

# Winter 2026 DSC 240: Introduction to Machine Learning

## Homework 3

Due: Monday, Feb 23th, 11:59 pm PST

---

### Notes:

1. **This assignment is to be done individually.** You may discuss the problems at a general level with others in the class (e.g., about the concepts underlying the question, or what lecture or reading material may be relevant), but the work you turn in must be solely your own.
2. Be aware of the late policy in the course syllabus – i.e., *no late days for all the assignments*, so **it is your responsibility to turn in your assignment to Gradescope by the due time.**
3. Justify every answer you give – **show the work** that achieves the answer or **explain** your response.
4. Any updates or corrections will be posted on Piazza, so check there occasionally.
5. It is recommended to typeset the homework in L<sup>A</sup>T<sub>E</sub>X or markdown for the submission. However, neatly, hand-written homework are also allowed.
6. This assignment requires **two parts of the submission**. Both of these submissions will be available on Gradescope, and **both submissions are MANDATORY** for receiving full credit for the assignment:
  - (a) Written form of the submission as a **PDF** file format. This needs to be submitted in **Homework 3 - Report** on Gradescope.
  - (b) Code submission as a **.py** python script or a **.ipynb** jupyter notebook file. This needs to be submitted in **Homework 3 - Code** on Gradescope.
7. You can complete the assignment in a Jupyter notebook, which allows you to write both the text (using L<sup>A</sup>T<sub>E</sub>X for math) and the code in one file. Once done, convert the notebook to a PDF and submit it to the **Homework 3 - Report** on Gradescope. Also, submit the .ipynb notebook file (with the code) to the **Homework 3 - Code** on Gradescope. This is just a suggestion – you can also submit the report using L<sup>A</sup>T<sub>E</sub>X/Markdown/handwriting and the code in a separate **.py/.ipynb** file if you prefer.
8. Be sure to re-read the “**Academic Integrity**” on the course syllabus. You must complete the section below. If you answered Yes to either of the following two questions, give corresponding full details.

*Did you receive any help whatsoever from anyone in solving this assignment?*  
*Did you give any help whatsoever to anyone in solving this assignment?*

---

### Question 1 (10 points): Maximum Likelihood Estimation for Bernoulli Distribution

Let  $X_1, X_2, \dots, X_n$  be  $n$  independent and identically distributed (i.i.d.) random variables following a Bernoulli distribution. The Bernoulli distribution for a random variable  $X$  is given by:

$$\mathbb{P}(X = x; p) = p^x(1 - p)^{1-x}$$

where  $x \in \{0, 1\}$  and  $p$  is the probability of  $X$  being 1. Derive the maximum likelihood estimator (MLE) for the parameter  $p$  using the given samples.

### Question 2 (20 points): Cross-Entropy Loss and Logistic Loss

Consider a binary classification problem with outputs  $y \in \{0, 1\}$ . Assume  $p(y|x)$  is Bernoulli( $\hat{p}(x; \theta^*)$ ) for some parameter vector  $\theta^* \in \mathbb{R}^d$ , where for each  $\theta$ ,  $\hat{p}(x; \theta)$  returns a number between  $[0, 1]$ .

- (a) **(10 points)** For each parameter  $\theta$ , the cross-entropy loss for a set of  $n$  examples with predictions  $\hat{p}_i := \hat{p}(x_i; \theta)$  and true labels  $y_i$  is given by:

$$\text{CE}(\theta) = -\frac{1}{n} \sum_{i=1}^n (y_i \log(\hat{p}_i) + (1 - y_i) \log(1 - \hat{p}_i)).$$

Derive this loss function starting from the principle of Maximum Likelihood Estimation (MLE).

- (b) **(10 points)** Instantiate the classifier's prediction  $\hat{p}(x; \theta)$  as the sigmoid function applied to a linear combination of input features, i.e.,  $\hat{p}(x; \theta) = \sigma(\theta^\top x)$ , where  $\sigma(z) = \frac{1}{1+e^{-z}}$ . Show that in this case, minimizing the cross-entropy loss above is equivalent to minimizing the familiar logistic loss, which is given by:

$$L(\theta) = \frac{1}{n} \sum_{i=1}^n \log(1 + e^{-\tilde{y}_i \theta^\top x_i}) \quad \text{where } \tilde{y}_i = \begin{cases} 1 & \text{if } y_i = 1 \\ -1 & \text{if } y_i = 0 \end{cases}.$$

### Question 3 (40 + 10 bonus points): Shape of Bayes Optimal Classifiers for Gaussian Data

Assume we have a binary classification problem where:

- Data points from class A are generated from a Gaussian distribution  $\mathcal{N}(\mu_1, I_d)$ .
- Data points from class B are generated from a Gaussian distribution  $\mathcal{N}(\mu_2, I_d)$ .
- The prior probabilities for class A and class B are equal,  $P(\text{class A}) = P(\text{class B}) = 0.5$ .

- (a) **(10 points)** Derive the posterior probability  $P(y|x)$  for class A and B and the decision rule for the Bayes optimal classifier. Given  $\mu_1 = [0, 0]$  and  $\mu_2 = [1, 1]$ , plot the decision boundary and determine if it is linear.

(Hint: Recall that the Bayes optimal classifier  $h^*$  is given by  $h^*(x) = \arg \max_y p(y|x)$ . So the strategies for solving Part (a) and (b)(c)(d) is to first derive  $p(y|x)$  using the Bayes rule. )

- (b) **(10 points)** Consider the case where the prior probabilities are not equal,  $P(\text{class A}) = p$  and  $P(\text{class B}) = 1 - p$ . Derive the Bayes optimal classifier. For the 2D example, let  $\mu_1 = [0, 0]^T$ ,  $\mu_2 = [1, 1]^T$ , and  $p = 0.8$ . Plot the decision boundary.
- (c) **(10 points)** Now let the covariance matrix for class A be  $\sigma^2 I_d$  while keeping class B's covariance matrix as  $I_d$ . Derive the Bayes optimal classifier. For the 2D example, let  $\mu_1 = [0, 0]^T$ ,  $\mu_2 = [1, 1]^T$ ,  $p = 0.5$ , and  $\sigma^2 = 0.25$ . Plot the decision boundary.
- (d) **(10 points)** Instantiate the decision rule from (c) with  $\mu_1 = \mu_2$  what is the shape of the decision boundary? For the scenario where  $\mu_1 = [0, 0]^T$  and  $\mu_2 = [0, 0]^T$  with  $p = 0.5$  and  $\sigma^2 = 0.25$ , plot the decision boundary.
- (e) **(Bonus 10 points)** Examine whether the decision boundaries from parts (a)-(d) can be *realized* by a Gaussian Naive Bayes model. Provide a reasoning for your answer. (Hint: Whether the Bayes Optimal classifier can be *realized by*  $\mathcal{H}$  means whether  $h^* \in \mathcal{H}$ , i.e., whether there is a parameter vector of the Gaussian Naive Bayes classifier that is the same as  $h^*$ )

**Deliverables** for each subproblem (a) (b) (c) and (d):

1. Analytical expressions for  $P(y|x)$ .
2. The decision rule of the Bayes optimal classifier / analytical expression of the decision boundary.
3. Plotted figures that clearly demonstrate the decision boundary along with 100 random samples (color-coded in red and blue) from each class in the same figure generated in the specified way.

#### Question 4 (30 points): Naive Bayes Classifier vs Linear Classifier

In this question, we have provided you with three carefully designed binary classification datasets (`data-1.npy`, `data-2.npy`, and `data-3.npy`).

- (a) **(15 points)** For each of the three datasets provided, fit a Gaussian Naive Bayes classifier and a linear logistic regression classifier using the `sklearn` library in Python. Plot the dataset with points color-coded according to their true labels. Additionally, plot the decision boundaries for both classifiers.

(*Hint:* You may use `matplotlib` for plotting and `sklearn`'s `GaussianNB` and `LogisticRegression` for model fitting. The first two columns in the dataset represents the features and the last column is labels.)

- (b) **(15 points)** Discuss why the Naive Bayes classifier and / or the linear classifier fail on each example. Justify your answers.