Machine Learning - SS 2021
Exercise 3: Synthetic Data

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

July 23, 2021

Contents

Introduction

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

1 Data Sets Description

Here we briefly introduce our four datasets, analyze the distributions of the input data and check for significant correlations between variables.

1.1 Income Data Set

The "income" data set[**?**] The distribution of the relevant features, as well as a scatter plot to highlight pair-wise correlations are shown in Fig. 5, while a more general view on the correlations are shown in Fig. 6.

1.2 Titanic Data Set

xxx

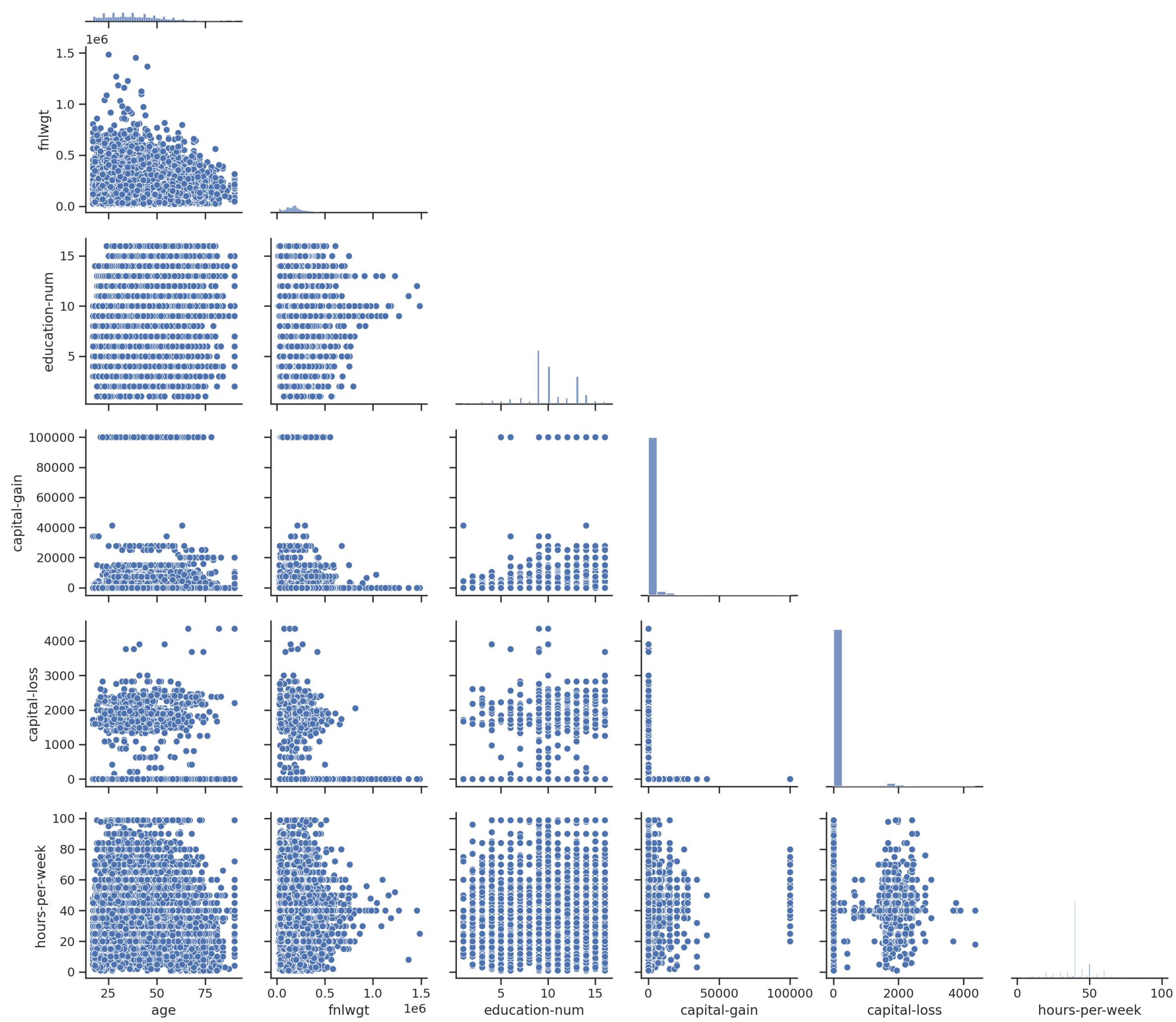

Figure 1: Distributions and pair-wise correlation for the features of the "income" data set.

