Homework 2

(1) Expected Loss & Bayes Optimality

a) i) Keeping every email: $\mathbb{E}[J(y,t)] = 0.2(0) + 0.8(1) = 0.8$

ii) Discarding every email: $\mathbb{E}[J(y,t)] = 0.2(0) + 0.8(500) = 400$

b) In order to make the Bayes optimal prediction, we compare the expected loss of each action (keeping vs discarding the email):

1) $\text{Risk}_{\text{keep}} = P(\text{spam}|x) \cdot 1 + P(\text{non-spam}|x) \cdot 0 = P(\text{spam}|x)$

2) $\text{Risk}_{\text{remove}} = P(\text{spam}|x) \cdot 0 + P(\text{non-spam}|x) \cdot 500 = (1 - P(\text{spam}|x)) \cdot 500$

We do whichever action is less risky; i.e, we choose based on

"$\text{Risk}_{\text{keep}} < \text{Risk}_{\text{remove}}$"?

Homework 2

(1) Expected Loss & Bayes Optimality

b) For instance, we only remove the email if

$$\text{Risk}_{\text{remove}} < \text{Risk}_{\text{keep}}$$

$$(1 - P(\text{Spam}|x)) \cdot 500 < P(\text{Spam}|x)$$

$$\Rightarrow P(\text{spam}|x) > 500\,(1 - P(\text{Spam}|x)) = 500 - 500\,P(\text{Spam}|x)$$

$$501\,P(\text{spam}|x) > 500$$

$$\therefore P(\text{spam}|x) > \frac{500}{501}$$

Homework 2

(1) Expected Loss & Bayes Optimality

c) We follow the same line of reasoning as before, but using (x_1, x_2) rather than $\vec{x}$:

i) $(x_1 = 0, x_2 = 0)$

$$P(x_1 = 0, x_2 = 0) = 0.2(0.45) + 0.8(0.996) = 0.8868$$

$$\text{Risk}_{\text{keep}} = P(\text{spam} \mid (x_1 = 0, x_2 = 0)) \cdot 1 = \frac{P(x_1, x_2 \mid \text{spam}) \cdot P(\text{spam})}{P(x_1, x_2)} = \frac{(0.45)(0.2)}{0.8868}$$

$$= 0.1015$$

$$\text{Risk}_{\text{remove}} = P(\text{nonspam} \mid (x_1 = 0, x_2 = 0)) \cdot 500 = \frac{P(x_1, x_2 \mid \text{non-spam}) \cdot P(\text{non-spam})}{P(x_1, x_2)} = \frac{(0.996)(0.8)}{0.8868}$$

$$= 449.3$$

$\therefore$ $(x_1 = 0, x_2 = 0) \Rightarrow \text{Risk}_{\text{keep}} < \text{Risk}_{\text{remove}} \Rightarrow$ keep the email

Homework 2

(1) Expected Loss & Bayes Optimality

c) ii) $(x_1 = 1, x_2 = 0)$

$$P(x_1 = 1, x_2 = 0) = 0.2(0.18) + 0.8(0.002) = 0.0376$$

$$\text{Risk}_{\text{keep}} = P(\text{spam} \mid (x_1 = 1, x_2 = 0)) \cdot 1 = \frac{P(x_1, x_2 \mid \text{spam}) \cdot P(\text{spam})}{P(x_1, x_2)} = \frac{(0.18)(0.2)}{0.0376}$$

$$= 0.9574$$

$$\text{Risk}_{\text{remove}} = P(\text{nonspam} \mid (x_1 = 1, x_2 = 0)) \cdot 500 = \frac{P(x_1, x_2 \mid \text{non-spam}) \cdot P(\text{non-spam})}{P(x_1, x_2)} = \frac{(0.002)(0.8)}{0.0376}$$

$$= 21.28$$

$\therefore \quad (x_1 = 1, x_2 = 0) \Rightarrow \text{Risk}_{\text{keep}} < \text{Risk}_{\text{remove}} \Rightarrow$ keep the email

