CS 273P: Machine Learning and Data Mining

Homework 1

Due date: **See Canvas**

Instructor: Xiaohui Xie

This homework (and most subsequent ones) will be graded automatically on **Gradescope** using an autograder.

Submission rules (READ CAREFULLY):

1. Download the homework starter files from Canvas.
2. Complete your work by editing only the three Python files: `problem1.py`, `problem2.py`, and `problem3.py`. Your functions must keep the original names and return the required values.
3. Create a plain text file called `statement.txt` containing your collaboration statement (see last page).
4. Submit your work to Gradescope by uploading a single **ZIP file** containing:
   - `problem1.py`
   - `problem2.py`
   - `problem3.py`
   - `statement.txt`

   No folders or subdirectories, **you must zip the files directly**, not zipping a folder containing them; all files must be at the top level of the zip; optionally, you may upload each file individually; **file names must match exactly**, you do not need to submit the dataset files.
5. After uploading, Gradescope will automatically run tests and report your score. You may resubmit as many times as you like before the deadline; your **highest score** counts.

If you run into issues with the autograder or submission format, post on Ed Discussion so others can benefit from the discussion.

Summary: (1) Edit `problem1.py`, `problem2.py`, `problem3.py`; (2) Add `statement.txt`; (3) Zip all four files and upload to Gradescope; (4) Autograder score determines your grade.

Points: This homework adds up to a total of **100 points**, as follows:

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 1: Python & Data Exploration (20 points)

In this problem, you will compute basic statistics and summaries of a small numerical data set. Function definitions for this problem are provided for you in `problem1.py`. Complete each function so that it returns the required values.

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)

No plotting is needed; the autograder evaluates return values only.

Problem 2: kNN Predictions (35 points)

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

1. `load_cifar10()` — return the training and testing data. (5 points)
2. `compute_distances(X_train, X_test)` — compute pairwise distances between samples. (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 accuracy for several k values (see template). (10 points)

Modify only the function bodies; the autograder imports your functions and checks return values.

Problem 3: Naïve Bayes Classifiers (40 points)

For the binary-feature email dataset provided to you, complete the following functions in `problem3.py`. $x_1, x_2, x_3, x_4, x_5, y$ represent: "know author?", "is long?", "has 'research'?", "has 'lottery'?", "read?", and label respectively.

1. `estimate_nb_params(X, Y)` — compute $p(y)$ and $p(x_i \mid y)$ for each feature and class. (10 points)
2. `predict_nb(x, params)` — return the predicted class for feature vector x. (10 points)
3. `posterior_nb(x, params)` — return $p(y = +1 \mid x)$ for a given feature vector. (5 points)
4. `drop_feature_and_retrain(X,Y)` — remove the first feature, recompute parameters, and return predictions before and after removal. (10 points)
5. `compare_accuracy()` — return a brief result stating whether accuracy improves, degrades, or stays the same after removing x_1. (5 points)

All grading is based strictly on returned values; printed messages are ignored.

Statement of Collaboration (5 points)

Add a plain text file `statement.txt` to your submission. List the names of collaborators and the nature of collaboration, or write "No collaboration." if you worked alone. Do not share code. Academic honesty policies apply.