

```
In [ ]: #=====
#Nafn: Orrí Leví Úlfarsson
#email: olu2@hi.is
#Nafn: Sigursteinn Pálsson Enos
#email: spe3@hi.is
#=====
```

```
In [1]: #Fall sem tekur inn test fylki og thetur
#teiknar síðan confusion matrix og fjölda villa
#skilar síðan fjölda villna
def Prufun(Test, supaArray):
    matrix = np.full((11,11),0)
    temp = 0
    current = 0
    errs = 0
    for l in range(len(y_test)):
        sum = -1
        for i in range(0,10):
            temp = np.dot(Test[l,:],supaArray[i,:])
            if temp > sum:
                sum = temp
                current = i
        if current != y_test[l]:
            errs += 1
        matrix[int(y_test[l]),current] += 1
        matrix[int(y_test[l]),10] += 1
        matrix[10,current] += 1
        matrix[10,10] +=1
    print("Fjöldi villna:",errs)
    print(matrix)
    return(errs)
```

```
In [2]: #tekur inn fylkið x, stærðina á q og random fylkið R
#margfaldar x við R og finnur síðan út hvort það eigi að setja inn 0 eða tölu
#skilar frá sér fylkinu x breyttu
def newMatt(x,q, R):
    newMat = np.full((x.shape[0], 785+q),0)
    for i in range(x.shape[0]):
        g = np.dot(x[i,:],R)
        for j in range(785):
            newMat[i,j] = x[i,j]
        for k in range(q):
            if g[k] <= 0:
                newMat[i,k+784] = 0
            else:
                newMat[i,k+784] = g[k]
    return newMat
```

```
In [16]: #tekur inn fylkið newMat
#Það finnur A og B með thetuna 1000 og finnur út Ax-B
#skilar frá sér fylki af thetum/vægi lambda
def lambMat(newMat):
    lamb = 1000

    x_train2 = np.delete(newMat,0,axis=1)

    A = np.vstack((x_train2, np.identity(784+q)*lamb))
    zeros = np.full((784+q),0)

    supaArray2 = np.empty((10,784+q))

    for i in range(0,10):
        y_temp = y_train.copy()
        for j in range(len(y_train)):
            if y_temp[j] == i:
                y_temp[j] = 1
            else:
                y_temp[j] = -1
        B = np.hstack((y_temp,zeros))
        supaArray2[i,:784+q],_,_,_ = LA.lstsq(A,B)
    return supaArray2
```

```

In [4]: import numpy as np
        from numpy import linalg as LA

        data=np.load('docmatrix.npz')
        X=data['X']
        y=data['y']
        terms=data['terms']

        #1.a
        # Skipta í gögnum í þjálfunar- og prófunargögn
        n=X.shape[0]
        rnd=np.random.permutation(n) # Slembin umröðun talnanna 1,...,n
        nfrac=0.7 # Hlutfall gagna sem er notað til þjálfunar
        n_train=int(nfrac*n)
        x_train=X[rnd[0:n_train],:]
        y_train=y[rnd[0:n_train]]
        x_test=X[rnd[n_train:],:]
        y_test=y[rnd[n_train:]]
        #fyrst þarf að flokka í 2 flokka. svo við breyttum menntun og Harry Potter
        í -1
        for i in range(len(y_train)):
            if (y_train[i] != 1):
                y_train[i] = -1;

        #Síðan er fundið vigur minnstu kvaðrata
        N = x_train.shape[0]
        A = np.c_[np.ones(N), x_train]

        theta,_,_,_ = LA.lstsq(A, y_train)

        #Síðan er prentað dot.productið og miðað þeim við hvort annað.
        #print(np.dot(x_train,r))
        #Því hærrí sem talan er því líklegri er hún til þess að vera í ípróttum
        #Þægilegasta leiðin til að horfa á þetta er að einfaldlega fylgjast með for
        merkjunum
        # mínus = ekki ípróttir
        # plús = ípróttir
        N = x_test.shape[0]
        A = np.c_[np.ones(N), x_test]
        print(A.shape)
        print(theta.shape)
        z = np.dot(A,theta)
        #print(z)
        theta = np.delete(theta,0)

        #1.b
        sortedList = (np.argsort(theta))

        #mismunandi gögn við hvert skipti en virðast mikilvæg
        for i in range(0,10):
            print(terms[sortedList[999-i]])

