Assignment 2（Writeup）

Created By:

Group: Chengyin Liu(cl93), Ran Jin(Oliver)(rj23)

Jan 28, 2018
COMP 540 – Statistical Machine Learning
Rice University

Problem 1)

a) Let $g(z) = \frac{1}{1+e^{-z}}$, want to show that $\frac{d}{dz}\, g(z) = g(z)(1-g(z))$

We write

$$\frac{1}{1+e^{-z}} = (1\ + e^{-z})^{-1}$$

Then, we have

$$\frac{d}{dz}\ [(1\ + e^{-z})^{-1}] = -\frac{\frac{d}{dz}(e^{-z}+1)}{(e^{-z}+1)^2} = -\frac{\frac{d}{dz}(e^{-z}) + \frac{d}{dz}(1)}{(e^{-z}+1)^2} = -\frac{-e^{-z}\frac{d}{dz}(z)}{(e^{-z}+1)^2}$$

$$= \frac{e^{-z}}{(e^{-z}+1)^2} = g(z)(1-g(z))$$

Hence the proof.

b) Derive the NLL for the logistic regression

First of all, $h_\theta(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$

Then the negative log-likelihood would be

$$NLL(\theta) = -\log\left(\prod_{i=1}^{m} P\left(y^{(i)}\middle|X^{(i)}\right)\right) = -\log\left(\prod_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)\right)^{y^{(i)}}\left(1-h_\theta\left(x^{(i)}\right)\right)^{1-y^{(i)}}\right)$$

$$= -(\sum_{i=1}^{m} y^{(i)} log\left(h_\theta\left(x^{(i)}\right)\right) + (1-y^{(i)}) log\left(1-h_\theta\left(x^{(i)}\right)\right))$$

Finally, by taking the derivative of NLL, we get

$$\frac{\partial}{\partial(\theta)} NLL(\theta) = -(\sum_{i=1}^{m}(y^{(i)}\frac{g(\theta^T x^{(i)})'}{g(\theta^T x^{(i)})} + (1-y^{(i)})\frac{(1-g(\theta^T x^{(i)}))'}{1-g(\theta^T x^{(i)})})\frac{\partial\left(\theta^T x^{(i)}\right)}{\partial(\theta)})$$

$$= -(\sum_{i=1}^{m}(y^{(i)}\frac{g\left(\theta^T x^{(i)}\right)\left(1-g\left(\theta^T x^{(i)}\right)\right)}{g(\theta^T x^{(i)})}$$

$$+ \left(1-y^{(i)}\right)\frac{g\left(\theta^T x^{(i)}\right)\left(g\left(\theta^T x^{(i)}\right)-1\right)}{\left(1-g(\theta^T x^{(i)})\right)})x^{(i)})$$

$$= \sum_{i=1}^{m}\left(-y^{(i)} + y^{(i)}g(\theta^T x^{(i)}) - y^{(i)}g(\theta^T x^{(i)}) + g(\theta^T x^{(i)})\right)x^{(i)}$$

$$= \sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x^{(i)}$$

Hence the proof.

c) Show that H is positive definite.

A matrix is positive definite if its determinant is positive. So, we will determine that H has a positive determinant.

First of all, we know that X is full rank which means the determinant of X has to be nonzero. And also $\det(X) = \det(X^T)$. Thus, $\det(X) * \det(X^T)$ has to be positive. Now, we only need to show that if S has a positive dominant. Then, H has a positive determinant as well which implies that H is positive definite.

S is diagonal matrix with all diagonal elements greater than 0 and less than 1. Thus, the determinant of S is also positive. Thus, the dominant of H is positive and we conclude that H is positive definite.

Problem 2)

a) J(θ) has multiple locally optimal solutions.

False. Asλ >0, there only exists one global optimal for J(θ).

b) Let $\theta^* = argmin_\theta J(\theta)$ be a global optimum. θ^* is sparse. (has many zero entries)

False. We can construct a linearly separable data set with certain set of features such that all of the features would be required to separate the classes.

c) If the training data is linearly separable, then some coefficients θ_j might become infinite if λ = 0.

True. If the training data is linearly separable, then $c * \theta^*$, where c is a positive constant, is always going to increase the likelihood.

