ELEN E4903: MACHINE LEARNING HOMEWORK 2

ELEN E4903: MACHINE LEARNING HOMEWORK 2

Problem 1:

$$y_0 = \arg\max_y p(y_0 = y|\pi) \prod_{d=1}^{D} p_d(x_{0,d}|\theta_y^{(d)})$$

$$\hat{\pi}, \hat{\theta}_y^{(1)}, \hat{\theta}_y^{(2)} = \arg\max_{\pi, \theta_y^{(1)}, \theta_y^{(2)}} \sum_{i=1}^{n} \ln p(y_i|\pi) + \sum_{i:y_i=y}^{n} \ln p(x_{i1}|\theta_y^{(1)}) + \sum_{i:y_i=y}^{n} \ln p(x_{i2}|\theta_y^{(2)})$$

(a) Solving for $\hat{\pi}$, $\nabla_\pi \sum_{i=1}^{n} \ln p(y_i|\pi) + \sum_{i:y_i=y}^{n} \ln p(x_{i1}|\theta_{y_i}^{(1)}) + \sum_{i:y_i=y}^{n} \ln p(x_{i2}|\theta_{y_i}^{(2)}) = 0$

$\nabla_\pi \sum_{i=1}^{n} \ln p(y_i|\pi) = 0$ since latter 2 terms don't have π

$\nabla_\pi \sum_{i=1}^{n} \ln\left(\pi^{y_i}(1-\pi)^{1-y_i}\right) = 0$ since $p(y_i = y|\pi) \sim$ Bernoulli $(y|\pi)$

$$\nabla_\pi \sum_{i=1}^{n} [y_i \ln \pi + (1-y_i)\ln(1-\pi)] = 0$$

$$\sum_{i=1}^{n} \left[\frac{y_i}{\hat{\pi}} - \frac{1-y_i}{1-\hat{\pi}}\right] = 0$$

$$\frac{\sum_{i=1}^{n} y_i}{n\hat{\pi}} - \frac{n - \sum_{i=1}^{n} y_i}{n - n\hat{\pi}} = 0$$

$$n^2\hat{\pi} - n\hat{\pi}\sum_{i=1}^{n} y_i = n\sum_{i=1}^{n} y_i - n\hat{\pi}\sum_{i=1}^{n} y_i$$

$$\therefore \hat{\pi} = \frac{\sum_{i=1}^{n} y_i}{n}$$

since y_i for $y=0$ is 0 and for $y=1$ is 1

This implies that $\hat{\pi}$ is the $\frac{\text{Total no of } y \text{ in class } y=1}{\text{Total no. of observations}}$

(b) It's worth noting that $\hat{\theta}_y^{(i)}$ is class conditional.

Solving for $\hat{\theta}_y^{(i)}$ leaving y arbitrary in the subscript,

$$\nabla_{\hat{\theta}_y^{(i)}} \sum_{i=1}^{n} \ln p(y_i|\pi) + \sum_{i=1}^{n} \ln p(x_{i1}|\theta_{y_i}^{(i)}) + \sum_{i=1}^{n} \ln p(x_{i2}|\theta_y^{(i)}) = 0$$

$$\nabla_{\hat{\theta}_y^{(i)}} \sum_{i=1}^{n} \ln p(x_{i1}|\theta_{y_i}^{(i)}) = 0 \quad \text{since the other terms don't contain } \hat{\theta}_y^{(i)}$$

$$\nabla_{\hat{\theta}_y^{(i)}} \sum_{i=1}^{n} \ln\left[(\theta_{y_i}^{(i)})^{x_{i1}}(1-\theta_{y_i}^{(i)})^{1-x_{i1}}\right] = 0 \quad \text{since p. is Bernoulli distribution}$$

$$\nabla_{\hat{\theta}_y^{(i)}} \sum_{i=1}^{n} \left[x_{i1} \ln \theta_{y_i}^{(i)} + (1-x_{i1}) \ln(1-\theta_{y_i}^{(i)})\right] = 0$$

$$\sum_{i=1}^{n} \left[\frac{x_{i1_{y_i=y}}}{\hat{\theta}_{y_i=y}^{(i)}} - \frac{1-x_{i1_{y_i=y}}}{1-\hat{\theta}_{y_i=y}^{(i)}}\right] = 0$$

with the subscript for $y_i=y$ since we only take the derivative if the class matches (class conditional)

$$\frac{\sum_{i=1}^{n} x_{i1_{y_i=y}}}{n_{y_i=y}\hat{\theta}_{y_i=y}^{(i)}} - \frac{n_{y_i=y} - \sum_{i=1}^{n} x_{i1_{y_i=y}}}{n_{y_i=y} - n_{y_i=y}\hat{\theta}_{y_i=y}^{(i)}} = 0$$

$$n_{y_i=y}^2 \hat{\theta}_{y_i=y}^{(i)} - n_{y_i=y}\hat{\theta}_{y_i=y}^{(i)} \sum_{i=1}^{n} x_{i1_{y_i=y}} = n_{y_i=y} \sum_{i=1}^{n} x_{i1_{y_i=y}} - n_{y_i=y}\hat{\theta}_y^{(i)} \sum_{i=1}^{n} x_{i1_{y_i=y}}$$

$$\therefore \hat{\theta}_y^{(i)} = \frac{\sum_{i=1}^{n} x_{i1_{y_i=y}}}{n_{y_i=y}}, \quad y \in \{0,1\}$$

This implies that for a particular class y, $\hat{\theta}_y^{(i)}$ is the $\frac{\text{Sum of all } x_{i1} \text{ across that class } y}{\text{No. of observations in that class } y}$ with y left arbitrary in the derivation and $\hat{\theta}_y^{(i)}$ class conditional.

(c) Like $\hat{\theta}_y^{(1)}$, $\hat{\theta}_y^{(2)}$ is also class conditional.

Solving for $\hat{\theta}_y^{(2)}$ leaving y arbitrary in the subscript,

$$\nabla\hat{\theta}_y^{(2)} \sum_{i=1}^{n} \ln p(y_i|\pi) + \sum_{i=1}^{n} \ln p(x_{i1}|\theta_{y_i}^{(1)}) + \sum_{i=1}^{n} \ln p(x_{i2}|\theta_{y_i}^{(2)}) = 0$$

$$\nabla\hat{\theta}_y^{(2)} \sum_{i=1}^{n} \ln p(x_{i2}|\theta_{y_i}^{(2)}) = 0 \quad \text{since the other terms don't contain } \theta_y^{(2)}$$

$$\nabla\hat{\theta}_y^{(2)} \sum_{i=1}^{n} \ln\left[\theta_{y_i}^{(2)} (x_{i2})^{-(\theta_{y_i}^{(2)}+1)}\right] = 0 \quad \text{since } p_2 \text{ is a Pareto distribution}$$

$$\nabla\hat{\theta}_y^{(2)} \sum_{i=1}^{n} \left[\ln\theta_{y_i}^{(2)} - (\theta_{y_i}^{(2)}+1)\ln x_{i2}\right] = 0$$

$$\sum_{i=1}^{n} \left(\frac{1}{\theta_{y_i=y}^{(2)}} - \ln x_{i2_{y_i=y}}\right) = 0$$

$$\frac{n_{y_i=y}}{\theta_{y_i=y}^{(2)}} - \sum_{i=1}^{n} \ln x_{i2_{y_i=y}} = 0$$

$$\therefore \hat{\theta}_y^{(2)} = \frac{n_{y_i=y}}{\sum_{i=1}^{n} \ln x_{i2_{y_i=y}}}, \quad y \in \{0,1\}$$

This implies that for a particular class y, $\hat{\theta}_y^{(2)}$ is the $\frac{\text{No of observations in that class } y}{\text{Sum of the } \ln x_{i2} \text{ across that class } y}$ with y left arbitrary in the derivation and $\hat{\theta}_y^{(2)}$ being class conditional.

ELEN E4903: MACHINE LEARNING HOMEWORK 2

Problem 2:

(a) Implementing the naïve Bayes classifier derived in problem 1, I was able to obtain a vector of the parameters θ_y^i for $i = 1, 2, 3 \ldots, 57$ where the first 54 are for the Bernoulli distribution and the last 3 for the Pareto distribution as well as for $y = \{0, 1\}$ since those parameters are class conditional. I was also able to obtain $\pi = 0.39397$. The parameters obtained using the solutions from Problem 1 are implemented on the training data are shown below:

For y = 0,

	V1.x	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14
1	0.1486091	0.09919473	0.2770864	0.002928258	0.2225476	0.1153001	0.01573939	0.07320644	0.07942899	0.1698389	0.05197657	0.4256955	0.1174963	0.04538799

V15	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28	V29
0.01793558	0.09114202	0.09553441	0.124817	0.5816252	0.01683748	0.3418741	0.008052709	0.02745242	0.01976574	0.3737189	0.2818448	0.2774524	0.1559297	0.1292094

V30	V31	V32	V33	V34	V35	V36	V37	V38	V39	V40	V41	V42	V43	V44
0.1614202	0.1035871	0.07320644	0.1237189	0.07393851	0.1577599	0.1749634	0.2620791	0.01830161	0.1145681	0.08931186	0.05270864	0.1156662	0.1039531	0.1002928

V40	V41	V42	V43	V44	V45	V46	V47	V48	V49	V50	V51	V52	V53	V54
0.08931186	0.05270864	0.1156662	0.1039531	0.1002928	0.2975842	0.1617862	0.01610542	0.06771596	0.1859444	0.5512445	0.1427526	0.2690337	0.1046852	0.08272328

V55	V56	V57
1.519735	0.461781	0.2507241

For y = 1,

V1.x	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15
0.3536036	0.3417793	0.6131757	0.0213964	0.6233108	0.3761261	0.419482	0.3440315	0.3079955	0.4583333	0.3136261	0.6295045	0.2871622	0.1278153	0.160473

V16	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28	V29	V30
0.545045	0.3834459	0.3811937	0.8862613	0.2100225	0.8063063	0.05292793	0.3344595	0.3766892	0.02759009	0.01463964	0.003941441	0.01689189	0.006756757	0.009572

V31	V32	V33	V34	V35	V36	V37	V38	V39	V40	V41	V42	V43	V44
0.001689189	0.001126126	0.03434685	0.005630631	0.02533784	0.0625	0.05574324	0.01801802	0.03322072	0.1120495	0.0005630631	0.01126126	0.04786036	0.0259009

V45	V46	V47	V48	V49	V50	V51	V52	V53	V54
0.2697072	0.03828829	0.01013514	0.009009009	0.151464	0.6531532	0.07207207	0.8333333	0.6148649	0.2865991

V55	V56	V57
0.7331172	0.2712874	0.185925

Using the parameters obtained from the training data, I then predicted on the testing data using the following rule where for each $x_{0,d}$ observed, the y_0 predicted is:

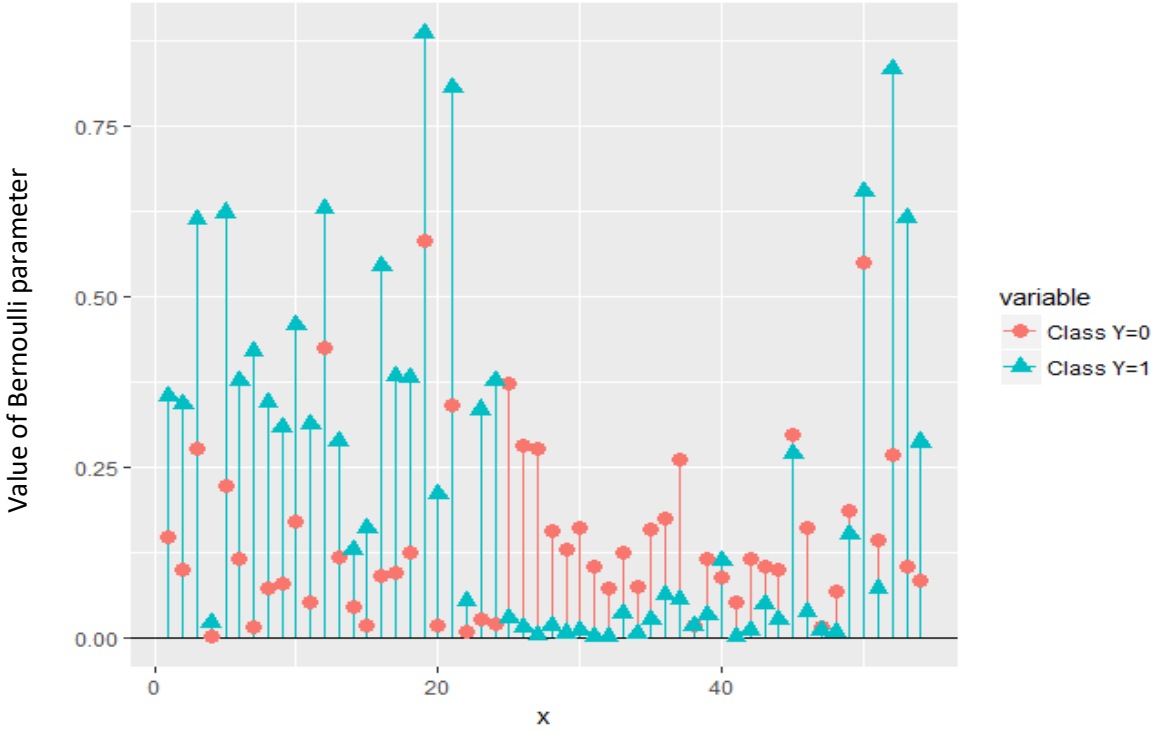
$$y_0 = argmax_y p(y_0 = y|\pi) \prod_{d=1}^{D} p_d(x_{0,d}|\theta_y^d)$$

Finally. comparing the ground truth in the test data for the class y data point with the model prediction y', the following table can be obtained:

	Ground Truth y = 0	Ground Truth y = 1
Model Prediction y' = 0	53	4
Model Prediction y' = 1	3	33

Prediction accuracy = $\frac{53+33}{93} = 0.925$

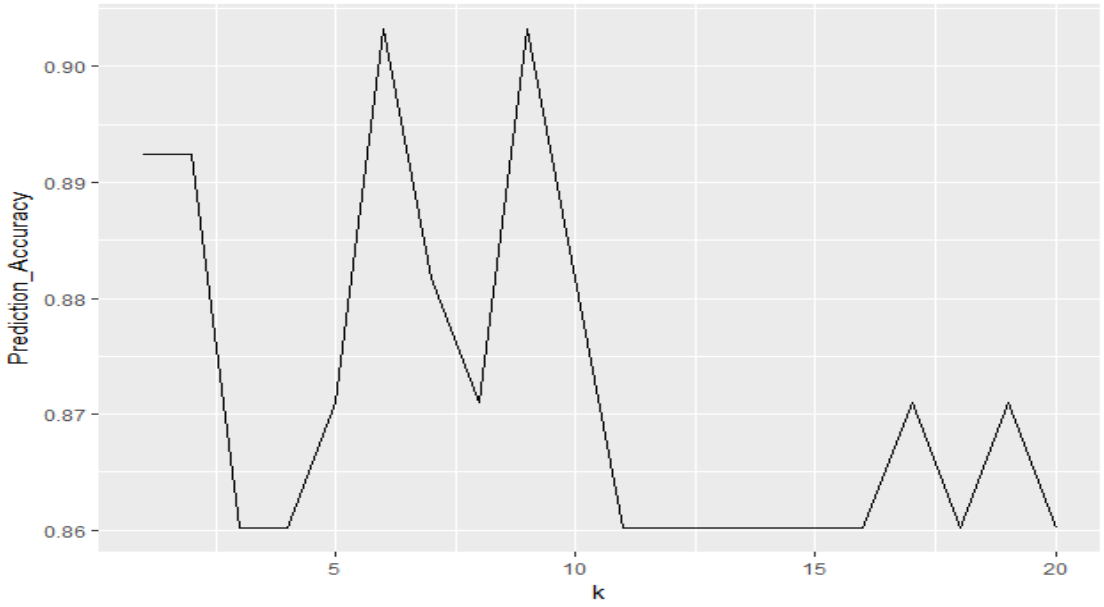
(b) The following is the stem plot for the 54 Bernoulli parameters for each class:



The file "spambase.names" consists of words and characters commonly found in an email. Dimension 16 and 52 in the file correspond to the word "free" and the character "!" respectively. Comparing that to the stemplot above, in dimension **16 and 52**, it can be seen that the **value of the Bernoulli parameter for class y = 1 (spam email) is larger than** that for class **y = 0 (non-spam email).** This **pushes up the probability of the email being classified as a spam once these dimensions** (for the word "free" and character "!") **are present** since $p_d(x_{0,d}|\theta_y^d)$ **for** $d = 16$ **and** $d = 52$ **would be larger for y = 1 than for y = 0**. This makes intuitive sense since "free" and "!" are elements commonly found in spam emails.

ELEN E4903: MACHINE LEARNING HOMEWORK 2

(c) By implementing the k-NN algorithm which is done by firstly **finding the k points closest to each X** (a vector of all 57 dimensions of x for each of the 93 inputs of X in the testing data) defining the **distance** between the X in the training data and testing data as the l_1 distance of $\sum_{d=1}^{57} |\boldsymbol{X}_{\boldsymbol{test,d}} - \boldsymbol{X}_{\boldsymbol{train,d}}|$, secondly **returning the majority vote of y (whether there are more y = 1 or y = 0 classes in the k closest points** based on the training data for that X), thirdly **breaking ties in both steps at random**, **lastly repeating the classification for all 93 observations of X** in the testing data and **for k = 1 to 20**, the following plot of prediction accuracy against k is obtained:



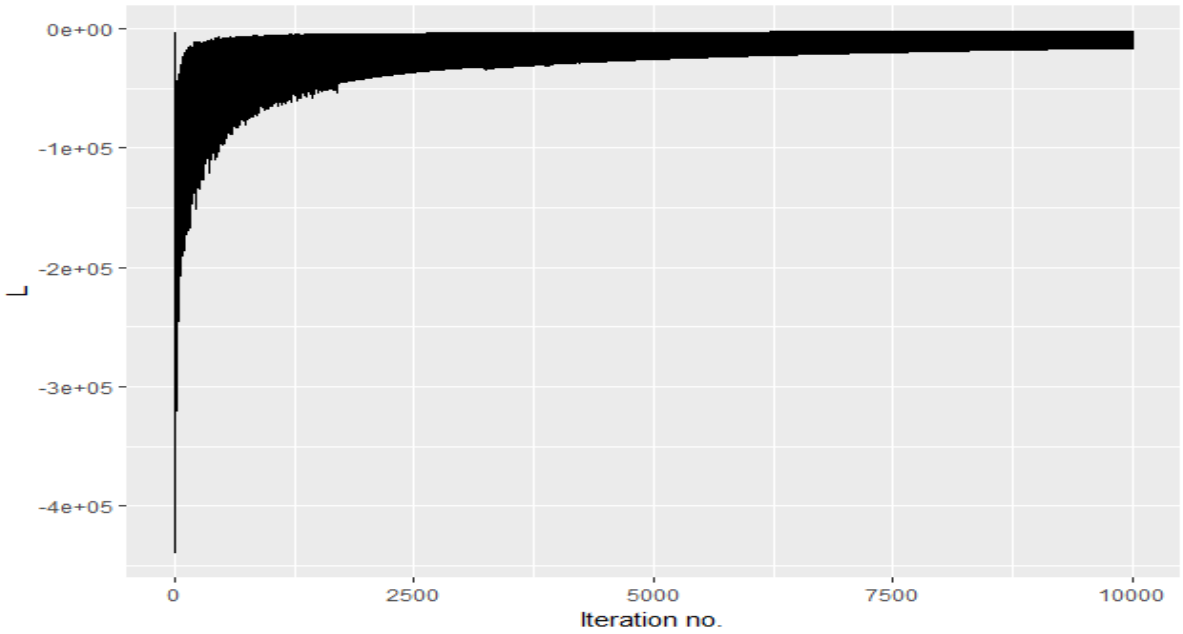
From the above plot of prediction accuracy against k, it can be deduced that the highest prediction accuracy occurs when k = 9, giving a prediction accuracy of 0.903 (or 90.3%) whilst the lowest prediction accuracy occurs when k = 3, giving a prediction accuracy of 0.860 (or 86.0%). It is also worth noting that since ties are broken at random, a slightly different graph is produced each time the algorithm is run.

ELEN E4903: MACHINE LEARNING HOMEWORK 2

(d) The steepest ascent algorithm for this part is carried out via the following steps:

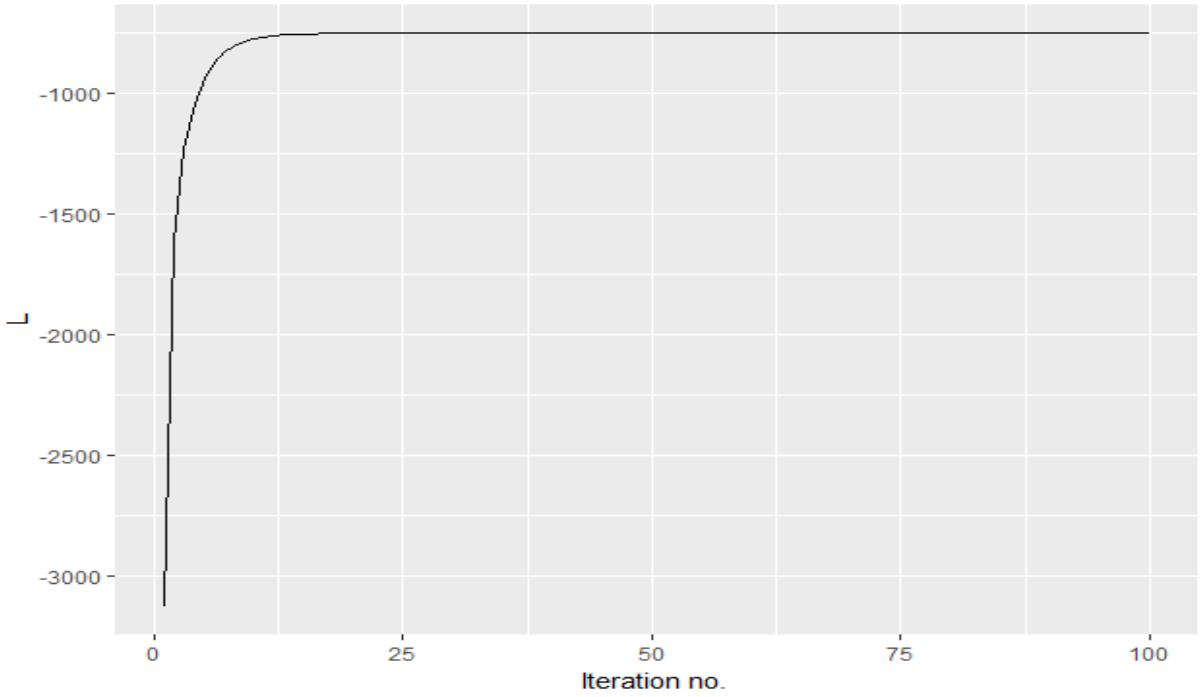
1. Using training data $(X_1, y_1), \dots, (X_{4508}, y_{4508})$, where X_i is 58 x 1 vector of all dimensions and y_i is a scalar representing the class, define a vector that is 58 x 1 called $w^{(1)} = \vec{0}$
2. For iteration $t = 1,2, \dots, 10000$, update $w^{(t+1)} = w^{(t)} + \eta_t \sum_{i=1}^{4508} \left(1 - \sigma_i\left(y_i . w^{(t)}\right)\right) y_i X_i$ where $\eta_t = \frac{1}{10^5\sqrt{(t+1)}}$ and $\sigma_i\left(y_i . w^{(t)}\right) = \frac{e^{y_i X_i^T w}}{1+e^{y_i X_i^T w}}$
3. Obtain logistic regression objective training function, $L = \sum_{i=1}^{4508} ln\, \sigma_i\left(y_i . w^{(t)}\right)$

