1.

a) when $y = \text{Keep}$, $E[J(y,t)] = 0 \cdot (1-0.2) + 1 \cdot 0.2$
$= 0.2$

when $y = \text{Remove}$, $E[J(y,t)] = 500 \cdot (1-0.2) + 0 \cdot 0.2$
$= 400$

b) Want to minimize expected loss given conditional probability i.e. $E[J(y,t)|x]$

$$E[J(y,t)|x] = J(t=\text{Spam}, y) \cdot Pr(t=\text{Spam}|x) + J(t=\text{NonSpam}, y) \cdot Pr(t=\text{NonSpam}|x)$$

Let $Pr(t=\text{Spam}|x) = p \Rightarrow Pr(t=\text{NonSpam}|x) = 1-p$

$\Rightarrow J(t=\text{Spam}, y) \cdot Pr(t=\text{Spam}|x) = 1 \cdot p = p$

$\Rightarrow J(t=\text{NonSpam}, y) \cdot Pr(t=\text{NonSpam}|x) = 500 \cdot (1-p)$

set $p = 500(1-p) \Rightarrow 500 = 501p$

$\Rightarrow p = \frac{500}{501}$

$\Rightarrow$ if $Pr(t=\text{Spam}|x) \leq \frac{500}{501}$, $y^* = \text{keep}$

if $Pr(t=\text{Spam}|x) > \frac{500}{501}$, $y^* = \text{remove}$

c) $Pr(Spam \mid x) = \dfrac{Pr(x \mid Spam)\, Pr(Spam)}{Pr(x)}$

$Pr(Spam \mid (x_1, x_2) = (0,0)) = \dfrac{0.45 \times 0.2}{0.45 \times 0.2 + 0.996 \times 0.8}$

$= \dfrac{0.09}{0.09 + 0.7968} = \dfrac{225}{2217}$

$Pr(Spam \mid (x_1, x_2) = (0,1)) = \dfrac{0.25 \times 0.2}{0.25 \times 0.2 + 0.002 \times 0.8}$

$= \dfrac{0.05}{0.05 + 0.0016} = \dfrac{125}{129}$

$Pr(Spam \mid (x_1, x_2) = (1,0)) = \dfrac{0.18 \times 0.2}{0.18 \times 0.2 + 0.002 \times 0.8}$

$= \dfrac{0.036}{0.036 + 0.0016} = \dfrac{45}{47}$

$Pr(Spam \mid (x_1, x_2) = (1,1)) = \dfrac{0.12 \times 0.2}{0.12 \times 0.2 + 0 \times 0.8}$

$= 1$

According to Part b)

when $(x_1, x_2) = (0,0), (0,1), (1,0)$

$y^* = \text{keep}$

when $(x_1, x_2) = (1,1)$, $y^* = \text{remove}$

d)

$$E[J(y_*, t)] = E[E(J(y_*, t)|x)]$$

$$= \sum_x E(J(y_*, t)|x) \cdot Pr_x(x)$$

$$Pr((x_1, x_2) = (0,0)) = 0.45 \times 0.2 + 0.996 \times 0.8$$
$$= 0.8868$$

$$Pr((x_1, x_2) = (0,1)) = 0.25 \times 0.2 + 0.002 \times 0.8$$
$$= 0.0516$$

$$Pr((x_1, x_2) = (1,0)) = 0.18 \times 0.2 + 0.002 \times 0.8$$
$$= 0.0376$$

$$\Pr((x_1, x_2) = (1,1)) = 0.12 \times 0.2 + 0 \times 0.8$$
$$= 0.024$$

$$\Rightarrow E[J(y_*, t)]$$
$$= 0.8868 \times \frac{225}{227} + 0.0516 \times \frac{125}{129}$$
$$+ 0.0376 \times \frac{45}{47} + 0.024 \times 0$$
$$= 0.09 + 0.05 + 0.036 + 0.024 \times 0$$
$$= 0.176$$

2

a) Suppose the dataset is linearly separable.

since when $(x_1, x_2) = (-2, -1)$ and $(2, 3)$,

$y = 1$ which is positive, in such a case,

any segment connecting these two points should also be positive if dataset is linearly separable

