29. Logistic Regression (Practical) (Binary Classification) (Multiple Inputs)

In [1]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:

```
dataset = pd.read_csv(r'Data/placement_2.csv')
dataset.head(3)
```

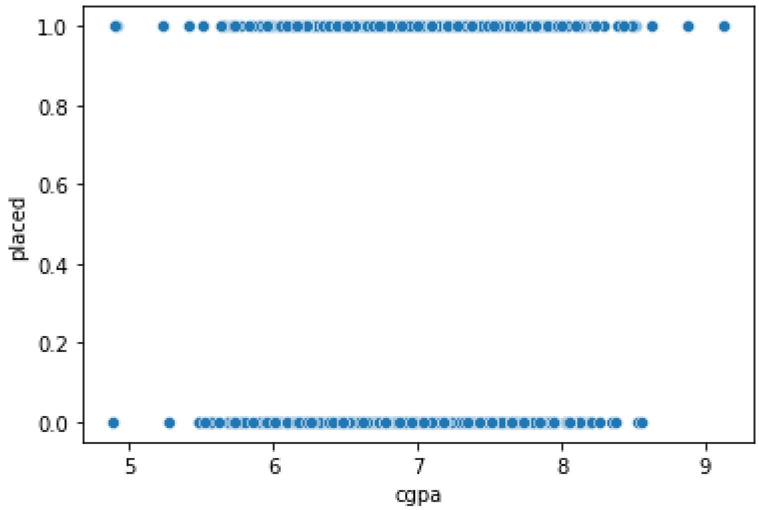
Out[2]:

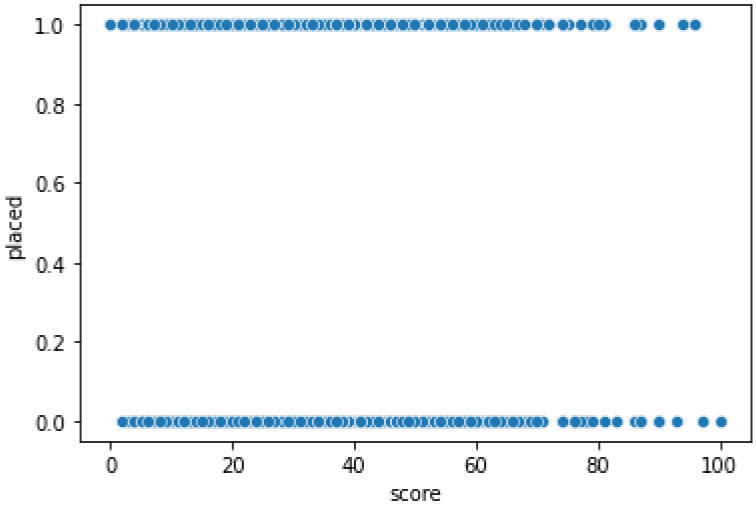
	cgpa	score	placed
0	7.19	26	1
1	7.46	38	1
2	7.54	40	1

Step 1: Check if the data follows logistic regression

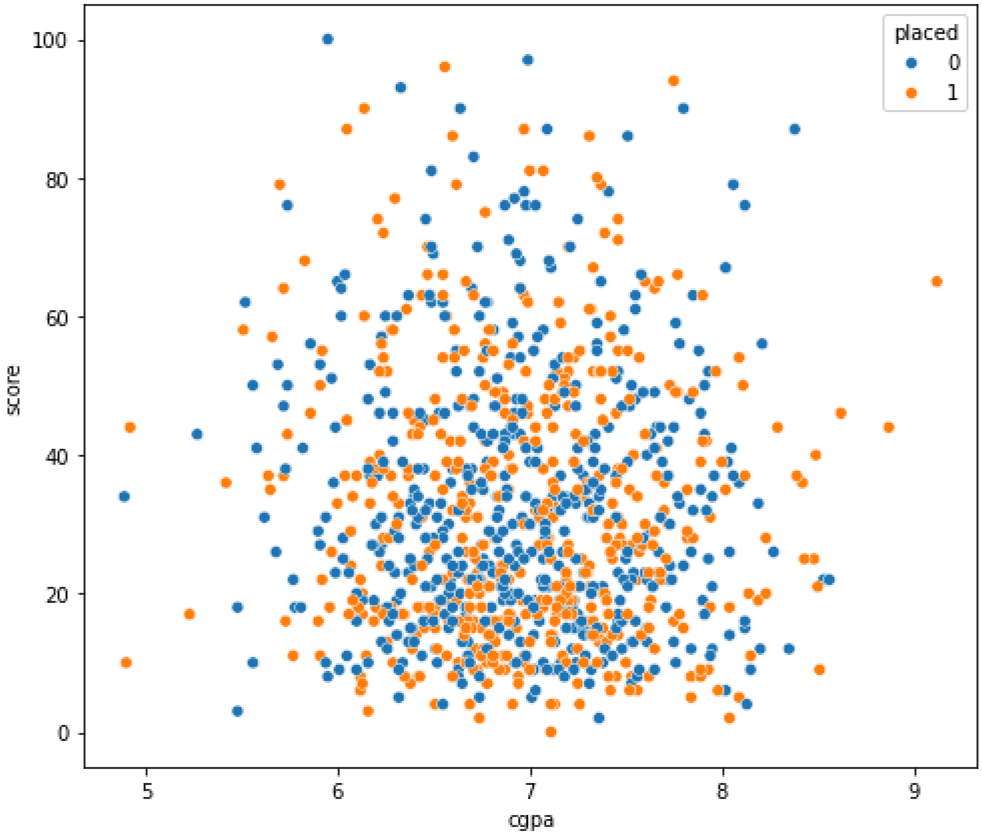
In [9]:

```
sns.scatterplot(x="cgpa", y="placed", data=dataset)
plt.show()
sns.scatterplot(x="score", y="placed", data=dataset)
plt.show()
```





```
In [24]: plt.figure(figsize=(8,7))
         sns.scatterplot(x="cgpa", y="score", data=dataset, hue='placed')
         plt.show()
```