The following graph of L against iteration number is obtained after coding the above steps:



The reason why the pattern looks strange is because the logistic regression objective training function L is not changing monotonically, it oscillates between a high value and low value after every iteration. Usually, after running all the iterations, the w that corresponds to the iteration that yielded the largest L would be chosen and used to predict in the test data. In this case, the highest value of L obtained after 10000 iterations is -2382.312 obtained during the 9998th iteration.

(e) For "Newton's method", the main change compared to steepest ascent is to change the way $w^{(t)}$ is updated. At iteration t, now set $w^{(t+1)} = w^{(t)} - \eta_t(\nabla_w^2 L)^{-1}\nabla_w L$ where $\eta_t = \frac{1}{\sqrt{(t+1)}}$ and only do 100 iterations. The following is the graph of L against iteration number for Newton's method:



	Ground Truth y = -1	Ground Truth y = 1
Model Prediction y' = -1	54	6
Model Prediction y' = 1	2	31

Prediction accuracy = $\frac{54+31}{93} = 0.914$

For this part, there is indeed a need to predict in the testing data to obtain the prediction accuracy. To do that, after running all the iterations, the w that corresponds to the iteration that yielded the largest L would be chosen. In this case, the highest value of L obtained after 10000 iterations is -750.7107 obtained during the 42nd iteration and lasting all the way till the 100th iteration. The corresponding vector w used for prediction is shown below with "newcolumn" being w_0 and the other V_i being the value of w corresponding to the respective dimension of X:

V1	-0.6201652608
V2	-0.1652014104
V3	-0.4859882879
V4	0.6559388180
V5	1.1136687338
V6	0.2640306777
V7	2.4910464106
V8	0.9832908593
V9	0.2639803414
V10	0.3582122463
V11	-0.4387315790
V12	-0.3387495125
V13	-0.9788182684
V14	0.8514318482
V15	1.3307721660
V16	1.5606466670
V17	1.0485624924

V18	-0.4874418958
V19	0.1818564196
V20	0.5153830487
V21	0.6654690561
V22	1.3657795459
V23	0.9549713150
V24	1.7468889501
V25	-3.6883380165
V26	-0.2325684906
V27	-6.2792906974
V28	2.1928061100
V29	-0.6371493500
V30	-0.3746895691
V31	-2.2364983547
V32	-1.5763510142
V33	-0.7866088825
V34	1.0528741029

V35	-1.7230387233
V36	0.4675550791
V37	-1.0828028276
V38	1.5877920961
V39	-0.5722115589
V40	-0.0987794236
V41	-6.3716405002
V42	-2.5331903316
V43	-1.2286304922
V44	-1.7571863307
V45	-1.0421209934
V46	-2.4493541620
V47	0.2762629460
V48	-2.2746944104
V49	-0.2580654409
V50	0.1838076557
V51	-0.3356871486

V52	1.3532155385
V53	1.9016671537
V54	-0.9006894507
V55	-0.0019486469
V56	0.0056613509
V57	0.0008598616
newcolumn	-1.9971222713