1.3 Social Data Set

xxx

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.

2.1 Data Overview

Here we provide an overview of the original data set data.

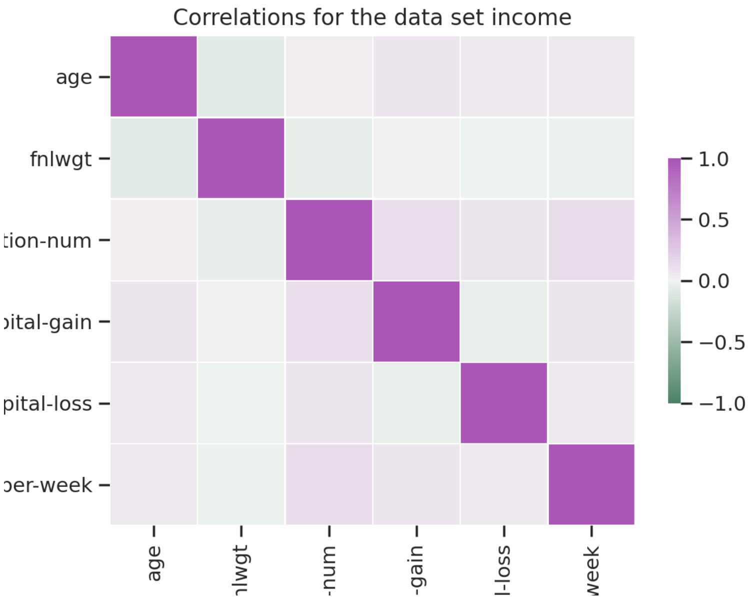

Figure 2: Correlation matrix for the "income" data set.

Missing Value The

Normalization The

Hot Encoding ? The

Outliers The

3 Generation of Synthetic Data

Here we briefly outlined the motivations and techniques used for the generation of synthetic data.

4 Model Implementation

4.1 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.

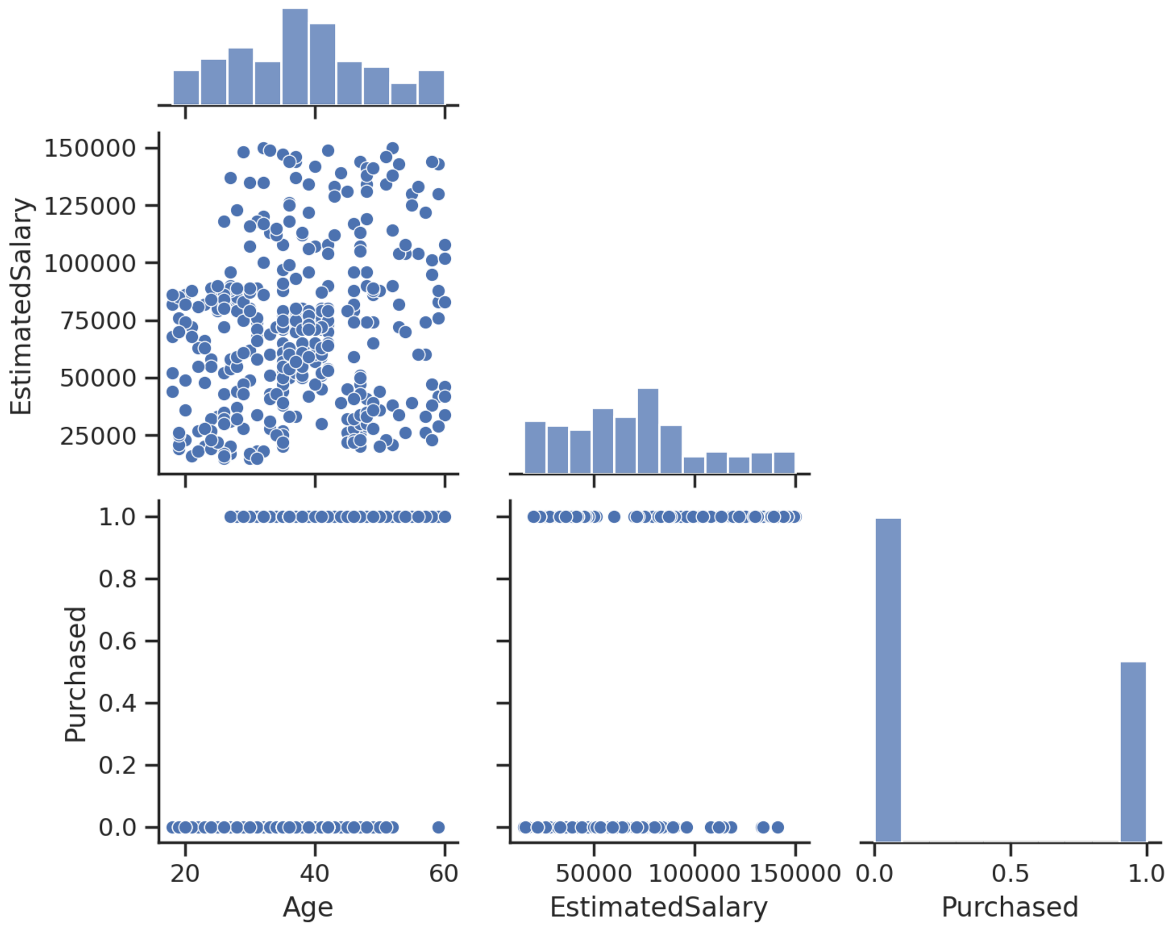

Figure 3: Distributions and pair-wise correlation for the features of the "social" data set.

5 Performance Tests

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. **??**.

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) , accuracy (A) and specificity S:

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

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:

1. TPs are the values in the diagonal;

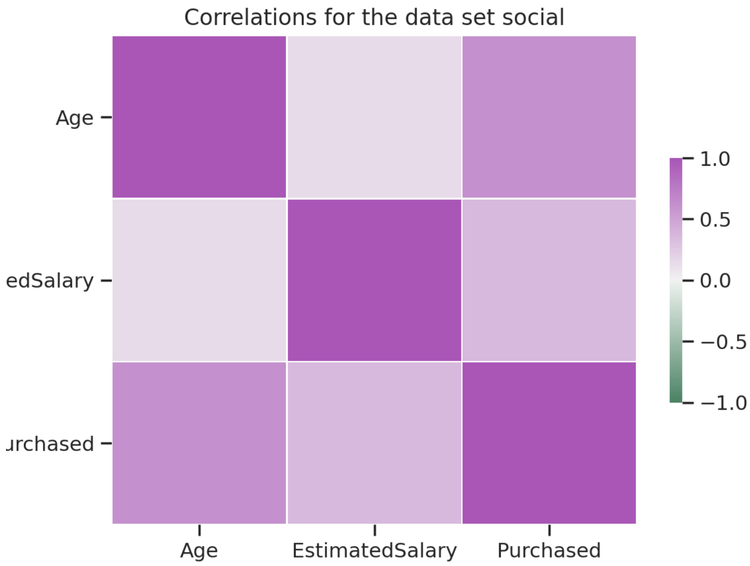

Figure 4: Correlation matrix for the "social" data set.

2. FNs for a certain class are the sum of values in the corresponding row excluding the TP;
3. FPs for a certain class are the sum of values in the corresponding column excluding the TP;
4. 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:

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

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

6 Conclusion

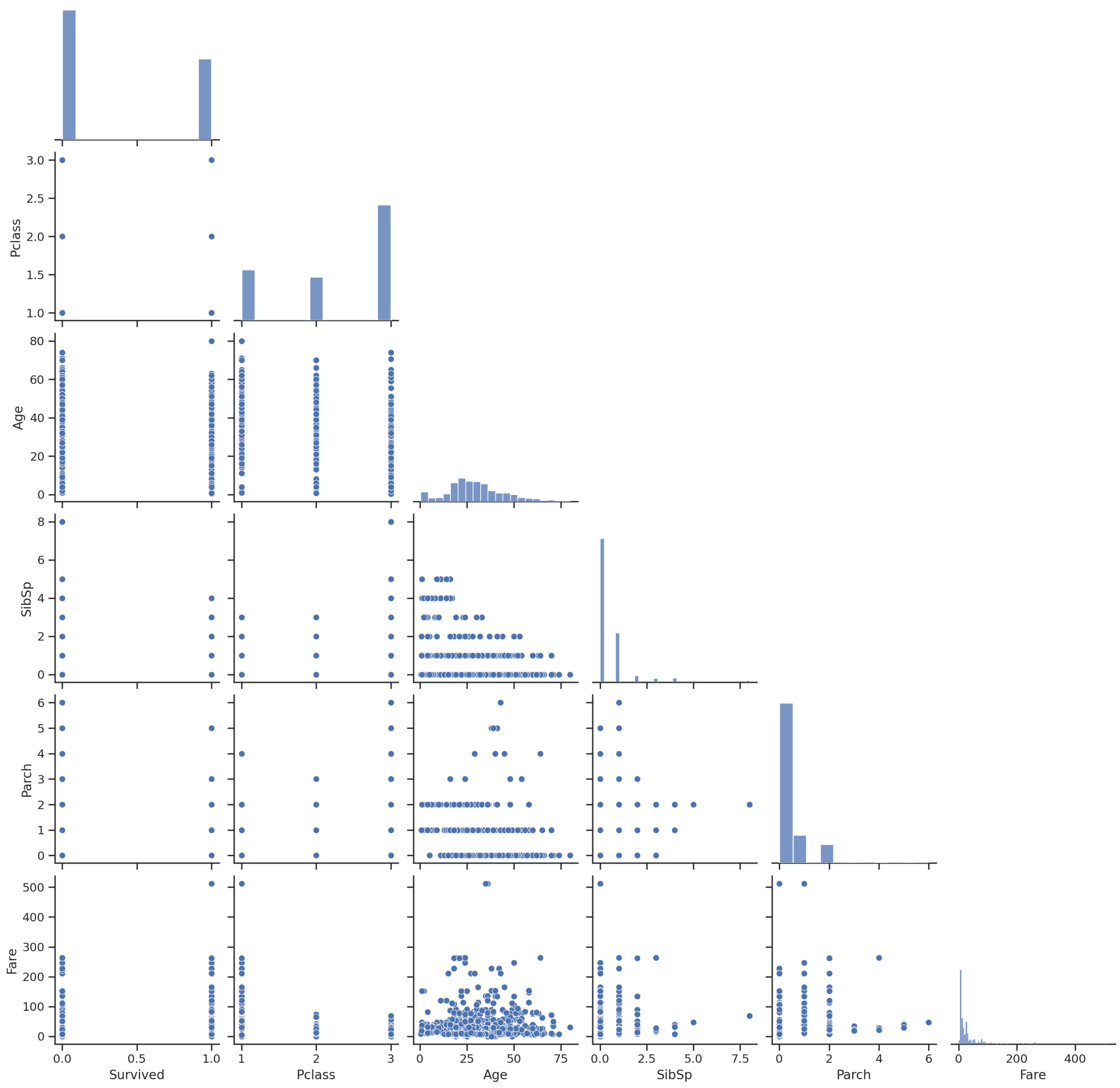

Figure 5: Distributions and pair-wise correlation for the features of the "Titanic" data set.

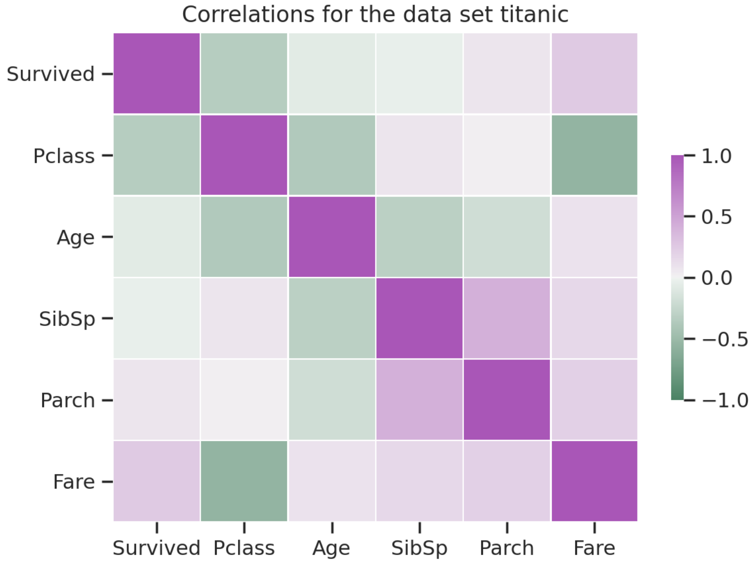

Figure 6: Correlation matrix for the "Titanic" data set.