In [ ]:
```
import numpy as np
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import LinearRegression
from sklearn.naive_bayes import MultinomialNB, BernoulliNB
```

In [ ]:
```
# Đọc dữ liệu từ tập csv
data = pd.read_csv('Admission_Predict.csv')

# Hiển thị thông tin cơ bản về dữ liệu
data.head()
```

Out[ ]:

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	1	337	118	4	4.5	4.5	9.65	1	0.92
1	2	324	107	4	4.0	4.5	8.87	1	0.76
2	3	316	104	3	3.0	3.5	8.00	1	0.72
3	4	322	110	3	3.5	2.5	8.67	1	0.80
4	5	314	103	2	2.0	3.0	8.21	0	0.65

In [ ]:
```
data = data.drop('Serial No.', axis=1)
```

In [ ]:
```
# Hàm sigmoid
def sigmoid(z):
    return 1 / (1 + np.exp(-z))

# Gradient descent
def logistic_regression(X, y, learning_rate, iterations = 10000, tol=1e-4):
    m, n = X.shape
    theta = np.zeros(n)
    for i in range(iterations):
        z = np.dot(X, theta)
        h = sigmoid(z)
        gradient = np.dot(X.T, (h - y)) / m
        if i % n == 0:
            if np.linalg.norm(theta - theta + learning_rate * gradient) < tol:
                return theta - learning_rate * gradient
        theta -= learning_rate * gradient

    return theta
```

In [ ]:
```
# Chia dữ liệu thành features (X) và target (y)
X = data.drop('Chance of Admit', axis=1)
y = np.where(data['Chance of Admit'] >= 0.75, 1, 0)
```

In [ ]:
```
# Thêm cột bias vào ma trận X
X = np.column_stack((np.ones(len(X)), X))

# Chia dữ liệu thành tập huấn luyện và tập kiểm tra
X_train, X_test = X[:350], X[350:]
y_train, y_test = y[:350], y[350:]
```

In [ ]:
```
eta = .05
iterations = 10 ** 8
```

In [ ]:
```
model = LogisticRegression(max_iter=iterations)
model.fit(X_train, y_train)

sk_y_pred = model.predict(X_test)
```

In [ ]:
```
# Đánh giá mô hình
accuracy = np.mean(sk_y_pred == y_test)
print("Độ chính xác:", accuracy)
```

Độ chính xác: 0.9

In [ ]:
```
# Thực hiện gradient descent để tìm các tham số theta
w = logistic_regression(X_train, y_train, eta, iterations)
```

```
C:\Users\Hoang Tu\AppData\Local\Temp\ipykernel_2416\2582513912.py:3: RuntimeWarning: overflow encountered in exp
  return 1 / (1 + np.exp(-z))
```

In [ ]:
```
# Dự đoán trên tập kiểm tra
z = np.dot(X_test, w)
h = sigmoid(z)
y_pred = np.where(h >= 0.75, 1, 0)
```

```
C:\Users\Hoang Tu\AppData\Local\Temp\ipykernel_2416\2582513912.py:3: RuntimeWarning: overflow encountered in exp
  return 1 / (1 + np.exp(-z))
```

In [ ]:
```
# Đánh giá mô hình
accuracy = np.mean(y_pred == y_test)
print('Độ chính xác:', accuracy)
```

Độ chính xác: 0.88

In [ ]:
```
df = pd.DataFrame({'Real label': y_test, 'My solution': y_pred, 'Sklearn': sk_y_pred, })

print(df)
```

```
     Real label  My solution  Sklearn
0             0            0        0
1             0            1        1
2             0            0        0
3             0            0        0
4             0            0        0
5             0            0        0
6             1            1        1
7             0            0        0
8             0            0        0
9             1            0        0
10            1            1        1
11            1            1        1
12            1            1        1
13            0            0        0
14            1            1        0
15            1            1        1
16            0            1        0
17            0            0        0
18            0            0        0
19            0            0        0
20            0            0        0
21            1            1        1
22            1            1        1
23            1            1        0
24            0            0        0
25            0            0        0
26            0            0        0
27            0            0        0
28            0            0        0
29            0            0        0
30            1            1        1
31            0            1        0
32            1            1        1
33            0            0        0
34            1            1        1
35            1            1        1
36            0            0        0
37            0            0        0
38            0            0        0
39            1            1        1
40            0            0        0
41            0            0        0
42            1            1        1
43            1            0        0
44            1            1        1
45            1            1        1
46            1            1        1
47            1            1        1
48            0            1        0
49            1            1        1
```

Linear Regressiong Model

In [ ]:
```
model = LinearRegression()
model.fit(X_train, y_train)
linear_predict = np.where(model.predict(X_test) >= 0.75, 1, 0)

accuracy = np.mean(linear_predict == y_test)
print("Độ chính xác:", accuracy)
```

Độ chính xác: 0.78

In [ ]:
```
df["Linear"] =  linear_predict
print(df)
```

```
     Real label  My solution  Sklearn  Linear
0             0            0        0       0
1             0            1        1       0
2             0            0        0       0
3             0            0        0       0
4             0            0        0       0
5             0            0        0       0
6             1            1        1       0
7             0            0        0       0
8             0            0        0       0
9             1            0        0       0
10            1            1        1       0
11            1            1        1       1
12            1            1        1       1
13            0            0        0       0
14            1            1        0       0
15            1            1        1       1
16            0            1        0       0
17            0            0        0       0
18            0            0        0       0
19            0            0        0       0
20            0            0        0       0
21            1            1        1       1
22            1            1        1       1
23            1            1        0       0
24            0            0        0       0
25            0            0        0       0
26            0            0        0       0
27            0            0        0       0
28            0            0        0       0
29            0            0        0       0
30            1            1        1       0
31            0            1        0       0
32            1            1        1       1
33            0            0        0       0
34            1            1        1       1
35            1            1        1       1
36            0            0        0       0
37            0            0        0       0
38            0            0        0       0
39            1            1        1       0
40            0            0        0       0
41            0            0        0       0
42            1            1        1       1
43            1            0        0       0
44            1            1        1       1
45            1            1        1       1
46            1            1        1       0
47            1            1        1       1
48            0            1        0       0
49            1            1        1       1
```

Navie Bayes

In [ ]:
```
model = MultinomialNB()
model.fit(X_train, y_train)
NB_predict = np.where(model.predict(X_test) >= 0.75, 1, 0)

accuracy = np.mean(NB_predict == y_test)
print("Độ chính xác:", accuracy)
```

Độ chính xác: 0.86

In [ ]:
```
df["MNB"] =  NB_predict
print(df)
```

```
     Real label  My solution  Sklearn  Linear  MNB
0             0            0        0       0    0
1             0            1        1       0    1
2             0            0        0       0    0
3             0            0        0       0    0
4             0            0        0       0    0
5             0            0        0       0    0
6             1            1        1       0    1
7             0            0        0       0    0
8             0            0        0       0    0
9             1            0        0       0    0
10            1            1        1       0    1
11            1            1        1       1    1
12            1            1        1       1    1
13            0            0        0       0    0
14            1            1        0       0    1
15            1            1        1       1    1
16            0            1        0       0    1
17            0            0        0       0    0
18            0            0        0       0    0
19            0            0        0       0    0
20            0            0        0       0    0
21            1            1        1       0    1
22            1            1        1       1    1
23            1            1        0       0    1
24            0            0        0       0    0
25            0            0        0       0    0
26            0            0        0       0    0
27            0            0        0       0    0
28            0            0        0       0    0
29            0            0        0       0    0
30            1            1        1       0    1
31            0            1        0       0    1
32            1            1        1       1    1
33            0            0        0       0    0
34            1            1        1       1    1
35            1            1        1       1    1
36            0            0        0       0    0
37            0            0        0       0    0
38            0            0        0       0    0
39            1            1        1       0    1
40            0            0        0       0    0
41            0            0        0       0    0
42            1            1        1       1    1
43            1            0        0       0    0
44            1            1        1       1    1
45            1            1        1       1    1
46            1            1        1       0    1
47            1            1        1       1    1
48            0            1        0       0    0
49            1            1        1       1    1
```

In [ ]:
```
model = BernoulliNB()
model.fit(X_train, y_train)
NB_predict = np.where(model.predict(X_test) >= 0.75, 1, 0)

accuracy = np.mean(NB_predict == y_test)
print("Độ chính xác:", accuracy)
```

Độ chính xác: 0.8

In [ ]:
```
df["BNB"] =  NB_predict
print(df)
```

```
     Real label  My solution  Sklearn  Linear  MNB  BNB
0             0            0        0       0    0    1
1             0            1        1       0    1    1
2             0            0        0       0    0    1
3             0            0        0       0    0    0
4             0            0        0       0    0    0
5             0            0        0       0    0    0
6             1            1        1       0    1    1
7             0            0        0       0    0    1
8             0            0        0       0    0    0
9             1            0        0       0    0    0
10            1            1        1       0    1    1
11            1            1        1       1    1    1
12            1            1        1       1    1    1
13            0            0        0       0    0    0
14            1            1        0       0    1    1
15            1            1        1       1    1    1
16            0            1        0       0    1    1
17            0            0        0       0    0    0
18            0            0        0       0    0    0
19            0            0        0       0    0    0
20            0            0        0       0    0    0
21            1            1        1       0    0    1
22            1            1        1       1    1    1
23            1            1        0       0    0    1
24            0            0        0       0    0    0
25            0            0        0       0    0    0
26            0            0        0       0    0    0
27            0            0        0       0    0    0
28            0            0        0       0    0    0
29            0            0        0       0    0    1
30            1            1        1       0    1    1
31            0            1        0       0    0    1
32            1            1        1       1    1    1
33            0            0        0       0    0    0
34            1            1        1       1    1    1
35            1            1        1       1    1    1
36            0            0        0       0    0    0
37            0            0        0       0    0    0
38            0            0        0       0    0    0
39            1            1        1       0    1    1
40            0            0        0       0    0    0
41            0            0        0       0    0    0
42            1            1        1       1    1    1
43            1            0        0       0    0    0
44            1            1        1       1    1    1
45            1            1        1       0    1    1
```

46	1	1	1	0	0	1
47	1	1	1	1	1	1
48	0	1	0	0	0	0
49	1	1	1	1	1	1