Machine Learning - SS 2021
Exercise 1: Classification

Federico Ambrogi, e1449911@student.tuwien.ac.at
Adam Höfler, e11847620@student.tuwien.ac.at
Matteo Panzeri 12039996@student.tuwien.ac.at
TU Wien

April 25, 2021

Contents

1 Introduction

In this document we describe the results of the implementation of three algorithms to solve classification problems on four different dataset.

1.1 Data Sets Description

Here we briefly introduce our four datasets.

Drug consumption data set This is the data set taken from exercise 0, so here we only provide basic details. The primary task of our implementation is to classify the likeliness of the frequency of usage of a certain type of drug (from never used to current usage), given personal aspects such as gender, ethnicity, education background etc. and psychological traits. In detail:

- **attributes**: 'age', 'gender', 'education', 'ethnicity', 'Nscore', 'Escore', 'Oscore', 'Ascore', 'Cscore', 'Impulsive', 'SS'

- **targets**: 'alcohol', 'amphetamines', 'amylNitrite', 'benzodiazepine', 'caffeine', 'cannabis', 'chocolate', 'cocaine', 'crack', 'ecstasy', 'heroin', 'ketamine', 'legal', 'LSD', 'methadone', 'mushrooms', 'nicotine', 'volatileSubstance' divided in **7 classes**: "Never", "¿10 Years Ago", "Last Decade", "Last Year", "Last Month", "Last Week", "Last Day"

Asteroids data set The asteroids data set is designed to perform a binary classification task. It consists of a total of astrophysical data, such as the distance from the Earth, the size of the major axis of the orbits, their mass etc. as well metadata regarding their name. The target feature for classification is a string variable that maps to boolean values "True" in the case the asteroid represents a concrete hazard for the Earth i.e. with high risk of impact, and "False" otherwise.

In detail:

- **attributes**:'Absolute Magnitude', 'Est Dia in KM(min)', 'Est Dia in KM(max)', 'Relative Velocity km per sec', 'Miss Dist.(kilometers)', 'Minimum Orbit Intersection', 'Jupiter Tisserand Invariant', 'Eccentricity', 'Semi Major Axis', 'Inclination', 'Asc Node Longitude', 'Orbital Period', 'Perihelion Distance', 'Perihelion Arg', 'Aphelion Dist', 'Perihelion Time', 'Mean Anomaly', 'Mean Motion'

- **target**: 'Hazardous' divided in **classes**: "True" and "False"

Breast Cancer (Kaggle Data Set) ccc

- **attributes**: 'radiusMean', 'textureMean', 'perimeterMean', 'areaMean', 'smoothnessMean', 'compactnessMean', 'concavityMean', 'concavePointsMean', 'symmetryMean', 'fractalDimensionMean', 'radiusStdErr', ' textureStdErr' ,'perimeterStdErr' ,'areaStdErr', 'smoothnessStdErr', 'compactnessStdErr', 'concavityStdErr', 'concavePointsStdErr' ,'symmetryStdErr' ,'fractalDimensionStdErr', 'radiusWorst', 'textureWorst' ,'perimeterWorst' ,'areaWorst', 'smoothnessWorst' ,'compactnessWorst' ,'concavityWorst', 'concavePointsWorst', 'symmetryWorst', 'fractalDimensionWorst'

- **target**: 'class' divided in **classes**: "True" and "False"

Advertising Bidding (Kaggle Data Set) In detail:

- **attributes**: 'Region', 'City', 'AdExchange', 'Adslotwidth', 'Adslotheight', 'Adslotfloorprice', 'CreativeID', 'Biddingprice', 'AdvertiserID', 'interest_news', 'eduation', 'automobile','interest_realestate', 'IT', 'electronicgame','interest_fashion', 'entertainment', 'luxury','homeandlifestyle', 'interest_health', 'food','interest_divine', 'interest_motherhood_parenting', 'sports','interest_travel_outdoors', 'interest_social', 'Inmarket_3cproduct','Inmarket_appliances', 'Inmarket_clothing_shoes_bags' ,

  'Inmarket_Beauty_PersonalCare', 'Inmarket_infant_momproducts', 'Inmarket_sportsitem', 'Inmarket_outdoor', 'Inmarket_healthcareproducts', 'Inmarket_luxury', 'Inmarket_realestate',

  'Inmarket_automobile', 'Inmarket_finance', 'Inmarket_travel', 'Inmarket_education', 'Inmarket_service', 'art_photography_design', 'onlineliterature', 'Inmarket_electronicgame', '3c', 'Inmarket_book', 'Inmarket_medicine', 'Inmarket_fooddrink', 'culture', 'sex', 'Demographic_gender_male',

  'Demographic_gender_famale', 'Inmarket_homeimprovement', 'Payingprice', 'imp', 'click', 'Browser', 'Adslotvisibility', 'Adslotformat'

- **target**: 'conv' divided in **classes**: "True" and "False"

In detail:

- **attributes**:'Absolute Magnitude', 'Est Dia in KM(min)', 'Est Dia in KM(max)', 'Relative Velocity km per sec', 'Miss Dist.(kilometers)', 'Minimum Orbit Intersection', 'Jupiter Tisserand Invariant', 'Eccentricity', 'Semi Major Axis', 'Inclination', 'Asc Node Longitude', 'Orbital Period', 'Perihelion Distance', 'Perihelion Arg', 'Aphelion Dist', 'Perihelion Time', 'Mean Anomaly', 'Mean Motion'

- **target**: 'Hazardous' divided in **classes**: "True" and "False"

1.2 Framework and Implementation

We implemented our analysis using the **python3.8** language, with its library for machine learning application *scikit-learn* as well as *numpy* and *pandas* for data statistical analysis and data manipulation. In the attached package, two distinct files can be found: the first, called *data_preparation.py*, cleans and standardizes the input data set, so that they are ready for data processing. The main script for the data analysis is called *classifier.py*. This code will run the different classifiers on the data sets, and will produce the comparison data and plots that will be found in this report.

1.3 Workflow

Without giving particular details on the implementations of the various classifier, which can be read inside the scripts, we summarize here how the workflow is constructed, as well as how the classification tasks have been implemented. First, the cleaned data (i.e. processed by *data_preparation.py*) are imported as *pandas* data frame. For each data set, a dictionary provided the list of names of features that will be used to predict the target feature. Then, we analyse the class balancing of the predicting features; it is possible to decide to apply a balancing procedure before processing further(see Section 3.4 for details). The data can now be trained: the splitting of the data into a training and testing set is done differently depending on the chosen methods between "hold out" (by the $model_selection.train_test_split$) or "kNN-cross validation" (by the XXX); see Section 3.3. At this point, the classification algorithms are applied to the training and testing sets. The following algorithms have been chosen.

Decision Three We use the *DecisionTreeClassifier* module, where we try two implementation of the "Gini" and "entropy" (parameter *criterion*) criteria for the Gini impurity and for the information gain,respectively, to measure the quality of a split.

kNN - k Nearest Neighbours We use the *KNeighborsClassifier* module, where we vary the number of neighbours (parameter $n_neighbors$) in the set $[5, 10, 50]$.

naive Bayes We use the *GaussianNB* module, with default parameters.

2 Data Exploration and Pre-processing

In this section we describe the necessary preliminary steps to import and prepare the data to make them suitable for the learning algorithms. The steps are handled by the script *data_preparation.py* which includes dedicated function to pre-process each data set.

Drug consumption Since this dataset was already preprocessed (transformation of the nominal features) and cleaned of missing values or outliers by the providers of the dataset [1], the data preparation is not given. As can be seen in Fig.2 we deal with heavily imbalanced data. Normally, the approach would be to resample the data (see Sec.3.4) but with our later on discussed approach we would run into problems due to low statistics - we decided to keep the dataset as it is.

As can be seen in Fig.2 the majority of the drugs can be grouped by their imbalance towards either CL0 ("Never used") or CL6 ("Last day") - those two groups give a rough representation of "everyday drugs" like chocolate and "marginal drugs" like heroin. Since the total number of respondents is constant, a vast majority of one class implies a small occurrence of the other classes and equalizing the samples results in low statistics and no usable outcome of the classification - this was the result of our tests.

2.1 Asteroids Data Set

The original data set contained 40 columns of different variable types (numeric and categorical). However, some columns were unnecessary, not usable or redundant for classification purposes, or contained constant values.

2.2 Advertising Bidding (Kaggle Data Set)

In the dataset we have both nominal and ordinal data. An example of nominal attribute is the Browser where is specified the browser used by the user in that observation. An example of ordinal data is the "Inmarket_realestate" which indicates the tendency of an user to bid (1) or not to bid (0) on a real estate. In this dataset there are 12500 observations and 65 attributes. In the dataset there are missing values in particular the 4% of the rows has the URL attribute missing and the 38This dataset is characterized by the presence of several attributes with low information, for example the attribute "Inmarket_sportsitem" has 98% of the observations with value 1(yes) or the attribute "Inmarket_outdoor" with the percentage above the 99%,

this is solved thanks at the balancing process that we will perform before splitting the dataset.

The following columns have been removed under the assumption that they depends only on the single observation and consequently generate noise without bringing additional information. the columns are: "RowID", "UserID", "BidID", "IP" , "Domain", "URL", "Time_Bid", "AdslotID". In the dataset the columns "Browser", "Adslotvisibility' , "Adslotformat" have been converted using a label encoder.

2.3 Breast Cancer (Kaggle Data Set)

In this dataset every attribute is a measurement done during a medical exam. The attributes are all numerical. Examples of the attributes are: "concavePointsMean", "symmetryMean","fractalDimensionMean", "radiusStdErr", "textureStdErr". There are no missing values and we did not preprocesse any attribute. Due the domain of the attributes there are no low information columns so we have used them all in order to fit the classifiers.

3 Performance Tests

3.1 Model Training Parameters

We remind here the definition of the metric parameters we will used to quantify the performance of our classifiers i.e. precision (P), recall (R) and accuracy (A):

$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN} \qquad A = \frac{TP + TN}{\#all} \tag{1}$$

It is straightforward to calculate these parameters for binary classification tasks out of the confusion matrix. In case of multiple labels, we need to calculate these parameter for each class, given that:

- TPs are the values in the diagonal;
- FNs for a certain class are the sum of values in the corresponding row excluding the TP;
- FPs for a certain class are the sum of values in the corresponding column excluding the TP;
- TNs for a certain class are the sum of all rows and columns, excluding the class's column and row.

Another convenient metric, particularly because it is calculated directly from the proper *scikit-learn* function, is called $f1 - score$, which is defined as the harmonic mean of the precision and recall:

$$f1 - score = 2 \times \frac{P\dot{R}}{P + R} \tag{2}$$

3.2 Confusion Matrix

For each classifier, we produce a confusion matrix where each entry i, j corresponds to the number of observations in group i, but predicted to be in group j. We chose to normalize the entries according to the sum of each row. In case of binary classification, the matrix reduces to the number of true negatives (TN), false positives (FP), false negatives (FN) and true positives (TP).

Examples can be see in Fig. 1.

Describe micro averaging and macro averaging, will we report both ?

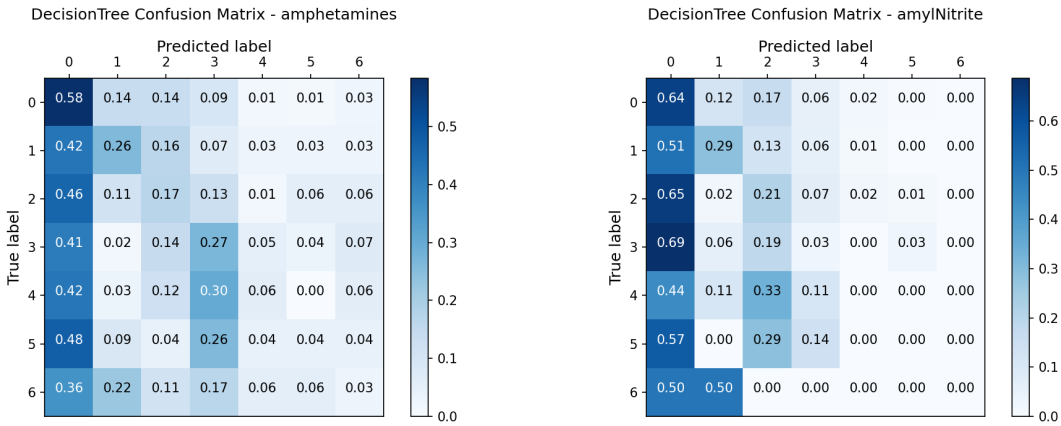

Figure 1: Example confusion matrices for REDO, just an example

3.3 Holdout and Cross Validation

Here we describe briefly the techniques of "holdout" and "cross validation" that are used to evaluate a model. The **holdout** method is essentially based on the splitting on the input data set into two subset, one used for training the model, and one used for testing the model, for example in $80\% - 20\%$ proportion, although there is not fix recipe for this split. Once the model is trained, the evaluation can be performed on the test data set, and this is therefore possible to check if the prediction of the model match the data. One big issue of this model is that the training strongly depend on the splitting of the initial data set, for example if the characteristic of the training data set are not representative of the whole data set.

