Part-1 Python basics print out: (T test and Linear model)

```
T- test:
Gauss [2.492625648511938, 6.526310224358953, 8.152245357678925, 10.894615049025047, 6.462624794056173,
 8.888520932424589, 6.738073279807356, 1.7558954027775422, 4.348655129521663, -0.745378369792268,
  6.466536091935655, 4.860273668804933, 9.182573552357514, 11.075987876886748, -0.7089836170915405,
   8.697064371533129, -3.152829935754405, -6.097258008778711, 0.32668267094167436, -3.908010093703446,
    -1.75333382182081017, -5.266629568354527, -5.829156283217352, 13.776925845123232, 4.876548686575962]

mean 3.5224229796271436
var 32.07767545290616
T 1.344014302718903
pval 0.8444436162934872

Linear model:
E noise is random normal, shape:  (100, 1)
X is random uniform with 1s stacked, shape:  (100, 4)
B [[1]
 [2]
 [3]
 [4]]
Y = XB+E, shape:  (100, 1)
Bhat = (XtX)^-1 XtY
Bhat =  [[1.31816727]
 [1.94712275]
 [3.31835405]
 [3.75955769]]
```

Part 2 k=2 logistic derivation turned in on paper

Part 3 k=3 logistic regression with MNIST data printout:

```
arr0 shape  (300, 28, 28) arr1 shape  (300,) arr2 shape  (3, 28, 28)
create custom test image which is mirrored version of arr_2[2] which is an 8
..displaying image
Define Y categories: 1=[1 0 0] 4=[0 1 0] 8=[0 0 1]
Y shape:  (300, 3)
flatted 28x28 x matrices to 1D of length 768 + constant 1
X shape:  (300, 785)
initialize B with uniform values 0.1
B Shape:  (785, 3)
begin descent with parameters: alpha=  0.002  with stopping condition dB > 0.001
define dB such that  dB = max| B[i,j] - Bnew[i,j] |
2 iteration dB =  0.09263893260066962
3 iteration dB =  0.12847462384600614
4 iteration dB =  0.17213797025021949
5 iteration dB =  0.266587752683672515
6 iteration dB =  0.1743652228219916
7 iteration dB =  0.06777251138304888
8 iteration dB =  0.019961923059446995
9 iteration dB =  0.008310829092577465
10 iteration dB =  0.0072197397508926566
11 iteration dB =  0.006661097527581198
12 iteration dB =  0.00615478219811294
13 iteration dB =  0.005756657723360037
14 iteration dB =  0.005408004098022751
15 iteration dB =  0.0051009039470377715
16 iteration dB =  0.0048285020082924244
17 iteration dB =  0.0045850615532614936
18 iteration dB =  0.004365890072616979
19 iteration dB =  0.0041671709428277772
20 iteration dB =  0.00398581564180607
21 iteration dB =  0.0038193208473577267
22 iteration dB =  0.003665649934493323
23 iteration dB =  0.003523137023366457
24 iteration dB =  0.0033904108164321545
25 iteration dB =  0.0032760814343914535
26 iteration dB =  0.00316965902768937
27 iteration dB =  0.003069225883546217
28 iteration dB =  0.0029756971511954444
29 iteration dB =  0.0029249122425786256
30 iteration dB =  0.0028767114874010957
31 iteration dB =  0.0028830759146785322
32 iteration dB =  0.00278677040202413
33 iteration dB =  0.002744503589002134
34 iteration dB =  0.0027037538646602632
35 iteration dB =  0.002664347972465564
```

```
36 iteration dB =   0.0026261398529470736
37 iteration dB =   0.0025890069073227218
38 iteration dB =   0.0025528467709592384
39 iteration dB =   0.0025175744491607666
40 iteration dB =   0.0024831200036980091
41 iteration dB =   0.0024494260787719563
42 iteration dB =   0.0024164460163247436
43 iteration dB =   0.0023841422161423598
44 iteration dB =   0.002352484448407179
45 iteration dB =   0.0023214485070272572
46 iteration dB =   0.002291015008306252
47 iteration dB =   0.0022611683465566708
48 iteration dB =   0.0022318958076548157
49 iteration dB =   0.0022031868220005765
50 iteration dB =   0.0021750323469278965
51 iteration dB =   0.002147424365121331
52 iteration dB =   0.0021203554858950646
53 iteration dB =   0.00209381863586805
54 iteration dB =   0.0020678066826269669
55 iteration dB =   0.0020423129846770527
56 iteration dB =   0.0020173298399719375
57 iteration dB =   0.0019928498504023784
58 iteration dB =   0.0019688651658111844
59 iteration dB =   0.0019453676162878852
60 iteration dB =   0.0019223487206594392
61 iteration dB =   0.0018997997093198604
62 iteration dB =   0.0018777115568907599
63 iteration dB =   0.0018560750210847021
64 iteration dB =   0.001834880684910556
65 iteration dB =   0.0018141190000176555
66 iteration dB =   0.0017937803295235422
67 iteration dB =   0.0017738549891275768
68 iteration dB =   0.0017543332856786442
69 iteration dB =   0.001735205552661434
70 iteration dB =   0.0017164621822951531
71 iteration dB =   0.0016980936541161062
72 iteration dB =   0.0016800905600463656
73 iteration dB =   0.001662443626049781
74 iteration dB =   0.00164514373054081
75 iteration dB =   0.0016281819197580538
76 iteration dB =   0.0016115494203398084
77 iteration dB =   0.0015952376493522102
78 iteration dB =   0.0015792382220219392
79 iteration dB =   0.0015635429574222448
80 iteration dB =   0.0015481438823510274
81 iteration dB =   0.0015330332336203623
82 iteration dB =   0.0015182034589741877
83 iteration dB =   0.0015036472168172277
84 iteration dB =   0.001489357374935052
85 iteration dB =   0.001475327008359839
86 iteration dB =   0.0014615493965239867
87 iteration dB =   0.0014480180198260806
88 iteration dB =   0.0014347265557196898
89 iteration dB =   0.0014216688744227435
90 iteration dB =   0.0014088390343319213
91 iteration dB =   0.0013962312772151653
92 iteration dB =   0.0013838400232467074
93 iteration dB =   0.0013716598865938124
94 iteration dB =   0.001359685567012603
95 iteration dB =   0.001347912051260891
96 iteration dB =   0.0013363344014126755
97 iteration dB =   0.0013249478530502068
98 iteration dB =   0.0013137477895872052
99 iteration dB =   0.0013027297373314761
100 iteration dB =   0.0012918893606456705
101 iteration dB =   0.001281222457218567
102 iteration dB =   0.00127072495345526
103 iteration dB =   0.0012603928999933567
104 iteration dB =   0.00125022246734946
105 iteration dB =   0.0012402099416996548
106 iteration dB =   0.0012303517207953307
107 iteration dB =   0.0012206443100145625
108 iteration dB =   0.0012110843185493825
109 iteration dB =   0.0012016684557268897
110 iteration dB =   0.0011923935274627454
111 iteration dB =   0.0011832564432844837
112 iteration dB =   0.0011742541608431578
113 iteration dB =   0.0011653837871444805
114 iteration dB =   0.0011566424711070833
```

```
115 iteration dB =  0.00114802745283199
116 iteration dB =  0.001139536050348
117 iteration dB =  0.0011311656569058526
118 iteration dB =  0.0011229137383776955
119 iteration dB =  0.0011147778307587508
120 iteration dB =  0.0011067555377667904
121 iteration dB =  0.0010988445285356474
122 iteration dB =  0.001091042535399711
123 iteration dB =  0.0010833473517650183
124 iteration dB =  0.0010757568300640008
125 iteration dB =  0.0010682688797902795
126 iteration dB =  0.0010608814656102306
127 iteration dB =  0.0010535926055479372
128 iteration dB =  0.001046400369240863
129 iteration dB =  0.0010393028762630818
130 iteration dB =  0.00103229829451279
131 iteration dB =  0.0010253848386619913
132 iteration dB =  0.0010185607686654663
133 iteration dB =  0.0010118243883265876
134 iteration dB =  0.0010051740439168677
135 iteration dB =  0.000998608122848299
Model trained, now testing with arr2
arr 2 contents [ [1],[4],[8], [8] ], last 8 is custom image appended to test array
P for Xtest and B
[[9.83775816e-01 1.15201394e-04 1.60949129e-02]
 [2.97847121e-04 9.95633578e-01 4.06347098e-03]
 [2.12117527e-02 2.72053343e-04 9.78515791e-01]
 [9.00379854e-02 2.20605905e-03 9.07754679e-01]]
Model prediction for test[i], with P > 0.5
test digit i= 0  classified as a 1
test digit i= 1  classified as a 4
test digit i= 2  classified as a 8
test digit i= 3  classified as a 8
```