a02_4

October 24, 2025

1 4 Maximum Aposteriori Estimation

Report of Jonas Ortner: 2265527 Marmee Pandya: 1963521

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
import sklearn

%load_ext autoreload
%autoreload 2

from a02_helper import *
from a02_functions import l, l_l2, dl_l2, gd, gd_l2, logsigma, classify,␣
↪optimize

from sklearn.preprocessing import StandardScaler
import pandas as  pd
```

1.1 4a Gradient Descent

Implement the function returning the log-density of the posterior of logistic regression, regularized with parameter lambda, in `a02_functions.py`. Then test it below.

[2]:
```
# this should give:
# [-47066.641667825766, -47312.623810682911]
[l_l2(y, Xz, np.linspace(-5, 5, D), 0), l_l2(y, Xz, np.linspace(-5, 5, D), 1)]
```

[2]:
```
[-47066.64166782574, -47312.62381068288]
```

Now implement the function to obtain its gradient and test it, in the same manner as above.

[3]:
```
# this should give:
# [array([  551.33985842,   143.84116318,   841.83373606,   156.87237578,
#           802.61217579,   795.96202907,   920.69045803,   621.96516752,
#           659.18724769,   470.81259805,   771.32406968,   352.40325626,
#           455.66972482,   234.36600888,   562.45454038,   864.83981264,
#           787.19723703,   649.48042176,   902.6478154 ,   544.00539886,
#          1174.78638035,   120.3598967 ,   839.61141672,   633.30453444,
#          -706.66815087,  -630.2039816 ,  -569.3451386 ,  -527.50996698,
```

```
#         -359.53701083,  -476.64334832,  -411.60620464,  -375.11950586,
#         -345.37195689,  -376.22044258,  -407.31761977,  -456.23251936,
#         -596.86960184,  -107.97072355,  -394.82170044,  -229.18125598,
#         -288.46356547,  -362.13402385,  -450.87896465,  -277.03932676,
#         -414.99293368,  -452.28771693,  -167.54649092,  -270.9043748 ,
#         -252.20140951,  -357.72497343,  -259.12468742,   418.35938483,
#          604.54173228,    43.10390907,   152.24258478,   378.16731033,
#          416.12032881]),
#  array([ 556.33985842,   148.66259175,   846.4765932 ,   161.33666149,
#          806.89789007,   800.06917193,   924.61902946,   625.71516752,
#          662.75867626,   474.20545519,   774.5383554 ,   355.43897054,
#          458.52686767,   237.04458031,   564.95454038,   867.16124121,
#          789.34009417,   651.44470748,   904.43352968,   545.61254171,
#         1176.21495178,   121.6098967 ,   840.68284529,   634.19739158,
#         -705.95386516,  -629.66826731,  -568.98799574,  -527.33139555,
#         -359.53701083,  -476.82191975,  -411.9633475 ,  -375.65522015,
#         -346.08624261,  -377.11329972,  -408.38904835,  -457.48251936,
#         -598.29817327,  -109.57786641,  -396.60741472,  -231.14554169,
#         -290.60642261,  -364.45545242,  -453.37896465,  -279.71789819,
#         -417.85007654,  -455.32343122,  -170.76077664,  -274.29723194,
#         -255.77283808,  -361.47497343,  -263.05325885,   414.25224198,
#          600.25601799,    38.63962335,   147.59972763,   373.34588176,
#          411.12032881])]
[dl_l2(y, Xz, np.linspace(-5, 5, D), 0), dl_l2(y, Xz, np.linspace(-5, 5, D), 1)]
```

[3]:

```
[array([ 551.33985842,  143.84116318,  841.83373606,  156.87237578,
         802.61217579,  795.96202907,  920.69045803,  621.96516752,
         659.18724769,  470.81259805,  771.32406968,  352.40325626,
         455.66972482,  234.36600888,  562.45454038,  864.83981264,
         787.19723703,  649.48042176,  902.6478154 ,  544.00539886,
        1174.78638035,  120.3598967 ,  839.61141672,  633.30453444,
        -706.66815087, -630.2039816 , -569.3451386 , -527.50996698,
        -359.53701083, -476.64334832, -411.60620464, -375.11950586,
        -345.37195689, -376.22044258, -407.31761977, -456.23251936,
        -596.86960184, -107.97072355, -394.82170044, -229.18125598,
        -288.46356547, -362.13402385, -450.87896465, -277.03932676,
        -414.99293368, -452.28771693, -167.54649092, -270.9043748 ,
        -252.20140951, -357.72497343, -259.12468742,  418.35938483,
         604.54173228,   43.10390907,  152.24258478,  378.16731033,
         416.12032881]),
 array([ 556.33985842,  148.66259175,  846.4765932 ,  161.33666149,
         806.89789007,  800.06917193,  924.61902946,  625.71516752,
         662.75867626,  474.20545519,  774.5383554 ,  355.43897054,
         458.52686767,  237.04458031,  564.95454038,  867.16124121,
         789.34009417,  651.44470748,  904.43352968,  545.61254171,
        1176.21495178,  121.6098967 ,  840.68284529,  634.19739158,
        -705.95386516, -629.66826731, -568.98799574, -527.33139555,
```

```
       -359.53701083, -476.82191975, -411.9633475 , -375.65522015,
       -346.08624261, -377.11329972, -408.38904835, -457.48251936,
       -598.29817327, -109.57786641, -396.60741472, -231.14554169,
       -290.60642261, -364.45545242, -453.37896465, -279.71789819,
       -417.85007654, -455.32343122, -170.76077664, -274.29723194,
       -255.77283808, -361.47497343, -263.05325885,  414.25224198,
        600.25601799,   38.63962335,  147.59972763,  373.34588176,
        411.12032881])]
```

Now define the (f,update) tuple handed to the `optimize` function for gradient descent on logistic regression with L2 regularization. Then run it below.

[4]:
```python
# let's run!
lambda_ = 100
w0 = np.random.normal(size=D)
wz_gd_l2, vz_gd_l2, ez_gd_l2 = optimize(gd_l2(y, Xz, lambda_), w0, nepochs=500)
```

```
Epoch   0: f=  7159.877, eps=0.010000000
Epoch   1: f= 16414.030, eps=0.005000000
Epoch   2: f=  3756.368, eps=0.005250000
Epoch   3: f=  1310.198, eps=0.005512500
Epoch   4: f=  1149.990, eps=0.005788125
Epoch   5: f=  1605.227, eps=0.002894063
Epoch   6: f=  1498.673, eps=0.003038766
Epoch   7: f=  1039.586, eps=0.003190704
Epoch   8: f=  1027.892, eps=0.003350239
Epoch   9: f=  1046.823, eps=0.001675120
Epoch  10: f=   996.131, eps=0.001758876
Epoch  11: f=   988.957, eps=0.001846819
Epoch  12: f=   988.582, eps=0.001939160
Epoch  13: f=   988.531, eps=0.002036118
Epoch  14: f=   988.519, eps=0.002137924
Epoch  15: f=   988.515, eps=0.002244820
Epoch  16: f=   988.513, eps=0.002357061
Epoch  17: f=   988.512, eps=0.002474914
Epoch  18: f=   988.512, eps=0.002598660
Epoch  19: f=   988.512, eps=0.002728593
Epoch  20: f=   988.512, eps=0.002865023
Epoch  21: f=   988.512, eps=0.003008274
Epoch  22: f=   988.512, eps=0.001504137
Epoch  23: f=   988.512, eps=0.001579344
Epoch  24: f=   988.512, eps=0.001658311
Epoch  25: f=   988.512, eps=0.001741227
Epoch  26: f=   988.512, eps=0.001828288
Epoch  27: f=   988.512, eps=0.001919702
Epoch  28: f=   988.512, eps=0.002015687
Epoch  29: f=   988.512, eps=0.002116472
Epoch  30: f=   988.512, eps=0.002222295
```

```
Epoch  31: f=   988.512, eps=0.002333410
Epoch  32: f=   988.512, eps=0.002450081
Epoch  33: f=   988.512, eps=0.002572585
Epoch  34: f=   988.512, eps=0.002701214
Epoch  35: f=   988.512, eps=0.002836275
Epoch  36: f=   988.512, eps=0.002978088
Epoch  37: f=   988.512, eps=0.003126993
Epoch  38: f=   988.512, eps=0.003283342
Epoch  39: f=   988.512, eps=0.003447510
Epoch  40: f=   988.512, eps=0.003619885
Epoch  41: f=   988.512, eps=0.003800879
Epoch  42: f=   988.512, eps=0.003990923
Epoch  43: f=   988.512, eps=0.001995462
Epoch  44: f=   988.512, eps=0.002095235
Epoch  45: f=   988.512, eps=0.002199996
Epoch  46: f=   988.512, eps=0.002309996
Epoch  47: f=   988.512, eps=0.002425496
Epoch  48: f=   988.512, eps=0.001212748
Epoch  49: f=   988.512, eps=0.001273385
Epoch  50: f=   988.512, eps=0.000636693
Epoch  51: f=   988.512, eps=0.000668527
Epoch  52: f=   988.512, eps=0.000701954
Epoch  53: f=   988.512, eps=0.000350977
Epoch  54: f=   988.512, eps=0.000175488
Epoch  55: f=   988.512, eps=0.000184263
Epoch  56: f=   988.512, eps=0.000092131
Epoch  57: f=   988.512, eps=0.000046066
Epoch  58: f=   988.512, eps=0.000048369
Epoch  59: f=   988.512, eps=0.000050787
Epoch  60: f=   988.512, eps=0.000025394
Epoch  61: f=   988.512, eps=0.000026663
Epoch  62: f=   988.512, eps=0.000013332
Epoch  63: f=   988.512, eps=0.000013998
Epoch  64: f=   988.512, eps=0.000014698
Epoch  65: f=   988.512, eps=0.000007349
Epoch  66: f=   988.512, eps=0.000007717
Epoch  67: f=   988.512, eps=0.000003858
Epoch  68: f=   988.512, eps=0.000004051
Epoch  69: f=   988.512, eps=0.000004254
Epoch  70: f=   988.512, eps=0.000002127
Epoch  71: f=   988.512, eps=0.000001063
Epoch  72: f=   988.512, eps=0.000001117
Epoch  73: f=   988.512, eps=0.000000558
Epoch  74: f=   988.512, eps=0.000000279
Epoch  75: f=   988.512, eps=0.000000140
Epoch  76: f=   988.512, eps=0.000000147
Epoch  77: f=   988.512, eps=0.000000073
Epoch  78: f=   988.512, eps=0.000000077
```

```
Epoch  79: f=   988.512, eps=0.000000038
Epoch  80: f=   988.512, eps=0.000000019
Epoch  81: f=   988.512, eps=0.000000020
Epoch  82: f=   988.512, eps=0.000000021
Epoch  83: f=   988.512, eps=0.000000022
Epoch  84: f=   988.512, eps=0.000000023
Epoch  85: f=   988.512, eps=0.000000025
Epoch  86: f=   988.512, eps=0.000000026
Epoch  87: f=   988.512, eps=0.000000027
Epoch  88: f=   988.512, eps=0.000000028
Epoch  89: f=   988.512, eps=0.000000014
Epoch  90: f=   988.512, eps=0.000000015
Epoch  91: f=   988.512, eps=0.000000016
Epoch  92: f=   988.512, eps=0.000000016
Epoch  93: f=   988.512, eps=0.000000008
Epoch  94: f=   988.512, eps=0.000000009
Epoch  95: f=   988.512, eps=0.000000009
Epoch  96: f=   988.512, eps=0.000000010
Epoch  97: f=   988.512, eps=0.000000010
Epoch  98: f=   988.512, eps=0.000000005
Epoch  99: f=   988.512, eps=0.000000005
Epoch 100: f=   988.512, eps=0.000000006
Epoch 101: f=   988.512, eps=0.000000006
Epoch 102: f=   988.512, eps=0.000000003
Epoch 103: f=   988.512, eps=0.000000003
Epoch 104: f=   988.512, eps=0.000000002
Epoch 105: f=   988.512, eps=0.000000002
Epoch 106: f=   988.512, eps=0.000000002
Epoch 107: f=   988.512, eps=0.000000001
Epoch 108: f=   988.512, eps=0.000000001
Epoch 109: f=   988.512, eps=0.000000001
Epoch 110: f=   988.512, eps=0.000000001
Epoch 111: f=   988.512, eps=0.000000001
Epoch 112: f=   988.512, eps=0.000000001
Epoch 113: f=   988.512, eps=0.000000001
Epoch 114: f=   988.512, eps=0.000000001
Epoch 115: f=   988.512, eps=0.000000001
Epoch 116: f=   988.512, eps=0.000000001
Epoch 117: f=   988.512, eps=0.000000001
Epoch 118: f=   988.512, eps=0.000000001
Epoch 119: f=   988.512, eps=0.000000001
Epoch 120: f=   988.512, eps=0.000000001
Epoch 121: f=   988.512, eps=0.000000001
Epoch 122: f=   988.512, eps=0.000000000
Epoch 123: f=   988.512, eps=0.000000000
Epoch 124: f=   988.512, eps=0.000000000
Epoch 125: f=   988.512, eps=0.000000000
Epoch 126: f=   988.512, eps=0.000000000
```

```
Epoch 127: f=   988.512, eps=0.000000000
Epoch 128: f=   988.512, eps=0.000000000
Epoch 129: f=   988.512, eps=0.000000000
Epoch 130: f=   988.512, eps=0.000000000
Epoch 131: f=   988.512, eps=0.000000000
Epoch 132: f=   988.512, eps=0.000000000
Epoch 133: f=   988.512, eps=0.000000000
Epoch 134: f=   988.512, eps=0.000000000
Epoch 135: f=   988.512, eps=0.000000000
Epoch 136: f=   988.512, eps=0.000000000
Epoch 137: f=   988.512, eps=0.000000000
Epoch 138: f=   988.512, eps=0.000000000
Epoch 139: f=   988.512, eps=0.000000000
Epoch 140: f=   988.512, eps=0.000000000
Epoch 141: f=   988.512, eps=0.000000000
Epoch 142: f=   988.512, eps=0.000000000
Epoch 143: f=   988.512, eps=0.000000000
Epoch 144: f=   988.512, eps=0.000000000
Epoch 145: f=   988.512, eps=0.000000000
Epoch 146: f=   988.512, eps=0.000000000
Epoch 147: f=   988.512, eps=0.000000000
Epoch 148: f=   988.512, eps=0.000000000
Epoch 149: f=   988.512, eps=0.000000000
Epoch 150: f=   988.512, eps=0.000000000
Epoch 151: f=   988.512, eps=0.000000000
Epoch 152: f=   988.512, eps=0.000000000
Epoch 153: f=   988.512, eps=0.000000000
Epoch 154: f=   988.512, eps=0.000000000
Epoch 155: f=   988.512, eps=0.000000000
Epoch 156: f=   988.512, eps=0.000000000
Epoch 157: f=   988.512, eps=0.000000000
Epoch 158: f=   988.512, eps=0.000000000
Epoch 159: f=   988.512, eps=0.000000000
Epoch 160: f=   988.512, eps=0.000000000
Epoch 161: f=   988.512, eps=0.000000000
Epoch 162: f=   988.512, eps=0.000000000
Epoch 163: f=   988.512, eps=0.000000000
Epoch 164: f=   988.512, eps=0.000000000
Epoch 165: f=   988.512, eps=0.000000000
Epoch 166: f=   988.512, eps=0.000000000
Epoch 167: f=   988.512, eps=0.000000000
Epoch 168: f=   988.512, eps=0.000000000
Epoch 169: f=   988.512, eps=0.000000000
Epoch 170: f=   988.512, eps=0.000000000
Epoch 171: f=   988.512, eps=0.000000000
Epoch 172: f=   988.512, eps=0.000000000
Epoch 173: f=   988.512, eps=0.000000000
Epoch 174: f=   988.512, eps=0.000000000
```

```
Epoch 175: f=   988.512, eps=0.000000000
Epoch 176: f=   988.512, eps=0.000000000
Epoch 177: f=   988.512, eps=0.000000000
Epoch 178: f=   988.512, eps=0.000000000
Epoch 179: f=   988.512, eps=0.000000000
Epoch 180: f=   988.512, eps=0.000000000
Epoch 181: f=   988.512, eps=0.000000000
Epoch 182: f=   988.512, eps=0.000000000
Epoch 183: f=   988.512, eps=0.000000000
Epoch 184: f=   988.512, eps=0.000000000
Epoch 185: f=   988.512, eps=0.000000000
Epoch 186: f=   988.512, eps=0.000000000
Epoch 187: f=   988.512, eps=0.000000000
Epoch 188: f=   988.512, eps=0.000000000
Epoch 189: f=   988.512, eps=0.000000000
Epoch 190: f=   988.512, eps=0.000000000
Epoch 191: f=   988.512, eps=0.000000000
Epoch 192: f=   988.512, eps=0.000000000
Epoch 193: f=   988.512, eps=0.000000000
Epoch 194: f=   988.512, eps=0.000000000
Epoch 195: f=   988.512, eps=0.000000000
Epoch 196: f=   988.512, eps=0.000000000
Epoch 197: f=   988.512, eps=0.000000000
Epoch 198: f=   988.512, eps=0.000000000
Epoch 199: f=   988.512, eps=0.000000000
Epoch 200: f=   988.512, eps=0.000000000
Epoch 201: f=   988.512, eps=0.000000000
Epoch 202: f=   988.512, eps=0.000000000
Epoch 203: f=   988.512, eps=0.000000000
Epoch 204: f=   988.512, eps=0.000000000
Epoch 205: f=   988.512, eps=0.000000000
Epoch 206: f=   988.512, eps=0.000000000
Epoch 207: f=   988.512, eps=0.000000000
Epoch 208: f=   988.512, eps=0.000000000
Epoch 209: f=   988.512, eps=0.000000000
Epoch 210: f=   988.512, eps=0.000000000
Epoch 211: f=   988.512, eps=0.000000000
Epoch 212: f=   988.512, eps=0.000000000
Epoch 213: f=   988.512, eps=0.000000000
Epoch 214: f=   988.512, eps=0.000000000
Epoch 215: f=   988.512, eps=0.000000000
Epoch 216: f=   988.512, eps=0.000000000
Epoch 217: f=   988.512, eps=0.000000000
Epoch 218: f=   988.512, eps=0.000000000
Epoch 219: f=   988.512, eps=0.000000000
Epoch 220: f=   988.512, eps=0.000000000
Epoch 221: f=   988.512, eps=0.000000000
Epoch 222: f=   988.512, eps=0.000000000
```

```
Epoch 223: f=   988.512, eps=0.000000000
Epoch 224: f=   988.512, eps=0.000000000
Epoch 225: f=   988.512, eps=0.000000000
Epoch 226: f=   988.512, eps=0.000000000
Epoch 227: f=   988.512, eps=0.000000000
Epoch 228: f=   988.512, eps=0.000000000
Epoch 229: f=   988.512, eps=0.000000000
Epoch 230: f=   988.512, eps=0.000000000
Epoch 231: f=   988.512, eps=0.000000000
Epoch 232: f=   988.512, eps=0.000000000
Epoch 233: f=   988.512, eps=0.000000000
Epoch 234: f=   988.512, eps=0.000000000
Epoch 235: f=   988.512, eps=0.000000000
Epoch 236: f=   988.512, eps=0.000000000
Epoch 237: f=   988.512, eps=0.000000000
Epoch 238: f=   988.512, eps=0.000000000
Epoch 239: f=   988.512, eps=0.000000000
Epoch 240: f=   988.512, eps=0.000000000
Epoch 241: f=   988.512, eps=0.000000000
Epoch 242: f=   988.512, eps=0.000000000
Epoch 243: f=   988.512, eps=0.000000000
Epoch 244: f=   988.512, eps=0.000000000
Epoch 245: f=   988.512, eps=0.000000000
Epoch 246: f=   988.512, eps=0.000000000
Epoch 247: f=   988.512, eps=0.000000000
Epoch 248: f=   988.512, eps=0.000000000
Epoch 249: f=   988.512, eps=0.000000000
Epoch 250: f=   988.512, eps=0.000000000
Epoch 251: f=   988.512, eps=0.000000000
Epoch 252: f=   988.512, eps=0.000000000
Epoch 253: f=   988.512, eps=0.000000000
Epoch 254: f=   988.512, eps=0.000000000
Epoch 255: f=   988.512, eps=0.000000000
Epoch 256: f=   988.512, eps=0.000000000
Epoch 257: f=   988.512, eps=0.000000000
Epoch 258: f=   988.512, eps=0.000000000
Epoch 259: f=   988.512, eps=0.000000000
Epoch 260: f=   988.512, eps=0.000000000
Epoch 261: f=   988.512, eps=0.000000000
Epoch 262: f=   988.512, eps=0.000000000
Epoch 263: f=   988.512, eps=0.000000000
Epoch 264: f=   988.512, eps=0.000000000
Epoch 265: f=   988.512, eps=0.000000000
Epoch 266: f=   988.512, eps=0.000000000
Epoch 267: f=   988.512, eps=0.000000000
Epoch 268: f=   988.512, eps=0.000000000
Epoch 269: f=   988.512, eps=0.000000000
Epoch 270: f=   988.512, eps=0.000000000
```

```
Epoch 271: f=   988.512, eps=0.000000000
Epoch 272: f=   988.512, eps=0.000000000
Epoch 273: f=   988.512, eps=0.000000000
Epoch 274: f=   988.512, eps=0.000000000
Epoch 275: f=   988.512, eps=0.000000000
Epoch 276: f=   988.512, eps=0.000000000
Epoch 277: f=   988.512, eps=0.000000000
Epoch 278: f=   988.512, eps=0.000000000
Epoch 279: f=   988.512, eps=0.000000000
Epoch 280: f=   988.512, eps=0.000000000
Epoch 281: f=   988.512, eps=0.000000000
Epoch 282: f=   988.512, eps=0.000000000
Epoch 283: f=   988.512, eps=0.000000000
Epoch 284: f=   988.512, eps=0.000000000
Epoch 285: f=   988.512, eps=0.000000000
Epoch 286: f=   988.512, eps=0.000000000
Epoch 287: f=   988.512, eps=0.000000000
Epoch 288: f=   988.512, eps=0.000000000
Epoch 289: f=   988.512, eps=0.000000000
Epoch 290: f=   988.512, eps=0.000000000
Epoch 291: f=   988.512, eps=0.000000000
Epoch 292: f=   988.512, eps=0.000000000
Epoch 293: f=   988.512, eps=0.000000000
Epoch 294: f=   988.512, eps=0.000000000
Epoch 295: f=   988.512, eps=0.000000000
Epoch 296: f=   988.512, eps=0.000000000
Epoch 297: f=   988.512, eps=0.000000000
Epoch 298: f=   988.512, eps=0.000000000
Epoch 299: f=   988.512, eps=0.000000000
Epoch 300: f=   988.512, eps=0.000000000
Epoch 301: f=   988.512, eps=0.000000000
Epoch 302: f=   988.512, eps=0.000000000
Epoch 303: f=   988.512, eps=0.000000000
Epoch 304: f=   988.512, eps=0.000000000
Epoch 305: f=   988.512, eps=0.000000000
Epoch 306: f=   988.512, eps=0.000000000
Epoch 307: f=   988.512, eps=0.000000000
Epoch 308: f=   988.512, eps=0.000000000
Epoch 309: f=   988.512, eps=0.000000000
Epoch 310: f=   988.512, eps=0.000000000
Epoch 311: f=   988.512, eps=0.000000000
Epoch 312: f=   988.512, eps=0.000000000
Epoch 313: f=   988.512, eps=0.000000000
Epoch 314: f=   988.512, eps=0.000000000
Epoch 315: f=   988.512, eps=0.000000000
Epoch 316: f=   988.512, eps=0.000000000
Epoch 317: f=   988.512, eps=0.000000000
Epoch 318: f=   988.512, eps=0.000000000
```

```
Epoch 319: f=   988.512, eps=0.000000000
Epoch 320: f=   988.512, eps=0.000000000
Epoch 321: f=   988.512, eps=0.000000000
Epoch 322: f=   988.512, eps=0.000000000
Epoch 323: f=   988.512, eps=0.000000000
Epoch 324: f=   988.512, eps=0.000000000
Epoch 325: f=   988.512, eps=0.000000000
Epoch 326: f=   988.512, eps=0.000000000
Epoch 327: f=   988.512, eps=0.000000000
Epoch 328: f=   988.512, eps=0.000000000
Epoch 329: f=   988.512, eps=0.000000000
Epoch 330: f=   988.512, eps=0.000000000
Epoch 331: f=   988.512, eps=0.000000000
Epoch 332: f=   988.512, eps=0.000000000
Epoch 333: f=   988.512, eps=0.000000000
Epoch 334: f=   988.512, eps=0.000000000
Epoch 335: f=   988.512, eps=0.000000000
Epoch 336: f=   988.512, eps=0.000000000
Epoch 337: f=   988.512, eps=0.000000000
Epoch 338: f=   988.512, eps=0.000000000
Epoch 339: f=   988.512, eps=0.000000000
Epoch 340: f=   988.512, eps=0.000000000
Epoch 341: f=   988.512, eps=0.000000000
Epoch 342: f=   988.512, eps=0.000000000
Epoch 343: f=   988.512, eps=0.000000000
Epoch 344: f=   988.512, eps=0.000000000
Epoch 345: f=   988.512, eps=0.000000000
Epoch 346: f=   988.512, eps=0.000000000
Epoch 347: f=   988.512, eps=0.000000000
Epoch 348: f=   988.512, eps=0.000000000
Epoch 349: f=   988.512, eps=0.000000000
Epoch 350: f=   988.512, eps=0.000000000
Epoch 351: f=   988.512, eps=0.000000000
Epoch 352: f=   988.512, eps=0.000000000
Epoch 353: f=   988.512, eps=0.000000000
Epoch 354: f=   988.512, eps=0.000000000
Epoch 355: f=   988.512, eps=0.000000000
Epoch 356: f=   988.512, eps=0.000000000
Epoch 357: f=   988.512, eps=0.000000000
Epoch 358: f=   988.512, eps=0.000000000
Epoch 359: f=   988.512, eps=0.000000000
Epoch 360: f=   988.512, eps=0.000000000
Epoch 361: f=   988.512, eps=0.000000000
Epoch 362: f=   988.512, eps=0.000000000
Epoch 363: f=   988.512, eps=0.000000000
Epoch 364: f=   988.512, eps=0.000000000
Epoch 365: f=   988.512, eps=0.000000000
Epoch 366: f=   988.512, eps=0.000000000
```

```
Epoch 367: f=   988.512, eps=0.000000000
Epoch 368: f=   988.512, eps=0.000000000
Epoch 369: f=   988.512, eps=0.000000000
Epoch 370: f=   988.512, eps=0.000000000
Epoch 371: f=   988.512, eps=0.000000000
Epoch 372: f=   988.512, eps=0.000000000
Epoch 373: f=   988.512, eps=0.000000000
Epoch 374: f=   988.512, eps=0.000000000
Epoch 375: f=   988.512, eps=0.000000000
Epoch 376: f=   988.512, eps=0.000000000
Epoch 377: f=   988.512, eps=0.000000000
Epoch 378: f=   988.512, eps=0.000000000
Epoch 379: f=   988.512, eps=0.000000000
Epoch 380: f=   988.512, eps=0.000000000
Epoch 381: f=   988.512, eps=0.000000000
Epoch 382: f=   988.512, eps=0.000000000
Epoch 383: f=   988.512, eps=0.000000000
Epoch 384: f=   988.512, eps=0.000000000
Epoch 385: f=   988.512, eps=0.000000000
Epoch 386: f=   988.512, eps=0.000000000
Epoch 387: f=   988.512, eps=0.000000000
Epoch 388: f=   988.512, eps=0.000000000
Epoch 389: f=   988.512, eps=0.000000000
Epoch 390: f=   988.512, eps=0.000000000
Epoch 391: f=   988.512, eps=0.000000000
Epoch 392: f=   988.512, eps=0.000000000
Epoch 393: f=   988.512, eps=0.000000000
Epoch 394: f=   988.512, eps=0.000000000
Epoch 395: f=   988.512, eps=0.000000000
Epoch 396: f=   988.512, eps=0.000000000
Epoch 397: f=   988.512, eps=0.000000000
Epoch 398: f=   988.512, eps=0.000000000
Epoch 399: f=   988.512, eps=0.000000000
Epoch 400: f=   988.512, eps=0.000000000
Epoch 401: f=   988.512, eps=0.000000000
Epoch 402: f=   988.512, eps=0.000000000
Epoch 403: f=   988.512, eps=0.000000000
Epoch 404: f=   988.512, eps=0.000000000
Epoch 405: f=   988.512, eps=0.000000000
Epoch 406: f=   988.512, eps=0.000000000
Epoch 407: f=   988.512, eps=0.000000000
Epoch 408: f=   988.512, eps=0.000000000
Epoch 409: f=   988.512, eps=0.000000000
Epoch 410: f=   988.512, eps=0.000000000
Epoch 411: f=   988.512, eps=0.000000000
Epoch 412: f=   988.512, eps=0.000000000
Epoch 413: f=   988.512, eps=0.000000000
Epoch 414: f=   988.512, eps=0.000000000
```

```
Epoch 415: f=   988.512, eps=0.000000000
Epoch 416: f=   988.512, eps=0.000000000
Epoch 417: f=   988.512, eps=0.000000000
Epoch 418: f=   988.512, eps=0.000000000
Epoch 419: f=   988.512, eps=0.000000000
Epoch 420: f=   988.512, eps=0.000000000
Epoch 421: f=   988.512, eps=0.000000000
Epoch 422: f=   988.512, eps=0.000000000
Epoch 423: f=   988.512, eps=0.000000000
Epoch 424: f=   988.512, eps=0.000000000
Epoch 425: f=   988.512, eps=0.000000000
Epoch 426: f=   988.512, eps=0.000000000
Epoch 427: f=   988.512, eps=0.000000000
Epoch 428: f=   988.512, eps=0.000000000
Epoch 429: f=   988.512, eps=0.000000000
Epoch 430: f=   988.512, eps=0.000000000
Epoch 431: f=   988.512, eps=0.000000000
Epoch 432: f=   988.512, eps=0.000000000
Epoch 433: f=   988.512, eps=0.000000000
Epoch 434: f=   988.512, eps=0.000000000
Epoch 435: f=   988.512, eps=0.000000000
Epoch 436: f=   988.512, eps=0.000000000
Epoch 437: f=   988.512, eps=0.000000000
Epoch 438: f=   988.512, eps=0.000000000
Epoch 439: f=   988.512, eps=0.000000000
Epoch 440: f=   988.512, eps=0.000000000
Epoch 441: f=   988.512, eps=0.000000000
Epoch 442: f=   988.512, eps=0.000000000
Epoch 443: f=   988.512, eps=0.000000000
Epoch 444: f=   988.512, eps=0.000000000
Epoch 445: f=   988.512, eps=0.000000000
Epoch 446: f=   988.512, eps=0.000000000
Epoch 447: f=   988.512, eps=0.000000000
Epoch 448: f=   988.512, eps=0.000000000
Epoch 449: f=   988.512, eps=0.000000000
Epoch 450: f=   988.512, eps=0.000000000
Epoch 451: f=   988.512, eps=0.000000000
Epoch 452: f=   988.512, eps=0.000000000
Epoch 453: f=   988.512, eps=0.000000000
Epoch 454: f=   988.512, eps=0.000000000
Epoch 455: f=   988.512, eps=0.000000000
Epoch 456: f=   988.512, eps=0.000000000
Epoch 457: f=   988.512, eps=0.000000000
Epoch 458: f=   988.512, eps=0.000000000
Epoch 459: f=   988.512, eps=0.000000000
Epoch 460: f=   988.512, eps=0.000000000
Epoch 461: f=   988.512, eps=0.000000000
Epoch 462: f=   988.512, eps=0.000000000
```

```
Epoch 463: f=   988.512, eps=0.000000000
Epoch 464: f=   988.512, eps=0.000000000
Epoch 465: f=   988.512, eps=0.000000000
Epoch 466: f=   988.512, eps=0.000000000
Epoch 467: f=   988.512, eps=0.000000000
Epoch 468: f=   988.512, eps=0.000000000
Epoch 469: f=   988.512, eps=0.000000000
Epoch 470: f=   988.512, eps=0.000000000
Epoch 471: f=   988.512, eps=0.000000000
Epoch 472: f=   988.512, eps=0.000000000
Epoch 473: f=   988.512, eps=0.000000000
Epoch 474: f=   988.512, eps=0.000000000
Epoch 475: f=   988.512, eps=0.000000000
Epoch 476: f=   988.512, eps=0.000000000
Epoch 477: f=   988.512, eps=0.000000000
Epoch 478: f=   988.512, eps=0.000000000
Epoch 479: f=   988.512, eps=0.000000000
Epoch 480: f=   988.512, eps=0.000000000
Epoch 481: f=   988.512, eps=0.000000000
Epoch 482: f=   988.512, eps=0.000000000
Epoch 483: f=   988.512, eps=0.000000000
Epoch 484: f=   988.512, eps=0.000000000
Epoch 485: f=   988.512, eps=0.000000000
Epoch 486: f=   988.512, eps=0.000000000
Epoch 487: f=   988.512, eps=0.000000000
Epoch 488: f=   988.512, eps=0.000000000
Epoch 489: f=   988.512, eps=0.000000000
Epoch 490: f=   988.512, eps=0.000000000
Epoch 491: f=   988.512, eps=0.000000000
Epoch 492: f=   988.512, eps=0.000000000
Epoch 493: f=   988.512, eps=0.000000000
Epoch 494: f=   988.512, eps=0.000000000
Epoch 495: f=   988.512, eps=0.000000000
Epoch 496: f=   988.512, eps=0.000000000
Epoch 497: f=   988.512, eps=0.000000000
Epoch 498: f=   988.512, eps=0.000000000
Epoch 499: f=   988.512, eps=0.000000000
Result after 500 epochs: f=988.5118396027028
```

1.2 4b Effect of Prior

[5]:
```
# %% [markdown]
# ## 4b Effect of Prior

# %%
# YOUR CODE HERE
# Compare training/test log-likelihood and accuracy for different lambda values
```

```
# Parameters
lambdas = [0, 5, 10, 50, 100, 200, 500, 1000]
nepochs = 300
eps0_init = 1e-3
max_retries = 3
use_zero_init = True  # for fair comparisons

# Prepare scaled features (safe even if Xz already z-scored)
scaler = StandardScaler()
Xz_scaled = scaler.fit_transform(Xz)
Xtestz_scaled = scaler.transform(Xtestz)

# Storage
rows = []
results = []  # keep small result dicts for later use (4c can reuse if desired)

for lam in lambdas:
    print(f"\nRunning lambda = {lam}")
    if use_zero_init:
        w0 = np.zeros(Xz_scaled.shape[1])
    else:
        w0 = np.random.normal(size=Xz_scaled.shape[1])

    eps_try = eps0_init
    success = False
    for attempt in range(max_retries):
        print(f" attempt {attempt+1} with eps0={eps_try:.1e} ... ", end="",␣
 ↪flush=True)
        try:
            w_map, vals_map, eps_map = optimize(gd_l2(y, Xz_scaled, lam), w0,␣
 ↪nepochs=nepochs, eps0=eps_try, verbose=False)
        except Exception as e:
            print(f"exception: {e}. reduce eps and retry.")
            eps_try *= 0.1
            continue

        if np.any(np.isnan(vals_map)) or np.any(np.isnan(w_map)):
            print("FAILED (nan). Reducing eps and retrying.")
            eps_try *= 0.1
            continue

        print("OK")
        success = True
        break

    if not success:
        print(f" All retries failed for lambda={lam}. Marking as failed.")
```

```
        rows.append({"lambda": lam, "train_ll": np.nan, "test_ll": np.nan,␣
↪"train_acc": np.nan, "test_acc": np.nan})
        results.append({"lambda": lam, "w": None, "objective": None, "eps":␣
↪None})
        continue

    # compute plain data log-likelihoods (no prior) and accuracies
    tr_ll = l(y, Xz_scaled, w_map)
    te_ll = l(ytest, Xtestz_scaled, w_map)
    ypred_tr = classify(Xz_scaled, w_map)
    ypred_te = classify(Xtestz_scaled, w_map)
    tr_acc = np.mean(ypred_tr == y)
    te_acc = np.mean(ypred_te == ytest)

    rows.append({"lambda": lam, "train_ll": tr_ll, "test_ll": te_ll,␣
↪"train_acc": tr_acc, "test_acc": te_acc})
    results.append({"lambda": lam, "w": w_map, "objective": vals_map, "eps":␣
↪eps_map})
```

```
Running lambda = 0
 attempt 1 with eps0=1.0e-03 … OK

Running lambda = 5
 attempt 1 with eps0=1.0e-03 … OK

Running lambda = 10
 attempt 1 with eps0=1.0e-03 … OK

Running lambda = 50
 attempt 1 with eps0=1.0e-03 … OK

Running lambda = 100
 attempt 1 with eps0=1.0e-03 … OK

Running lambda = 200
 attempt 1 with eps0=1.0e-03 … OK

Running lambda = 500
 attempt 1 with eps0=1.0e-03 … OK

Running lambda = 1000
 attempt 1 with eps0=1.0e-03 … OK
```

[6]:
```
# Summary DataFrame
df = pd.DataFrame(rows).set_index("lambda")
pd.set_option("display.float_format", "{: .4f}".format)
```

