In the name of God
Machine Learning Course (Spring 2023)
Assignment #4 : Decision Tree & Ensemble Algorithms
Due date: 13th May (23th Ordibehesht)

There are two main parts to this assignment, and in each, you are expected to learn about the concepts of decision trees, ensembles, AdaBoost, and imbalanced data. The descriptions of every part are provided below.

Part A (Decision Tree): Dataset: Covid19HDDT.csv

In this part, you are about to implement HDDT, which is a decision tree based on Hellinger distance, and it is suitable for unbalanced data. Read more in the attached documents for more information about the HDDT algorithm and its implementation.

Algorithm 1. *Calc_Hellinger*

Input: Training set T, Feature f

1: **for** each value v of f **do**
2: $Hellinger+ = (\sqrt{|T_{x_f=v,y=+}|/|T_{y=+}|} - \sqrt{|T_{x_f=v,y=+}|/|T_{y=+}|})^2$
3: **end for**
4: **return** $\sqrt{Hellinger}$

Algorithm 2. *HDDT*

Input: Training set T, Cut-off size C

1: **if** $|T| < C$ **then**
2: **return**
3: **end if**
4: **for** each feature f of T **do**
5: $H_f = Calc\_Hellinger(T, f)$
6: **end for**
7: $b = \max(H)$
8: **for** each value v of b **do**
9: $HDDT(T_{x_b=v}, C)$
10: **end for**

The following algorithm outlines the approach to incorporating Hellinger distance in learning decision trees. We will refer to Hellinger distance and Hellinger distance-based decision trees as HDDT for the rest of the paper. In our algorithm, $T_{y=i}$ indicates the subset of training set T that has class i, $T_{xk=j}$ specifies the subset with value j for feature k, and $T_{xk=j,y=i}$ identifies the subset with class i and has value j for feature k .

You are required to evaluate the performance of HDDT on the Coronavirus data set. Since the data set is unbalanced, you should use **Precision**, **Recall**, **F-measure**, and **AUC** measures to evaluate the performance of your tree. It is worth mentioning that these measures are one-class measures; in other words, you should compute these metrics just for the minority class. Moreover, **G-mean** should be reported.

In the name of God
Machine Learning Course (Spring 2023)
Assignment #4 : Decision Tree & Ensemble Algorithms
Due date: 13th May (23th Ordibehesht)

Split the data set(Covid19HDDT.csv) to train and test parts. Use 70% of the data for training phase and the remaining 30% for testing phase. Run your codes for 10 individual runs and report the average of 10 runs for each performance metric.

Step1:

Since the data set has three classes and HDDT is a two-class algorithm you are required to convert the data to a two-class data set by keeping the smallest class as minority and the rest as majority. Implement the two-class HDDT and evaluate its performance in terms of all mentioned metrics.

After you implement the two-class HDDT, you need to extend it to handle the multiclass data with OVO (One Versus One) and OVA (One Versus All) approaches. Use the original version of the data sets without converting it to multiclass (Covid19HDDT.csv). Then evaluate the performance of both approaches with the mentioned metrics and report the results.

Step2:

Repeat both of the experiments in previous parts with the pruned HDDT trees. Simply put, consider the longest height of the tree and prune it with different heights of $MaxHeight = \{2, 3, 4, 5\}$ where MaxHeight is the height of the HDDT tree. Compare your results with the unpruned versions of the trees and report the results.

Questions

Part A:

1. What are the properties of the HDDT algorithm? Why is it suitable for unbalanced data?
2. What are the differences between Hellinger distances, Gini index, and information gain?
3. Is pruning lead to better results? Why?

In the name of God
Machine Learning Course (Spring 2023)
Assignment #4 : Decision Tree & Ensemble Algorithms
Due date: 13th May (23th Ordibehesht)

Part B (Ensemble Learning): Dataset: Covid.csv

In this Part, you should implement a bootstrap ensemble algorithm which is Bagging for imbalanced data. Also, you can implement another ensemble as a bonus which is AdaBoost.M1.

Method I: Bagging with UnderSampling

For each iteration in Bagging, draw a bootstrap sample from the minority class. Randomly draw the same number of samples, with replacement, from the majority class. Repeat this step for the ***T*** number of times. Aggregate the predictions of base learners and make the final prediction.

Note 1: In the Bagging Algorithm, the number of learners or iterations ***T*** should be selected from the set {11, 31, 51, 101}, and the results should be reported accordingly.

