

#for #8-10

```
import numpy as np
import random
```

```
def createData(n):
    matx = np.zeros(shape=(n,3))
    vecty = np.zeros(shape=(n,1))
    for i in range(0, n):
        x = np.matrix([1, random.uniform(-1, 1), random.uniform(-1, 1)])
        matx[i] = x
        vecty[i] = np.sign(np.square(x[0,1]) + np.square(x[0,2]) - 0.6)
    randomIndices = random.sample(range(0, n), int(n/10))
    for index in randomIndices:
        vecty[index] *= -1

    return (matx, vecty)
```

#8

```
def linRegIn(dataset, n):
    matx, vecty = dataset
    w = np.linalg.pinv(matx).dot(vecty)
    errorCount = 0
    for i in range(0, n):
        if np.sign(w.transpose().dot(matx[i,:])) != np.sign(vecty[i]):
            errorCount += 1
    return (errorCount / n)
```

#9

```
def transformLinReg(dataset, n):
    #transform
    matx, vecty = dataset
    matz = np.zeros(shape=(n,6))
    for i in range(0, n):
        x = matx[i,:]
        matz[i, 0:3] = x
        matz[i, 3] = x[1] * x[2]
        matz[i, 4:6] = np.matrix([np.square(x[1]), np.square(x[2])])

    #find w
    w = np.linalg.pinv(matz).dot(vecty)
    return w
```

#10

```
def linRegOut():
    w = transformLinReg(createData(1000), 1000)
    dataset = createData(1000)

    #transform new dataset
    matx, vecty = dataset
    matz = np.zeros(shape=(1000,6))
    for i in range(0, 1000):
        x = matx[i,:]
        matz[i, 0:3] = x
        matz[i, 3] = x[1] * x[2]
        matz[i, 4:6] = np.matrix([np.square(x[1]), np.square(x[2])])

    errorCount = 0
    for i in range(0, 1000):
        if np.sign(w.transpose().dot(matz[i,:])) != np.sign(vecty[i]):
```

```
        errorCount += 1
    return (errorCount / 1000)
```

```
#8: Calculate E_in w/o transformation
totalErrorFreq = 0
for i in range(0, 1000):
    totalErrorFreq += linRegIn(createData(1000), 1000)
print(totalErrorFreq / 1000)
```

```
#9: transform linear regression
totalw = np.zeros(shape=(6,1))
for i in range(0, 1000):
    totalw += transformLinReg(createData(1000), 1000)
print (totalw/1000)
```

```
#10: Calculate E_out w/ transformation
totalErrorFreq = 0
for i in range(0, 1000):
    totalErrorFreq += linRegOut()
print(totalErrorFreq / 1000)
```