A more powerful method is the **cross-validation** or "k-fold cross validation". Form the full dataset, a test is held out for final evaluation, but the validation set is no longer needed. For this, the training set is split into k sets, so that the model is trained using $k-1$ folds as training data, and the remaining one is used for the validation as a test set (to compute the interesting metric of the model). Once we obtain such k number of metrics, the final result is the average of these parameters, obtained for each iteration of the cross-validation on each distinct fold.

3.4 Class Balance

In this section we discuss the problem of imbalanced data. Whenever classes are not represented equally (imbalanced data), the result of the training and testing of the model might be biased. In particular, a metric such the accuracy might appear very satisfying while, in fact, it is only representing the underlying class distribution. The model in in fact favoured to learn the class with the majority of instances.

Fig. 2 shows the distribution of the various class attributes for our datasets. Starting with the binary classification problems we see that the "breast cancer" dataset is slightly imbalanced $(37,5\%\text{- }62,5\%)$ recurrence events vs non recurrence, while the imbalance is more pronounced for the other dataset, in particular we see $(17\%\text{-}83\%)$ hazardous vs non hazardous, and $(2\%\text{-}98\%)$ buy vs non buy classes.

In the case of the drug consumption dataset, the situation it is more complicated to analyse since it varies largely depending on the considered drug.

In order to balance the dataset, we implemented and algorithm that performs the following steps. First, it extracts the number of entries in the under-represented class (which by chance is always class "True" for the three binary classification tasks). Then we randomly select a number of rows from the remaining data of where class is equal to "False", adding an extra 10% to this total. Then, we create a final dataset combining the rows with the class "True" and the randomly selected "False". This step is done at a preliminary phase before any splitting into the training and testing subsets.

