

Programming Assignment-4 (Lab-7&8)  
CSL2050 - Pattern Recognition and Machine Learning

**NOTE:**

1. This Programming Assignment is associated with your Lab-7 and Lab-8.
  2. **Maximum Points:** 150 Points
  3. **Deadline:** March 16, 2024, 10:30 PM.
  4. **Lab Attendance:** Attending labs and finishing parts of the tasks during the lab is highly encouraged.
  5. **Late Submission Policy:** Late submissions beyond the due date will incur a 10% penalty for each day. Plan the submission ahead, and do not wait until the last minute.
  6. **Academic code of honor:** Please refer to course policy regarding academic code of conduct.
  7. Detailed submission instructions will be shared separately.
-

**Problems:**

1. **(LDA)** You are given a dataset of 2-dimensional samples in the following (x, y, label) format (refer data.csv, points can be in decimal)  
 1,2,0  
 1,1,0  
 5,6,1  
 8,9,1

**Task-1 (25 pts):** Compute the following terms and print them:

(For this Task, you are given a sample data.csv and helper code `pa_4_problem_1_task_1.py`, use it for the template, and write the function definitions there).

- (i). Difference of class wise means  $= m_1 - m_2$
- (ii). Total Within-class Scatter Matrix  $S_W$
- (iii). Between-class Scatter Matrix  $S_B$
- (iv). The EigenVector of matrix  $S_W^{-1} S_B$  corresponding to highest EigenValue
- (v). For any input 2-D point, print its projection according to LDA.

**Deliverable:** (i) myLDA.py that performs all these tasks we will test it on our version of data.csv (ii) For this task there is no requirement of any report.

**Code/Data Link:** <https://github.com/anandmishra22/PRML-Spring-2023.git>

(Task-2 and 3 can be done in Google Colab and observations in the report need to be submitted)

**Task-2 (5 pts):** Show the LDA projection vector on a plot.

**Task-3 (10 pts):** Compare the performance of 1-NN neighbor classifier on original data vs projected data. Write down your observations.

2. **(Naive Bayes)** You are given dataset (ref: naive\_bayes.csv) describing weather conditions and whether or not people played a certain outdoor sport. The features are Outlook, Temperature (Temp), Humidity, and Windy, and the target variable is Play (whether they played or not). **Dataset link:**

<https://github.com/anandmishra22/PRML-Spring-2023.git>

**Task-0 (0 pt):** Split the dataset into train-test so that randomly chosen 12 out of 14 samples go to train split and the remaining two samples go to test split.

**Task-1 (5 pts):** Calculate Prior Probabilities, i.e. the probability of playing ( $P(\text{Play}=\text{yes})$ ) and not playing ( $P(\text{Play}=\text{no})$ ).

**Task-2 (10 pts):** Calculate Likelihood Probabilities: i.e. the likelihood probabilities for each feature given the class ( $\text{Play} = \text{yes}$  or  $\text{Play} = \text{no}$ ). For

example, calculate  $P(\text{Outlook} = \text{Sunny} | \text{Play} = \text{yes})$ ,  $P(\text{Temperature} = \text{Mild} | \text{Play} = \text{yes})$ , and so on.

**Task-3 (10 pts):** Calculate Posterior Probabilities: Using the Naive Bayes formula, calculate the posterior probabilities for both classes (Play = yes and Play = no) for the testing split.

**Task-1 (5 pts):** Make Predictions: Based on the posterior probabilities, predict whether the given test split examples will result in playing the sport or not.

**Task-1 (10 pts):** Use Laplace Smoothing: Laplace smoothing is an essential technique in probabilistic models like Naive Bayes. It mitigates the challenge of zero probabilities for unseen events by introducing a small pseudocount. This adjustment ensures a more reliable and adaptable model, particularly when encountering unobserved combinations of feature values during classification.

Reference: <https://towardsdatascience.com/laplace-smoothing-in-na%C3%AFve-bayes-algorithm-9c237a8bdece>

Incorporating Laplace Smoothing, recalculate the Likelihood and Posterior Probabilities and make predictions on the test split. Report the observed differences in your predictions justify the results in the report.

---

**Rubrics:**

**Task completion with proper documentation/comments and variable naming:** 80 Points

**Viva:** 30 Points

**Report:** 40 Points

---

End of Paper