CIS 4190/5190: Applied Machine Learning — Spring 2026

Homework 2

Handed Out: February 16 — *Due: March 2, 8 p.m.*

- You are encouraged to format your solutions using LaTeX. Handwritten solutions are permitted, but remember that you bear the risk that we may not be able to read your work and grade it properly — we will not accept post hoc explanations for illegible work. You will submit your solution manuscript for written HW 2 as a single PDF file.
- Familiarize yourself with the "Human and AI Assistance in Homework" Policy included in the administrivia slides from the first lecture. https://www.seas.upenn.edu/~cis5190/fall2025/schedule.html
- The homework is **due at 8 PM** on the due date. We will be using Gradescope for collecting the homework assignments. Please submit your solution manuscript as a PDF file via Gradescope. Post on Ed Discussion and contact the TAs if you are having technical difficulties in submitting the assignment.
- Make sure to assign pages to each question when submitting homework to Gradescope. The TA may deduct 0.2 points per sub-question if a page is not assigned to a question.
- Items marked [**5190 Only**] are mandatory for students enrolled in CIS 5190, and optional for CIS 4190. More information on the administrivia slides.

1 Written Questions

Note: You do not need to show work for multiple choice questions. If formatting your answer in LaTeX, use our LaTeX template hw_template.tex (This is a read-only link. You'll need to make a copy before you can edit. Make sure you make only private copies.).

1. **[Logistic Regression] (8 pts)**

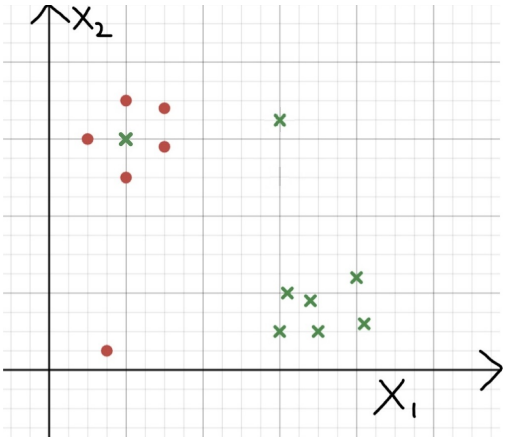

Figure 1: Data for Logistic Regression Question

Let the data distribution, as shown in Figure1, represent the binary classification problem where we fit the model $p(y = 1|x, \beta) = \sigma(\beta_0 + \beta_1 x_1 + \beta_2 x_2)$. As seen in class, we do this by minimizing the negative log loss (same as maximizing the likelihood), as shown below:

$$L(\beta) = -\ell_Z(\beta)$$

where $\ell_Z(\beta)$ represents the log likelihood on the training set Z.

For the questions below, submit the answer to each question as a separate figure. We just expect an approximation of the figures if you submit hand-drawn solutions, also be careful about the clarity of your submitted figures.

(a) (2 pts) Show a decision boundary that possibly would correspond to $\hat{\beta}$ (final weights) after training the regressor. How many datapoints are wrongly classified on the training data?

(b) (2 pts) For this part, consider that a strong regularization is applied to the β_0 parameter and we minimize

$$L_0(\beta) = -\ell_Z(\beta) + \lambda\beta_0^2$$

Assume that λ is effectively infinite, so, β_0 is pulled down all the way to 0, but all other parameters are unregularized. Draw a decision boundary that approximately corresponds to the optimal classifier with parameters $\hat{\beta}$. How many datapoints are wrongly classified on the training data?

(c) (2 pts) Now, heavy regularization is performed only on the β_1 parameter, i.e., we minimize

$$L_1(\beta) = -\ell_Z(\beta) + \lambda\beta_1^2$$

Show a decision boundary that possibly would correspond to $\hat{w}$. How many datapoints are wrongly classified on the training data?

(d) (2 pts) Finally, heavy regularization is done only on the β_2 parameter. Show a decision boundary that possibly would correspond to $\hat{\beta}$. How many datapoints are wrongly classified on the training data?

2. **[Neural Networks] (6 pts)**

a. (1 pts) What happens when we initialize all weights zero in Neural Networks?

b. (1 pts) Why do we use a non-linear activation function in a feed forward neural network?

c. (4 pts) Consider a fully connected neural network with the following architecture:
Input layer: 40×30 grayscale image
Hidden layer 1: 64 units
Hidden layer 2: 32 units
Output layer: 10 units

Assume biases are included in the network. Calculate the total number of trainable parameters in this network.

3. [k **Nearest Neighbors**] **(6 pts)**

   Consider properties of kNN models:

   a. (2 pts) Suppose we are using 1-nearest neighbor (kNN with $k = 1$) for binary classification under Euclidean distance.

      i. If the training set contains only two points with different labels, describe the resulting decision boundary. [Hint: Try drawing a picture]

      ii. Now assume we have full control over the training dataset (and can even construct an infinitely large one). Can the classifier represent any possible decision boundary? If yes, explain how the dataset could be arranged so that the classifier exactly matches the desired boundary. If no, provide an example of a decision boundary that 1-NN cannot represent.

   b. (2 pts) Suppose we take $k \to n$, where n is the size of the dataset; what would kNN output for a regression problem? What about for a classification problem?

   c. (2 pts) What effect does increasing the number of nearest neighbors k have on the bias-variance tradeoff? Explain your answer. [Hint: Use parts (a) and (b) in your explanation.]

4. [k **Nearest Neighbors**] **(5pts)** Imagine you want to apply k Nearest Neighbors to the binary classification problem of predicting whether a house is worth more than \$600, 000 (y = +1) or less than \$600, 000 (y = -1). Suppose you have two features: the square footage of the house, and the distance of the house from the nearest school (in miles). Most houses in the dataset are located in residential neighborhoods with schools nearby.

   a. (2 pts) Why might Euclidean distance between feature vectors representing houses be a bad metric of similarity / dissimilarity? What might you do to solve this problem?

   b. (1 pts) Suppose you found another potential feature: the type of roof the house has, expressed as a categorical feature via integer IDs (i.e. 0 = shingles, 1 = padded, etc.). Is it a good idea to use this feature alongside the other two features, with Euclidean distance? Explain why or why not.

   c. (2pts) Briefly describe the curse of dimensionality. Why can k-NN still perform well on some datasets like the handwritten digits, despite these images being very high dimensional?

5. **[Decision Trees] (8 pts)**

   In class, we discussed early stopping of generating splits and post-pruning. Here, we consider how they interact.

a. (2 pts) Decision tree training often combines early stopping criteria (such as maximum depth or minimum samples per split) with post-pruning on a validation set. If post-pruning can remove unnecessary branches, why might it still be beneficial for model performance to apply early stopping during tree construction?

b. (4 pts) Suppose we are training a decision tree with both early-stopping conditions and post-pruning. For each of the following, describe how it affects bias and variance: (i) increase the maximum depth of the decision tree, (ii) increase the minimum number of samples needed to split, (iii) disable post-pruning, and (iv) assuming we are using a feature map, add more features to the feature map.

c. (2 pts) For the given boolean functions, please provide the corresponding decision tree representations. Here, $\wedge$ denotes logical *AND* and $\vee$ denotes logical *OR*.

   i. $A \wedge (B \vee C)$

   ii. $(A \wedge B) \vee (C \wedge D)$

6. **[Decision Trees] (17 pts)** Consider the following set of training examples for a decision tree classifier: [Hint: a1, a2 are attributes, e.g. High Income? Y/N]

Instance	a1	a2	Classification
1	-	+	F
2	-	-	T
3	+	+	T
4	+	+	T

Recall the following definitions of entropy (H), Information Gain, and Gini index, which are useful for this problem:

$$H(\mathcal{D}) = -\sum_c P(Y=c)\,\log_2 P(Y=c),$$

$$\mathrm{IG}(\mathcal{D}, X_j) = H(\mathcal{D}) \;-\; \sum_v P(X_j=v)\; H\big(\mathcal{D}[X_j=v]\big),$$

$$\mathrm{Gini}(\mathcal{D}) = \sum_c P(Y=c)\big(1-P(Y=c)\big) \;=\; 1-\sum_c P(Y=c)^2.$$

a. (2 pts) What is the class entropy of this collection of training examples? Please write your final answer in decimal form.

b. (6 pts) What is the information gain of the two attributes respectively relative to these training examples? Please write your final answer in decimal form.

c. (3 pts) Draw the complete decision tree, showing the class predictions at the leaves.
Assuming you are using LaTeX, you may (i) very neatly hand draw the tree, photograph it, and include it as a figure, (ii) draw it using a graphics program or PowerPoint, (iii) express the tree in a series of if statements, preferably using LaTeX's `verbatim` environment, or (iv) typeset the tree directly with TikZ.

d. (1 pts) Given this decision tree, what would be the classification for a new instance with:a1 = +, a2 = -?

e. (5 pts) **[5190 ONLY]** What is the Gini gain of each of the two attributes respectively relative to these training examples? Be sure to clearly show your work, including for Gini index of parent and weighted average Gini index of children.