Data Science Homework 4

Andrés Ponce, **彭思安** P76107116

May 8, 2022

1 Introduction

Our data's properties will determine the pre-processing steps we take to ensure our model performs well with new data. For instance, since machine learning methods can for the most part deal only with numerical values, its is necessary to convert string-based values into numbers. Such methods range from just assigning a numerical value to a class index to using hash tables to store indices of the column to which it belongs.

Another common issue in the data processing stage is imbalanced data. This happens when some categories in our data happen less often than others, sometimes much less often. These categories are known as **minority class**, and more common classes or categories are known as **majority class**. To ensure our model is exposed to the category, we can either remove elements from the majority class with **undersampling methods** or create new samples from the minority class using **oversampling methods**.

In this assignment we investigate the relation between datasets and different encoding and sampling methods. First, we describe the datasets used, then conduct experiments on different combinations of encoding and sampling methods.

2 Datasets

In this assignment, we used 15 classification datasets, most of which have a class imbalance. The datasets were mostly obtained mostly from Kaggle, the UCI repository, and OpenML. Table 2 shows basic information about our datasets. Some of the tasks involve doing binary classification, where the target class has only two distinct values, while others have multiple target values. Imbalanced datasets are often handled using measures such as **precision** and **recall**, which measure the percentage of correct classifications and and the percentage of samples we correctly classified as belonging to a different class, respectively.

With multiclass datasets, we can take each class c_i as the positive class and all the other classes as the negatives. The `imblearn` library provides functions to calculate the precision, recall, F score, and support for each class.

Dataset	Features	Categorical	Numerical	Size	Task
Heart Disease	14	11	3	319,795	Multi-class
Bank Marketing	20	9	11	32561	Binary
Income Evalutaion	15	9	6	32,562	Binary
Telco Customers	21	19	2	7044	Binary
Abalone	8	1	8	4178	Multi-Class
IBM Attrition	13	4	9	1470	Multi-Class
Biostar Degradation	42	0	42	1055	Binary
Credit Card	25	20	4	30,000	Binary
Fuel Consumption	15	9	6	947	Multi-Class
Travel Insurance	10	4	6	1986	Binary
Diabetes	15	1	14	520	Binary
Loans	10	5	5	399	Binary
Onlne Shopper intention	18	8	10	12,330	Binary
Teacher Assistant	6	5	1	150	Multiclass
Churn Modelling	14	2	12	10,000	Binary

Table 1: Properties of the datasets used.

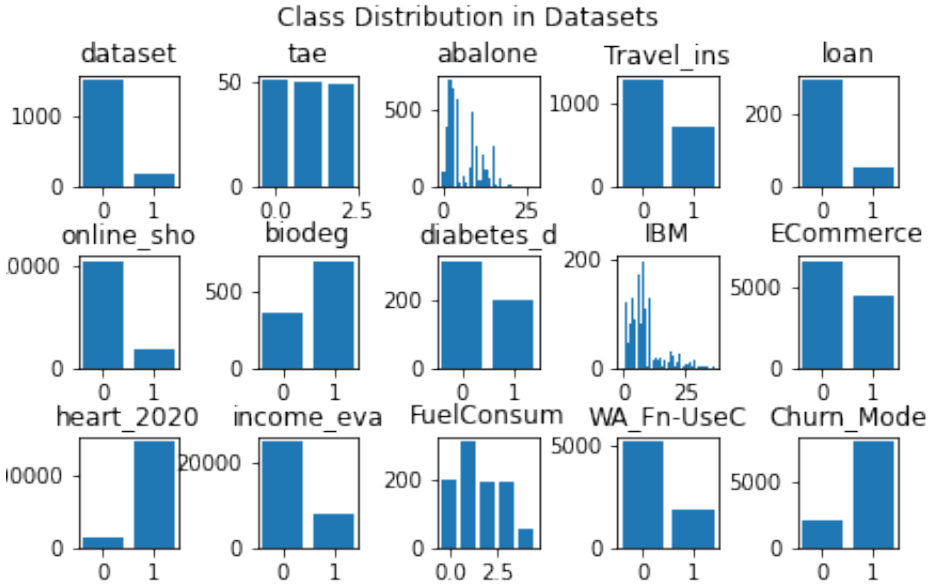

Figure 1: Distribution of the target labels of each file tested.

Heart Disease	87.9	87.9
Bank Marketing	88.6	87.8
Income Evaluation	84.8	85.1
Telco Customers	78.3	78.8
Abalone	23.3	23.3
IBM Attrition	15.9	14.2
Biostar Degradation	86.6	85.5
Credit Card	88.6	87.8
Fuel Consumption	80	76
Travel Insurance	80.6	80.7
Diabetes	98.4	97.6
Loans	81.2	82.0
Onlne Shopper intention	89.6	89.4
Teacher Assistant	61.3	62
Churn Modelling	85.0	84.8

Table 2: Effects of applying feature standardizaiton.

3 How does feature scaling (i.e. doing standardization) affect the performance?

To test the effect of standardization on the model performance, we train a `RandomForest` classifier with 10 decision trees. Since we want to measure the effect of standardization on performance as a whole, we use classification accuracy as our measure.

As seen in Table 2, in our experiments feature scaling has only a small influence in our accuracy. Some datasets such as IBM Attrition and Abalone had poor performance because there are classes that have only one sample. This poses problems when using a `StratifiedKFold` cross-validation method. It is important that we use this evalutation method so we can roughly preserve the original distribution of data in each fold. Most results fall within one percentage point difference between the standardized and non-standardized approaches.

4 When using tree-based algorithms, will using one-hot encoding for categorical features generate worse performance than label encoding? Why?

Using one-hot encoding for a RandomForest classifier does indead lead to worse performance, sometimes by significant amounts. We run our experiments again using a `RandomForest` classifier with 10 decision trees.

One-hot algorithms greatly increase the amount of columns in the dataset. Tree-based classifiers using an impurity metric such as GINI have a greater chance of using the new columns in their splits at every stage. This means that we can end with deep decision trees

that rely on these columns.