DS 303: Introduction to Machine Learning

Quiz 3, Total Points: 25 points
Submit by 11.59pm, 18th March on Moodle

March 17, 2024

Question 1: Random Forest [15 points]

[points] Part 1: Complete the functions – Open the `RandomForest.ipnb` from https://github.com/soumenkm/DS303/blob/main/RandomForest.ipynb. You are required to write the codes for the following 3 functions –

1. [1 points] `entropy`
2. [5 points] `randomforestclassifier.fit`
3. [2 points] `randomforestclassifier.predict`

Do not modify/change the code of any other part of the notebook. Write your code only in the designated space provided within the functions. Change the `return` variable if the function returns meaningful results.

[4 points] Part 2: Use standard ML library – Calculate the `Accuracy` for the Random Forest classifier applied to the same dataset used in `Part 1` using the standard `Scikit Learn` library.

[3 points] Part 3: Find bugs in the code – Open the `buggy_code.py` from https://github.com/soumenkm/DS303/blob/main/buggy_code.py. This code has a major logical error. Find the logical error and give a solution to address the issue. Moreover, the model is intentionally overfitted. Find the reason behind overfitting in the code and give a solution to prevent overfitting.

Question 2: AdaBoost [10 points]

[5 points] Part 1: Complete the functions – Open the `adaboost.py` from https://github.com/soumenkm/DS303/blob/main/adaboost.py. You are required to write the codes for the following 2 functions –

1. [5 points] `Adaboost.fit`
2. [2 points] `Adaboost.predict`

Do not modify/change the code of any other part of the notebook. Write your code only in the designated space provided within the functions. Change the `return` variable if the function returns meaningful results.

[3 points] Part 2: Use standard ML library – Calculate the `Accuracy` for the Adaboost classifier applied to the same dataset used in `Part 1` using the standard `Scikit Learn` library.

Extra credit question: [6 points]

This is continuation of Question 1 on Random forests. Answer the following on Cross Validation.

1. [2 points] Recall that a random forest is created from a number of decision trees, with each decision tree created from a bootstrapped version of the original training set. One hyperparameter of a random forest is the number of decision trees we train to create the random forest. Define T to be the number of decision trees used to create the random forest. Let's say we have two candidate values for T: T_1 and T_2. We want to perform T_3 - fold cross-validation to determine the optimal value of T. Assume T_1, T_2 and T_3 are integers. Write the answers for the following in terms of T_i, $\forall i \in \{1, 2, 3\}$

   (a) [1 points] In this cross-validation process, how many random forests will we train?

   (b) [1 points] In this cross-validation process, how many decision trees will we train?

2. [1 points] Let's say we pick three hyperparameters to tune with cross-validation. We have 5 candidate values for hyperparameter 1, 6 candidate values for hyperparameter 2, and 7 candidate values for hyperparameter 3. We perform 4-fold cross validation to find the optimal combination of hyperparameters, across all possible combinations. In this cross-validation process, how many random forests will we train?

3. [3 points] Here is some code that attempts to implement the cross-validation procedure described above. Your task is to complete the function as follows.

```
from sklearn.model_selection import KFold
from sklearn import ensemble
import numpy as np
import pandas as pd
from sklearn.datasets import load_iris

def cross_validate(X_train, Y_train, cands1, cand
"""Hint 1: Use KFold class
Hint 2: Use ensemble.RandomForestClassifier model
Hint 3: Calculate accuracy and
return the list of accuracies
"""
# Write your code here
pass
return cv_scores

# Load the Iris dataset
iris = load_iris()
X = pd.DataFrame(iris.data, columns=iris.feature_n
Y = pd.Series(iris.target)
```

```
# Example usage:
cands1 = [5, 10, 15]  # max_depth candidates
cands2 = [50, 100, 150]  # n_estimators candidate

# Run the function
output = cross_validate(X, Y, cands1, cands2)

# Sample output
print("Cross-validation scores:", output)
```