Let $\lambda = 0.25$, $1 - \lambda = 0.75$

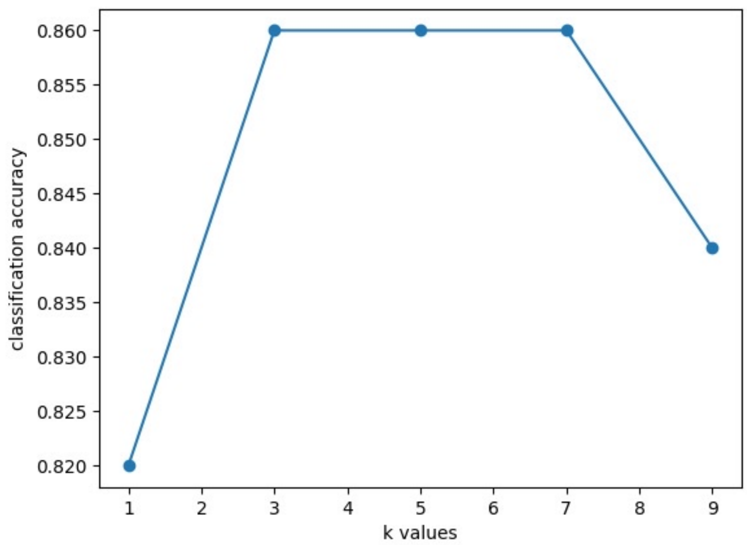
$\lambda(-2,-1) + (1-\lambda)(2,3) = 0.25(-2,-1) + 0.75(2,3) = (1,2)$

which according to database, $y=0$ not positive

so dataset is not linear separable.

3.1

a)



b) From the graph above, we can see classification accuracy keeps at highest which is 0.86 for $k=3$ to $k=7$.

when $k<3$, model is overfit.

when $k>7$, model is underfit

Therefore, we choose $k^*=5$, $k^*-2=3$, $k^*+2=7$, all keep the highest classification accuracy 0.86.

$k^*=5$ is neither underfit nor overfit.

The validation accuracy for k^*-2, k^*, k^*+2 is 0.86

The test accuracy for k^*-2 is 0.92, for k^* and k^*+2 is 0.94.

Test accuracy are all higher than validation accuracy.

3.2

a)

For mnist_train, learning rate is 0.05, number of iterations are 800.

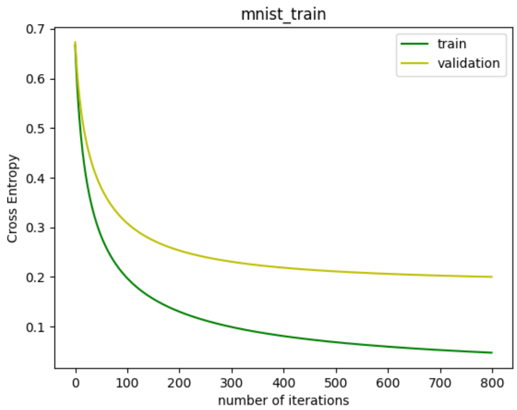
	cross entropy	classification accuracy
training data	0.04733	1
validation data	0.20011	0.88
test data	0.20143	0.92

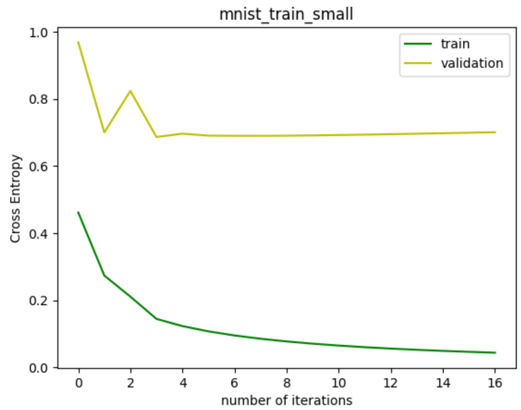
For mnist_train_small, learning rate is 0.3, number of iterations are 17.

	cross entropy	classification error
training data	0.04391	1
validation data	0.7009	0.66
test data	0.57908	0.78

b) In both graph, the cross entropy for train keeps decreasing. The cross entropy for validation would fluctuate when iteration is small and dataset is small.

To choose best parameter, I would take an average of the learning rate and number of iterations





4

a) $\frac{\lambda}{2}||w||^2 = \frac{\lambda}{2}\sum_{i=1}^{N} w_i^2$

$$\Rightarrow f = \frac{1}{2}\sum_{i=1}^{N} a^{(i)}(y^{(i)} - w^T x^{(i)})^2 + \frac{\lambda}{2}\sum_{i=1}^{N} w_i^2$$

$$\frac{\partial f}{\partial w_j} = \sum_{i=1}^{N} a^{(i)}(y^{(i)} - w^T x^{(i)})(-x_j^{(i)}) + \lambda w_j$$

$$= \sum_{i=1}^{N} a^{(i)} x_j^{(i)}\left(\sum_{k=1}^{D} w_k x_k^{(i)} - y^{(i)}\right) + \lambda w_j$$

