MLA0202 - Machine Learning for Patterns Discovery

Model Practical

Monika P
192425059
B.Tech-AIML
SET-8

1. Linear Regression

Aim :- Predict Sales using Advertising

Algorithm :-

* Load data
* Show first rows, state, datatypes.
* Plot Scatter
* Check & report nulls with mode
* Split data into train / test
* Train Linear Regression model
* Predict test values
* Print predictions

Code :-

```
import pandas as pd
from sklearn.model_selection import
train_test_split
from sklearn.learn_model import
Linear Regression
df = pd.DataFrame ({
        'Sales' : [200, 250, 300, 350, 400],
        'Adv' : [20, 25, 30, 35, 40]
})
Print (df.head())
Print (df.describe())
```

```
print (df.dtypes)
print (df.isnull().sum())
df = df.fillna (df.mode().iloc[0])
x = df [['Adv']]
y = df ['Sales']
x_train, x_test, y_train, y_test = train_test_split
        (x, y, test_size = 0.2)
model = LinearRegression()
model.fit (x_train, y_train)
print ("Prediction:", model.predict (x_test))
```

Output :-

Prediction : [345.0]

Result :- Successfully found the Prediction is 345.0

2. Candidate - Elimination Algorithm.

Aim :- Find all consistent hypothesis

Algorithm :-

* Initialize S = most specific, G = most general.
* for each example :
  - If positive → generalize S and remove inconsistent G.
  - If negative → specilize G to exclude the example
* Remove inconsistent hypothesis
* Output S and G.

Code:-

```
import pandas as pd
data = [
 ['Big', 'Red', 'Circle', 'No']
 ['small', 'Red', 'Triangle', 'No']
 ['small', 'Red', 'Circle', 'Yes']
 ['Big', 'Blue', 'Circle', 'No']
]
df = pd.DataFrame(data, columns = ['size', 'Color',
                    'shape', 'class'])
S = [0, 0, 0]
G = [['?', '?', '?']]
for i, row in df.iterrows():
    h = row[:-1]; c = row[-1]
    if c == 'Yes':
        for j in range(3):
            if S[j] == 0: S[j] = h[j]
            elif S[j] != h: S[j] = '?'
        G = [g for g in G if all(g[k] == '?' or
    g[k] == h[k] for k in range(3))]
    else:
        newG = []
        for g in G:
            for j in range(3):
                g[j] == '?'
                    new = list(g); new[j] = S[j]
                    newG.append(new)
        G = newG
```

```
print ("S:", S)
print ("G:", G)
```

Output:-

S: [small, ?, Circle]

G: [[?, Red, ?], [?, Blue, ?]]

Result:- Succesfully find the output.

3. # Logistic Regression.

Aim:- Predict "Byy"

Algorithm:-

* Load dataset
* Split into train/test
* Train Logistic Regression model of traning data
* Predict on data test
* Compute accuracy
* Print predictions.

Code:-

```
from sklearn.linear_model import
Logistic Regression
from sklearn.model_selection import
train_test_split
import pandas as pd.
```

```
df = pd.DataFrame({
    Age : [20, 30, 40, 50, 60]
    Income : [20, 30, 40, 50, 60]
    Buy : [0, 0, 1, 1, 1]
})
x = df[['Age', 'Income']]
y = df['Buy']
xtr, xte, ytr, yte = train_test_split(x, y, test_size
                                      = 0.2)
model = logisticRegression()
model.fit(xtr, ytr)
print(Pred, model.predict(xte))
```

Output:-

Pred : [1]

Result:- successfully found the predict [1]

4. Naive Bayes Algorithm

Aim:- Predict "Pass" based on study & sptslup.

Algorithm:-

* Load dataset
* Split into train / test
* Train Naive Bayes model
* Predict test labels
* Calculate accuracy
* Print predictions.

Code:-

```
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
import pandas as pd
df = pd.DataFrame({
    'study': [2, 4, 6, 8, 10],
    'sleep': [9, 8, 7, 6, 5],
    'Pass': [0, 0, 1, 1, 1]
})
x = df[['study', 'sleep']]
y = df['Pass']
xtr, xte, ytr, yte = train_test_split(x, y, test_size=0.2)

model = GaussianNB()
model.fit(xtr, ytr)
print("Pred", model.predict(xte))
```

Output

Pred: [1]

Result:- successfully found the predict [1]