Part-1: Basic Concepts

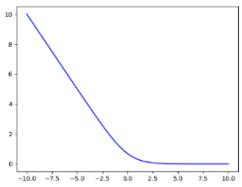
1. Cross-entropy loss for classification (18 points)

Notations for (1) and (2):
x_n is an input data sample, and it is a vector.
y_n is the ground-truth class label of x_n.
$\hat{y}_n$ is the output "soft-label"/confidence of a logistic regression classifier given the input x_n
The number of classes is K

(1: 1 point) Assume there are only two classes (K=2): class-0, class-1, and the data point x_n is in class-1 ($y_n = 1$).
Assume the output is $\hat{y}_n = 0.9$ from a binary logistic regression classifier.
Compute the binary cross-entropy loss associated with the single data sample x_n.
note: show the steps of your calculations. You will get zero point if only a number is shown.

(2: 1 point) Assume there are three classes (K=3): class-0, class-1 and class-2, and the data point x_n is in class-2 ($y_n = 2$). Assume the output is $\hat{y}_n = [0.01, 0.09, 0.9]^T$ from a multi-class logistic regression classifier. Do one-hot-encoding on y_n, and then Compute the cross-entropy loss associated with the single data sample x_n.
note: show the steps of your calculations. You will get zero point if only a number is shown.

(3: 2 points) Show that the function $f(x) = -log\left(\frac{1}{1+e^{-x}}\right)$ is convex in x, where log is the natural log .



Here is a plot of the function, and it seems that the function is convex.
Hint: show that $\frac{\partial^2 f}{\partial x^2} \geq 0$ then it is convex.
Note: show the steps of your calculations

(4: 2 points) Explain why cross entropy loss is convex with respect to the parameters of a logistic regression classifier.
Note: a few bullet points are just fine, and you may use anything in the lecture notes.

(5: 5 points) Let **L** be the cross entropy loss of a logistic regression classifier for binary-class classification, and let $\hat{y}$ be the scalar output of the classifier for an input sample x. $\hat{y} = \frac{1}{1+e^{-z}}$ and $z = w^T x + b$. Compute the derivative $\frac{\partial \boldsymbol{L}}{\partial z}$
note: show the steps of your calculations. You will get zero point if only the result is shown.

(6: 7 points) Let **L** be the cross entropy loss of a logistic regression classifier for multi-class classification, and let $\hat{y}$ be the vector output of the classifier for an input sample x. $\hat{y} = softmax(\boldsymbol{z})$, where $\boldsymbol{z} = [z_1, \dots, z_K]^T$ and $z_k = w_k^T x + b_k$. Compute the derivative $\frac{\partial \boldsymbol{L}}{\partial \boldsymbol{z}}$, which is a vector.
note: show the steps of your calculations. You will get zero point if only the result is shown.

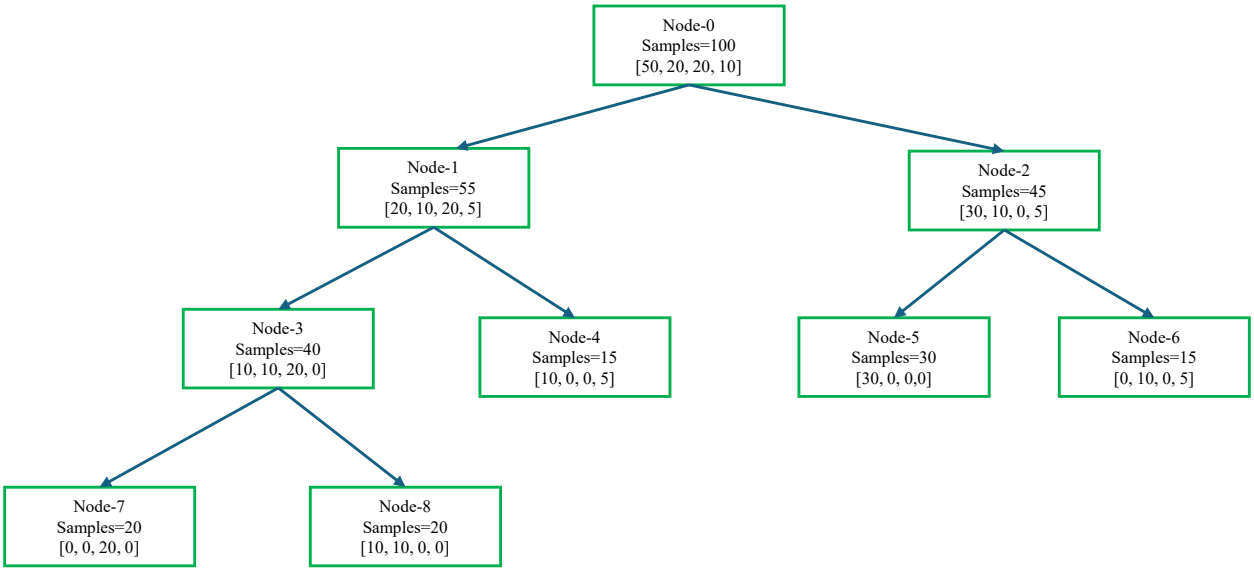
Note: this results of (5) and (6) are very useful when we apply the cross entropy loss to neural networks.

2. Regression with multiple outputs (3 points)

x_n is an input data sample, and it is a vector.
y_n is the ground-truth .
y_n is a vector and it has **two elements** $[y_{n,1}, y_{n,2}]$. For example, $y_{n,1}$ is income, and $y_{n,2}$ is age.
$\hat{y}_n$ is the output of a regressor (e.g., linear regressor) given the input x_n .
y_n is a vector and it has **two elements** $[\hat{y}_{n,1}, \hat{y}_{n,2}]$.
There are N data points.

(1: 1 point) write down the formula of MSE loss, using $y_{n,1}$, $y_{n,2}$, $\hat{y}_{n,1}$, $\hat{y}_{n,2}$, and N, where n is from 1 to N
(2: 1 point) write down the formula of MAE loss, using $y_{n,1}$, $y_{n,2}$, $\hat{y}_{n,1}$, $\hat{y}_{n,2}$, and N, where n is from 1 to N
(3: 1 point) write down the formula of MAPE loss, using $y_{n,1}$, $y_{n,2}$, $\hat{y}_{n,1}$, $\hat{y}_{n,2}$, and N, where n is from 1 to N
Note: the loss in (1)/(2)/(3) is the loss averaged across all of the samples (n is from 1 to N)

3. Decision Tree (5 points)
A decision tree is a partition of the input space.
Every leaf node of the tree corresponds to a region of the final partition of the input space.
(1)~(5) are related to the tree below:



(1: 1 point) What is the total number of training samples according to the above tree for classification?
(2: 1 point) What is the max-depth of the tree?
(3: 1 point) What is the entropy on Node-0? (using log base 2)
(4: 1 point) How many 'pure' nodes (entropy =0) does this tree have?
(5: 1 point) How many leaf/terminal nodes does this tree have?

4. Bagging and Random Forest (2 points)
(1: 1 point) Bagging will NOT work under a condition: what is this condition?
(2: 1 point) The trees in a random-forest is only weakly correlated in theory: why?

5. Boosting (2 points)
What is the difference between boosting (e.g., XGBoost) and bagging (e.g., Random-forest) from the perspective of variance and bias?

6. Stacking (2 points)
(1: 1 point) Could it be useful to stack many polynomial models of the same degree?
(2: 1 point) Could it be useful to stack models of different types/structures?

7. Overfitting and Underfitting (2 points)
It is easy to understand Overfitting and Underfitting, but it is hard to detect them.

Consider two scenarios in a classification task:
(1) the training accuracy is 100% and the testing accuracy is 50%
(2) the training accuracy is 80% and the testing accuracy is 70%
In which scenario is overfitting likely present? (1 point)

Consider two new scenarios in a classification task:
(1) the training accuracy is 80% and the testing accuracy is 70%
(2) the training accuracy is 50% and the testing accuracy is 50%
In which scenario is underfitting likely present? (1 point)

Keep in mind that, in real applications, the numbers in different scenarios may be very similar.
We can always increase model complexity to avoid underfitting.
We need to find the model with the "right" complexity (i.e., the best hyper-parameters) to reduce overfitting if possible.

