Output

April 11, 2022

[29]:
```
import sys
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import numpy as np
```

[30]:
```
sys.path.append("../")
from dataGen.o_data import OData
from models.logit import Logit
from models.probit import Probit
from models.lpm import LPM
```

[31]:
```
num_obs = 100
num_features = 2
y, X, *_ = OData.make_data(num_obs, num_features, "linear_simple",␣
 ↪error_dist="normal")

# X = StandardScaler().fit_transform(X)
X = np.concatenate((np.ones((X.shape[0], 1)), X), axis=1)
```

```
Generating noise with normal distribution
```

[32]:
```
# LPM
coef_est = LPM.estimate(X, y)
print("LPM Coefficients: ", coef_est)
print(f"LPM R^2: {LPM.rsq(y, LPM.predict_y(X, y))}")
```

```
LPM Coefficients:  [ 0.48424492 -0.03792858 -0.03649747]
LPM R^2: -0.3605442176870748
```

[33]:
```
# Logit
est = Logit().logit_model(X, y)
print("Logit Coefficients: ", est)
print(f"McFadden's R^2 for Logit: {Logit().rsq(X, y)}")
```

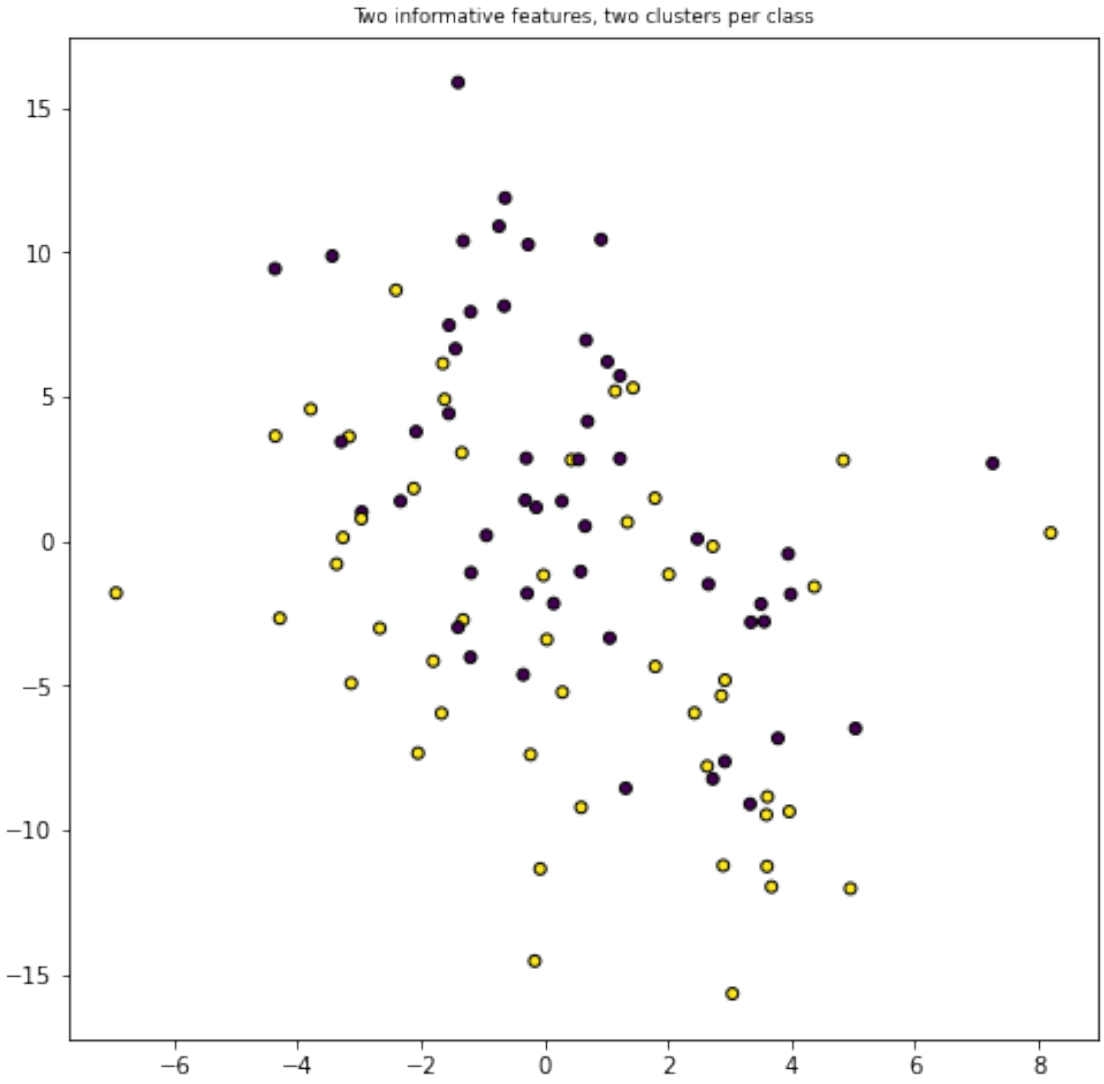
```
Logit Coefficients:        fun: 59.314976868709614
 hess_inv: <3x3 LbfgsInvHessProduct with dtype=float64>
      jac: array([-0.0006807 , -0.18054465,  0.33622101])
  message: 'CONVERGENCE: REL_REDUCTION_OF_F_<=_FACTR*EPSMCH'
```

```
     nfev: 44
      nit: 9
     njev: 11
   status: 0
  success: True
        x: array([-0.08114816, -0.18236779, -0.17404283])
McFadden's R^2 for Logit: 0.5383385473000828
```

[34]:
```
# Probit
est = Probit().probit_model(X, y)
print("Probit Coefficients: ", coef_est)
print(f"McFadden's R^2 for Probit: {Probit().rsq(X, y)}")
```

```
Probit Coefficients:  [ 0.48424492 -0.03792858 -0.03649747]
McFadden's R^2 for Probit: 0.5014932568002144
```

[35]:
```
plt.figure(figsize=(8, 8))
plt.title("Two informative features, two clusters per class", fontsize="small")
plt.scatter(X[:, 1], X[:, 2], marker="o", c=y, s=25, edgecolor="k")
plt.show()
```


Two informative features, two clusters per class
15
10
5
0
−5
−10
−15
−6
−4
−2
0
2
4
6
8

[ ]: