

MANCHE Marie-Amandine

Master 2 SE

Année 2020-21

Rapport de projet : Machine learning & Brain tumor



Enseignants :

Dubois Jérôme

Grassia Filippo

Sommaire

Introduction.....	p3
I/ Analyses prédictives des algorithmes de régression.....	p4
1)Dataset RFE	p4
2)Analyse prédictive	p4
II/ Etude des IRM du dataset brats.....	p5
1)Définition.....	p5
2)Programmation sur Colab.....	p6
III/Analyse de chaque partie de la tumeur.....	p7
1)Volume de la tumeur entière	p7
2)Volume de la partie « enhancing tumor ».....	p8
3)Volume de la nécrose.....	p8
4)Volume de l'œdème.....	p9
5)Volume du cœur de la tumeur.....	p9
6)Ratio.....	p10
Conclusion.....	p11

Introduction

Le premier semestre consacré au projet est basé sur l'intelligence artificielle notamment l'apprentissage supervisé qui consiste à étudier les différents algorithmes de classification et régression.

Concernant mon avancement, il me restait à étudier les algorithmes de régression.

Nous verrons dans une première partie, le résultat des algorithmes de régression sur notre ensemble de données réalisé précédemment. Et dans une seconde partie, l'analyse en profondeur de la tumeur avec notamment le volume de chaque partie de la tumeur ainsi que ses proportions.

I/ Analyses prédictives des algorithmes de régression

1) Dataset RFE

Construit à partir du dataset Brats, pour chaque irm des patients, j'ai extrait des informations contenues dans l'image afin d'obtenir un fichier.csv

Code permettant l'extraction des valeurs dans les pixels.

```
image1 = sitk.ReadImage('/content/drive/MyDrive/precision/Brats2020/Brats520_Training_202/Brats520_Training_202_flair.nii')
label1 = sitk.ReadImage('/content/drive/MyDrive/precision/Brats2020/Brats520_Training_202/Brats520_Training_202_seg.nii')
image = sitk.GetArrayFromImage(image1)
label = sitk.GetArrayFromImage(label1)

plt.imshow(image[100])
plt.show()
plt.imshow(label[100])
plt.show()

firstOrderFeatures = firstorder.RadiomicsFirstOrder(image1, label1)
firstOrderFeatures.enableAllFeatures()

n=0
results = firstOrderFeatures.execute()
for (key, val) in six.iteritems(results):
    print(n, key, ': ', val)
    n += 1
```



```
0 10Percentile : 282.0
1 90Percentile : 456.0
2 Energy : 2414395572.0
3 Entropy : 3.4519283240682572
4 InterquartileRange : 88.0
5 Kurtosis : 3.224283655538636
6 Maximum : 552.0
7 MeanAbsoluteDeviation : 53.53415944002464
8 Mean : 359.7836173990235
9 Median : 355.0
10 Minimum : 55.0
11 Range : 497.0
12 RobustMeanAbsoluteDeviation : 37.158288989113885
13 RootMeanSquared : 365.9979253468373
14 Skewness : -0.856712047976869934
15 TotalEnergy : 2414395572.0
16 Uniformity : 0.10842622003822293
17 Variance : 4567.788988245982
```

Ensemble de données du dataset RFE

id	10Percentile	90Percentile	Energy	Entropy	InterquartileRange	Kurtosis	Maximum
Brats17_TCIA_331	536.0	750.0	311177478.0	3.8662834684416145	112.25	4.03201803743317	953.0
Brats17_TCIA_332	330.4	508.0	1129535978.0	3.4111425657328858	71.0	3.7860128343804824	632.0
Brats17_TCIA_335	182.0	396.0	1202952208.0	3.37327072686738	164.0	1.5432032187083697	471.0
Brats17_TCIA_338	231.0	338.0	255362843.0	2.8815451477065657	55.0	3.7637667834579407	449.0
Brats17_TCIA_343	132.0	195.0	31278605.0	2.079769292090513	34.0	2.880471108685313	215.0
Brats17_TCIA_361	328.0	481.0	707740405.0	3.000898459380014	64.0	3.133773122750965	539.0

2) Analyse prédictive :

```
ridge_params = {
    "alpha": 0.9999189637151178,
    "tol": 0.8668539399622242,
    "solver": "cholesky",
}

ridge = Ridge(**ridge_params, random_state=0)

ridge_scores = ridge.fit( X_train, y_train)

y_pred1=ridge_scores.predict(X_test)
y_pred1 = np.exp(y_pred1)
ridge_result=round((RMSLE(y_test, y_pred1)*100),2)
```

