1)

Aim :- To predict future Sales using Linear Regression based on advertising expenditure

Algorithm :-

1. Load dataset
2. Split data into train and test
3. Apply Linear Regression.
4. predict sales.
5. Display result

Program :-

```
import Pandas as pd
from sklearn.Linear_model import
Linear Regression
data = pd. read_csv ("Sales. csv")
X,y = data [['Advertising']].
data ['Sales']
model = Linear Regression ().fit (x,y)
```

Output:

[210.5 198.3 225.1 230.7]

Result: sales value are successfully predicted using Linear Regression

4)

Aim :- To classify data using Naive Bayes algorithm.

Algorithm :-

1. Load dataset
2. Train Naive Bayes classifier
3. Predict results.
4. Display accuracy.

Program :-

```
from sklearn.datasets import load-iris
from sklearn.naive-bayes import gaussian
X,Y = load -iris (return-X-Y= True)
model = Gaussian NB(). fit(x,y)
print ("Accuracy"; , model-score (x,y)
```

Output :-
Accuracy : 0.95

Result :- Naive Bayes classifier successfully classifiers the dataset

3)

Aim:
To classify data using Logistic Regression

Algorithm:-

1. Load dataset
2. Train Logistic Regression model
3. Predict output
4. Display accuracy.

Program:-

```
from sklearn.datasets import load_iris
from sklearn.linear_model import
Logistic Regression
x,y = load_iris (return_x-y = True)
model = Logistic Regression (max-iter =200).fit(x,y)
Print ('Accuracy :', model.score (x,y)),
```

Output:-

Accuracy : 0.97

Result:-
Logistic Regression model classifies data with high accuracy.

2) **Aim :-** To find the hypothises consistent with given training example.

Algorithm :-

1. Initialize S with first positive example.
2. Initialize G with most general hypothesis
3. update S & G for each example
4. output final S & G.

Program :-

```
S = ['small', Red, 'circle'].
data = [['small', 'Red', 'circle', 'Yes'),
        ['Big', 'Blue', 'circle', No]]
for x in data:
    for i in range(3):
        if x[3] == 'Yes' & x[i] != s[i]:
s[i] = '?'
Print ('S:",s)
```

output :-

S: [small, '?', circle]

Result : final hypothsis correctly represents all positive examples