m3_case_study_2

October 22, 2024

0.0.1 Module 3 – OOP Packages Modules Try-Except

0.1 Case Study – 2

0.1.1 Approach to Solve the Problem

In this task, we need to create a program that helps Bank of Portugal optimize their marketing campaign by focusing on eligible clients based on their profession. The task can be broken down into the following steps:

1. Read the CSV file (bank-data.csv) containing client data.
2. Extract and build a set of unique professions from the dataset.
3. Input a profession from the user (tele-caller).
4. Check if the input profession is in the set of eligible professions.
5. Output whether the client is eligible to be approached for the campaign based on their profession.

[9]:
```
import pandas as pd

# Step 1: Read the CSV file
bank_data = pd.read_csv('bank-data.csv')

print(bank_data.info())
print("\n--------------------\n")
print(bank_data.head())
print("\n--------------------\n")



# Step 2: Build a set of unique professions
unique_professions = set(bank_data['job'].unique())
print('Set of unique professions',str(len(unique_professions)), ␣
 ↪unique_professions)

# Step 3: Function to check eligibility
def check_eligibility(profession):
    if profession in unique_professions:
        print(f"The profession '{profession}' is eligible for the marketing␣
 ↪campaign.")
    else:
        print(f"The profession '{profession}' is not eligible for the marketing␣
 ↪campaign.")

# Step 4: Get profession input from the user
input_profession = input("Enter the profession of the client: ").strip().lower()

# Step 5: Check if the profession is eligible
check_eligibility(input_profession)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 447 entries, 0 to 446
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   age      447 non-null    int64
 1   job      447 non-null    object
 2   marital  447 non-null    object
 3   y        447 non-null    object
dtypes: int64(1), object(3)
memory usage: 14.1+ KB
None

-------------------


   age          job  marital    y
0   20      student   single  yes
1   32   management   single  yes
2   49   technician  married  yes
3   32  blue-collar  married  yes
4   33   management  married  yes

-------------------


Set of unique professions 9 {'technician', 'student', 'blue-collar', 'admin.',
'entrepreneur', 'self-employed', 'management', 'services', 'housemaid'}

Enter the profession of the client:  student

The profession 'student' is eligible for the marketing campaign.
```

0.1.2 Here's an enhanced version of the code that includes the following features:

1. Compute max and min age for loan eligibility based on the data in the CSV file.
2. Store max and min age in a dictionary.
3. Make the profession check case insensitive.
4. Keep taking input in a while loop, and end only if the user types "END" for the profession.

[10]:
```
import pandas as pd
```

```
# Step 1: Read the CSV file
bank_data = pd.read_csv('bank-data.csv')

print(bank_data.info())
print("\n--------------------\n")
print(bank_data.head())
print("\n--------------------\n")

# Step 2: Build a set of unique professions (case-insensitive)
unique_professions = set(bank_data['job'].str.lower().unique())

# Step 3: Compute max and min age for loan eligibility based on the dataset
max_age = bank_data['age'].max()
min_age = bank_data['age'].min()

# Store max and min age in a dictionary
age_eligibility = {
    "min_age": min_age,
    "max_age": max_age
}

print("age_eligibility", age_eligibility)

# Function to check eligibility
def check_eligibility(profession, age):
    profession = profession.lower()  # Making the profession check␣
 ↪case-insensitive
    if profession in unique_professions:
        # Check if the age is within the eligibility range
        if age_eligibility['min_age'] <= age <= age_eligibility['max_age']:
            print(f"The profession '{profession}' is eligible for the marketing␣
 ↪campaign.")
        else:
            print(f"The profession '{profession}' is eligible, but age {age} is␣
 ↪not in the eligible range ({min_age}-{max_age}).")
    else:
        print(f"The profession '{profession}' is not eligible for the marketing␣
 ↪campaign.")

# Step 4: Start a while loop to continuously ask for input
while True:
    # Get profession and age input from the user
    input_profession = input("Enter the profession of the client (type 'END' to␣
 ↪stop): ").strip()

    if input_profession.upper() == "END":
        print("Program terminated.")
```

```python
        break  # Exit the loop if the user types "END"

    try:
        input_age = int(input("Enter the age of the client: "))
    except ValueError:
        print("Invalid age input. Please enter a valid integer for age.")
        continue  # Skip to the next iteration of the loop

    # Check if the profession is eligible and age is within the range
    check_eligibility(input_profession, input_age)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 447 entries, 0 to 446
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   age      447 non-null    int64 
 1   job      447 non-null    object
 2   marital  447 non-null    object
 3   y        447 non-null    object
dtypes: int64(1), object(3)
memory usage: 14.1+ KB
None

-------------------


   age          job  marital    y
0   20      student   single  yes
1   32   management   single  yes
2   49   technician  married  yes
3   32  blue-collar  married  yes
4   33   management  married  yes

-------------------


age_eligibility {'min_age': np.int64(19), 'max_age': np.int64(80)}

Enter the profession of the client (type 'END' to stop):  Student
Enter the age of the client:  23

The profession 'student' is eligible for the marketing campaign.

Enter the profession of the client (type 'END' to stop):  Artist
Enter the age of the client:  23

The profession 'artist' is not eligible for the marketing campaign.

Enter the profession of the client (type 'END' to stop):  student
Enter the age of the client:  17
```

```
The profession 'student' is eligible, but age 17 is not in the eligible range
(19-80).

Enter the profession of the client (type 'END' to stop):  END

Program terminated.
```

```
[8]: #### Mr Akram M'Tir 12/22-10-2024
```