Module 3: Functions, OOP, modules, Errors & Exceptions

Case Study II

Domain – Banking Marketing

Business challenge/requirement

Bank of Portugal runs marketing campaign to offer loans to clients. Loan is offered to only clients with particular professions. List of successful campaigns (with client data) is given in attached dataset. You have to come up with program which reads the file and builds a set of unique profession list and given input profession of client –system tells whether client is eligible to be approached for marketing campaign.

Key issues
Tele Caller can only make x number of cold calls in a day. Hence to increase her effectiveness only eligible customers should be called

Considerations
Current system does not differentiate clients based on age and profession
Data volume: 447 records in bank-data.csv

Business benefits
Company can achieve between 15% to 20% higher conversion by targeting right clients

Approach to Solve
You have to use fundamentals of Python taught in module 3

1. Read file bank-data.csv
2. Build a set of unique jobs
3. Read the input from command line –profession
4. Check if profession is in list
5. Print whether client is eligible

Enhancements for code
You can try these enhancements in code

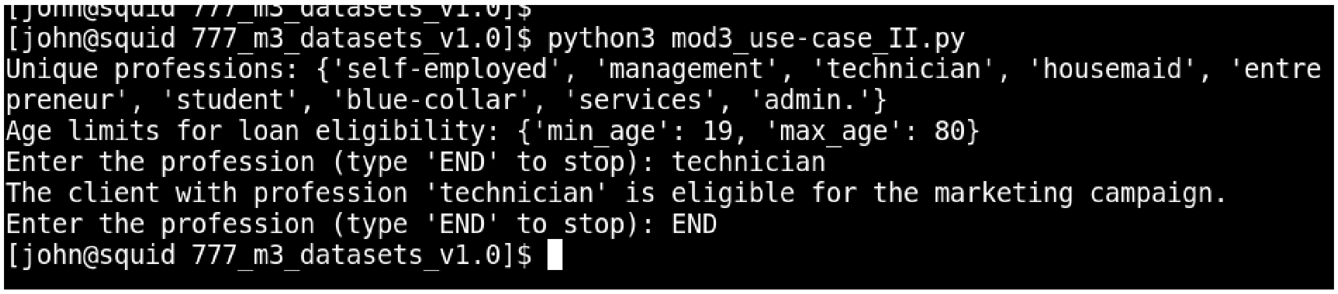
1. Compute max and min age for loan eligibility based on data in csv file
2. Store max and min age in dictionary
3. Make the profession check case insensitive
4. Currently program ends after the check. Take the input in while loop and end only if user types "END" for profession

```
import csv

def read_csv(file_name):
    with open(file_name, mode='r') as file:
        csv_reader = csv.DictReader(file)
        data = [row for row in csv_reader]
    return data

def get_unique_jobs(data):
    return set(row['job'].lower() for row in data)

def get_age_limits(data):
    ages = [int(row['age']) for row in data]
    return {'min_age': min(ages), 'max_age': max(ages)}

def check_eligibility(profession, unique_jobs):
    return profession.lower() in unique_jobs

def main():
    file_name = 'bank-data.csv'
    data = read_csv(file_name)
    unique_jobs = get_unique_jobs(data)
    age_limits = get_age_limits(data)
    
    print(f"Unique professions: {unique_jobs}")
    print(f"Age limits for loan eligibility: {age_limits}")
    
    while True:
        profession = input("Enter the profession (type 'END' to stop): ")
        if profession == "END":
            break
        if check_eligibility(profession, unique_jobs):
            print(f"The client with profession '{profession}' is eligible for the marketing campaign.")
        else:
            print(f"The client with profession '{profession}' is not eligible for the marketing campaign.")
        
if __name__ == "__main__":
    main()
```

Here's how the code works:

1. **Read the CSV File**: The `read_csv` function reads the `bank-data.csv` file and stores each row as a dictionary in a list.

2. **Get Unique Jobs**: The `get_unique_jobs` function extracts unique job titles from the data and stores them in a set (making it case insensitive by converting to lowercase).
3. **Get Age Limits**: The `get_age_limits` function calculates the minimum and maximum ages from the data and returns them as a dictionary.
4. **Check Eligibility**: The `check_eligibility` function checks if the given profession is in the list of unique jobs.
5. **Main Function**: The `main` function reads the data, gets the unique jobs and age limits, and then enters a loop to continuously check if entered professions are eligible until the user types "END".



```
[john@squid 777_m3_datasets_v1.0]$
[john@squid 777_m3_datasets_v1.0]$ python3 mod3_use-case_II.py
Unique professions: {'self-employed', 'management', 'technician', 'housemaid', 'entre
preneur', 'student', 'blue-collar', 'services', 'admin.'}
Age limits for loan eligibility: {'min_age': 19, 'max_age': 80}
Enter the profession (type 'END' to stop): technician
The client with profession 'technician' is eligible for the marketing campaign.
Enter the profession (type 'END' to stop): END
[john@squid 777_m3_datasets_v1.0]$ 
```