Documentation on

Web Application to Predict and Analyze Sales

Project submitted towards partial fulfillment
of the requirements for the degree of

B.Tech in Information Technology

Submitted by
Aritra Banerjee (16500218055)
Indranuj Ganguly (16500218043)
Ranadip Ghosh (16500218030)
Rounak Das (16500218023)
Sarat Sen (16500218017)

Under the guidance of
Tuhina Sinha

Calcutta Institute of Engineering and Management
24, 1A, Chandi Ghosh Road, Ashok Nagar, Tollygunge, Kolkata, West Bengal 700040

Course affiliated to
Maulana Abul Kalam Azad University of Technology
(Formerly known as, West Bengal University of Technology)
Kolkata - 700064, India

B.Tech in Information Technology

Course affiliated to

Maulana Abul Kalam Azad University of Technology

(Formerly known as, West Bengal University of Technology)

Kolkata - 700064, India

CERTIFICATE OF RECOMMENDATION

This is to certify that the project entitled *"Web Application to Predict and Analyze Sales"* is a bona fide work carried out by Aritra Banerjee, Indranuj Ganguly, Ranadip Ghosh, Rounak Das & Sarat under our supervision and guidance for partial fulfillment of the requirements for the degree of B.Tech in Information Technology in Calcutta Institute of Engineering and Management , during the academic session 2018-2022.

SUPERVISOR/MENTOR

Calcutta Institute of Engineering and Management

Maulana Abul Kalam Azad University of Technology, Kolkata-700 040

Teacher-in-Charge/HoD

Calcutta Institute of Engineering and Management

Maulana Abul Kalam Azad University of Technology, Kolkata-700 040

DEAN/PRINCIPAL

Calcutta Institute of Engineering and Management

Maulana Abul Kalam Azad University of Technology, Kolkata-700 040

B.Tech in Information Technology

Course affiliated to

Maulana Abul Kalam Azad University of Technology

(Formerly known as, West Bengal University of Technology)

Kolkata - 700064, India

CERTIFICATE OF APPROVAL**

This foregoing documentation of the project is hereby approved as a credible study of an engineering subject carried out and presented in a manner satisfactory to warranty its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not endorse or approve any statement made or opinion expressed or conclusion drawn therein but approve the report only for purpose for which it has been submitted.

.

Committee of final examination

for evaluation of Project

** *Only in case the project-report is approved*

DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC ETHICS

We hereby declare that this project contains development and original research work by the undersigned candidate, as part of their B.Tech in Information Technology studies.

All information in this document has been obtained and presented in accordance with academic rules and ethical conduct.

We also declare that, as required by this rule and conduct, we have fully cited and referenced all materials and results that are not original to this work.

ARITRA BANERJEE (16500218055)
INDRANUJ GANGULY (16500218043)
RANADIP GHOSH (16500218030)
ROUNAK DAS (16500218023)
SARAT SEN (16500218017)

Project title: **WEB APPLICATION TO PREDICT AND ANALYZE SALES**
Demonstration: https://share.streamlit.io/virtualbookworm/wapas/main/app.py
Source code: https://github.com/VirtualBookWorm/WAPAS

SIGNATURE

DATE

LICENSE

The project is developed under open source license by all of the following developers & also made available virtually for further development & improvement with open source contribution. Detailed terms & conditions for further development, distribution of software and/or requirements are provided in the link, given below.

Refer to https://github.com/VirtualBookWorm/WAPAS/blob/main/LICENSE

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to my supervisor **Mrs. Tuhina Sinha** for her continuous support, guidance and valuable suggestions in our project work and related research. We are very much thankful to her for constant encouragement, motivation and patience during the entire project work.

Besides this, we would also like to thank **Prof. Samir Biswas** for his continuous encouragement and support during our entire period of study in ***Calcutta Institute of Engineering and Management***. I would like to express my gratitude to the entire staff, lab assistant and research workers for their cooperation and opinion.

Our thanks and appreciation goes to our classmates from B.Tech in Information Technology. We wish to thank all the departmental support staff and everyone else who has made different contributions in this dissertation.

Finally, we would like to thank our parents for always supporting me in every ups and downs during our entire period of work.

_____ *Date:*

ARITRA BANERJEE (16500218055)
INDRANUJ GANGULY (16500218043)
RANADIP GHOSH (16500218030)
ROUNAK DAS (16500218023)
SARAT SEN (16500218017)

B.Tech in Information Technology

Calcutta Institute of Engineering and Management
Maulana Abul Kalam Azad University of Technology, Kolkata – 700040

Contents

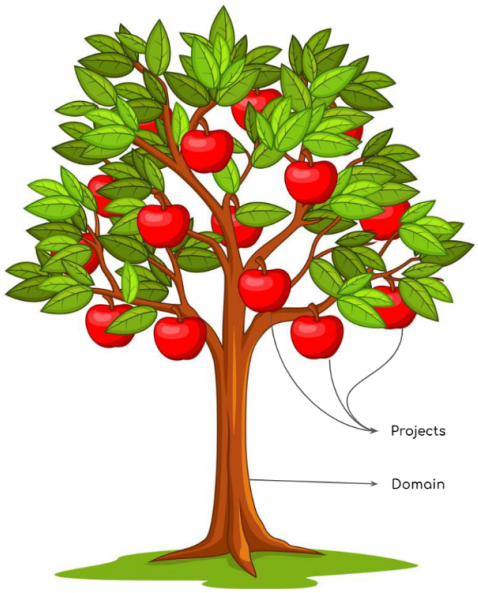
What is Sales Analytics?

- ➔ Data-driven decision making
- ➔ Commercially relevant insights
- ➔ Business estimates
- ➔ Cost analysis
- ➔ Consumer behavioral understanding
- ➔ Brand perceptions

....

How WAPAS stands out?

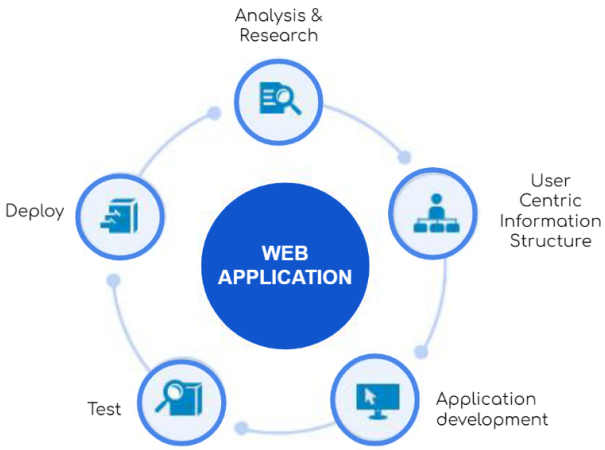
- ❑ Simple, clean & intuitive UI
- ❑ Data visualization feature
- ❑ Reliable accuracy
- ❑ Efficient business outlook
- ❑ Secured by Unique user ID
- ❑ Reusability & maintenance



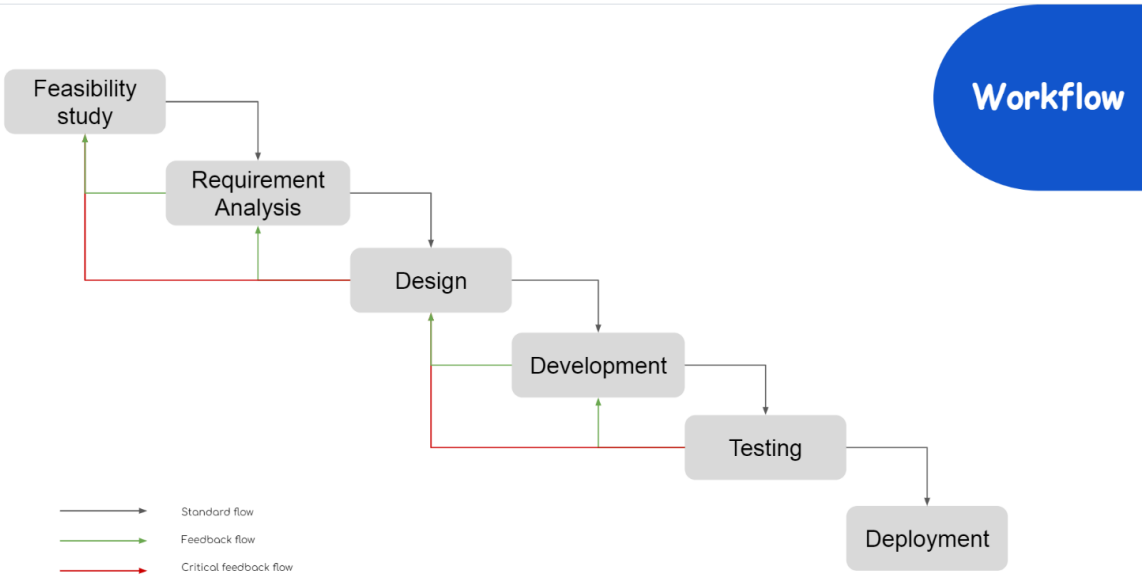
....

Why Web Application?

- User-friendly deployment
- Easily accessible
- Cross platform compatibility
- Comparatively less expensive
- Widespread manageability
- Simple authentication system

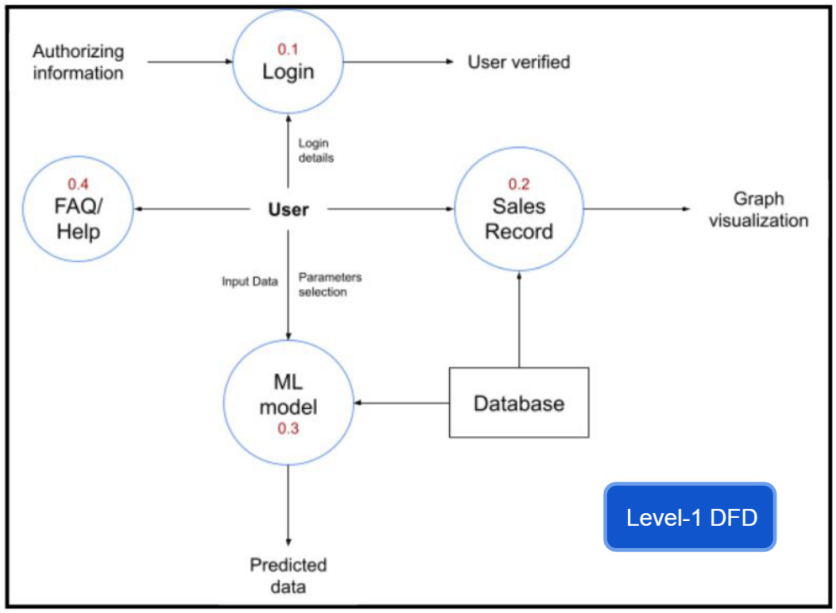


....

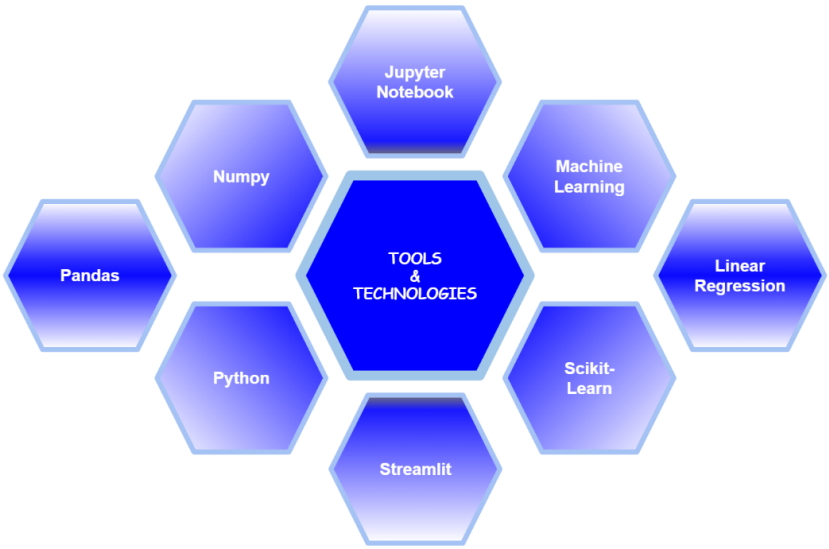


Royce's Waterfall model

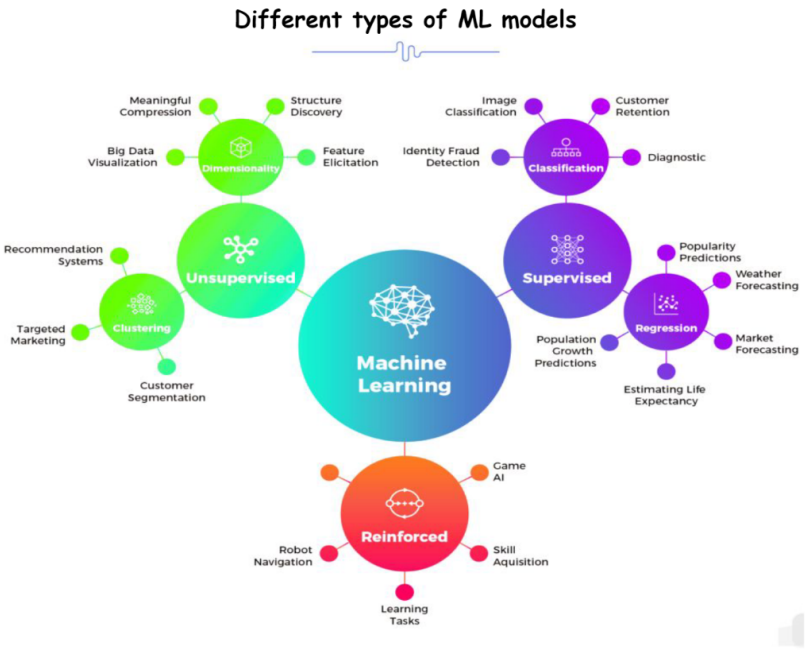
....


Authorizing information
0.1
Login
User verified
Login details
0.4
FAQ/ Help
User
0.2
Sales Record
Graph visualization
Input Data
Parameters selection
ML model
0.3
Database
Level-1 DFD
Predicted data

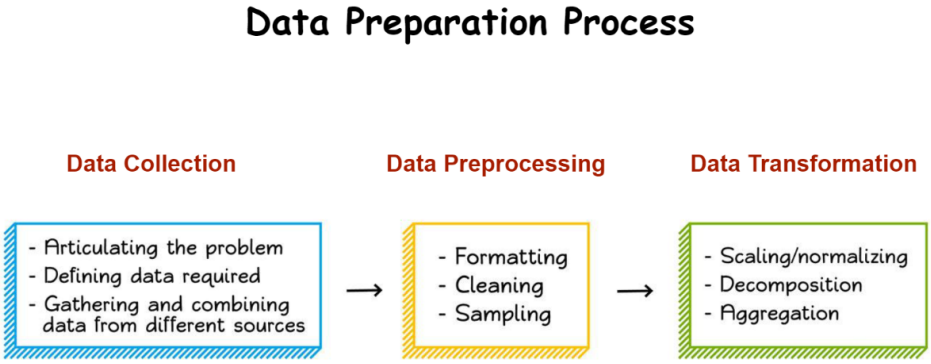
....


Jupyter Notebook
Numpy
Machine Learning
Pandas
TOOLS & TECHNOLOGIES
Linear Regression
Python
Scikit-Learn
Streamlit

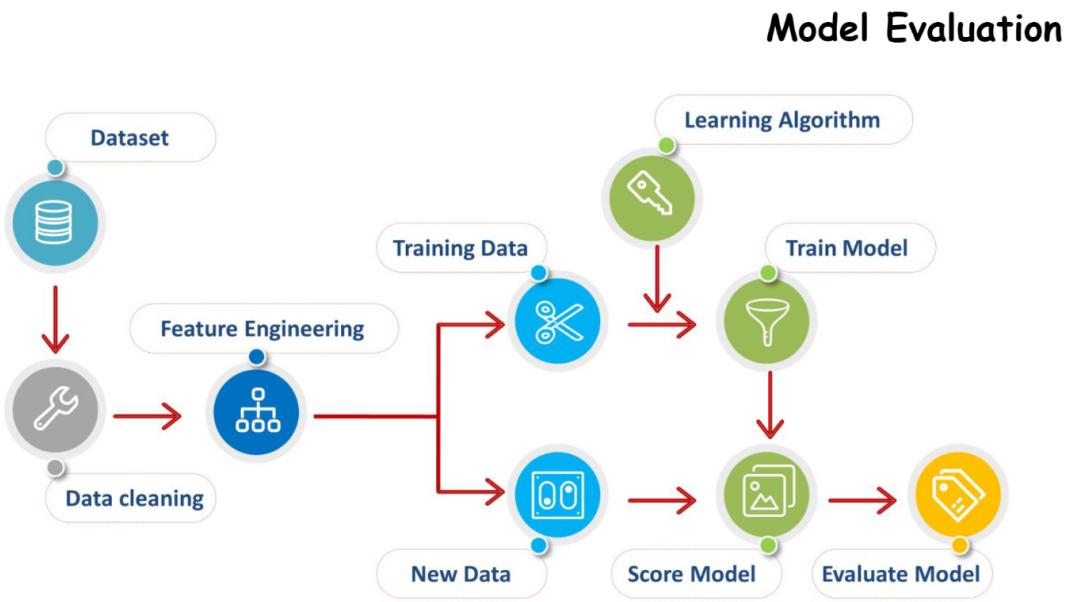
....


Different types of ML models
Machine Learning
Unsupervised
Dimensionality
Meaningful Compression
Structure Discovery
Big Data Visualization
Feature Elicitation
Clustering
Recommendation Systems
Targeted Marketing
Customer Segmentation
Supervised
Classification
Image Classification
Customer Retention
Identity Fraud Detection
Diagnostic
Regression
Popularity Predictions
Weather Forecasting
Population Growth Predictions
Market Forecasting
Estimating Life Expectancy
Reinforced
Game AI
Robot Navigation
Skill Aquisition
Learning Tasks

....


Data Preparation Process
Data Collection
Data Preprocessing
Data Transformation
- Articulating the problem
- Defining data required
- Gathering and combining data from different sources
- Formatting
- Cleaning
- Sampling
- Scaling/normalizing
- Decomposition
- Aggregation

....



....

Sales Prediction (Simple Linear Regression)

Problem Statement

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

Data

Use the advertising dataset given in ISLR and analyse the relationship between 'TV advertising' and 'sales' using a simple linear regression model.

In this notebook, we'll build a linear regression model to predict Sales using an appropriate predictor variable.

Reading and Understanding the Data

In [ ]:
```
# 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 [ ]:
```
df = pd.DataFrame(pd.read_csv("/content/advert_database.csv"))
df.head()
```

Out[ ]:

	TV	Radio	Newspaper	Sales

	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 [ ]:
```
df.shape
```

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

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

Out[ ]:

	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 [ ]:
```
# Checking Null values
df.isnull().sum()*100/df.shape[0]
```

Out[ ]:
```
TV           0.0
Radio        0.0
```

```
Newspaper    0.0
Sales        0.0
dtype: float64
```

There are no NULL values in the dataset, hence it is clean.

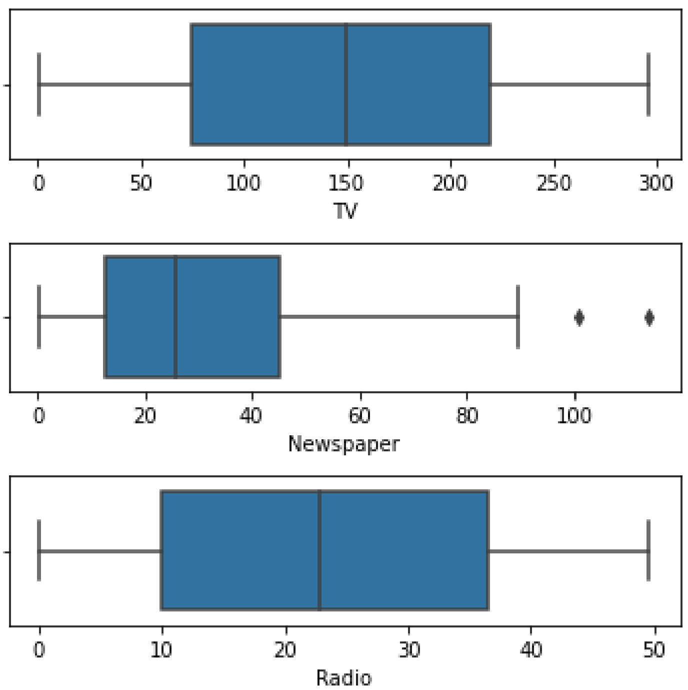
Exploratory Data Analysis

Univariate Analysis

In [ ]:

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

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pas
s the following variable as a keyword arg: x. From version 0.12, the only valid posi
tional argument will be `data`, and passing other arguments without an explicit keyw
ord will result in an error or misinterpretation.
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pas
s the following variable as a keyword arg: x. From version 0.12, the only valid posi
tional argument will be `data`, and passing other arguments without an explicit keyw
ord will result in an error or misinterpretation.
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pas
s the following variable as a keyword arg: x. From version 0.12, the only valid posi
tional argument will be `data`, and passing other arguments without an explicit keyw
ord will result in an error or misinterpretation.
  FutureWarning
```



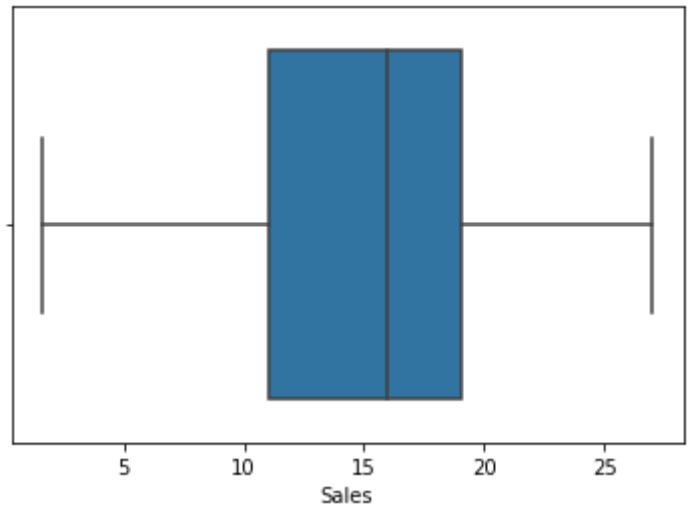
There are no considerable outliers present in the data.