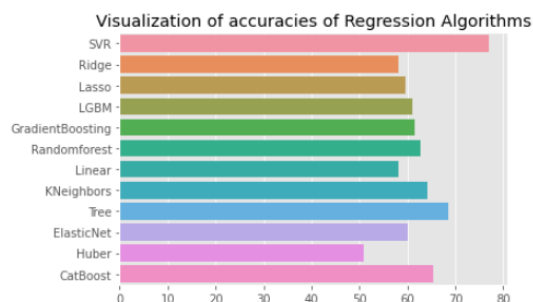
Déclaration des différents paramètres propres à chaque algorithme

Instanciation du modèle

Entraînement du modèle

-Utilisation du modèle pour prédire de nouvelles données

-Affichage du résultat :
RMSLE méthode pour l'évaluation des problèmes de régression

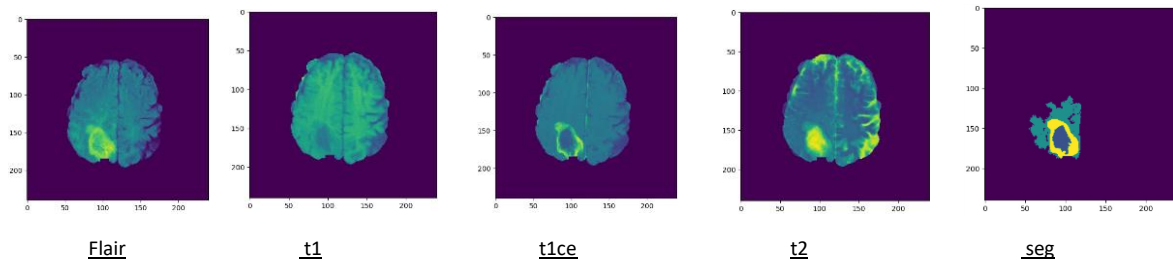


Concernant cet ensemble de données, l'algorithme obtenant la meilleure prédiction est SVR(support vector regression)avec un score de 77 %.

II/ Etude des IRM du dataset brats

1) Définition

Le dataset brats est un ensemble de données de segmentation des tumeurs cérébrales disposant pour chaque patient cinq images de l'irm du cerveau :



Explication :

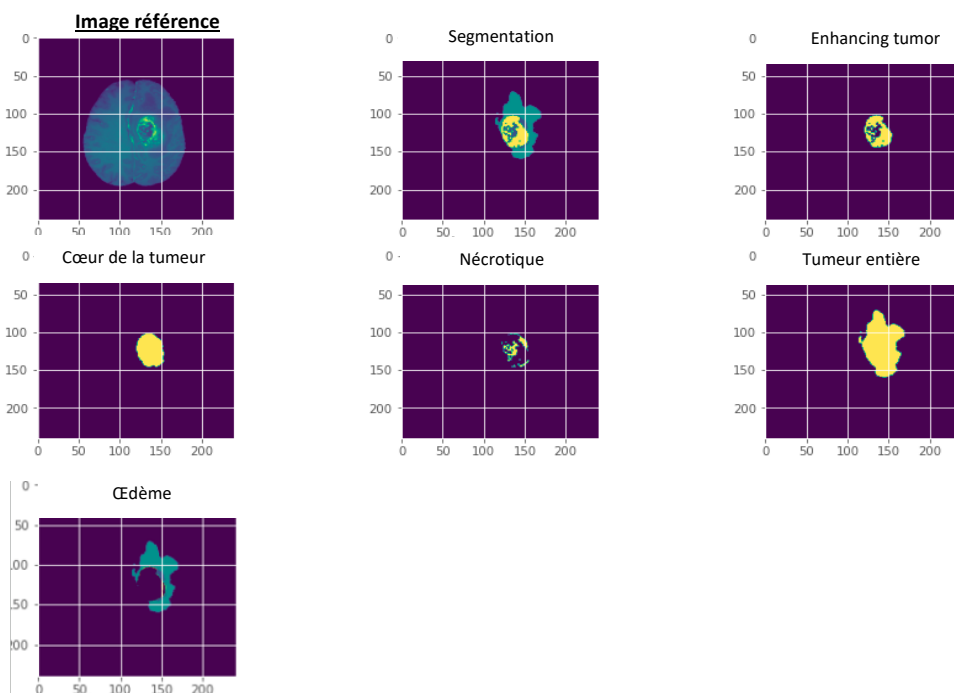
t2 : la totalité de la tumeur et son noyau sont visibles

flair : après avoir supprimé le signal de liquide céphalo-rachidien de l'image t2 cela met en évidence la région tumorale uniquement.

t1ce : l'injection de contraste permet de visualiser les parties : la partie « enhancing » et la partie nécrotique de la tumeur.

