Xiaoqi Zhou (xqzhou@stanford.edu)
06237147

1. **Solution:**
$\hat{\theta} = \underset{\theta}{argmax} LL(\theta)$
$LL(\theta) = \sum_{i=1}^{n} \log \lambda e^{-\lambda X_i} = \sum_{i=1}^{n} \log \lambda - \lambda X_i$
$\frac{\delta LL(\lambda)}{\delta \lambda} = \frac{n}{\lambda} - \sum_{i=1}^{n} X_i = 0$
Answer:
$\hat{\lambda} = \frac{n}{\sum_{i=1}^{n} X_i}$

2. **Solution:**
$Y_i \sim N(\theta_1 X_i + \theta_2, \sigma^2)$
$LL(\theta_1, \theta_2) = \sum_{i=1}^{n} \log(\frac{1}{\sqrt{2\pi}\sigma}) - \frac{(Y_i - \theta_1 X_i - \theta_2)^2}{2\sigma^2}$
Answer:
$LL(\theta_1, \theta_2) = -\frac{n}{2}\log(2\pi) - n\log(\sigma) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(Y_i - \theta_1 X_i - \theta_2)^2$

3. **Answer:**
One important assumption of the Naive Bayesian Classifier is all the inputs X_i from the training set it independent. If the first k input of the training are always identical copies.
$P(X_1, X_2, ..., X_n|Y) \neq \prod_{i=1}^{n} P(X_i|Y)$
And we have no data can be used as reference to estimate the $P(X_1, X_2, ..., X_n|Y)$, when $X_1, X_2, ...X_k$ are NOT copies of each other.

4. (a) $\frac{\delta LL(\theta)}{\theta_1} = \frac{\delta LL(\hat{y})}{\hat{y}} \frac{\delta \hat{y}}{\delta(\theta_3 g + \theta_4 h)} \frac{\delta(\theta_3 g + \theta_4 h)}{\delta g} \frac{\delta g}{\delta \theta_1 x} \frac{\delta \theta_1 x}{\delta \theta_1}$
$\frac{\delta LL(\theta)}{\theta_1} = \left[\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}\right] \hat{y}(1-\hat{y})\theta_3 g(1-g)x$
$\frac{\delta LL(\theta)}{\theta_2} = \left[\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}\right] \hat{y}(1-\hat{y})\theta_4 h(1-h)x$
$\frac{\delta LL(\theta)}{\theta_3} = \left[\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}\right] \hat{y}(1-\hat{y})g$
$\frac{\delta LL(\theta)}{\theta_4} = \left[\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}\right] \hat{y}(1-\hat{y})h$

   (b) The results show the chain rule of derivative. We can use computer to calculate the derviative of all the parameters based on a given neural network.
We can define a small step η, update the derivative and iterate the procedure enough times. Essentially, the estimation of each particular parameter θ_i will provide the highest log-likelihood. These are the final values of θ_i.
By using the parameter θ_i from the training data set we can predict the output Y from any given X.

5. (a) Both MLE and Laplace Estimators provide 100% accuracy
   (b) MLE: $P(Y = 1) = 0.496$
Laplace: $P(Y = 1) = 0.496$

Xiaoqi Zhou (xqzhou@stanford.edu)
06237147

$P(X_i = 1\|Y = y)$	MLE		Laplace	
i	$y = 0$	$y = 1$	$y = 0$	$y = 1$
0	0.586	0.671	0.586	0.671
1	0.684	0.667	0.683	0.666
2	0.676	0.594	0.675	0.594
3	0.657	0.605	0.657	0.605
4	0.568	0.722	0.568	0.721
5	0.598	0.607	0.598	0.607
6	0.622	0.601	0.622	0.601
7	0.600	0.696	0.600	0.695
8	0.862	0.861	0.862	0.861
9	0.697	0.658	0.697	0.658
10	0.664	0.680	0.664	0.679
11	0.643	0.662	0.643	0.661
12	0.876	0.876	0.875	0.876
13	0.672	0.671	0.672	0.671
14	0.567	0.509	0.567	0.509
15	0.555	0.516	0.555	0.516
16	0.761	0.750	0.760	0.750
17	0.567	0.709	0.567	0.709
18	0.312	0.748	0.313	0.747