8. Training, Validation, and Testing for Classification and Regression (3 points)
(1: 1 point) What are hyper-parameters of a model? Give some examples.
(2: 1 point) Why do we need a validation set? Why don't we just find the optimal hyper-parameters of a model on the training set? e.g., find the model that performs the best on the training set.
(3: 1 point) Why don't we optimize the optimal hyper-parameters of a model using the testing set?
Terminologies: training(train) set(dataset), testing(test) set(dataset), validation (val) set(dataset)

9. SVM (3 points)
(1: 1 point) Why maximizing the margin in the input space will improve classifier robustness against noises?
(2: 1 point) Will the margin in the original input space be maximized by a nonlinear SVM?
(3: 1 point) What is the purpose of using a kernel function in a nonlinear SVM?

10. Handle class-imbalance for classification tasks (2 points)
We have a class-imbalanced dataset, and the task is to build a classifier on this dataset. From the perspective of PDF, there are two types/scenarios of class-imbalance (see lecture notes). Now, assume we are in scenario-1.
(1: 1 point) Why do we use weighted-accuracy (a.k.a. balanced-accuracy) to measure the performance of a classifier? i.e., What is the problem of the standard accuracy?
(2: 1 point) When class-weight is not an option for a classifier, what other options do we have to handle class-imbalance?

11. Handle data-imbalance for regression tasks (2 points)
For regression tasks, is there an issue similar to class-imbalance ? If so, describe the issue and list some possible methods to handle this issue. (read http://dir.csail.mit.edu/)

12. Entropy (6 points)
The PMF for a discrete random variable X is $[p_1, p_2, p_3, \dots p_K]$ where $\sum_k p_k = 1$ and $0 \leq p_k \leq 1$
Write down the entropy and prove that:
(1: 1 point) entropy is non-negative
(2: 5 points) entropy reaches the maximum when the PMF is a uniform distribution, i.e., $p_k = 1/K$
Hint: you can use Jensen's inequality or Lagrange Multiplier

13. KL Divergence for probability distributions of discrete random variables. (5 points)
There are two probability distributions for the same discrete random variable X:
Distribution P: $[p_1, p_2, p_3, \dots p_K]$ where $\sum_k p_k = 1$ and $0 \leq p_k \leq 1$
Distribution Q: $[q_1, q_2, q_3, \dots q_K]$ where $\sum_k q_k = 1$ and $0 \leq q_k \leq 1$
The KL Divergence measures the difference between P and Q, and it is defined as

$$D_{KL}(P||Q) = \sum_{k=1}^{K} p_k log \frac{p_k}{q_k}$$

(1: 2 points) prove that the KL Divergence is non-negative
Hint: you can use Jensen's inequality
(2: 3 points) show that the KL Divergence is equivalent to cross-entropy when the distribution P is known and fixed
Hint: read lecture notes

Part-2: Programming on classification and regression

Read the instructions in H3P2T1.ipynb, H3P2T2.ipynb, H3P2T3.ipynb

Grading: (points for each question/task)

	Undergraduate Student	Graduate Student
Question 1	18	18
Question 2	3	3
Question 3	5	5
Question 4	2	2
Question 5	2	2
Question 6	2	2
Question 7	2	2
Question 8	3	3
Question 9	3	3
Question 10	2	2
Question 11	2	2
Question 12	Bonus (6 points)	6
Question 13	NA	Bonus (5 points)
H3P2T1	25	25
H3P2T2	21	15
H3P2T3	10	10
Total	100 + 6	100 + 5

Attention:

If you use test sets for optimizing any model, you will get zero score.

Make sure you run each and every code cell of your program files. If you do not run a code cell, you will lose the points of that cell.

LLM (e.g. ChatGPT) may give you wrong answers.

Homework 3

Sloan Atkins
CSC 546
Fall 25

(1: 1 point) Assume there are only two classes (K=2): class-0, class-1, and the data point x_n is in class-1 ($y_n = 1$). Assume the output is $\hat{y}_n = 0.9$ from a binary logistic regression classifier. Compute the binary cross-entropy loss associated with the single data sample x_n.
note: show the steps of your calculations. You will get zero point if only a number is shown.

$$L = -[y_n \log(\hat{y}_n) + (1 - y_n)\log(1 - \hat{y}_n)] \qquad y_n = 1$$
$$= -[1 \cdot \log(0.9) + (1-1)\log(1-0.9)]$$
$$= -[\log(0.9) + 0]$$
$$= -\log(0.9)$$

$$\log(0.9) = -0.10536$$

$$L = -(-0.10536) = 0.10536$$

(2: 1 point) Assume there are three classes (K=3): class-0, class-1 and class-2, and the data point x_n is in class-2 ($y_n = 2$). Assume the output is $\hat{y}_n = [0.01, 0.09, 0.9]^T$ from a multi-class logistic regression classifier. Do one-hot-encoding on y_n, and then Compute the cross-entropy loss associated with the single data sample x_n.
note: show the steps of your calculations. You will get zero point if only a number is shown.

$y_n = 2$, one-hot encoding: $y_n [0, 0, 1]^T$

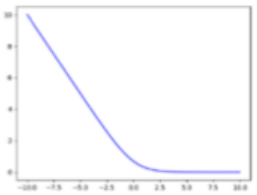
$$L = -\sum_{k=1}^{K} y_{n,k} \log(\hat{y}_{n,k})$$

$$= -[0 \cdot \log(0.01) + 0 \cdot \log(0.09) + 1 \cdot \log(0.9)]$$
$$= -\log(0.9)$$

$$\log(0.9) = -0.10536$$

$$L = -(-0.10536) = 0.10536$$

(3: 2 points) Show that the function $f(x) = -log\left(\frac{1}{1+e^{-x}}\right)$ is convex in x, where log is the natural log .



Here is a plot of the function, and it seems that the function is convex.
Hint: show that $\frac{\partial^2 f}{\partial x^2} \geq 0$ then it is convex.
Note: show the steps of your calculations

$$f(x) = -\log\left(\frac{1}{1+e^{-x}}\right)$$
$$= \log(1+e^{-x})$$

$$f'(x) = \frac{d}{dx}\log(1+e^{-x})$$
$$= \frac{1}{1+e^{-x}} \cdot (e^{-x})$$
$$= -\frac{e^{-x}}{1+e^{-x}}$$

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

$$f'(x) = -[1-\sigma(x)] = \sigma(x) - 1$$

$$f''(x) = \frac{d}{dx}(\sigma(x)-1) = \frac{d\sigma(x)}{dx}$$
$$\frac{d\sigma(x)}{dx} = \sigma(x)(1-\sigma(x))$$
$$f''(x) = \sigma(x)(1-\sigma(x))$$

Since $0 < \sigma(x) < 1$ for all x:
$\sigma(x)(1-\sigma(x)) > 0$
$f''(x) \geq 0$

$f(x) = -\log\left(\frac{1}{1+e^{-x}}\right)$ is convex in x

(4: 2 points) Explain why cross entropy loss is convex with respect to the parameters of a logistic regression classifier.
Note: a few bullet points are just fine, and you may use anything in the lecture notes.

Logistic regression model : $\hat{y} = \sigma(w^T x + b) = \frac{1}{1+e^{-(w^T x+b)}}$

Binary cross-entropy loss: $L = -[y \log(\hat{y}) + (1-y)\log(1-\hat{y})]$

$\Rightarrow L = \log(1+e^{-y(w^T x+b)})$

$f(z) = \log(1+e^{-z})$ is convex in z because its second derivative

$f''(z) = \sigma(z)(1-\sigma(z)) \geq 0$

Since $z = y(w^T x+b)$ is linear in the parameters w, b, and a convex function of a linear function is still convex,

$L(w, b)$ is convex with respect to the model parameters

(5: 5 points) Let **L** be the cross entropy loss of a logistic regression classifier for binary-class classification, and let $\hat{y}$ be the scalar output of the classifier for an input sample x. $\hat{y} = \frac{1}{1+e^{-z}}$ and $z = w^T x + b$. Compute the derivative $\frac{\partial L}{\partial z}$
note: show the steps of your calculations. You will get zero point if only the result is shown.

$$\frac{\partial L}{\partial \hat{y}} = -\frac{y}{\hat{y}} + \frac{1-y}{1-\hat{y}}$$

$$\frac{d\hat{y}}{dz} = \hat{y}(1-\hat{y})$$

$$\frac{dL}{dz} = \frac{dL}{d\hat{y}} \cdot \frac{d\hat{y}}{dz}$$

$$= \left(-\frac{y}{\hat{y}} + \frac{1-y}{1-\hat{y}}\right)\hat{y}(1-\hat{y})$$

$$= -y(1-\hat{y}) + (1-y)\hat{y}$$

$$= \hat{y} - y$$

$$\frac{\partial L}{\partial z} = \hat{y} - y$$

(6: 7 points) Let **L** be the cross entropy loss of a logistic regression classifier for multi-class classification, and let $\hat{y}$ be the vector output of the classifier for an input sample x. $\hat{y} = softmax(\boldsymbol{z})$, where $\boldsymbol{z} = [z_1, \dots, z_K]^T$ and $z_k = w_k^T x + b_k$. Compute the derivative $\frac{\partial \boldsymbol{L}}{\partial \boldsymbol{z}}$, which is a vector.
note: show the steps of your calculations. You will get zero point if only the result is shown.

Note: this results of (5) and (6) are very useful when we apply the cross entropy loss to neural networks.

$$\frac{\partial L}{\partial \hat{y}_k} = -\frac{y_k}{\hat{y}_k}$$

$$\frac{\partial \hat{y}_k}{\partial z_i} = \begin{cases} \hat{y}_k(1-\hat{y}_k) & \text{if } i = k \\ -\hat{y}_k \hat{y}_i & \text{if } i \neq k \end{cases}$$

$$\frac{\partial L}{z_i} = \sum_{k=1}^{K} \frac{\partial L}{\partial \hat{y}_k} \cdot \frac{\partial \hat{y}_k}{\partial z_i}$$

$$= \sum_{k=1}^{K} \left(-\frac{y_k}{\hat{y}_k}\right) \frac{\partial \hat{y}_k}{\partial z_i}$$

$$\frac{\partial L}{\partial z} = \hat{y} - y$$

2. Regression with multiple outputs (3 points)
x_n is an input data sample, and it is a vector.
y_n is the ground-truth .
y_n is a vector and it has **two elements** $[y_{n,1}, y_{n,2}]$. For example, $y_{n,1}$ is income, and $y_{n,2}$ is age.
$\hat{y}_n$ is the output of a regressor (e.g., linear regressor) given the input x_n .
y_n is a vector and it has **two elements** $[\hat{y}_{n,1}, \hat{y}_{n,2}]$.
There are N data points.

(1: 1 point) write down the formula of MSE loss, using $y_{n,1}$, $y_{n,2}$, $\hat{y}_{n,1}$, $\hat{y}_{n,2}$, and N, where n is from 1 to N
(2: 1 point) write down the formula of MAE loss, using $y_{n,1}$, $y_{n,2}$, $\hat{y}_{n,1}$, $\hat{y}_{n,2}$, and N, where n is from 1 to N
(3: 1 point) write down the formula of MAPE loss, using $y_{n,1}$, $y_{n,2}$, $\hat{y}_{n,1}$, $\hat{y}_{n,2}$, and N, where n is from 1 to N
Note: the loss in (1)/(2)/(3) is the loss averaged across all of the samples (n is from 1 to N)

1) $$L_{MSE} = \frac{1}{N} \sum_{n=1}^{N} \left[(y_{n,1} - \hat{y}_{n,1})^2 + (y_{n,2} - \hat{y}_{n,2})^2 \right]$$

2) $$L_{MAE} = \frac{1}{N} \sum_{n=1}^{N} \left[|y_{n,1} - \hat{y}_{n,1}| + |y_{n,2} - \hat{y}_{n,2}| \right]$$

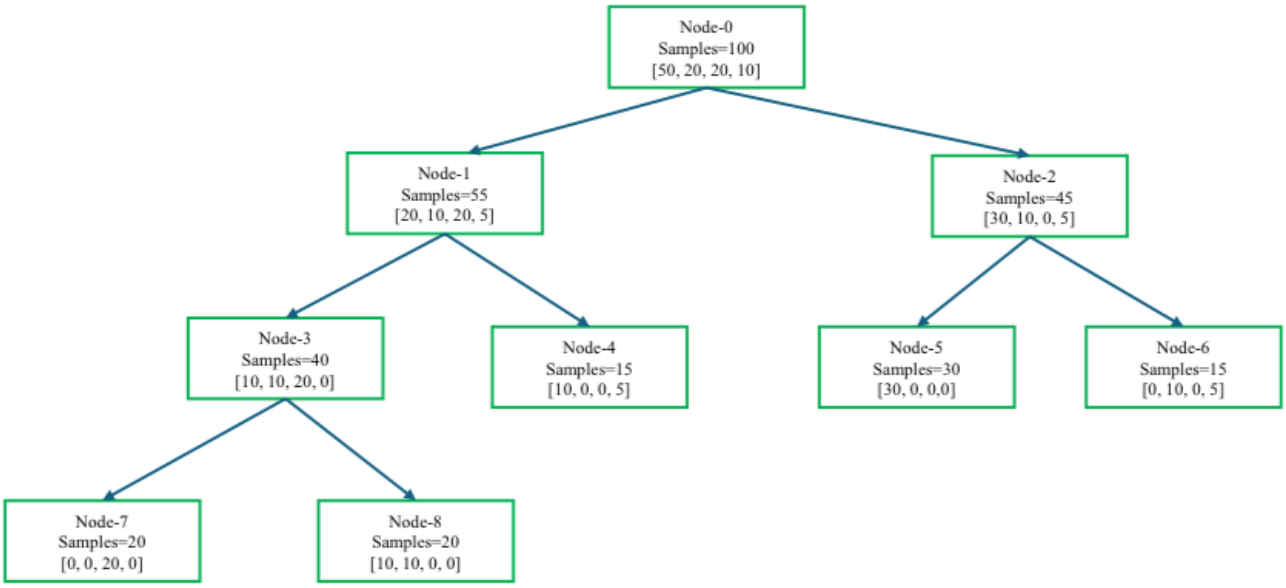
3) $$L_{MAPE} = \frac{100\%}{N} \sum_{n=1}^{N} \left[\frac{|y_{n,1} - \hat{y}_{n,1}|}{|y_{n,1}|} + \frac{|y_{n,2} - \hat{y}_{n,2}|}{|y_{n,2}|} \right]$$

3. Decision Tree (5 points)

A decision tree is a partition of the input space.

Every leaf node of the tree corresponds to a region of the final partition of the input space.

(1)~(5) are related to the tree below:



(1: 1 point) What is the total number of training samples according to the above tree for classification?

(2: 1 point) What is the max-depth of the tree?

(3: 1 point) What is the entropy on Node-0? (using log base 2)

(4: 1 point) How many 'pure' nodes (entropy =0) does this tree have?

(5: 1 point) How many leaf/terminal nodes does this tree have?

1) Root node (Node-0) : Samples = 100

2) Node 0 → Node 1 → Node 3 → Node 7 is the longest path

4 levels

↳ Max depth = 4

3) Node-0 distribution: [50, 20, 20, 10]

Total = 100

$p = [0.5, 0.2, 0.2, 0.1]$

$H = -\sum_i p_i \log_2(p_i)$

$= -(0.5 \log_2 0.5 + 0.2 \log_2 0.2 + 0.2 \log_2 0.2 + 0.1 \log_2 0.1)$

$= -(0.5(-1) + 0.2(-2.322) + 0.2(-2.322) + 0.1(-3.322))$

$= 0.5 + 0.464 + 0.464 + 0.332 = 1.76$ bits

4) Node-7 → [0, 0, 20, 0] pure
Node-8 → [10, 10, 0, 0]
Node-4 → [10, 0, 0, 5]
Node-5 → [30, 0, 0, 0] pure
Node-6 → [0, 10, 0, 5]

2 pure nodes (7 & 5)

5) Leaves: Node 4, 5, 6, 7, 8

5 leaf nodes

4. Bagging and Random Forest (2 points)
(1: 1 point) Bagging will NOT work under a condition: what is this condition?
(2: 1 point) The trees in a random-forest is only weakly correlated in theory: why?

1) Condition:
It relies on model diversity - if all learners are the same, averaging doesn't reduce variance

2) Each tree is trained on a different bootstrap sample (random subset of data)
At each split, only a random subset of features is considered
↳ This randomness decorrelates the trees

5. Boosting (2 points)
What is the difference between boosting (e.g., XGBoost) and bagging (e.g., Random-forest) from the perspective of variance and bias?

Bagging (Random Forest) = reduces variance by averaging many independent models

Boosting (XGBoost) = reduces bias by combining weak learns sequentially, each correcting the previous one's errors

6. Stacking (2 points)
(1: 1 point) Could it be useful to stack many polynomial models of the same degree?
(2: 1 point) Could it be useful to stack models of different types/structures?

1) Stacking many polynomials models of the same degree is not useful. They will have similar behavior and add no diversity

2) Stacking different model types/structures is useful. It combines models that capture different data patterns, improving overall performance

7. Overfitting and Underfitting (2 points)
It is easy to understand Overfitting and Underfitting, but it is hard to detect them.

Consider two scenarios in a classification task:
(1) the training accuracy is 100% and the testing accuracy is 50%
(2) the training accuracy is 80% and the testing accuracy is 70%
In which scenario is overfitting likely present? (1 point)

Consider two new scenarios in a classification task:
(1) the training accuracy is 80% and the testing accuracy is 70%
(2) the training accuracy is 50% and the testing accuracy is 50%
In which scenario is underfitting likely present? (1 point)

Overfitting is more likely in scenario 1 because model fits training data too closely and fails to generalize

Underfitting is more likely in scenario 2 because model is too simple to learn underlying patterns

8. Training, Validation, and Testing for Classification and Regression (3 points)
(1: 1 point) What are hyper-parameters of a model? Give some examples.
(2: 1 point) Why do we need a validation set? Why don't we just find the optimal hyper-parameters of a model on the training set? e.g., find the model that performs the best on the training set.
(3: 1 point) Why don't we optimize the optimal hyper-parameters of a model using the testing set?
Terminologies: training(train) set(dataset), testing(test) set(dataset), validation (val) set(dataset)

1) Hyperparameters are parameters set before training that control behavior
Examples: learning rate, number of epochs, number of neighbors, max depth in decision trees, regularization strength

2) We need a validation set to tune hyperparameters because:
- Choosing hyperparameters that perform best on the training set causes overfitting
- The validation set helps estimate model performance on unseen data for fair tuning

3) We don't optimize using the testing set because:
- The test set is meant for final evaluation only
- Using it for optimization would cause data leakage and yield overly optimistic results

9. SVM (3 points)
(1: 1 point) Why maximizing the margin in the input space will improve classifier robustness against noises?
(2: 1 point) Will the margin in the original input space be maximized by a nonlinear SVM?
(3: 1 point) What is the purpose of using a kernel function in a nonlinear SVM?

1) Maximizing the margin in input space improves robustness because a larger margin means the classifier can better tolerate noise or small perturbations in the data

2) The margin in the original input space is not necessarily maximized by a nonlinear SVM, it's maximized in the feature space created by the kernal function

3) The kernal function allows SVM to map non-linearly seperable data into a higher-dimensional feature space where a linear seperator can be found

10. Handle class-imbalance for classification tasks (2 points)
We have a class-imbalanced dataset, and the task is to build a classifier on this dataset. From the perspective of PDF, there are two types/scenarios of class-imbalance (see lecture notes). Now, assume we are in scenario-1.
(1: 1 point) Why do we use weighted-accuracy (a.k.a. balanced-accuracy) to measure the performance of a classifier? i.e., What is the problem of the standard accuracy?
(2: 1 point) When class-weight is not an option for a classifier, what other options do we have to handle class-imbalance?

1) Because standard accuracy is biased toward the majority class and weighted accuracy ensures each class contributes equally to the overall score

2) Options:
- Oversampling (duplicate or generate synthetic minority samples with SMOTE)
- Undersampling (reduce majority samples)
- Ensemble methods designed for imbalance (Balanced Random Forest)

11. Handle data-imbalance for regression tasks (2 points)
For regression tasks, is there an issue similar to class-imbalance ? If so, describe the issue and list some possible methods to handle this issue. (read http://dir.csail.mit.edu/)

Yes, regression can have a target imbalance issue - where some ranges of output values are more frequent than others causing poor performance for rare target values

Methods to handle it:
- Reweight samples to emphasize rare target ranges
- Oversample or undersample based on target distribution
- Apply data transformation
- Use quantile regression or stratified binning across target ranges

12. Entropy (6 points)

The PMF for a discrete random variable X is $[p_1, p_2, p_3, \dots p_K]$ where $\sum_k p_k = 1$ and $0 \le p_k \le 1$

Write down the entropy and prove that:

(1: 1 point) entropy is non-negative

(2: 5 points) entropy reaches the maximum when the PMF is a uniform distribution, i.e., $p_k = 1/K$

Hint: you can use Jensen's inequality or Lagrange Multiplier

Entropy formula: $H(X) = -\sum_{k=1}^{K} p_k \log(p_k)$

1) Non-negativity:

Since $0 \le p_k \le 1$ and $\log(p_k) \le 0$, each term $-p_k \log(p_k) \ge 0$

$\Rightarrow H(X) \ge 0$

2) Using Lagrange multipliers with constraint $\sum_i p_k = 1$:

$$\frac{\partial}{\partial p_k}\left[-\sum_i p_k \log(p_k) + \lambda\left(\sum_i p_k - 1\right)\right] = 0$$

$\Rightarrow -(\log p_k + 1) + \lambda = 0$

$\Rightarrow p_k = e^{\lambda - 1}$

Because all p_k are equal, $p_k = \frac{1}{K}$

$\hookrightarrow H_{max} = -K\left(\frac{1}{K}\log\frac{1}{K}\right) = \log K$

$\therefore$ Entropy reaches its maximum for a uniform distribution

13. KL Divergence for probability distributions of discrete random variables. (5 points)

There are two probability distributions for the same discrete random variable X:

Distribution P: $[p_1, p_2, p_3, \dots p_K]$ where $\sum_k p_k = 1$ and $0 \le p_k \le 1$

Distribution Q: $[q_1, q_2, q_3, \dots q_K]$ where $\sum_k q_k = 1$ and $0 \le q_k \le 1$

The KL Divergence measures the difference between P and Q, and it is defined as

$$D_{KL}(P||Q) = \sum_{k=1}^{K} p_k log \frac{p_k}{q_k}$$

(1: 2 points) prove that the KL Divergence is non-negative
Hint: you can use Jensen's inequality

(2: 3 points) show that the KL Divergence is equivalent to cross-entropy when the distribution P is known and fixed
Hint: read lecture notes

$$D_{KL}(P\|Q) = \sum_{K=1}^{K} P_K \log \frac{P_K}{q_K}$$

1) By Jensen's inequality, $D_{KL}(P\|Q) \geq 0$ with equality only when $P_K = q_K$ for all K

2) Cross-entropy is $H(P, Q) = -\sum_K P_K \log(q_K)$

and entropy is $H(P) = -\sum_K P_K \log(P_K)$

Then: $D_{KL}(P\|Q) = H(P, Q) - H(P)$

When P is fixed, minimizing D_{KL} is equivalent to minimizing cross-entropy