Part-1: Basic Concepts

1. Classification and Cross-entropy loss

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
n is from 1 to N
the number of classes is K

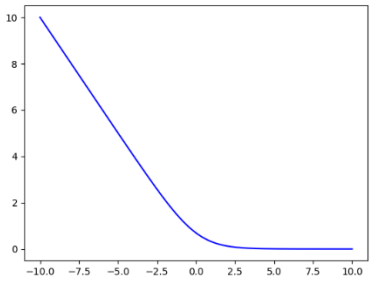
(1) write down the formula of binary cross-entropy (K=2), assuming y_n is an integer

(2) write down the formula of cross-entropy (K≥2), assuming y_n is a one-hot vector.

(3) Assume there are only two classes: class-0, class-1, and the data point x_n is in class-1 ($y_n = 1$)
Assume the output is $\hat{y}_n = 0.8$ from a binary logistic regression classifier
Compute the binary cross-entropy loss associated with the single data sample x_n
note: show the steps

(4) Assume there are three classes: class-0, class-1 and class-2, and the data point x_n is in class-2 ($y_n = 2$).
Assume the output is $\hat{y}_n = [0.1, 0.2, 0.7]^T$ 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

(5) Show that the function $f(x) = -log\left(\frac{1}{1+e^{-x}}\right)$ is convex in x. 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. This explains why cross entropy loss is convex.
The concept of cross entropy is from information theory

2. Regression

x_n is an input data sample, and it is a vector.
M is the number of elements/features in x_n.
y_n is the ground-truth
y_n is a vector that has **two elements** $[y_{n,1}, y_{n,2}]$. For example, $y_{n,1}$ is income, and $y_{n,2}$ is age, given input image x_n
$\hat{y}_n$ is the output of a regressor (e.g., linear regressor) given the input x_n
n is from 1 to N

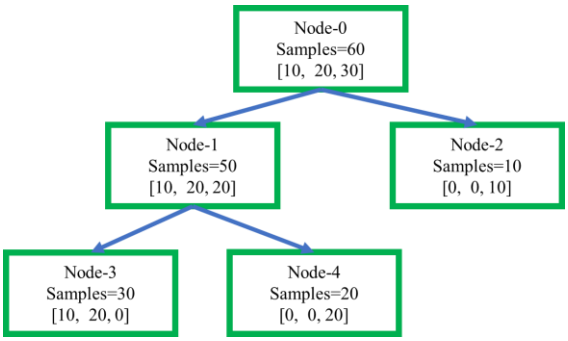
(1) write down the formula of MSE loss

(2) write down the formula of MAE loss

(3) write down the formula of MAPE loss

3. Decision Tree
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) The output of a decision tree for regression looks like a staircase. Why?
(2) Is it a good strategy to build a deep tree such that every leaf-node of the tree is a pure node?
(3)~(6) are related to the tree below



(3) What is the total number of training samples according to the above tree?
(4) What is the max-depth of the tree?
(5) How many 'pure' nodes (entropy =0) does this tree have? What is the entropy on Node-0?
(6) How many leaf/terminal nodes?

4. Bagging and Random Forest
(1) Under what condition will Bagging work?
(2) Random-forest uses a strategy to reduce the correlation between trees. What is the strategy?

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

6. Stacking
(1) What is the difference between stacking and bagging?
(2) Could it be useful to stack many polynomial models of the same degree?
(3) Could it be useful to stack models of different types/structures?

7. Overfitting and Underfitting
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?

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?

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

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

9. SVM

(1) Why maximizing the margin in the input space will improve classifier robustness against noises?
(2) Will the margin in the input space be maximized by a nonlinear SVM?
(3) Why could SVM cause "out-of-memory" error for a large dataset?
(4) What is the purpose of using a kernel function?

10. Handle class-imbalance

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) Why do we use weighted-accuracy to measure the performance of a classifier? i.e., What is the problem of the standard accuracy?
(2) When class-weight is not an option for a classifier, what other options do we have to handle class-imbalance?

11. Entropy

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

12. Joint Entropy and Mutual Information

For a pair of discrete random variables X and Y (scalars, not vectors) with the joint distribution $p(x, y)$, the joint entropy $H(X, Y)$ is defined as

$$H(X,Y) = -\sum_x \sum_y p(x,y) log(p(x,y))$$

which can also be expressed using mathematical expectation:

$$H(X,Y) = -E[log(p(x,y))]$$

The entropy of X (PDF is $p(x)$) is defined as

$$H(X) = -\sum_x p(x) log(p(x))$$

The entropy of Y (PDF is $p(y)$) is defined as

$$H(Y) = -\sum_y p(y) log(p(y))$$

note: $p(x)$ and $p(y)$ represent different PDFs.
The mutual information is defined as

$$I(X,Y) = \sum_x \sum_y p(x,y) log\left(\frac{p(x,y)}{p(x)p(y)}\right)$$

Jensen's inequality says: $E[f(X)] \geq f(E[X])$ where $f(x)$ is a convex function.

Prove that $H(X,Y) \leq H(X) + H(Y)$
Hint: show that $I(X,Y) = H(X) + H(Y) - H(X,Y)$, and then show $I(X,Y) \geq 0$ using Jensen's inequality.

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	5	5
Question 2	2	2
Question 3	6	6
Question 4	2	2
Question 5	2	2
Question 6	2	2
Question 7	2	2
Question 8	3	3
Question 9	3	2
Question 10	3	2
Question 11	Bonus (5 points)	2
Question 12	N.A.	Bonus (5 points)
H3P2T1	30	30
H3P2T2	30	30
H3P2T3	10	10

Attention:

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