Assignment 1: Classification

Mohamed Darkaoui

April 16, 2023

1 Introduction

For this project, I primarily focused on decision tree-based models since they are known to perform well for this type of tasks. I began with a standard decision tree and then optimized it using cost complexity pruning. Next, I explored ensemble classifiers, namely the Random Forest and AdaBoost methods.

2 Data Pre-processing

2.1 Missing values

I found out that there are missing values in the columns workclass, occupation and native-country. These are all categorical columns. Removing all rows that contain at least one missing value will result in a relatively small loss of data, however in the real world setting we have to deal with missing data. When a tuple with missing values comes in, I want the model to predict it instead of throwing it away. So I chose to create a separate category for the missing values.

2.2 encoding

For encoding the data, I employed the one-hot encoding technique, which considerably expands the dataset's size. I chose one-hot encoding over label encoding for categorical variables because I've read that label encoding can introduce a false sense of hierarchy into the model. Meanwhile, the target labels are encoded using the label encoding method.

2.3 Normalization

Since I'm using decision tree based models only in my project, there is no need for normalization of data.

3 Training

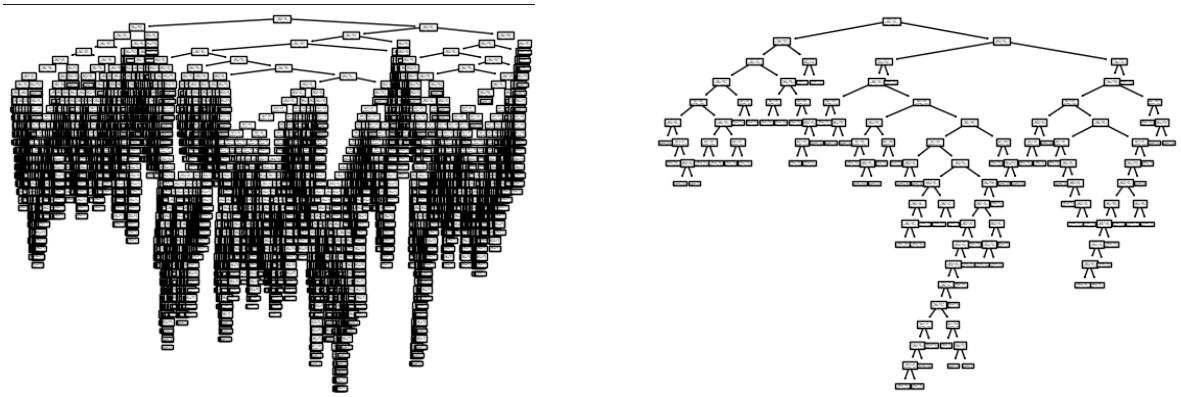
I calculated the expected return using the following formula:

$$pN * 0.1 * 980 - (1 - p)N * 0.05 * 310 - 10N$$

Where p is the precision and N is the number of customers predicted to be earning more than 50K per year.

3.1 Decision Tree

The regular decision tree had an accuracy of 82%, precision of 63% and recall of 62%. The expected return is 172986 euro. After cost complexity pruning where I maximize the precision, recall and accuracy consecutively, I get higher expected returns which are shown in the table below. Figure1 shows the image of the tree before and after applying the cost complexity pruning.



(a) *unpimized* (b) *opimized*

Figure 1: Visualization of the optimized vs the un-optimized decision tree

3.2 Random Forest

I applied the random forest classifier on the data, and got results that are slightly lower than those of the optimized decision tree, even after optimization using grid search.

3.3 AdaBoost

I used this classifier without parameter optimization. The results were also lower than thos of the optimized decision tree.

3.4 Results

metric	decision tree	random forest	ada boost
precision	0.72	0.71	0.71
recall	0.65	0.62	0.62
accuracy	0.85	0.84	0.85
exp.return	192447.7	189925.6	180854.8

Note that only the best results are shown of each type of classifier.