logisticregression

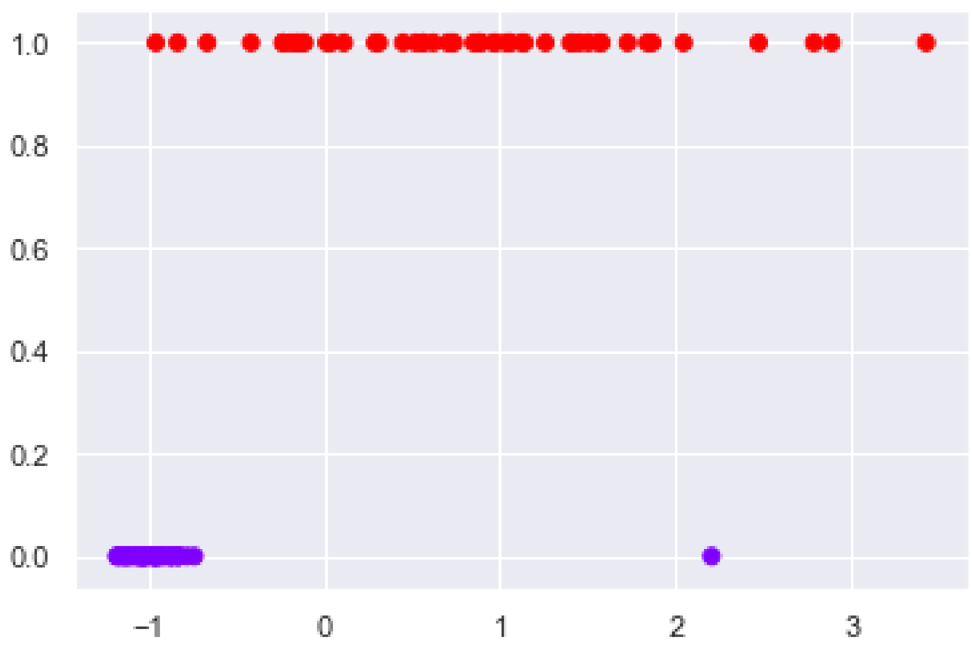
COSC3337

[1]:
```
from sklearn.datasets import make_classification
from matplotlib import pyplot as plt
from sklearn.linear_model import LogisticRegression
import seaborn as sns
sns.set()
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
import pandas as pd
```

[2]:
```
x, y = make_classification(
    n_samples=100,
    n_features=1,
    n_classes=2,
    n_clusters_per_class=1,
    flip_y=0.03,
    n_informative=1,
    n_redundant=0,
    n_repeated=0
)
```

[3]:
```
plt.scatter(x, y, c=y, cmap='rainbow')
```

[3]:
```
<matplotlib.collections.PathCollection at 0x22de1e47c08>
```



[4]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=1)
```

[5]:
```
#lbfgs - Stands for Limited-memory Broyden-Fletcher-Goldfarb-Shanno. It␣
 ↪approximates
#the second derivative matrix updates with gradient evaluations.
#It stores only the last few updates, so it saves memory
lr = LogisticRegression()
#fit is to find the new coefiicent of the logistic regression
lr.fit(x_train, y_train)
```

[5]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

[6]:
```
print(lr.coef_)
print(lr.intercept_)
```

```
[[3.14697545]]
[1.21465271]
```

[11]:
```
y_pred = lr.predict(x_test)
```

```
[0 0 1 1 0 0 0 1 1 1 0 1 1 0 1 0 0 1 1 0 0 0 0 1 0]
```

```
[13]: print(y_pred)
      print(y_test)
      confusion_matrix(y_test, y_pred)
```

