Rendu Machine learning

Groupe:

Nael Mezdari
Amine Baha Eddine
Sayed Drissi
Hugo Sinprasith

Lien du Repo (branche “test_branch”)

Comment on a abordé le problème

Tout d’abord, nous avons fait un bilan des compétences de chacun. Ainsi, nous avons pu découvrir les affinités de chacun pour travailler efficacement.
Ensuite, nous avons analysé ensemble le jeu de données pour définir les bases et pour s’assurer qu’on parte tous sur la même longueur d’onde.
Enfin, nous avons chacun de notre côté essayé de manipuler le jeu de données avec Jupyter, Pandas, etc… car même si nous savions ce qu’on voulait réaliser, nous avions besoin de pratiquer pour au moins avoir les bases.

Exploration du dataset

Forme : 319795 lignes et 18 colonnes

```
df.shape
✓ 0.4s
(319795, 18)
```

Types des différentes colonnes (float64 ou object)
Deux types de données :

- Variables de catégories (14)

	HeartDisease	Smoking	AlcoholDrinking	Stroke	DiffWalking	Sex	AgeCategory	Race	Diabetic	PhysicalActivity	GenHealth	Asthma	KidneyDisease	SkinCancer
0	No	Yes	No	No	No	Female	55-59	White	Yes	Yes	Very good	Yes	No	Yes
1	No	No	No	Yes	No	Female	80 or older	White	No	Yes	Very good	No	No	No

- Variables discrètes (4)

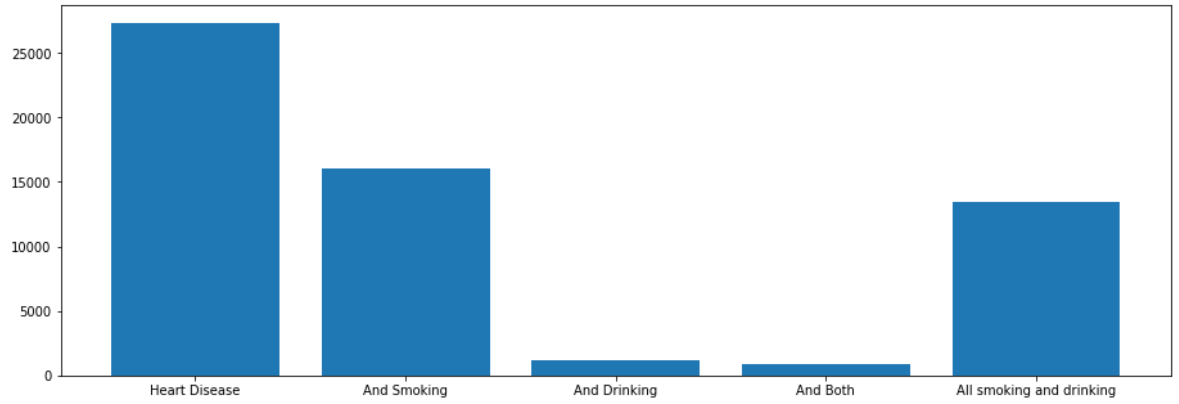
	BMI	PhysicalHealth	MentalHealth	SleepTime
0	16.60	3.0	30.0	5.0
1	20.34	0.0	0.0	7.0

Aucune valeur nulle.

Remarques supplémentaires :

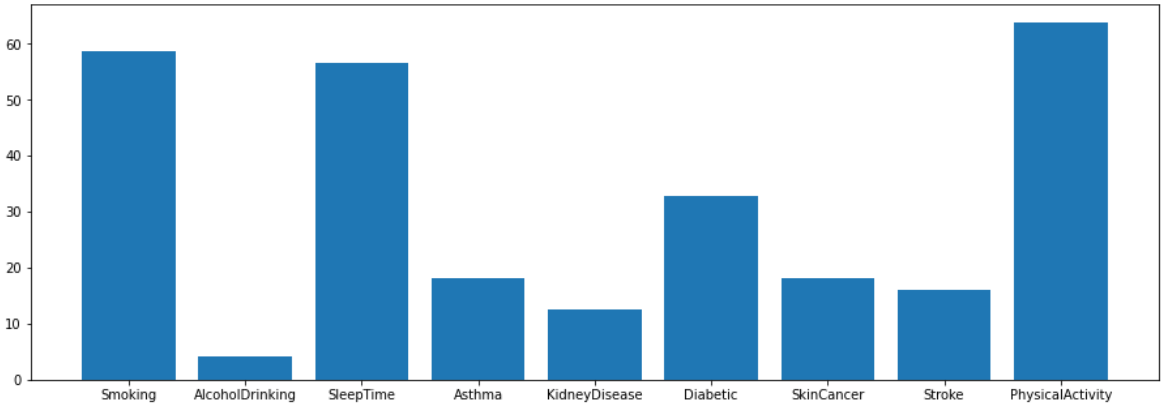
- 27373 personnes sur les 319795 qui souffrent de maladies cardiaques.
- Sur ces 27373 personnes, 16037 d'entre elles fument, 1141 boivent de l'alcool, 890 fument et boivent
- Aussi, sur les 27373 personnes, 15497 ont un temps de sommeil inférieur aux 8 heures recommandées

```
Heart Disease : 27373
And Smoking : 16037
And Drinking : 1141
And Both : 890
All Smoking and Drinking : 13415
```



Etude de chaque colonne (sur les personnes qui souffrent de maladies cardiaques) :

- la majorité des sujets sont entre 23 et 33 BMI
- 58.5 % des sujets fument, 41.5% ne fument pas
- 4.1% de sujets boivent de l'alcool
- 56.6% de sleeptime < 8 heures
- 18% qui font de l'asthme
- 12% qui ont une maladie des reins
- 32% qui ont le diabète
- 18% qui ont un cancer de la peau
- 16% qui ont déjà eu une crise cardiaque
- 63.9% font une activité physique régulière




```
Smoking : 58.586928725386336
AlcoholDrinking : 4.168341066013955
SleepTime : 56.614181858035295
Asthma : 18.021407956745698
KidneyDisease : 12.621926716107112
Diabetic : 32.722025353450476
SkinCancer : 18.19310999890403
Stroke : 16.034048149636504
PhysicalActivity : 63.89142585759691
```

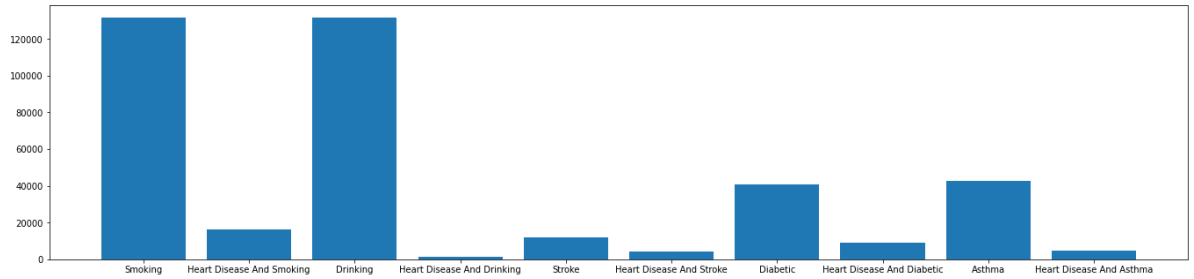
Combinaisons de données

Après l'étude initiale, on essaie de comparer les sujets qui sont malades et ceux qui ne le sont pas à travers plusieurs différentes caratéristiques.

On commence par récupérer le nombre total de personnes qui on “True” sur la caractéristique (par exemple Smoking) et on récupère le nombre de personnes qui souffrent de maladies cardiaques et de la caractéristique choisie aussi.

On compare les deux valeurs qu'on trouve :

```
Smoking : 131908
Heart Disease And Smoking : 16037, 12.157715983867543
Drinking : 131908
Heart Disease And Drinking : 1141, 0.8649968159626407
Stroke : 12069
Heart Disease And Stroke : 4389, 36.36589609743972
Diabetic : 40802
Heart Disease And Diabetic : 8957, 21.952355276702125
Asthma : 42872
Heart Disease And Asthma : 4933, 11.506344467251353
```



Théories qu’on peut émettre

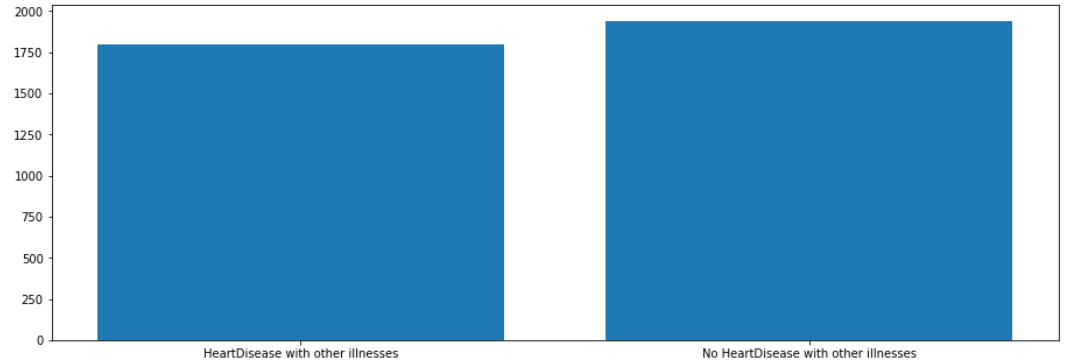
En observant les données qu’on a,

On peut voir que 36% des personnes qui souffrent de maladies cardiaques ont subi une crise cardiaque.

On peut aussi voir que 21% des personnes qui souffrent de maladies cardiaques sont diabétiques.

On peut alors combiner les deux pour voir combien de personnes qui sont diabétiques et qui ont subi une crise cardiaque souffrent d’une maladie du coeur.

```
HeartDisease with other illnesses: 1801
No HeartDisease with other illnesses: 1943
48.103632478632478% of the people that are diabetic and suffered a stroke have a heart disease.
```



Résultat: 48% sont malades d'une maladie du cœur.

Etude à travers des modèles

Split des données

Pour appliquer nos différents modèles, on doit faire en sorte qu'on ai le meme nombre de lignes et de colonnes pour X(L1) et Y(L2)

```
df.describe()
X = df.drop(columns =["HeartDisease"], axis = 1)
Y = df["HeartDisease"]
X_train, x_test, Y_train, y_test = train_test_split(X,Y,shuffle = True, test_size=

ros = RandomOverSampler(random_state =42)
X_train_resampled , y_train_resampled , = ros.fit_resample(X_train , Y_train)
X_train_test , y_train_test , = ros.fit_resample(x_test , y_test)
y_train_resampled.value_counts()
```

✓ 1.1s CVE

```
0    263310
1    263310
Name: HeartDisease, dtype: int64
```

Modèle 1: Logistic Regression

```
lr=LogisticRegression(solver='liblinear', max_iter=1000)
#lbfgs warning : Limited-memory Broyden-Fletcher-Goldfarb-Shanno Algorithm.
#That's why we recude the number of max iteration and select a specific solver.
lr.fit(X_train_resampled , y_train_resampled)
prediction = lr.predict(x_test)
print(classification_report(y_test,prediction))
#print(lr.score(X_train_resampled,y_train_resampled)) to get the accuracy alone
```

✓ 3.1s CVE

	precision	recall	f1-score	support
0	0.97	0.75	0.84	29112
1	0.23	0.78	0.36	2868
accuracy			0.75	31980
macro avg	0.60	0.76	0.60	31980
weighted avg	0.90	0.75	0.80	31980

On remarque que notre taux de précision est de 76%.

Cette valeur est due au fait que ce modèle est plus précis avec des valeurs continues, alors que nous disposons principalement de valeurs discrètes.

Nos données comme "Smoking" et "Diabetic" sont plus importantes que d'autres valeurs pour déterminer si le sujet est malade d'une maladie de cœur. Hors ce modèle accorde la même importance à chaque feature.

Ce résultat est peut-être aussi dû à un surentraînement à cause de nos valeurs répétitives d'où l'importance de préparer nos données afin de l'éviter.

Modèle 2: Random Forest

```
rf=RandomForestClassifier(n_estimators=20)
rf.fit(X_train_resampled , y_train_resampled)
prediction2 = rf.predict(x_test)
print(classification_report(y_test,prediction2))
```

✓ 15.9s CVE

	precision	recall	f1-score	support
0	0.92	0.95	0.94	29112
1	0.29	0.21	0.24	2868
accuracy			0.88	31980
macro avg	0.61	0.58	0.59	31980
weighted avg	0.87	0.88	0.87	31980

On remarque que notre taux de précision est de 58%

Le taux de précision varie suivant le paramètre "n_estimators".
Ce paramètre correspond au nombre d'arbres qui va "prendre une décision" (choix binaire).
On constate qu'avec 20 arbres, le taux de précision est assez faible.

On en déduit que les différentes données n'ont pas la même importance sur le fait qu'une personne soit malade ou non.

On suppose que la combinaison de certaines valeurs entre elles entraînera un meilleur taux de réussite.

Par exemple, une personne, dont les caractéristiques "smoking" et "Diabetic" sont plus élevées, est plus susceptible d'avoir une maladie du cœur.

```
rf=RandomForestClassifier(n_estimators=20,max_depth=10)
rf.fit(X_train_resampled , y_train_resampled)
prediction2 = rf.predict(x_test)
print(classification_report(y_test,prediction2))
```

✓ 8.9s CVE

```
              precision    recall  f1-score   support

           0       0.97      0.75      0.85     29112
           1       0.23      0.75      0.35      2868

    accuracy                           0.75     31980
   macro avg       0.60      0.75      0.60     31980
weighted avg       0.90      0.75      0.80     31980
```

Dans ce deuxième essai, on remarque que la combinaison des facteurs impacte grandement le taux de précision et la rapidité d'exécution.

Modèle 3: XGBoost

```
xgb_clf = XGBClassifier()
grid = {
    'objective':['binary:logistic'],
    'max_depth': [2,4,6],
    'alpha': [3],
    'learning_rate': [1.0],
    'n_estimators':[10]
}
model = GridSearchCV(xgb_clf, grid, scoring='accuracy', verbose=1)
model.fit(X_train_resampled,y_train_resampled)
print(model.best_score_)
print(model.best_params_)
xgb_clf.fit(X_train_resampled,y_train_resampled)
prediction3 = model.predict(x_test)
print(classification_report(y_test,prediction3))
```

✓ 43.3s CVE

```
Fitting 5 folds for each of 3 candidates, totalling 15 fits
0.7691941058068437
{'alpha': 3, 'learning_rate': 1.0, 'max_depth': 6, 'n_estimators': 10, 'objective':
'binary:logistic'}
              precision    recall  f1-score   support

           0       0.97      0.73      0.83     29112
           1       0.22      0.78      0.34      2868

    accuracy                           0.73     31980
   macro avg       0.60      0.76      0.59     31980
weighted avg       0.90      0.73      0.79     31980
```

On remarque que notre taux de précision est de 76%

Ce modèle est très similaire au modèle Random Forest, sauf qu'il ne se charge pas d'optimiser le modèle lui-même.

On utilise donc GridSearchCV qui cette charge de trouver les paramètres optimaux pour le XGBoost.

Conclusion

On conclut que les modèles basés sur des arbres de décision sont plus efficaces et évitent le surapprentissage.
Grâce à ce data set, on a réussi à trouver les facteurs essentiels de la détection d'une maladie du coeur (le diabète et le fait de fumer).