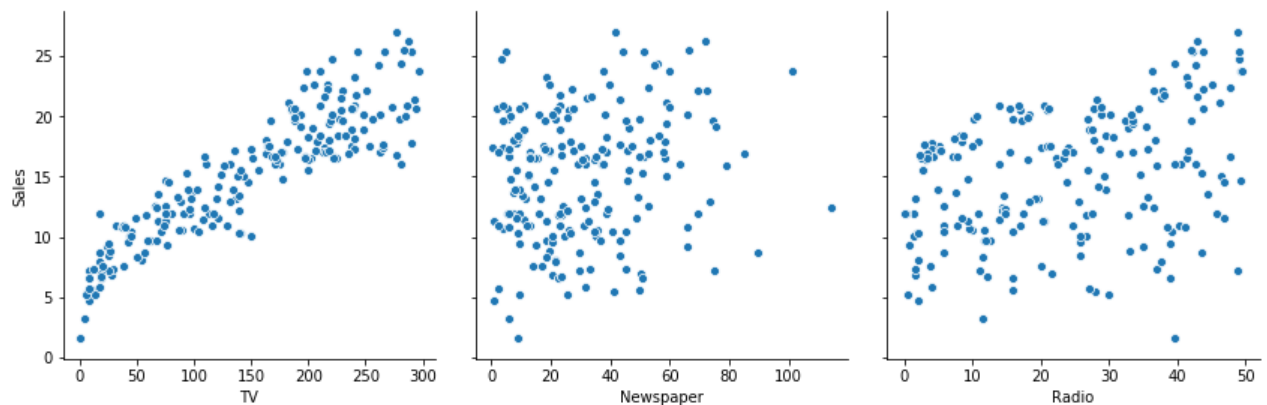
RESULTS AND CONCLUSION

5. RESULTS:

Training for the model workflows was done using Jupyter Notebook offered by Anaconda Navigator. In this project I have taken dataset from Kaggle , the dataset contains advertising data of a sales company with various platforms like TV, radio and newspaper . It consists about 200 data.

Configuration: Nvidia GeForce RTX 3050, Unspecified 8 CPU cores, 16 GB RAM

Model Training, testing and deployment was done on a local machine.



LIST OF FIGURES:

Dataset of Sales:

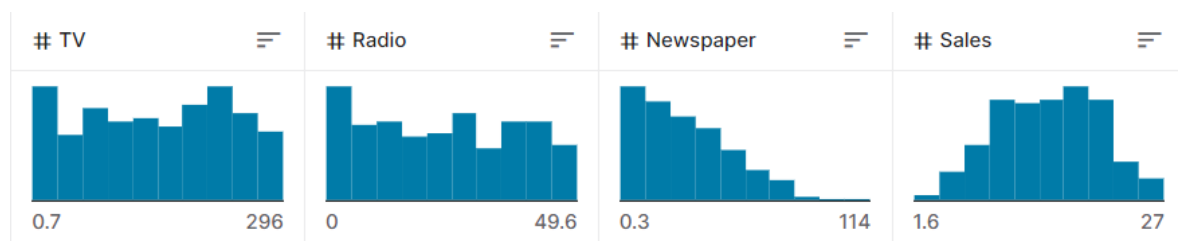


Fig.1: Dataset of sales

Reading and Understanding the Data:

	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

Fig.2: Visualized Data Table

Data Inspection:

	TV	Radio	Newspaper	Sales
count	200.000000	200.000000	200.000000	200.000000
mean	147.042500	23.264000	30.554000	15.130500
std	85.854236	14.846809	21.778621	5.283892
min	0.700000	0.000000	0.300000	1.600000
25%	74.375000	9.975000	12.750000	11.000000
50%	149.750000	22.900000	25.750000	16.000000
75%	218.825000	36.525000	45.100000	19.050000
max	296.400000	49.600000	114.000000	27.000000

Fig.3: Description Table of Dataset

Data Cleaning:

