In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
```

In [2]:
```
df=pd.read_csv('Placement_Data_Full_Class.csv')
```

In [3]:
```
df
```

Out[3]:

	\	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	status	salary
0	1	M	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tech	No	55.0	Mkt&HR	58.80	Placed	270000.0
1	2	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	Placed	200000.0
2	3	M	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No	75.0	Mkt&Fin	57.80	Placed	250000.0
3	4	M	56.00	Central	52.00	Central	Science	52.00	Sci&Tech	No	66.0	Mkt&HR	59.43	Not Placed	NaN
4	5	M	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	Placed	425000.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
210	211	M	80.60	Others	82.00	Others	Commerce	77.60	Comm&Mgmt	No	91.0	Mkt&Fin	74.49	Placed	400000.0
211	212	M	58.00	Others	60.00	Others	Science	72.00	Sci&Tech	No	74.0	Mkt&Fin	53.62	Placed	275000.0
212	213	M	67.00	Others	67.00	Others	Commerce	73.00	Comm&Mgmt	Yes	59.0	Mkt&Fin	69.72	Placed	295000.0
213	214	F	74.00	Others	66.00	Others	Commerce	58.00	Comm&Mgmt	No	70.0	Mkt&HR	60.23	Placed	204000.0
214	215	M	62.00	Central	58.00	Others	Science	53.00	Comm&Mgmt	No	89.0	Mkt&HR	60.22	Not Placed	NaN

215 rows × 15 columns

In [4]:
```
df=df.drop(columns=["salary"])
```

In [5]:
```
df
```

Out[5]:

	\	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	status
0	1	M	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tech	No	55.0	Mkt&HR	58.80	Placed
1	2	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	Placed
2	3	M	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No	75.0	Mkt&Fin	57.80	Placed
3	4	M	56.00	Central	52.00	Central	Science	52.00	Sci&Tech	No	66.0	Mkt&HR	59.43	Not Placed
4	5	M	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	Placed
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
210	211	M	80.60	Others	82.00	Others	Commerce	77.60	Comm&Mgmt	No	91.0	Mkt&Fin	74.49	Placed
211	212	M	58.00	Others	60.00	Others	Science	72.00	Sci&Tech	No	74.0	Mkt&Fin	53.62	Placed
212	213	M	67.00	Others	67.00	Others	Commerce	73.00	Comm&Mgmt	Yes	59.0	Mkt&Fin	69.72	Placed
213	214	F	74.00	Others	66.00	Others	Commerce	58.00	Comm&Mgmt	No	70.0	Mkt&HR	60.23	Placed
214	215	M	62.00	Central	58.00	Others	Science	53.00	Comm&Mgmt	No	89.0	Mkt&HR	60.22	Not Placed

215 rows × 14 columns

In [6]:
```
from sklearn.preprocessing import LabelEncoder
label_encoders={}
for column in df.select_dtypes(include=['object']).columns:
    le=LabelEncoder()
    df[column]=le.fit_transform(df[column])
    label_encoders[column]=le
```

In [7]:
```
print(df.dtypes)
```

```
\                  int64
gender             int32
ssc_p            float64
ssc_b              int32
hsc_p            float64
hsc_b              int32
hsc_s              int32
degree_p         float64
degree_t           int32
workex             int32
etest_p          float64
specialisation     int32
mba_p            float64
status             int32
dtype: object
```

In [8]:
```
independent=df[["gender","ssc_p","ssc_b","hsc_p","hsc_b","hsc_s","degree_p","degree_t","workex","etest_p","specialisation","mba_p"]]
dependent=df[["status"]]
```

In [9]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(independent,dependent,test_size=0.2,random_state=42)
```

In [10]:
```
model=RandomForestClassifier(n_estimators=100,random_state=42)
model.fit(X_train,Y_train)
```

```
C:\Users\CSELAB\AppData\Local\Temp\ipykernel_11524\402067526.py:2: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please ch
ange the shape of y to (n_samples,), for example using ravel().
  model.fit(X_train,Y_train)
```

Out[10]:
```
RandomForestClassifier(random_state=42)
```

In [11]:
```
Y_pred=model.predict(X_test)
```

In [12]:
```
accuracy=accuracy_score(Y_test,Y_pred)
print(f"model Accuracy:{accuracy:2f}")
```

```
model Accuracy:0.790698
```

In [13]:
```
user_input=[]
```

In [14]:
```
for col in independent.columns:
    value=int(input(f"Enter value for {col}:"))
    user_input.append(value)
predicted_status=model.predict([user_input])
print("predicted status(0=Not Placed,1=Placed):",predicted_status[0])
```

```
Enter value for gender:1
Enter value for ssc_p:62
Enter value for ssc_b:0
Enter value for hsc_p:68
Enter value for hsc_b:0
Enter value for hsc_s:2
Enter value for degree_p:58
Enter value for degree_t:0
Enter value for workex:1
Enter value for etest_p:86
Enter value for specialisation:0
Enter value for mba_p:69
predicted status(0=Not Placed,1=Placed): 1
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but RandomForestClassifier was fitted with f
eature names
  warnings.warn(
```

In [ ]: