Sipna Shikshan Prasarak Mandal's

SIPNA COLLEGE OF ENGINEERING & TECHNOLOGY, AMRAVATI

Accredited by **NAAC** with **Grade 'A'** | **NBA** Accredited
IAO Certified | **ISO** Certified

SIPNA COLLEGE OF ENGINEERING & TECHNOLOGY,

AMRAVATI

Department of Computer Science and Engineering

Academic Year: 2022-2023

Semester: Fifth

A PROJECT REPORT ON

DATA SCIENCE AND STATISTIC'S

Submitted for

Emerging Tech Lab:PE-1

Submitted in

December 2022

Under The Guidance Of

Prof.A.D.Shah

SIPNA COLLEGE OF ENGINEERING & TECHNOLOGY,

AMRAVATI

CERTIFICATE

This is to certify that this project report entitled

"DATA SCIENCE AND STATISTICS"

has been completed by the following students in the partial fulfillment of project work of the fifth semester, Department of Computer Science and Engineering , During the Academic Session of 2022-2023 This is the record of their work under my guidance and to my immense satisfaction.

Prof.A.D.Shah

Project Guide

Dr. V.K. Shandilya
HOD

Dept. Computer Science & Engineering

Submitted By:-

Group Members:-

Student Id	Name	Alloted Practical
21BE0011	Darshan Rahate	Practical 3,4
21BE0014	Amit Tayade	Practical 2
21BE0017	Nikhil Patil	Practical 5,6
21BE0073	Sarvesh Moharil	Practical 7,8

INDEX:

Aim:

Use Cases Study of different domain along with dataset.

Software Requirements : Python 3.10

Code:

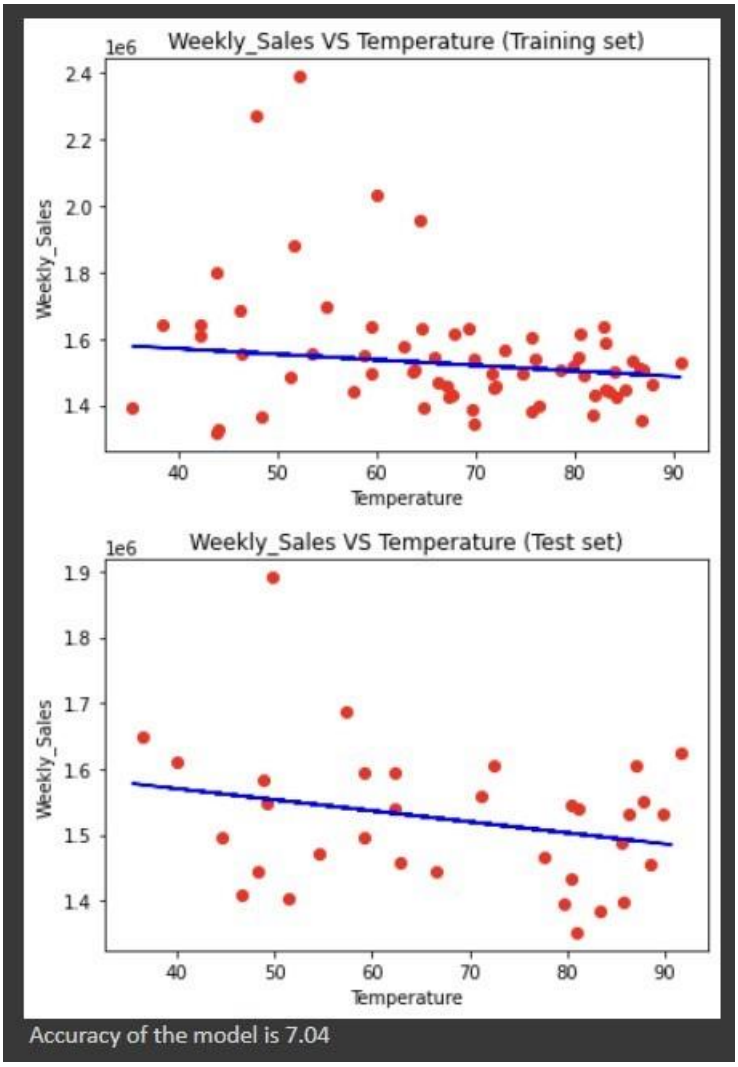
1. Algorithm Name : Linear Regression

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np #
Importing the dataset
dataset = pd.read_csv("/content/sample_data/Walmart.csv")
X = dataset.iloc[:, :-1].values #get a copy of dataset exclude last column
y = dataset.iloc[:, 1].values #get array of dataset in column 1st #

Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=1/3,
random_state=0)

"""
# Scaling
from sklearn.preprocessing import StandardScaler sc
_X = StandardScaler()
```

```
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
"""

# Fitting Simple Linear Regression to the Training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_test,y_test)

# Predicting the Test set results
y_pred = regressor.predict(X_test)

# Visualizing the Training set results
viz_train = plt
viz_train.scatter(X_train, y_train, color='red')
viz_train.plot(X_train, regressor.predict(X_train), color='blue')
viz_train.title('Weekly_sales VS Temperature (Training set)')
viz_train.xlabel('Year of Experience')
viz_train.ylabel('Weekly_sales')
viz_train.show()

# Visualizing the Test set results
viz_test = plt
viz_test.scatter(X_test, y_test, color='red')
viz_test.plot(X_train, regressor.predict(X_train), color='blue')
viz_test.title('Weekly_sales VS Temperature (Test set)')
viz_test.xlabel('Year of Experience')
viz_test.ylabel('Weekly_sales')
viz_test.show()

from sklearn.metrics import r2_score
```

```
r2_score(y_test,y_pred)
Accuracy=r2_score(y_test,y_pred)*100
print(" Accuracy of the model is %.2f" %Accuracy)
```

Output:



2. Algorithm Name : Multi-linear Regression

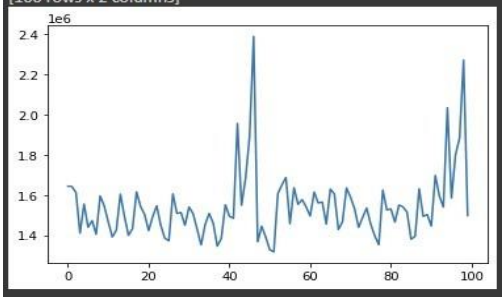
Code:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

df = pd.read_csv('/content/Walmart.csv ')
print(df)

# Here, profit is the only dependent var.
#separate the other attributes from the predicting
attribute x = df.drop('Weekly_sales',axis=1)
#separte the predicting attribute into Y for model
training y = df['Weekly_sales']

df['Weekly_sales'].plot(kind
= 'line ') plt.show()

# handle categorical variable
states=pd.get_dummies(x,drop_first=True)

from sklearn.model_selection import train_test_
split # splitting the data
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2,
random_state = 42)

from sklearn.linear_model import
LinearRegression LR = LinearRegression()
# fitting the training
data LR.fit(x_train,y_
train)
```

```
print("\n\n")
y_prediction =  LR.predict(x_test)
print(y_prediction)
print("\n")

from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
# predicting the accuracy score
score=r2_score(y_test,y_prediction)
print('r2 socre is ',score)
# print('mean_sqrd_error is
=',mean_squared_error(y_test,y_prediction))
# print('root_mean_squared error of is
=',np.sqrt(mean_squared_error(y_test,y_prediction)))

#Evaluate the model
from sklearn.metrics import r2_score
r2_score(y_test,y_prediction)
Accuracy=r2_score(y_test,y_prediction)*100
print(" Accuracy of the model is %.2f" %Accuracy)
```

Output:

```
   Temperature  Weekly_Sales
0        42.31   1643690.90
1        38.51   1641957.44
2        39.93   1611968.17
3        46.63   1409727.59
4        46.50   1554806.68
..         ...          ...
95       48.91   1584083.95
96       43.93   1799682.38
97       51.63   1881176.67
98       47.96   2270188.99
99       44.55   1497462.72

[100 rows x 2 columns]
```




```
[1520191.01848962 1625428.0885779  1501247.81450756 1589693.71360473
 1599019.18989884 1566047.91901281 1508288.41626807 1483154.79639846
 1545909.14114689 1609699.649928   1508421.25781072 1506348.92974536
 1494818.28384324 1507066.27407568 1545244.93343364 1598567.52865383
 1494233.78105557 1491417.54035137 1543013.1955171  1513044.14349498]


r2 socre is  0.3163344149803312
 Accuracy of the model is 31.63
```

3. Algorithm Name : Logistic Regression

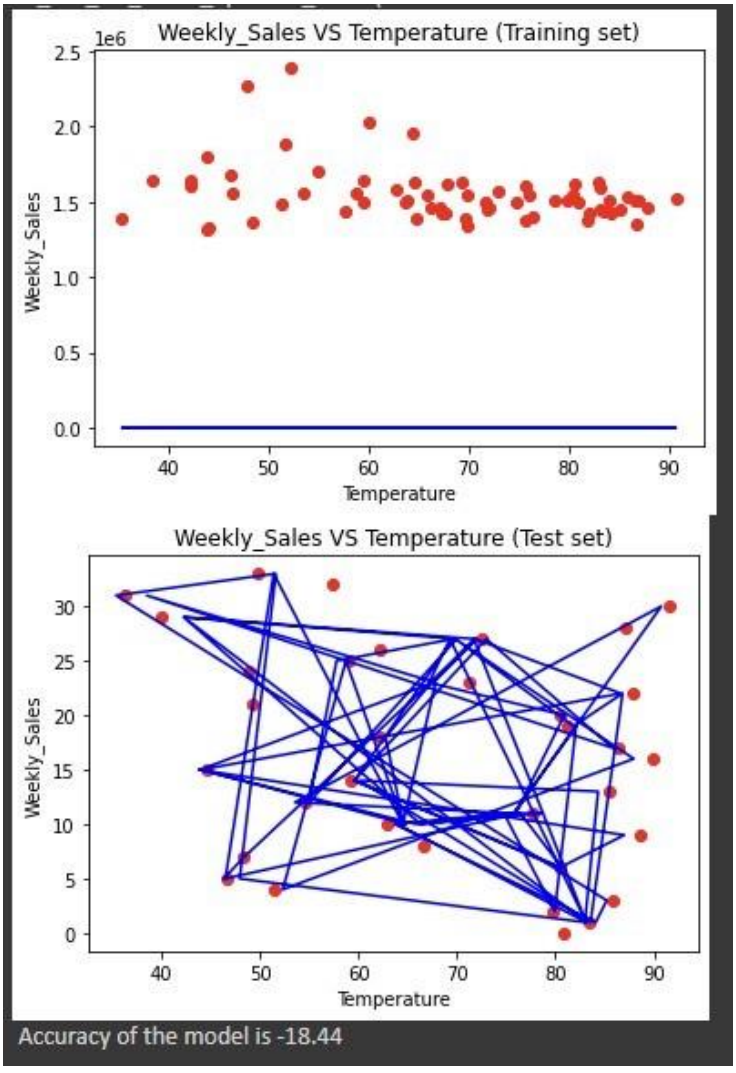
Code:

```
 import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv("/content/sample_data/Walmart.csv")
```

```
X = dataset.iloc[:, :-1].values #get a copy of dataset exclude last
column
y = dataset.iloc[:, 1].values #get array of dataset in column 1st

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=1/3,
random_state=0)

"""
# Scaling
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
"""

# Fitting Logistic Regression to the Training set
from sklearn.linear_model import LogisticRegression
regressor = LogisticRegression()
regressor.fit(X_test,y_test)

# Predicting the Test set results
y_pred = regressor.predict(X_test)

# Visualizing the Training set results
viz_train = plt
viz_train.scatter(X_train, y_train, color='red')
viz_train.plot(X_train, regressor.predict(X_train), color='blue')
viz_train.title('Weekly_sales VS Temperature (Training set)')
viz_train.ylabel('Weekly_sales')
viz_train.show()
```

```
# Visualizing the Test set results
viz_test = plt
viz_test.scatter(X_test, y_test, color='red')
viz_test.plot(X_train, regressor.predict(X_train), color='blue')
viz_test.title('Weekly_sales VS Temperature (Test set)')
viz_test.ylabel('Weekly_sales')
viz_test.show()

from sklearn.metrics import r2_score
r2_score(y_test,y_pred)
Accuracy=r2_score(y_test,y_pred)*100
print(" Accuracy of the model is %.2f" %Accuracy)
```

Output:


Weekly_Sales VS Temperature (Training set)
1e6
2.5
2.0
1.5
1.0
0.5
0.0
Weekly_Sales
40
50
60
70
80
90
Temperature
Weekly_Sales VS Temperature (Test set)
30
25
20
15
10
5
0
Weekly_Sales
40
50
60
70
80
90
Temperature
Accuracy of the model is -18.44

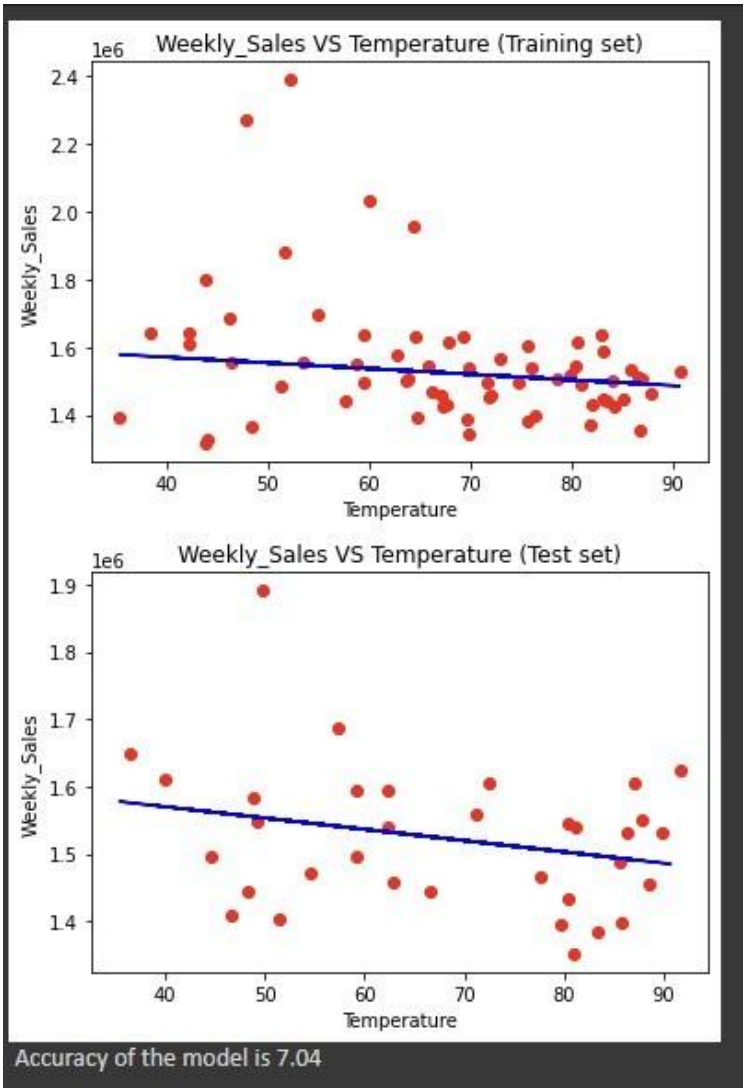
4. Algorithm Name : Ridge Regression

Code:

```
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv("/content/sample_data/Walmart.csv")
X = dataset.iloc[:, :-1].values #get a copy of dataset exclude last
y = dataset.iloc[:, 1].values #get array of dataset in column 1st

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=1/3,
random_state=0)

"""
# Scaling
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
"""

# Fitting Ridge Regression to the Training set
from sklearn.linear_model import Ridge
regressor = Ridge()
regressor.fit(X_test,y_test)
```

```
# Predicting the Test set results
y_pred = regressor.predict(X_test)

# Visualizing the Training set results
viz_train = plt
viz_train.scatter(X_train, y_train, color='red')
viz_train.plot(X_train, regressor.predict(X_train), color='blue')
viz_train.title('Weekly_sales VS Temperature (Training set)')
viz_train.ylabel('Weekly_sales')
viz_train.show()

# Visualizing the Test set results
viz_test = plt
viz_test.scatter(X_test, y_test, color='red')
viz_test.plot(X_train, regressor.predict(X_train), color='blue')
viz_test.title('Weekly_sales VS Temperature (Test set)')
viz_test.ylabel('Weekly_sales')
viz_test.show()

from sklearn.metrics import r2_score
r2_score(y_test,y_pred)
Accuracy=r2_score(y_test,y_pred)*100
print(" Accuracy of the model is %.2f" %Accuracy)
```

Output:


Weekly_Sales VS Temperature (Training set)
1e6
2.4
2.2
2.0
1.8
1.6
1.4
Weekly_Sales
40
50
60
70
80
90
Temperature
Weekly_Sales VS Temperature (Test set)
1e6
1.9
1.8
1.7
1.6
1.5
1.4
Weekly_Sales
40
50
60
70
80
90
Temperature
Accuracy of the model is 7.04

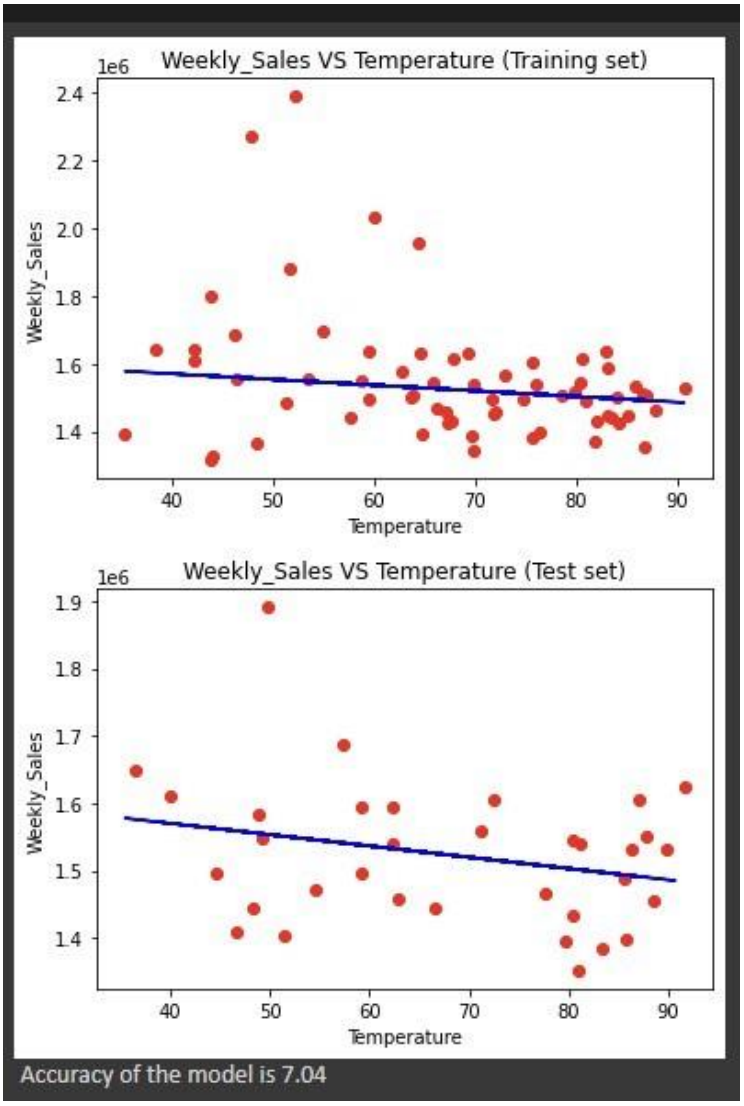
5. Algorithm Name : Lasso Regression

Code:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv("/content/sample_data/Walmart.csv")
X = dataset.iloc[:, :-1].values #get a copy of dataset exclude last
y = dataset.iloc[:, 1].values #get array of dataset in column 1st

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=1/3,
random_state=0)

"""
# Scaling
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
"""

# Fitting Lasso Regression to the Training set
from sklearn.linear_model import Lasso
```

```
regressor = Lasso()
regressor.fit(X_test,y_test)

# Predicting the Test set results
y_pred = regressor.predict(X_test)

# Visualizing the Training set results
viz_train = plt
viz_train.scatter(X_train, y_train, color='red')
viz_train.plot(X_train, regressor.predict(X_train), color='blue')
viz_train.title('Weekly_sales VS Temperature (Training set)')
viz_train.ylabel('Weekly_sales')
viz_train.show()

# Visualizing the Test set results
viz_test = plt
viz_test.scatter(X_test, y_test, color='red')
viz_test.plot(X_train, regressor.predict(X_train), color='blue')
viz_test.title('Weekly_sales VS Temperature (Test set)')
viz_test.ylabel('Weekly_sales')
viz_test.show()

from sklearn.metrics import r2_score
r2_score(y_test,y_pred)
Accuracy=r2_score(y_test,y_pred)*100
print(" Accuracy of the model is %.2f" %Accuracy)
```

Output:


Weekly_Sales VS Temperature (Training set)
1e6
2.4
2.2
2.0
1.8
1.6
1.4
Weekly_Sales
40
50
60
70
80
90
Temperature
Weekly_Sales VS Temperature (Test set)
1e6
1.9
1.8
1.7
1.6
1.5
1.4
Weekly_Sales
40
50
60
70
80
90
Temperature
Accuracy of the model is 7.04

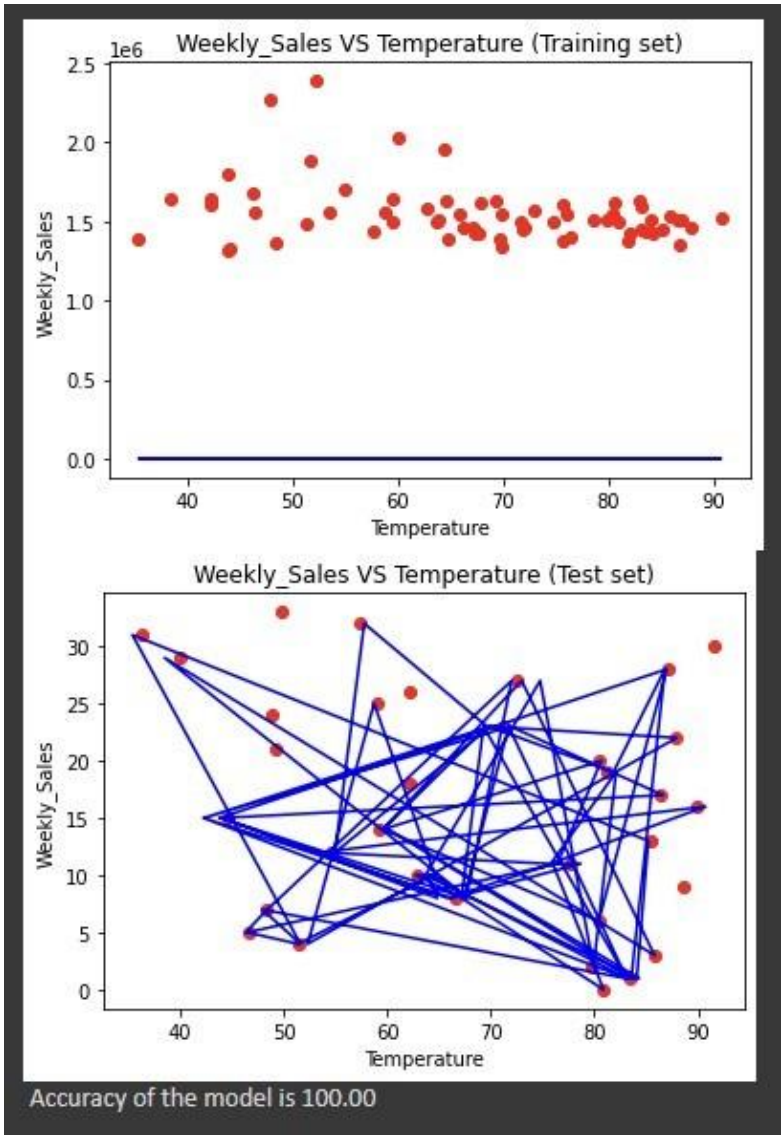
6. Algorithm Name : Decision Tree

Code:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv("/content/sample_data/Walmart.csv")
X = dataset.iloc[:, :-1].values #get a copy of dataset exclude last
y = dataset.iloc[:, 1].values #get array of dataset in column 1st

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=1/3,
random_state=0)

"""
# Scaling
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
"""

# Fitting Decision tree to the Training set
from sklearn.tree import DecisionTreeClassifier
```

```
regressor = DecisionTreeClassifier()
regressor.fit(X_test,y_test)

# Predicting the Test set results
y_pred = regressor.predict(X_test)

# Visualizing the Training set results
viz_train = plt
viz_train.scatter(X_train, y_train, color='red')
viz_train.plot(X_train, regressor.predict(X_train), color='blue')
viz_train.title('Weekly_sales VS Temperature (Training set)')
viz_train.ylabel('Weekly_sales')
viz_train.show()

# Visualizing the Test set results
viz_test = plt
viz_test.scatter(X_test, y_test, color='red')
viz_test.plot(X_train, regressor.predict(X_train), color='blue')
viz_test.title('Weekly_sales VS Temperature (Test set)')
viz_test.ylabel('Weekly_sales')
viz_test.show()

from sklearn.metrics import r2_score
r2_score(y_test,y_pred)
Accuracy=r2_score(y_test,y_pred)*100
print(" Accuracy of the model is %.2f" %Accuracy)
```

Output:


Weekly_Sales VS Temperature (Training set)
1e6
2.5
2.0
1.5
1.0
0.5
0.0
Weekly_Sales
40
50
60
70
80
90
Temperature
Weekly_Sales VS Temperature (Test set)
30
25
20
15
10
5
0
Weekly_Sales
40
50
60
70
80
90
Temperature
Accuracy of the model is 100.00

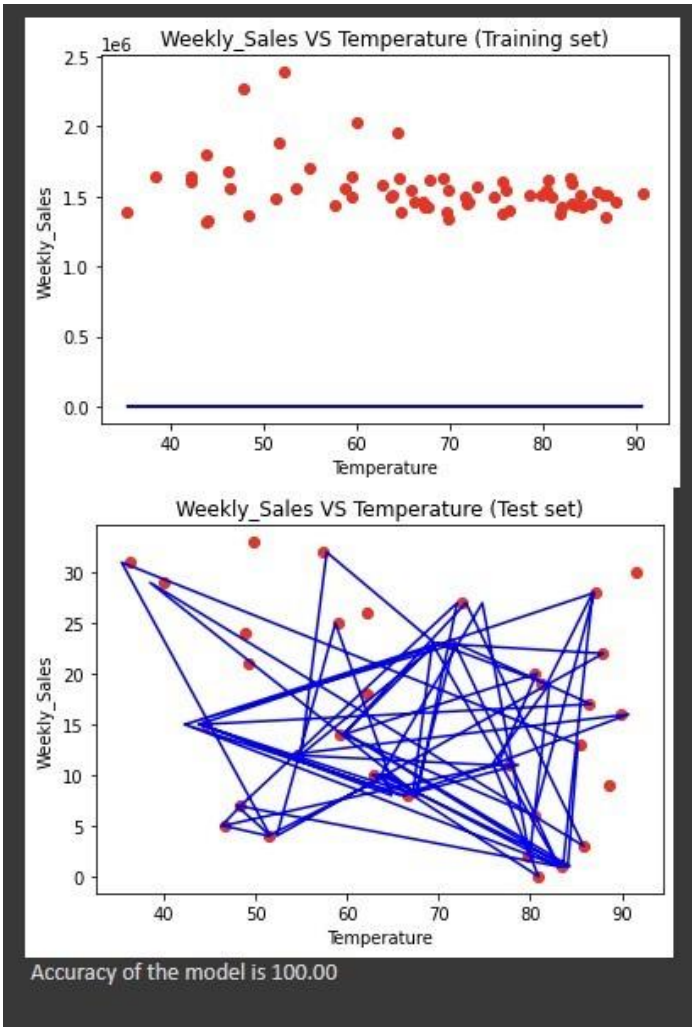
7. Algorithm Name : SVM

Code:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv("/content/sample_data/Walmart.csv")
X = dataset.iloc[:, :-1].values #get a copy of dataset exclude last
y = dataset.iloc[:, 1].values #get array of dataset in column 1st

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=1/3,
random_state=0)

"""
# Scaling
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
"""

# Fitting SVM to the Training set
from sklearn.svm import SVC
regressor = SVC()
regressor.fit(X_test,y_test)
```

```
# Predicting the Test set results
y_pred = regressor.predict(X_test)

# Visualizing the Training set results
viz_train = plt
viz_train.scatter(X_train, y_train, color='red')
viz_train.plot(X_train, regressor.predict(X_train), color='blue')
viz_train.title('Weekly_sales VS Temperature (Training set)')
viz_train.ylabel('Weekly_sales')
viz_train.show()

# Visualizing the Test set results
viz_test = plt
viz_test.scatter(X_test, y_test, color='red')
viz_test.plot(X_train, regressor.predict(X_train), color='blue')
viz_test.title('Weekly_sales VS Temperature (Test set)')
viz_test.ylabel('Weekly_sales')
viz_test.show()

from sklearn.metrics import r2_score
r2_score(y_test,y_pred)
Accuracy=r2_score(y_test,y_pred)*100
print(" Accuracy of the model is %.2f" %Accuracy)
```

Output:


Weekly_Sales VS Temperature (Training set)
1e6
2.5
2.0
1.5
1.0
0.5
0.0
Weekly_Sales
40
50
60
70
80
90
Temperature
Weekly_Sales VS Temperature (Test set)
30
25
20
15
10
5
0
Weekly_Sales
40
50
60
70
80
90
Temperature
Accuracy of the model is 100.00

Analysis Table:

Algorithm Name	Practical Dataset	Project Dataset (Walmart.csv)
Linear Regression	Accuracy: 97.96 Dataset:None	Accuracy: 7.04
Multilinear Regression	Accuracy: 89.87 Dataset:Multilinear.csv	Accuracy:31.63
Logistic Regression	Accuracy : 37.71 Dataset:None	Accuracy:-18.44
Ridge Regression	Accuracy: 97.95 Dataset:None	Accuracy:7.04
Lasso Regression	Accuracy:65.23 Dataset:mtcars.csv	Accuracy:7.04
Decision Tree	Accuracy:73.4 Dataset:user_data.csv	Accuracy:100.00
SVM Classifier	Accuracy:97.78 Dataset: IRIS.csv	Accuracy:100.00

Result :-

According to our sales dataset above all algorithm gives different Accuracy .Thus we have discussed and applied different algorithms on a single dataset in order to get the highest Accuracy.