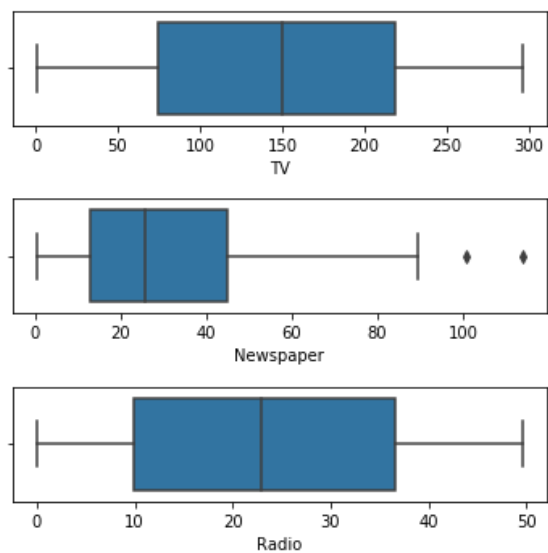


Fig.4: Data Outliers Box Plot

Data Analysis:

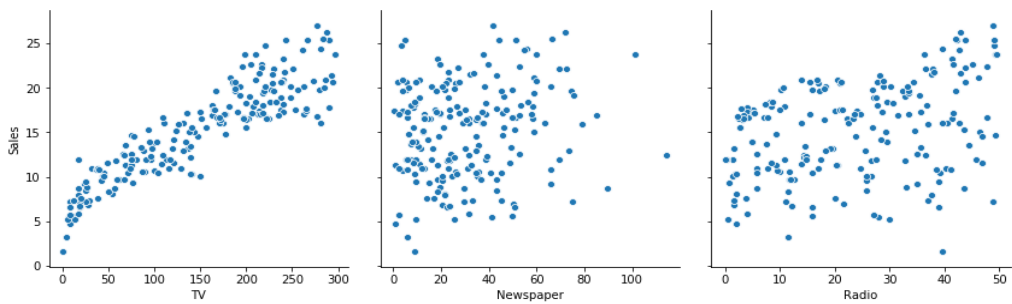


Fig.5: Scatter Plot

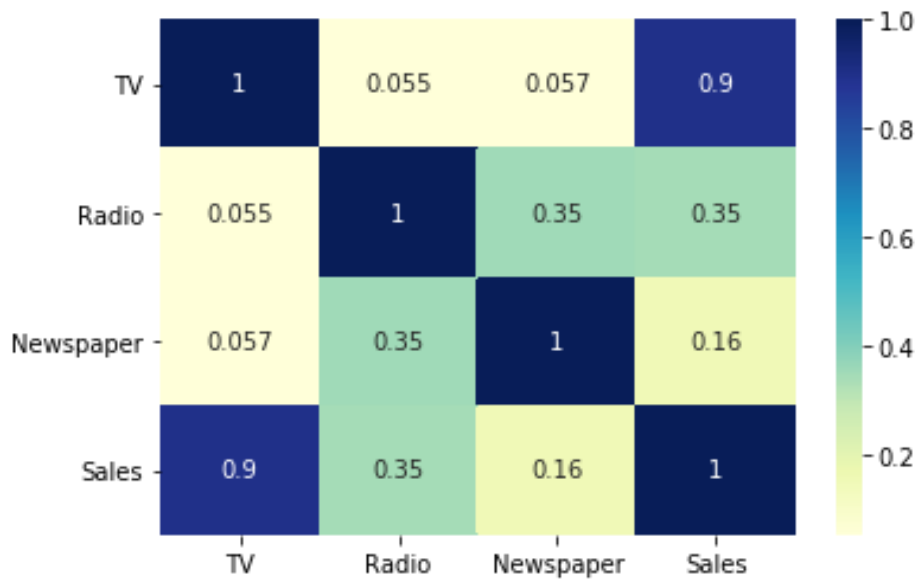


Fig.5.1: Heat Map

Model Building:

OLS Regression Results						
=====						
Dep. Variable:	Sales	R-squared:	0.816			
Model:	OLS	Adj. R-squared:	0.814			
Method:	Least Squares	F-statistic:	611.2			
Date:	Fri, 21 Apr 2023	Prob (F-statistic):	1.52e-52			
Time:	13:11:33	Log-Likelihood:	-321.12			
No. Observations:	140	AIC:	646.2			
Df Residuals:	138	BIC:	652.1			
Df Model:	1					
Covariance Type:	nonrobust					
=====						
	coef	std err	t	P> t	[0.025	0.975]

const	6.9487	0.385	18.068	0.000	6.188	7.709
TV	0.0545	0.002	24.722	0.000	0.050	0.059
=====						
Omnibus:	0.027	Durbin-Watson:	2.196			
Prob(Omnibus):	0.987	Jarque-Bera (JB):	0.150			
Skew:	-0.006	Prob(JB):	0.928			
Kurtosis:	2.840	Cond. No.	328.			
=====						

Fig.6: OLS Regression Summary

Model Fitting:

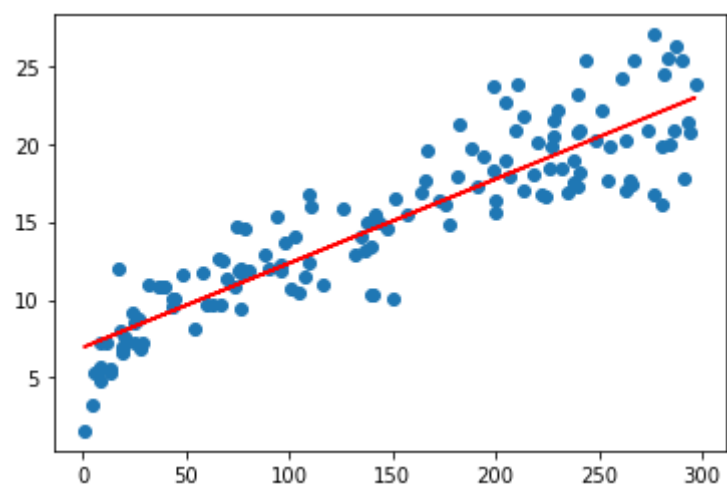


Fig.7: Model Building-Scatter Plot

Model Evaluation:

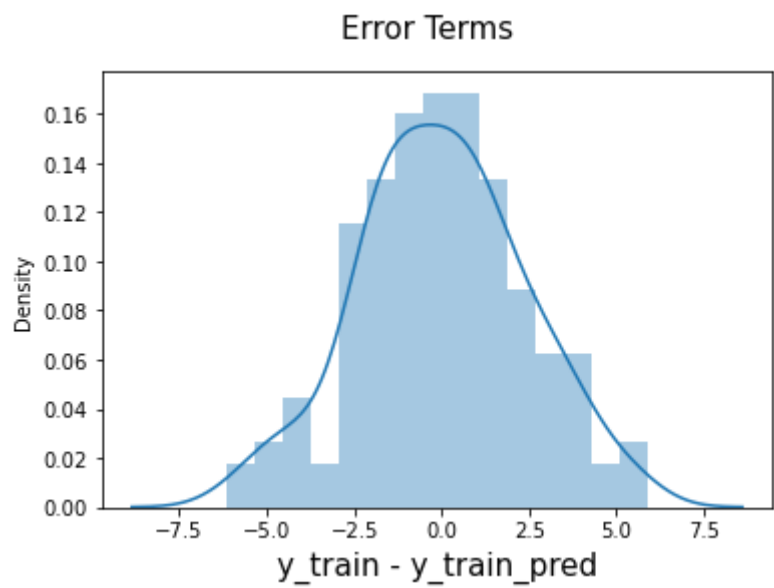


Fig.8: Histogram of Error Terms

Visualizing Model Fit on the Test Set:

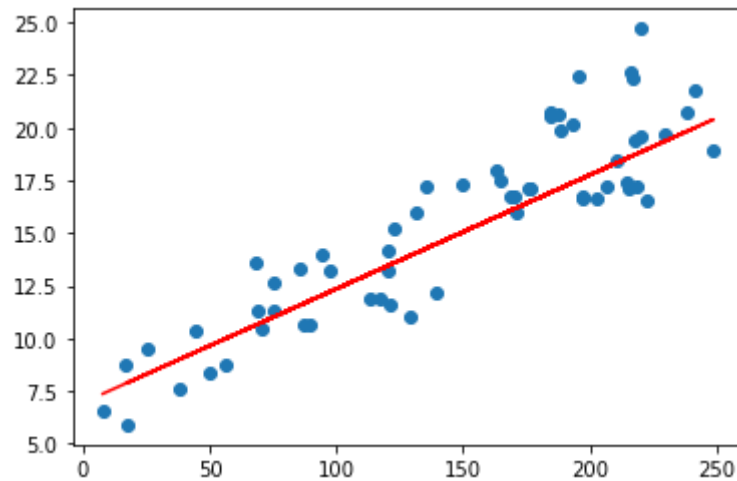


Fig.9: Model Fit on Test Set

5.1. Performance Analysis: -

Accuracy = $\frac{\text{No. of correct predictions}}{\text{Total no. of predictions}}$

PERFORMANCE	ACCURACY
Best Training Accuracy	~81.6%
Best Validation Accuracy	~79.2%
Real World Accuracy	~80%
Run Time	14.2 s

5.2. CONCLUSION

In this project, we have used linear regression algorithm to find the sales of a company using an old advertising data from various platforms like TV, newspaper and radio. This project uses some training model algorithms and plotting techniques for Sales Prediction such as Linear Regression algorithm which gives better efficiency to manipulate the trending sales analysis. The traditional approach of sales and marketing goals no longer help the companies, to cope up with the pace of competitive market, as they are carried out with no insights to customers' purchasing patterns.

In future, adding more algorithms to this project we can get more accuracy of the sales in a company that can help the owners to improve their sales and production in the organization. In the present world of digitally connected world, every shopping mall desires to know the customer demands in advance to avoid the shortfall of sale items in all seasons. Day to day, companies, malls and trend-based organizations are predicting the demand of product sales more accurately. Extensive research in this area at enterprise level is happening for accurate sales prediction.

CHAPTER 6

APPENDIX

6.1 SCREENSHOTS:

Reading and Understanding the Data

```
In [1]: # Suppress Warnings

import warnings
warnings.filterwarnings('ignore')

# Import the numpy and pandas package

import numpy as np
import pandas as pd

# Data Visualisation
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [2]: advertising = pd.DataFrame(pd.read_csv("advertising.csv"))
advertising.head()
```

```
Out[2]:
```

	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

Data Inspection

```
In [3]: advertising.shape
```

```
Out[3]: (200, 4)
```

```
In [4]: advertising.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
```

```
In [5]: advertising.describe()
```

```
Out[5]:
```

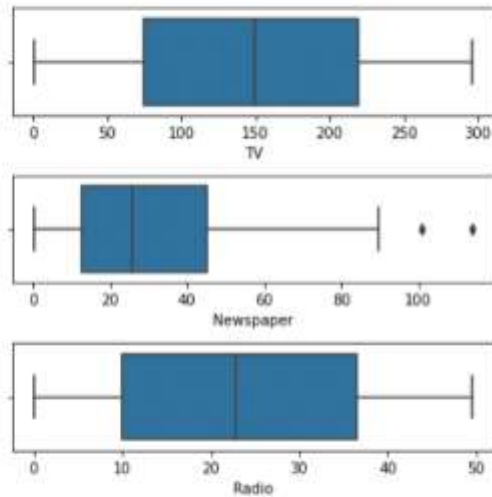
	TV	Radio	Newspaper	Sales
count	200.000000	200.000000	200.000000	200.000000
mean	147.042500	23.264000	30.554000	15.130500
std	85.854236	14.846809	21.778621	5.283892
min	0.700000	0.000000	0.300000	1.600000
25%	74.375000	9.975000	12.750000	11.000000
50%	149.750000	22.900000	25.750000	16.000000
75%	218.825000	36.525000	45.100000	19.050000
max	296.400000	49.600000	114.000000	27.000000