Homework 2

(1) Expected Loss & Bayes Optimality

c) iii) $(x_1 = 0, x_2 = 1)$

$$P(x_1 = 0, x_2 = 1) = 0.2(0.25) + 0.8(0.002) = 0.0516$$

$$\text{Risk}_{\text{keep}} = P(\text{spam} \mid (x_1 = 0, x_2 = 1)) \cdot 1 = \frac{P(x_1, x_2 \mid \text{spam}) \cdot P(\text{spam})}{P(x_1, x_2)} = \frac{0.2(0.25)}{0.0516}$$

$$= 0.9690$$

$$\text{Risk}_{\text{remove}} = P(\text{nonspam} \mid (x_1 = 0, x_2 = 1)) \cdot 500 = \frac{P(x_1, x_2 \mid \text{non-spam}) \cdot P(\text{non-spam})}{P(x_1, x_2)} = \frac{0.8(0.002)}{0.0516}$$

$$= 15.50$$

$\therefore \quad (x_1 = 0, x_2 = 1) \Rightarrow \text{Risk}_{\text{keep}} < \text{Risk}_{\text{remove}} \Rightarrow$ keep the email

Homework 2

(1) Expected Loss & Bayes Optimality

c) iv) $(x_1 = 1, x_2 = 1)$

$$P(x_1 = 1, x_2 = 1) = 0.2(0.12) + 0.8(0) = 0.024$$

$$\text{Risk}_{\text{keep}} = P(\text{spam} \mid (x_1 = 1, x_2 = 1)) \cdot 1 = \frac{P(x_1, x_2 \mid \text{spam}) \cdot P(\text{spam})}{P(x_1, x_2)} = \frac{0.2(0.12)}{0.024} = 1$$

$$\text{Risk}_{\text{remove}} = P(\text{nonspam} \mid (x_1 = 1, x_2 = 1)) \cdot 500 = \frac{P(x_1, x_2 \mid \text{non-spam}) \cdot P(\text{non-spam})}{P(x_1, x_2)} = \frac{0.8(0)}{0.024} = 0$$

$\therefore \quad (x_1 = 1, x_2 = 1) \Rightarrow \text{Risk}_{\text{keep}} > \text{Risk}_{\text{remove}} \Rightarrow$ remove the email

Homework 2

(1) Expected Loss & Bayes Optimality

d) $\mathbb{E}[\mathcal{T}(y_*, t)] = \sum_{(x_1, x_2)} P(x_1, x_2) \cdot \mathbb{E}[\mathcal{T}(y_*, t)]$

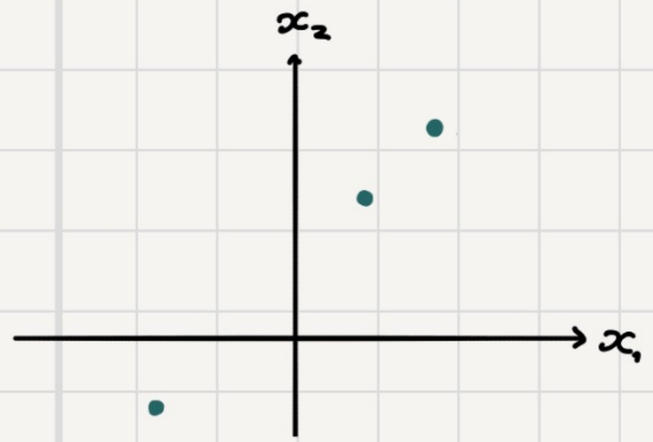
$= 0.8868\,(0.1015) + 0.0516(0.9690) + 0.0376(0.9574) + 0.024\,(0)$

$= 0.1760$

Homework 2

(2) Feature Maps

x_1	x_2	y
-2	-1	1
1	2	0
2	3	1



a) Let S be the set of points defined by the (x_1, x_2) pairs in the table above.

In order for this dataset to be linearly separable, there would need to exist some linear boundary to separate the points corresponding to $y = 1$ (i.e, $(-2,-1)$ and $(2,3)$) from the point corresponding to $y = 0$ (i.e, $(1,2)$).