```
print("\nSummary (data log-likelihood and accuracies):")
display(df)
```

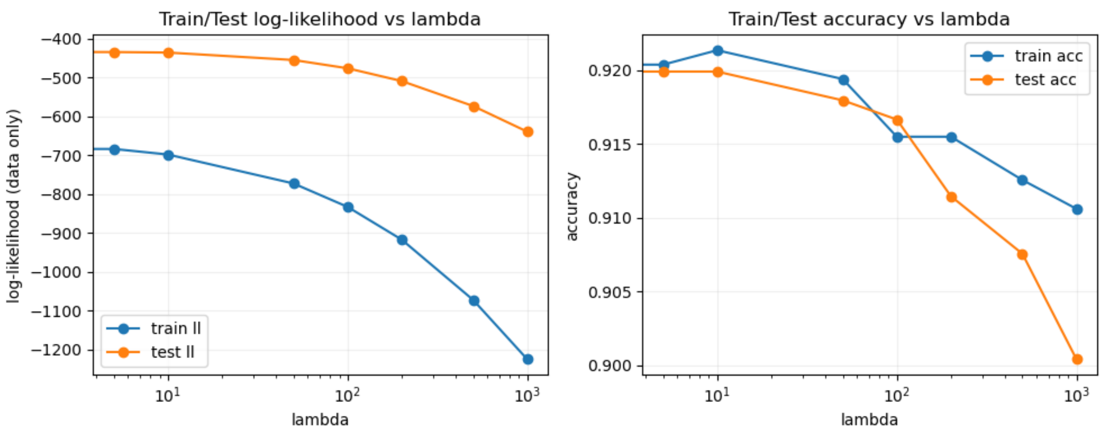
Summary (data log-likelihood and accuracies):

```
          train_ll   test_ll  train_acc  test_acc
lambda
0        -661.5186 -430.3692     0.9214    0.9173
5        -684.1570 -435.0073     0.9204    0.9199
10       -698.4214 -436.3227     0.9214    0.9199
50       -772.8450 -455.4549     0.9194    0.9180
100      -833.0105 -476.7408     0.9155    0.9167
200      -917.1361 -509.5574     0.9155    0.9115
500     -1072.2064 -573.8156     0.9126    0.9076
1000    -1224.7230 -639.6407     0.9106    0.9004
```

[7]:

```
# Plot plain train/test log-likelihood (no prior)
plt.figure(figsize=(10,4))
plt.subplot(1,2,1)
plt.plot(df.index, df["train_ll"], marker='o', label="train ll")
plt.plot(df.index, df["test_ll"], marker='o', label="test ll")
plt.xscale("log")
plt.xlabel("lambda")
plt.ylabel("log-likelihood (data only)")
plt.title("Train/Test log-likelihood vs lambda")
plt.legend()
plt.grid(alpha=0.2)

# Plot accuracies
plt.subplot(1,2,2)
plt.plot(df.index, df["train_acc"], marker='o', label="train acc")
plt.plot(df.index, df["test_acc"], marker='o', label="test acc")
plt.xscale("log")
plt.xlabel("lambda")
plt.ylabel("accuracy")
plt.title("Train/Test accuracy vs lambda")
plt.legend()
plt.grid(alpha=0.2)
plt.tight_layout()
plt.show()
```



1.3 4c Composition of Weight Vector

[8]:
```python
# %%
# YOUR CODE HERE
# PARAMETERS
try:
    results  # noqa: F821
except NameError:
    # Recompute (same safe logic as in 4b but shorter)
    results = []
    scaler = StandardScaler()
    Xz_scaled = scaler.fit_transform(Xz)
    for lam in lambdas:
        w0 = np.zeros(Xz_scaled.shape[1])
        eps_try = eps0_init
        success = False
        for attempt in range(max_retries):
            try:
                w_map, vals_map, eps_map = optimize(gd_l2(y, Xz_scaled, lam),␣
↪w0, nepochs=nepochs, eps0=eps_try, verbose=False)
            except Exception:
                eps_try *= 0.1
                continue
            if np.any(np.isnan(w_map)) or np.any(np.isnan(vals_map)):
                eps_try *= 0.1
                continue
            success = True
            break
        if not success:
            results.append({"lambda": lam, "w": None, "objective": None, "eps":␣
↪None})
```

```python
        else:
            results.append({"lambda": lam, "w": w_map, "objective": vals_map, "eps": eps_map})
```

[9]:
```python
# Parameters for display
k = 12  # top-k per lambda to consider
union_idx = set()
for r in results:
    if r["w"] is None:
        continue
    order = np.argsort(np.abs(r["w"]))[::-1][:k]
    union_idx.update(order)
union_idx = sorted(list(union_idx))
if len(union_idx) == 0:
    raise RuntimeError("No successful runs to inspect. Try lowering eps0 or lambdas list.")

feat_names = [features[i] for i in union_idx]

