MODEL LAB EXAM

MLA0102 - Fundamentals of Machine Learning

A Aishaya
192425110
SET - 6

① **Aim:-** To predict house prices for mark using historical data.

Algorithm:-

* Load dataset with pandas
* check first 5 rows and basic statistics.
* check columns, data types, & handle nulls.
* Explore correlations with a heatmap.
* Split data into train/test sets.
* Train a linear regression model.
* predict house prices & evaluate performance.

Program:-

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
df = pd.read_csv('house_data.csv').fillna(0)
X = pd.get_dummies(df.drop('price', axis=1), drop_first=True);
y = df['price']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
model = LinearRegression().fit(X_train, y_train)
y_pred = model.predict(X_test)
print(pd.DataFrame({'Actual': y_test, 'predicted': y_pred}).head())
```

Output:-

	Actual	predicted
0	500000	510000.0
1	650000	640000.0
2	200000	220000.0
3	400000	410000.0
4	900000	880000.0

Result:- Model predicts house prices accurately. actual & predicted values are close.

② **Aim:-** To find the most specific (S) & most general (G) hypothesis consistent with training data.

Algorithm:-

① Initialize S = most specific, G

② If +ve → generalize S, remove G

If -ve → specialize G, remove S

③ Output S & G

Program:-

```
data = [['Circular', 'Large', 'Light', 'Smooth', 'Thick', '+'], ['oval',
    'Large', 'Light', 'Irregular', 'Thick', '+']]
S = ['0']*5; G = [['?']*5]
for x in data:
    x = x[:-1]
    if x[-1] == '+':
        G = [g for g in G if all(a=='?' or a==b for a,
    b in zip(g,x))]
```

```
G_new.append(g[:j] + ['?'] + g[j+1:])
G = [g for g in G_new if
all(c == '?' or c == a for c, a in
zip(s, g))]
print("S:", S)
print("G:", G)
```

Output:-

```
S: ['Circular', 'Large', '?', '?', 'Thick']
G: [['Circular', 'Large', '?', '?', 'Thick'],
['?', 'Large', 'Light', '?', 'Thick']]
```

Result:- S and G show all hypotheses consistent with the training data.

(2) Aim:- To predict a continuous target variable using Linear Regression

Algorithm:-

1) Load dataset
2) Split into features (X) & target (Y)
3) Split into train/test sets.
4) Train Linear Regression model.
5) Predict and evaluate.

Program:-

```
from sklearn.model_selection import
train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score import pandas as pd
df = pd.DataFrame({'X': [1,2,3,4,5], 'Y': [2,4,5,4,5]})
X = df[['X']]; Y = df['Y']
X_train, X_test, y_train, y_test = train_test_split(X, Y,
test_size = 0.2, random_state = 0)
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
print("Predicted:", y_pred)
print("R2 Score:", r2_score(y_test, y_pred))
```

Output:-

```
Predicted: [4.2]
R2 Score: 0.6
```

Result:- Linear Regression predicts target values; R^2 shows model performance.

(4) Aim:- To cluster data using the expectation-maximization (EM) algorithm.

Algorithm:-

1. Load or generate data.
2. Fit a gaussian mixture model (gmm).
3. Predict cluster labels.
4. Evaluate clusters.

program:-

```
from sklearn.mixture import GaussianMixture
import numpy as np
X = np.array([[1,2], [1,4], [1,0], [10,2],
[10,4], [10,0]])
gmm = GaussianMixture(n_components = 2,
random_state = 0).fit(X)
labels = gmm.predict(X)
print("cluster labels:", labels)
print("Means:", gmm.means_)
```

Output:-

```
cluster labels: [0 0 0 1 1 1]
means: [[1 2]
        [10. 2.]]
```

Result:-

EM algorithm clusters the data; means & labels show the learned Gaussian distributions.