```

(180, 1001)  
(1001,)  
later  
alreadi  
seem  
increas  
major  
stori  
put  
must  
part  
teach

```
In [5]: import numpy as np
import matplotlib.pyplot as plt

data=np.load('mnist_small.npz')
x_train=data['x_train']
y_train=data['y_train']
x_test=data['x_test']
y_test=data['y_test']
n_train = len(y_train)
n_test = len(y_test)

#Hérna er bætt fastanum við
x_train=np.c_[np.ones(n_train), x_train]
x_test=np.c_[np.ones(n_test), x_test]
print(x_train.shape)
print(x_test.shape)

#2A. Hérna er hannað fylkið með 10 mismunandi lóðunum til að sjá hver tala
er líklegust
supaArray = np.empty((10,785))

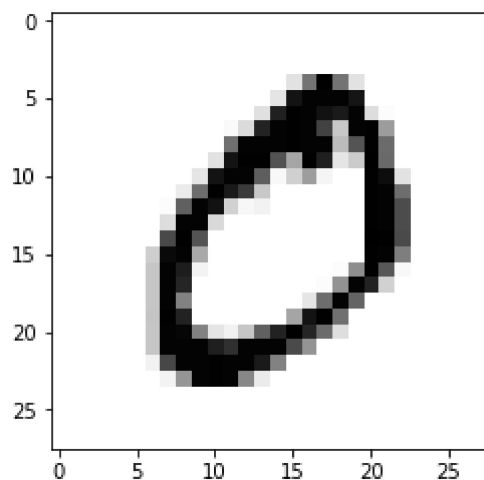
for i in range(0,10):
    y_temp = y_train.copy()
    for j in range(len(y_train)):
        if y_temp[j] == i:
            y_temp[j] = 1
        else:
            y_temp[j] = -1
    supaArray[i,:785],_,_,_ = LA.lstsq(x_train,y_temp)

print(supaArray.shape)

#Lítið prófunarfall, giskar hvað því þykir líklegasta talan
sum = -100
current = -1
for i in range(0,10):
    temp = np.dot(x_train[1,:],supaArray[i,:])
    if temp > sum:
        sum = temp
        current = i

print(current)
plotimage = np.delete(x_train,0,axis=1)
plt.imshow(plotimage[1,:].reshape(28,28),cmap='Greys') # Fyrsta myndin í þj
álfunargögnunum
plt.show()
```

```
(10000, 785)
(1000, 785)
(10, 785)
0
```



```
In [6]: #2B Hérna prófum við þjálfunarforritið til endans og fáum út villutíðni og got
t confusion matrix
fjoldierrors = Prufun(x_test,supaArray)
```

Fjöldi villna: 204

```
[ [ 82  0  0  0  0  0  2  0  1  0  85]
  [  0 122  0  1  1  0  1  0  1  0 126]
  [  2 13 75  4  0  0  4  6 12  0 116]
  [  1  0  2 86  2  3  2  3  5  3 107]
  [  0  2  1  0 95  3  1  0  2  6 110]
  [  4  1  0 14  1 50  3  3 10  1  87]
  [  3  2  0  0  3  4 75  0  0  0  87]
  [  0  7  2  1  3  0  0 78  0  8  99]
  [  0  3  3  7  5  2  1  1 65  2  89]
  [  0  2  0  2 10  0  0  9  3 68  94]
  [ 92 152 83 115 120 62 89 100 99 88 1000]]
```

```

In [9]: #2C. Hérna var prófað "featured engineering addition" dótið. Við prófuðum 100,
        1000, 5000 og 10000.
        #Það besta sem virkaði hjá okkur var 1000

q=1000
R = (np.random.randint(low=0, high=2,size=(785,q))*2)-1
newR = np.delete(R,0,axis=0)

#Hér er hannað x_train með features
newTrain = np.full((x_train.shape[0], 785+q),0)
newTrain = newMatt(x_train,q,R)

#Hérna er hannað test með sömu features
newTest = np.full((x_test.shape[0], 785+q),0)
newTest = newMatt(x_test,q,R)

#Hérna er hannað thetafylkið með features
supaArray = np.empty((10,785+q))
for i in range(0,10):
    y_temp = y_train.copy()
    for j in range(len(y_train)):
        if y_temp[j] == i:
            y_temp[j] = 1
        else:
            y_temp[j] = -1
    supaArray[i,:785+q],_,_,_ = LA.lstsq(newTrain,y_temp)

#Hérna er hannað test með sömu features
print("done2")

```

done2

```

In [10]: #Í þessu celli prófuðum við mismunandi q-in
        errors = Prufun(newTest, supaArray)

```

Fjöldi villna: 91

```

[[ 84   0   0   0   0   0   1   0   0   0   85]
 [  0 124   1   1   0   0   0   0   0   0 126]
 [  1   0  97   1   0   1   1   3  12   0 116]
 [  0   1   0  94   0   7   0   2   1   2 107]
 [  0   0   2   0 102   0   0   0   2   4 110]
 [  1   0   0   3   1  78   1   0   2   1  87]
 [  3   0   0   0   1   2  81   0   0   0  87]
 [  0   1   2   1   3   0   0  89   1   2  99]
 [  1   0   2   4   2   1   0   3  75   1  89]
 [  0   0   0   0   4   1   0   1   3  85  94]
 [ 90 126 104 104 113 90  84  98  96  95 1000]]

```

```
In [11]: #2D. Hérna er Lambdafallið okkar. Þetta er upprunarlega fallið og við tökum í
          burt fastann að framan
          lambMat = np.empty((8,1))
          errorMat = np.empty((8,1))
          lamb = 0.1
          for index in range(8):

              lamb = lamb*10
              lambMat[index,0] = lamb
              x_train2 = np.delete(x_train,0,axis=1)

              A = np.vstack((x_train2, np.identity(784)*lamb))
              zeros = np.full((784),0)

              supaArray2 = np.empty((10,784))

              for i in range(0,10):
                  y_temp = y_train.copy()
                  for j in range(len(y_train)):
                      if y_temp[j] == i:
                          y_temp[j] = 1
                      else:
                          y_temp[j] = -1
                  B = np.hstack((y_temp,zeros))
                  supaArray2[i,:784],_,_,_ = LA.lstsq(A,B)

              x_test2 = np.delete(x_test,0,axis=1)

          #Hérna er annað prófunarfall sem fyllir inn villutiðni fyrir framtíðar plot
          error = Prufun(x_test2, supaArray2)
          errorMat[index,0] = error/1000
          print(errorMat)
          print(lambMat)
```



Fjöldi villna: 205

```

[[ 79  0  1  0  0  0  2  0  3  0  85]
 [  0 122  0  1  1  0  1  0  1  0 126]
 [  2 12 75  5  1  0  3  6 12  0 116]
 [  0  1  2 86  1  3  2  4  5  3 107]
 [  0  2  0  0 94  2  1  0  4  7 110]
 [  4  1  0 14  1 48  2  4 12  1  87]
 [  3  2  0  0  2  2 76  0  2  0  87]
 [  0  7  1  0  4  0  0 79  0  8  99]
 [  0  4  3  7  4  1  0  2 66  2  89]
 [  0  2  0  2  9  0  0  9  2 70  94]
 [ 88 153 82 115 117 56 87 104 107 91 1000]]

```

Fjöldi villna: 203

```

[[ 79  0  1  0  0  0  2  0  3  0  85]
 [  0 122  0  1  1  0  1  0  1  0 126]
 [  2 12 76  4  1  0  3  6 12  0 116]
 [  0  1  2 86  1  3  2  4  5  3 107]
 [  0  2  0  0 94  2  1  0  4  7 110]
 [  4  1  0 14  1 49  2  3 12  1  87]
 [  3  2  0  0  2  2 76  0  2  0  87]
 [  0  7  1  0  4  0  0 79  0  8  99]
 [  0  4  3  7  4  1  0  2 66  2  89]
 [  0  2  0  2  9  0  0  9  2 70  94]
 [ 88 153 83 114 117 57 87 103 107 91 1000]]

