model pratecol exam

192425047
SET-8

Y. siddartha
Achari

(1)

Aim: To build a linear regression model to predict future sales.

Algorithm:-

- load the dataset and check basic details
- Handle missing values and visualize data.
- split data into training and testing sets
- Train linear Regression model and predict sales

code:

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
df = pd.read_csv("sales.csv")
Print(df.head())
Print(df.describe())
Print (df.dtypes)
df.scatter(df['Advertising'], df['sales'])
Plt.xlabel ("Advertising")
Plt.ylabel ("sales")
Plt.show()
x = df(['Advertising'])
y = df ['sales']
```

```
model = linear Regression()
model.fit (x_train, y_train)
Pred = model.predict(x_test)
Print
Print ("predictions:", pred)
```

output:-

Predection: [120.5 134.2 150.7]

Result: The model successfully predicts sales based on advertising.

(2)

Aim:- To find the best hypotheses using condedate elimination algorithm.

Algorithm:-

* initialize specific and general hypotheses
* For positive example, specialize
* for negative example, specialize G.
* output final S and G

Code:

```
import numpy as np
data=np.array([['small', 'Red', 'circle', 'yes'], ['small', 'Blue', 'circle', 'yes']])
S = data [0,:-1].copy()
 for row in date:
          if row [-1] = = 'yes':
                  for i in range (len(s)):
                          if row [i] ! = S[i]:
                             S[i] = '?'
   Print(s)
```

output :-

```
['small' '?' 'circle']
```

Result:- The final hypotheses matches positive examples.

(3)

Aim:- To classify data using logistic Regression.

Algorithm:-

- load dataset and split into train / test
- Train logistic Regression model
- Predict test data
- measure accuracy

Code:-

```
from sklearn.dataset import load_iris
from sklearn.linear_model import logisticRegression

x,y = load_iris(return_x_y=True)

x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=02)
model = logisticRegression (max_iter=200)
model fit (x_train, y_train)
Print (model.score(x_test, y_test))
```

output:-

```
0.96
```

Result:- logistic Regression gives high classification Accuracy.

(4)

Aim:- To classify data using naive Bayes classifier

Algorithm:-

* load dataset and split data
* Train Naive Bayes model
* predict test data
* check accuracy.

code:-

```
Form sklearn. dataset import load_iris
From sklearn. naive_bayes import GaussianND
from sklean . model_ Selection import train_ test_ split
 X, y = load_iris (return_x_y= True)
X_train, X_ test, y_train, y_test = train_test_split (x,y, test_size =0.2)
 model= GaussianNB( )
model. fit (X_train, y_train)
Print (model. Score (x_test, y_test))
```

output:-

```
0.93
```

Result:- Naive Bayes classifier performs well on classification tasks.