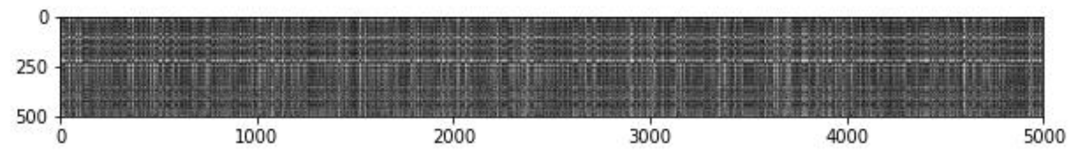
d) The first term of J(θ) always increases as we increase λ.

False. As λ increases, the likelihood on the test data will increase till a certain point that when $\theta^* = 0$, then it would typically not be a good predictor.

Problem 3)

a) Creating a knn classifier

Visualize the distance matrix: each row is a single test example and its distances to training examples.



Question: Notice the structured patterns in the distance matrix, where some rows or columns are visible brighter. (Note that with the default color scheme black indicates low distances while white indicates high distances.)

What in the data is the cause behind the distinctly bright rows?

What causes the columns?

Answer:

1. The test samples corresponding to these distinctly bright rows are very different from most of the training samples, which means they may be relatively more identifiable.

2. The training samples corresponding to these bright columns are very different from most of the test samples, which means these training samples have a high degree of differentiation between classes because they may be only close to the samples within its own class on the feature domain.

b) Distance matrix computation with two loops

```
Training data shape:    (50000L, 32L, 32L, 3L)
Training labels shape:    (50000L,)
Test data shape:    (10000L, 32L, 32L, 3L)
Test labels shape:    (10000L,)
y_train[1]    9
```

c) Compute majority label

When k=1, the accuracy is:

```
Got  137  /  500  correct  =>  accuracy:  0.274000
```

When k=5, the accuracy is:

```
Got  139  /  500  correct  =>  accuracy:  0.278000
```

d) Distance matrix computation with one loop

```
Difference was: 0.000000
Good! The distance matrices are the same
```

e) Distance matrix computation with no loops

```
Difference was: 0.000000
Good! The distance matrices are the same
```

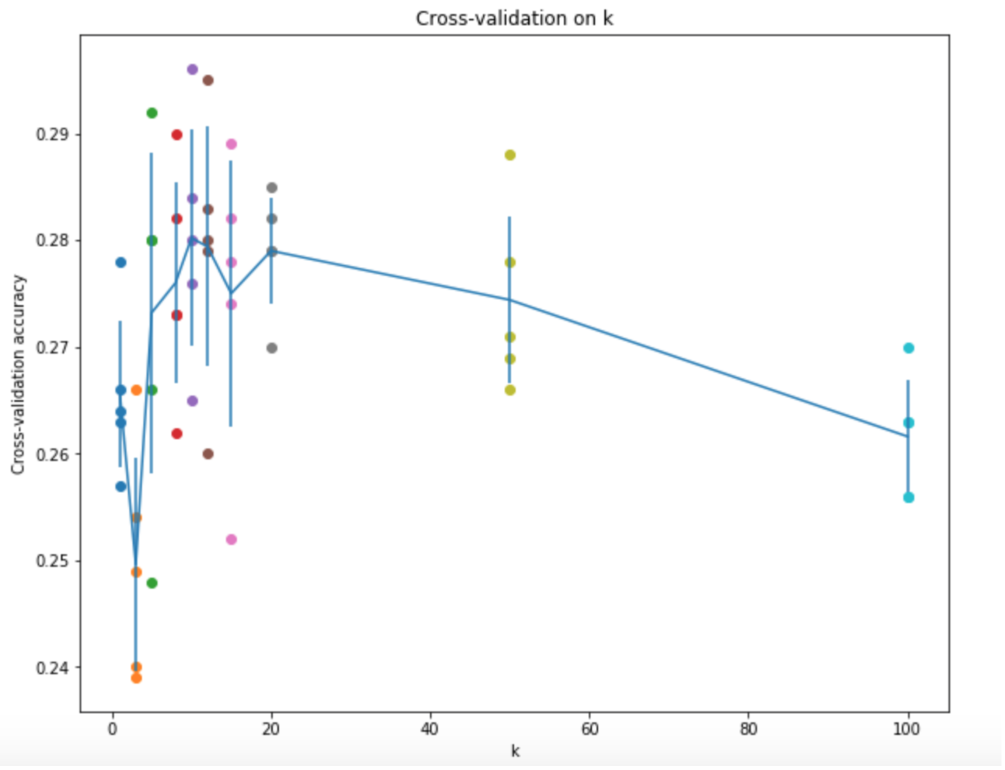
```
Two loop version took 42.182000 seconds
One loop version took 78.941000 seconds
No loop version took 0.656000 seconds
```

Thus, No loop is significantly faster.

f) Choosing k by cross validation

```
k = 1, accuracy = 0.263000
k = 1, accuracy = 0.257000
k = 1, accuracy = 0.264000
k = 1, accuracy = 0.278000
k = 1, accuracy = 0.266000
k = 3, accuracy = 0.239000
k = 3, accuracy = 0.249000
k = 3, accuracy = 0.240000
k = 3, accuracy = 0.266000
k = 3, accuracy = 0.254000
k = 5, accuracy = 0.248000
k = 5, accuracy = 0.266000
k = 5, accuracy = 0.280000
k = 5, accuracy = 0.292000
k = 5, accuracy = 0.280000
k = 8, accuracy = 0.262000
k = 8, accuracy = 0.282000
k = 8, accuracy = 0.273000
k = 8, accuracy = 0.290000
k = 8, accuracy = 0.273000
k = 10, accuracy = 0.265000
k = 10, accuracy = 0.296000
k = 10, accuracy = 0.276000
k = 10, accuracy = 0.284000
k = 10, accuracy = 0.280000
k = 12, accuracy = 0.260000
k = 12, accuracy = 0.295000
k = 12, accuracy = 0.279000
k = 12, accuracy = 0.283000
k = 12, accuracy = 0.280000
k = 15, accuracy = 0.252000
```

```
k = 15, accuracy = 0.289000
k = 15, accuracy = 0.278000
k = 15, accuracy = 0.282000
k = 15, accuracy = 0.274000
k = 20, accuracy = 0.270000
k = 20, accuracy = 0.279000
k = 20, accuracy = 0.279000
k = 20, accuracy = 0.282000
k = 20, accuracy = 0.285000
k = 50, accuracy = 0.271000
k = 50, accuracy = 0.288000
k = 50, accuracy = 0.278000
k = 50, accuracy = 0.269000
k = 50, accuracy = 0.266000
k = 100, accuracy = 0.256000
k = 100, accuracy = 0.270000
k = 100, accuracy = 0.263000
k = 100, accuracy = 0.256000
k = 100, accuracy = 0.263000
```



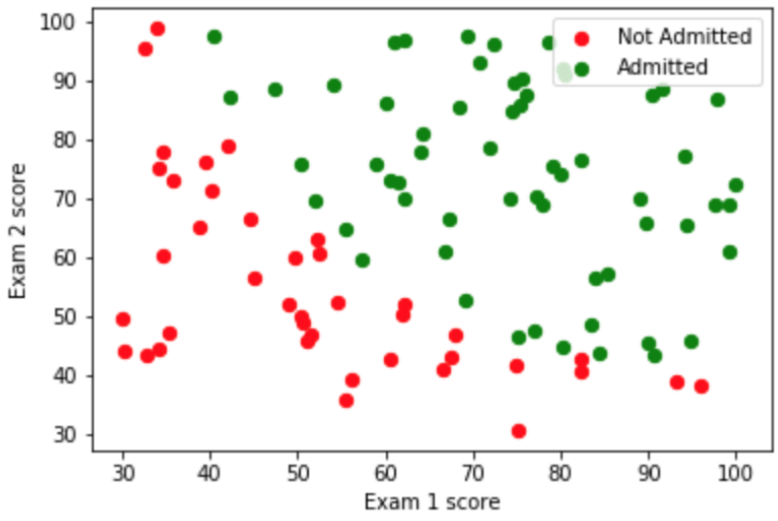
Thus, we picked the best k by cross validation that k=10.
And the computed accuracy is:

```
Got 141 / 500 correct => accuracy: 0.282000
```

Problem 4)