Data Cleaning

```
In [6]: # Checking Null values  
advertising.isnull().sum()*100/advertising.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  
Sales        0.0  
dtype: float64
```

```
In [7]: # Outlier Analysis
fig, axs = plt.subplots(3, figsize = (5,5))
plt1 = sns.boxplot(advertising['TV'], ax = axs[0])
plt2 = sns.boxplot(advertising['Newspaper'], ax = axs[1])
plt3 = sns.boxplot(advertising['Radio'], ax = axs[2])
plt.tight_layout()
```



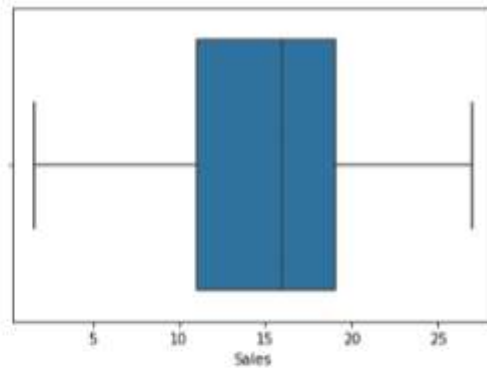
```
In [8]: # There are no considerable outliers present in the data.
```

Exploratory Data Analysis

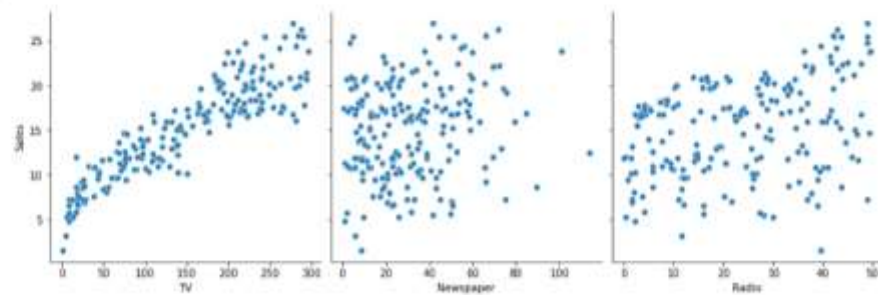
Univariate Analysis

Sales (Target Variable)

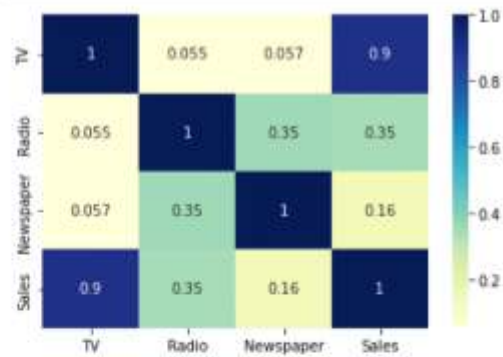
```
In [9]: sns.boxplot(advertising['Sales'])  
plt.show()
```



```
In [10]: # Let's see how Sales are related with other variables using scatter plot.  
sns.pairplot(advertising, x_vars=['TV', 'Newspaper', 'Radio'], y_vars='Sales', height=10)  
plt.show()
```



```
In [11]: # Let's see the correlation between different variables.
sns.heatmap(advertising.corr(), cmap="YlGnBu", annot = True)
plt.show()
```



Model Building

```
In [12]: X = advertising['TV']
y = advertising['Sales']
```

```
In [13]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size = 0.7, test_
```

```
In [14]: # Let's now take a look at the train dataset
X_train.head()
```

```
Out[14]: 74    213.4
3      151.5
185    205.0
26     142.9
90     134.3
Name: TV, dtype: float64
```

```
In [15]: y_train.head()
```

```
Out[15]: 74     17.0
3       16.5
185     22.6
26      15.0
90      14.0
Name: Sales, dtype: float64
```

```
In [16]: import statsmodels.api as sm
```

```
In [17]: # Add a constant to get an intercept
X_train_sm = sm.add_constant(X_train)

# Fit the regression line using 'OLS'
lr = sm.OLS(y_train, X_train_sm).fit()
```

```
In [18]: # Print the parameters, i.e. the intercept and the slope of the regression line
lr.params
```

```
Out[18]: const    6.948683
TV          0.054546
dtype: float64
```

```
In [19]: # Performing a summary operation lists out all the different parameters of the model
print(lr.summary())
```

```

=====
                        OLS Regression Results
=====
Dep. Variable:          Sales    R-squared:                0.816
Model:                  OLS      Adj. R-squared:           0.814
Method:                 Least Squares    F-statistic:           611.2
Date:                  Thu, 03 Nov 2022    Prob (F-statistic):      1.52e-52
Time:                  23:30:24    Log-Likelihood:         -321.12
No. Observations:      140      AIC:                    646.2
Df Residuals:          138      BIC:                    652.1
Df Model:               1
Covariance Type:       nonrobust
=====

```

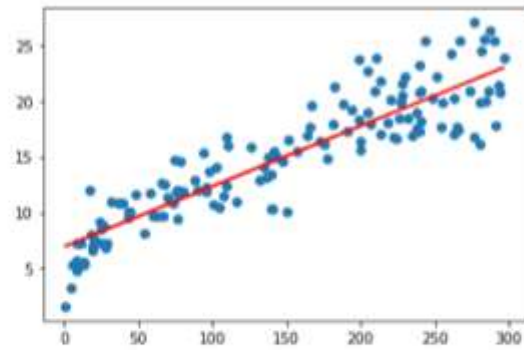
	coef	std err	t	P> t	[0.025	0.975]
const	6.9487	0.385	18.068	0.000	6.188	7.709
TV	0.0545	0.002	24.722	0.000	0.050	0.059

```
=====
Omnibus:                 0.027    Durbin-Watson:           2.196
Prob(Omnibus):            0.987    Jarque-Bera (JB):         0.150
Skew:                    -0.006    Prob(JB):                 0.928
Kurtosis:                 2.840    Cond. No.                  328.
=====
```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

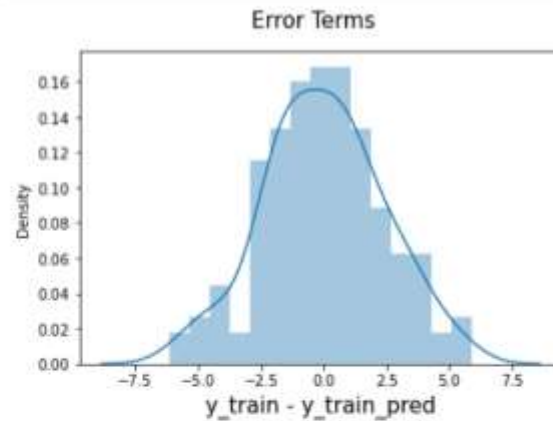
```
In [20]: plt.scatter(X_train, y_train)
plt.plot(X_train, 6.948 + 0.054*X_train, 'r')
plt.show()
```



Model Evaluation

```
In [21]: y_train_pred = lr.predict(X_train_sm)
res = (y_train - y_train_pred)
```

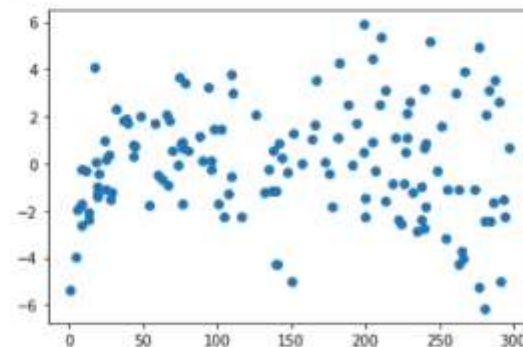
```
In [22]: fig = plt.figure()
sns.distplot(res, bins = 15)
fig.suptitle('Error Terms', fontsize = 15)      # Plot heading
plt.xlabel('y_train - y_train_pred', fontsize = 15)  # X-Label
plt.show()
```



The residuals are following the normally distributed with a mean 0.

Looking for patterns in the residuals

```
In [23]: plt.scatter(X_train, res)
plt.show()
```



Therefore, we are confident that the model fit isn't by chance, and has decent predictive power. The normality of residual terms allows some inference on the coefficients.

Although, the variance of residuals increasing with X indicates that there is significant variation that this model is unable to explain.

As you can see, the regression line is a pretty good fit to the data

Predictions on the Test Set

```
In [24]: # Add a constant to X_test
X_test_sm = sm.add_constant(X_test)

# Predict the y values corresponding to X_test_sm
y_pred = lr.predict(X_test_sm)
```

```
In [25]: y_pred.head()
```

```
Out[25]: 126    7.374140
        104    19.941482
         99    14.323269
         92    18.823294
        111    20.132392
        dtype: float64
```

```
In [26]: from sklearn.metrics import mean_squared_error
        from sklearn.metrics import r2_score
```

Looking at the RMSE

```
In [27]: #Returns the mean squared error; we'll take a square root
np.sqrt(mean_squared_error(y_test, y_pred))
```

```
Out[27]: 2.019296008966233
```

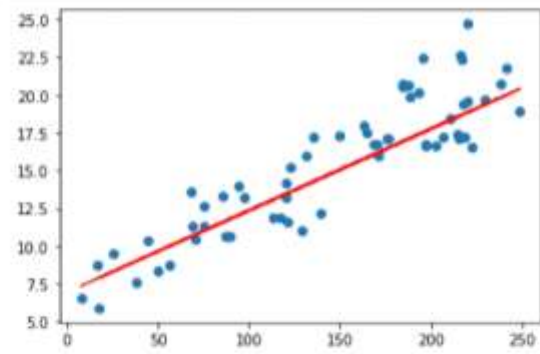
Checking the R-squared on the test set

```
In [28]: r_squared = r2_score(y_test, y_pred)
        r_squared
```

```
Out[28]: 0.7921031601245658
```

Visualizing the fit on the test set

```
In [29]: plt.scatter(X_test, y_test)
plt.plot(X_test, 6.948 + 0.054 * X_test, 'r')
plt.show()
```



7. SOURCE CODE:

```
# Supress Warnings
```

```
import warnings
warnings.filterwarnings('ignore')
```

```
# Import the numpy and pandas package
```

```
import numpy as np
import pandas as pd
```

```
# Data Visualisation
import matplotlib.pyplot as plt
import seaborn as sns
```

```
advertising = pd.DataFrame(pd.read_csv("advertising.csv"))
advertising.head()
```

```
advertising.shape
```

```
advertising.describe()
```

```
# Checking Null values
advertising.isnull().sum()*100/advertising.shape[0]
# There are no NULL values in the dataset, hence it is clean.
```

```
# Outlier Analysis
fig, axs = plt.subplots(3, figsize = (5,5))
plt1 = sns.boxplot(advertising['TV'], ax = axs[0])
plt2 = sns.boxplot(advertising['Newspaper'], ax = axs[1])
plt3 = sns.boxplot(advertising['Radio'], ax = axs[2])
plt.tight_layout()
```

```
sns.boxplot(advertising['Sales'])
plt.show()
```

```
# Let's see how Sales are related with other variables using scatter plot.
sns.pairplot(advertising, x_vars=['TV', 'Newspaper', 'Radio'], y_vars='Sales',
height=4, aspect=1, kind='scatter')
plt.show()
```

```
# Let's see the correlation between different variables.
sns.heatmap(advertising.corr(), cmap="YlGnBu", annot = True)
plt.show()
```

```
X = advertising['TV']
y = advertising['Sales']
```

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size = 0.7, test_size = 0.3, random_state = 100)
```

```
# Let's now take a look at the train dataset
```

```
X_train.head()
```

```
y_train.head()
```

```
import statsmodels.api as sm
```

```
# Add a constant to get an intercept
```

```
X_train_sm = sm.add_constant(X_train)
```

```
# Fit the regression line using 'OLS'
```

```
lr = sm.OLS(y_train, X_train_sm).fit()
```

```
# Print the parameters, i.e. the intercept and the slope of the regression line fitted  
lr.params
```

```
# Performing a summary operation lists out all the different parameters of the  
regression line fitted
```

```
print(lr.summary())
```

```
plt.scatter(X_train, y_train)
```

```
plt.plot(X_train, 6.948 + 0.054*X_train, 'r')
```

```
plt.show()
```

```
y_train_pred = lr.predict(X_train_sm)
```

```
res = (y_train - y_train_pred)
```

```
fig = plt.figure()
```

```
sns.distplot(res, bins = 15)
```

```
fig.suptitle('Error Terms', fontsize = 15)
```

```
# Plot heading
```

```
plt.xlabel('y_train - y_train_pred', fontsize = 15)
```

```
# X-label
```

```
plt.show()
```

```
plt.scatter(X_train, res)
```

```
plt.show()
```

```
# Add a constant to X_test
```

```
X_test_sm = sm.add_constant(X_test)
```

```
# Predict the y values corresponding to X_test_sm
```

```
y_pred = lr.predict(X_test_sm)
```

```
y_pred.head()
```

```
from sklearn.metrics import mean_squared_error
```

```
from sklearn.metrics import r2_score
```

```
#Returns the mean squared error; we'll take a square root
np.sqrt(mean_squared_error(y_test, y_pred))

r_squared = r2_score(y_test, y_pred)
r_squared

plt.scatter(X_test, y_test)
plt.plot(X_test, 6.948 + 0.054 * X_test, 'r')
plt.show()
```

6.3 REFERENCES:

- 1.) <https://www.ijcrt.org/>
- 2.) <https://www.kaggle.com/datasets/ashydv/advertising-dataset>
- 3.) <https://aip.scitation.org/>
- 4.) <https://aicourse.cognibot.ml/>
- 5.) <https://www.ijarsct.co.in/Paper7611.pdf>
- 6.) <https://pubs.aip.org/aip/acp/article/2387/1/140038/1000110/Sales-prediction-using-machine-learning-approaches>
- 7.) <https://ieeexplore.ieee.org/document/8777620>
- 8.) <https://www.analyticsvidhya.com/blog/2020/08/building-sales-prediction-web-application-using-machine-learning-dataset/>
- 9.) <https://medium.com/mlearning-ai/sales-prediction-using-a-linear-regression-model-ffeec84eede1>
- 10.) <https://towardsdatascience.com/5-machine-learning-techniques-for-sales-forecasting-598e4984b109>
- 11.) <https://www.neuraldesigner.com/blog/retail-store-sales-forecasting>
- 12.) <https://www.warse.org/IJATCSE/static/pdf/file/ijatcse951022021.pdf>
- 13.) <https://www.ritchieng.com/machine-learning-linear-regression/>
- 14.) <https://www.mdpi.com/2504-4990/4/1/6>
- 15.) <https://www.knowledgehut.com/blog/data-science/linear-regression-for-machine-learning>