Step 2: Split the data into independent/input (x: cgpa, score) and dependent variables/output (y: placed)

```
In [10]: x = dataset.iloc[:,:-1]
         x
```

Out[10]:

	cgpa	score
0	7.19	26
1	7.46	38
2	7.54	40
3	6.42	8
4	7.23	17
...	...	...
995	8.87	44
996	9.12	65
997	4.89	34
998	8.62	46
999	4.90	10

1000 rows × 2 columns

In [11]:

```
y = dataset['placed']
y
```

Out[11]:

```
0      1
1      1
2      1
3      1
4      0
      ..
995    1
996    1
997    0
998    1
999    1
Name: placed, Length: 1000, dtype: int64
```

Step 3: Split the data into training and test data

In [12]:

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

In [13]:

```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.20, random_stat
```

Step 4: Build Logistic Regression Model

In [14]:

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

In [16]:

```
lr = LogisticRegression()
lr.fit(x_train, y_train)
```

```
Out[16]:  ▾ LogisticRegression
          LogisticRegression()
```

Step 5: Check the accuracy of Model

```
In [18]: lr.score(x_test, y_test)*100
```

```
Out[18]: 51.5
```

Step 6: Do prediction based on the built model

```
In [20]: lr.predict([[6,53]])
```

```
C:\Users\rashi\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\bas
e.py:450: UserWarning: X does not have valid feature names, but LogisticRegression w
as fitted with feature names
  warnings.warn(
```

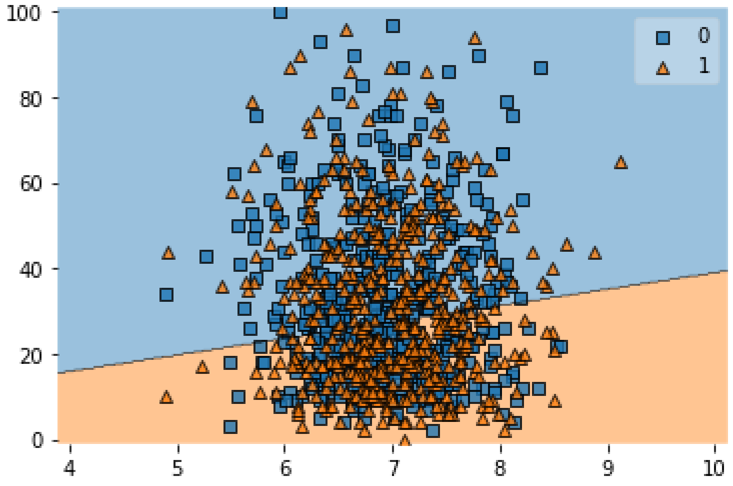
```
Out[20]: array([0], dtype=int64)
```

Step 7: Create Classifier boundary

```
In [26]: from mlxtend.plotting import plot_decision_regions
```

```
In [27]: plot_decision_regions(x.to_numpy(), y.to_numpy(), clf=lr)
         plt.show()
```

```
C:\Users\rashi\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\bas
e.py:450: UserWarning: X does not have valid feature names, but LogisticRegression w
as fitted with feature names
  warnings.warn(
```



```
In [ ]:
```