In [1]:

```
import numpy as np
import pandas as pd
```

In [2]:

```
df = pd.read_csv('Social_Network_Ads.csv')
df.head()
```

Out[2]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0

In [3]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   User ID          400 non-null    int64 
 1   Gender           400 non-null    object
 2   Age              400 non-null    int64 
 3   EstimatedSalary  400 non-null    int64 
 4   Purchased        400 non-null    int64 
dtypes: int64(4), object(1)
memory usage: 15.8+ KB
```

In [4]:

```
# performing encoding using dummies
data = df.drop(columns=['User ID'])

data = pd.get_dummies(data)
data
```

Out[4]:

	Age	EstimatedSalary	Purchased	Gender_Female	Gender_Male
0	19	19000	0	False	True
1	35	20000	0	False	True
2	26	43000	0	True	False
3	27	57000	0	True	False
4	19	76000	0	False	True
...	...	...	...	...	...
395	46	41000	1	True	False
396	51	23000	1	False	True
397	50	20000	1	True	False
398	36	33000	0	False	True
399	49	36000	1	True	False

400 rows × 5 columns

In [5]:

```
predictions = ['Age', 'EstimatedSalary','Gender_Female', 'Gender_Male']
x = data[predictions]
y = data['Purchased']
```

In [6]:

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(x)
scaled_data = scaler.transform(x)
scaled_data = pd.DataFrame(scaled_data, columns=x.columns)
scaled_data.head()
```

Out[6]:

	Age	EstimatedSalary	Gender_Female	Gender_Male
0	-1.781797	-1.490046	-1.020204	1.020204
1	-0.253587	-1.460681	-1.020204	1.020204
2	-1.113206	-0.785290	0.980196	-0.980196
3	-1.017692	-0.374182	0.980196	-0.980196
4	-1.781797	0.183751	-1.020204	1.020204

In [7]:

```
# train test split
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(scaled_data, y, test_size=0.2, random_state=1
```

In [8]:

```
# training the model
from sklearn import linear_model

model = linear_model.LogisticRegression()
model.fit(x_train, y_train)
```

Out[8]:

```
▾ LogisticRegression ⓘ ?
LogisticRegression()
```

(https://scikit-learn.org/1.4/modules/generated/sklearn.linear_model.LogisticRegression.html)

In [9]:

```
# evaluating the model
from sklearn.metrics import accuracy_score

y_pred = model.predict(x_test)

accuracy_score = round(accuracy_score(y_pred, y_test), 4)
print('Accuracy: ', accuracy_score)
```

```
Accuracy:  0.825
```

In [ ]: