Lab 5 Solution

April 22, 2024

1 LAB 05 - Python version

Luca Catalano, Daniele Rege Cambrin, Eleonora Poeta

1.0.1 Disclaimer

The purpose of creating this material is to enhance the knowledge of students who are interested in learning how to solve problems presented in laboratory classes using Python. This decision stems from the observation that some students have opted to utilize Python for tackling exam projects in recent years.

To solve these exercises using Python, you need to install Python (version 3.9.6 or later) and some libraries using pip or conda.

Here's a list of the libraries needed for this case:

- `os`: Provides operating system dependent functionality, commonly used for file operations such as reading and writing files, interacting with the filesystem, etc.
- `pandas`: A data manipulation and analysis library that offers data structures and functions to efficiently work with structured data.
- `numpy`: A numerical computing library that provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.
- `matplotlib.pyplot`: A plotting library for creating visualizations like charts, graphs, histograms, etc.
- `sklearn`: Machine learning algorithms and tools.
- `xlrd`: A Python library used for reading data and formatting information from Excel files (.xls and .xlsx formats). It provides functionality to extract data from Excel worksheets, including cells, rows, columns, and formatting details.

You can download Python from here and follow the installation instructions for your operating system.

For installing libraries using pip or conda, you can use the following commands:

- Using pip:

  ```
  pip install pandas numpy matplotlib ltk scikit-learn xlrd
  ```

- Using conda:

  ```
  conda install pandas numpy matplotlib scikit-learn xlrd
  ```

Make sure to run these commands in your terminal or command prompt after installing Python. You can also execute them in a cell of a Jupyter Notebook file (`.ipynb`) by starting the command with '!'.

2 Exercise 1

Import some libraries

[1]:
```
import pandas as pd

from sklearn.preprocessing import LabelEncoder
from sklearn.tree import export_text
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import cross_val_predict
from sklearn.metrics import confusion_matrix
```

2.1 Read file excel "user.xlsx"

To read the Excel file using a function integrated into the pandas library, you can use the `pd.read_excel()` function. Rewrite the instruction with the argument as the path of the file to be read

[2]:
```
# Read file excel
dataset = pd.read_excel("/Users/luca/Library/Mobile Documents/
 ↪com~apple~CloudDocs/Business Intelligence per Big Data/Laboratories/LAB05/
 ↪Lab5Materiale/user.xlsx")
```

```
/Users/luca/Library/Python/3.9/lib/python/site-
packages/openpyxl/styles/stylesheet.py:226: UserWarning: Workbook contains no
default style, apply openpyxl's default
  warn("Workbook contains no default style, apply openpyxl's default")
```

In a Jupyter Notebook cell, you can print a subset of the representation by simply calling the name of the variable containing the DataFrame.

[3]:
```
# print dataset
dataset
```

[3]:
```
      Age         Workclass    Education      Marital Status  \
0    39.0         State-gov    Bachelors       Never-married
1    50.0  Self-emp-not-inc    Bachelors  Married-civ-spouse
2    38.0           Private      HS-grad            Divorced
3    53.0           Private         11th  Married-civ-spouse
4    28.0           Private    Bachelors  Married-civ-spouse
..    …               …          …                …
995  56.0           Private      HS-grad  Married-civ-spouse
996  45.0           Private      Masters            Divorced
```

```
997  48.0       Federal-gov     Bachelors           Divorced
998  40.0           Private  Some-college  Married-civ-spouse
999  39.0      Self-emp-inc     Bachelors  Married-civ-spouse

            Occupation   Relationship   Race     Sex Native Country  Response
0         Adm-clerical  Not-in-family  White    Male  United-States  Negative
1      Exec-managerial        Husband  White    Male  United-States  Negative
2    Handlers-cleaners  Not-in-family  White    Male  United-States  Negative
3    Handlers-cleaners        Husband  Black    Male  United-States  Negative
4       Prof-specialty           Wife  Black  Female           Cuba  Negative
..                 …              …      …     …              …         …
995    Exec-managerial        Husband  White    Male  United-States  Positive
996     Prof-specialty  Not-in-family  White    Male  United-States  Negative
997    Exec-managerial      Unmarried  White    Male  United-States  Positive
998  Machine-op-inspct        Husband  White    Male  United-States  Negative
999    Exec-managerial        Husband  White    Male  United-States  Positive

[1000 rows x 10 columns]
```

2.2 Define the label column in the dataset data frame

Rename the 'Response' column to 'Label' [use dataset.rename(columns={'actual_col_name': 'new_col_name'})]

[4]:
```
# rename column Response to Label
dataset = dataset.rename(columns={'Response': 'Label'})
```

[5]:
```
# print datsaset to check if the column has been renamed
dataset
```

[5]:
```
      Age         Workclass     Education      Marital Status  \
0    39.0         State-gov     Bachelors       Never-married
1    50.0  Self-emp-not-inc     Bachelors  Married-civ-spouse
2    38.0           Private       HS-grad            Divorced
3    53.0           Private          11th  Married-civ-spouse
4    28.0           Private     Bachelors  Married-civ-spouse
..    …               …            …                 …
995  56.0           Private       HS-grad  Married-civ-spouse
996  45.0           Private       Masters            Divorced
997  48.0       Federal-gov     Bachelors            Divorced
998  40.0           Private  Some-college  Married-civ-spouse
999  39.0      Self-emp-inc     Bachelors  Married-civ-spouse

            Occupation   Relationship   Race     Sex Native Country     Label
0         Adm-clerical  Not-in-family  White    Male  United-States  Negative
1      Exec-managerial        Husband  White    Male  United-States  Negative
2    Handlers-cleaners  Not-in-family  White    Male  United-States  Negative
```

```
3    Handlers-cleaners        Husband  Black    Male  United-States  Negative
4       Prof-specialty           Wife  Black  Female           Cuba  Negative
..                 …                …     …       …               …        …
995    Exec-managerial        Husband  White    Male  United-States  Positive
996     Prof-specialty  Not-in-family  White    Male  United-States  Negative
997    Exec-managerial      Unmarried  White    Male  United-States  Positive
998  Machine-op-inspct        Husband  White    Male  United-States  Negative
999    Exec-managerial        Husband  White    Male  United-States  Positive

[1000 rows x 10 columns]
```

2.3 Separate the dataset into features, referred to as X, and labels, referred to as y. Afterwards, utilize Label Encoder to encode the categorical features.

[You can achieve this by selecting columns using the [] operator on the dataframe, then initializing the Label Encoder and applying its fit_transform method]

[6]:
```python
# Split the dataset into features (X) and target variable (y)
X = dataset.drop(columns=['Label'])  # Features
y = dataset['Label']  # Target variable



# Label encoding
labelencoder = LabelEncoder()
# Apply label encoding to each column, except for the age column
for column in X.columns:
    if column != 'Age':
        X[column] = labelencoder.fit_transform(X[column])
# print X
X
```

[6]:
```
      Age  Workclass  Education  Marital Status  Occupation  Relationship  \
0    39.0          5          9               4           0             1
1    50.0          4          9               2           3             0
2    38.0          2         11               0           5             1
3    53.0          2          1               2           5             0
4    28.0          2          9               2           9             5
..    …          …          …               …           …             …
995  56.0          2         11               2           3             0
996  45.0          2         12               0           9             1
997  48.0          0          9               0           3             4
998  40.0          2         15               2           6             0
999  39.0          3          9               2           3             0

     Race  Sex  Native Country
0       4    1              27
1       4    1              27
```

```
2        4    1              27
3        2    1              27
4        2    0               4
..     …   …              …
995      4    1              27
996      4    1              27
997      4    1              27
998      4    1              27
999      4    1              27

[1000 rows x 9 columns]
```

2.4 Use the decision tree classifier model.

Set these parameters:

- Criterion: 'entropy'
- Max Depth: 20
- Min Impurity Decrease: 0.001

[Use DecisionTreeClassifier() and its .fit function]

[7]:
```
# Initialize the Decision Tree Classifier
clf = DecisionTreeClassifier(criterion='entropy', max_depth=20,␣
 ↪min_impurity_decrease=0.001)
# Train the Decision Tree Classifier
clf.fit(X, y)
```

[7]:
```
DecisionTreeClassifier(criterion='entropy', max_depth=20,
                       min_impurity_decrease=0.001)
```

2.5 Print the structure of the decision tree

[use export_text(classifier_name, feature_names=list(x.columns))]

[8]:
```
# Print the structure of the decision tree
tree_structure = export_text(clf, feature_names=list(X.columns))
print(tree_structure)
```

```
|--- Marital Status <= 2.50
|   |--- Marital Status <= 1.50
|   |   |--- Education <= 10.50
|   |   |   |--- Workclass <= 0.50
|   |   |   |   |--- class: Positive
|   |   |   |--- Workclass >  0.50
|   |   |   |   |--- Education <= 8.50
|   |   |   |   |   |--- Age <= 42.50
|   |   |   |   |   |   |--- Age <= 34.50
|   |   |   |   |   |   |   |--- class: Negative
```

```
|   |   |   |   |   |   |--- Age >  34.50
|   |   |   |   |   |   |   |--- Age <= 35.50
|   |   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |   |   |   |--- Age >  35.50
|   |   |   |   |   |   |   |   |--- Age <= 40.00
|   |   |   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |   |   |   |--- Age >  40.00
|   |   |   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |   |--- Age >  42.50
|   |   |   |   |   |   |--- class: Negative
|   |   |   |   |--- Education >  8.50
|   |   |   |   |   |--- Education <= 9.50
|   |   |   |   |   |   |--- Race <= 3.00
|   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |   |   |--- Race >  3.00
|   |   |   |   |   |   |   |--- Workclass <= 2.50
|   |   |   |   |   |   |   |   |--- Age <= 38.50
|   |   |   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |   |   |   |--- Age >  38.50
|   |   |   |   |   |   |   |   |   |--- Age <= 44.50
|   |   |   |   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |   |   |   |   |   |--- Age >  44.50
|   |   |   |   |   |   |   |   |   |   |--- Age <= 55.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |   |   |   |   |   |--- Age >  55.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |--- Workclass >  2.50
|   |   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |--- Education >  9.50
|   |   |   |   |   |   |--- class: Positive
|   |   |--- Education >  10.50
|   |   |   |--- Age <= 42.50
|   |   |   |   |--- class: Negative
|   |   |   |--- Age >  42.50
|   |   |   |   |--- Age <= 44.50
|   |   |   |   |   |--- class: Negative
|   |   |   |   |--- Age >  44.50
|   |   |   |   |   |--- Occupation <= 2.50
|   |   |   |   |   |   |--- Occupation <= 1.00
|   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |   |--- Occupation >  1.00
|   |   |   |   |   |   |   |--- Relationship <= 2.50
|   |   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |   |   |   |--- Relationship >  2.50
|   |   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |--- Occupation >  2.50
|   |   |   |   |   |   |--- class: Negative
|   |--- Marital Status >  1.50
```

```
|   |   |--- Age <= 28.50
|   |   |   |--- Occupation <= 9.50
|   |   |   |   |--- Education <= 8.50
|   |   |   |   |   |--- class: Negative
|   |   |   |   |--- Education >  8.50
|   |   |   |   |   |--- Education <= 10.00
|   |   |   |   |   |   |--- Race <= 3.00
|   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |   |--- Race >  3.00
|   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |   |--- Education >  10.00
|   |   |   |   |   |   |--- Relationship <= 2.50
|   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |   |--- Relationship >  2.50
|   |   |   |   |   |   |   |--- Native Country <= 23.00
|   |   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |   |   |--- Native Country >  23.00
|   |   |   |   |   |   |   |   |--- Race <= 3.50
|   |   |   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |   |   |   |--- Race >  3.50
|   |   |   |   |   |   |   |   |   |--- class: Positive
|   |   |   |--- Occupation >  9.50
|   |   |   |   |--- class: Negative
|   |   |--- Age >  28.50
|   |   |   |--- Education <= 6.50
|   |   |   |   |--- Occupation <= 6.50
|   |   |   |   |   |--- Workclass <= 2.50
|   |   |   |   |   |   |--- Age <= 40.50
|   |   |   |   |   |   |   |--- Age <= 36.00
|   |   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |   |   |--- Age >  36.00
|   |   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |   |   |--- Age >  40.50
|   |   |   |   |   |   |   |--- Education <= 0.50
|   |   |   |   |   |   |   |   |--- Age <= 53.50