netflix-test MLE accuracy = 0.730
netflix-test Laplace accuracy = 0.730

(c) $P(Y = 1|X_i = 1) = \frac{P(X_i=1|Y=1)P(Y=1)}{P(X_i=1)}$
$P(Y = 1|X_i = 0) = \frac{P(X_i=0|Y=1)P(Y=1)}{P(X_i=0)}$
$r = \frac{P(Y=1|X_i=1)}{P(Y=1|X_i=0} = \frac{P(X_i=1|Y=1)P(X_i=0)}{P(X_i=0|Y=1)P(X_i=1)}$

i	r_i
0	1.208
1	0.962
2	0.840
3	0.895
4	1.433
5	1.020
6	0.956
7	1.245
8	0.996
9	0.917
10	1.036
11	1.041
12	1.001
13	1.000
14	0.890
15	0.923
16	0.971
17	1.386
18	2.646

The index of the most indicative movie that a user will like Love Actually is
18+1 = 19: When Harry Met Sally,
4+1 = 5: How to Lose a Guy in 10 Days,
17+1 = 18: What Women Want,
7+1 = 8: La Vita E Bella,
0+1 = 1: Idiots

Xiaoqi Zhou (xqzhou@stanford.edu)
06237147

There may be a miss-prediction of 13: Pirates. Since the total sample number when $X_{12} = 0$ is much lower than $X_{12} = 1$. The estimation of $P(Y = 1|X_{12} = 0)$ may not be accurate and the difference between $Y = 1|X_{12} = 0$ and $Y = 1|X_{12} = 1$ are very small already.

(d) ancestry-test MLE accuracy = 0.788
ancestry-test Laplace accuracy = 0.777

(e) heart-test MLE accruacy = 0.775
heart-test Laplace accruacy = 0.749

6. (a) Logistic Regression provide 100% accuracy

(b) Final parameter

i	θ_i
0	-1.265
1	0.147
2	-0.032
3	-0.193
4	-0.111
5	0.328
6	0.032
7	-0.112
8	0.216
9	-0.038
10	-0.083
11	0.0822
12	0.048
13	0.021
14	0.001
15	-0.105
16	0.008
17	-0.017
18	0.278
19	1.756

netflix-test logistic regressing accuracy = 0.740

19: When Harry Met Sally,
5: How to Lose a Guy in 10 Days,
18: What Women Want,
8: La Vita E Bella,
1: Idiots
The log likelihood of the train data when all the parameters are 0 is
$LL(0) = -346.574$
The log likelihood of the train data when all the parameters are after training is
$LL(\theta) = -282.978$

(c) ancestry-test logistic regressing accuracy = 0.837

(d) We can find the smaller step length can provide better performance, but in the same time the $LL(\theta)$ reduces after $\eta = 0.0002$. Because in the beginning the smaller step can make the iteration arrive the top of $LL(\theta)$ more accurate, but if the step is too small based on the same iteration

Xiaoqi Zhou (xqzhou@stanford.edu)
06237147

step number, there is a change that θ never reach the optimized point.
By checking the test data set, most of the Y are one. So even we have a non-optimized θ, as long as the most estimation results are ones, we can still get a good results for this particular data set.

η	Classification Accuracy	$LL(\theta)$
0.0025	0.679	-154.5
0.0005	0.732	-103.0
0.0001	0.775	-83.1
0.00002	0.786	-79.1
0.000004	0.882	-87.4
0.0000008	0.936	-110.9
0.00000016	0.936	- 124.9

The code is attached from next page. Both problem 5 and problem 6 are using the same entrance main.py
File list:
main.py
dataLoader.py
MLETrainer.py
laplaceTrainer.py
logRegTrainer.py