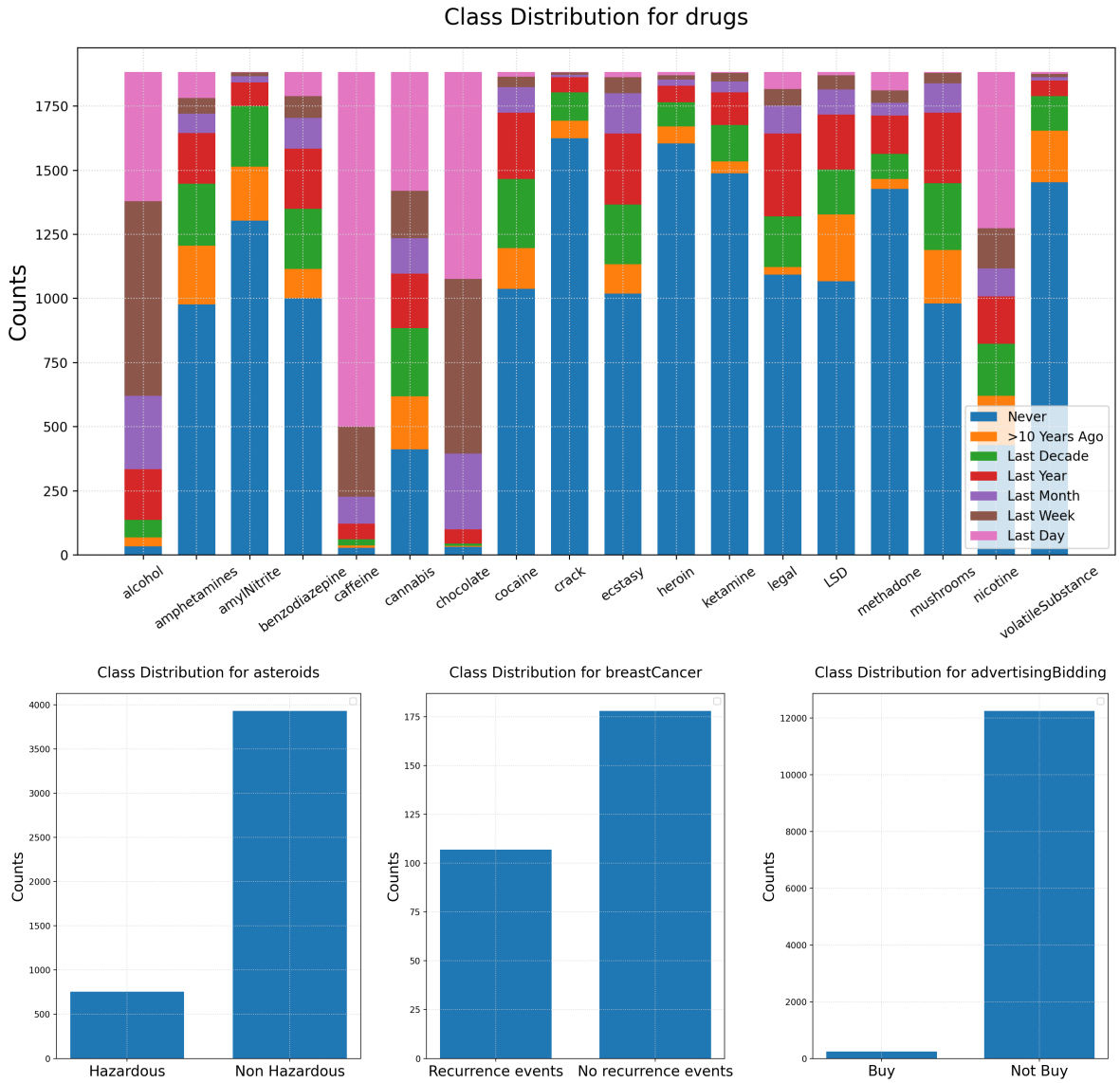

Figure 2: Distribution of classes for the target attributes for the multiclass classification (top panel, drug dataset) and binary classification (bottom panel, asteroids, breast cancer and advertising bidding datasets).

4 Comparison

4.1 Drug consumption

Overall the performance of the classification is not well. Looking at the confusion matrices we can see the expected influence of the datas imbalance. The overall fitting strategy is towards the most dominant class whereas the decision tree seems to be the most resistant against this bias (looking at the distribution around the main diagonal).

Decision Tree

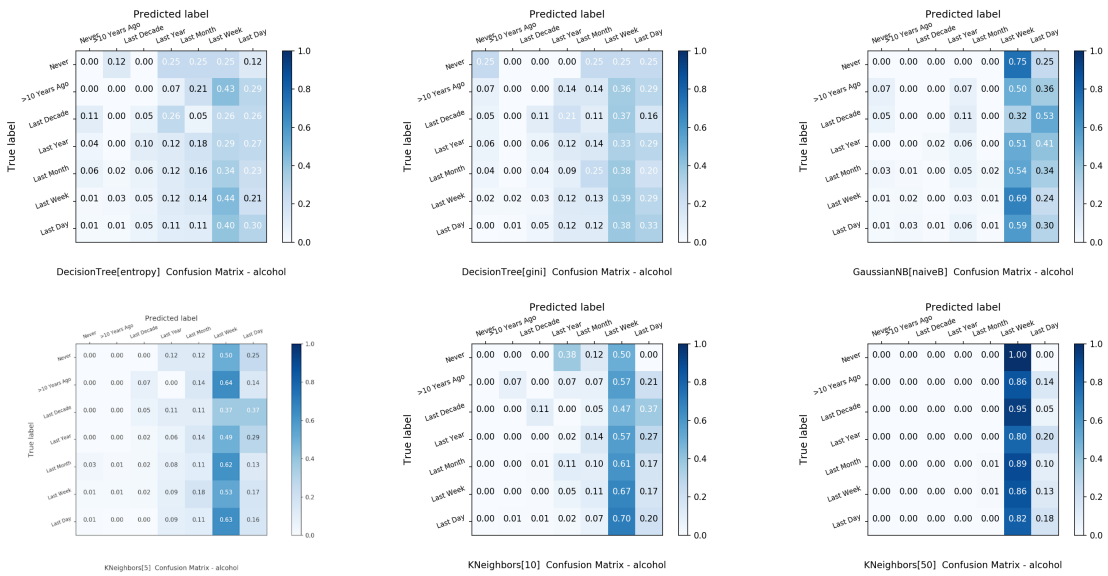

Figure 3: Confusion matrices for the used classifiers on the drug cocaine.

References

[1] E. Fehrman, A. K. Muhammad, E. M. Mirkes, V. Egan, and A. N. Gorban. The five factor model of personality and evaluation of drug consumption risk. 2017.