healthcare_dataset

presentation projet integree

validation finale

plan

Etape 1 : describtion du projet

Etape 2 : machine learning

Etape 3 : Conception du data warehouse et Alimentation du data warehouse avec Talend

Etape 4 :Visualisation des données avec Power BI

Etape 1 :describtion du projet

1. Introduction
2. Objectifs
3. Problématique
4. Étude de l'existant
5. Description du jeu de données

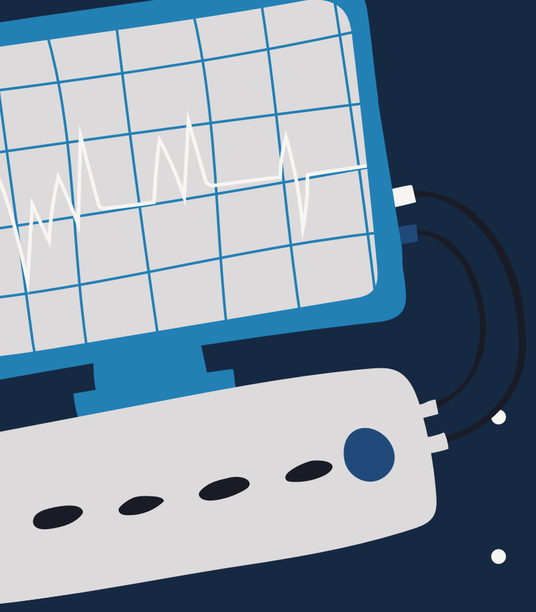
Etape 1 : describtion du projet

Introduction

Avec la croissance exponentielle des données médicales dans les établissements de santé, il devient essentiel d'exploiter ces données de manière intelligente pour améliorer les processus décisionnels. L’analyse de données hospitalières permet de mieux comprendre les coûts, les flux d’admission, les traitements, et d’optimiser la gestion des ressources.

Objectifs

- Mettre en place un entrepôt de données structuré pour centraliser les données hospitalières.
- Construire un tableau de bord décisionnel permettant une visualisation dynamique des indicateurs de performance.
- Réaliser une analyse descriptive et prédictive pour identifier des tendances clés.



Problématique

Comment exploiter efficacement les données hospitalières pour améliorer la qualité des soins, la gestion administrative et la maîtrise des coûts médicaux ?

HEALTHCARE_DATASET

Étude de l'existant

Les établissements de santé utilisent de plus en plus des solutions BI pour :

- Améliorer la gestion des patients et du personnel.
- Réduire les coûts d’exploitation.
- Optimiser la planification des ressources hospitalières.

Description du jeu de données

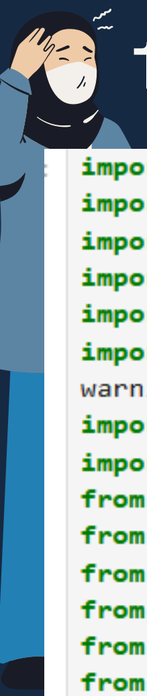
Titre : Healthcare Dataset
Lien : https://www.kaggle.com/datasets/prasad22/healthcare-dataset
Format : CSV
Taille : 55 500 lignes
Nature : Données hospitalières simulées

Structure du fichier healthcare_dataset.csv :

- **Name** : Nom du patient (variable qualitative)
- **Age :** Âge du patient (variable quantitative)
- **Gender :** Sexe (Male / Female) (variable qualitative)
- **Blood Type :** Groupe sanguin (A, B, AB, O) (variable qualitative)
- **Medical Condition :** Diagnostic ou pathologie (variable qualitative)
- **Date of Admission :** Date d'entrée à l'hôpital (variable qualitative)
- **Doctor :** Médecin en charge (variable qualitative)
- **Hospital :** Établissement hospitalier (variable qualitative)
- **Insurance Provider :** Compagnie d'assurance (variable qualitative)
- **Billing Amount :** Montant facturé au patient (variable quantitative)
- **Room Number :** Numéro de chambre attribuée (variable quantitative)
- **Admission Type :** Type d'admission (urgente, planifiée…) (variable qualitative)
- **Discharge Date :** Date de sortie de l'hôpital (variable qualitative)
- **Medication :** Médicaments prescrits (variable qualitative)
- **Test Results :** Résultats des tests médicaux (variable qualitative)

Etape 2 : Machine learning

1. Import Libraries and Dataset:
2. Exploration des Données (EDA)Problématique
3. analyse des donnees
4. modele

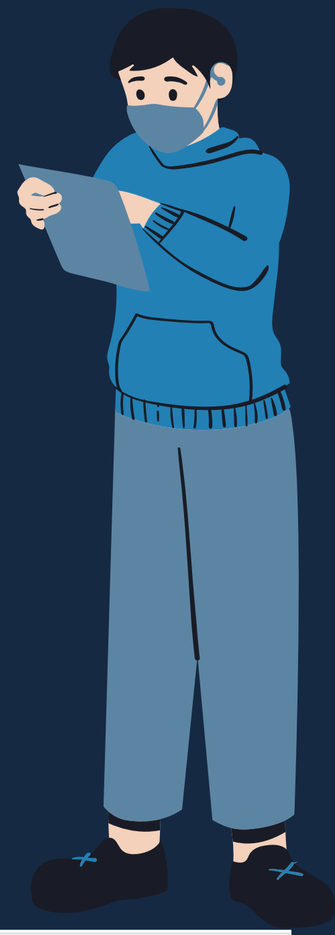


1) Import Libraries and Dataset:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
import warnings
warnings.filterwarnings('ignore')
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, r2_score
```

```
df= pd.read_csv("C:/Users/souha/OneDrive/Desktop/pi sem2/healthcare_dataset.csv")
```

Etape 2 : Machine learning



```
df.head()
```

	Name	Age	Gender	Blood Type	Medical Condition	Date of Admission	Doctor	Hospital	Insurance Provider	Billing Amount	Room Number	Admission Type	Discharge Date	Medication	Test Results
0	Bobby JacksOn	30	Male	B-	Cancer	2024-01-31	Matthew Smith	Sons and Miller	Blue Cross	18856.281306	328	Urgent	2024-02-02	Paracetamol	Normal
1	LesLie TErRy	62	Male	A+	Obesity	2019-08-20	Samantha Davies	Kim Inc	Medicare	33643.327287	265	Emergency	2019-08-26	Ibuprofen	Inconclusive
2	DaNnY sMitH	76	Female	A-	Obesity	2022-09-22	Tiffany Mitchell	Cook PLC	Aetna	27955.096079	205	Emergency	2022-10-07	Aspirin	Normal
3	andrEw waTtS	28	Female	O+	Diabetes	2020-11-18	Kevin Wells	Hernandez Rogers and Vang,	Medicare	37909.782410	450	Elective	2020-12-18	Ibuprofen	Abnormal
4	adrIENNE bEll	43	Female	AB+	Cancer	2022-09-19	Kathleen Hanna	White-White	Aetna	14238.317814	458	Urgent	2022-10-09	Penicillin	Abnormal

2)Exploration des Données (EDA)

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 55500 entries, 0 to 55499
Data columns (total 15 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Name                55500 non-null  object 
 1   Age                 55500 non-null  int64  
 2   Gender              55500 non-null  object 
 3   Blood Type          55500 non-null  object 
 4   Medical Condition   55500 non-null  object 
 5   Date of Admission   55500 non-null  object 
 6   Doctor              55500 non-null  object 
 7   Hospital            55500 non-null  object 
 8   Insurance Provider  55500 non-null  object 
 9   Billing Amount      55500 non-null  float64
 10  Room Number         55500 non-null  int64  
 11  Admission Type      55500 non-null  object 
 12  Discharge Date      55500 non-null  object 
 13  Medication          55500 non-null  object 
 14  Test Results        55500 non-null  object 
dtypes: float64(1), int64(2), object(12)
memory usage: 6.4+ MB
```

```
df.describe()
```

	Age	Billing Amount	Room Number
count	55500.000000	55500.000000	55500.000000
mean	51.539459	25539.316097	301.134829
std	19.602454	14211.454431	115.243069
min	13.000000	-2008.492140	101.000000
25%	35.000000	13241.224652	202.000000
50%	52.000000	25538.069376	302.000000
75%	68.000000	37820.508436	401.000000
max	89.000000	52764.276736	500.000000

Statistiques descriptives :

Âge moyen : 51.5 ans (écart : 13–89 ans).

Conditions médicales fréquentes : Arthrite (9 308 cas), Diabète (9 304 cas).

Types de sang les plus courants : A- (6 969 patients), A+ (6 956).

2)Exploration des Données (EDA)

```
df['Medical Condition'].value_counts()
```

```
Medical Condition
Arthritis          9308
Diabetes           9304
Hypertension       9245
Obesity            9231
Cancer             9227
Asthma             9185
Name: count, dtype: int64
```

```
df['Blood Type'].value_counts()
```

```
Blood Type
A-      6969
A+      6956
AB+     6947
AB-     6945
B+      6945
B-      6944
O+      6917
O-      6877
Name: count, dtype: int64
```

```
df['Insurance Provider'].value_counts()
```

```
Insurance Provider
Cigna                11249
Medicare             11154
UnitedHealthcare     11125
Blue Cross           11059
Aetna                10913
Name: count, dtype: int64
```

```
print(f"Age Range: {df['Age'].min()} to {df['Age'].max()}")
```

```
Age Range: 13 to 89
```

3) Preprocessing:

Nettoyage du texte: **Conversion des noms en minuscules.**

```python
# Convert the 'Name' column to lowercase
df['Name'] = df['Name'].str.lower()

# Display the updated DataFrame
df.head()
```

	Name
0	Bobby JacksOn
1	LesLie TErRy
2	DaNnY sMitH
3	andrEw waTtS
4	adrIENNE bEll

	Name	Age	Gender	Blood Type	Medical Condition	Date of Admission	Doctor	Hospital
0	bobby jackson	30	Male	B-	Cancer	2024-01-31	Matthew Smith	Sons and Miller
1	leslie terry	62	Male	A+	Obesity	2019-08-20	Samantha Davies	Kim Inc
2	danny smith	76	Female	A-	Obesity	2022-09-22	Tiffany Mitchell	Cook PLC
3	andrew watts	28	Female	O+	Diabetes	2020-11-18	Kevin Wells	Hernandez Rogers and Vang,
4	adrienne bell	43	Female	AB+	Cancer	2022-09-19	Kathleen Hanna	White-White

4) Exploratory Data Analysis (EDA)

comprendre la Distribution des colonnes suivantes :

- **conditions médicales: Arthritis" est la plus fréquente (9308 cas)**
- **Groupes sanguins: Répartition assez uniforme, avec "A-" légèrement plus fréquent.**
- **Fournisseurs d'assurance: "Cigna" est le plus courant**
- **age : min 13 max 89**

```
df['Medical Condition'].value_counts()
```

```
Medical Condition
Arthritis       9308
Diabetes        9304
Hypertension    9245
Obesity         9231
Cancer          9227
Asthma          9185
Name: count, dtype: int64
```

```
df['Blood Type'].value_counts()
```

```
Blood Type
A-      6969
A+      6956
AB+     6947
AB-     6945
B+      6945
B-      6944
O+      6917
O-      6877
Name: count, dtype: int64
```

```
df['Insurance Provider'].value_counts()
```

```
Insurance Provider
Cigna                11249
Medicare             11154
UnitedHealthcare     11125
Blue Cross           11059
Aetna                10913
Name: count, dtype: int64
```

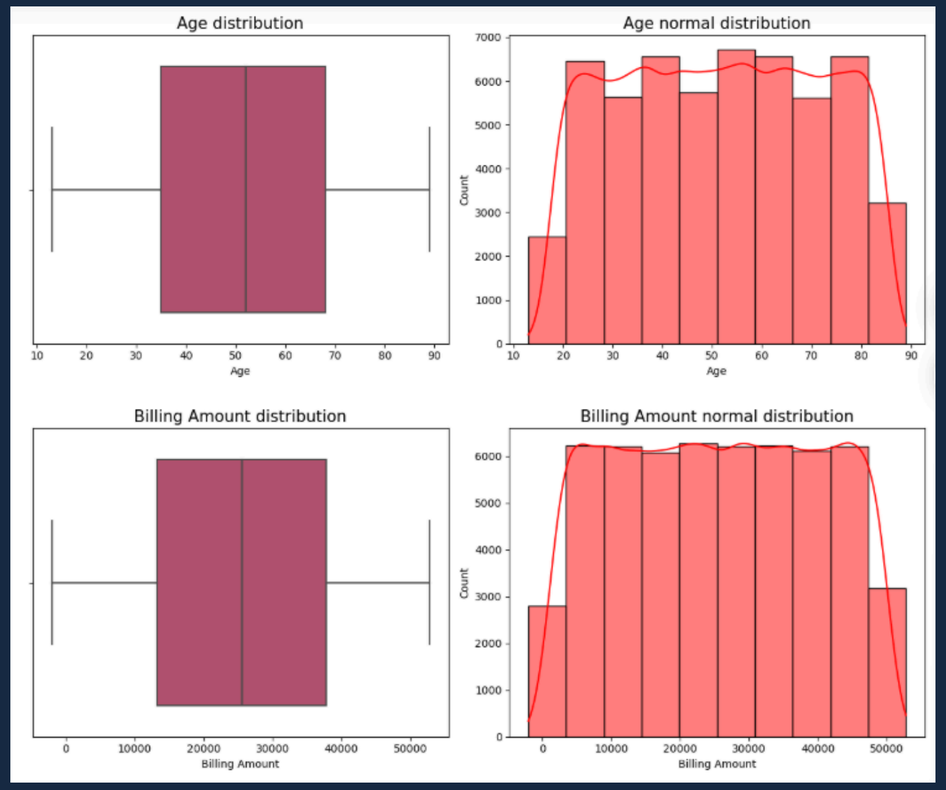
```
print(f"Age Range: {df['Age'].min()} to {df['Age'].max()}")
```

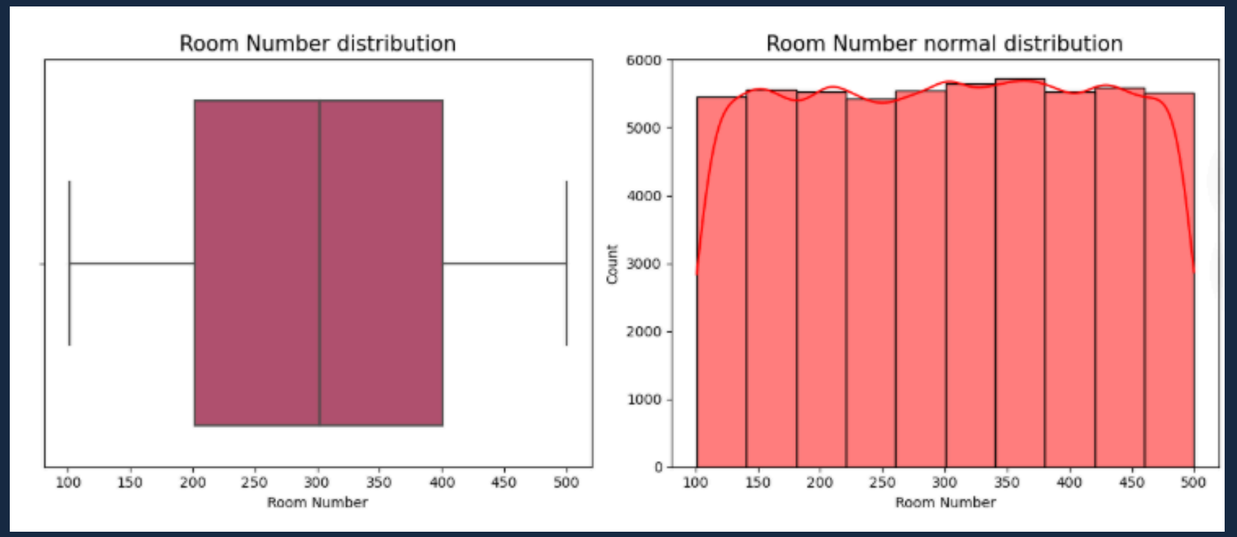
```
Age Range: 13 to 89
```

Visualisation de la distribution des colonnes numériques :

```
numerical_cols = ['Age', 'Billing Amount', 'Room Number']
for i in numerical_cols:
    fig, ax = plt.subplots(1,2,figsize=(12,5))
    sns.boxplot(data=df, x=i, ax=ax[0], palette = 'flare')
    ax[0].set_title(f'{i} distribution', fontsize=15)
    sns.histplot(data=df, x=i, kde=True, bins = 10, ax=ax[1], color="red")
    ax[1].set_title(f'{i} normal distribution', fontsize=15)
    plt.tight_layout()
    plt.show()
```

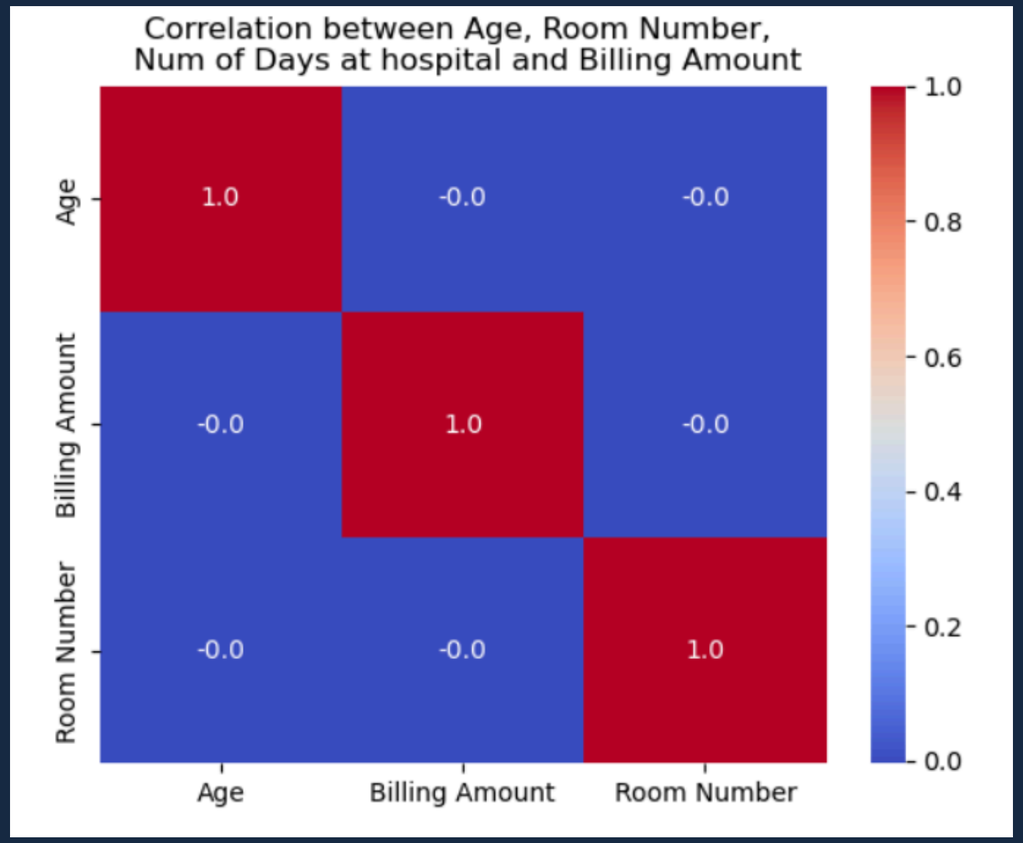
- Age
- Billing Amount
- Room Number.





Analyse de la Corrélation entre l'Âge, le Montant de Facturation et le Numéro de Chambre

```
: fig = plt.figsize=(15,5)
  sns.heatmap(data=df[numerical_cols].corr(), annot=True, cmap='coolwarm', fmt=".1f" )
  plt.title('Correlation between Age, Room Number, \n Num of Days at hospital and Billing Amount')
  plt.show()
```



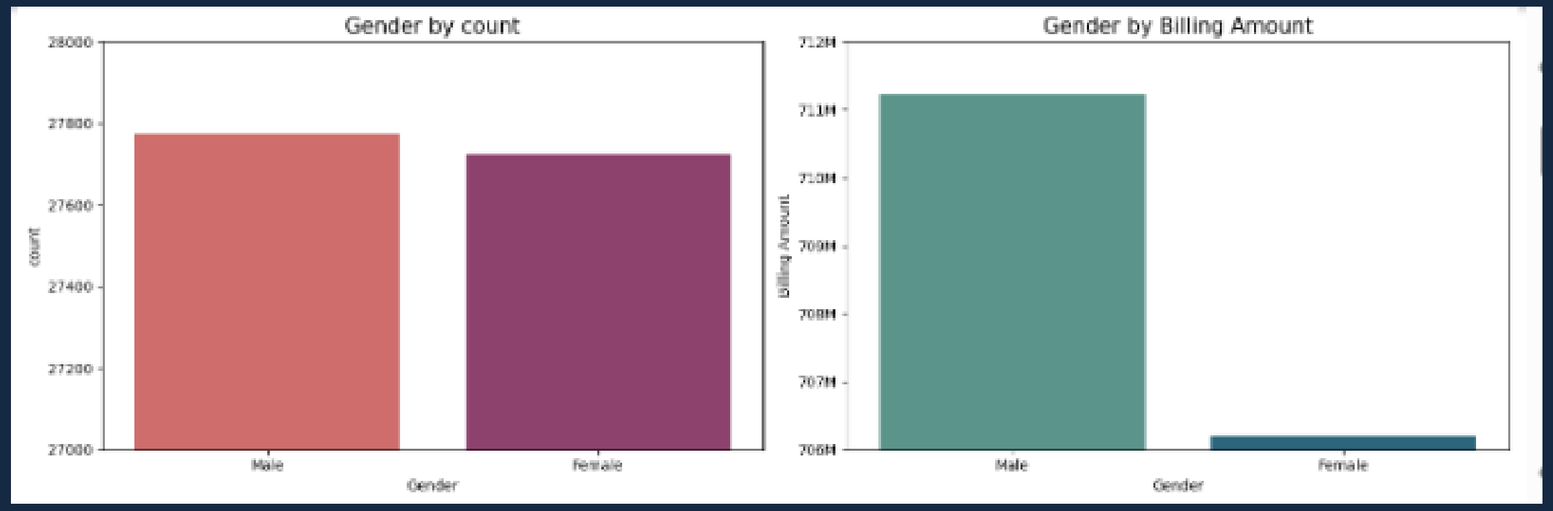
Corrélations Observées :

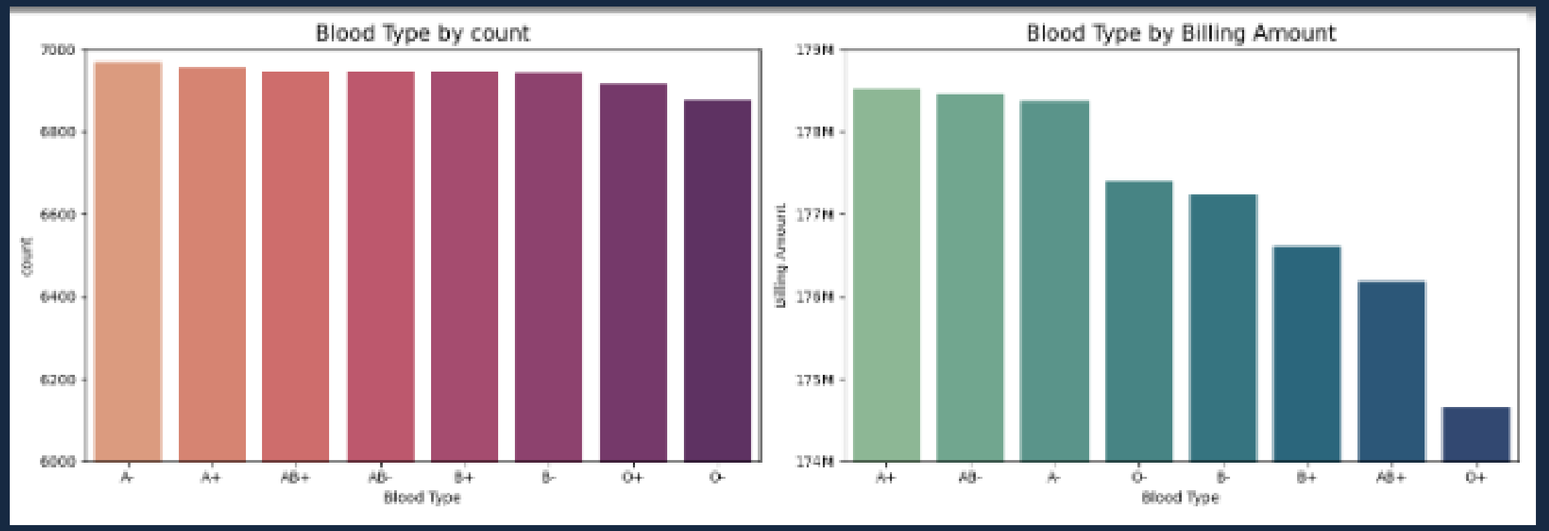
- **Âge et billing amount** : Corrélation de 1.0, indiquant une très forte corrélation positive.
- **Âge et Numéro de Chambre** : Corrélation de 0.0, indiquant aucune relation.
- **billing amountet Numéro de Chambre** : Corrélation de 0.0, indiquant également aucune relation.

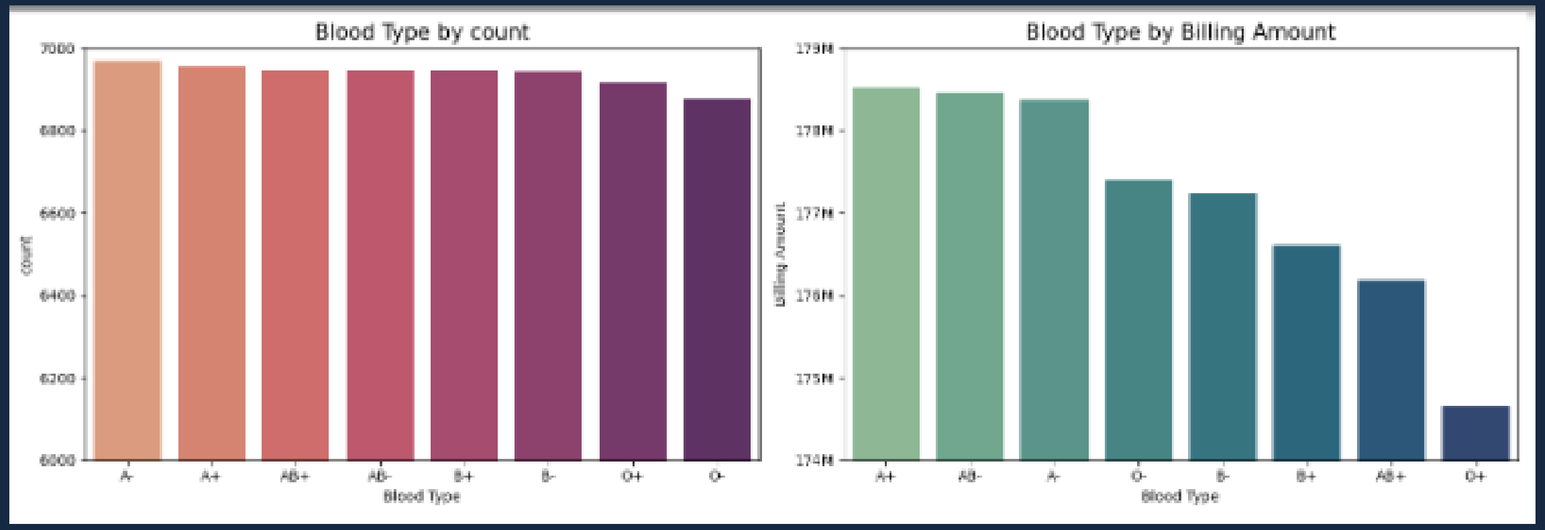
visualisation des variables categorielle

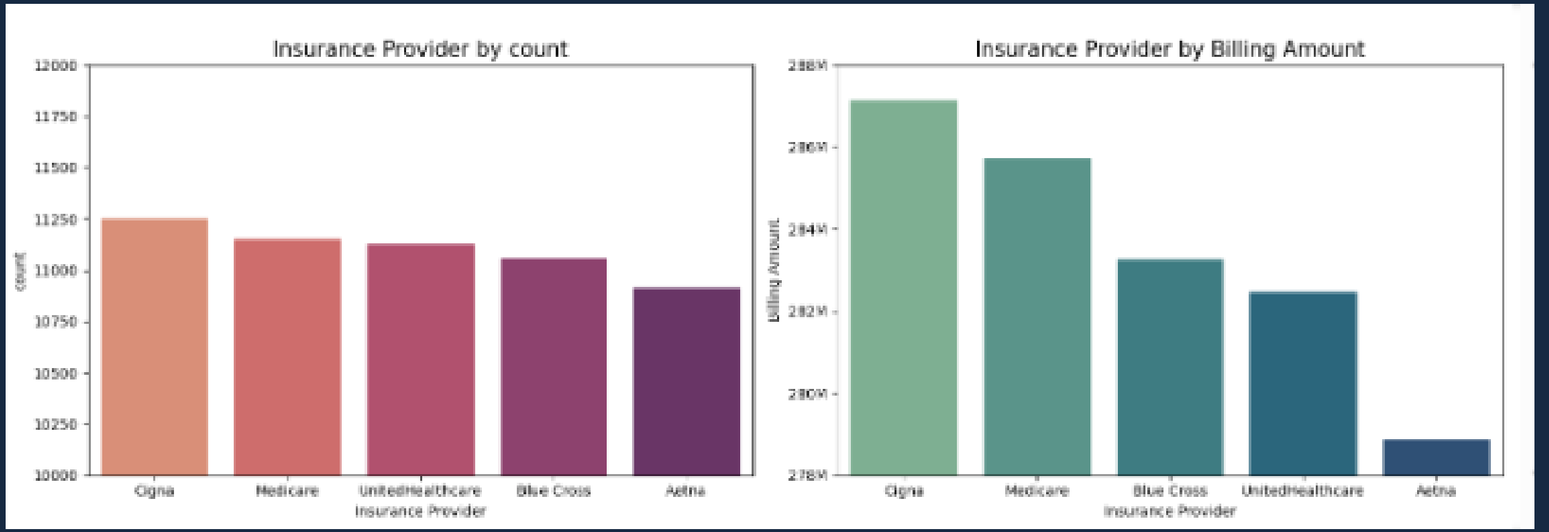
```
for i in categorical_cols:
    fig, ax = plt.subplots(1,2, figsize=(15,5))
    value_counts_df = df[i].value_counts(sort='ascending').reset_index()
    value_counts_df.columns = [i, 'count']
    y_min = value_counts_df['count'].min()
    y_max = value_counts_df['count'].max()
    y_min = max(0, (y_min // 1_000) * 1_000)
    y_max = ((y_max // 1_000) + 1) * 1_000
    sns.barplot(data=value_counts_df, x=i, y='count', palette = 'flare', ax=ax[0])
    ax[0].set_ylim(y_min, y_max)
    ax[0].set_title(f'{i} by count', fontsize = 15)
    group = df.groupby([i], sort=False)['Billing Amount'].sum().reset_index()
    group = group.sort_values('Billing Amount', ascending=False)
    billing_order = group[i]
    sns.barplot(data=group , x=i, y='Billing Amount', ax=ax[1], palette = 'crest', order=billing_order)
    y_min = group['Billing Amount'].min()
    y_max = group['Billing Amount'].max()
    y_min_rounded = max(0, (y_min // 1_000_000) * 1_000_000)
    y_max_rounded = ((y_max // 1_000_000) + 1) * 1_000_000
    ax[1].set_ylim(y_min_rounded, y_max_rounded)
    ax[1].set_title(f'{i} by Billing Amount', fontsize =15)
    ax[1].yaxis.set_major_formatter(ticker.FuncFormatter(lambda x, _: f'{x*1e-6:.0f}M'))
    plt.tight_layout()
    plt.show()
```

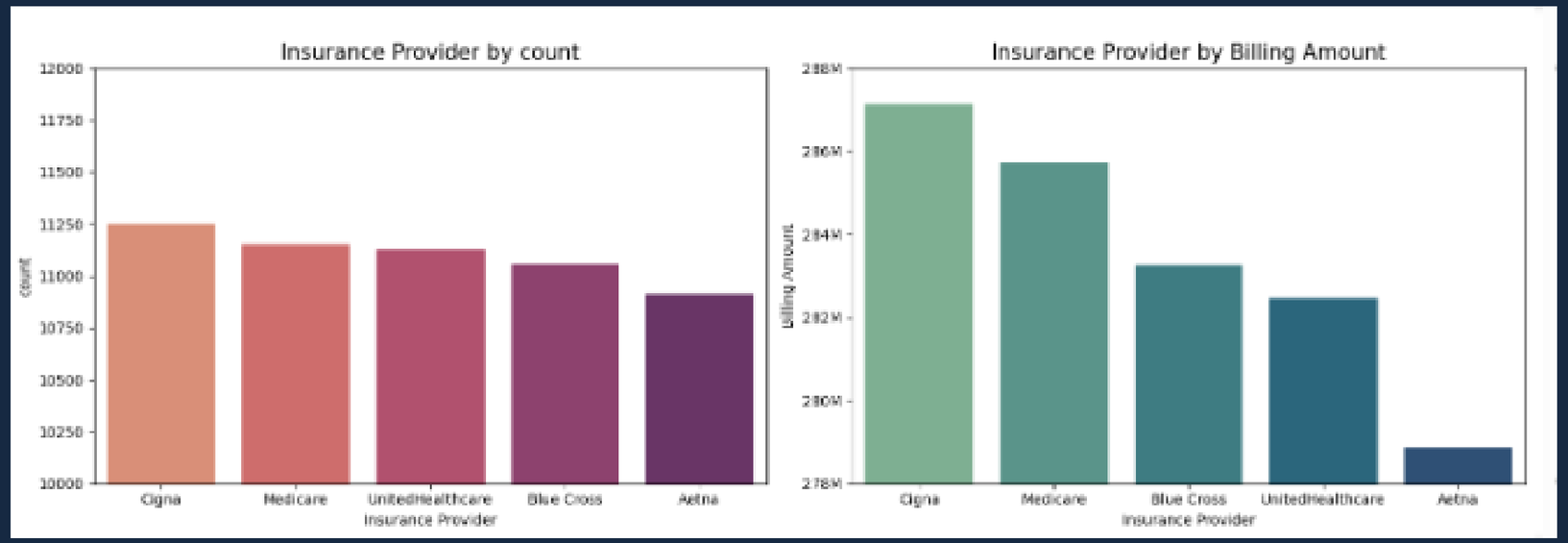
output

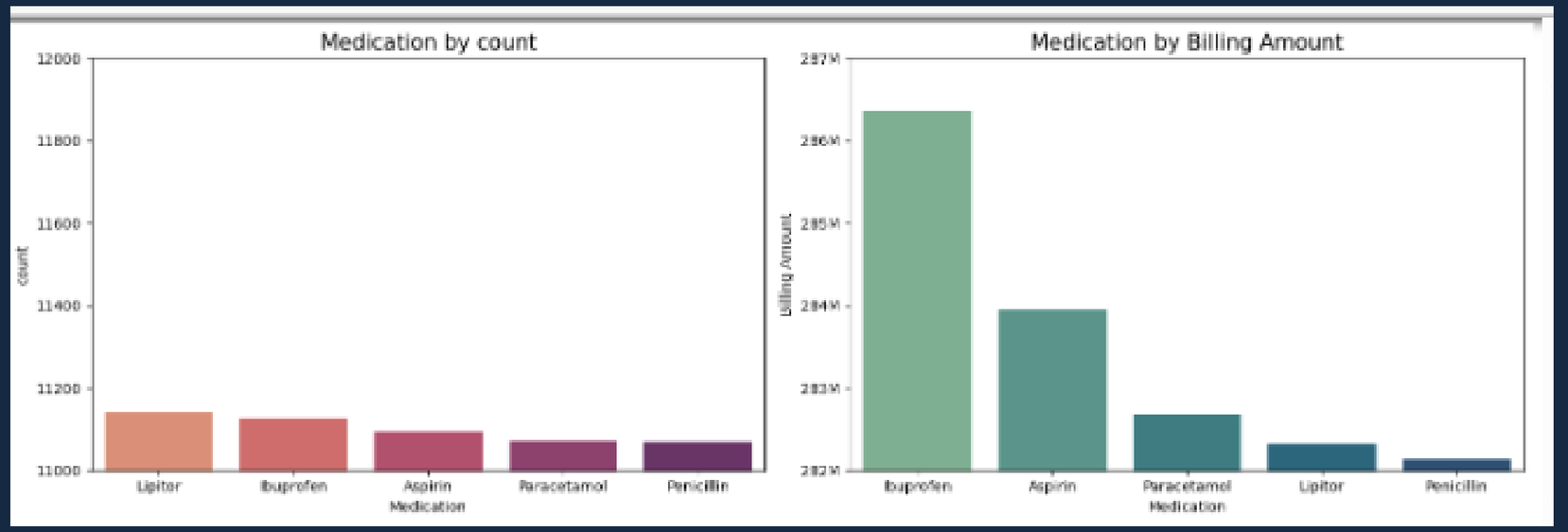

Gender by count
count
28000
27800
27600
27400
27200
27000
Male
Female
Gender
Gender by Billing Amount
Billing Amount
712M
711M
710M
709M
708M
707M
706M
Male
Female
Gender

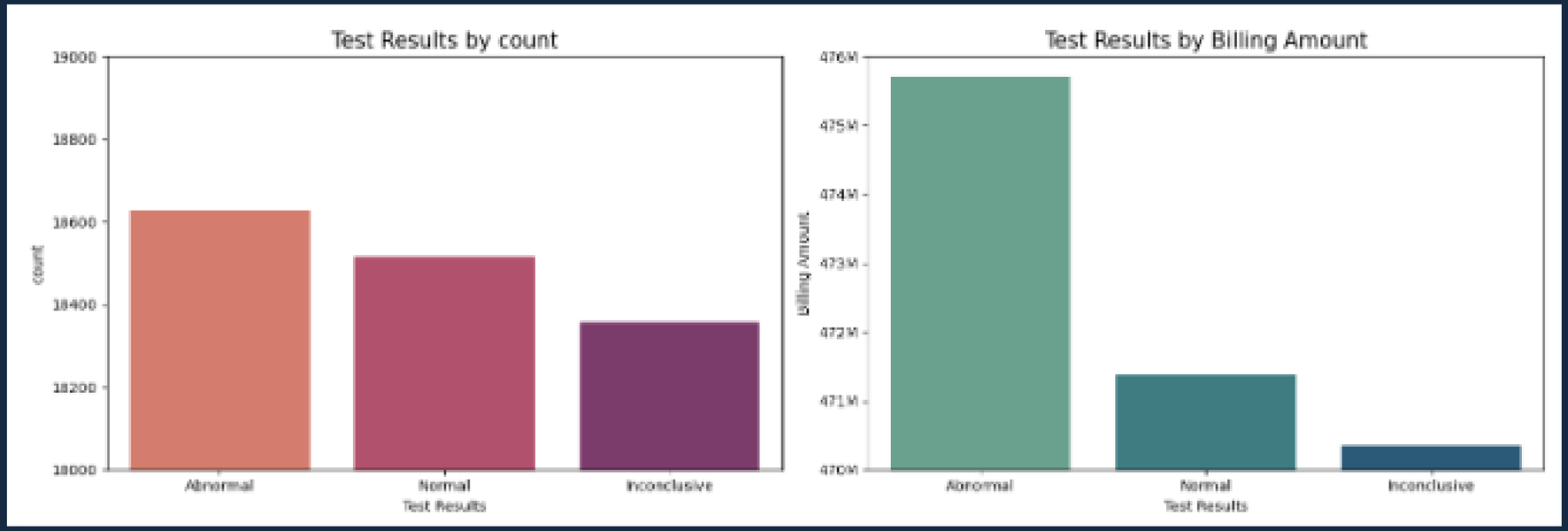

Blood Type by count
count
7000
6800
6600
6400
6200
6000
A-
A+
AB+
AB-
B+
B-
O+
O-
Blood Type
Blood Type by Billing Amount
Billing Amount
179M
178M
177M
176M
175M
174M
A+
AB-
A-
O-
B-
B+
AB+
O+
Blood Type


Blood Type by count
Count
7000
6800
6600
6400
6200
6000
A-
A+
AB+
AB-
B+
B-
O+
O-
Blood Type
Blood Type by Billing Amount
Billing Amount
179M
178M
177M
176M
175M
174M
A+
AB-
A-
O-
B-
B+
AB+
O+
Blood Type


Insurance Provider by count
count
12000
11750
11500
11250
11000
10750
10500
10250
10000
Cigna
Medicare
UnitedHealthcare
Blue Cross
Aetna
Insurance Provider
Insurance Provider by Billing Amount
Billing Amount
288M
286M
284M
282M
280M
278M
Cigna
Medicare
Blue Cross
UnitedHealthcare
Aetna
Insurance Provider


Insurance Provider by count
count
12000
11750
11500
11250
11000
10750
10500
10250
10000
Cigna
Medicare
UnitedHealthcare
Blue Cross
Aetna
Insurance Provider
Insurance Provider by Billing Amount
Billing Amount
288M
286M
284M
282M
280M
278M
Cigna
Medicare
Blue Cross
UnitedHealthcare
Aetna
Insurance Provider


Medication by count
count
12000
11800
11600
11400
11200
11000
Lipitor
Ibuprofen
Aspirin
Paracetamol
Penicillin
Medication
Medication by Billing Amount
Billing Amount
287M
286M
285M
284M
283M
282M
Ibuprofen
Aspirin
Paracetamol
Lipitor
Penicillin
Medication


Test Results by count
19000
18800
18600
18400
18200
18000
count
Abnormal
Normal
Inconclusive
Test Results
Test Results by Billing Amount
476M
475M
474M
473M
472M
471M
470M
Billing Amount
Abnormal
Normal
Inconclusive
Test Results

Préparation à la modélisation :

Encodage des variables catégorielles

```
df_encoded = df.copy()  #pour ne pas modifier le database originale
label_encoders = {}

for col in df_encoded.select_dtypes(include=['object']).columns:
    le = LabelEncoder()
    df_encoded[col] = le.fit_transform(df_encoded[col].astype(str))
    label_encoders[col] = le
```

Pour que les algorithmes de machine learning puissent traiter les données textuelles, toutes les colonnes catégorielles (textes) doivent être converties en valeurs numériques.

creation du modele:

```
y = df_encoded.iloc[:,-1]
y.values
```

```
array([2, 1, 2, ..., 0, 0, 0])
```

```
X = df_encoded.iloc[:,:-1].values
y = df_encoded.iloc[:,-1].values
```

Séparation entre les variables explicatives (X) et la cible (y) à prédire.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print(f'Training data shape {X_train.shape}')
print(f'Testing data shape {X_test.shape}')
```

```
Training data shape (44400, 14)
Testing data shape (11100, 14)
```

Division du dataset en 80% pour l’entraînement et 20% pour le test, avec un tirage aléatoire constant.

Entraînement des modèles

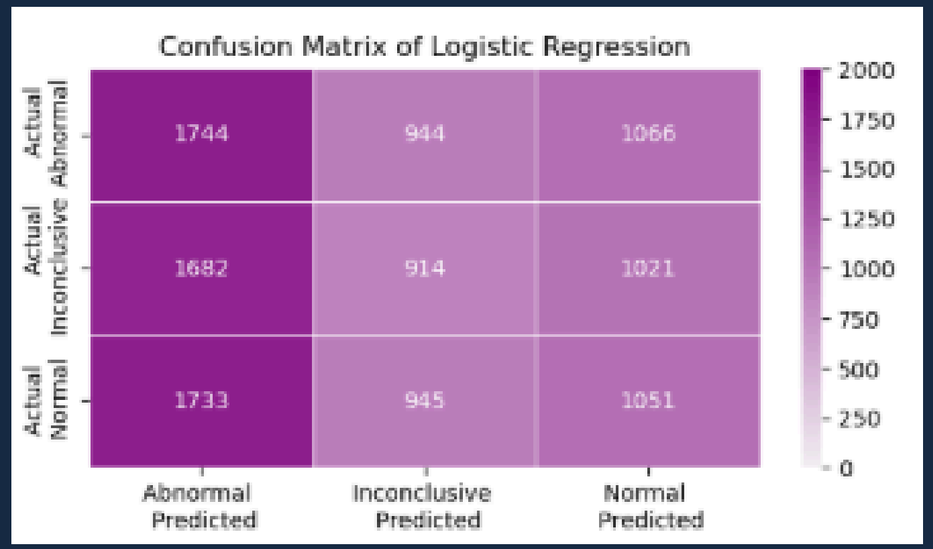
Entraînement de 4 modèles de classification, évaluation de leur performance avec la précision et la matrice de confusion.

```python
models = {
    "Logistic Regression":LogisticRegression(),
    "K Means": KNeighborsClassifier(),
    "Decision Tree": DecisionTreeClassifier(),
    "Random Forest": RandomForestClassifier()
}
results = {}
n = len(cols)
rows, cols_per_row = 2, 2  # adjust as needed
fig, axes = plt.subplots(rows, cols_per_row, figsize=(15, 8))
axes = axes.flatten()  # make 2D array into 1D for easy indexing
idx=0
for model_name, model in models.items():
    model.fit(X_train, y_train)
    predictions = model.predict(X_test)
    accuracy_scores = round(accuracy_score(y_test, predictions),3)
    results[model_name] = {'accuracy_scores': accuracy_scores}
    sns.heatmap(confusion_matrix(y_test, predictions), annot=True,  cmap=sns.light_palette("purple", as_cmap=True), linewidths=0
                xticklabels=['Abnormal \n Predicted', 'Inconclusive \n Predicted', 'Normal \n Predicted'],
                yticklabels=['Actual \n Abnormal', 'Actual \n Inconclusive', 'Actual \n Normal'], ax = axes[idx], vmin=0, vmax=2
    axes[idx].set_title(f'Confusion Matrix of {model_name}')
    print(f'\033[1mAccuracy score of {model_name} is {accuracy_scores}\033[1m \n')
    idx+=1

plt.subplots_adjust(wspace=0.25, hspace=0.5)
plt.show()
```

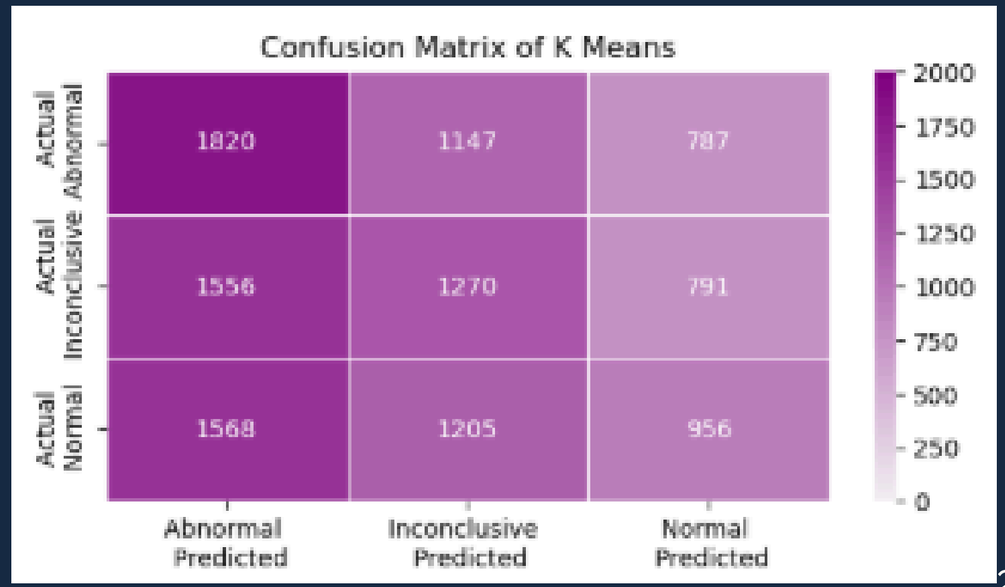
output:

```
Accuracy score of Logistic Regression is 0.334

Accuracy score of K Means is 0.365

Accuracy score of Decision Tree is 0.436

Accuracy score of Random Forest is 0.445
```

Evaluation des modèles

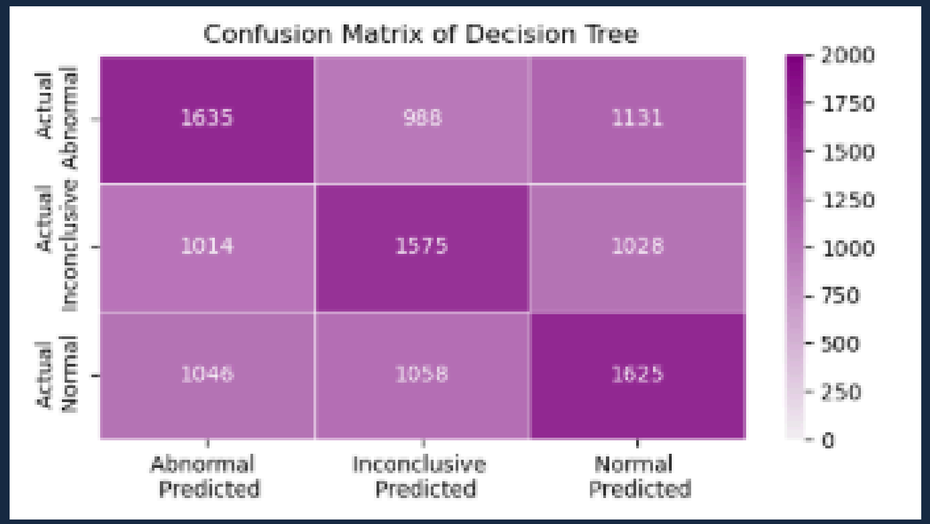


Interprétation :
Très peu de prédictions précises. Seules ~30-33% des prédictions sont correctes par classe.
Les classes sont fortement confondues, notamment entre "Abnormal" et "Normal".
Le modèle n'arrive pas à différencier les profils, indiquant une mauvaise capacité de généralisation.
📌 **Conclusion :** Trop d'erreurs. Modèle non adapté aux données dans sa forme actuelle.



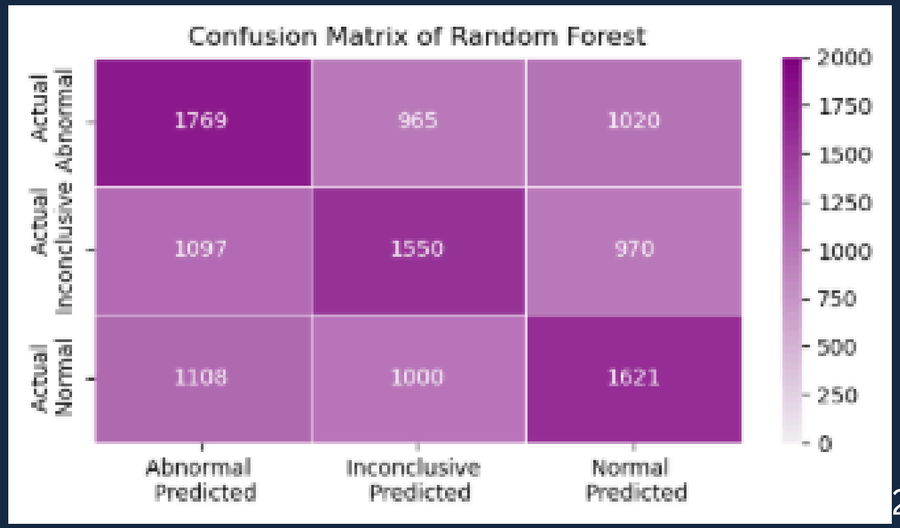
🗒️ **Interprétation :**
Léger gain en précision par rapport à la régression logistique.
Beaucoup de confusion entre "Abnormal" et "Inconclusive", et entre "Normal" et "Abnormal".
La répartition n'est pas homogène, ce qui montre que KNN suit les zones de densité locale sans bien généraliser.
📌 **Conclusion :** Meilleur que la régression, mais encore beaucoup de bruit.

Evaluation des modèles



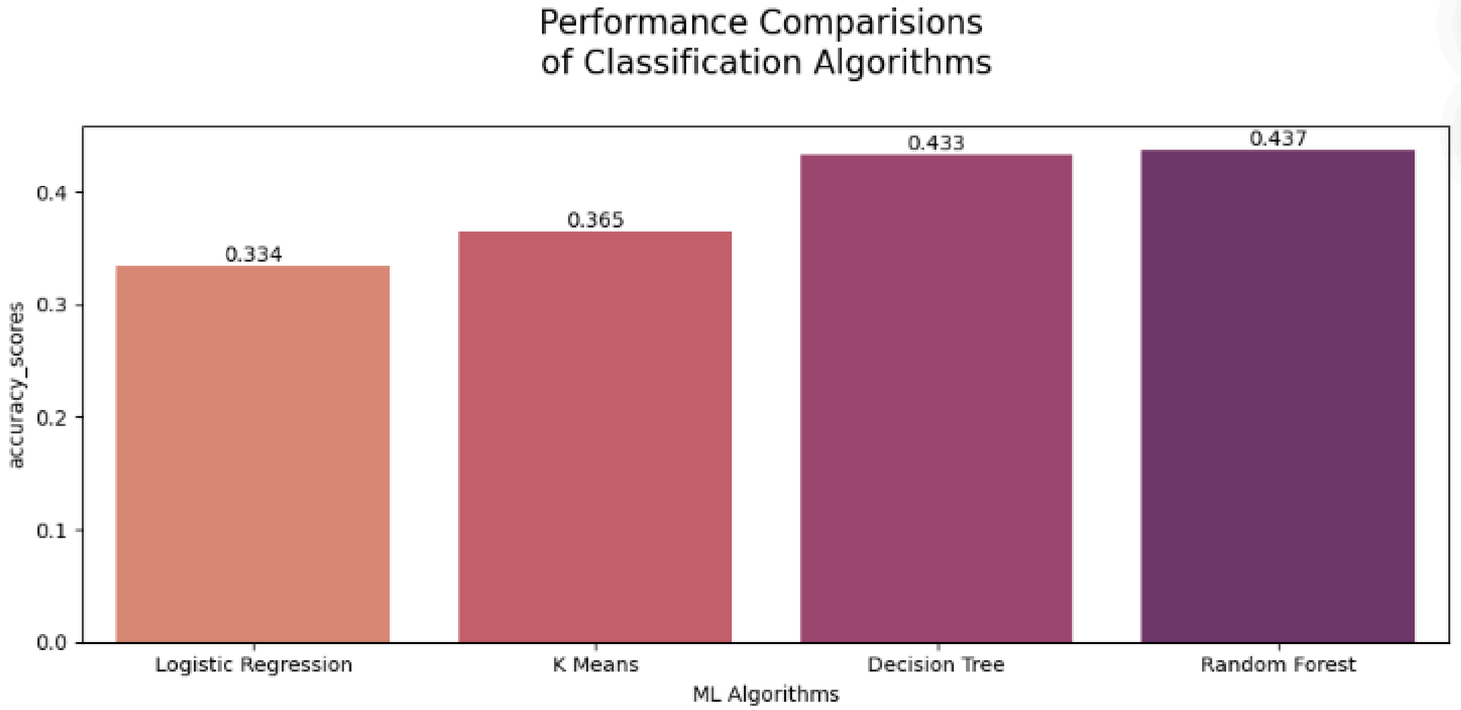
📄 **Interprétation :**
Bonne performance sur la classe Inconclusive (1575 prédictions correctes).
Le modèle commence à mieux distinguer la classe "Normal" (1625 bons).
Il reste des erreurs, surtout entre "Abnormal" et "Normal", mais la diagonale s'améliore.
📌 **Conclusion :** Meilleur équilibre, le modèle apprend des règles pertinentes.

📄 **Interprétation :**
Meilleures performances globales, surtout sur la classe "Normal" (1621).
Bonne séparation pour toutes les classes : la diagonale est dominante.
Les erreurs persistent mais sont moins marquées et mieux réparties.
📌 **Conclusion :** Modèle le plus robuste, avec une bonne capacité de généralisation.



Evaluation des modèles

```
]: results_df = pd.DataFrame(results).T
   plt.figure(figsize = (10,5))
   ax = sns.barplot(data = results_df, x=results_df.index, y=results_df['accurac
   ax.bar_label(ax.containers[0], fontsize=10);
   ax.set_xlabel('ML Algorithms')
   plt.title('Performance Comparisions \n of Classification Algorithms \n', font
   plt.tight_layout()
   plt.show()
```



Etape 3 : Conception du data warehouse et Alimentation du data warehouse avec Talend

1. Introduction
2. Définition des Faits et Mesures
3. Définition des Dimensions
4. visualisation sur talend
5. visualisation sur pg admin

Introduction

Cette partie présente la conception d’un Data Warehouse orienté santé, basé sur un schéma en étoile. Il intègre des dimensions descriptives (patients, médecins, hôpitaux, médicaments, dates) et une table de faits centralisant les mesures économiques.

Définition des Faits et Mesures

Faits :

• - dwh Fait : représente les interactions médicales (actes ou prescriptions)

Mesures :

• - montant : coût ou montant associé à l'acte médical

Définition des Dimensions

Dimension Temps (dwh dim-date) :

• - id-date • - date • - année • - mois • - jours

Dimension Médecin (dwh dim-doctor):

• - id-doctor • - name 1

Dimension Patient (dwh dim-patient) :

• - id-patient • - name • - age • - gender • - typeblood

Dimension Hôpital (dwh dim-hopitale) :

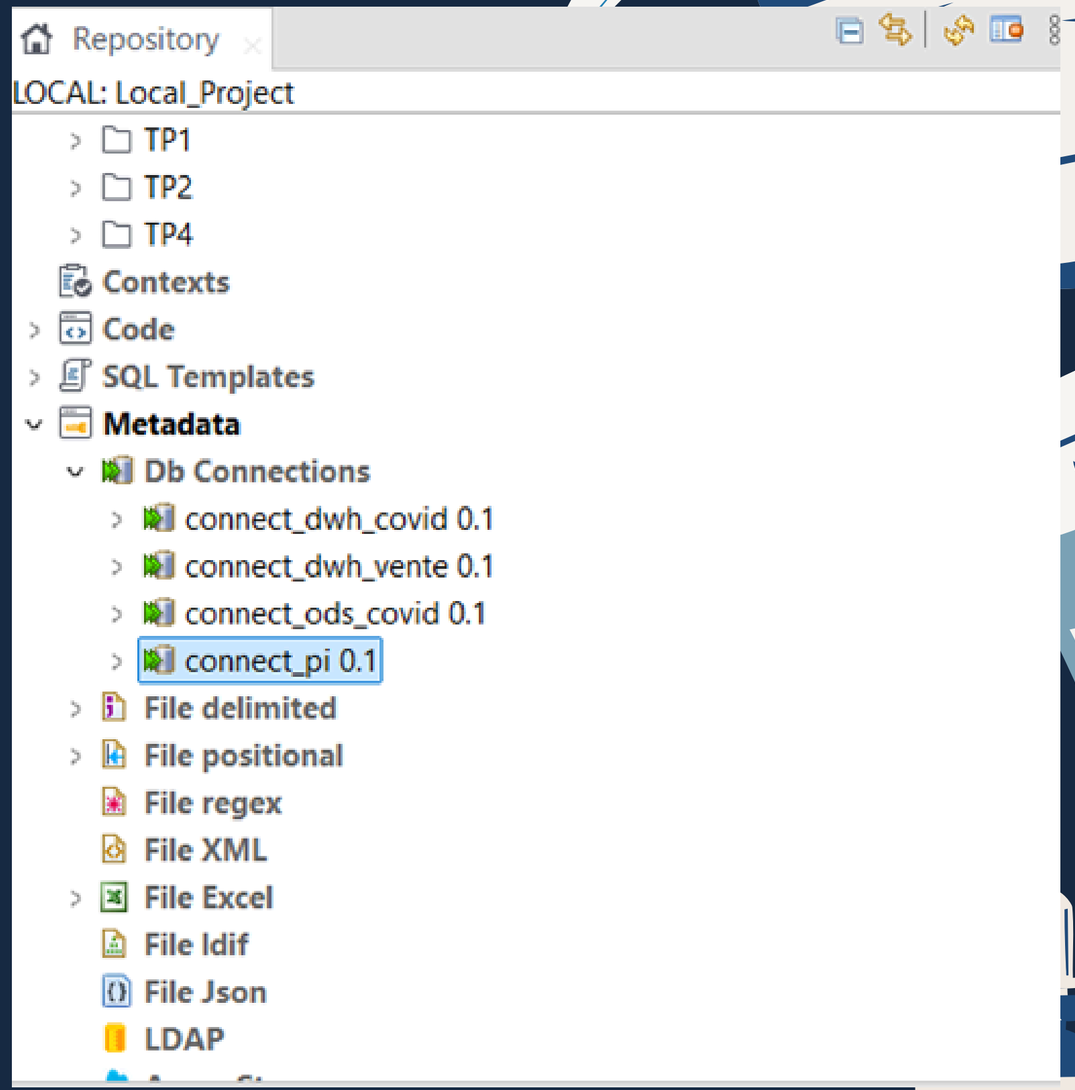
• - id-hopitale • - hopitale Dimension

Médicament (dwh dim-medicament) :

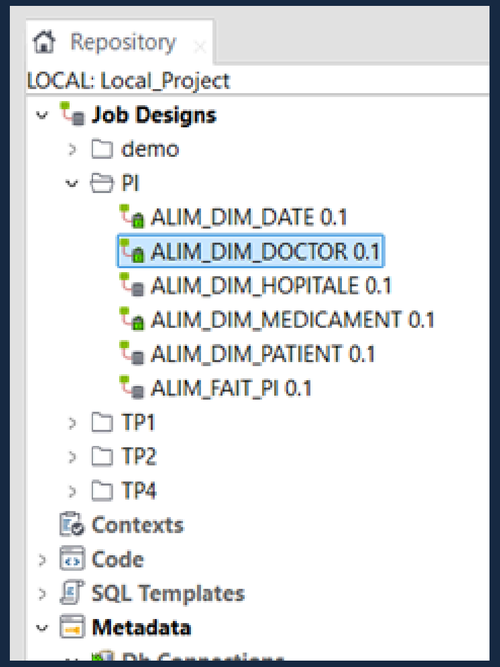
• - id-medicament • - name

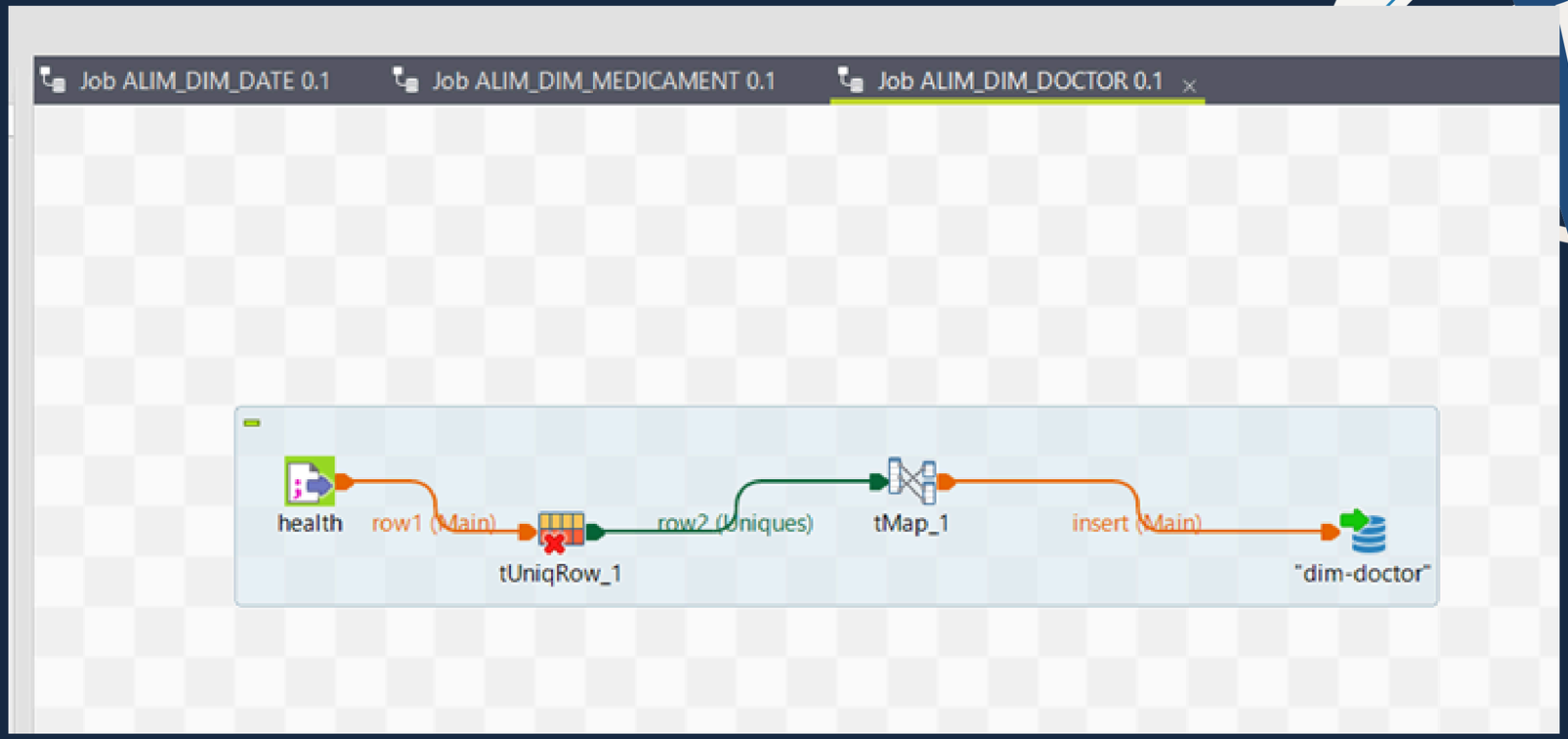
Visualitation talend

L’importation du base de donne et l’alimentation des variable sur chaque table



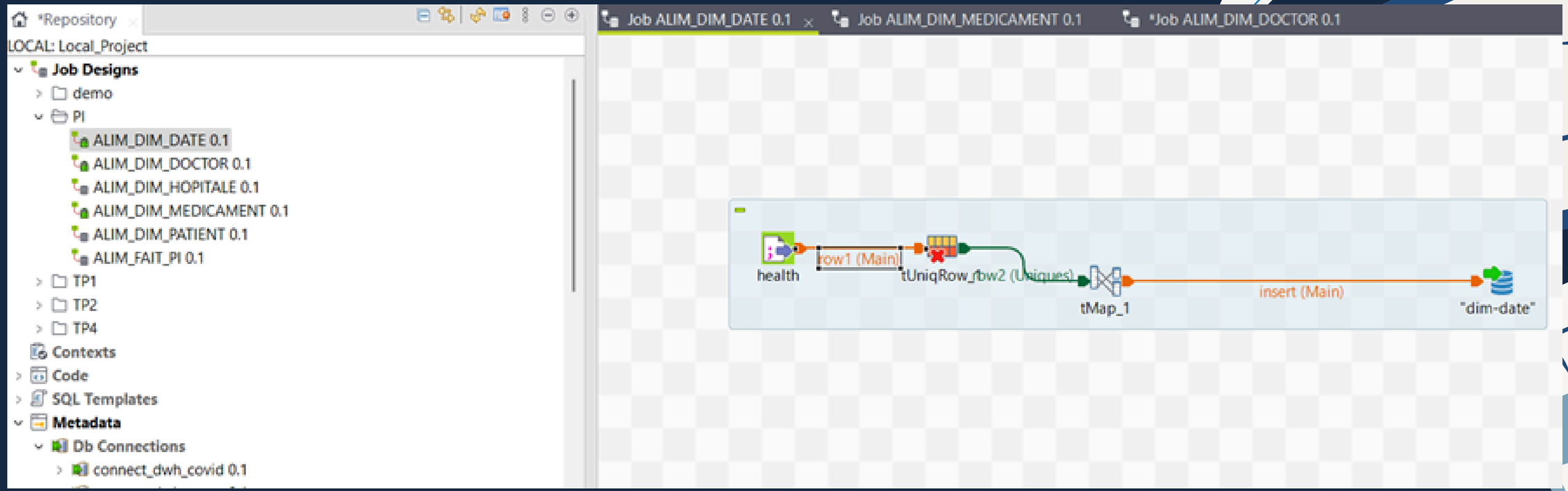
Visualitation talend





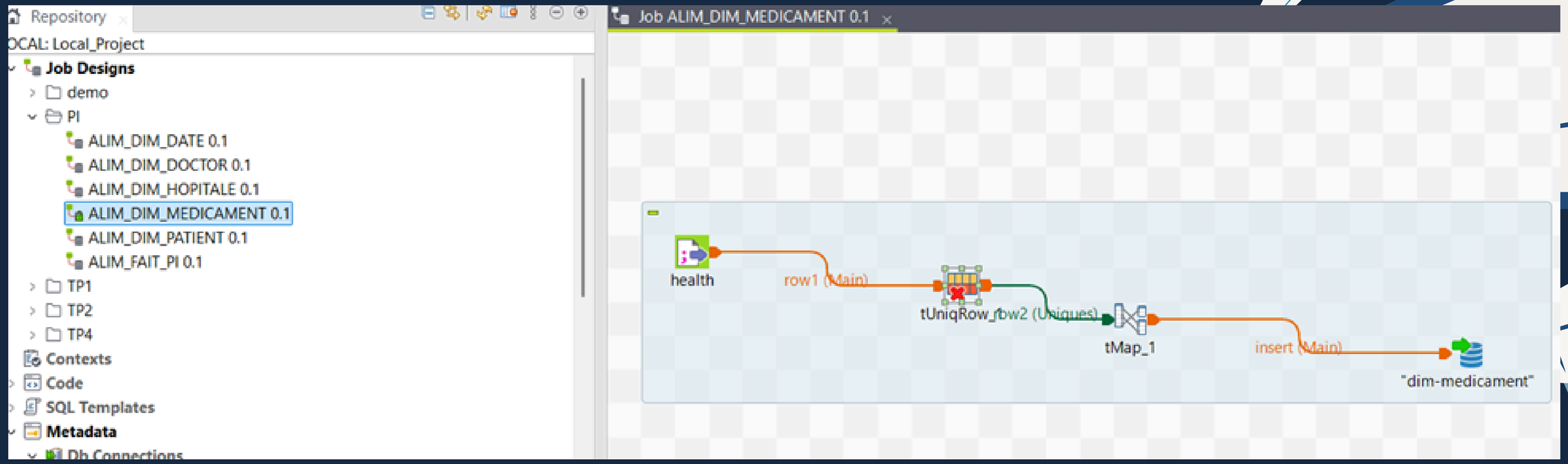
On a implémenter notre base donner avec create job format input et le table dim doctor output et en a fait une relation avec tmap pour la jointure des variables et aussi on a utilise la tunique nous affiche le name de doctor une seul foit pour avoir un plus de capacite pour les autre fonction

Visualitation talend



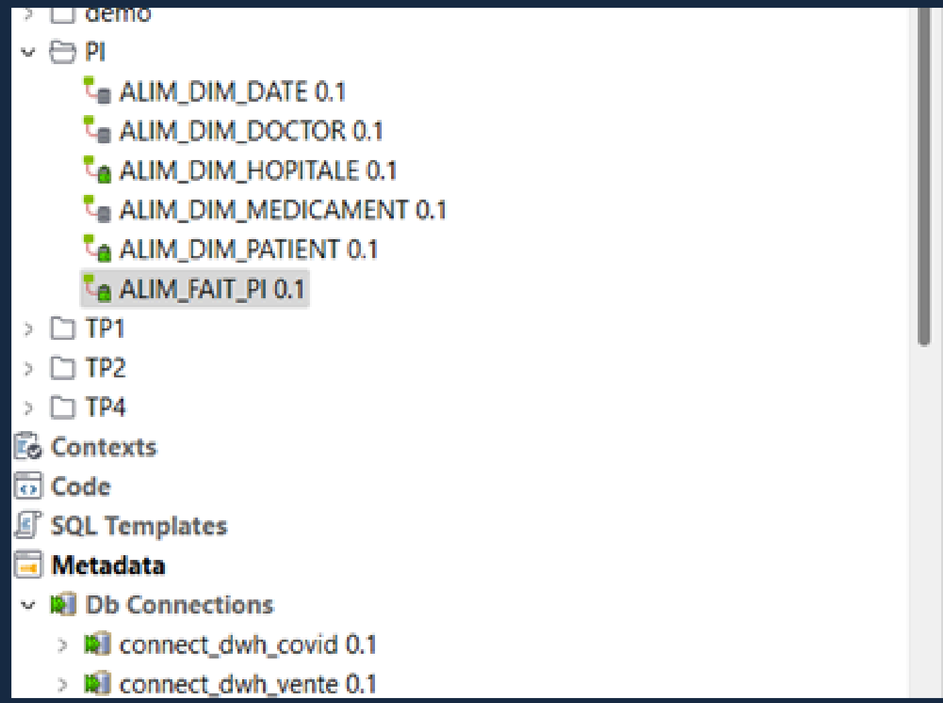
On a implémenter notre base donner avec create job format input et le table dim date output et en a fait une relation avec tmap pour la jointure des variables et aussi on a utilise la tunique nous affiche le date of addmission une seul foit pour avoir un plus de capacite pour les autre fonction

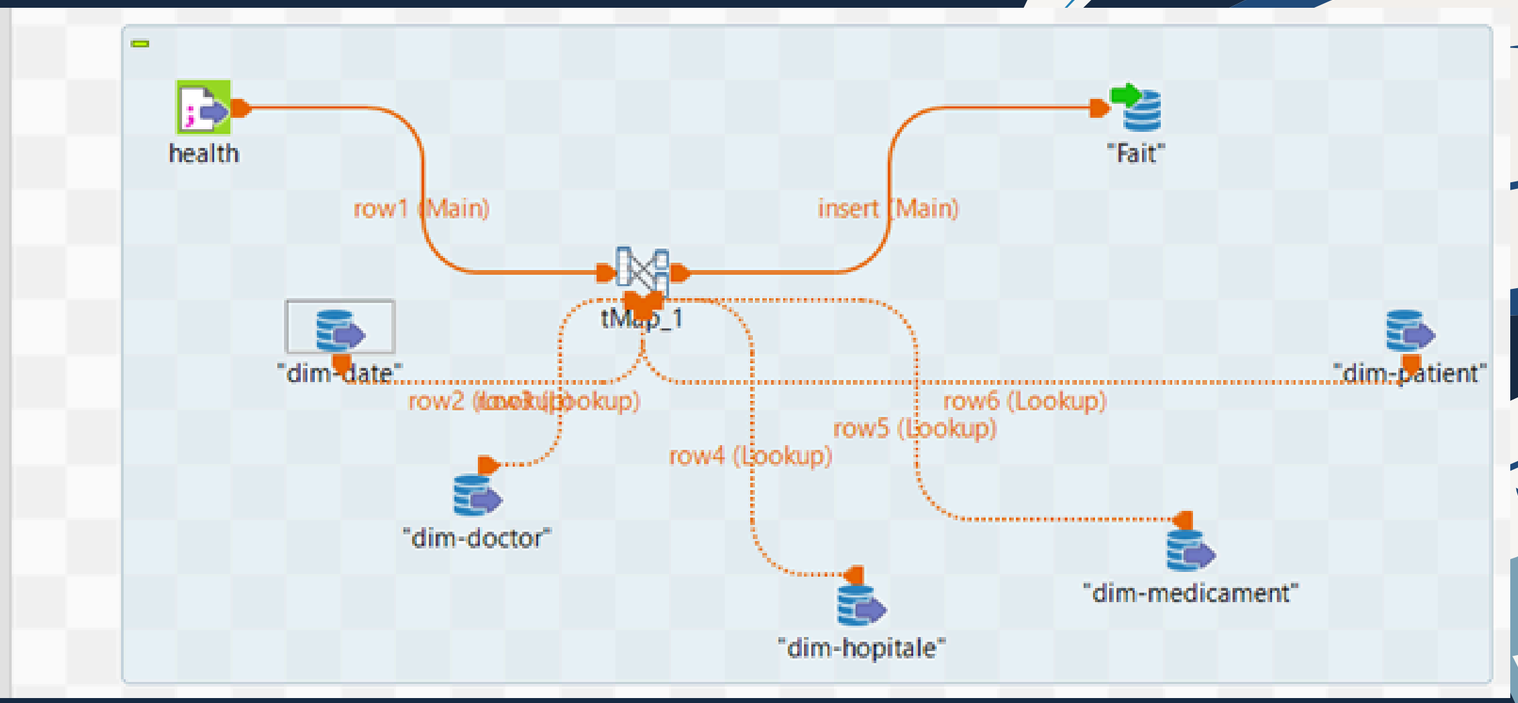
Visualitation talend



On a implémenter notre base donner avec create job format input et le table dim médicament output et en a fait une relation avec tmap pour la jointure des variables et aussi on a utilisé la tunique nous affiche le médication une seul foit pour avoir un plus de capacite pour les autre fonction

Visualitation talend



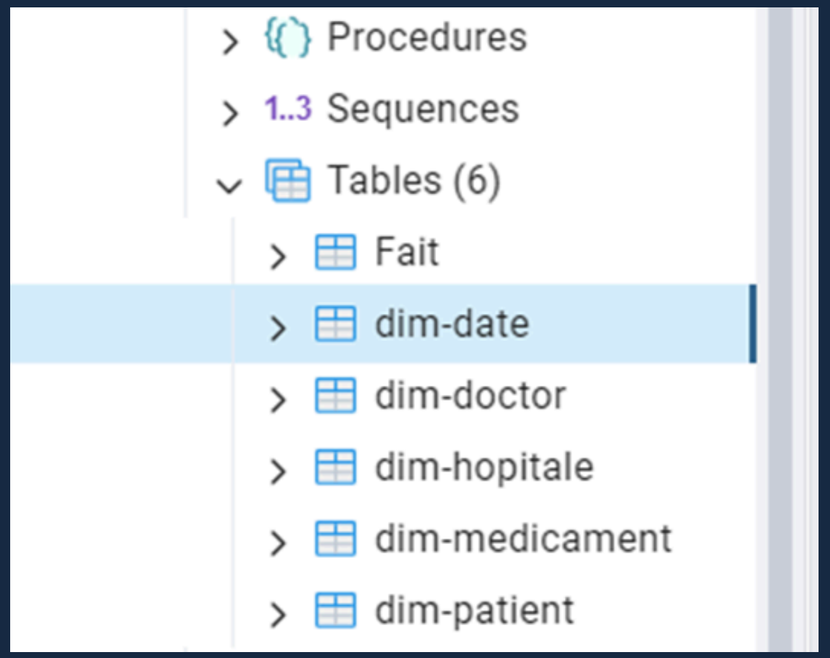


On implémenter le table fait avec format output et le table dim medicament / dim patient/ dim doctor /dim date /dim hospitale et base donnercomme un input et en a fait une relation avec tmap pour la jointure des variables

voir talend

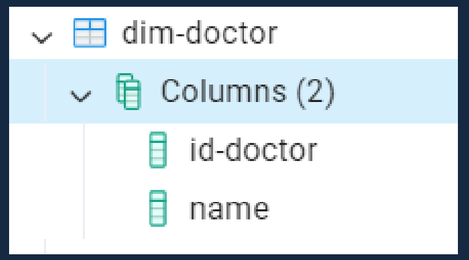
Visualitation PG admin

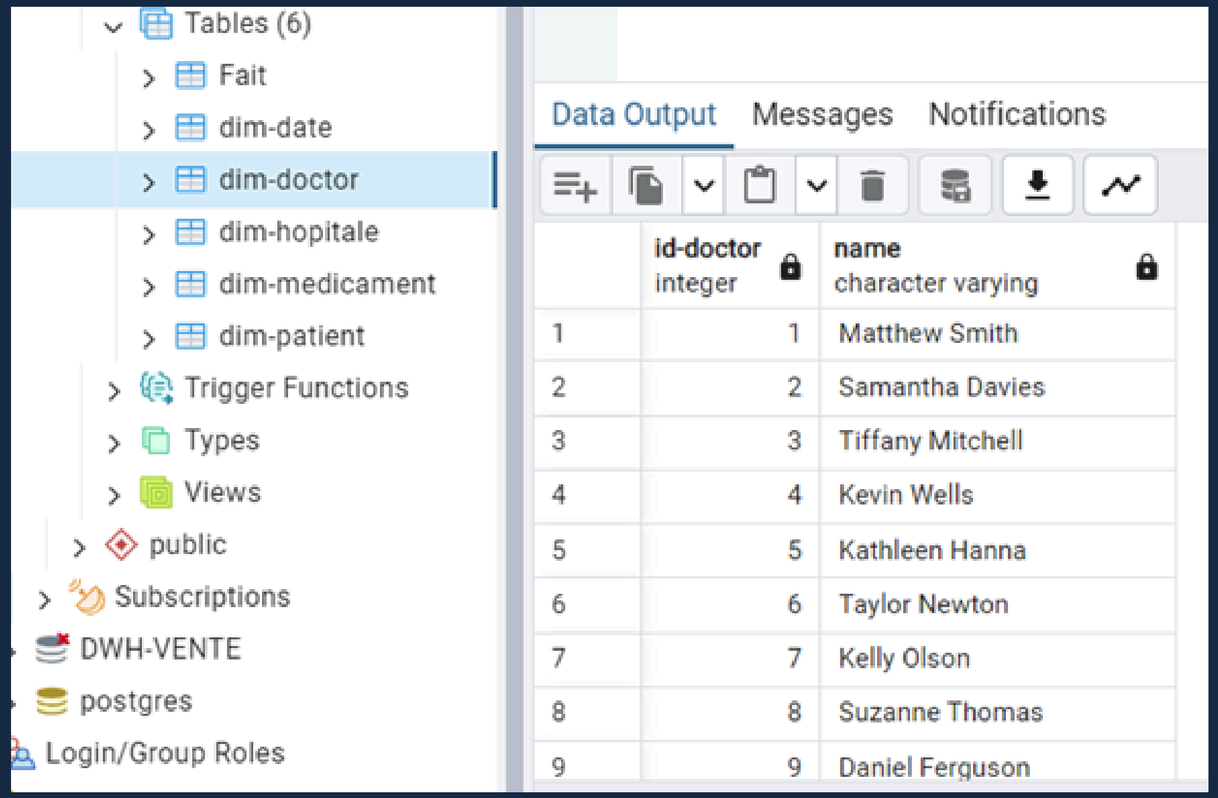
On a l’imprter notre base de donne format csv sur pg admin



Visualitation PG admin

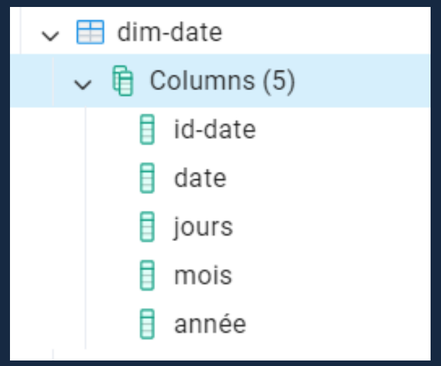
Et ca le output L'exucution sur Le pg admin du dim doctor (les id doctor et name)





Visualitation PG admin

Et ca le output L'exucution sur
Le pg admin du dim date(les id date date /jours/mois/annee)

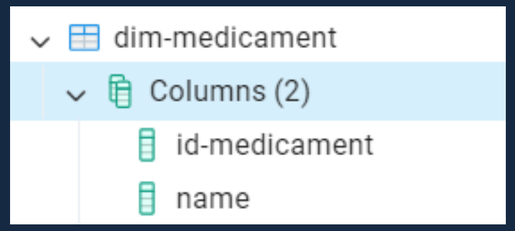


Data Output | Messages | Notifications

	id-date integer	date date	jours character varying	mois character varying	année character varying
1	1	2024-01-31	31	01	2024
2	2	2019-08-20	20	08	2019
3	3	2022-09-22	22	09	2022
4	4	2020-11-18	18	11	2020
5	5	2022-09-19	19	09	2022
6	6	2023-12-20	20	12	2023
7	7	2020-11-03	03	11	2020
8	8	2021-12-28	28	12	2021
9	9	2020-07-01	01	07	2020

Visualitation PG admin

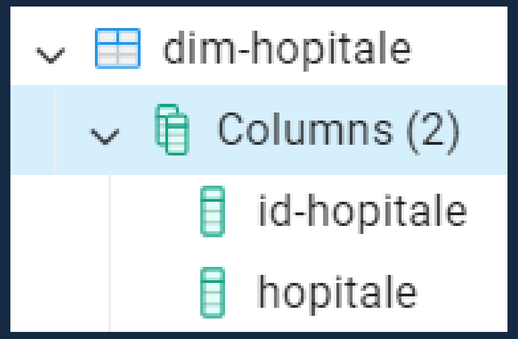
Et ca le output L’exucution sur
Le pg admin du dim médicament (les id medicament /name)



id-medicament integer	name character varying
1	Paracetamol
2	Ibuprofen
3	Aspirin
4	2020-12-18
5	Penicillin
6	2022-01-07
7	Lipitor

Visualitation PG admin

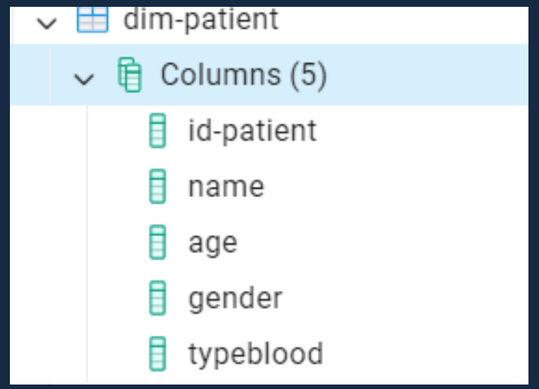
Et ca le output L’exucution sur
Le pg admin du dim hospital (les id hospital /hospital)



	id-hopitale integer	hopitale character varying
1	1	Sons and Miller
2	2	Kim Inc
3	3	Cook PLC
4	4	"Hernandez Rogers and Vang
5	5	White-White
6	6	Nunez-Humphrey
7	7	Group Middleton
8	8	"Powell Robinson and Valdez

Visualitation PG admin

Et ca le output L'exucution sur
Le pg admin du dim patient (les id patients / name/ gender/typeblood)

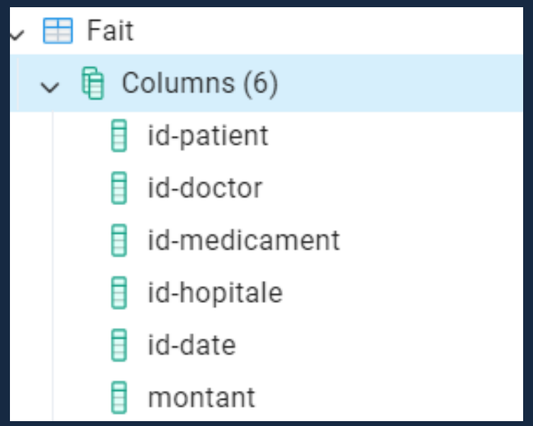


Data Output | Messages | Notifications

	id-patient integer	name character varying	age integer	gender character varying	typeblood character varying
1	1	Bobby JacksOn	30	Male	B-
2	2	LesLie TErRy	62	Male	A+
3	3	DaNnY sMitH	76	Female	A-
4	4	andrEw waTtS	28	Female	O+
5	5	adrIENNE bEll	43	Female	AB+
6	6	EMILY JOHNSOn	36	Male	A+
7	7	edwArD EDWaRDs	21	Female	AB-
8	8	CHrisTInA MARtinez	20	Female	A+
9	9	JASmINe aGuIlaR	82	Male	AB+

Visualitation PG admin

Et ca le output L'exucution sur Le pg admin du dim fait (id patient /id doctor/id medicament/ id hospital/ id date/ montant)



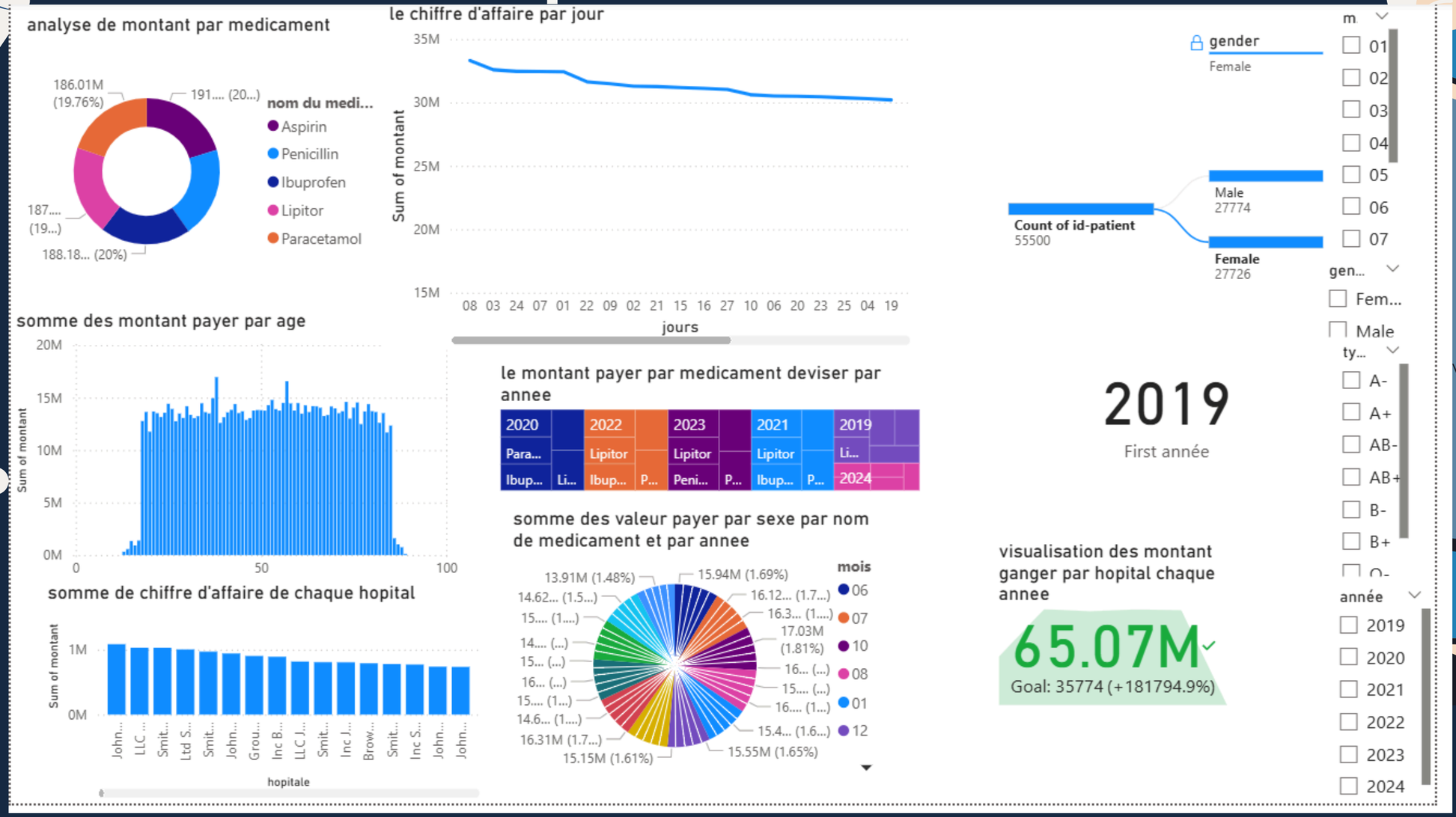
Data Output | Messages | Notifications

	id-patient integer	id-doctor integer	id-medicament integer	id-hopitale integer	id-date integer	montant numeric
1	1	1	1	1	1	18856.281305978155
2	2	2	2	2	2	33643.327286577885
3	3	3	3	3	3	27955.096078842456
4	50145	5	5	5	5	14238.317813937623
5	6	6	2	6	6	48145.11095104189
6	7	7	1	7	7	19580.87234486093
7	9	9	3	9	9	50119.222791548505
8	10	10	1	10	10	19784.63106221073
9	11	11	1	11	11	12576.795609050234

voir pg admin

Etape 4 :Visualisation des données avec Power BI

on va voir cette partie sur power bi



MERCI DE VOTRE ATTENTION

HEALTHCARE_DATASET