CS 273P: Machine Learning and Data Mining

Homework 1

Due date: **Thursday, Jan 22, 2026 at 23:59**

Instructor: Xiaohui Xie

This homework (and most subsequent ones) uses **GitHub Classroom** and is graded automatically using **GitHub Actions**.

Submission rules (READ CAREFULLY):

1. Accept the assignment link on GitHub Classroom (posted on Canvas). This will create a private repository for you.
2. Clone your repository locally and complete all required code inside the provided Python scripts.
3. Submit by committing and pushing your code when you have finished. **Every push triggers the autograder.** Your test result will show up as "failed" unless you have completed all questions correctly. The last successful autograder score counts.
4. You do not need to submit anything to Gradescope or Canvas.
5. The repository should contain **all your code in its intended structures**, so that the autograder can evaluate correctness. If you modify the code structures that is not meant to be changed, such as Python file names or function names, the test will fail.

If you run into issues using GitHub Classroom, ask questions on Ed Discussion so others benefit from the discussion.

Summary: (1) Work in your GitHub Classroom repository; (2) Push code to trigger autograding; (3) Make sure all your functions run correctly in the provided files; (4) **No other submission needed.**

Points: This homework adds up to a total of **105 points**, as follows:

Problem 0: Get Connected	5 points
Problem 1: Python & Data Exploration	20 points
Problem 2: kNN Predictions	35 points
Problem 3: Naïve Bayes Classifiers	40 points
Statement of Collaboration	5 points

Problem 0: Get Connected (5 points)

Visit our class forum on Ed Discussion. This is where all class questions should be posted instead of emailing staff. Your participation is automatically tracked; nothing to submit.

Problem 1: Python & Data Exploration (20 points)

In this problem, we will explore some basic statistics and visualizations of an example data set. Function definitions for this problem are provided for you in `problem1.py`. Complete each function so that it returns the required values using the Iris dataset stored in the `data` folder. The Iris data consist of four real-valued features used to predict which of three types of iris flower was measured (a three-class classification problem).

1. `get_shape(X)` — return the number of data points and number of features. (5 points)
2. `feature_histograms(X)` — compute histogram counts for each feature and return them in any reasonable structure (e.g. list of arrays). (5 points)

3. `compute_stats(X)` — return mean and standard deviation for each feature. (5 points)
4. `scatter_pairs(X, Y)` — compute and return the coordinate pairs for (1,2), (1,3), (1,4), grouped by class. No plotting required; just return the values the scatter plot would use. (5 points)

You do not generate figures, you may look at visualizations if you want to; you only compute and return the numerical values.

Problem 2: kNN Predictions (35 points)

Complete the functions in `problem2.py` to implement a k-nearest neighbors classifier on CIFAR-10.

1. `load_cifar10()` — return the training and testing data. (5 points)
2. `compute_distances(X_train, X_test)` — compute pairwise distances between images. (10 points)
3. `predict_knn(dists, Y_train, k)` — return predicted labels for test data. (10 points)
4. `evaluate_accuracy(Y_pred, Y_true)` — compute and return accuracy for several k values (see template). (10 points)

Modify only the function bodies; the autograder will import your functions.

Problem 3: Naïve Bayes Classifiers (40 points)

For the binary-feature email dataset in `data/email_data.csv`, complete the following functions in `problem3.py`. $x_1, x_2, x_3, x_4, x_5, y$ stands for "know author?", "is long?", "has 'research'", "has 'lottery'", "read?" respectfully.

1. `estimate_nb_params(X, Y)` — compute $p(y)$ and $p(x_i \mid y)$ for each feature and class, and return them in a structured format. (10 points)
2. `predict_nb(x, params)` — return the predicted class for feature vector x using naive Bayes multiplicative likelihoods, breaking ties in favor of $+1$. Test your function on $x = (0, 0, 0, 0, 0)$ and $(1, 1, 0, 1, 0)$. (10 points)
3. `posterior_nb(x, params)` — compute and return $p(y = +1 \mid x)$ for $x = (1, 1, 0, 1, 0)$. (5 points)
4. `drop_feature_and_retrain(X,Y)` — remove the first feature, recompute model parameters, and return predictions before and after removing x_1. (10 points)
5. `compare_accuracy()` — return a string describing whether accuracy improves, degrades, or stays the same after removing x_1. (5 points)

All grading is based on return values; no printed explanations needed.

Statement of Collaboration (5 points)

Add your names and collaboration statement to `collaboration.txt` included in your repo. Do not share code. Follow course academic honesty rules. If you did not collaborate with anyone on this homework, say "No collaboration".