[Project Code : CADT]
Credit Approval using Decision Tree based Learning Model

Project Duration : 21-Jan-2024 ~~ 10-Feb-2024
Submission Information : (via) CSE-Moodle

Objective:

The primary goal of this initiative is to revolutionize our credit approval process by developing and implementing a sophisticated decision tree-based system. Traditional credit evaluation methods often lack the precision and adaptability required to navigate the dynamic landscape of financial risk. This project aims to address this challenge by leveraging the rich dataset available and advanced decision tree algorithms to enhance the accuracy, efficiency, and transparency of our credit approval procedures.

Your task is to build a decision tree learning model to help find the class(approved/denied) of each data. In particular, you shall be doing the following tasks:

1. Based on the dataset (described later), you will write a program to learn a decision tree. You have to use any 1 of two methods in such decision tree learning:
   a. *Method-1:* Decision tree learning should use *entropy-based information gain* in selecting attribute choices while building the tree.
   b. *Method-2:* Decision tree learning should use *gini index* based criteria for choosing the attribute for splitting.
2. Use decision tree pruning techniques to eliminate overfitting

   Tree pruning should be performed at different depths. For pruning you may create a simple function to run your model using different values for a function maxdepth (say, from 1 to 25) and visualize its results to see how the accuracy differs for each criterion value (i.e. for *gini index* and *information gain*). Based on the best accuracy criteria at a certain depth the tree structure should be printed as output.
3. Both methods should be compared before and after pruning with the help of validation data you create in this assignment .
4. Compare the results with the results generated by the decision tree learning algorithm from a pre-created package such as sklearn. (Code snippet provided)

Note: *The program can be written in C / C++ / Java / Python programming language from scratch. No machine learning /data science /statistics package / library should be used for model creation.*

Dataset information:

Source: Quinlan,J. R.. Credit Approval. UCI Machine Learning Repository. https://doi.org/10.24432/C5FS30.

Dataset Filename: `crx.data`

Data Description:

Number of Instances: 690

Number of Attributes: 15 (discrete/continuous) + class name = 16

Attribute Information:

A1:	b, a.
A2:	continuous.
A3:	continuous.
A4:	u, y, l, t.
A5:	g, p, gg.
A6:	c, d, cc, i, j, k, m, r, q, w, x, e, aa, ff.
A7:	v, h, bb, j, n, z, dd, ff, o.
A8:	continuous.
A9:	t, f.
A10:	t, f.
A11:	continuous.
A12:	t, f.
A13:	g, p, s.
A14:	continuous.
A15:	continuous.
A16: +,-	(class attribute)

For more information on the dataset: https://archive.ics.uci.edu/dataset/27/credit+approval

Tasks to be done:

1. In the dataset,some of the features(A2,A3,A8,A11,A14,A15) are continuous so they have to be discretized.Also handle the missing values in the dataset.

2. The dataset is not divided into train, validation and test sets. The first task is to randomly partition the train dataset into train and test sets using 80-20 split.
3. Decision Tree Model without Pruning:
    a. Implement the standard **ID3 or Gini Decision tree algorithm** using information gain to choose which attribute to split at each point. Do NOT use scikit-learn for this part.
    b. Test the implementation of Decision Tree Classifier from scikit-learn package, using information gain (code snippet provided).
4. Revised Decision Tree Model with Pruning. To prune the tree, you have to use Reduced Error Pruning.
5. Classification Report
    a. Create a classification report for both the trees in tabular form. (with and without pruning).
    b. You need to calculate precision, recall, f1-score and accuracy of the model.

Submission Details: (to be submitted under the specified entry in CSE-Moodle)

1. ZIPPED Code Distribution in CSE-Moodle
2. A brief (2-3 page) report/manual of your work
   (with your hyperparameter tuning results also presented in that report)

Submission Guidelines:

1. You may use one of the following languages: C/C++/Java/Python.
2. Your Programs should run on a Linux Environment.
3. You are **not** allowed to use any library apart from these (Also explore all these libraries if doing in Python, or equivalent of these):

```
import numpy # linear algebra
import csv # data processing, CSV file I/O
import pandas # data processing, CSV file I/O
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.model_selection import KFold
from sklearn.tree import DecisionTreeClassifier # sklearn Decision Tree
import operator
from math import log
from collections import Counter
```

   Your program should be standalone and should **not** use any *special purpose* library for Machine Learning for the decision tree creation algorithm. Numpy and Pandas may be used. And, you can use libraries for other purposes, such as generation and formatting of data.

4. You should submit the program file and README file and **not** the output/input file.
5. You should name your file as <GroupNo_ProjectCode.extension>.
   (e.g., *Group99_CADT.zip* for code-distribution and *Group99_CADT.pdf* for report)
6. The submitted program file *should* have the following header comments:
   # Group Number
   # Roll Numbers : Names of members (listed line wise)
   # Project Number
   # Project Title

You should not use any code available on the Web. Submissions found to be plagiarized or having used ML libraries (except for parts where specifically allowed) will be awarded zero marks.

For any questions about the assignment, contact the following TA:
Akash kundu (Email: kunduakash91@gmail.com)