Machine Learning Using Trees

A. Nagar

Due Date Mentioned in eLearning

Instructions

- This assignment requires you to build and compare tree models in Python using standard machine learning libraries.
- You should store your dataset under your account in the UTD server or any other public location, such as Google drive. Do not submit the dataset (which could be quite large) on eLearning,
- You are allowed to work in teams of maximum two students. Please write the names and NetIDs of each group member on the cover page.
  Only 1 final submission per team.
- **You have a total of 4 free late days for the entire semester. You can use at most 2 days for any one assignment. After four days have been used up, there will be a penalty of 10% for each late day. The submission for this assignment will be closed 2 days after the due date.**
- Please ask all questions on Piazza, not via email.

1 Project and Dataset Selection

For this assignment, you will need to choose a dataset from the UCI ML repository, which is located at: https://archive.ics.uci.edu/ml/datasets.php OR the Kaggle datasets available at https://www.kaggle.com/datasets.

You can choose either a regression or classification task. You will need to understand the dataset by reading its description and making sure you know which is the predicted variable and which are the predictors i.e. independent variables.

2 Tree Model Building

In this section, you will perform data pre-processing, loading, model creation and parameter tuning using GridSearchCV and results analysis. You will need to create the following tree models:

1. Plain Decision Tree Classifier / Regressor
2. Random Forest Classifier / Regressor
3. Adaboot Classifier / Regressor
4. XGBoost Classifier / Regressor

2.1 Pre-Processing

The following are the required pre-processing steps. Of course, you can add more as per your requirements.

- Loading the data into Pandas DataFrame object. Remember to use public URLs to read the file.
- Examining data for consistency: Check for null values, missing data, and any data inconsistency and handle them before proceeding forward.
- Examining attributes and target variable(s): Be sure you clearly understand each of the attributes and the target variable. Examine the various attributes and convert any categorical ones to numerical ones, if needed. Obtain and output summary of the attributes. Are the attributes normally distributed? If not, what could be the reason?
- Standardize and normalize the attributes.
- Find how the attributes are correlated to each other and the target variable. Perform numerical and visual analysis and output plots and results.
- Identify a few important attributes and proceed forward. Do not use all attributes blindly.

2.2 Model Construction

As stated earlier, you need to create four tree of models using scikit-learn or any other library of your choice.

You will need to use GridSearchCV or similar libraries for hyper-parameter tuning and cross-validation. It is up to you how many to choose, but you have to convince yourself and us that you have found the best values.

2.3 Tree Visualization

Wherever possible, you will need to visualize the tree or the best estimator tree. Use visually appealing libraries like graphviz or treeviz.

2.4 Result Analysis

By now, you should be familiar with results analysis using parameters other than just accuracy. You will need to output and **analyze** confusion matrix, and other metrics such as precision, recall, and F-statistic. You will also need to output and **interpret** in your report at least the following plots: ROC, Precision-Recall curve. Remember, result analysis is the key to success in a data science project.

Also, you need to compare the four models and write in your report which method works best and your analysis of the reason.

Please do not include code or code snippets in your report. Instead, submit them as a separate file.

3 Requirements

The following are requirements that **cannot be changed**

1. You are allowed to work in teams of maximum size 2
2. Treat this as a data science project. You have to interpret the output diagnostics. Also, try to include as many plots as you can. As stated previously, your interpretation and analysis of results is what we want to see.
3. You cannot copy any publicly available solutions. There will be penalty for plagiarism.
4. Submit your Python code file and report file. Please do not hard code any paths in your code. You can put the data in your UTD web account and read from that link.
5. Python code can be on Google Colab or Jupyter Notebook.

If you have made any assumptions, please state them completely. Also include instructions on how to compile and run your code in a README file.