# Build DataFrame (rows = features, cols = lambdas)
df_plot = pd.DataFrame(index=feat_names, columns=[f" ={r['lambda']}" for r in results], dtype=float)
for r in results:
    lam = r["lambda"]
    col = f" ={lam}"
    if r["w"] is None:
        df_plot[col] = np.nan
    else:
        df_plot[col] = [r["w"][i] for i in union_idx]
```

[10]:
```python
pd.set_option("display.float_format", "{: .4f}".format)
print("\nTop features (union across lambdas) - numeric table:")
display(df_plot)
```

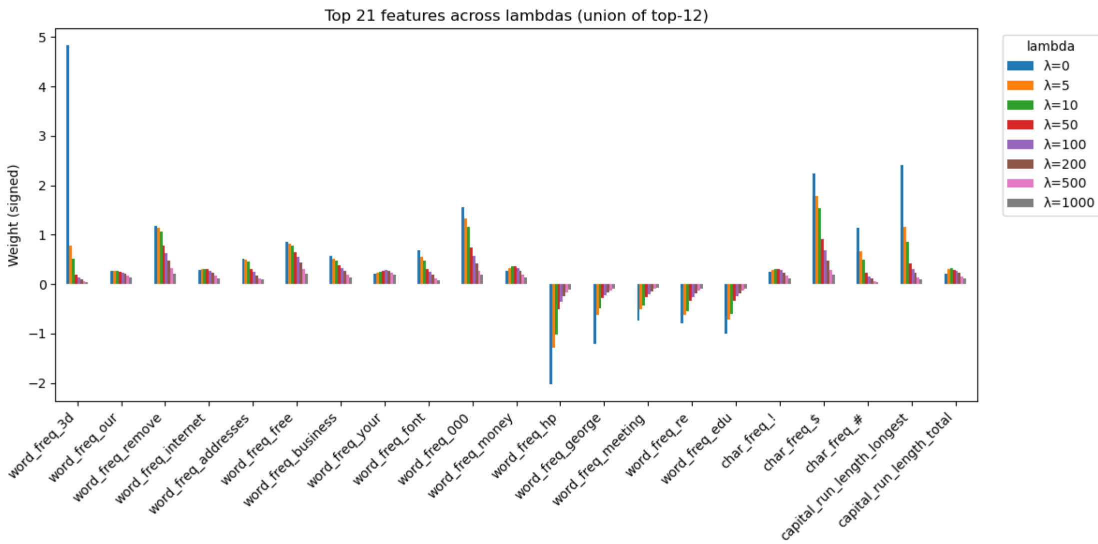
Top features (union across lambdas) - numeric table:

	=0	=5	=10	=50	=100	=200	\
word_freq_3d	4.8299	0.7814	0.5049	0.1973	0.1359	0.0967	
word_freq_our	0.2595	0.2624	0.2633	0.2531	0.2365	0.2104	
word_freq_remove	1.1758	1.1353	1.0718	0.7876	0.6324	0.4828	
word_freq_internet	0.2912	0.3080	0.3124	0.2985	0.2723	0.2316	
word_freq_addresses	0.5172	0.4930	0.4533	0.3084	0.2404	0.1803	
word_freq_free	0.8470	0.8176	0.7870	0.6436	0.5482	0.4409	
word_freq_business	0.5697	0.5135	0.4836	0.3869	0.3319	0.2706	
word_freq_your	0.2156	0.2328	0.2430	0.2715	0.2781	0.2726	
word_freq_font	0.6824	0.5458	0.4756	0.3066	0.2426	0.1854	
word_freq_000	1.5633	1.3370	1.1663	0.7343	0.5680	0.4238	

```
word_freq_money              0.2620  0.3319  0.3524  0.3577  0.3202  0.2634
word_freq_hp                -2.0325 -1.2908 -1.0322 -0.5080 -0.3577 -0.2525
word_freq_george            -1.2220 -0.6153 -0.4969 -0.2829 -0.2201 -0.1724
word_freq_meeting           -0.7349 -0.5166 -0.4399 -0.2604 -0.1981 -0.1489
word_freq_re                -0.7933 -0.6224 -0.5417 -0.3357 -0.2569 -0.1910
word_freq_edu               -1.0132 -0.7159 -0.6043 -0.3417 -0.2540 -0.1873
char_freq_!                  0.2404  0.2812  0.2992  0.3081  0.2815  0.2376
char_freq_$                  2.2336  1.7888  1.5457  0.9132  0.6759  0.4820
char_freq_#                  1.1374  0.6678  0.5022  0.2285  0.1599  0.1105
capital_run_length_longest   2.4013  1.1502  0.8529  0.4161  0.3039  0.2196
capital_run_length_total     0.2095  0.3037  0.3163  0.2939  0.2630  0.2221

                              =500   =1000
word_freq_3d                 0.0613  0.0418
word_freq_our                0.1640  0.1248
word_freq_remove             0.3150  0.2166
word_freq_internet           0.1657  0.1183
word_freq_addresses          0.1198  0.0871
word_freq_free               0.3018  0.2119
word_freq_business           0.1902  0.1375
word_freq_your               0.2364  0.1903
word_freq_font               0.1208  0.0814
word_freq_000                0.2728  0.1884
word_freq_money              0.1830  0.1305
word_freq_hp                -0.1625 -0.1168
word_freq_george            -0.1243 -0.0939
word_freq_meeting           -0.0998 -0.0716
word_freq_re                -0.1247 -0.0872
word_freq_edu               -0.1240 -0.0884
char_freq_!                  0.1695  0.1219
char_freq_$                  0.2943  0.1963
char_freq_#                  0.0663  0.0442
capital_run_length_longest   0.1391  0.0961
capital_run_length_total     0.1614  0.1184
```

[11]:
```python
# Plot grouped bar chart for the union-top features
ax = df_plot.plot(kind="bar", figsize=(12, 6))
ax.set_ylabel("Weight (signed)")
ax.set_title(f"Top {len(feat_names)} features across lambdas (union of␣
 ↪top-{k})")
plt.xticks(rotation=45, ha="right")
plt.legend(title="lambda", bbox_to_anchor=(1.02, 1), loc="upper left")
plt.tight_layout()
plt.show()
```



[12]:
```python
# print textual top-10 for smallest and largest successful lambda
successful = [r for r in results if r["w"] is not None]
if successful:
    lam_small = successful[0]["lambda"]
    lam_large = successful[-1]["lambda"]
    w_small = successful[0]["w"]
    w_large = successful[-1]["w"]
    print(f"\nTop 10 features for lambda={lam_small}:")
    for idx in np.argsort(np.abs(w_small))[::-1][:10]:
        print(f"  {features[idx]:30s} {w_small[idx]: .4f}")
    print(f"\nTop 10 features for lambda={lam_large}:")
    for idx in np.argsort(np.abs(w_large))[::-1][:10]:
        print(f"  {features[idx]:30s} {w_large[idx]: .4f}")
```

```
Top 10 features for lambda=0:
  word_freq_3d                    4.8299
  capital_run_length_longest      2.4013
  char_freq_$                     2.2336
  word_freq_hp                   -2.0325
  word_freq_000                   1.5633
  word_freq_george               -1.2220
  word_freq_remove                1.1758
  char_freq_#                     1.1374
  word_freq_edu                  -1.0132
  word_freq_free                  0.8470

Top 10 features for lambda=1000:
  word_freq_remove                0.2166
```

word_freq_free	0.2119
char_freq_$	0.1963
word_freq_your	0.1903
word_freq_000	0.1884
word_freq_business	0.1375
word_freq_money	0.1305
word_freq_our	0.1248
char_freq_!	0.1219
capital_run_length_total	0.1184