Part A: Logistic regression

a) Implementing logistic regression: the sigmoid function



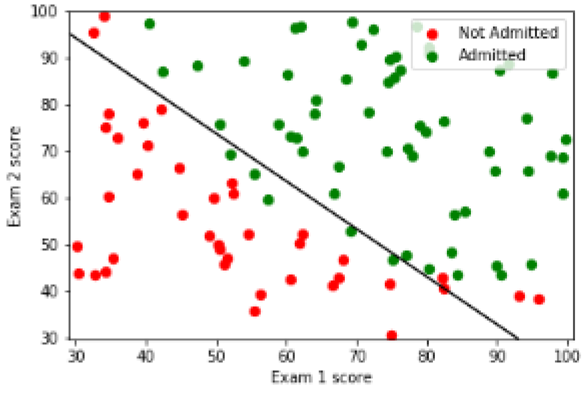
b) Cost function and gradient of logistic regression

```
Loss on all-zeros theta vector (should be around 0.693)
= 0.69314718056
Gradient of loss wrt all-zeros theta vector (should be around [-0.1, -12.01, -11.26])
= [ -0.1,  -12.00921659, -11.26284221]
Optimization terminated successfully.
         Current function value: 0.203498
         Iterations: 19
         Function evaluations: 20
         Gradient evaluations: 20
Theta found by fmin_bfgs:  [-25.16056945   0.20622963   0.20146073]
Final loss =  0.203497702351
```

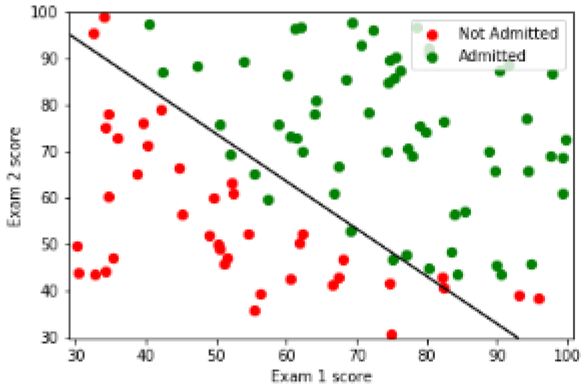
c) Prediction using a logistic regression model

```
For a student with 45 on exam 1 and 85 on exam 2, the probability of admission = 0.
776246678481
[0 0 0 1 1 0 1 0 1 1 1 0 1 1 0 1 0 0 1 1 0 1 0 0 1 1 1 1 0 0 1 1 0 0 0 0 1  1
0 0 1 0 1 1 0 0 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 0 0 1 0 1 1 0 1 1 1  1 1 1
 1 0 1 1 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1]
Accuracy on the training set =  0.89
```

d) Visualizing the decision boundary



```
Theta found by sklearn:  [[-25.15293066   0.20616459   0.20140349]]
```



Part B: Regularized logistic regression

a) Cost function and gradient for regularized logistic regression and plotting the decision boundary

Optimization terminated successfully.

Current function value: 0.529003

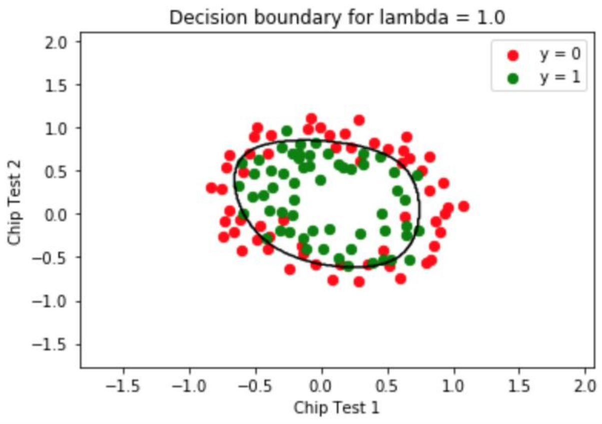
Iterations: 47

Function evaluations: 48

Gradient evaluations: 48

Theta found by fmin_bfgs: [1.27268739 0.62557016 1.1809665 -2.01919822 -0.91761468 -1.43194199 0.12375921 -0.36513086 -0.35703388 -0.17485805 -1.45843772 -0.05129676 -0.61603963 -0.2746414 -1.19282569 -0.24270336 -0.20570022 -0.04499768 -0.27782709 -0.29525851 -0.45613294 -1.04377851 0.02762813 -0.29265642 0.01543393 -0.32759318 -0.14389199 -0.92460119]