|   |   |   |   |   |   |   |   |   |--- Age <= 46.50
|   |   |   |   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |   |   |   |   |--- Age >  46.50
|   |   |   |   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |   |   |   |   |--- Age >  53.50
|   |   |   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |   |   |--- Education >  0.50
|   |   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |--- Workclass >  2.50
|   |   |   |   |   |   |--- Education <= 0.50
|   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |   |--- Education >  0.50
|   |   |   |   |   |   |   |--- Occupation <= 2.50
```

```
|   |   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |   |   |--- Occupation >  2.50
|   |   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |--- Occupation >  6.50
|   |   |   |   |   |--- class: Negative
|   |   |   |--- Education >  6.50
|   |   |   |   |--- Education <= 10.50
|   |   |   |   |   |--- Occupation <= 4.50
|   |   |   |   |   |   |--- Occupation <= 2.50
|   |   |   |   |   |   |   |--- Age <= 45.00
|   |   |   |   |   |   |   |   |--- Age <= 38.50
|   |   |   |   |   |   |   |   |   |--- Native Country <= 24.50
|   |   |   |   |   |   |   |   |   |   |--- Education <= 7.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |   |   |   |   |   |--- Education >  7.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |--- Native Country >  24.50
|   |   |   |   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |   |   |   |   |--- Age >  38.50
|   |   |   |   |   |   |   |   |   |--- Race <= 1.50
|   |   |   |   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |   |   |   |   |   |--- Race >  1.50
|   |   |   |   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |   |   |--- Age >  45.00
|   |   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |   |   |--- Occupation >  2.50
|   |   |   |   |   |   |   |--- Age <= 39.50
|   |   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |   |   |   |--- Age >  39.50
|   |   |   |   |   |   |   |   |--- Workclass <= 3.50
|   |   |   |   |   |   |   |   |   |--- Age <= 40.50
|   |   |   |   |   |   |   |   |   |   |--- Education <= 8.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |   |   |   |   |   |   |--- Education >  8.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |   |   |   |   |--- Age >  40.50
|   |   |   |   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |   |   |   |   |--- Workclass >  3.50
|   |   |   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |--- Occupation >  4.50
|   |   |   |   |   |   |--- Race <= 3.00
|   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |   |   |--- Race >  3.00
|   |   |   |   |   |   |   |--- Education <= 9.50
|   |   |   |   |   |   |   |   |--- Occupation <= 12.50
|   |   |   |   |   |   |   |   |   |--- Age <= 31.50
|   |   |   |   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |   |   |   |   |   |--- Age >  31.50
```

```
|   |   |   |   |   |   |   |   |   |   |--- Native Country <= 25.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |   |   |   |   |   |--- Native Country >  25.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 5
|   |   |   |   |   |   |   |   |--- Occupation >  12.50
|   |   |   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |   |   |--- Education >  9.50
|   |   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |--- Education >  10.50
|   |   |   |   |   |--- Education <= 11.50
|   |   |   |   |   |   |--- Native Country <= 8.50
|   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |   |   |--- Native Country >  8.50
|   |   |   |   |   |   |   |--- Race <= 3.00
|   |   |   |   |   |   |   |   |--- Age <= 63.50
|   |   |   |   |   |   |   |   |   |--- Occupation <= 3.50
|   |   |   |   |   |   |   |   |   |   |--- Age <= 46.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |   |--- Age >  46.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |   |   |   |   |--- Occupation >  3.50
|   |   |   |   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |   |   |   |--- Age >  63.50
|   |   |   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |   |   |   |--- Race >  3.00
|   |   |   |   |   |   |   |   |--- Age <= 64.50
|   |   |   |   |   |   |   |   |   |--- Workclass <= 0.50
|   |   |   |   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |   |   |   |   |   |--- Workclass >  0.50
|   |   |   |   |   |   |   |   |   |   |--- Occupation <= 8.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 10
|   |   |   |   |   |   |   |   |   |   |--- Occupation >  8.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 9
|   |   |   |   |   |   |   |   |--- Age >  64.50
|   |   |   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |--- Education >  11.50
|   |   |   |   |   |   |--- Occupation <= 12.50
|   |   |   |   |   |   |   |--- Age <= 32.50
|   |   |   |   |   |   |   |   |--- Occupation <= 10.00
|   |   |   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |   |   |   |--- Occupation >  10.00
|   |   |   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |   |   |   |--- Age >  32.50
|   |   |   |   |   |   |   |   |--- Age <= 76.50
|   |   |   |   |   |   |   |   |   |--- Workclass <= 0.50
|   |   |   |   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |   |   |   |   |   |--- Workclass >  0.50
|   |   |   |   |   |   |   |   |   |   |--- Occupation <= 2.50
```

```
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 5
|   |   |   |   |   |   |   |   |   |   |   |--- Occupation >  2.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 10
|   |   |   |   |   |   |   |   |   |--- Age >  76.50
|   |   |   |   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |   |   |--- Occupation >  12.50
|   |   |   |   |   |   |   |   |--- class: Negative
|--- Marital Status >  2.50
|   |--- Age <= 36.50
|   |   |--- Age <= 27.50
|   |   |   |--- class: Negative
|   |   |--- Age >  27.50
|   |   |   |--- Occupation <= 8.50
|   |   |   |   |--- class: Negative
|   |   |   |--- Occupation >  8.50
|   |   |   |   |--- Education <= 11.50
|   |   |   |   |   |--- class: Negative
|   |   |   |   |--- Education >  11.50
|   |   |   |   |   |--- Occupation <= 9.50
|   |   |   |   |   |   |--- Age <= 35.50
|   |   |   |   |   |   |   |--- Sex <= 0.50
|   |   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |   |   |--- Sex >  0.50
|   |   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |   |   |--- Age >  35.50
|   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |--- Occupation >  9.50
|   |   |   |   |   |   |--- class: Negative
|   |--- Age >  36.50
|   |   |--- Occupation <= 8.50
|   |   |   |--- Occupation <= 3.50
|   |   |   |   |--- Occupation <= 2.50
|   |   |   |   |   |--- class: Negative
|   |   |   |   |--- Occupation >  2.50
|   |   |   |   |   |--- Education <= 8.50
|   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |--- Education >  8.50
|   |   |   |   |   |   |--- Education <= 10.00
|   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |   |   |--- Education >  10.00
|   |   |   |   |   |   |   |--- Marital Status <= 4.50
|   |   |   |   |   |   |   |   |--- Native Country <= 16.00
|   |   |   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |   |   |   |--- Native Country >  16.00
|   |   |   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |   |   |   |--- Marital Status >  4.50
|   |   |   |   |   |   |   |   |--- class: Negative
|   |   |   |--- Occupation >  3.50
```

```
|   |   |   |   |--- class: Negative
|   |   |--- Occupation >  8.50
|   |   |   |--- Workclass <= 3.50
|   |   |   |   |--- Age <= 61.50
|   |   |   |   |   |--- Age <= 59.00
|   |   |   |   |   |   |--- Education <= 13.50
|   |   |   |   |   |   |   |--- Marital Status <= 4.50
|   |   |   |   |   |   |   |   |--- Age <= 40.50
|   |   |   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |   |   |   |   |--- Age >  40.50
|   |   |   |   |   |   |   |   |   |--- Workclass <= 0.50
|   |   |   |   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |   |   |   |   |   |--- Workclass >  0.50
|   |   |   |   |   |   |   |   |   |   |--- Age <= 53.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |--- Age >  53.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |   |   |--- Marital Status >  4.50
|   |   |   |   |   |   |   |   |--- class: Positive
|   |   |   |   |   |   |--- Education >  13.50
|   |   |   |   |   |   |   |--- class: Negative
|   |   |   |   |   |--- Age >  59.00
|   |   |   |   |   |   |--- class: Positive
|   |   |   |   |--- Age >  61.50
|   |   |   |   |   |--- class: Negative
|   |   |   |--- Workclass >  3.50
|   |   |   |   |--- class: Negative
```

2.6 Use the trained model on unseen data

Now that we have trained the model using the `fit` function, we can apply it to a dataset that the model hasn't seen before and evaluate its performance. [We'll use the variable `clf` that was declared previously (without redefining it) and apply the `predict` function to make predictions on the new dataset]

Another way to store the trained model for later reuse is by using serialization techniques such as `joblib` or `pickle`. These libraries allow you to save the trained model to a file, which can then be loaded and used whenever needed without having to retrain the model from scratch.

2.6.1 Load the new dataset "prospects.xlsx"

[10]:
```
# load the new dataset. [Use pd.read_excel() function to load the dataset. Use␣
↪the path of the file as an argument of the function.]
new_dataset = pd.read_excel("/Users/luca/Library/Mobile Documents/
↪com~apple~CloudDocs/Business Intelligence per Big Data/Laboratories/LAB05/
↪Lab5Materiale/prospect.xlsx")
```

```
/Users/luca/Library/Python/3.9/lib/python/site-
packages/openpyxl/styles/stylesheet.py:226: UserWarning: Workbook contains no
default style, apply openpyxl's default
  warn("Workbook contains no default style, apply openpyxl's default")
```

[11]:
```python
# print the new dataset
new_dataset
```

[11]:
```
        Age     Workclass     Education      Marital Status  \
0      25.0       Private       HS-grad       Never-married
1      46.0       Private           9th  Married-civ-spouse
2      37.0       Private       1st-4th  Married-civ-spouse
3      41.0       Private  Some-college  Married-civ-spouse
4      44.0       Private       HS-grad       Never-married
…       …            …            …                  …
28255  27.0       Private    Assoc-acdm  Married-civ-spouse
28256  40.0       Private       HS-grad  Married-civ-spouse
28257  58.0       Private       HS-grad             Widowed
28258  22.0       Private       HS-grad       Never-married
28259  52.0  Self-emp-inc       HS-grad  Married-civ-spouse

              Occupation Relationship                Race     Sex  \
0        Farming-fishing    Unmarried               White    Male
1          Other-service      Husband               White    Male
2           Craft-repair      Husband  Asian-Pac-Islander    Male
3           Craft-repair      Husband               White    Male
4           Adm-clerical    Own-child               White    Male
…                    …           …                  …      …
28255       Tech-support         Wife               White  Female
28256  Machine-op-inspct      Husband               White    Male
28257       Adm-clerical    Unmarried               White  Female
28258       Adm-clerical    Own-child               White    Male
28259    Exec-managerial         Wife               White  Female

      Native Country
0      United-States
1      United-States
2           Cambodia
3      United-States
4      United-States
…                …
28255  United-States
28256  United-States
28257  United-States
28258  United-States
28259  United-States
```

```
[28260 rows x 9 columns]
```

Please be mindful that in this scenario, we lack the variable "Label" (nor "Response"). As a matter of fact, we are unaware of the outcomes, yet we aim to forecast them using a model pre-trained on actual values.

2.7 Utilize Label Encoder to encode the categorical features.

[Rename the dataframeas X, then initializing the Label Encoder and applying the fit_transform method]

[12]:
```
X = new_dataset
# Label encoding for the new_dataset
labelencoder = LabelEncoder()
# Apply label encoding to each column, except for the age column
for column in X.columns:
    if column != 'Age':
        X[column] = labelencoder.fit_transform(X[column])
# print X
X
```

[12]:
```
        Age  Workclass  Education  Marital Status  Occupation  Relationship  \
0      25.0          3         11               4           4             4
1      46.0          3          6               2           7             0
2      37.0          3          3               2           2             0
3      41.0          3         15               2           2             0
4      44.0          3         11               4           0             3
…        …          …          …               …           …             …
28255  27.0          3          7               2          12             5
28256  40.0          3         11               2           6             0
28257  58.0          3         11               6           0             4
28258  22.0          3         11               4           0             3
28259  52.0          4         11               2           3             5

       Race  Sex  Native Country
0         4    1              38
1         4    1              38
2         1    1               0
3         4    1              38
4         4    1              38
…        …    …               …
28255     4    0              38
28256     4    1              38
28257     4    0              38
28258     4    1              38
28259     4    0              38

[28260 rows x 9 columns]
```

2.8 Apply the pretrained Decision Tree model

[13]:
```
# Predict the target variable of the new dataset
y_pred = clf.predict(X)
# print the prediction
y_pred
```

[13]:
```
array(['Negative', 'Negative', 'Negative', …, 'Negative', 'Negative',
       'Positive'], dtype=object)
```

3 Exercise 2

3.1 Read file excel "user.xlsx"

To read the Excel file using a function integrated into the pandas library, you can use the `pd.read_excel()` function. Rewrite the instruction with the argument as the path of the file to be read

[14]:
```
dataset = pd.read_excel("/Users/luca/Library/Mobile Documents/
 ↪com~apple~CloudDocs/Business Intelligence per Big Data/Laboratories/LAB05/
 ↪Lab5Materiale/user.xlsx")
```

```
/Users/luca/Library/Python/3.9/lib/python/site-
packages/openpyxl/styles/stylesheet.py:226: UserWarning: Workbook contains no
default style, apply openpyxl's default
  warn("Workbook contains no default style, apply openpyxl's default")
```

In a Jupyter Notebook cell, you can print a subset of the representation by simply calling the name of the variable containing the DataFrame.

[15]:
```
# print dataset
dataset
```

[15]:
```
      Age         Workclass     Education      Marital Status  \
0    39.0         State-gov     Bachelors       Never-married
1    50.0  Self-emp-not-inc     Bachelors  Married-civ-spouse
2    38.0           Private       HS-grad            Divorced
3    53.0           Private          11th  Married-civ-spouse
4    28.0           Private     Bachelors  Married-civ-spouse
..    …                 …           …                 …
995  56.0           Private       HS-grad  Married-civ-spouse
996  45.0           Private       Masters            Divorced
997  48.0       Federal-gov     Bachelors            Divorced
998  40.0           Private  Some-college  Married-civ-spouse
999  39.0      Self-emp-inc     Bachelors  Married-civ-spouse

           Occupation   Relationship   Race     Sex Native Country  Response
0        Adm-clerical  Not-in-family  White    Male  United-States  Negative
```

```
1      Exec-managerial        Husband  White    Male  United-States  Negative
2    Handlers-cleaners  Not-in-family  White    Male  United-States  Negative
3    Handlers-cleaners        Husband  Black    Male  United-States  Negative
4       Prof-specialty           Wife  Black  Female           Cuba  Negative
..                 …              …      …       …             …        …
995    Exec-managerial        Husband  White    Male  United-States  Positive
996     Prof-specialty  Not-in-family  White    Male  United-States  Negative
997    Exec-managerial      Unmarried  White    Male  United-States  Positive
998  Machine-op-inspct        Husband  White    Male  United-States  Negative
999    Exec-managerial        Husband  White    Male  United-States  Positive

[1000 rows x 10 columns]
```

3.2 Define the label column in the dataset data frame

Rename the 'Response' column to 'Label' [use dataset.rename(columns={'actual_col_name': 'new_col_name'})]

[16]:
```
# rename column Response to Label
dataset = dataset.rename(columns={'Response': 'Label'})
```

[17]:
```
# print datsaset to check if the column has been renamed
dataset
```

[17]:
```
      Age          Workclass     Education      Marital Status  \
0    39.0          State-gov     Bachelors       Never-married
1    50.0   Self-emp-not-inc     Bachelors  Married-civ-spouse
2    38.0            Private       HS-grad            Divorced
3    53.0            Private          11th  Married-civ-spouse
4    28.0            Private     Bachelors  Married-civ-spouse
..    …                  …             …                  …
995  56.0            Private       HS-grad  Married-civ-spouse
996  45.0            Private       Masters            Divorced
997  48.0        Federal-gov     Bachelors            Divorced
998  40.0            Private  Some-college  Married-civ-spouse
999  39.0       Self-emp-inc     Bachelors  Married-civ-spouse

            Occupation   Relationship   Race     Sex Native Country     Label
0         Adm-clerical  Not-in-family  White    Male  United-States  Negative
1      Exec-managerial        Husband  White    Male  United-States  Negative
2    Handlers-cleaners  Not-in-family  White    Male  United-States  Negative
3    Handlers-cleaners        Husband  Black    Male  United-States  Negative
4       Prof-specialty           Wife  Black  Female           Cuba  Negative
..                 …              …      …       …             …        …
995    Exec-managerial        Husband  White    Male  United-States  Positive
996     Prof-specialty  Not-in-family  White    Male  United-States  Negative
997    Exec-managerial      Unmarried  White    Male  United-States  Positive
```

```
998  Machine-op-inspct         Husband  White    Male  United-States  Negative
999    Exec-managerial         Husband  White    Male  United-States  Positive

[1000 rows x 10 columns]
```

3.3 Separate the dataset into features, referred to as X, and labels, referred to as y. Afterwards, utilize Label Encoder to encode the categorical features.

[You can achieve this by selecting columns using the [] operator on the dataframe, then initializing the Label Encoder and applying the fit_transform method]

[18]:
```python
# Split the dataset into features (X) and target variable (y)
X = dataset.drop(columns=['Label'])  # Features
y = dataset['Label']   # Target variable



# Label encoding
labelencoder = LabelEncoder()
# Apply label encoding to each column, except for the age column
for column in X.columns:
    if column != 'Age':
        X[column] = labelencoder.fit_transform(X[column])
# print X
X
```

[18]:
```
      Age  Workclass  Education  Marital Status  Occupation  Relationship  \
0    39.0          5          9               4           0             1
1    50.0          4          9               2           3             0
2    38.0          2         11               0           5             1
3    53.0          2          1               2           5             0
4    28.0          2          9               2           9             5
..    …         …         …              …          …           …
995  56.0          2         11               2           3             0
996  45.0          2         12               0           9             1
997  48.0          0          9               0           3             4
998  40.0          2         15               2           6             0
999  39.0          3          9               2           3             0

     Race  Sex  Native Country
0       4    1              27
1       4    1              27
2       4    1              27
3       2    1              27
4       2    0               4
..    …   …             …
995     4    1              27
996     4    1              27
```

```
997       4     1                 27
998       4     1                 27
999       4     1                 27

[1000 rows x 9 columns]
```

3.4 Validation of Decision Tree classification model using Cross Validation

Cross-validation is a technique used to assess the performance and generalization ability of machine learning models, particularly in the context of classification tasks. It involves partitioning the dataset into multiple subsets, known as folds.

1. **Partitioning the Dataset**: The dataset is divided into k equal-sized folds.
2. **Training and Testing**: The model is trained k times, each time using k-1 folds for training and the remaining fold for testing.
3. **Evaluation**: The performance of the model is evaluated on each fold, and the results are averaged to obtain a robust estimate of the model's performance.
4. **Advantages**: Cross-validation provides a more reliable estimate of the model's performance compared to a single train-test split. It helps to detect overfitting and assesses the model's ability to generalize to unseen data.

[Use `cross_val_score` and `cross_val_predict` to perform cross-validation easily. Follow the same instruction of Exercise 1 to initialise and use the model]

Set these parameters for Decision Classfier model:

- Criterion: 'entropy'
- Max Depth: 25
- Min Impurity Decrease: 0.01

[19]:
```python
# Initialize the decision tree classifier
clf = DecisionTreeClassifier(criterion='entropy', max_depth=25,␣
↪min_impurity_decrease=0.01)

# Perform cross-validation predictions
y_pred = cross_val_predict(clf, X, y, cv=5)

# Calculate confusion matrix
conf_matrix = confusion_matrix(y, y_pred)

# Evaluate accuracy
accuracy = accuracy_score(y, y_pred)
# Print accuracy
print("Accuracy:", accuracy)

# Print confusion matrix
conf_matrix = pd.DataFrame(conf_matrix, columns=['Predicted No', 'Predicted␣
↪Yes'], index=['Actual No', 'Actual Yes'])
```

```
conf_matrix
```

Accuracy: 0.808

[19]:

	Predicted No	Predicted Yes
Actual No	730	38
Actual Yes	154	78