

```

4 # Print the probabilities for each one of the three data points
5 import pandas as pd
6 import numpy as np
7 from matplotlib import pyplot as plt
8 from sklearn import linear_model
9
10 df = pd.read_csv('Student-Pass-Fail.csv')
11 print(df)
12 X = np.array(df.iloc[:, 0:2])
13 y = np.array(df.iloc[:, 2])
14 print(X)
15 logReg = linear_model.LogisticRegression().fit(X, y)
16 print(f'Coef: {logReg.coef_}')
17 print(f'Coef shape: {logReg.coef_.shape}')
18 print(f'Intercept: {logReg.intercept_}')
19 log_odds = logReg.coef_
20 odds = np.exp(log_odds)
21 print(f'Odds: {odds}\n')
22 pred = np.array([[7, 28],
23                 [10, 34],
24                 [2, 39]])
25 Ypred = logReg.predict(pred)
26 print(f'Predicted: {Ypred}')
27 Ymodel = logReg.intercept_ + logReg.coef_[0, 0]*pred[0, 0] + logReg.coef_[0,
28 1]*pred[0, 1]
29 Ymodel2 = logReg.intercept_ + logReg.coef_[0, 0]*pred[1, 0] + logReg.coef_[0,
30 1]*pred[1, 1]
31 Ymodel3 = logReg.intercept_ + logReg.coef_[0, 0]*pred[2, 0] + logReg.coef_[0,
32 1]*pred[2, 1]
33 probability = np.exp(Ymodel) / (np.exp(Ymodel) + 1)
34 print(f'Probabilities: {probability}')
35 probability2 = np.exp(Ymodel2) / (np.exp(Ymodel2) + 1)
36 print(f'Probabilities: {probability2}')
37 probability3 = np.exp(Ymodel3) / (np.exp(Ymodel3) + 1)
38 print(f'Probabilities: {probability3}')

```

```
● raulrodriguez@Rauls-Air WorkSpaceVSPython % /usr/local
Coef: [[ 2.52529854 -0.84899742]]
Coef shape: (1, 2)
Intercept: [7.21234405]
Odds: [[12.49462487  0.42784367]]

Predicted: [1 1 0]
Probabilities: [0.75352584]
Probabilities: [0.97338792]
Probabilities: [8.8276849e-10]
○ raulrodriguez@Rauls-Air WorkSpaceVSPython %
```

```
6 import pandas as pd
7 import numpy as np
8 from matplotlib import pyplot as plt
9 from sklearn import linear_model
10 from sklearn.metrics import classification_report, confusion_matrix
11
12 df = pd.read_csv('Student-Pass-Fail.csv')
13 #print(df)
14 X = np.array(df.iloc[:, 0:2])
15 y = np.array(df.iloc[:, 2])
16 #print(X)
17 logReg = linear_model.LogisticRegression().fit(X, y)
18 print(f'Coef: {logReg.coef_}')
19 print(f'Coef shape: {logReg.coef_.shape}')
20 print(f'Intercept: {logReg.intercept_}')
21 log_odds = logReg.coef_
22 odds = np.exp(log_odds)
23 print(f'Odds: {odds}\n')
24
25 Ypredicted = logReg.predict(X)
26 confusionMatrix = confusion_matrix(y, Ypredicted)
27 print(confusionMatrix)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

- driguez/Documents/WorkspaceVSPython/Lab7_2ML.py
Coef: $\begin{bmatrix} 2.52529854 & -0.84899742 \end{bmatrix}$
Coef shape: (1, 2)
Intercept: [7.21234405]
Odds: $\begin{bmatrix} 12.49462487 & 0.42784367 \end{bmatrix}$

 $\begin{bmatrix} 478 & 23 \\ 1 & 498 \end{bmatrix}$
- raulrodriguez@Rauls-Air WorkspaceVSPython %

```
6 import pandas as pd
7 import numpy as np
8 from matplotlib import pyplot as plt
9 from sklearn import linear_model
10
11 df = pd.read_csv('Bank-data.csv')
12 #print(df)
13 df=df.drop(df.columns[0],axis=1)
14 #print(df)
15 X = np.array(df.iloc[:, 0:6])
16 y = np.array(df.iloc[:, 6])
17 print(X)
18 logReg = linear_model.LogisticRegression().fit(X, y)
19 print(f'Coef: {logReg.coef_}')
20 print(f'Coef shape: {logReg.coef_.shape}')
21 print(f'Intercept: {logReg.intercept_}')
22 log_odds = logReg.coef_
23 odds = np.exp(log_odds)
24 print(f'Odds: {odds}\n')
25 pred = np.array([[1.335, 0, 1, 0, 0, 109],
26                 [1.25, 0, 0, 1, 0, 279]])
27 Ypred = logReg.predict(pred)
28 print(f'Predicted: {Ypred}')
29 Ymodel = logReg.intercept_ + logReg.coef_[0, 0]*pred[0, 0] + logReg.coef_[0,
30 1]*pred[0, 1]
31 Ymodel2 = logReg.intercept_ + logReg.coef_[0, 0]*pred[1, 0] + logReg.coef_[0,
32 1]*pred[1, 1]
33 probability = np.exp(Ymodel) / (np.exp(Ymodel) + 1)
34 print(f'Probabilities: {probability}')
35 probability2 = np.exp(Ymodel2) / (np.exp(Ymodel2) + 1)
36 print(f'Probabilities: {probability2}')
```

```
/usr/local/bin/python3 /Users/raulrodriguez/Documents/WorkspaceVSPython/Lab7_3ML.py
● driguez/Documents/WorkspaceVSPython/Lab7_3ML.py
Coef: [[-0.77233164  1.23548459 -1.66070557  0.22072494  1.03857191  0.00679389]]
Coef shape: (1, 6)
Intercept: [-0.09072394]
Odds: [[0.46193475  3.44004513  0.19000487  1.24698038  2.82517952  1.00681702]]

Predicted: ['no' 'yes']
Probabilities: [0.24568002]
Probabilities: [0.25804809]
○ raulrodriguez@Rauls-Air WorkspaceVSPython %
```