

Aufgabe12

November 15, 2018

Aufgabe 12

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from numpy.linalg import inv

In [2]: P0 = pd.read_hdf('zwei_populationen.h5', key='P_0_10000')
P1 = pd.read_hdf('zwei_populationen.h5', key='P_1')

In [3]: P0_x = P0['x']
P0_y = P0['y']
P1_x = P1['x']
P1_y = P1['y']
```

Teil a)

```
In [4]: mu_P0_x = np.mean(P0_x)
mu_P0_y = np.mean(P0_y)
mu_P1_x = np.mean(P1_x)
mu_P1_y = np.mean(P1_y)
print("mu_P0")
print(mu_P0_x, mu_P0_y)
print("mu_P1")
print(mu_P1_x, mu_P1_y)

mu_P0
-0.02743075223963595  2.9799446468232453
mu_P1
5.986448205069931  3.085282896934817
```

Teil b)

```
In [5]: V_P0 = np.cov(P0_x - mu_P0_x, P0_y - mu_P0_y)
V_P1 = np.cov(P1_x - mu_P1_x, P1_y - mu_P1_y)
V_P0_P1 = V_P0 + V_P1
mat_mu = np.mat(((mu_P0_x - mu_P1_x), (mu_P0_y - mu_P1_y))).T
```

```

V_B = mat_mu*mat_mu.T
print('V_P0', V_P0)
print('V_P1', V_P1)
print('V_P0_P1', V_P0_P1)
print('V_B', V_B)

V_P0 [[ 12.20892862  8.15840984]
 [ 8.15840984  6.72286327]]
V_P1 [[ 12.35218537  7.41075614]
 [ 7.41075614  5.47731503]]
V_P0_P1 [[ 24.56111399 15.56916598]
 [15.56916598 12.20017829]]
V_B [[ 3.61667401e+01  6.33491486e-01]
 [ 6.33491486e-01  1.10961469e-02]]

```

Teil c

```

In [6]: lambda1 = inv(V_P0_P1)*mat_mu
print(lambda1)
# ay= bx <=> y = b/a x
a = np.round(-float(lambda1[0]/lambda1[1]))
print('Geradengleichung: y =', np.round(float(-lambda1[0]/lambda1[1]), 4), 'x')

#S_W_S_B = inv(V_P0_P1)*V_B
#print(S_W_S_B)
#Det = (S_W_S_B[0,0])
#print(Det)

[[-1.2529094 ]
 [ 1.59025678]]
Geradengleichung: y = 0.7879 x

```

```

In [7]: #mat_mu = np.mat((-5.4), (-4)).T
#V_B = mat_mu*mat_mu.T
#V_P0_P1 = np.mat(((13.2, -2.2), (-2.2, 26.4)))
#lambda1 = inv(V_P0_P1)*mat_mu
#print(lambda1)
#a = [1,2,3]
#b= [5,0,4]
#print(min(min(a),min(b)))

```

```

In [8]: def lin(x,a):
        return x*a

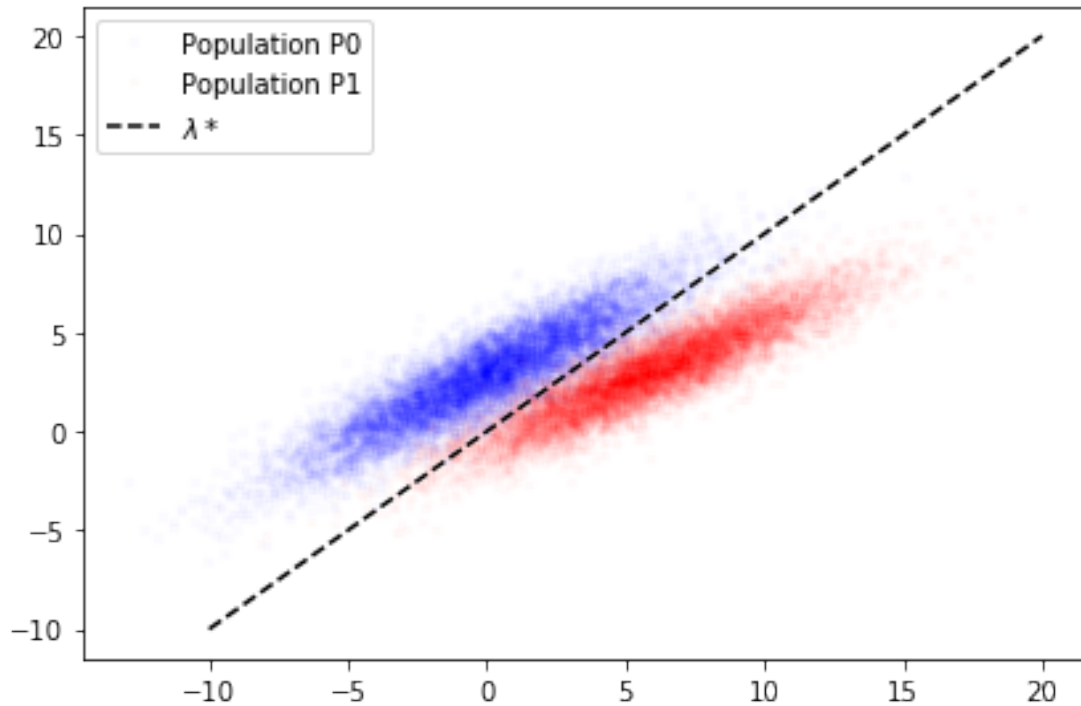
```

```

In [9]: x = np.linspace(-10,20)
plt.plot(P0_x,P0_y,'b.',alpha=0.01,label='Population P0')
plt.plot(P1_x,P1_y,'r.',alpha=0.01,label='Population P1')