Final loss = 0.4624583499

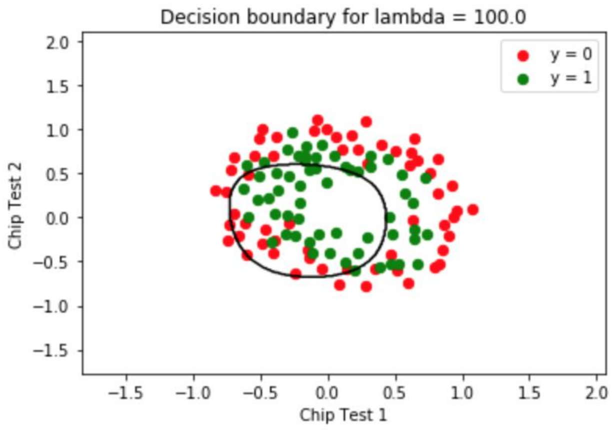


b) Prediction using the model

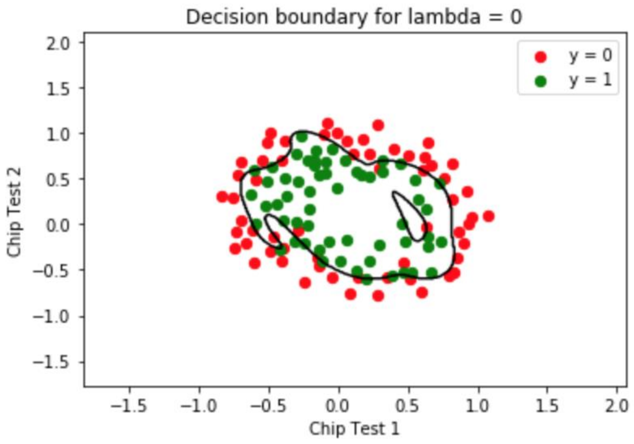
Accuracy on the training set = 0.830508474576

c) Varying λ

The under-fitting plot

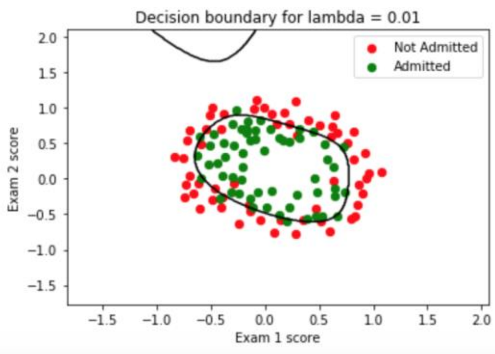


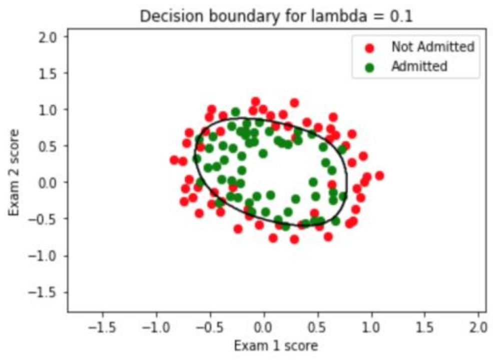
The over-fitting plot

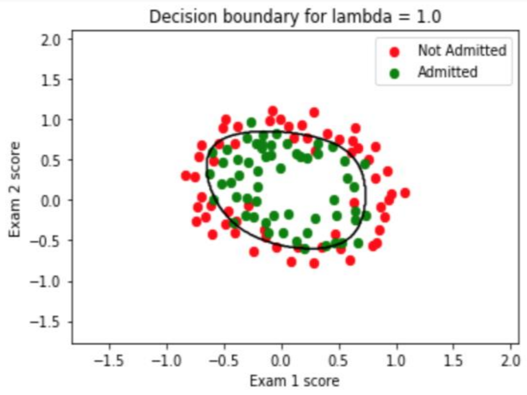


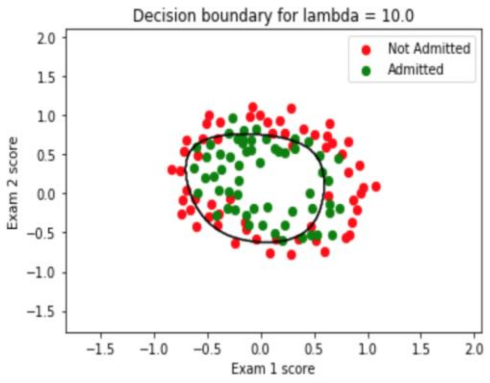
d) Exploring L1 and L2 penalized logistic regression

