```
import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression

cam=pd.read_csv("/content/drive/MyDrive/Campus_Selection.csv")
cam.head()
```


```
{"summary":"{\n  \"name\": \"cam\",\n  \"rows\": 215,\n  \"fields\":
[\n    {\n      \"column\": \"sl_no\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 62,\n        \"min\": 1,\n
\"max\": 215,\n        \"num_unique_values\": 215,\n
\"samples\": [\n          201,\n          213,\n          139\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"gender\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
2,\n        \"samples\": [\n          \"F\",\n          \"M\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"ssc_p\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
10.827205398231452,\n        \"min\": 40.89,\n        \"max\": 89.4,\n
\"num_unique_values\": 103,\n        \"samples\": [\n          74.0,\n
73.96\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"ssc_b\",\n      \"properties\": {\n        \"dtype\": \"category\",\
n        \"num_unique_values\": 2,\n        \"samples\": [\n
\"Central\",\n          \"Others\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"hsc_p\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 10.89750915750298,\n
\"min\": 37.0,\n        \"max\": 97.7,\n        \"num_unique_values\":
97,\n        \"samples\": [\n          82.0,\n          73.2\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"hsc_b\",\n      \"properties\": {\n        \"dtype\": \"category\",\
n        \"num_unique_values\": 2,\n        \"samples\": [\n
\"Central\",\n          \"Others\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"hsc_s\",\n      \"properties\": {\
n        \"dtype\": \"category\",\n        \"num_unique_values\": 3,\n
\"samples\": [\n          \"Commerce\",\n          \"Science\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"degree_p\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
7.358743287339439,\n        \"min\": 50.0,\n        \"max\": 91.0,\n
\"num_unique_values\": 89,\n        \"samples\": [\n          71.72,\n
76.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"degree_t\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          \"Sci&Tech\",\n          \"Comm&Mgmt\"\n        ],\n
```

\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"workex\",\n   \"properties\": {\n   \"dtype\": \"category\",\n   \"num_unique_values\": 2,\n   \"samples\": [\n   \"Yes\",\n   \"No\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"etest_p\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 13.275956401653835,\n   \"min\": 50.0,\n   \"max\": 98.0,\n   \"num_unique_values\": 100,\n   \"samples\": [\n   93.4,\n   69.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"specialisation\",\n   \"properties\": {\n   \"dtype\": \"category\",\n   \"num_unique_values\": 2,\n   \"samples\": [\n   \"Mkt&Fin\",\n   \"Mkt&HR\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"mba_p\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 5.8333845806838,\n   \"min\": 51.21,\n   \"max\": 77.89,\n   \"num_unique_values\": 205,\n   \"samples\": [\n   64.66,\n   52.21\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"status\",\n   \"properties\": {\n   \"dtype\": \"category\",\n   \"num_unique_values\": 2,\n   \"samples\": [\n   \"Not Placed\",\n   \"Placed\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe","variable_name":"cam"}

```
cam.isnull().sum()
```

```
sl_no             0
gender            0
ssc_p             0
ssc_b             0
hsc_p             0
hsc_b             0
hsc_s             0
degree_p          0
degree_t          0
workex            0
etest_p           0
specialisation    0
mba_p             0
status            0
dtype: int64
```

```
ind=cam[['gender','ssc_p','hsc_p','degree_p']]
dep=cam['status']
```

```
Logr=LogisticRegression()
ind_encoded = pd.get_dummies(ind, columns=['gender'], drop_first=True)
Logr.fit(ind_encoded,dep)
```

```
LogisticRegression()
```

```
gender_input = input("enter gender (Male/Female):")
if gender_input.lower() == 'male':
    gender_encoded = 1  # Assuming gender_M is True for Male
elif gender_input.lower() == 'female':
    gender_encoded = 0  # Assuming gender_M is False for Female
else:
    print("Invalid gender input. Please enter 'Male' or 'Female'.")
    # You might want to handle this error more robustly, e.g., by
exiting or re-prompting.
    gender_encoded = None # Placeholder, will cause error if not
handled

if gender_encoded is not None:
    ssc_p=int(input("enter ssc_p:"))
    hsc_p=int(input("enter the hsc_p:"))
    degree_p=int(input("enter the degree_p:"))

    # The order of features in Logr.predict should match ind_encoded:
ssc_p, hsc_p, degree_p, gender_M
    pred=Logr.predict([[ssc_p, hsc_p, degree_p, gender_encoded]])
    print(pred)
```

```
Logr.score(ind_encoded,dep)
```

```
from sklearn.metrics import accuracy_score
pval=Logr.predict(ind_encoded)
accuracy_score(dep,pval)
```