Assignment 2

Due on 06.04.2025 (23:00)
Programming Language: Python 3.9

Introduction

The goal of this project is to make you understand and familiarize yourself with Decision Tree Algorithm. You will experiment with the decision tree model by using the ID3 algorithm.

The Financial Risk Assessment Dataset contains information about 15,000 people. There are 19 attributes for each people (some attributes for some people may be missing) and the aim of this project is predicting financial risk of people. Missing data must be handled up to your approach (the logic behind your approach must be explained clearly) and data must be split data into train (70%), test (15%), and validation (15%) sets randomly. This project consists of two parts. The first part involves implementing a decision tree (by using the ID3 algorithm) where the second part focuses on pruning the implemented decision tree for the sake of getting better generalization, which means having higher accuracy on the data that is not in train set.

Dataset: Financial Risk Assessment

The information about attributes and class are as follows:

- Age: The age of the individual, a continuous variable influencing financial stability.
- Gender: Gender of the individual, categorized into Male, Female, and Non-binary.
- Education Level: Highest level of education achieved, ranging from High School to PhD.
- Marital Status: Current marital status, categorized as Single, Married, Divorced, or Widowed.
- Income: Annual income in USD, representing the individual's earning capacity.
- Credit Score: Numeric value indicating creditworthiness, ranging from 600 to 800.
- Loan Amount: The amount of loan requested by the individual, representing financial needs
- Loan Purpose: The purpose of the loan, categorized into Home, Auto, Personal, or Business.

- Employment Status: Employment situation of the individual, including Employed, Unemployed, or Self-employed.
- Years at Current Job: Duration of employment at the current job, reflecting job stability.
- Payment History: Historical payment performance, categorized into Excellent, Good, Fair, or Poor.
- Debt-to-Income Ratio: Ratio of debt to income, indicating financial leverage and risk.
- Assets Value: Total value of assets owned by the individual.
- Number of Dependents: Number of dependents supported by the individual, affecting financial responsibilities.
- City: City where the individual resides, providing geographic context.
- State: State where the individual resides, giving further geographic detail.
- Country: Country of residence, adding a global perspective.
- Previous Defaults: Number of previous loan defaults, indicating historical financial risk.
- Marital Status Change: Number of changes in marital status, reflecting personal life changes
- Risk Rating: Target column categorizing financial risk into Low, Medium, or High.

Part 1: Implementing Decision Tree

1. Import and visualize the data in any aspects that you think it is beneficial for the reader's better understanding of the data.
2. Apply necessary preprocessing to the data. You are expected to handle missing data, obtain a confusion matrix from Scikit-Learn, and more beneficial preprocessing operations. **Note that you must do all preprocessing operations on your own (without using any external libraries for encoding, scaling etc. but built-ins and functions of numpy and pandas can be used).**
3. Split data into train, test, and validation set randomly (you can use 70% of the data for training, 15% of it for the testing, and 15% for validation purposes).
4. Train your ID3 decision tree model with respect to features that you selected.
5. Try to determine risk rating for the people at the test set that you separated at the third step.

6. Compute and report your Accuracy, Precision, Recall, and F_1 Score of your different ID3 model parameters. Finally, write the rules for your best decision tree model variation with respect to these four metrics and the dataset. While writing a decision tree model's rules, you must print all root-to-leaf paths in left-to-right order.

$$\textbf{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \qquad \textbf{Precision} = \frac{TP}{TP+FP}$$
$$\textbf{Recall} = \frac{TP}{TP+FN} \qquad F_1\textbf{-Score} = \frac{2}{recall^{-1}+precision^{-1}}$$

7. Find a few misclassified people and comment on why you think they were hard to classify.

8. Compare the performance of different ID3 model variation choices for your dataset. Comment about trade-off between computation time and classification rate in detail.

Part 2: Pruning Decision Tree

The tree that has been generated at the first part has inherently memorized the training data instead of learning from it, which results with overfitting. It means that the model performs very well at the training data but fails on the test and validation data. The solution that can be applied here is pruning the leaves that gives less, keeping the leaves that gives more. The logic is simple, if effect of a leaf is less, it means that it only works on some specific cases which means memorizing which results with overfitting, but if effect of a leaf is more, it means that it covers much cases which means that it is the result of learning from the data instead of memorizing it. For the pruning process, you will use the the train, test, and validation split of the dataset that you separated at third step of first part. The algorithm must be followed starts with creating a "Last Accuracy" variable and set this accuracy to the accuracy of your decision tree model on the validation set before the pruning process. Following steps are:

- 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 the twig as a leaf (Set the majority of "Low", "Medium" or "High" as leaf value)
- Step 5: Measure the accuracy value of your decision tree model with removed twig on the validation set ("Current Accuracy")
- Step 6: If "Current Accuracy $\geq$ Last Accuracy" : Jump to Step 1
  Else : Revert the last changes done in Step 3 and 4, then terminate

P.S.: You can also work with F_1 Score in addition to accuracy to determine which approach gives the best. Moreover, you can also hold last n versions of the tree in the memory and wait until last n-1 versions does not exceed the accuracy (and/or F_1 Score) of the n-th model from the last, and then revert back to n-th model from the last for the sake of avoiding sudden decreases at accuracy (and/or F_1 Score).

After the pruning process, you must write the rules; accuracy, precision, recall, and F_1 Score values on the train, test, and validation set for both your pre-pruning decision tree and post-pruning decision tree. Also, you must compare them and state in your report the differences between these two models. Report after the pruning process, which redundant features/attributes are pruned, and comment about why you think that features are pruned.

What to Hand In

You are required to submit all your code in a Jupyter notebook, along with a report in ipynb format, which should also be prepared using Jupyter notebook. **The code you submit should be thoroughly commented and your notebook must be ran and have outputs for each cell in the order of the cells before submission.** Your report should be self-contained and include a concise overview of the problem and the details of your implemented solution. Note that your report also has to contain necessary libraries to be installed with the versions that are used (!pip install commands are preferred). Feel free to include pseudo-code or figures to highlight or clarify specific aspects of your solution. Submission hierarchy must be as follows:

```
- <GroupID>.zip
    - assignment2.ipynb
    - *.(jpg|jpeg|png|gif|tif|tiff|bmp|svg|webp) (optional)
```

Do not send the dataset.

Preparing a good report is important as well as the correctness of your solutions! You should explain your choices and their effects on the results. You can create a table (or any content you believe that it is beneficial to show your all work) to report your results.

Note that submission format is crucial and submit system is set to give you score as one if you follow the submission hierarchy, which is really easy (there might be some issues for the MacOS users but it can be overcomed via the mini guide that is shared at the Piazza). If you do not score one from the submit system **you will penalized by 20% even if your submission hierarchy is correct**.

P.S.: You can use libraries for holding (such as numpy, pandas etc.) and representing (such as matplotlib, seaborn etc. for visualization and explanation) data, but **you must implement decision tree and things related to its mechanics from scratch, which means any library usage is forbidden except for the holding and representing the data purposes**. Note that you can use train_test_split from scikit-learn, it is an exception.

Academic Integrity

All work on assignments must be done on your own group 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 pseudo-code) 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.