Data Science and Computer Programming

Project

Introduction

In the “Data Science and Computer Programming” course at the National Taiwan Normal University, I chose a data set from the Internet for the project in order to extract valuable information and present it in an appealing and easy-to-understand form. The goal is to discover patterns and correlations in the data and present them clearly through visualizations such as charts and graphs.

Dataset

I downloaded the dataset called “Predict Students' Dropout and Academic Success” from the UC Irvine Machine Learning Repository. The dataset consists of 36 columns and over 4200 entries. Since I don't need all 36 columns for the project, I thought about which data I can work well with.

The following columns have emerged:

- **Marital status:** Single or Married
- **Application order:** Number between 0 (first choice) - 9 (last choice)
- **Course:** Number that refers to the course name
- **Nationality:** Nationality of the students
- **Mother's qualification:** Educational qualification of the mother
- **Father's qualification:** Educational qualification of the father
- **Displaced:** Is displaced or not
- **Gender:** Male or Female
- **Scholarship holder:** Has a scholarship or not
- **Age at enrollment:** Age of the student
- **Target:** Dropout, Enrolled or Graduate

Data manipulation

Right at the beginning, I deleted the rows in which the students were neither single nor married. As this only accounts for a very small proportion, it has no further major influence on the work.

However, because the data set consists almost exclusively of numbers, I converted these into the corresponding text.

Example 1:

The following code snippet shows how the ones are changed to “Single” and the rest (only twos) to “Married”.

```
"""Changes column to Single/Married (text) instead of 1/2 (num)"""

data["Marital status"] = data["Marital status"].apply(
                                lambda x: "Single" if x == 1 else "Married")
```

Example 2:

As this is a bit more extensive, I have written a function that converts the corresponding number into the corresponding text. Due to the large number of lines, I have only included the first few here as an example.

```
def get_country(num):
    try:
        num = int(num)
    except ValueError:
        num = -1
    match num:
        case -1:
            return "CastingError"
        case 1:
            return "Portuguese"
        case 2:
            return "German"
        case 6:
            return "Spanish"
        case 11:
            return "Italian"
        case 13:
            return "Dutch"
        Case _:
            return "UNKNOWN"
```

Graphics

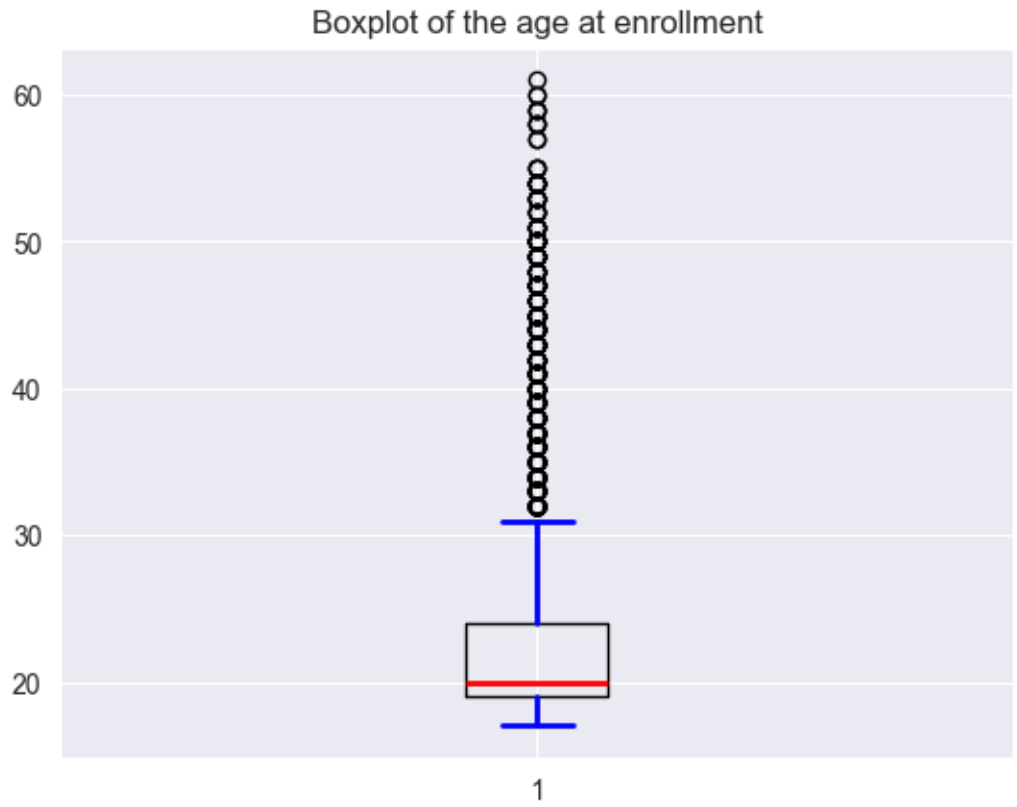
Again, I will only show a small section of the code.

Example 1:

This code generates a boxplot of the distribution of the students' ages.

```
"""Prints a boxplot to show the age at enrollment"""
plt.boxplot(data["Age at enrollment"],
        whiskerprops=dict(color='blue', linewidth=2),
        capprops=dict(color='blue', linewidth=2),
        medianprops=dict(color='red', linewidth=2))
plt.title('Boxplot of the age at enrollment')
plt.grid(True)
plt.show()
```

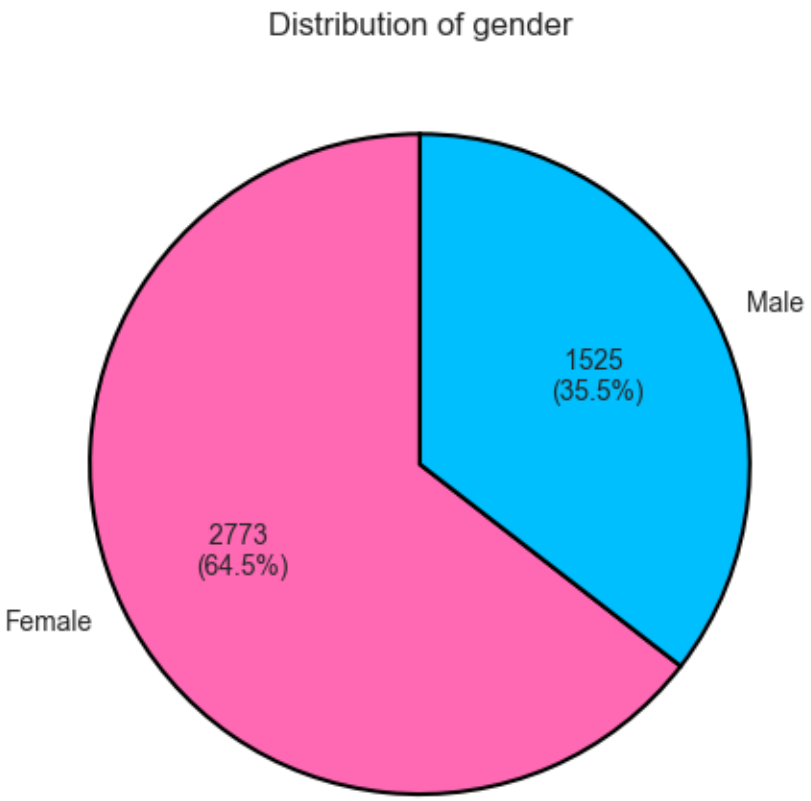
This boxplot shows the age distribution of students, with the majority aged between 20 and 30. The red line inside the box marks the median, which is 20.



Example 2:

The following pie chart shows the distribution of students between men and women.

```
"""Shows a pie chart of the students' gender"""
pieChartGender = data["Gender"].value_counts()
colors = ['#FF69B4' if x == 'Female' else '#00BFFF'
          for x in pieChartGender.index]
pieChartGender.plot(kind='pie', startangle=90, colors=colors,
                    wedgeprops={'edgecolor': 'black', 'linewidth': 1.5},
                    autopct=lambda pct: f"{
                    int(pct / 100. * sum(data["Gender"].value_counts()))
                    } \n({pct:.1f}%)")
plt.title('Distribution of gender')
plt.xlabel('')
plt.ylabel('')
plt.tight_layout()
plt.show()
```



Example 3:

I have omitted the code snippet for the following example due to its length, but I would still like to discuss the result.

The code creates a crosstab from the data sets “Scholarship holder” and “Target” and calculates the χ^2 test and the p-value.

The result is as follows:

	Dropout	**Graduate**	**All**
No Scholarship	1234	1347	2581
Scholarship	129	816	945
All	1263	2163	3526

χ^2-Value: 338.96

p-Value: 1.0746e-75

The value for χ^2 is approximately 340, which is a fairly high value and indicates a possible dependency between the two variables. In addition, the p-value is extremely low and approaches 0, which means that the null hypothesis can be rejected.

This means nothing other than that there is a dependency between scholarship and degree.

In simple terms, this means that if you receive a scholarship, you are more likely to graduate.

Link to the GitHub-Page:

https://silashage.github.io/Data-Science-and-Computer-Programming/ or see

https://github.com/silashage