

LogisticRegression

August 9, 2023

```
[ ]: import torch
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
import numpy as np
```

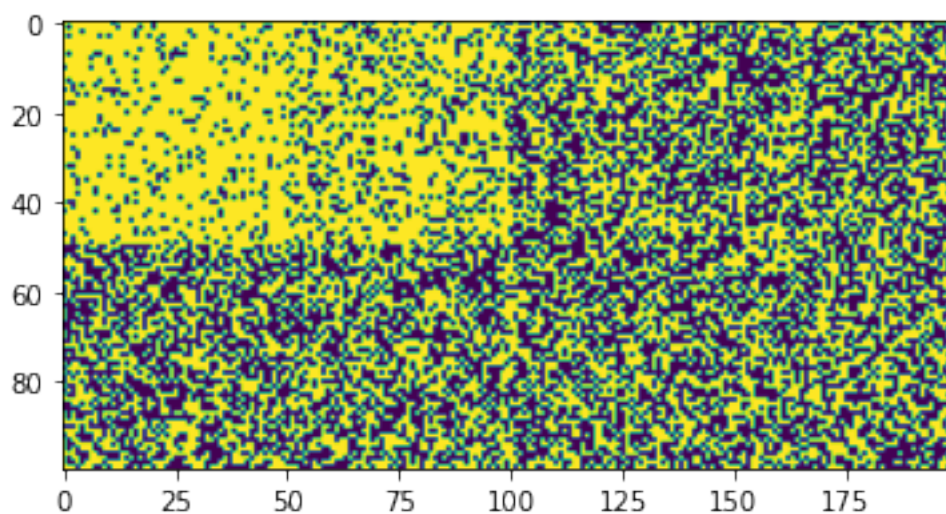
```
[ ]: x = torch.randint(0, 2, (100, 200), dtype=bool)
x1 = torch.randint(0, 2, ( 50, 100), dtype=bool)
x2 = torch.randint(0, 2, ( 50,  50), dtype=bool)

x[:50, :100] |= x1
x[:50, : 50] |= x2

plt.imshow(x)
plt.show()

y = torch.concat([torch.ones(50), torch.zeros(50)], dim=0)
y = y.type(torch.bool)

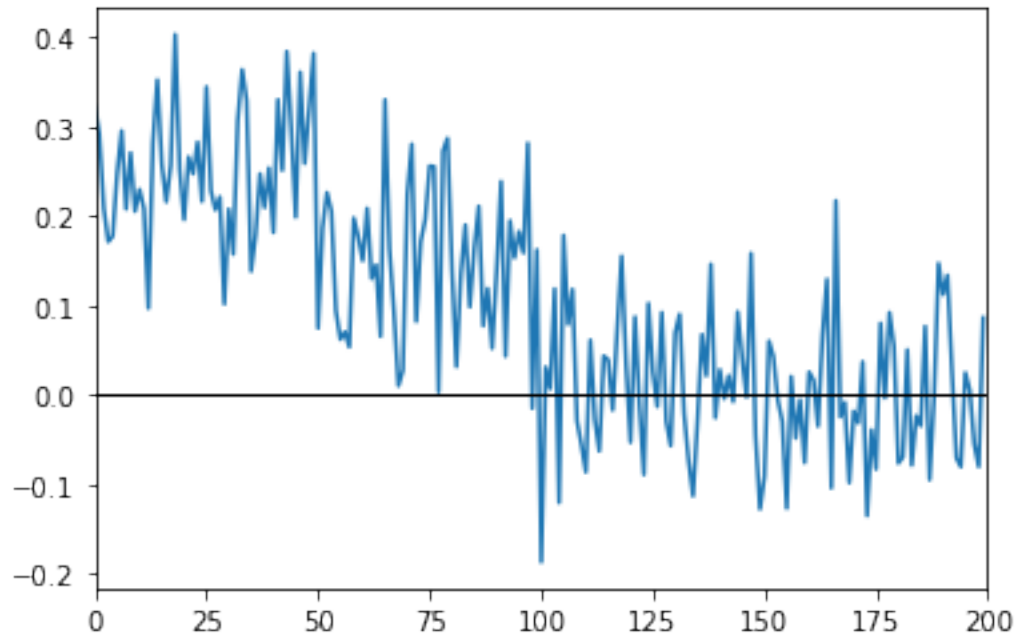
is_train = torch.randperm(100) < 80
```



```
[ ]: reg = LogisticRegression()

reg.fit(x[is_train], y[is_train])
plt.plot(reg.coef_[0])
plt.xlim(0, 200)
plt.axhline(0, c='black', lw=1)
plt.show()

confusion_matrix(reg.predict(x[~is_train]), y[~is_train])
```



```
[ ]: array([[ 9,  0],
           [ 0, 11]])
```

```
[ ]: OR = np.exp(reg.coef_[0])
OR = torch.tensor(OR, dtype=torch.float32)

patient_wise_sum = (x.type(torch.FloatTensor) * OR).sum(dim=1)
patient_wise_sum_positive = patient_wise_sum[is_train][y[is_train]]
patient_wise_sum_negative = patient_wise_sum[is_train][~y[is_train]]

plt.figure(figsize=(6, 6))
plt.boxplot(x=[patient_wise_sum_positive, patient_wise_sum_negative])
plt.xticks([1, 2], ['positivie', 'negative'])
plt.show()
```