```
[0 0 1 1 0 0 0 1 1 1 0 1 1 0 1 0 0 1 1 0 0 0 0 1 0]
[0 0 1 1 0 0 1 1 1 1 0 1 1 0 1 0 0 1 0 0 0 0 0 1 0]
```

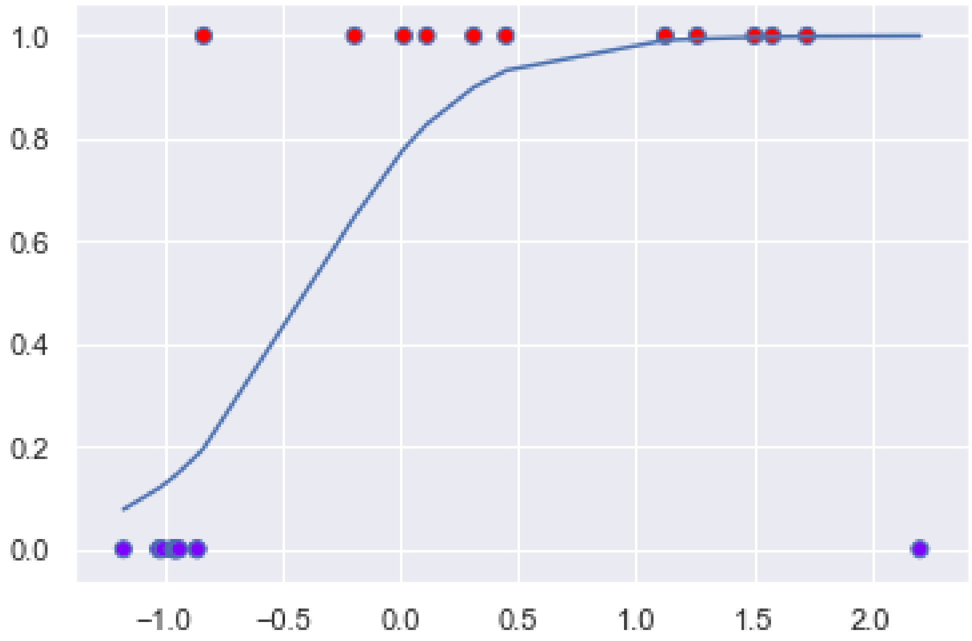
```
[13]: array([[13,  1],
             [ 1, 10]], dtype=int64)
```

```
[9]: lr.predict_proba(x_test)
```

```
[9]: array([[8.53553221e-01, 1.46446779e-01],
            [8.75151521e-01, 1.24848479e-01],
            [1.29905685e-03, 9.98700943e-01],
            [3.52222160e-01, 6.47777840e-01],
            [8.52160291e-01, 1.47839709e-01],
            [8.15862061e-01, 1.84137939e-01],
            [8.03100393e-01, 1.96899607e-01],
            [9.99889242e-02, 9.00011076e-01],
            [8.55322015e-03, 9.91446780e-01],
            [5.60192326e-03, 9.94398077e-01],
            [8.80394863e-01, 1.19605137e-01],
            [2.20114864e-01, 7.79885136e-01],
            [2.05734213e-03, 9.97942658e-01],
            [8.80854908e-01, 1.19145092e-01],
            [2.60995116e-03, 9.97390049e-01],
            [8.56549956e-01, 1.43450044e-01],
            [8.17846968e-01, 1.82153032e-01],
            [6.73171776e-02, 9.32682822e-01],
            [2.88240923e-04, 9.99711759e-01],
            [8.50008009e-01, 1.49991991e-01],
            [8.56992914e-01, 1.43007086e-01],
            [9.22559050e-01, 7.74409499e-02],
            [8.59507503e-01, 1.40492497e-01],
            [1.71948649e-01, 8.28051351e-01],
            [8.57947379e-01, 1.42052621e-01]])
```

```
[10]: df = pd.DataFrame({'x': x_test[:,0], 'y': y_test})
      df = df.sort_values(by='x')
      from scipy.special import expit
      sigmoid_function = expit(df['x'] * lr.coef_[0][0] + lr.intercept_[0]).ravel()
      plt.plot(df['x'], sigmoid_function)
      plt.scatter(df['x'], df['y'], c=df['y'], cmap='rainbow', edgecolors='b')
```

```
[10]: <matplotlib.collections.PathCollection at 0x22de1e8be88>
```


1.0
0.8
0.6
0.4
0.2
0.0
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

[ ]: