Machine Learning (CS60050): 2023-2024 Autumn

Assignment 1 Part 2

Submission deadline: 28-Aug-2023, 11:55 PM

Tarun Mohanty * and Ashwamegh Rathore *

*Indian Institute of Technology, Kharagpur

Instructions

1. The submission deadline is hard. There may be unforeseen glitches during submission. So, for safety, submit your files well ahead.
2. All submissions should be on microsoft teams only. No email submission will be accepted. Special consideration may be made for medical emergencies.
3. Please download the dataset and directory structure to be maintained from here.

   ```
   dataset
   ├── cross-validation.csv
   └── decision-tree.csv
   ```

4. For each part of the assignment implement the model in a **code.ipynb** file. Report the model performance by attaching a **report.pdf** which contains the output of all cells of the jupyter notebook. Refer to this article for converting the ipynb file to pdfs. Finally attach the list of package dependencies for each model in the **requirements.txt** file.
5. After implementing the required machine learning models, you must submit only the `<rollno>_<name>` directory. No need to submit the dataset directory. Directory structure to be submitted-

   ```
   rollno_name
   ├── decision-trees
   │   ├── code.ipynb
   │   └── report.pdf
   ├── cross-validation
   │   ├── code.ipynb
   │   └── report.pdf
   └── requirements.txt
   ```

   The name of the zip file should be your roll number, followed by an underscore, followed by your name. For example, if your roll number is `22CS0100` and name is `John Doe`, then the zip file should be named as `22CS0100_JohnDoe.zip`.

   Failing this, your assignment will not be evaluated.

6. **Unless explicitly asked,you cannot use any library/module meant for Machine Learning or Deep Learning.** You can use libraries for other purposes, such as formatting and pre-processing of data, but NOT for the ML part. Also you should not use any code available on the Web. Submissions found to be plagiarised or having used ML libraries will be awarded zero marks for all the students concerned.

1 Decision Trees

30 **marks**

Decision trees are graphical models that make decisions based on conditions, branching into outcomes or actions. They represent choices in a tree-like structure, aiding in classification or regression tasks by recursively partitioning data based on features, enabling interpretable and effective decision-making.

For this assignment,

1. Split the dataset into 80% for training and 20% for testing. Normalize/Regularize data if necessary. Encode categorical variables using appropriate encoding method if necessary.
2. Implement the standard **ID3 Decision tree** algorithm as discussed in class,using **Information Gain** to choose which attribute to split at eachpoint. Stop splitting a node if it has less than 10 data points. **Do NOT use scikit-learn for this part.**
3. Perform **reduced error pruning** operation over the tree obtained in (2). Plot a graph showing the variation in test accuracy with varying depths. Print the pruned tree obtained in hierarchical fashion with the attributes clearly shown at each level.
4. Report the **mean macro accuracy, macro precision and macro recall** for the classifier. You may or may not use the scikit-learn implementations for computing these metrics.

2 k-fold Cross Validation

20 **marks**

K-fold cross-validation is a technique used to assess and optimize the performance of machine learning models. The dataset is divided into K subsets, or "folds." The model is trained on K-1 folds and tested on the remaining one. This process is repeated K times, and the average performance is used to gauge the model's generalization ability.

For this assignment,

1. Split the dataset into 80% for training and 20% for testing. Normalize/Regularize data if necessary. Encode categorical variables using appropriate encoding method if necessary.
2. Train a **Logistic Regression model** on the dataset using **saga solver from scikit-learn package** and using **no regularization penalty**.
3. Cross Validate the classifier with **5-folds** and print the **mean accuracy, precision and recall** for the class 1(good) for the classifier. You may or may not use the scikit-learn implementations for computing these metrics. However, **you cannot use any ML package for the cross validation logic**.