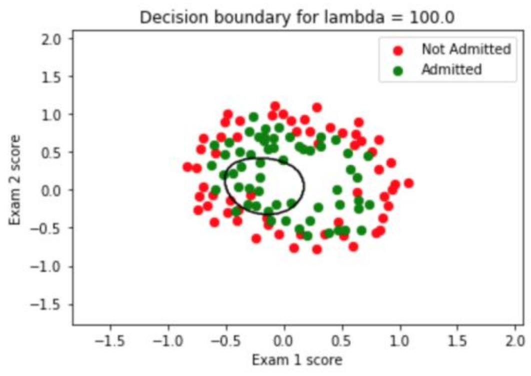
For L2,



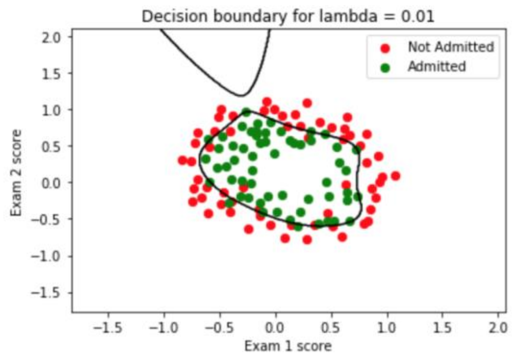


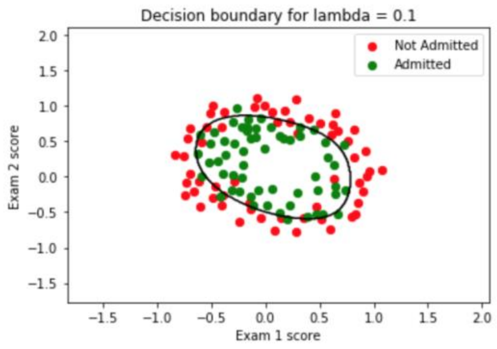


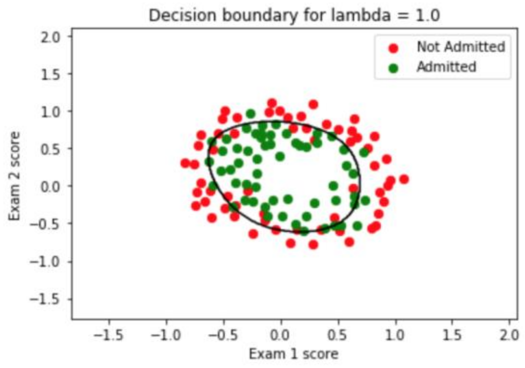


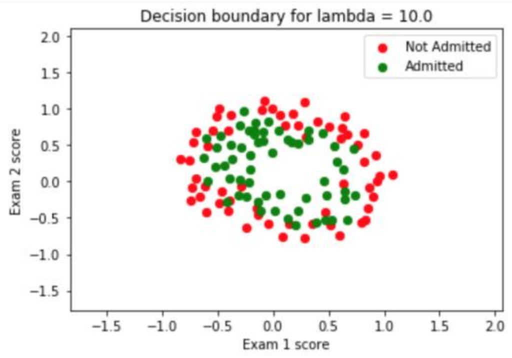


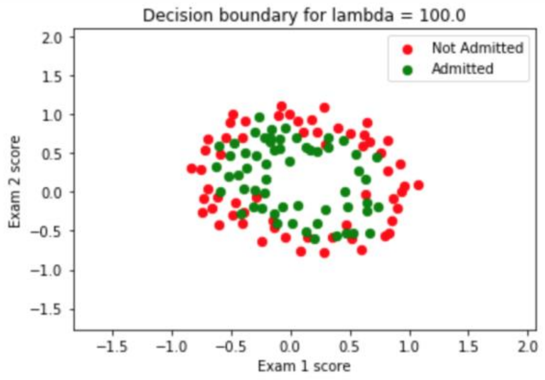
For L1,



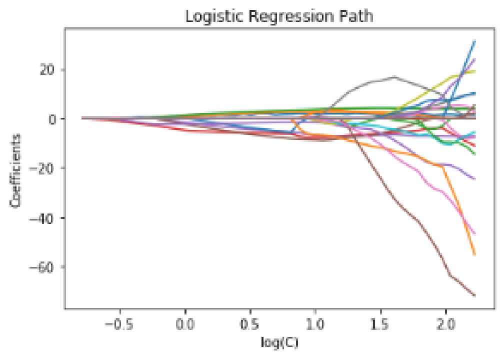








```
Computing regularization path ...
```



As λ increases from 0.01 to 100.0, both the models with L1 and L2 penalization change from overfitting to under-fitting. Besides the plots, the values of Loss of these models, which increase from about 0.3 to about 0.7 with the increasement of the regularization strength, also have this result confirmed.
The regression path diagram shows the number of non-zero coefficients of L1 regularization decreases when λ increases, especially when λ is approaching 10, all the coefficients become 0 and it cannot even decide the boundary. This also verifies the trend from overfitting to under-fitting of the model, since more non-zero coefficients represent a more complicated model, which is tend to overfit the training data

Part C: Logistic regression for spam classification

a) Feature transformation

b) Fitting regularized logistic regression models (L2 and L1)

```
L2  Penalty  experiments  -----------
best_lambda  =    0.1
Coefficients  =    [-4.86311363]  [[  -2.74146423e-02    -2.25297597e-01     1.21840933e-01     2.
29362879e+00
    2.70425715e-01     2.32851163e-01     9.28595395e-01     2.95200236e-01
    1.62205936e-01     6.78260362e-02    -8.32604386e-02    -1.60373354e-01
   -4.72247682e-02     1.07677111e-02     1.87903360e-01     8.19771812e-01
    5.09528973e-01     3.98711504e-02     2.67729695e-01     3.47047564e-01
    2.60498923e-01     3.64605215e-01     7.25019578e-01     1.96728249e-01
   -3.15395701e+00    -4.03133789e-01    -1.25451044e+01    -6.16580960e-02
   -1.56114609e+00    -5.51429801e-02    -3.00815864e-02     4.07263543e-01
   -3.68156446e-01    -1.43611787e+00    -5.87180606e-01     4.44294891e-01
    4.23159462e-02    -1.56897094e-01    -4.55330838e-01    -1.02250289e-01
   -3.54273295e+00    -1.72944487e+00    -4.37529300e-01    -1.05999941e+00
   -9.18599328e-01    -1.75490328e+00    -1.67475856e-01    -9.56875266e-01
   -3.65653149e-01    -1.36535510e-01    -6.58692488e-02     2.06714030e-01
    1.70694385e+00     1.21460313e+00    -3.35269880e-01     1.56141411e+00
    3.68775701e-01]]
Accuracy  on  set  aside  test  set  for    std    =    0.9296875
best_lambda  =    0.6
Coefficients  =    [-4.60944667]  [[-0.45145887  -0.28466502  -0.06327819    0.68295804    1.210
53197    0.91504987
   2.83046244    1.43677856    0.24145457    0.35775723  -0.38642638  -0.48142751
  -0.69586861    0.37457025    0.64885478    1.5395627     1.38118339    0.07197747
   0.37642217    0.63501926    0.52274838    0.38563703    2.00138627    1.50817467
  -3.14060836  -0.66617179  -4.90648468  -0.03260466  -1.28886313  -0.15745822
  -0.6389963    -0.30229058  -1.00990215  -0.42568565  -1.08721623    1.28432651
  -0.90558978  -0.35285879  -1.12971405  -0.62589095  -1.40337046  -2.44123337
  -1.55653325  -1.9477813   -1.13113514  -2.79991122  -0.7512231   -2.11601915
  -1.68510766  -0.66773402  -0.69125555   2.06913245   4.21977733   0.7630898
   0.70345803   0.17008574   0.43018819]]
Accuracy on set aside test set for  logt  =  0.943359375
best_lambda =  1.1
Coefficients =  [-1.83742964] [[ -1.91463199e-01  -1.66872958e-01  -3.93802023e-01   2.
39462779e-01
    9.83292893e-01   1.75311414e-01   2.12183419e+00   7.92547596e-01
    1.94566579e-01   3.34388296e-01  -2.90824615e-01  -4.20297341e-01
   -9.06380381e-01   2.56299856e-01   5.15189474e-01   1.47014136e+00
    8.76696476e-01  -8.32760956e-02   2.41264180e-01   5.01801273e-01
    7.37046896e-01   1.15518007e+00   9.11195183e-01   1.36902984e+00
   -2.35248856e+00  -4.17190306e-01  -3.79772643e+00   6.88337611e-01
   -6.07237597e-01  -1.61622832e-01  -9.24671804e-01  -6.04558748e-01
