Réalisé par : BONNET Julie, ENJALBERT Anthony et FROMENT Jean-François
Supervisé par : M. WEHRLE

RAPPORT

Régression linéaire par descente de gradient

B.U.T Informatique 2ème année
SAE S4.D.01
Outil d'aide à la détermination du prix de vente d'un appartement

SOMMAIRE

INTRODUCTION

Dans le cadre de la SAE du quatrième semestre de notre formation, le projet de réalisation d'un "Outil d'aide à la détermination du prix de vente d'un appartement" nous a été confié. Ce projet nous a été donné par L. WEHRLE, notre enseignant de l'IUT de Rodez.

L'objectif du projet est la détermination d'un modèle permettant le calcul du prix de vente d'un appartement, à partir d'un jeu de données. La recherche du modèle se fait grâce à deux méthodes : par résolution analytique ou par descente de gradient.

L'outil est codé en python et la base de donnée est un fichier txt. Les participants au projet sont : Anthony ENJALBERT, Jean-François FROMENT et Julie BONNET.

I - PRÉSENTATION DE L'ALGORITHME 1

Code source de la méthode principale :

```
def algol(indiX, indiY, indiXY) :

    #Calcul de a = covarianceXY/varianceX
    a = indiXY[0]/indiX[2]

    #Calcul de b = moyenneY-a*moyenneX
    b = indiY[0]-a*indiX[0]

    return a,b
```

En suivant la méthode analytique, on cherche le moment où le gradient s'annule. On obtient un minimum pour a = covariance(x,y)/variance(x) et b = la moyenne de y - a * moyenne de x.

De nombreux indicateurs sont nécessaires pour cet algorithme. Ils sont calculé séparément, dans une autre méthode, de la manière suivante :

```
def calculIndicateurs() : #Méthode utile pour l'affichage des indicateurs et pour l'algol

    #Initialisations des variables
    indicateursX = []  # Moyenne des x, médiane des x, variance de x, écartType de x
    indicateursY = []  # Moyenne des y, médiane des y, variance de y, écartType de y
    indicateursXY = [] # Covariance, 'coefficient de corrélation linéaire'
    sommeAll = 0       # Permet de calculer la somme des x et y
    sommeQuadraX = 0   # Sert à faire la somme des x au carré
    sommeQuadraY = 0   # Sert à faire la somme des y au carré
    #Fin des initialisations

    #Calculs concernant les surfaces(valeur x) du jeu de données
    #Moyenne des x
    moyenneX = sum(surface_appartements)/len(surface_appartements)
    indicateursX.append(moyenneX) #Ajout au tableau

    #Médiane des x
    tailleListe = len(surface_appartements)   #Méthode qui donne la taille de la liste
    listeTriee = sorted(surface_appartements) #Méthode qui trie la liste
    if (tailleListe % 2 == 0) :
        Xmedian = (listeTriee[tailleListe//2-1] + listeTriee[tailleListe//2])/2
    else :
        Xmedian = listeTriee[(tailleListe//2)]
    indicateursX.append(Xmedian) #Ajout au tableau

    #Moyenne quadratique des x
    for i in range(len(surface_appartements)) :
        #Somme des Xi au carré
        sommeQuadraX = sommeQuadraX + surface_appartements[i]**2
    moyenneQuadraX = sommeQuadraX/len(surface_appartements)

    #Moyenne des x au carré
    moyenneXCarre = moyenneX**2

    #Variance de x
    varianceX = moyenneQuadraX-moyenneXCarre
    indicateursX.append(varianceX) #Ajout au tableau

    #Ecart type de x
    ecartTypeX = math.sqrt(varianceX) #math.sqrt permet le calcul de la racine carrée
    indicateursX.append(ecartTypeX) #Ajout au tableau
```

[Le code source n'est pas complet pour ne pas prendre trop de place]

Ainsi, la méthode algo1() récupère les tableaux contenant les indicateurs calculés par la méthode calculIndicateurs(). Elle sélectionne ensuite dans les tableaux seulement les indicateurs qui sont utiles pour la méthode analytique.

Nous avons choisi de faire une méthode à part pour calculer les indicateurs car ils nous sont utiles pour plusieurs méthodes, et de cette façon on ne les calcule qu'une seule fois.

II - PRÉSENTATION DE L'ALGORITHME 2

Code source de la méthode principale :

```
def algo2(a, b, pas):
    isFini = True
    compteur = 0
    while(isFini):
        valA = functionA(a, b)
        valB = functionB(a, b)
        if(abs(valA) <= 0.001 and abs(valB) <= 0.001):
            isFini = False
        if compteur == 10000000 :
            isFini = False
        a = a-valA*pas
        b = b-valB*pas
        compteur = compteur + 1
    return a,b
```

Pour cet algorithme qui utilise la descente de gradient, nous avons besoin de calculer les dérivées de nos fonctions. Ces calculs sont effectués séparément dans deux autres méthodes.
En cas de trop nombreuses répétitions, l'algorithme s'arrêtera et proposera une solution. Tant que la convergence n'est pas établie à un epsilon près, l'algorithme continue.

```
# Fonction derivée à partir de a
def functionA(a, b):
    global sommeCarreX
    global sommeX
    global sommeXY
    deriver = 2*(a*sommeCarreX+b*sommeX-sommeXY)
    return deriver

# Fonction derivée à partir de b
def functionB(a, b):
    global sommeX
    global sommeY
    deriver = 2*(a*sommeX+len(surface_appartements)*b-sommeY)
    return deriver
```

Pour calculer la dérivée de a ou de b on récupère certaines sommes calculées simplement dans le programme principal. Le calcul des dérivées pose ensuite aucun problème, on a :

$$\frac{\partial f}{\partial a}((a,b)) = 2\left(a\sum_{i=1}^{n} x_i^2 + b\sum_{i=1}^{n} x_i - \sum_{i=1}^{n} x_i y_i\right)$$

$$\frac{\partial f}{\partial b}((a,b)) = 2\left(a\sum_{i=1}^{n} x_i + nb - \sum_{i=1}^{n} y_i\right)$$

III - EVALUATION ET BILANS

A - Evaluation concertée, sincère et objective de l'apport par étudiant

Dès le départ, nous avons fait en sorte de répartir le travail de façon à peu près équitable. Toute l'équipe à participé à la V1 de la conception, puis nous nous sommes divisé les tâches :

- Julie s'est occupée de la méthode de lecture de fichiers et de la mise en place du drive (journal de temps, squelette du livrable, diagramme au propre, …).
- Anthony à travaillé sur l'Algorithme 1.
- Jean François à commencé les recherches sur l'Algorithme 2.

Une fois ces tâches réalisées, nous avons revu la conception et répartit les nouvelles tâches :

- Julie à fusionné et optimisé l'algorithme 1, les calculs d'indicateurs, la méthode permettant le prix de vente d'un appartement, … dans un seul et même fichier. Un menu a été ajouté pour l'utilisateur.
- Anthony et Jean François ont travaillé sur l'Algorithme 2.

Une fois l'Algorithme 2 terminé, nous nous sommes répartis de nouvelles tâches :

- Anthony à travaillé sur la mise en place du graphique, l'optimisation de l'Algorithme 2 et l'exécutable.
- Julie et Jean François ont travaillé sur l'optimisation du programme Outil, sur le travail facultatif "affichage des indicateurs" et sur la rédaction du livrable.

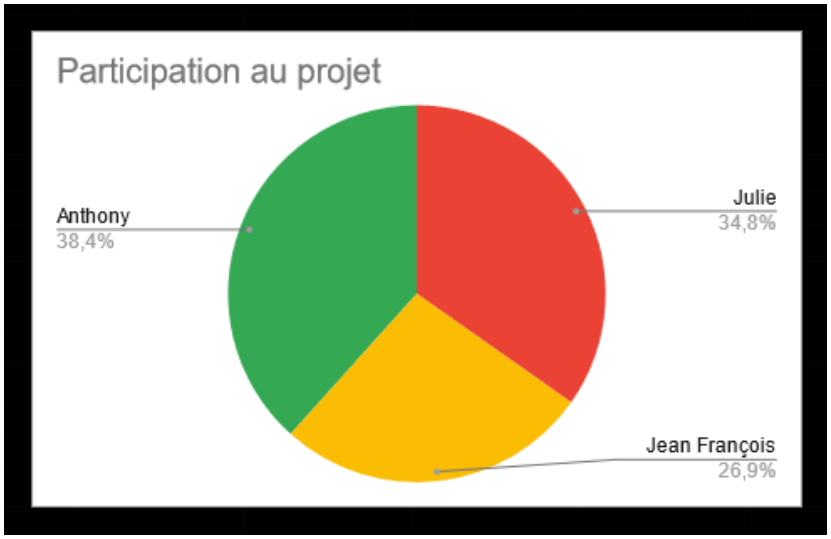

Diagramme représentant la répartition des temps de travail

Temps total : 23h et 15 minutes

Période de travail : du 7 au 15 février et le 20 mars pour le rendu

B - Bilans individuels

1. Anthony

Ce projet m'a plu puisque, tout d'abord, lorsque nous avons appris la descente de gradient en cours d'optimisation, j'avais bien apprécié travailler dessus. Donc, programmer ceci en Python était assez ludique pour moi. Ensuite, sur la partie résolution analytique, cela reprenait les bases de ce que nous avions vu au semestre 2 (écart-type, variance, médiane, …). Il y a eu un bon esprit d'équipe, nous nous sommes bien répartis les tâches ce qui fait que nous avons pu finir le projet à temps. Nous avons eu quelques problèmes au niveau de l'algorithme 2 (résolution par descente de gradient), mais nous avons pu corriger cela grâce aux conseils de M. WEHRLE. Nous avons également eu des problèmes à la fin du projet pour créer l'exécutable, vu qu'il fallait inclure les différents modules ainsi

que python pour que l'utilisateur qui récupère le projet puisse l'exécuter sans avoir à rien installer sur son PC hormis notre projet. Grâce à PyInstaller, nous avons pu régler ce problème.

2. Jean François

Le projet s'est bien déroulé. Les ressources que nous avons à disposition (les cours) m'ont permis de comprendre et réaliser le projet en équipe. La partie mathématiques du projet, notamment la descente de gradient ne m'ont pas subjugué, heureusement pour moi qu'il y avait la situation concrète des appartements pour apporter un côté plus enrichissant. Ce côté maths m'a causé certaines difficultés surtout pour la descente de gradient. Enfin, l'ambiance dans l'équipe était bonne et du bon travail a été effectué.

3. Julie

J'ai beaucoup apprécié travailler sur ce projet, surtout car on a réussi à réaliser ce qui avait à faire dans les délais et qu'on a même pu faire des points facultatifs. Je suis satisfaite de ce que j'ai pu produire, autant sur la partie mathématiques que sur la partie programmation, même si j'ai eu du mal à comprendre l'algorithme 2. J'ai aussi beaucoup apprécié travailler sur la programmation de l'outil et du menu. Je pense m'être améliorée en python grâce à ce projet.

C - Bilan collectif

L'ensemble de l'équipe est satisfaite de ce qui a été produit, surtout car nous avons fini dans les délais (voir en avance) et que nous avons pu réaliser de nombreuses parties facultatives. Nous avons atteint nos objectifs en termes de délais et de qualité, certainement grâce à notre bonne cohésion de groupe et notre organisation (par paires).
De plus, tous les membres sont satisfaits car ils ont gagné en compétence, que ça soit en programmation python, en mathématiques, en rédaction de livrables, en travail collaboratif, …