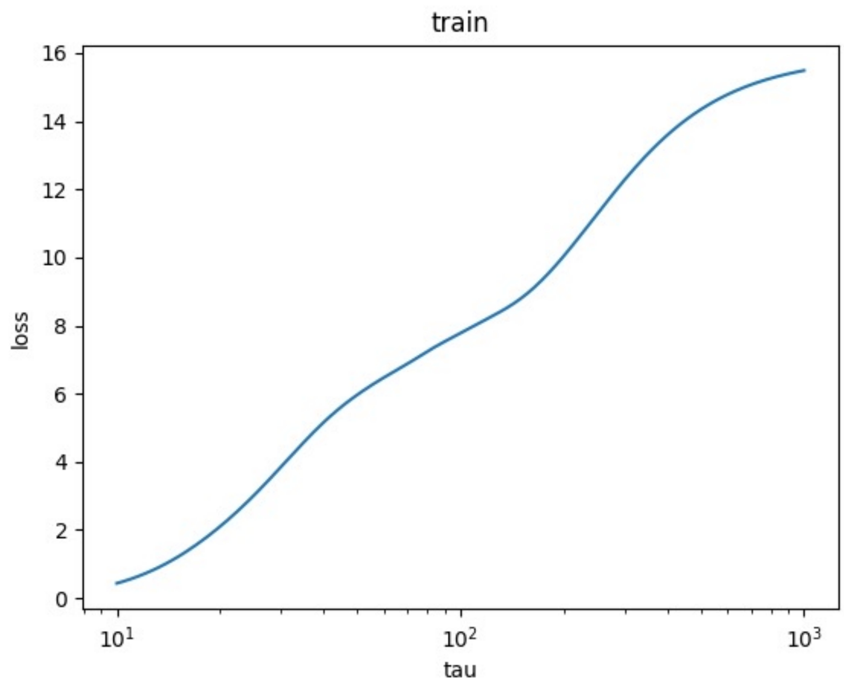
$$= \sum_{k=1}^{D} w_k \sum_{i=1}^{N} a^{(i)} x_k^{(i)} x_j^{(i)} - \sum_{i=1}^{N} a^{(i)} y^{(i)} x_j^{(i)} + \lambda w_j$$

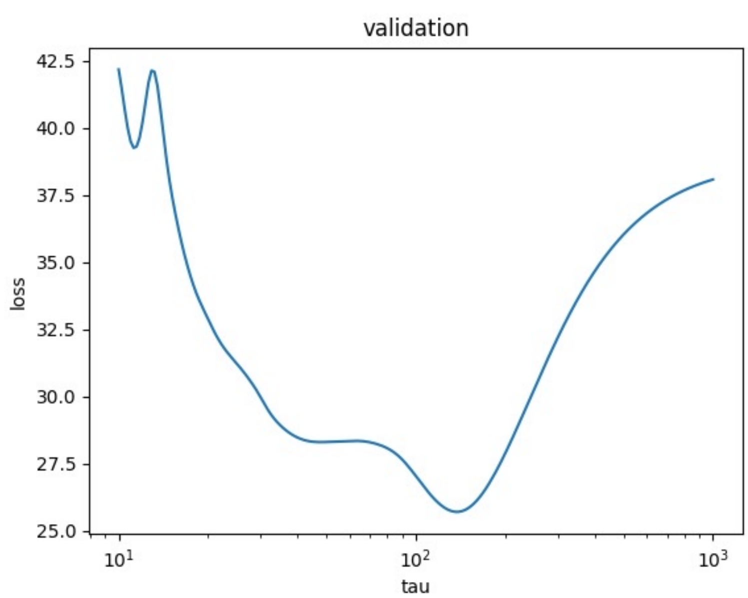
$$\Rightarrow x^T A x w - (y^T A^T x)^T + \lambda I w = 0$$

$$(x^T A x + \lambda I)w = x^T A y$$

$$w = (x^T A x + \lambda I)^{-1} x^T A y$$

c)


train
16
14
12
10
8
6
4
2
0
loss
10^1
10^2
10^3
tau


validation
42.5
40.0
37.5
35.0
32.5
30.0
27.5
25.0
loss
10^1
10^2
10^3
tau

d)

$$\lim_{\tau \to \infty} a^{(i)} = \lim_{\tau \to \infty} \frac{\exp\left(-\frac{\|x-x^{(i)}\|^2}{2\tau^2}\right)}{\sum_j \exp\left(-\frac{\|x-x^{(j)}\|^2}{2\tau^2}\right)}$$

$$= \lim_{\tau \to \infty} \frac{1}{\sum_j \exp\left(\frac{-\|x-x^{(j)}\|^2 + \|x-x^{(i)}\|^2}{2\tau^2}\right)}$$

$$= \lim \frac{1}{\sum_j 1}$$

$$= \frac{1}{N}$$

as $\tau \to \infty$, for both train and validation loss, would converge to constant

As $\tau \to 0$, $a^{(i)} \to \infty$, in graph, training loss close to 0, which means overfitting.

e)
advantage: Since linear regression model may result in too high error when the data set cannot be fit into a straight line, locally weighted regression could help in such a case.

disadvantage: locally weighted regression could cause overfit.