   -6.91161481e-01  -3.85638231e-02  -6.71440136e-01   3.52732370e-01
   -1.05408408e+00   5.28551480e-01  -7.65306731e-01  -2.46067578e-01
   -1.27643951e+00  -1.90613122e+00  -7.90184279e-01  -1.57619158e+00
   -7.64312034e-01  -2.22366816e+00  -8.34144234e-02  -1.39371572e+00
   -3.06993897e-01   2.00231957e-01  -1.70968577e-01   1.20762876e+00
    1.45771409e+00   3.79908693e-02   5.31813313e-04   5.31813313e-04
    5.31813313e-04]]
Accuracy on set aside test set for  bin  =  0.927734375

L1 Penalty experiments -----------
best_lambda =  4.6
Coefficients =  [-1.59528405] [[-0.01066616 -0.15872646  0.12269348  0.20794984  0.24
914094  0.1770254
   0.91014923  0.28982802  0.13971357  0.04855675 -0.02304995 -0.13981996
  -0.00706598  0.0091887   0.15331756  0.75699007  0.45992978  0.07035737
   0.25407585  0.19633242  0.24326464  0.3465077   0.72631226  0.23410927
  -2.34056308 -0.35781535 -3.18097986 -0.01105547 -0.37047654  0.          0.
   0.         -0.32746025  0.         -0.06139001  0.24253727  0.
  -0.11600145 -0.31161077 -0.04369546 -0.23960745 -0.7957659  -0.19073827
  -0.56433853 -0.73384605 -1.18124337 -0.08551656 -0.51372379 -0.25639933
  -0.13349696 -0.05666312  0.21824111  1.647554    0.22120305  0.
   0.64797357  0.33268323]]
Accuracy on set aside test set for  std  =  0.921875
best_lambda =  1.6
Coefficients =  [-4.46325713] [[-0.34343126 -0.09561861  0.          0.12935319  1.18
459629  0.69052579
   2.9132594   1.37677777  0.          0.29514542  0.          -0.48050956
  -0.32797159   0.1073368     0.            1.49525228   1.34902212   0.
   0.35495488   0.19675206   0.49438339   0.34756986   1.78473926   1.32956644
  -3.49784925  -0.2696488   -7.49408231   0.           -0.41744167   0.            0.
   0.           -0.79141606   0.           -0.23866668   0.87194694  -0.77437753
   0.           -0.88292016   0.           -0.30404293  -2.35514287  -0.69461756
  -1.6613016   -1.13789558  -2.98259264   0.           -1.90116358  -1.24510114
  -0.303593     0.            2.01475269   5.36669125   0.            0.63671312
   0.20187949   0.3881424  ]]
Accuracy on set aside test set for  logt  =  0.944010416667
best_lambda =  3.6
Coefficients =  [-0.6073033]  [[ 0.            0.           -0.19319604   0.            0.865
46434   0.
   2.02981932   0.63291657   0.02637316   0.21313141   0.           -0.42109053
  -0.68156116   0.            0.            1.31585884   0.76539192   0.
   0.10412766   0.12293073   0.63680137   0.73007646   0.6219361    1.18349939
  -2.42457594  -0.12536816  -3.73175969   0.            0.            0.            0.
   0.           -0.28795755   0.           -0.21895667   0.           -1.01577875
   0.           -0.40494512   0.           -0.11589481  -1.69459205  -0.03936507
  -1.10978283  -0.68778616  -2.21928382   0.           -1.02579743  -0.12514299
   0.07431402   0.            1.15123975   1.49962828   0.           -0.55358153
  -0.08874693  -0.41567272]]
Accuracy on set aside test set for  bin  =  0.92578125
```

c) Comment on the model sparsities with L1 and L2 regularization.

Which class of models will you recommend for this data set and why?

L1 regularization will have more sparse coefficients after regularization. And I may recommend the L1 models for this data set because not all the features of the data is crucial for a spam classification problem. But some of the features should have a high weight when we decide if the emails are spams. Thus, the sparsity of L1 regularization will help us focus on the important features and reduce the impact of insignificant features.