A line divides the space into two convex regions (halfspaces). But the point labeled $y = 0$ lies exactly between the two $y = 1$ points, so any line that includes those two points will also include the $y = 0$ point. Therefore, no such separating line exists, and the dataset is not linearly separable

Homework 2

x_1	x_2	y
-2	-1	1
1	2	0
2	3	1

(2) Feature Maps

b) $z = w_1 x_1 + w_2 x_2^2$. $\quad y = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{otherwise} \end{cases}$

$(-2,-1) \rightarrow 1 \Rightarrow -2w_1 + w_2 \geq 0$

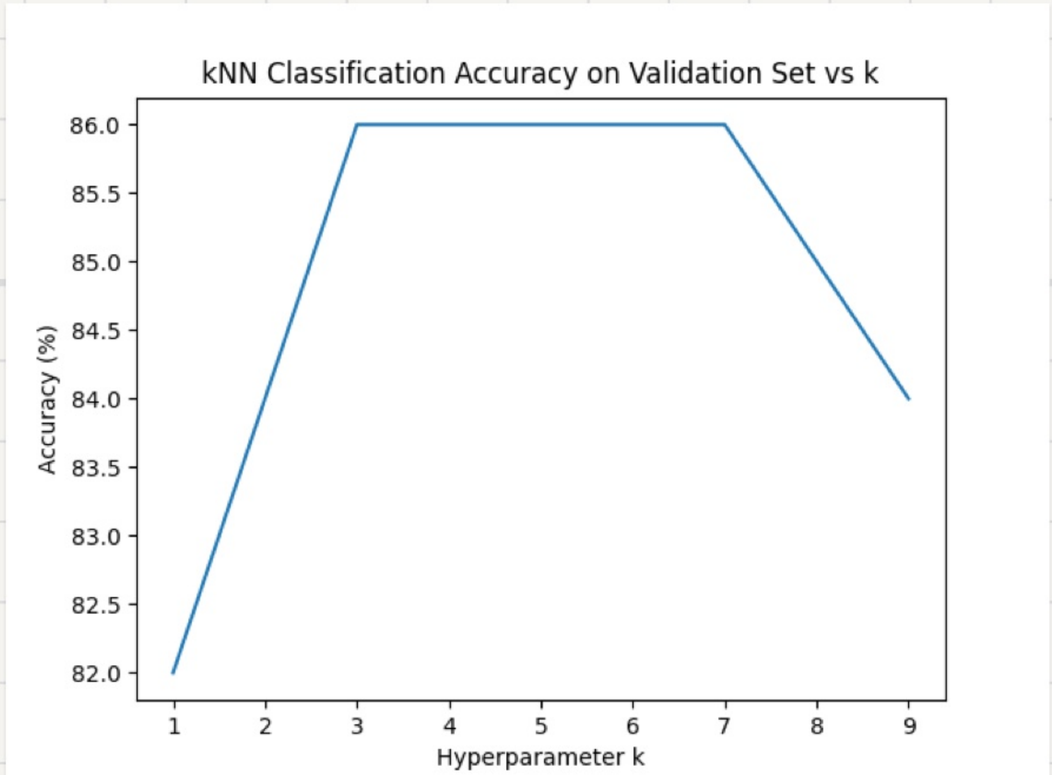
$(1,2) \rightarrow 0 \Rightarrow w_1 + 4w_2 < 0$

$(2,3) \rightarrow 1 \Rightarrow 2w_1 + 9w_2 \geq 0$

Homework 2

③ kNN vs Logistic Regression

a)



b) My choice is $k^* = 5$, since this value of k lies at the center of the plateau corresponding to maximum accuracy. Since k = 3, 5, 7 all achieve the same performance, choosing the middle value provides a conservative/balanced choice, reducing the risk of overfitting or underfitting without sacrificing accuracy.

k	Validation Acc.	Test Acc.
3	86.0%	92%
5	86.0%	94%
7	86.0%	94%

Overall, test accuracy improves slightly with higher k (5,7), while validation accuracy remains constant, suggesting stability in the model's generalization ability.

Homework 2

③ Logistic Regression

b) Best hyperparameters & final values of CE & Accuracy on all sets:

using small set

```
diff = 1.8555157593291063e-08
Best hyper-parameters: {'learning_rate': 0.05, 'weight_regularization': 0.0, 'num_iterations': 400}
Stopped at iteration 30
Train  CE = 0.1423, Acc = 1.0000
Valid  CE = 0.6724, Acc = 0.6200
Test   CE = 0.5435, Acc = 0.7400
```

using full set

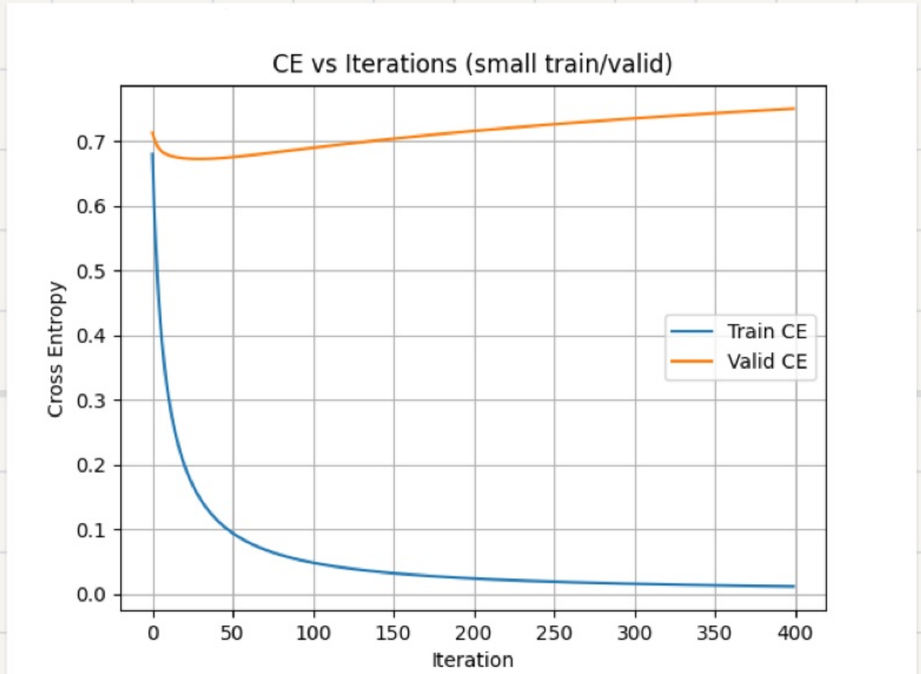
```
diff = 3.259058275280241e-08
Best hyper-parameters: {'learning_rate': 0.05, 'weight_regularization': 0.0, 'num_iterations': 400}
Stopped at iteration 399
Train  CE = 0.0808, Acc = 1.0000
Valid  CE = 0.2176, Acc = 0.8800
Test   CE = 0.2078, Acc = 0.9200
```

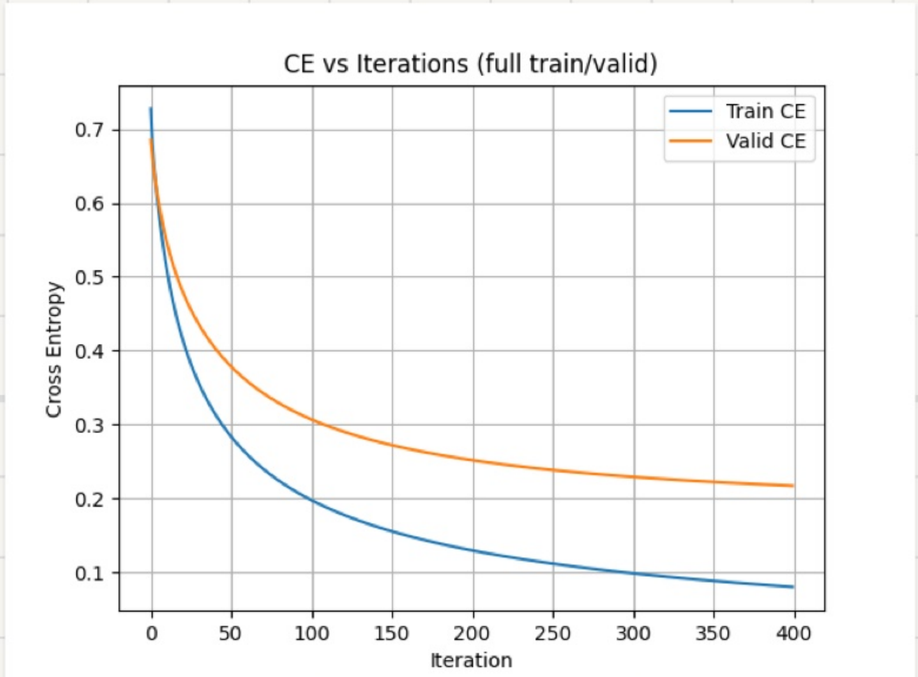
After selecting the best hyperparameters based on validation performance, the model was evaluated once on the test set. The final test CE and accuracy values shown above were computed at the best iteration for each run

Homework 2

3) Logistic Regression

c)





Although the weights are randomly initialized, I multiplied them all by a factor of 0.01 to reduce noise, which made the results very consistent across runs.

Homework 2

(4) Locally-Weighted Regression

a) In order to find the closed-form expression for w^*, we'll set the derivative of the loss function w.r.t w and set it to zero:

$$\frac{\partial J}{\partial w} = \frac{\partial}{\partial w}\left(\frac{1}{2}\sum_{i=1}^{N} a^{(i)}\left(y^{(i)} - w^T x^{(i)}\right)^2 + \frac{\lambda}{2}\|w\|^2\right) = 0$$

Writing this in matrix form, we have:

$$\frac{\partial J}{\partial w} = \frac{\partial}{\partial w}\left(\frac{1}{2}(y - Xw)^T A (y - Xw) + \frac{\lambda}{2} w^T w\right)$$

$$= -X^T A(y - Xw) + \lambda w = X^T A(Xw - y) + \lambda w$$

$$\Rightarrow X^T A(Xw - y) + \lambda w = 0$$

Homework 2

(4) Locally-Weighted Regression

a) From before:

$$X^T A(Xw - y) + \lambda w = 0$$

$$X^T AXw - X^T Ay + \lambda w = 0$$

$$(X^T AX + \lambda I)w - X^T Ay = 0$$

Solving for w:

$$w = w^* = (X^T AX + \lambda I)^{-1} X^T Ay$$

Homework 2

(4) Locally-Weighted Regression

d) • As $\tau \to \infty$, all training points are weighted equally, so locally weighted regression behaves like standard linear regression.

• As $\tau \to 0$, the single nearest training point dominates the prediction, making the model extremely sensitive to noise (prone to overfitting)

e) • Advantage: can capture local structure in the data better than traditional linear regression, making it better for datasets that aren't totally uniform, or only uniform in clusters.

• Disadvantage: it's computationally expensive at test time, since you need to fit a model to each test point.

Homework 2

(5) Loss Functions for Binary Classification

- **Squared Loss**
  a) Used mainly for regression
  b) Assumes errors follow a Gaussian distribution
  c) An advantage is that it's smooth and differentiable, has a closed-form solution in linear regression. A disadvantage is that it's sensitive to outliers, as it penalizes large errors far more strongly than small ones.

- **Mean Absolute Error Loss**
  a) Also used mainly for regression
  b) Assumes Laplace (double exponential) noise in the data
  c) An advantage is that it's not sensitive to outliers (doesn't overly penalize large errors); a disadvantage is that it's not differentiable at zero, so optimization can be harder and lacks a closed-form solution

Homework 2

(5) Loss Functions for Binary Classification

- **Binary Cross-Entropy Loss**
  a) Classification
  b) Assumes the model's outputs (y) are probabilities (i.e, $y \in [0,1]$) and the data is drawn from a Bernoulli Distribution
  c) An advantage is that it interprets probabilities well – i.e, it strongly penalizes confident wrong predictions, and rewards confident correct ones. A disadvantage is that it can be numerically unstable (e.g taking log(0)).

- **Huber Loss**
  a) Regression
  b) Assumes that errors follow a Gaussian distribution for the most part, but with occasional outliers
  c) An advantage is that it's robust to outliers, and that it's smooth and differentiable everywhere; a disadvantage is that it may be less optimal than pure MSE or MAE if the dataset is either very uniform or very noisy