Project 1: Loan Application

- Project starts on: **Tuesday, Feb. 2nd**
- Project is due on: **Monday, Feb. 15th**
- This is an individual project

**Useful material**

- This is a useful tutorial on how to use functions: [Functions](#).
- Overview on Booleans, conditions, and If/else statements: [Nested-if](#)

- Please read the entire handout before you start coding.
- Did you spot an error in this document or you think you can improve it? Please send me an e-mail.
  (mzahran@purdue.edu)

**IMPORTANT:** When you submit your project, **PLEASE CHECK** to ensure that your submission has been accepted. If you do not do this, and the submission fails, you will get a zero for the project because there is nothing to grade. The graders are unable to handle late submissions.

**You are allowed to discuss about the project in a high level manner with your classmates. However, you cannot show your code to other students.** We will use MOSS before grading this project. **If a plagiarism or cheating is detected, you will be reported to Dean of Students.**

**Introduction**

When you apply for a loan, you submit an application that contains some financial information about you such as: payment history, amount of funds owed, length of credit history, number of credit cards owned and annual income. Then, the application is reviewed by possible lenders as banks. In the review process they calculate a credit score based on your financial information provided in the application. If the credit score is above certain threshold value, they grant you the loan, if not they decline your loan application.

A bank hired you to implement the decision process for loan applications, which means you are required to write a Python program to read the financial information from applicants and calculate the credit score, then show whether the applicant is granted the loan or not.
Project skeleton

Your project skeleton is skeleton. Download this file and start working on it!

Loan Application details

For each application, the bank is interested in the following financial information, which we will call factors:

1. **Factor 1: Payment history**: Whether the applicant paid past credit accounts on time. E.g. yes
2. **Factor 2: Amount of funds owed**: In dollars, How much money currently the applicant owes. E.g. 3500 dollars
3. **Factor 3: Length of credit history**: In months, How long ago was the start of the applicant's credit history. E.g. 23 months
4. **Factor 4: Number of loan applications rejected**: How many times the applicant was rejected for a loan by any bank. E.g. 2 times
5. **Factor 5: Annual Income**: In dollars, how much money the applicant make in a year. E.g. 50000 dollars per year

Each factor will get a number of points depending on its value, the points of all factors will be used in the calculation of the final credit score.

The following table shows the conversion of the factor values to points:

<table>
<thead>
<tr>
<th>Factor</th>
<th>Value ranges</th>
<th>FPoints</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Payment history</td>
<td>yes</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>no</td>
<td>0</td>
</tr>
<tr>
<td>2 Amount of funds owed</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>1 to 999</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>1000 to 4999</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>5000 or more</td>
<td>0</td>
</tr>
<tr>
<td>3 Length of credit history</td>
<td>0 to 11</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>12 to 47</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>48 or more</td>
<td>100</td>
</tr>
<tr>
<td>4 Number of loan applications rejected</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>1 to 4</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>5 or more</td>
<td>0</td>
</tr>
<tr>
<td>5 Annual Income</td>
<td>0 to 4999</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>5000 to 9999</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>10000 or more</td>
<td>100</td>
</tr>
</tbody>
</table>

A simple way to get the credit score is to add the points of all factors together, But since there are some factors more important than others, we want to give them more **weight** in the final calculation of the credit score. It's up to the bank to decide the important factors and assign weights accordingly, so that the final credit score will be calculated as follows:

\[
\text{credit score} = \text{FPoints1}\times\text{weight1} + \text{FPoints2}\times\text{weight2} + \text{FPoints3}\times\text{weight3} + \text{FPoints4}\times\text{weight4} + \ldots
\]
FPoints5*weight5

If the credit score is greater than or equal to a certain value let's call it “Threshold” then this application is accepted, otherwise it's rejected.

The weights and the threshold are **fixed values** provided to you in the project skeleton.

**Examples**

Consider the following examples:

<table>
<thead>
<tr>
<th>Factors</th>
<th>Applicant 1</th>
<th>Applicant 1 Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Payment history</td>
<td>yes</td>
<td>100</td>
</tr>
<tr>
<td>2 Amount of funds owed</td>
<td>1500</td>
<td>30</td>
</tr>
<tr>
<td>3 Length of credit history</td>
<td>12</td>
<td>80</td>
</tr>
<tr>
<td>4 Number of loan applications rejected</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>5 Annual Income</td>
<td>6000</td>
<td>50</td>
</tr>
<tr>
<td>For w1=w2=w3=w4=w5=0.2, threshold=50</td>
<td></td>
<td>Credit score = 52 Decision: ACCEPTED</td>
</tr>
<tr>
<td>For w1=0.1, w2=0.25, w3=0.15, w4=0.3, w5=0.2, threshold=50</td>
<td></td>
<td>Credit score = 39.5 Decision: REJECTED</td>
</tr>
<tr>
<td>For w1=0.1, w2=0.25, w3=0.15, w4=0.3, w5=0.2, threshold=30</td>
<td></td>
<td>Credit score = 39.5 Decision: ACCEPTED</td>
</tr>
<tr>
<td>For w1=0.15, w2=0.2, w3=0.25, w4=0.05, w5=0.35, threshold=60</td>
<td></td>
<td>Credit score = ? Decision: REJECTED</td>
</tr>
</tbody>
</table>

Let's calculate the credit score for the previous applicant using the following values for weights and threshold:

**w1=w2=w3=w4=w5=0.2, threshold=50**

- credit score = 0.2*100 + 0.2*30 + 0.2*80 + 0.2*0 + 0.2*50 = 52 and since 52>50, then applicant 1 is ACCEPTED.

**w1=0.1, w2=0.25, w3=0.15, w4=0.3, w5=0.2, threshold=50**

- credit score = 0.1*100 + 0.25*30 + 0.15*80 + 0.3*0 + 0.2*50 = 39.5 and since 39.5<50, then applicant 1 is REJECTED.

**w1=0.1, w2=0.25, w3=0.15, w4=0.3, w5=0.2, threshold=30**

- credit score = 0.1*100 + 0.25*30 + 0.15*80 + 0.3*0 + 0.2*50 = 39.5 and since 39.5>30, then applicant 1 is ACCEPTED.

**For w1=0.15, w2=0.2, w3=0.25, w4=0.05, w5=0.35, threshold=60**

- As an exercise calculate the credit score manually and make sure the decision will be REJECTED.

**Other examples:**
## Tasks

The following is the skeleton of the project, marked with the tasks you should do. The main of the project should be like this:

```
# Tasks Summary:
#1) Read form the user the number of applications (n).
#2) Loop n times, in each iteration call the function makeDecision, then print decision returned by the function.
#3) Inside the function makeDecision: Read the factors of a new application from the user.
#4) Inside the function makeDecision: Calculate the points of each factor.
#5) Inside the function makeDecision: Using the factor points and the weights, calculate the credit score.
#6) Inside the function makeDecision: Using the threshold and the credit score return whether the application is ACCEPTED or REJECTED
```

#PARAMETERS:
#1)w1: weight for factor 1, Type: float
#2)w2: weight for factor 2, Type: float
#3) w3: weight for factor 3, Type: float
#4) w4: weight for factor 4, Type: float
#5) w5: weight for factor 5, Type: float
#6) threshold: credit score acceptance threshold, Type: float

#RETURNS:
#1) decision: the final decision, Type: string, either 'ACCEPTED' or 'REJECTED'

def makeDecision(w1, w2, w3, w4, w5, threshold):

    ###########################
    #Task 3
    #######
    #read application
    #TODO

    ###########################
    #Task 4
    #######
    #Calculate the points of each factor
    #TODO

    ###########################
    #Task 5
    #######
    #Calculate the credit score
    #TODO

    ###########################
    #Task 6
    #######
    #Make decision either accepted or rejected, and return the decision
    #TODO

def main():
    #Weights and threshold are given below.
    w1 = 0.1
    w2 = 0.25
    w3 = 0.15
    w4 = 0.3
    w5 = 0.2
    threshold = 50

    ###########################
    #Task 1
    #######
    #read the number of applications (n).
    #TODO

    ###########################
    #Task 2
    #######
#Loop n times
#TODO
    #call makeDecision  
    #TODO  
    #print the application status (decision).  
    #TODO
    ################################################################################
main()

Sample input/output for the project

Enter the number of applications: 2

Reading Application

Payment history: yes
Amount of funds owed: 1500
Length of credit history: 12
Number of loan applications rejected: 7
Annual Income: 6000
Decision: REJECTED

Reading Application

Payment history: yes
Amount of funds owed: 500
Length of credit history: 23
Number of loan applications rejected: 2
Annual Income: 50000
Decision: ACCEPTED

Final words: Please STICK to the given skeleton, any alteration with the skeleton will incur losing points.

Setting up the environment

Go to your working directory in “data.cs.purdue.edu” and create a directory “cs177/project1”. Refer to the lab1 if you need to remember the steps to do so.

Turnin Instructions

Run putty and login to data.cs.purdue.edu. Turn in your project by typing:

data 51 $ cd cs177
data 51 $ turnin -v -c cs177=COMMON -p project1 project1
Grading Rubric

This project is worth 100 points.

<table>
<thead>
<tr>
<th>TODO</th>
<th>MAX POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>5</td>
</tr>
<tr>
<td>Task 2</td>
<td>15</td>
</tr>
<tr>
<td>Task 3</td>
<td>25</td>
</tr>
<tr>
<td>Task 4</td>
<td>25</td>
</tr>
<tr>
<td>Task 5</td>
<td>20</td>
</tr>
<tr>
<td>Task 6</td>
<td>5</td>
</tr>
<tr>
<td>Skeleton</td>
<td>5</td>
</tr>
</tbody>
</table>

From: http://courses.cs.purdue.edu/ - Computer Science Courses

Permanent link: http://courses.cs.purdue.edu/cs17700:spring16:projects:project1

Last update: 2016/02/09 15:52