Lab-3&4

CSL2050 - Pattern Recognition and Machine Learning

NOTE:

1. This Programming Assignment is associated with your Lab-3 and Lab-4. Please use the labs to understand problem statements and demo code prop erly.
2. Perform all tasks in a single Google Colab file. Prepare a report regarding the steps followed while performing the given tasks. The report should not include excessive unscaled preprocessing plots.
3. Try to modularize the code for readability wherever possible. Submit a zip with the Colab file [.ipynb] and report [.pdf] in the classroom. Name your files with your roll-number. (ex: B20CS003.ipynb and B20CS003.pdf)
4. Please refer to the Academic Code of Honor for this course (ref: Lecture-1) before submission of this programming assignment. Additionally, you are not allowed to use LLM for this assignment.
5. Maximum Points: 80
6. Deadline: Feb 5, 2024, 10:30 PM.
7. 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.

Practice: Please go through the following Demo Code before attempting the problems.

1. Demo code: https://colab.research.google.com/drive/1YMBqWjjCUOCE_AYHCiy03yP5X_d9BhBT?usp=sharing

Problems:

1. (Decision Tree) This problem is designed to help you understand the imple mentation aspects of a decision tree (DT). You are given a dataset: https:// github.com/datasciencedojo/datasets/blob/master/titanic.csv. It contains data to classify whether someone will survive in the Titanic wreck. You need to implement a classification decision tree (DT) from scratch (you are not permitted to use any 3rd-party library's function for the classifier, e.g., Scikit. You may, however, use built-in functions for auxiliary tasks like train/test split, etc.).

   The implementation includes the following tasks.

   Task-1: Perform pre-processing and visualization of the dataset. Also, men tion in your report whether the given features are ordinal, nominal or cate gorical. Perform categorical encoding wherever applicable and split the data into train, validation and test. Use 70-20-10 split. (5 points)

   Task-2: Implement the entropy as the cost function to calculate the split. (5 points).

   Task-3: In order for the decision tree to work successfully, continuous vari ables need to be converted to categorical variables first. To this end, you need to implement a decision function that makes this split. Let us call that conTocat(). The details of the function are the following: (i) Assume that the continuous variables are independent of each other i.e., assuming two continuous variables A and B, the split of A does not in any way affect the split you will perform in B. (ii) The continuous variables should only be split into two categories, and the optimal split is one that divides the samples the best, based on the value of the function you have been allotted. (10 points)

   Task-4: After conversion to categorical variables, you can now go ahead and implement the training function. This would include implementing the following helper functions: (a) Get the attribute that leads to the best split (b) Make that split (c) Repeat these steps for the newly-created split

   Further, the DT should also include the following properties in the train function: (a) There should be a max depth that should be defined i.e. a depth after which the tree shouldn't be allowed to grow. (b) The algorithm should self-identify when there is no information gain being done, i.e., the model has plateaued in it's training and shouldn't grow further. (10

points)

Task-5: Write a function, namely, *Infer* that takes one sample as input and uses the decision tree to classify it into one of the two classes, i.e., survived (1) or not survived (0). (3 points)

Task-6: Compute the accuracy you get on the training and test splits. (overall and class-wise). (2 points)

Task-7: Show the Confusion Matrix on the test data. (2 points)

Task-8: Compute precision, recall, F1-score of the Decsision Tree on the test split. (3 points)

2. (Linear Regression-1) You are given a dataset with two features: TV mar keting budget and sales. Dataset Link: https://raw.githubusercontent. com/devzohaib/Simple-Linear-Regression/master/tvmarketing.csv.

   Now, perform the following tasks:

   Task-1 Dataset Exploration: (a) Load the dataset and display the first few rows. (b) Plot a scatter plot to visualize the relationship between the TV marketing budget and sales. Comment on the trend observed in the scatter plot. (c) Calculate and display basic statistical measures (mean, standard deviation) for both TV marketing budget and sales. (5 Points)

   Task-2: Data Preprocessing: (a) Check for any missing values in the dataset and handle them appropriately. (b) Normalize the TV marketing budget and sales columns if needed. (c) Split the dataset into training and testing sets using 80-20 split. (5 Points)

   Task-3: Linear Regression Implementation: (a) Implement the hypothesis function for linear regression($y = w_1x + w_0$) using Gradient Descent. Use mean squared error (MSE) cost function. (b) Plot the regression line on the scatter plot from Task-1. (8 Points)

   Task-4: Evaluation: On test split compute mean square error and absolute error. (2 Points)

3. (Linear Regression-2) Now repeat all the tasks in Problem-2 for predicting house rent on the Boston Housing dataset (Link: http://lib.stat.cmu. edu/datasets/boston). Note that in this dataset, you need to perform multivariate linear regression. (20 Points).

End of Paper