You need to use both:

1. **Hellinger decision tree**
2. **Built-in Decision tree**

as the base learner of your ensemble models.

- Split the data set into train and test parts. Use 70% of the data for the training phase and the remaining 30% for the testing phase.
- Run your codes for 10 individual runs and report the *mean* and *standard deviation* of 10 runs for each performance metric.
- 1. **max_depth** parameter of the base learners in the HDDT algorithm should be tuned **experimentally** so that the decision tree performs a little better than a random classifier.
  2. For the Bagging algorithm, use the default parameters of the base learner decision tree.
- Since the dataset (Covid.csv) is imbalanced, you should use **Precision**, **Recall**, **F-measure**, **ROC curve,** and **AUC** measures to evaluate the performance of these algorithms.
  It is worth mentioning that these measures are one-class measures; in other words, you should compute these metrics just for the minority class. Moreover, **G-mean** should be reported.

In the name of God
Machine Learning Course (Spring 2023)
Assignment #4 : Decision Tree & Ensemble Algorithms
Due date: 13th May (23th Ordibehesht)

$$True\ Positive\ Rate\ (Acc_+) = \frac{TP}{TP + FN}$$

$$True\ Negative\ Rate\ (Acc_-) = \frac{TN}{TN + FP}$$

$$G - mean = \sqrt{Acc_+ \times Acc_-}$$

Questions

1. Why should we set max_depth parameter in Bagging with HDDT so that the base classifiers become a little better than random guess?
2. What do we mean by stable, unstable, and weak classifier?
3. What kind of classifiers should be used in Bagging? How about AdaBoost.M1? Why?

Method II: Adaboost with UnderSampling (We consider **just this part** as Bonus)

The motivation of this method is to keep the high efficiency of under-sampling but reduce the possibility of ignoring potentially useful information contained in the majority class examples. This method randomly generates multiple sample sets $\{N_1, N_2, \ldots, N_T\}$ from the majority class N. The size of each sample set is equal to the size of the minority class, i.e., $|N_i| = |P|$. In each iteration, the union of pairs N_i and P is used to train the AdaBoost ensemble. The final ensemble is formed by combining all the base learners in all the AdaBoost ensembles. The algorithm is shown in Figure below.

In the name of God
Machine Learning Course (Spring 2023)
Assignment #4 : Decision Tree & Ensemble Algorithms
Due date: 13th May (23th Ordibehesht)

Input: Training data set $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$;
Minority class examples $P \subseteq D$;
Majority class examples $N \subseteq D$;
Number of subsets T to sample from N;
Number of iterations s_i to train an AdaBoost ensemble H_i.

Process:

1. **for** $i = 1$ to T:
2. Randomly sample a subset N_i from N with $|N_i| = |P|$;
3. Use P and N_i to learn an AdaBoost ensemble H_i, which is with s_i weak classifiers $h_{i,j}$ and corresponding weights $\alpha_{i,j}$:

$$H_i(\boldsymbol{x}) = \texttt{sign}\left(\sum_{j=1}^{s_i} \alpha_{i,j} h_{i,j}(\boldsymbol{x})\right).$$

4. **end**

Output: $H(\boldsymbol{x}) = \texttt{sign}\left(\sum_{i=1}^{T} \sum_{j=1}^{s_i} \alpha_{i,j} h_{i,j}(\boldsymbol{x})\right).$

This algorithm generates T balanced sub-problems. The output of the ith sub-problem is AdaBoost classifier H_i , an ensemble with s_i weak classifiers (consider built-in decision tree as weak classifier) $\{h_{i,j}\}$. Finally, instead of counting votes from $\{H_i\}_{i=1,\ldots,T}$, we collect all the $h_{i,j}$ (i = 1, 2, . . ., T; j = 1, 2, . . . , s_i) and form an ensemble classifier out of them.

The output of the above algorithm is a single ensemble, but it looks like an "**ensemble of ensembles**".

Note 2: The number of iterations T, and the number of rounds s_i should be selected from the sets {10, 15} and {11, 31, 51, 101}, respectively. The classification results of all these parameters should be provided in your report.

Notes:

- Pay extra attention to the due date. It will not extend.
- Be advised that submissions after the deadline would not grade.
- Prepare your entire report in PDF format and include the figures and results.
- Submit your assignment using a zipped file with the name of "StdNum_FirstName_LastName".zip
- Using other students' codes or the codes available on the internet will lead to zero.