PROJET: BUSINESS INTELLIGENCE

Sujet: Prédire les jours d’hospitalisation des patients

EL FALLEH Wissam
EL GHOULI Moussa
YAAKOUBI Salma

PLAN

1. INTRODUCTION GENERALE
2. SUJET
3. PARTIE BI
4. PARTIE ML
5. CONCLUSION

Table des matières

1. INTRODUCTION GENERALE

1.1. Introduction

Le terme Business Intelligence (BI), ou informatique décisionnelle, désigne les applications, les infrastructures, les outils et les pratiques offrant l'accès à l'information, et permettant d'analyser l'information pour améliorer et optimiser les décisions et les performances d'une entreprise.

En d'autres termes, la Business Intelligence est le processus d'analyse de données dirigé par la technologie dans le but de déceler des informations utilisables pour aider les dirigeants d'entreprises et autres utilisateurs finaux à prendre des décisions plus informées.

Ainsi**,** la BI regroupe une large variété d'outils, d'applications et de méthodologies permettant de collecter des données en provenance de systèmes internes et de sources externes, de les préparer pour l'analyse, de les développer et de lancer des requêtes au sein de ces ensembles de données**.**

Ces outils permettent ensuite de créer des rapports, des tableaux de bord et des visualisations de données pour rendre les résultats des analyses disponibles pour les preneurs de décisions.

2. SUJET

2.1. Presentation

L'analyse prédictive est un outil de plus en plus important dans le domaine de la santé, car les méthodes modernes d'apprentissage automatique (ML) peuvent utiliser de grandes quantités de données disponibles pour prédire les résultats individuels des patients.
Par exemple, les prédictions ML peuvent aider les prestataires de soins de santé à déterminer les probabilités de maladie, à faciliter le diagnostic, à recommander un traitement et à prédire le bien-être futur.
Pour ce projet, j'ai choisi de me concentrer sur une métrique plus logistique des soins de santé, la durée d'hospitalisation.

3. PARTIE BI

3.1 Partie BI

Comme point de départ pour la recherche de données, mon intuition était que l'ensemble de données devrait idéalement inclure des caractéristiques telles que la catégorie du patient (par exemple, maladie cardiaque, accouchement, etc.) L'âge, le sexe.

Après avoir cherché une base de données médicale utile, j'ai fini par choisir la base de données（Prédire les jours d'hospitalisation des patients）de la part de KAGGEL.

En raison de la grande quantité d'informations qu'elle contenait.

La première etape de notre projet c'est de crée la base de donne.
En fait on a crée 2 base de donne SA-Hopital et DW-Hopital.

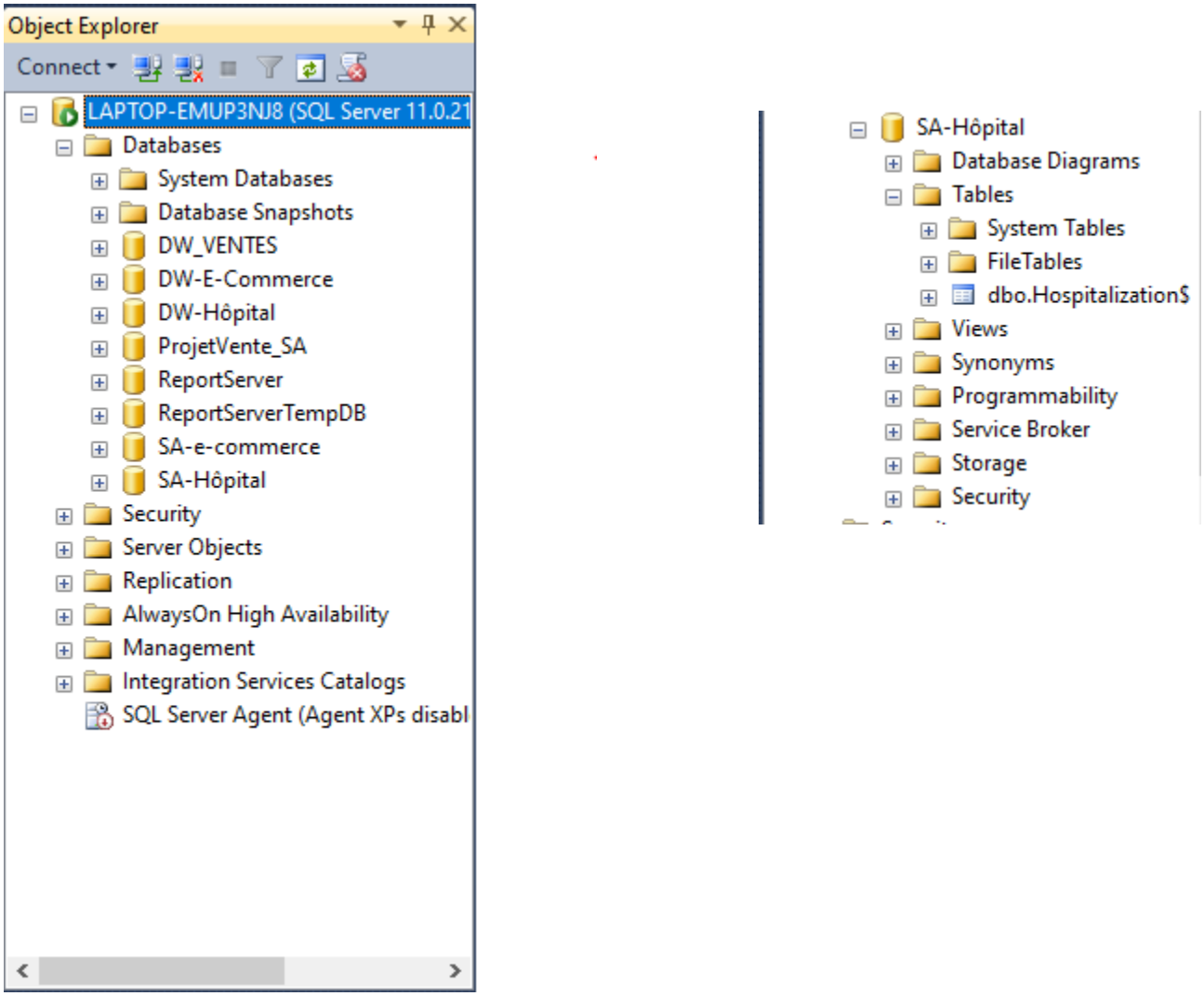

Figure 1:création d'une base sur SQL Server

En suite on a charger la base:

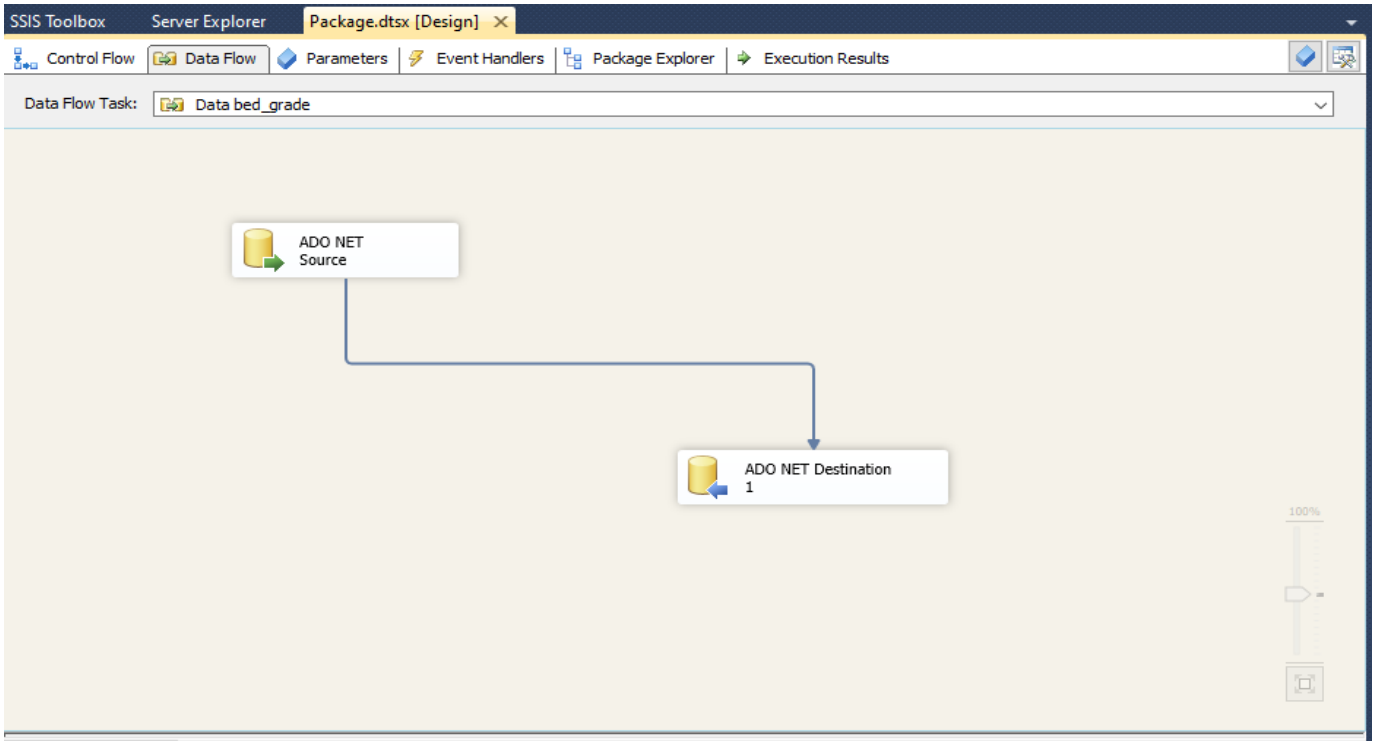

Figure 2:charger la base

Par la suite on a cree la table de faites:

