Optimal Average Grade Calculator

Created	@August 21, 2024 12:50 PM
Tags	

Cómo calcular el promedio mínimo a obtenerse en las próximas k intancias de evaluación para alcanzar cierta meta de promedio académico general.

averageScore.py

- *Se posee una lista de las n notas registradas hasta el momento.*

```
grades = [ 8.00, 9.00, 10.0, 10.0, 10.0, 8.50,
                     8.50, 8.00, 8.20, 9.00, 9.88, 8.00,
                     9.00, 10.0, 8.50, 8.80, 9.00, 9.00,
                     9.00, 8.50, 9.00, 9.20, 10.0, 7.00,
                     8.50, 9.00, 8.25, 8.50, 8.50, 10.0 ]

n = len(grades)   # 35 notas registradas
```

- *En base al listado puede calcularse el promedio actual:*

$$AvgGrade = \frac{\sum_{i=1}^{n} g_i}{n}$$

```
def getAverage(grades):
    n = len(grades)
    return sum(grades)/n
# Promedio actual: 314.83/35 = 8.99
```

- *La idea del programa, como lo dice el título, es calcular el promedio mínimo que debe obtenerse en las siguientes k instancias de evaluación para alcanzar un objetivo planteado previamente.*
  La función que lleva esto a cabo sigue la siguiente lógica:
  - *k: número de instancias de evaluación en las que se quiere alcanzar el objetivo. Parámetro aportado por el usuario.*
  - *GoalAvgGrade: meta de promedio general en el rango [1-10]. Parámetro aportado por el usuario.*

$$GoalAvgGrade \leq \frac{\sum_{i=1}^{n+k} g_i}{n+k}$$

Separando las listas de notas obtenemos la siguiente expresión:

$$GoalAvgGrade \leq \frac{\sum_{i=1}^{n} g_i}{n+k} + \frac{\sum_{j=1}^{k} g_j}{n+k}$$

El primer término puede calcularse a partir de la función anterior modificada:

$$\frac{\sum_{i=1}^{n} g_i}{n+k} = getAverage(grades, k) = Avg'$$

El segundo término puede simplificarse como la sumatoria de k notas iguales g':

$$\frac{\sum_{j=1}^{k} g_j}{n+k} = \frac{k \cdot g'}{n+k}$$

Reemplazando en la expresión inicial:

$$GoalAvgGrade \leq Avg' + \frac{k \cdot g'}{n+k}$$

Despejando la incógnita:

$$g' = \frac{(GoalAvgGrade - Avg') \cdot (n+k)}{k}$$

  - *g' : valor buscado. Si el usuario obtiene un promedio de g' en las próximas k instancias, alcanzará su objetivo.*
  - *La modificación de getAverage() consta de agregar un parámetro offset inicializado en 0 por defecto. Así puede lograrse dualidad de modo en la función, evitando duplicar código.*

```
def getAverage(grades,offset=0):
    total = len(grades) + offset   # n+k
    return sum(grades)/total
```

El método *getMinTryAverage()* lleva a cabo el cálculo detallado anteriormente:

```
def getMinTryAverage(grades, k, goalAvgGrade):
    n = len(grades)
    A' = getAverage(grades,k)
    result = ((goalAvgGrade - A') * (n+k)) / k
    if result>0 and result<=10:    # La meta es alcanzable en k
        return result
    return None                    # La meta es inalcanzable en
```

Ejemplo

- *Se propone el siguiente caso:*
  - *El usuario tiene un historial de notas idéntico al que se detalla al principio.*
  - *El usuario tiene un promedio meta de **9.00**.*
  - *El usuario quiere saber cuánto debe sacarse en su próximo exámen para alcanzar su meta (si es que es posible hacerlo).*

```
k = 1
goalAvg = 9.00
grades = [ 8.00, 9.00, 10.0, 10.0, 10.0,
                   8.50, 8.00, 8.20, 9.
                   9.00, 10.0, 8.50, 8.
                   9.00, 8.50, 9.00, 9.
                   8.50, 9.00, 8.25, 8.

g = getMinTryAverage(grades,k,goalAvg)
print(g)
```

```
Your current average is:  8.894 .
You wish to attain a goal of 9.0 in 1 try.
Sorry, it appears that your goal is not possible.
```

El objetivo no es posible. Si el usuario sacara 10, que es la máxima nota, aún así su promedio general se encontraría por debajo de 9.00.

```
k = 4
g = getMinTryAverage(grades,k,goalAvg
print(g)
```

```
Your current average is: 8.894 .
You wish to attain a goal of 9.0 in 4 tries.
You need an average of 9.792 to achieve such goal.
```

Recién con 4 intentos es posible alzanzar el objetivo. El usuario debería sacarse tres 10s para lograrlo.

```
updated_grades = grades + [8,10,10,10
k = 6
g = getMinTryAverage(grades,k,goalAvg
```

```
Your current average is: 8.995 .
You wish to attain a goal of 9.0 in 6 tries.
You need an average of 9.027 to achieve such goal.
```

Supongamos que faltaron cargar algunas notas. Se las agrega. Se quiere calcular g' para las seis materias del cuatrimestre. La meta es alcanzable para el usuario.