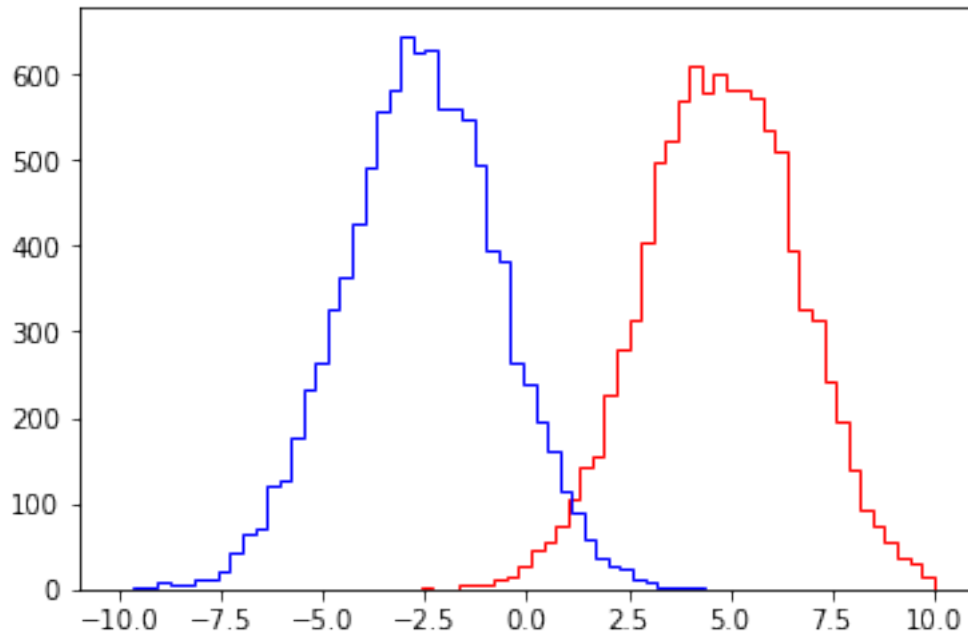
```

```
plt.plot(x, lin(x,a), 'k--',label=r'$\lambda *$')
plt.legend(loc='best')
plt.tight_layout()
plt.show()
```



Teil d)

```
In [10]: datahist_P0 = lambda1.T * np.mat((P0_x),(P0_y))
datahist_P1 = lambda1.T * np.mat((P1_x),(P1_y))
#print(datahist_P0)
plt.hist(datahist_P0.T,bins=50,range=(-5,10),color='r',histtype='step',label='Signal')
plt.hist(datahist_P1.T,bins=50,range=(-10,5),color='b',histtype='step', label='Untergru')
plt.show()
```



Teil e)-g)

In [11]: Ich habe es nicht hinbekommen λ_{cut} als Fkt. von λ bzw. die jeweiligen
 Ich habe den Rest so geschrieben, dass wenn man diesen Fehler korrigiert direkt weiter

```
In [12]: # Reinheit
reinh = []
#Effizienz
effiz = []
value = []
# Signal zu Hintergrund
S_B = []
# Signifikanz
sqrt_S_B = []
tp = 0
eff = 0
fp = 0
sq_s_b = 0
fn = 0
i = 0
# Hier muss ein Fehler sein
for x_value in range(-10,10):
    while i < 10000:
        if x_value < datahist_P0.T[i]:
            tp += 1
        if x_value < datahist_P1.T[i]:
```

```

        fp += 1
    else:
        fn += 1
    i += 1
    rein = tp / (tp + fp)
    eff = tp / (tp + fn)
    s_b = tp / fp
    sq_s_b = tp / (np.sqrt(tp + fp))
    Reinh.append(rein)
    Effiz.append(eff)
    S_B.append(s_b)
    sqrt_S_B.append(sq_s_b)
    value.append(x_value)

[0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
[-10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

```

```

In [13]: #x_value = np.linspace(4, 6)
#x_value = np.linspace(min(min(datahist_P0),min(datahist_P1)),max(max(datahist_P0),max(
plt.plot(value, Reinh, 'b-',label='Reinheit')
plt.plot(value, Effiz, 'g-',label='Effizienz')
plt.plot(value, S_B, 'r-',label='Signal zu Hintergrund')
plt.plot(value, sqrt_S_B, 'g-',label='Signifikanz')
#plt.legend(loc='best')
#plt.tight_layout()
plt.show()

```