Seg : résultat obtenu : en vert la partie œdème, en jaune la partie « enhancing » et en bleu la partie nécrotique.

Le but étant de déterminer le volume des différentes parties de la tumeur affichée sur les images ci-dessous : pour cela, je vais me baser sur l'image seg.nii.



2) Programmation sur Colab

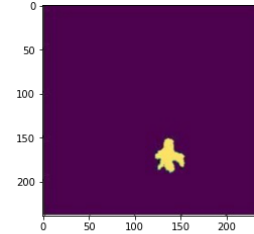
1^{ère} étape : afficher l'image au format .nii afin de l'enregistrer au format jpeg pour l'étudier.

```
img = sitk.ReadImage('/content/drive/MyDrive/pixel/Brats3/Brats17_TCIA_361_1_seg.nii.gz')
img = sitk.GetArrayFromImage(img)
data = np.array(img)
print('dimension de l image :', data.shape)
plt.imshow(img[100])
plt.show()
```

dimension de l image : (155, 240, 240)

```
image = Image.open('/content/drive/MyDrive/pixel/tumeur1.jpeg')
image=np.array(image)
print('dimension de l image :', image.shape)
plt.imshow(image)
plt.show()
```

dimension de l image : (239, 240, 3)



2^{ème} étape : Parcourir tous les pixels de l'image afin de calculer le volume de la tumeur (partie jaune).

La taille de l'image : dimension de l image : (239, 240, 3)

img.shape[0]=hauteur de l'image

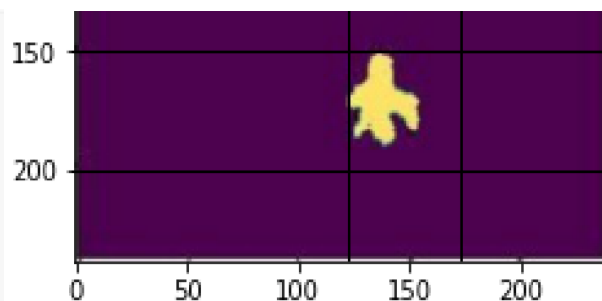
img.shape[1]=largeur de l'image

img.shape[2]=channels de l'image : représente le nombre de composants utilisés pour représenter chaque pixel.(Rouge, Vert, Bleu)

Analyse : le traitement de l'image va donc se faire pixels par pixels soit environ l'analyse de 57 360.

Cela va nous prendre un certain nombre d'heures, trop important pour notre étude, je vais donc réduire la fenêtre suivant l'emplacement de la tumeur voir ci-dessous.

```
s=0;
for i in range(145,190):
    for j in range(110,165):
        #x, y, z = image[i,j];
        #image[i, j] = (x, y, z);
        for x in range(200,256):
            for y in range(200,256):
                for z in range(40,200):
                    if np.all(image[i,j]==[x, y, z]):
                        print(image[i,j],[i,j]);
                        s=s+1;
print(s)
```



Détails du code :

Lignes 2-3 : Les boucles for i et j permettent de parcourir chaque pixel de l'image, les lignes et colonnes dans l'intervalle donné.

Lignes 6-8 : représente les pixels

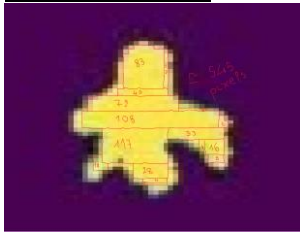
Décomposition d'un pixel : (exemple pixel bleu)

R, G, B = [11, 65, 181]

= [x, y, z] pour x,y,z je définis une plage possible de variations de pixels qui permettent d'avoir une couleur plus ou moins foncée de la couleur bleu.

