

# Learning and Decision Making

## Homework 4

(a)

In [112]:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

x = [1, 2, 2, 3, 5]
y = [3, 2, 1, 1, 4]
classAttr = [0, 1, 0, 0, 1]

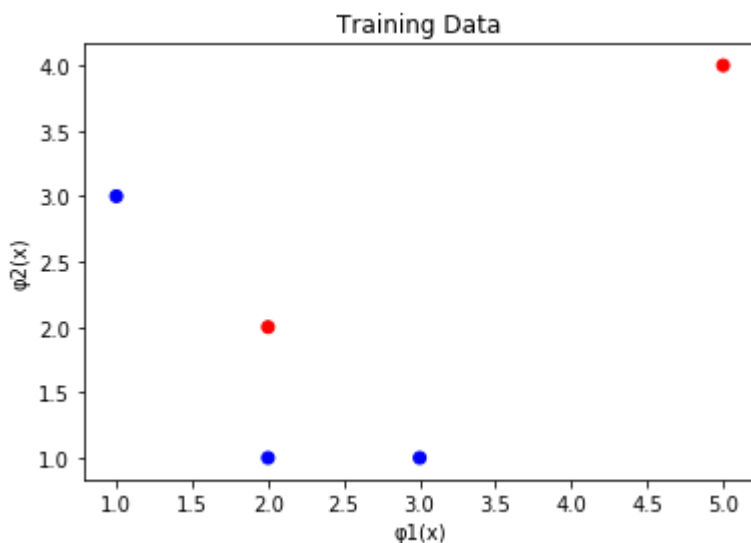
df = pd.DataFrame(dict(x=x, y=y, classattr=classAttr))

fig, ax = plt.subplots()

colors = {0: 'blue', 1: 'red'}
ax.scatter(df['x'], df['y'], c=df['classattr'].apply(lambda x: colors[x]))

plt.xlabel(" $\phi_1(x)$ ")
plt.ylabel(" $\phi_2(x)$ ")
plt.title("Training Data")
plt.show()

print("Os pontos não são linearmente separáveis, pois não é possível traçar uma recta que d
    " com a sua classe corretamente.")
```



Os pontos não são linearmente separáveis, pois não é possível traçar uma recta que divida os pontos de acordo com a sua classe corretamente.

(b)

In [113]:

```
from math import exp

def calcPi(point,weights):
    pi = weights[0]

    for i in range(len(point)-1):
        pi += weights[i + 1] * point[i]

    return 1.0/(1.0 + exp(-pi))

def calcWeights(data):
    alpha = 1
    weights = [0.0 for i in range(len(data[0]))]

    for point in data:
        pi = calcPi(point,weights)

        weights[0] += alpha * (point[2] - pi) * pi * (1 - pi)

        for i in range(0, len(point)-1):
            weights[i + 1] += alpha * (point[2] - pi) * pi * (1 - pi) * point[i]

    return weights

data = zip(x,y,classAttr)
data = list(data)
updatedWeights = calcWeights(data)
print(updatedWeights)
```

```
[-0.15566362061593841, -0.14056196132121454, -0.14791074869132217]
```

(c)

In [114]:

```

fig, ax = plt.subplots()

colors = {0: 'blue', 1: 'red'}
ax.scatter(df['x'], df['y'], c=df['classattr'].apply(lambda x: colors[x]))

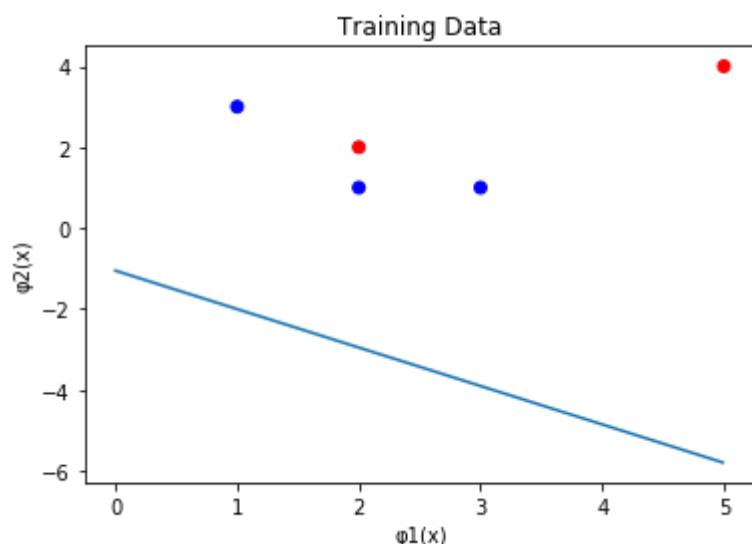
x1 = np.arange(0.0, 5.0, 0.01)
y1 = -(updatedWeights[0] + updatedWeights[1]*x1)/updatedWeights[2]

plt.plot(x1,y1)

plt.xlabel(" $\phi_1(x)$ ")
plt.ylabel(" $\phi_2(x)$ ")
plt.title("Training Data")
plt.show()

print("A recta não classifica propriamente os pontos. Isto deve-se provavelmente ao facto de
      " uma iteração do gradient descent. Testámos para 10000 iterações e a recta já conseg
      " os pontos.")

```



A recta não classifica propriamente os pontos. Isto deve-se provavelmente ao facto de só termos corrido uma iteração do gradient descent. Testámos para 10000 iterações e a recta já conseguia dividir melhor os pontos.

In [ ]: