ALDA Fall 2021
HW2
9/01/2021

HW2 contains 3 questions. Please read and follow the instructions.

- **DUE DATE FOR SUBMISSION: 9/17/2021 11:45 PM**
- **TOTAL NUMBER OF POINTS: 100**
- **NO PARTIAL CREDIT** will be given so provide concise answers.
- **You MUST manually add ALL team members in the submission portal when you submit through Gradescope.**
- Make sure you clearly list **your homework team ID, all team members' names and Unity IDs, for those who have contributed to the homework contribution** at the top of your submission.
- **[GradeScope and NCSU Github]:** Submit a PDF on GradeScope. **You must submit your code in Github, and give the instructors access.**

  Follow these Github instructions: Please use the same homework repository you created for ALDA class and shared with us for HW1. The only thing you need to do for HW2 is to **create a new folder named HW2**. **All code MUST be in HW2 folder before the deadline. No credit will be given if code is not submitted for a programming question.** In your PDF submitted on GradeScope, reference the script/function for each question (e.g. "For solution to question 2, see matrix.py"). **Include your team's GitHub repository link in the PDF.**
- The materials on this course website are only for use of students enrolled in this course and **MUST NOT** be retained or disseminated to others.
- By uploading your submission, you agree that you have not violated any university policies related to the student code of conduct (`https://policies.ncsu.edu/policy/pol-11-35-01/`), and you are signing the Pack Pledge: **"I have neither given nor received unauthorized aid on this test or assignment"**.

---

1. (40 points) [**PCA**] [**John Wesley Hostetter & Ge Gao**] PCA is an unsupervised learning algorithm which uses an orthogonal transformation to convert data to new dimensions. In this problem, you will perform a PCA on the provided training dataset ("pca_train.csv") and the testing dataset ("pca_test.csv"). In both datasets, each row represents a data point or sample and the last column "Class" is a feature which indicate a class for each sample. The rest of the columns are input features.
Write code in Python to perform the following tasks. You can use Numpy, Pandas, Matplotlib, SciPy, and Sklearn to solve this problem. Please *report your output and relevant code* in the document file, and also include your code file (ends with .py) in your GitHub repository.

   (a) (2 points) Load the data. Report the size of the training and testing sets. How many Class (1) and Class (0) samples are in the training set and the testing set, respectively?

   (b) (18 points) **Preprocessing Data-Normalization**: Please run normalization on all input features in both the training and testing datasets to obtain the *normalized* training and the *normalized* testing datasets. (**Hint:** you need to use the *min/max of each column* in the training dataset to normalize the testing dataset, and do NOT normalize the output "Class" of data.)

   Use the **NEW** normalized datasets for the following tasks :

      i. (2 points) Calculate the covariance matrix of the *NEW* training dataset. Please 1) specify the dimension of the resulted covariance matrix and 2) given the space limitation, please report the first $5 * 5$ of the covariance matrix, that is, only reporting the first five rows and the first five columns of the entire covariance matrix.

      ii. (2 points) Calculate the eigenvalues and the eigenvectors based on the entire covariance matrix in (i) above. Report the size of the covariance matrix and the 5 largest eigenvalues.

      iii. (1 point) Display the eigenvalues using a bar graph or a plot, and choose a reasonable number(s) of eigenvectors. Justify your answer.

      iv. (13 points) Next, you will combine PCA with a *K-nearest neighbor (KNN)* classifier. More specifically, PCA will be applied to reduce the dimensionality of data by transforming the original data into p ($p \leq 30$) principal components; and then KNN ($K = 5$, euclidean distance as distance metric) will be employed to the p principal components for classification.
         - (5 points) Report the accuracy of the *NEW* testing dataset when using PCA ($p = 10$) with 5NN. To show your work, please submit the corresponding .csv file (including the name of .csv file in your answer below). Your .csv file should have 12 columns: columns 1-10 are the 10 principal components, column 11 is the original ground truth output "Class", and the last column is the *predicted* output "Class".

- (6 points) Plot your results by varying p: 2, 4, 8, 10, 20, 25 and 30 respectively. In your plot, the x-axis represents the number of principal components and the y-axis refers to the accuracy of the *NEW* testing dataset using the corresponding number of principal components and 5NN.
- (2 point) Based upon the (PCA + 5NN)'s results above, what is the **most "reasonable" number** of principal components among all the choices? Justify your answer.

(c) (18 points) **Preprocess Data-Standardization**: Similarly, please run standardization on all input features to obtain the *standardized* training and the *standardized* testing datasets. Then repeat the four steps i-iv in (b) above on the two **NEW** *standardized* datasets.

(d) (2 points) Comparing the results from (b) and (c), which of the two data-processing procedures, normalization or standardization, would you prefer for the given datasets? And why? (Answer without any justification will get zero point.)

2. (30 points) [**Decision Tree**] [**Angela Zhang**] For this exercise, you will use Internships.csv, from 12 students who applied for internships.

   The output label *class* has the following values:

   1. **Yes**: The student received an internship offer.
   2. **No**: The student did not receive an internship offer.

   The attributes can be described as:

   1. Department: Electrical Engineering, CS - Engineering, Business
   2. Academic Year: Sophomore, Junior, Senior
   3. Time of Interview: Morning, Afternoon, Late Afternoon
   4. Semester: Fall, Spring

   Complete the following tasks using the decision tree algorithm discussed in the lecture. Note that multiple-way splitting is allowed. *In the case of ties, break ties in favor of the leftmost attribute.* (You can hand-draw all of your trees on paper and scan your results into the final PDF.)

   (a) (15 points) Construct the tree *manually* using ID3/entropy computations, write down the computation process by showing the number of cases in each class for each node before splitting and show your tree step by step. (No partial credit)

   (b) (15 points) Construct the tree *manually* using the GINI index, take the attribute *academic year* as the root (then generate the next two levels of the remaining tree by choosing the best split), write down the computation process by showing the number of cases in each class for each node before splitting and show your tree step by step. (No partial credit)

3. (30 points) [**Evaluate Classifier**] [**Angela Zhang**] Joe pays careful attention to the weather. Unfortunately, the weather report is not reliable. It frequently predicts rain or storm incorrectly. Joe decides to create his own decision tree to help him stay dry.

   Figure 1 below is the decision tree created by Joe (*yes* if it will rain; *no* if it will not rain). Your task is to determine whether to prune the given decision tree. Additionally, you will use the provided test dataset in "RainPredict.csv" to determine the effectiveness of resulted decision trees.

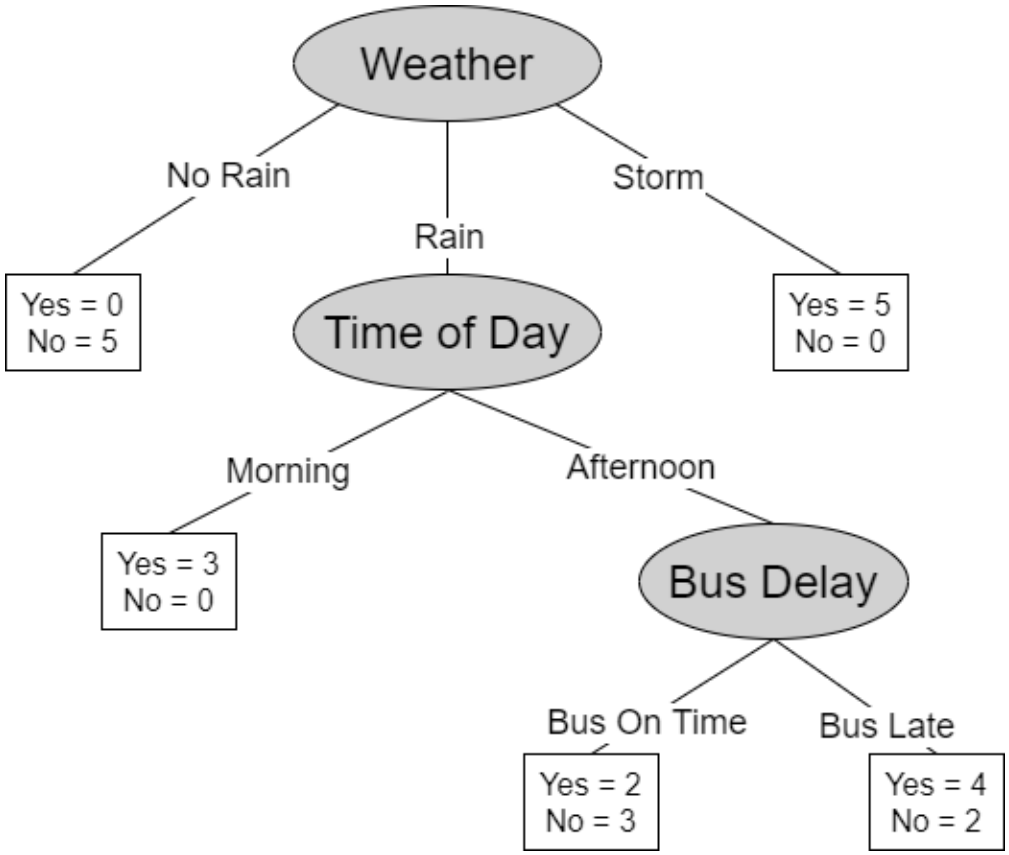

Figure 1: The Rain Prediction decision tree.

   (a) (10 points) Post-pruning based on **optimistic errors**.
      i. (3 points) Calculate the optimistic errors before splitting and after splitting using the attribute *Bus Delay* respectively.
      ii. (2 points) Based upon the optimistic errors, would the subtree be pruned or retained? If it is pruned, draw the resulting decision tree and use it for the next question; otherwise, use the original decision tree shown in Figure 1 for the next question.
      iii. (5 points) Use the decision tree from (a).ii above to classify the test dataset (RainPredict.csv). Report its performance on the following five evaluation metrics: Accuracy, Recall (Sensitivity), Precision, Specificity, and F1 Measure.

   (b) (10 points) Post-pruning based on **pessimistic errors**. When calculating pessimistic errors, each leaf node will add a factor of **2** to the error.

i. (3 points) Calculate the pessimistic errors before splitting and after splitting using the attribute *Bus Delay* respectively.

ii. (2 points) Based on the pessimistic errors, would the subtree be pruned or retained? If it is pruned, draw the resulting decision tree and use it for the next question; otherwise, use the original decision tree shown in Figure 1 for the next question.

iii. (5 points) Use the decision tree from (b).ii above to classify the test dataset (RainPredict.csv). Report its corresponding five evaluation metrics: Accuracy, Recall(Sensitivity), Precision, Specificity, and F1 Measure.

(c) (10 points) We will compare the performance of the decision trees from (a).ii and from (b).ii using the test dataset (RainPredict.csv). For the task of predicting if Joe will get caught in the rain, which of the five evaluation metrics: Accuracy, Recall(Sensitivity), Precision, Specificity, and F1 Measure, are the most important? Based on your selected evaluation metrics, which decision tree, (a).ii or (b).ii, is better for this task? Justify your answers.