Assignment 2

Due on November 17, 2021 (23:59:59)

Instructions. The goal of this problem set is to make you understand and familiarize with decision tree algorithm.You will experiment with decision tree model (by using ID3 algorithm) on the Diabetes Risk Prediction dataset.

Part 1: Diabetes Risk Prediction

In this assignment, you will implement a decision tree model to predict whether a patient is a potential diabetic or not.

A dataset [1] is provided for your training phase. You should split your training dataset into two set; training set which will be used to learn model, and validation set which will be used to measure the success of your model. You will use 5-fold cross-validation method which is explained in the class. You will implement ID3 concept within the scope of your decision tree model. **You will implement ID3 for on discrete attributes on the dataset and you will apply the discretization process on continuous attribute ("Age" attribute).**

Classification Dataset: Diabetes Risk Prediction Dataset [1]

- You can download the dataset from given link.
- Dataset consists of 520 samples with 16 features two class types. ("Positive" and "Negative")
- **Attribute and Class Information:**
  1. Age (Continuous Attribute)
  2. Gender (Discrete Attribute)
  3. Polyuria (Discrete Attribute)
  4. Polydipsia (Discrete Attribute)
  5. Sudden Weight Loss (Discrete Attribute)
  6. Weakness (Discrete Attribute)
  7. Polyphagia (Discrete Attribute)
  8. Genital Thrush (Discrete Attribute)
  9. Visual Blurring (Discrete Attribute)

10. Itching (Discrete Attribute)
11. Irritability (Discrete Attribute)
12. Delayed Healing (Discrete Attribute)
13. Partial Paresis (Discrete Attribute)
14. Muscle Stiffness (Discrete Attribute)
15. Alopecia (Discrete Attribute)
16. Obesity (Discrete Attribute)
17. Class (Output Prediction Class Information, "Positive" or "Negative")

Classification Performance Metric

You will compute "Accuracy", "Precision", "Recall" and "F1 Score" of your model to measure the success of your classification method based on your constructed confusion matrix, in which TP means "True Positive", TN means "True Negative", FP means "False Positive" and finally FN means "False Negative":

$$\textbf{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

$$\textbf{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\textbf{Recall} = \frac{TP}{TP + FN} \tag{3}$$

$$\textbf{F1Score} = \frac{2 * (Recall * Precision)}{(Recall + Precision)} \tag{4}$$

You will report these four measurement metrics for your of each test with respect to 5-fold cross validation and **finally also write the rules for your best decision tree model variation with respect to your 5-fold cross validation**.

Error Analysis for Classification

- Find a few misclassified samples and comment on why you think they were hard to classify.
- Compare performance of different ID3 model variation choices with respect to 5-fold cross-validation. Wherever relevant, feel free to discuss computation time in addition to classification rate.

Steps to Follow for Classification

1. Read your classification data and transform it to the Numpy array collection.
2. Prepare your training and testing sets with respect to the 5-fold cross validation.
3. Train your ID3 decision tree model (also including continuous "Age" attribute by discretization process) with respect to your training set.
4. **You can simply extract minimum and maximum value of your "Age" colon and then creating five intervals between your range of minimum and maximum values for the discretization process.**
5. For the test samples
   - predict their classes using your ID3 decision tree model
6. Compute and report your "Accuracy", "Precision", "Recall" and "F1 Score" of your different ID3 model parameters on 5-fold cross validation and finally **also write the rules of your best performing decision tree model with respect to these four metrics mentioned on "Classification Performance Metric"**.
7. Report your findings in "Error Analysis for Classification" section.

Part 2: Pruning Decision Tree

You are also expected to prevent overfit your decision tree by pruning the twigs of your tree. **The twigs are the nodes whose children are all leaves.**, For the pruning process, you will split the dataset into three parts, which are "training", "validation" and "test" sets. Training set should consist of 60 percent of the dataset, validation set should consist of 20 percent of the dataset and finally test set should consist of 20 percent of the dataset. The algorithm you follow for pruning process is below:

- Create an "Last Accuracy" variable and set this accuracy to the accuracy of your decision tree model on validation set before pruning process.
  - Step 1: Catalog all twigs in the tree
  - Step 2: Find the twig with the least Information Gain
  - Step 3: Remove all child nodes of the twig
  - Step 4: Relabel twig as a leaf (Set the majority of "Positive" or "Negative" as leaf value)
  - Step 5: Measure the **accuracy** value of your decision tree model with removed twig on the validation set ("Current Accuracy")
  - If "Current Accuracy $\geq$ Last Accuracy" : Jump to "Step1"
    Else : Revert the last changes done in Step 3,4 and then terminate

After the pruning process, you must write the rules and accuracy values on the test set for both of for your pre-pruning decision tree and post-pruning decision tree. Also you must compare them and state in your report the differences in these two models.

Implementation Details

- **You can't use ready-made libraries for your decision tree implementation and pruning process implementation. You must implement these on your own**
- You can use ready-made libraries for your K-fold cross-validation/Training-Test-Split/Shuffle methods for your data.
- You can use ready-made libraries for computing "Accuracy", "Precision", "Recall" and "F1 Score" metrics and for creating your confusion matrix.
- You may use Numpy array functions for your intermediate implementation steps for your implementations.
- You may use "Pandas" library for reading and writing/creating .csv files: `https://pandas.pydata.org/docs/index.html`

Submit

You are required to submit all your code (*all your code should be written in Jupyter notebook* long with a report in ipynb format (should be prepared using Jupyter notebook). The codes you will submit should be well commented. Your report should be self-contained and should contain a brief overview of the problem and the details of your implemented solution. You can include pseudocode or figures to highlight or clarify certain aspects of your solution.

Finally, prepare a ZIP file named **name-surname-pset2.zip** containing

- reportcode.ipynb (".ipynb" file including your code and report)

Grading

- Code (60): Part 1: (40 Points), Part 2: (20 Points)
- Report(40): Analysis of the results for Decision Tree Classification and Pruning

Note: Preparing a good report is important as well as the correctness of your solutions! You should explain your choices and their effects to the results. You can create a table to report your results.

Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else's work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.

References

[1] `https://www.kaggle.com/yasserhessein/early-stage-diabetes-risk-prediction-dataset`