```

Fjöldi villna: 197

```

[[ 79  0  1  0  0  0  2  0  3  0  85]
 [  0 122  0  1  1  0  1  0  1  0 126]
 [  2 11 77  4  0  0  3  5 13  1 116]
 [  0  1  3 88  0  3  2  3  4  3 107]
 [  0  2  0  0 94  2  1  0  4  7 110]
 [  4  1  0 14  1 51  2  3 10  1  87]
 [  3  2  0  0  2  1 77  0  2  0  87]
 [  0  7  1  1  3  0  0 79  0  8  99]
 [  0  4  3  7  4  1  0  2 66  2  89]
 [  0  2  0  2  9  0  0  9  2 70  94]
 [ 88 152 85 117 114 58 88 101 105 92 1000]]

```

Fjöldi villna: 192

```

[[ 80  0  0  0  0  0  2  0  3  0  85]
 [  0 122  0  1  1  0  1  0  1  0 126]
 [  1 11 80  4  0  0  4  5 10  1 116]
 [  0  1  1 88  0  3  3  3  5  3 107]
 [  0  2  0  0 94  2  1  0  4  7 110]
 [  3  1  0 14  1 50  2  4 11  1  87]
 [  3  2  0  0  2  1 77  0  2  0  87]
 [  0  6  1  1  3  0  0 81  0  7  99]
 [  1  3  3  6  6  1  0  2 64  3  89]
 [  0  1  0  2  8  0  0  8  3 72  94]
 [ 88 149 85 116 115 57 90 103 103 94 1000]]

```

Fjöldi villna: 197

```

[[ 80  0  0  0  0  0  3  0  2  0  85]
 [  0 122  0  1  0  1  1  0  1  0 126]
 [  1 15 80  3  0  0  2  5  9  1 116]
 [  0  0  1 90  0  3  4  3  5  1 107]
 [  1  1  1  0 92  1  2  0  3  9 110]
 [  4  2  0 20  3 42  4  4  7  1  87]
 [  4  3  1  0  1  1 76  0  1  0  87]
 [  1  9  2  1  2  0  0 78  0  6  99]

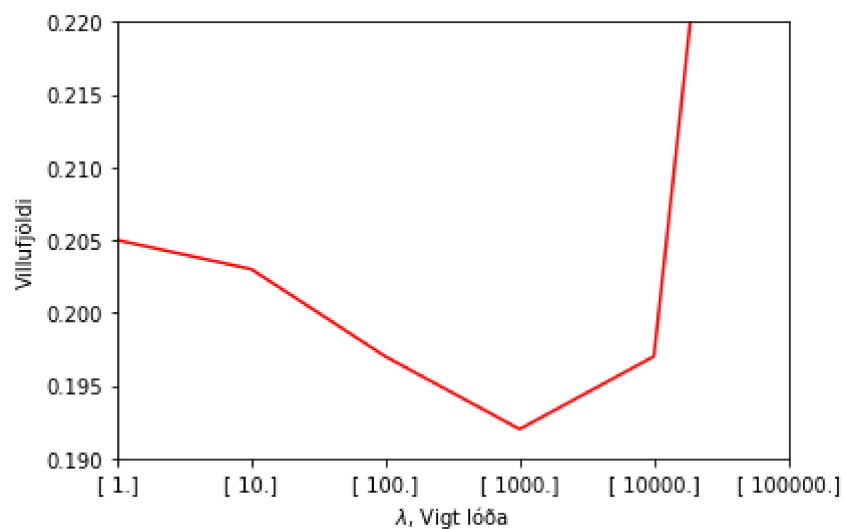
```

```

[ 0 4 1 7 2 0 4 4 65 2 89]
[ 0 1 0 2 7 0 0 5 1 78 94]
[ 91 157 86 124 107 48 96 99 94 98 1000]]
Fjöldi villna: 282
[[ 82 0 0 1 0 0 2 0 0 0 85]
[ 0 119 0 4 0 0 2 0 1 0 126]
[ 11 12 71 8 0 0 4 5 5 0 116]
[ 2 1 1 93 0 0 4 3 2 1 107]
[ 1 2 0 1 79 0 6 0 1 20 110]
[ 20 3 0 38 1 1 3 9 10 2 87]
[ 8 1 1 0 1 0 75 0 1 0 87]
[ 2 11 2 1 1 0 0 79 0 3 99]
[ 5 2 2 22 0 0 2 5 49 2 89]
[ 1 3 0 5 4 0 0 9 2 70 94]
[ 132 154 77 173 86 1 98 110 71 98 1000]]
Fjöldi villna: 335
[[ 83 0 0 2 0 0 0 0 0 0 85]
[ 0 107 1 12 0 0 1 0 5 0 126]
[ 17 1 69 16 0 0 3 2 8 0 116]
[ 4 0 1 95 0 0 2 0 3 2 107]
[ 8 0 0 6 52 0 12 0 5 27 110]
[ 48 0 0 28 0 0 1 3 7 0 87]
[ 20 0 0 0 0 0 67 0 0 0 87]
[ 5 5 1 5 0 0 1 72 6 4 99]
[ 11 0 0 26 0 0 1 1 48 2 89]
[ 4 1 0 5 0 0 0 7 5 72 94]
[ 200 114 72 195 52 0 88 85 87 107 1000]]
Fjöldi villna: 335
[[ 83 0 0 2 0 0 0 0 0 0 85]
[ 0 107 1 12 0 0 1 0 5 0 126]
[ 18 1 69 15 0 0 3 2 8 0 116]
[ 4 0 1 96 0 0 2 0 3 1 107]
[ 9 0 0 6 51 0 12 0 5 27 110]
[ 48 0 0 28 0 0 1 3 7 0 87]
[ 20 0 0 0 0 0 67 0 0 0 87]
[ 5 4 1 6 0 0 1 72 6 4 99]
[ 11 0 0 26 0 0 1 1 48 2 89]
[ 4 1 0 5 0 0 0 7 5 72 94]
[ 202 113 72 196 51 0 88 85 87 106 1000]]
[[ 0.205]
[ 0.203]
[ 0.197]
[ 0.192]
[ 0.197]
[ 0.282]
[ 0.335]
[ 0.335]]
[[ 1.00000000e+00]
[ 1.00000000e+01]
[ 1.00000000e+02]
[ 1.00000000e+03]
[ 1.00000000e+04]
[ 1.00000000e+05]
[ 1.00000000e+06]
[ 1.00000000e+07]]

