Define Problem : Predict whether a student will get placed based on real-world factors like skills, projects, certifications, networking, and company interactions.

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

```
df = pd.read_csv("/content/placement_data_updated.csv")
df.head()
```

	random_id	cgpa	projects	certifications	soft_skills	linkedin_connections	hackathons	interview_attempts	favorite_subject	p
0	8270	7.81	0	4	1	145	3	9	History	
1	1860	6.91	1	4	5	713	0	7	Art	
2	6390	9.86	0	0	5	712	2	9	Physics	
3	6191	9.24	5	2	1	239	2	1	History	
4	6734	8.61	3	1	5	785	0	4	Physics	

```
df.columns
```

```
Index(['random_id', 'cgpa', 'projects', 'certifications', 'soft_skills',
       'linkedin_connections', 'hackathons', 'interview_attempts',
       'favorite_subject', 'placed'],
      dtype='object')
```

select dependent and independent var or X and y var

```
X = df[['cgpa', 'projects', 'certifications', 'soft_skills', 'hackathons', 'interview_attempts']]
y = df['placed']
```

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=42)
```

```
from sklearn.linear_model import LogisticRegression
```

```
model = LogisticRegression()
```

```
model.fit(X_train, y_train)
```

```
▾ LogisticRegression
LogisticRegression()
```

```
from sklearn.metrics import accuracy_score
```

```
predictions = model.predict(X_test)
```

```
acc = accuracy_score(y_test, predictions)
```

```
print(acc)
```

```
0.6208333333333333
```

```
import joblib
```

```
joblib.dump(model,'model.pkl')
```

```
['model.pkl']
```

```
# 'cgpa', 'projects', 'certifications', 'soft_skills', 'hackathons', 'interview_attempts'
new_student = np.array([[9, 2, 3, 0, 1, 2]])
```

```
prediction = model.predict(new_student)
print(prediction)
print("Prediction:", "Placed" if prediction[0] == 1 else "Not Placed")
```

```
[1]
Prediction: Placed
/usr/local/lib/python3.11/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but Log
  warnings.warn(
```