1) Interprétation du résultat

Résultat attendu : estimation du volume de la tumeur



Valeur attendue = 545 mm³

Résultat obtenu :

```
[221 216 161] [188, 139]
[219 215 144] [188, 140]
[204 203 139] [188, 141]
551
```

```
print('la taille de la tumeur vaut', s)
la taille de la tumeur vaut 551 mm3
```

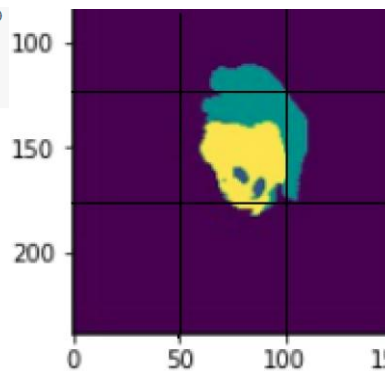
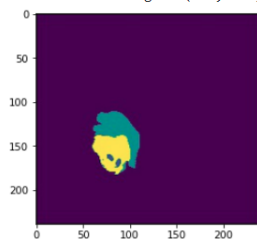
Conclusion : le résultat obtenu est proche du résultat attendu. On peut en déduire que le code a bien compté le nombre de pixels avec une meilleure précision.

III/Analyse de chaque partie de la tumeur

On choisit une irm adaptée : qui contient les trois parties distinctes de la tumeur

```
image = Image.open('/content/drive/MyDrive/pixel/tumeur2.jpg')
image=np.array(image)
print('dimension de l image :', image.shape)
plt.imshow(image)
plt.show()
```

dimension de l image : (239, 240, 3)



1) Volume de la tumeur entière

```
s=0;
for i in range(100,190):
    for j in range(50,120):
        x, y, z = image[i,j];
        image[i, j] = (x, y,z);
        if x<65 or x>75:
            if y>5 :
                if z<75 or z>86:
                    print(image[i,j],[i,j]);
                    s=s+1;
print(s)
```

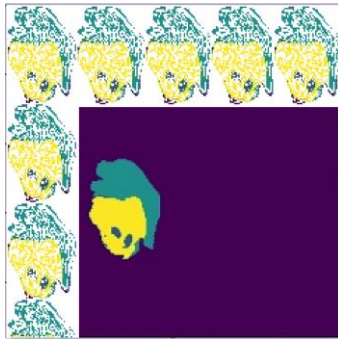
```
[164 159 127] [182, 87]
[46 10 72] [182, 89]
[61 7 65] [183, 83]
[37 7 41] [183, 84]
[13 8 30] [183, 85]
[13 9 46] [183, 86]
[62 7 72] [184, 85]
[52 9 88] [184, 86]
[56 6 95] [184, 87]
2671
```

Affiche la valeur du pixel ainsi que sa position

```
print('le volume de la tumeur entière vaut', s , 'mm^3')
le volume de la tumeur entière vaut 2671 mm^3
```

Volume de la tumeur entière = 2671 mm³

Vérification



Taille de l'image = $239 \times 240 = 57\,360$ pixels

Taille de la tumeur = 2671

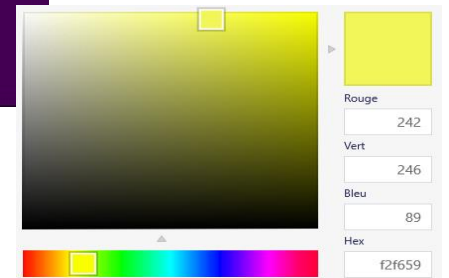
$r = 57\,360 / 2671 = 21$ tumeurs possible dans l'image

Sur l'image environ $5 \times 4 = 20$ tumeurs + les pertes

Donc, le volume de la tumeur entière semble justifié

2) Volume de la partie « enhancing tumor » (en jaune)

```
zone_couleur=0;
for i in range(135,185): #fenetre de la tumeur
    for j in range(55,110):
        #x, y, z = image[i,j];
        #image[i, j] = (x, y,z);
        for x in range(200,256):
            for y in range(200,256):
                for z in range(0,120):
                    if np.all([x, y,z]==image[i,j]):
                        print(image[i,j],[i,j]);
                        zone_couleur=zone_couleur+1;
print(zone_couleur)
[255 243 35] [180, 86]
[247 236 4] [180, 87]
[244 250 64] [180, 88]
[242 246 89] [181, 86]
[236 236 78] [181, 87]
1089
```



```
print('le volume de la partie "enhancing tumor" vaut', zone_couleur, 'mm^3')
le volume de la partie "enhancing tumor" vaut 1089 mm^3
```

Les valeurs des pixels correspondent bien à la couleur jaune de la tumeur

3) Volume de la nécrose (partie en bleu)

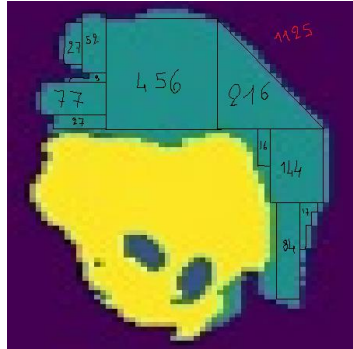
```
s=0;
for i in range(152,180):
    for j in range(65,99):
        #x, y, z = image[i,j];
        #image[i, j] = (x, y,z);
        for x in range(35,115):
            for y in range(75,130):
                for z in range(85,145):
                    if np.all(image[i,j]==[x, y,z]):
                        print(image[i,j],[i,j]);
                        s=s+1;
print(s)
```



```
[ 71 107 131] [178, 93]
[ 51 104 122] [179, 91]
[ 72 94 143] [179, 92]
40
print('le volume représentant la nécrose est de : ', s, 'mm^3')
le volume représentant la nécrose est de : 40 mm^3
```


4) Volume de l'œdème (en vert)

```
s=0;
for i in range(100,185):
    for j in range(55,115):
        #x, y, z = image[i,j];
        #image[i, j] = (x, y,z);
        for x in range(0,75):
            for y in range(100,155):
                for z in range(120,160):
                    if np.all(image[i,j]==[x, y,z]):
                        print(image[i,j],[i,j]);
                        s=s+1;
print(s)
```



```
[ 54 141 122] [177, 93]
[ 59 103 132] [177, 94]
[ 52 138 125] [178, 92]
[ 71 107 131] [178, 93]
[ 51 104 122] [179, 91]
1217
```

```
print('le volume représentant l oedeme est de : ', s , 'mm^3')

le volume représentant l oedeme est de : 1217 mm^3
```

5) Volume du cœur de la tumeur

Le cœur de la tumeur représente le volume de la nécrose et de la partie « enhancing tumor » : les résultats trouvés précédemment sont pour le volume nécrose= 40 et volume « enhancing tumor »= 1089.

Si on interprète ces résultats, nous devons obtenir un volume pour le cœur de la tumeur d'environ Vcoeur= 1130 mm³

```
zone_couleur=0;
for i in range(135,185): #fenetre de la tumeur
    for j in range(59,105):
        #x, y, z = image[i,j];
        #image[i, j] = (x, y,z);
        for x in range(200,256):
            for y in range(200,256):
                for z in range(0,120):
                    if np.all([x, y,z]==image[i,j]):
                        print(image[i,j],[i,j]);
                        zone_couleur=zone_couleur+1;
print(zone_couleur)
```



```
[243 234 43] [179, 86]
[249 233 26] [179, 87]
[248 233 52] [179, 88]
[224 230 108] [179, 89]
[251 239 67] [180, 86]
[245 229 46] [180, 87]
[253 237 100] [180, 88]
[235 230 112] [181, 87]
1183
```

```
print('la taille du coeur de la tumeur vaut', zone_couleur, 'mm3')

la taille du coeur de la tumeur vaut 1183 mm3
```

Le résultat obtenu est proche du résultat estimé

6) Ratio

$$\text{Ratio} = n/N$$

n est l'effectif du caractère dont on cherche la proportion

N est l'effectif total

$$x = \frac{\text{ratio nécrose}}{\text{ratio oedeme}} = \frac{40}{1217} = 3,2 \%$$

On peut en déduire que la partie **nécrose** représente environ 3 % de la taille de l'**œdème**.

$$x = \frac{\text{coeur de la tumeur}}{\text{oedeme}} = \frac{1183}{1217} = 97\%$$

On en conclut que le cœur **de la tumeur** représente environ 97 % de la taille de l'**œdème**. Le volume du cœur et le volume de l'œdème ont une taille quasiment identique.

$$x = \frac{\text{coeur de la tumeur}}{\text{tumeur entière}} = \frac{1183}{2671} = 44\%$$

Le **cœur** de la tumeur représente environ la moitié de la **tumeur entière**

Sachant que l'on connaît le volume de toutes les parties de la tumeur on peut effectuer le ratio suivant les parties qui nous intéressent.

Conclusion

Pour la première partie, l'analyse prédictive réalisée avec les différents algorithmes de régression sur le même ensemble de données a permis de montrer que suivant le dataset utilisé certains algorithmes sont plus efficaces que d'autres. Il est donc utile de connaître les atouts de chaque algorithme afin d'opter pour le plus adapté à notre ensemble de données.

Pour la seconde partie, grâce à la mesure de chaque volume des différentes parties de la tumeur, nous pouvons en déduire quelle partie prend le plus d'ampleur dans la tumeur.

Ce projet m'a permis d'acquérir plus de facilité dans la programmation en python et de développer une compétence : la technologie machine Learning qui pourra être un atout lors d'un futur travail.