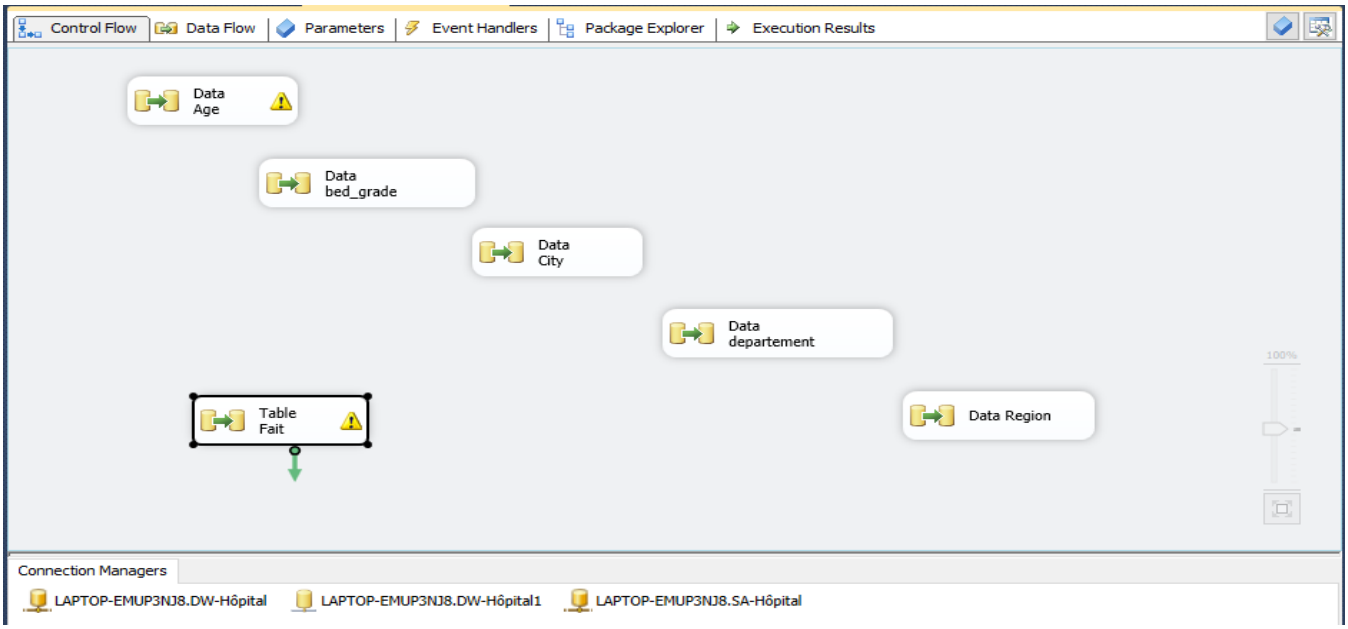

Figure 3:Création du table de Faites

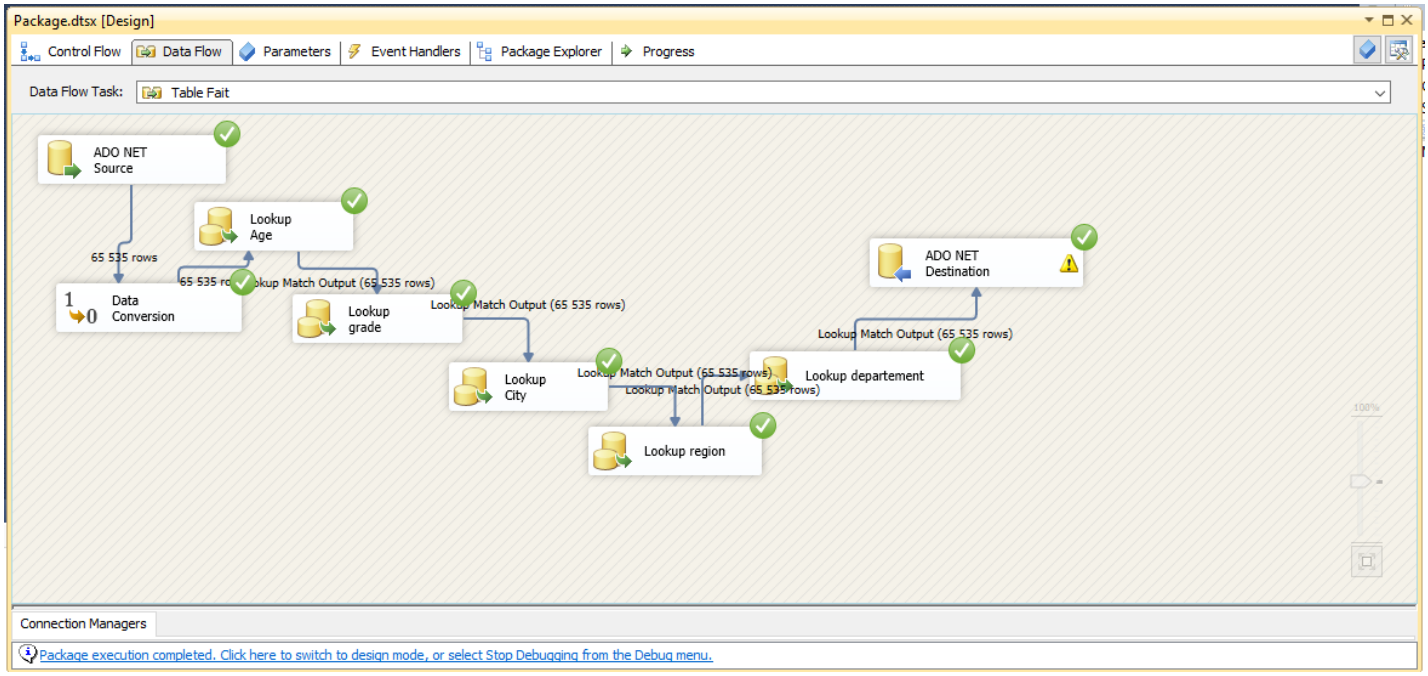

Figure 4:Remplier table de Faites

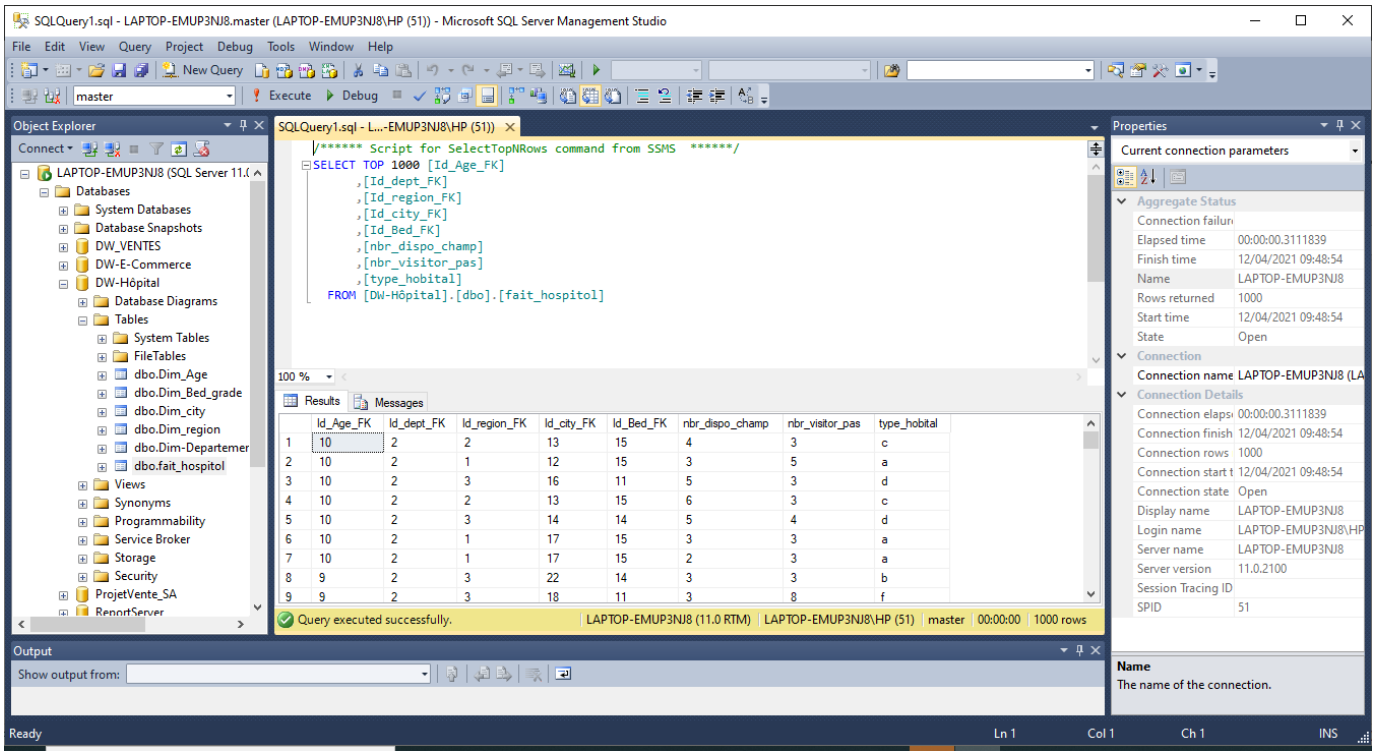

Figure 5:table de Faites charge

Après la creation des tables et les dimension et la charge de la base on a fait le cube

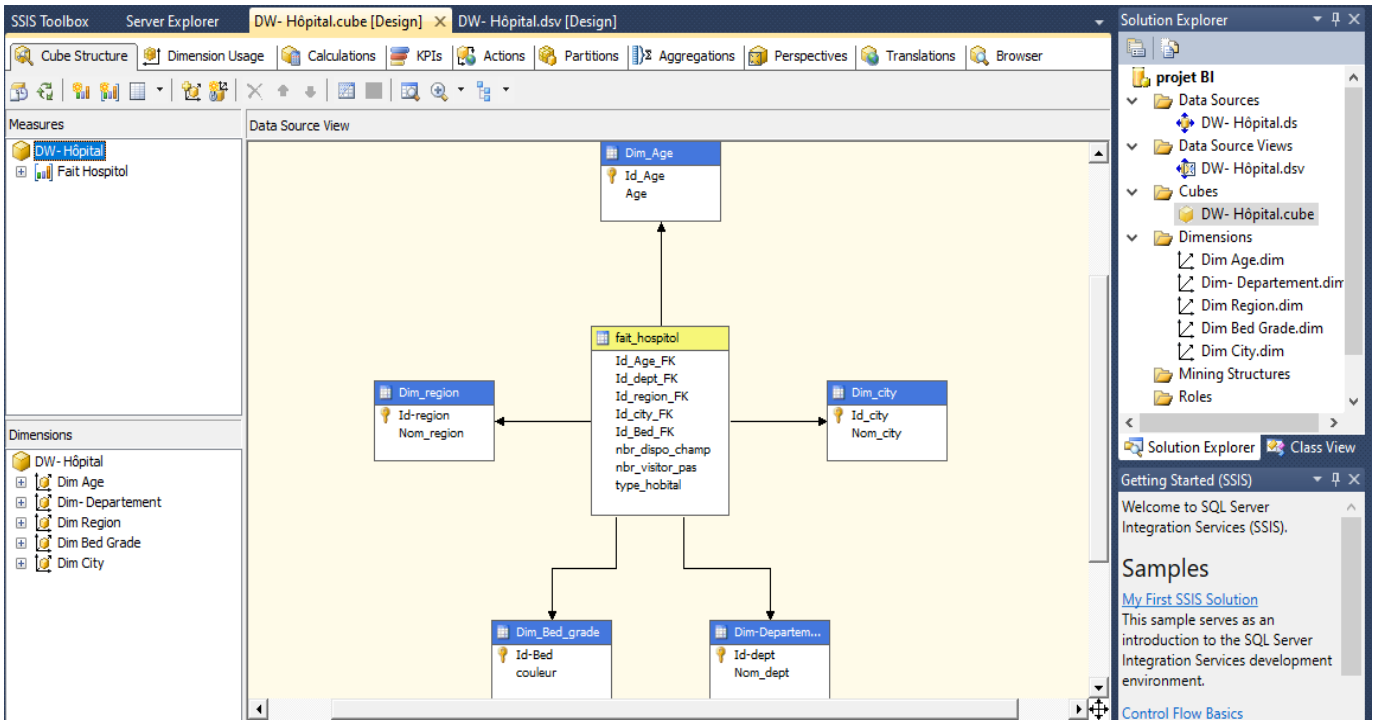

Figure 6:Cube

Et finalement on a construit notre power BI

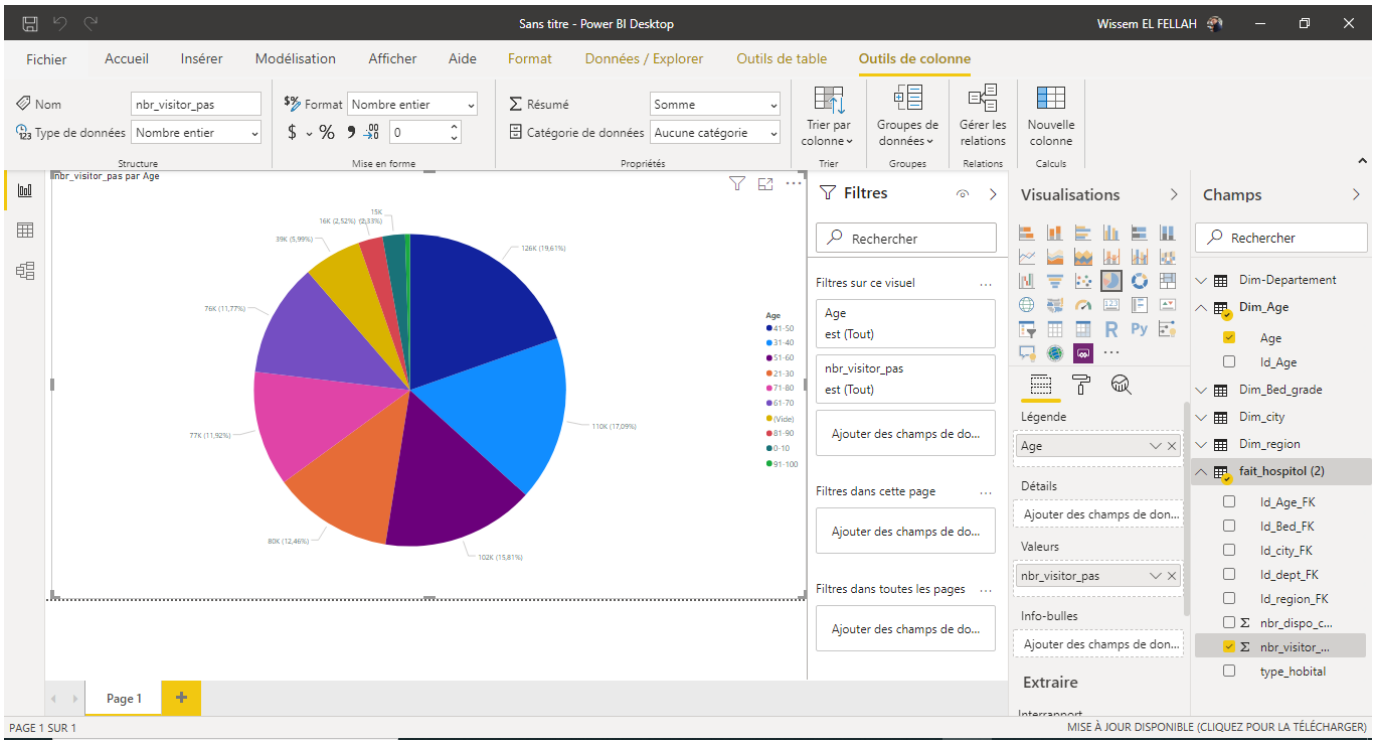

Figure 7:nombre de visiteur par Age

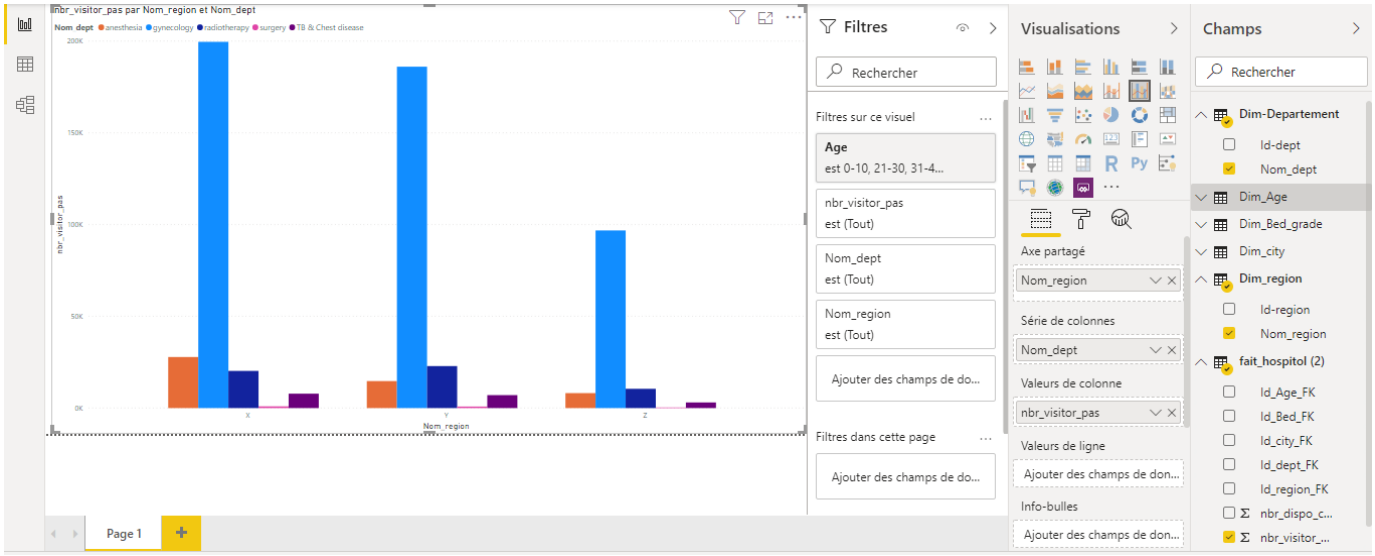

Figure 8:nombre de visiteur par département & région

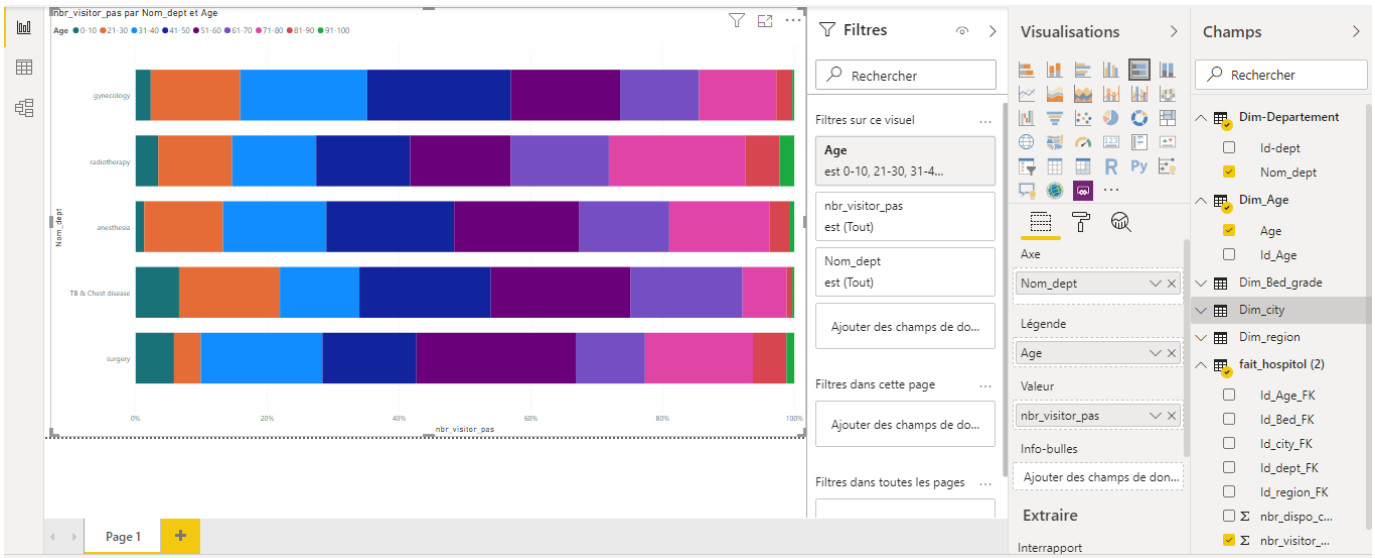

Figure 9:nombre de visiteur par Age & département

4. PARTIE ML

4. Partie ML

4.1. Architecture de la solution:

Notre système est composé de deux phases principales:
La première étape consterne la partie BI, elle est composée de trois sous-étapes qui sont:
1) la collecte de donnée,
2) la transformation des données,
3) le stockage de données.
La deuxième Cette étape est composée de deux sous étapes, à savoir le prétraitement et le nettoyage de données, et la classification qui est basée sur les algorithmes de l'apprentissage automatique (Machine Learning: ML).

4.2 Dataset used:

Un hôpital dispose de plusieurs types de données. Ces données concernent la ville, le patient, le département, etc. – il y a tellement de chiffres et de statistiques que nous pourrions collecter, c'est époustouflant! Nous réduirons notre champ d'application à certains domaines spécifiques pour ce projet:
Case_id: Identifiant du cas (ce n'est pas crucial pour l'analyse, mais au niveau de la base de données, il sera utile d'avoir cette information)
Hospital_code: indique Code de l'hôpital
City_Code_Hospital: indique le code d'hôpital par ville
Hospital_region_code : indique le code d'hôpital par région
Available Extra Rooms in Hospital: indique la disponibilité de chambre vide dans l'hôpital
Department: c'est le non de département dans l'hôpital
Ward_Type: indique Types de services hospitaliers
Ward_Facility_Code: Code de installation
Bed Grade: Catégorie de lit
Patient id: Identifiant du patient

City_Code_Patient: Code de ville de Patient
Type of Admission: Type d'admission (Lors d'une hospitalisation, le régime obligatoire de la Sécurité sociale laisse à la charge du patient des frais pouvant évoluer selon le type d'admission).
Severity of Illness: Gravité de la maladie
Visitors with Patient: Visiteurs avec patient
Age: indique l'âge de patient
Admission_Deposit: Caution d'admission (Pendant l'admission et tout au long de votre séjour
Stay: temps de séjour de patient dans l'hôpital.

4.3. Les algorithms utilisés:

Dans cette étape, nous étudions les algorithmes d'apprentissage automatique qu'on a utilisé.
On a commencé par l'algorithme Random Forest :

```
"4)Random forest

model=RandomForestRegressor(random_state=0, n_estimators=100)
model=LinearRegression()

import time
debut=time.time()
model.fit(Data, y)
fin=time.time()-debut
ypred=model.predict(Data)

from sklearn.metrics import mean_squared_error
mse=mean_squared_error(y,ypred)
import math
print('RMSE', math.sqrt(mse))

from sklearn.metrics import explained_variance_score
EV=explained_variance_score(y,ypred)
print("Explained variance : %f" % (EV))
```

Figure 10:Algorithmes Random Forest

En suit on a utilise XGBOOST:

```
"6)XGBoost
import xgboost as xgb
model=xgb.XGBRegressor(objective='reg:linear', linearing_rate=0.3, n_estimators=

import time
debut=time.time()
model.fit(Data, y)
fin=time.time()-debut
pred=model.predict(Data)

from sklearn.metrics import mean_squared_error
mse=mean_squared_error(y,pred)
import math
print('RMSE', math.sqrt(mse))

from sklearn.metrics import explained_variance_score
EV=explained_variance_score(y,pred)
print("Explained variance : %f" % (EV))
```

Figure 11:Algorithmes XGBOOST

Et on a terminer avec la Regression linear multiple:

```
"7)Regression linear multiple
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(Data,y)

a=model.coef_
print(a)

b=model.intercept_
print(b)
X=np.array(Data)

pred=model.predict(Data)

from sklearn.metrics import mean_squared_error
mse=mean_squared_error(y,pred)
import math
print('RMSE', math.sqrt(mse))

import pandas as pd
Y1=pd.DataFrame(y)
Y1.describe()

from sklearn.metrics import explained_variance_score
EV=explained_variance_score(y,pred)
print("Explained variance : %f" % (EV))

from sklearn.metrics import r2_score
r=r2_score(y,pred)
print(r)
```

Figure 12:Algorithmes Régression linear multiple

4.4. Les protocols expérimentaux:

Pour notre premiere algorithmes on a utilise RMSE et EV comme des protocoles experimentaux car avec l'algorithme de Random Forest il est primordial d'utilser RMSE et surtout EV

Et pour notre deuxieme algorithme XGBOOST, on a utilizer aussi RMSE et EV

Et finalement Regression linear multiple aussi on a utilizer RMSE, et EV

4.5. Résultats et discussion:

	Reg-Mult	Random-F	XGBOOST
RMSE	1.66932	1.45193	1.39624
EV	0.23545	0.38833	0.43435
CC	0.0984	0.1036	0.2364

En terme de RMSE et EV, nous clairement remarque que XGBOOST donne meilleur resultat que Reg-Mult et Random-Forest.

➔XGBOOST donne une Meilleur resultat que Random-F car XGBOOST est un algo sequential (boosting), et Random-F est un algo independent (bagging)

➔Random-F done une Meilleur resultat que Reg-Multi car R-F est ensemble des predicteur alors que Reg-Multi est un seul predictaur

En terme de complexité de calcule XGBOOST a une grande complexité de calcule car il est sequential.

En suite Random-F un peu faible car plusieur predicteur, et finallement Reg-Multi donne Meilleur CC.

5. CONCLUSION

Ce rapport est destiné à vous inspirer sur la façon d'utiliser les données.
Il existe de plus en plus d'outils disponibles que vous pouvez utiliser pour tirer des enseignements des données publiques. J'espère que cette procédure pas à pas vous donnera quelques idées sur la façon de faire fonctionner les données pour vous.
Vous pouvez également utiliser cette analyse pour créer des modèles d'apprentissage automatique. Nous avons effectué le nettoyage et l'exploration des données - faites-le avancer et utilisez vos algorithmes préférés pour prédire les jours d'hospitalisation des patients. Les possibilités sont infinies.