```

```
In [12]: #Hérna er plottað  
#Virðist vera að sirka 1000 lambda er besta valið  
intMat = ([1,2,3,4,5,6,7,8])  
plt.plot(intMat,errorMat,color="red")  
plt.xlabel("$\lambda$, Vigt lóða")  
plt.ylabel("Villufjöldi")  
plt.xticks(intMat,lambMat)  
axes = plt.gca()  
axes.set_ylim([0.19,0.22])  
axes.set_xlim([1,6])  
plt.show()
```



```
In [17]: #2e
#Gerum alveg eins og við gerðum áðan nema bæði random tölur og lambda
q=1000
R = (np.random.randint(low=0, high=2,size=(785,q))*2)-1
newR = np.delete(R,0,axis=0)

newTrain = np.full((x_train.shape[0], 785+q),0)
newTrain = newMatt(x_train,q,R)

newTest = np.full((x_test.shape[0], 785+q),0)
newTest = newMatt(x_test,q,R)
newTest = np.delete(newTest,0,axis=1)

supaArray = np.empty((10,784))
supaArray = lambMat(newTrain)

errors = Prufun(newTest, supaArray)
#Þetta gekk eh ekki upp, vitum ekki alveg hvað við vorum að gera vitlaust
```

Fjöldi villna: 82

```
[ [ 83  0  0  0  0  0  2  0  0  0  85]
  [  0 124  0  1  0  0  0  0  1  0 126]
  [  1  2 102  2  0  0  1  2  6  0 116]
  [  0  0  3 92  0  5  3  2  1  1 107]
  [  0  1  0  0 97  0  2  0  2  8 110]
  [  0  0  1  2  0 82  1  0  1  0  87]
  [  3  0  1  0  1  1 81  0  0  0  87]
  [  0  2  2  1  1  0  0 92  1  0  99]
  [  0  0  2  2  3  1  0  2 78  1  89]
  [  0  1  0  1  2  1  0  0  2 87  94]
  [ 87 130 111 101 104 90 90 98 92 97 1000]]
```