In [ ]:

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

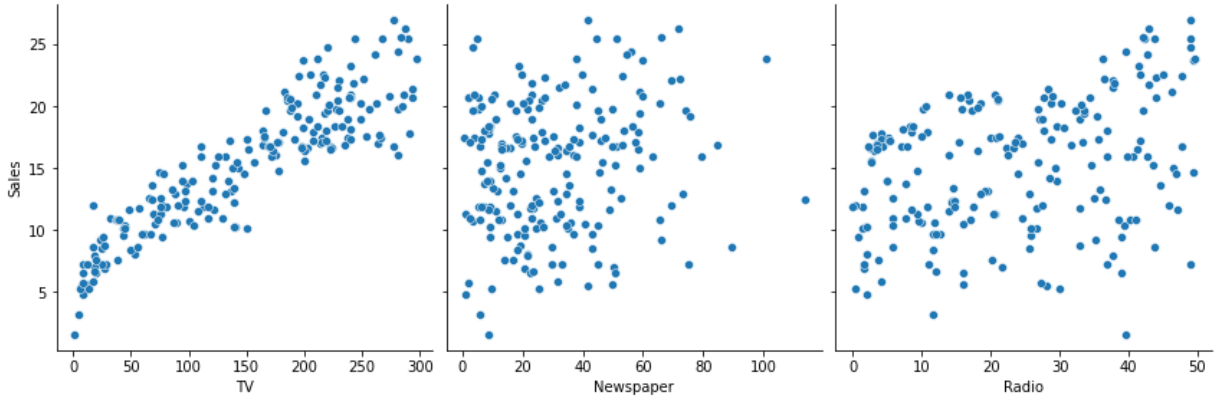
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pas
s the following variable as a keyword arg: x. From version 0.12, the only valid posi
tional argument will be `data`, and passing other arguments without an explicit keyw
```

```
ord will result in an error or misinterpretation.
  FutureWarning
```



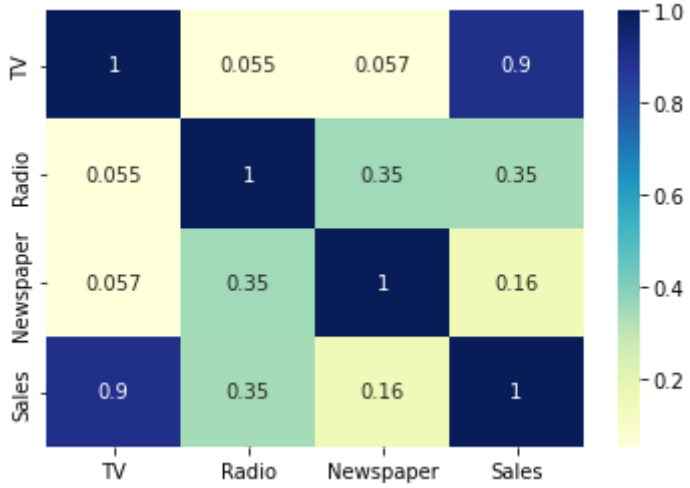
In [ ]:

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



In [ ]:

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



As is visible from the pairplot and the heatmap, the variable TV seems to be most correlated with Sales. So let's go ahead and perform simple linear regression using TV as our feature variable.

Model Building

Performing Simple Linear Regression

Equation of linear regression $y = c + m_1x_1 + m_2x_2 + \ldots + m_nx_n$ y is the response c is the intercept m_1 is the coefficient for the first feature m_n 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**.

Generic Steps in model building using *statsmodels*

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

In [ ]:
```
X = df['TV']
y = df['Sales']
```

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 70% of the data in your train dataset and the rest 30% in your test dataset

In [ ]:
```
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_siz
```

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

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

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

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

Building a Linear Model

You first need to import the statsmodel.api library using which you'll perform the linear regression.

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

By default, the statsmodels library fits a line on the dataset which passes through the origin. But in order to have an intercept, you need to manually use the add_constant attribute of statsmodels. And once you've added the constant to your X_train dataset, you can go ahead and fit a regression line using the OLS (Ordinary Least Squares) attribute of statsmodels as shown below

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

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

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/tsatools.py:117: FutureWarnin
g: In a future version of pandas all arguments of concat except for the argument 'ob
js' will be keyword-only
  x = pd.concat(x[::order], 1)
```

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

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

In [ ]:
```
# Performing a summary operation lists out all the different parameters of the regre
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:                Sun, 13 Feb 2022   Prob (F-statistic):           1.52e-52
Time:                        05:39:13   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.
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
```

Looking at some key statistics from the summary

The values we are concerned with are -

1. The coefficients and significance (p-values)

2. R-squared
3. F statistic and its significance

1. The coefficient for TV is 0.046, with a very low p value

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

2. R - squared is 0.613

Meaning that 61.3% of the variance in Sales is explained by TV

This is near to 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.

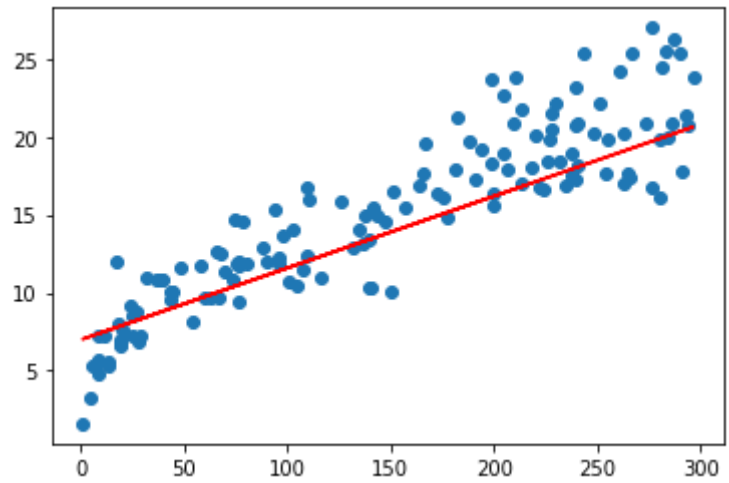
The fit is significant. Let's visualize how well the model fit the data.

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

$Sales = 6.989 + 0.046 \times TV$

Save model

In [ ]:
```
plt.scatter(X_train, y_train)
plt.plot(X_train, 6.989 + 0.046*X_train, 'r')
plt.show()
```



Model Evaluation

Residual analysis

To validate assumptions of the model, and hence the reliability for inference

Distribution of the error terms

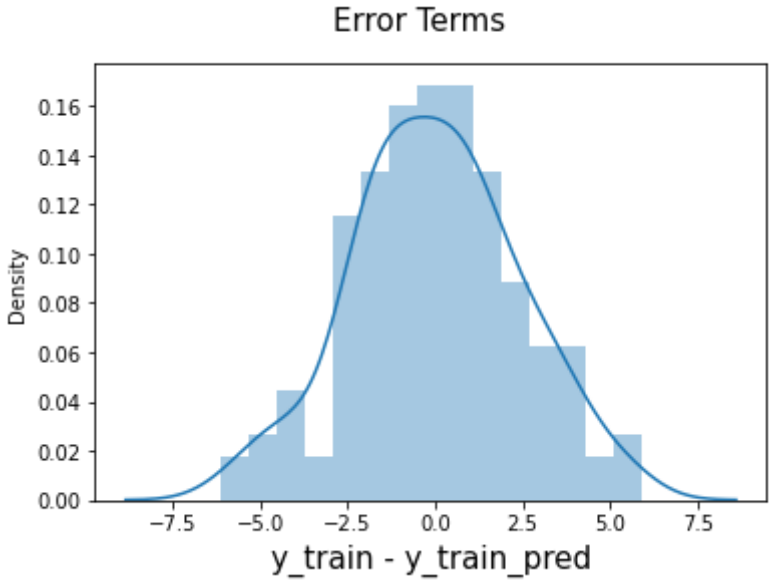
We need to check if the error terms are also normally distributed (which is infact, one of the major assumptions of linear regression), let us plot the histogram of the error terms and see

what it looks like.

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

In [ ]:
```
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()
```

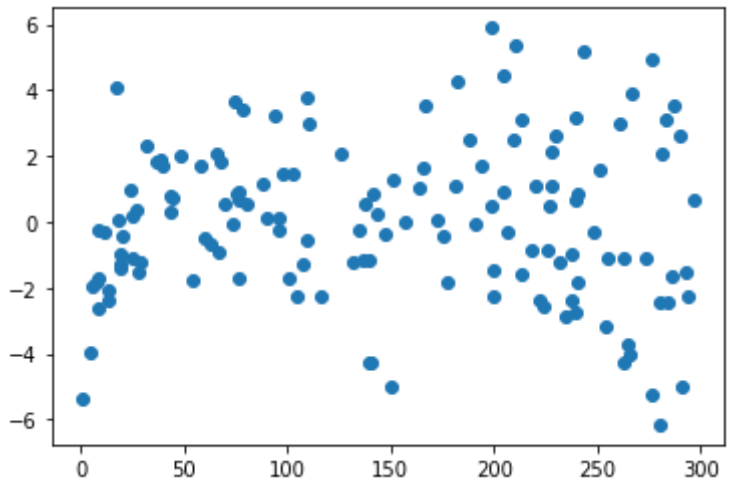
```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please
adapt your code to use either `displot` (a figure-level function with similar flexib
ility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



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

Looking for patterns in the residuals

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



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

Now that you have fitted a regression line on your train dataset, it's time to make some predictions on the test data. For this, you first need to add a constant to the X_test data like you did for X_train and then you can simply go on and predict the y values corresponding to X_test using the predict attribute of the fitted regression line.

In [ ]:
```
# 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)
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tsa/tsatools.py:117: FutureWarnin
g: In a future version of pandas all arguments of concat except for the argument 'ob
js' will be keyword-only
  x = pd.concat(x[::order], 1)
```

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

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

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

Looking at the RMSE

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

Out[ ]:
```
2.019296008966232
```

Checking the R-squared on the test set

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

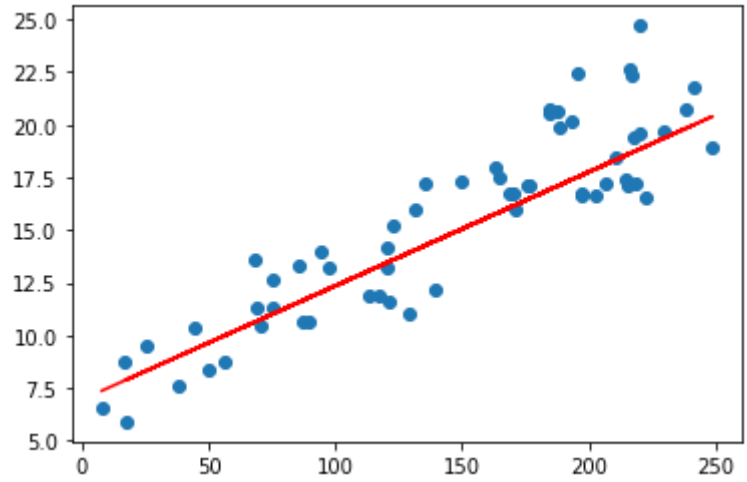
Out[ ]:
```
0.792103160124566
```

Therefore the model is showing 79% accuracy on the given data

Visualizing the fit on the test set

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

```
plt.show()
```



Save model

In [ ]:

```
import joblib

# Save the model as a pickle in a file
joblib.dump(lr, 'model.pkl')
```

Out[ ]:

```
['model.pkl']
```

Source code - Snapshot

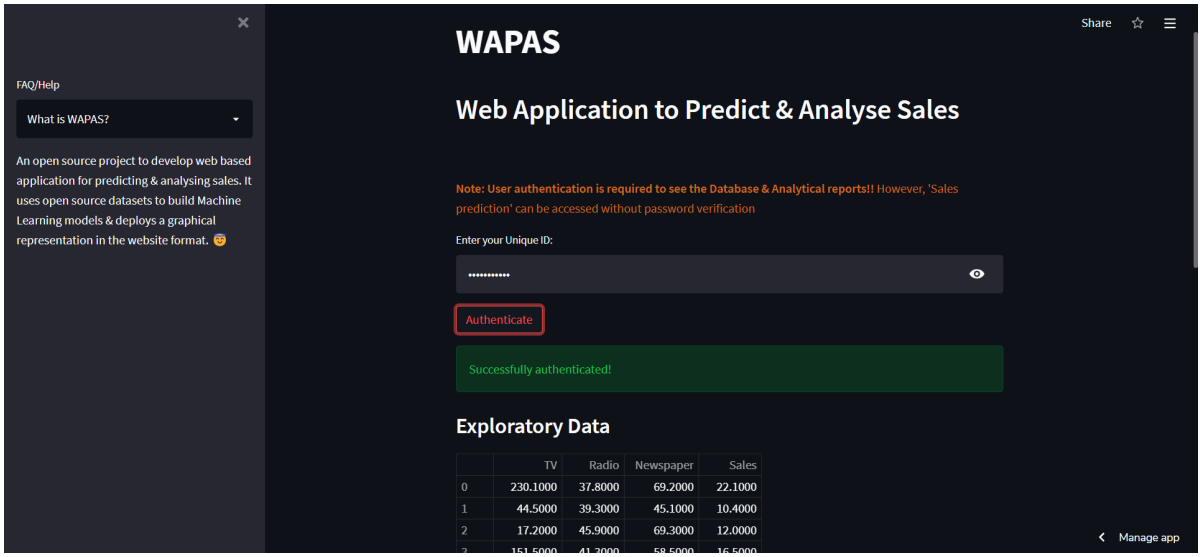
```
ID = st.text_input("Enter your Unique ID: ", type='password')
if st.button("Authenticate"):
        if ID==UID:
                st.success("Successfully authenticated!")
                data = load_data("assets/advert_database.csv")
                st.subheader("Exploratory Data")
                st.dataframe(data)
                with st.expander("See detailed plotting..."):
                        plot = px.scatter(data_frame=data, x='TV', y='Sales')
                        st.plotly_chart(plot)
                        plot = px.scatter(data_frame=data, x='Newspaper', y='Sales')
                        st.plotly_chart(plot)
                        plot = px.scatter(data_frame=data, x='Radio', y='Sales')
                        st.plotly_chart(plot)
                        st.write("***Note: Expenditure on X-axis in millions of $***")
                        st.write("***Note: Sales amount on Y-axis in Billions of $***")
                with st.expander("See comparative analysis..."):
                        st.image('assets/images/plots.png')
                        st.write("Therefore, **TV** adverts have the *highest/most correlative* weight...")
        else:
                st.error("Enter a valid Unique ID")

st.text("")
st.text("")

st.subheader("Predict sales")
choice = st.selectbox("Select expenditure method...", ['None', 'Advertisement on TV', 'Advertisement on Radio', 'Advertisement on Newspaper (not recommended)'])
if choice=='None':
        pass
if choice=='Advertisement on TV':
        TV = st.slider("Advertisement expenditure on TV (in millions of $)", 0, 300, 1)
        if st.button("Predict"):
                sample = np.array([0,TV]).reshape(2)
                model = load_model("models/model.pkl")
                prediction = model.predict(sample)
```

....

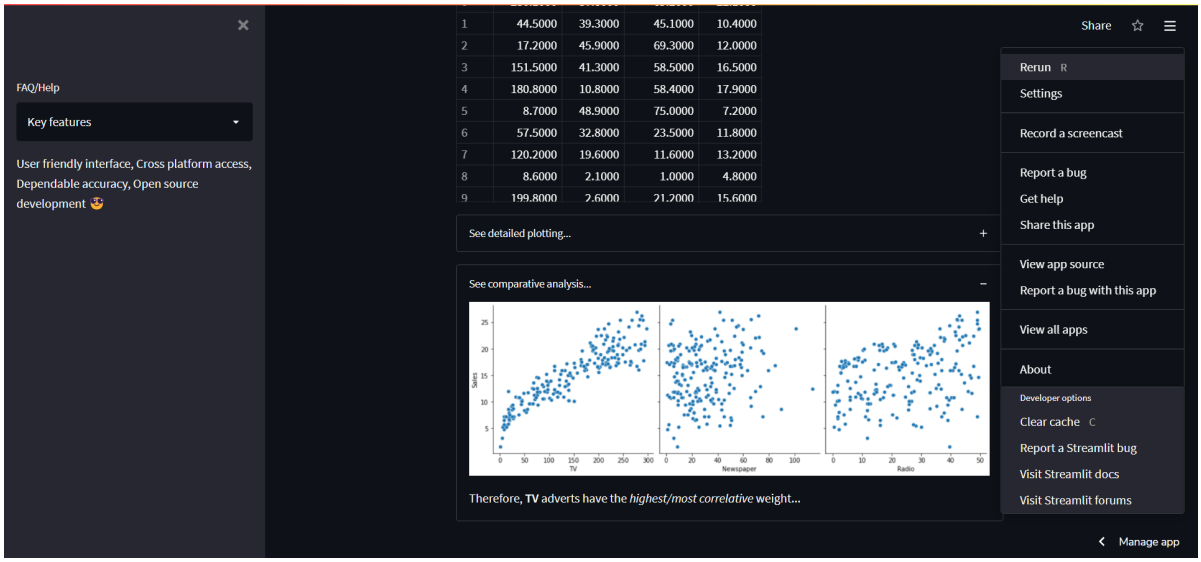
....

....

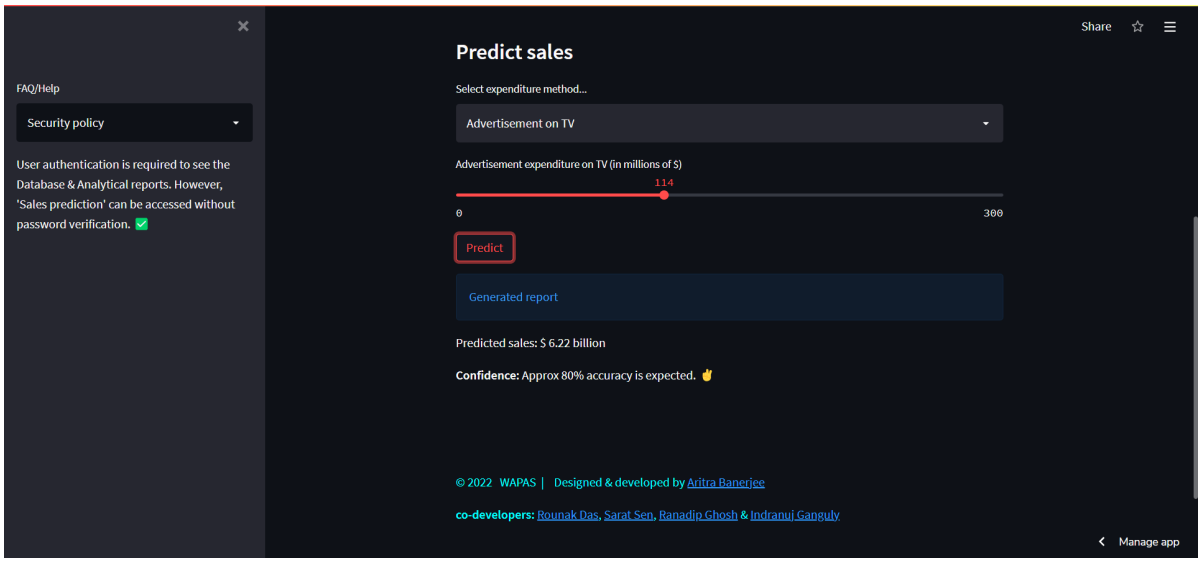
Web App - Snapshots


FAQ/Help
What is WAPAS?
An open source project to develop web based application for predicting & analysing sales. It uses open source datasets to build Machine Learning models & deploys a graphical representation in the website format.
Share
WAPAS
Web Application to Predict & Analyse Sales
Note: User authentication is required to see the Database & Analytical reports!! However, 'Sales prediction' can be accessed without password verification
Enter your Unique ID:
Authenticate
Successfully authenticated!
Exploratory Data
TV Radio Newspaper Sales
0 230.1000 37.8000 69.2000 22.1000
1 44.5000 39.3000 45.1000 10.4000
2 17.2000 45.9000 69.3000 12.0000
Manage app

....


FAQ/Help
Key features
User friendly interface, Cross platform access, Dependable accuracy, Open source development
1 44.5000 39.3000 45.1000 10.4000
2 17.2000 45.9000 69.3000 12.0000
3 151.5000 41.3000 58.5000 16.5000
4 180.8000 10.8000 58.4000 17.9000
5 8.7000 48.9000 75.0000 7.2000
6 57.5000 32.8000 23.5000 11.8000
7 120.2000 19.6000 11.6000 13.2000
8 8.6000 2.1000 1.0000 4.8000
9 199.8000 2.6000 21.2000 15.6000
See detailed plotting...
See comparative analysis...
Therefore, TV adverts have the highest/most correlative weight...
Share
Rerun R
Settings
Record a screencast
Report a bug
Get help
Share this app
View app source
Report a bug with this app
View all apps
About
Developer options
Clear cache C
Report a Streamlit bug
Visit Streamlit docs
Visit Streamlit forums
Manage app

....


FAQ/Help
Security policy
User authentication is required to see the Database & Analytical reports. However, 'Sales prediction' can be accessed without password verification.
Share
Predict sales
Select expenditure method...
Advertisement on TV
Advertisement expenditure on TV (in millions of $)
114
0
300
Predict
Generated report
Predicted sales: $ 6.22 billion
Confidence: Approx 80% accuracy is expected.
© 2022 WAPAS | Designed & developed by Aritra Banerjee
co-developers: Rounak Das, Sarat Sen, Ranadip Ghosh & Indranuj Ganguly
Manage app

....

Challenges faced

- Lack of clean data to directly work with might have slowed down our progress.
- The loss to value of information in a real scenario for sales is very high.
- Content based classification is just a part of the whole picture.
- Distinguish between sales fraud and actual sale.

....

Future scope

- ➔ **Accuracy increment**
- ➔ **Data verification**
- ➔ **Additional model deployment**
- ➔ **Relevant data sources**
- ➔ **Range optimization**
- ➔ **Metrics adjustment**
- ➔ **Cross platform development**
- ➔ **Business outlook improvement**
- ➔ **Experienced market research**

....

References

Dataset: www.kaggle.com/aungpyaeap/supermarket-sales

Waterfall model used: en.wikipedia.org/wiki/Waterfall_model

Standard ML pipeline: www.datanami.com/2018/09/05/how-to-build-a-better-machine-learning-pipeline/

Different types of ML models: www.rocketsource.co/blog/machine-learning-models/