Une image contenant texte

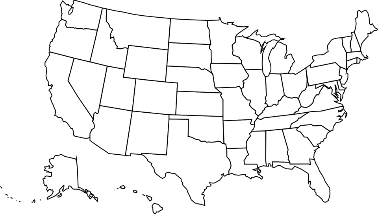
Description générée automatiquement

Masse de données et fouille de données

-

**Etude du Covid-19 aux Etats-Unis**

-

CHEVRIER Jean-Christophe

LUC Tristan

NOIROT Quentin

Sommaire

[1. Contexte 4](#_Toc98700133)

[2. Analyse exploratoire des données 4](#_Toc98700134)

[2.1. Volumétrie des données 4](#_Toc98700135)

[2.2. Structure des données 4](#_Toc98700136)

[2.3. Analyse exploratoire introductive 4](#_Toc98700137)

[2.5. Analyse exploratoire de la colonne « date » 4](#_Toc98700138)

[2.6. Analyse exploratoire de la colonne « county » 4](#_Toc98700139)

[2.7. Analyse exploratoire de la colonne « state » 4](#_Toc98700140)

[2.8. Analyse exploratoire de la colonne « fips » 4](#_Toc98700141)

[2.9. Analyse exploratoire de la colonne « cases » 4](#_Toc98700142)

[2.10. Analyse exploratoire de la colonne « deaths » 4](#_Toc98700143)

[2.11. Vérification de l’unicité du couple (date, comté) 4](#_Toc98700144)

[3. Nettoyage des données 4](#_Toc98700145)

[3.1. Suppression des lignes avec des valeurs manquantes 4](#_Toc98700146)

[3.2. Conversion au même format pour les dates 4](#_Toc98700147)

[3.3. Suppression des "-" dans les colonnes quantitatives 4](#_Toc98700148)

[3.4. Transformation des colonnes quantitatives en entiers 4](#_Toc98700149)

[4. Fouille des données 4](#_Toc98700150)

[4.1. Complétion des données 4](#_Toc98700151)

[4.2. Sélection des données 4](#_Toc98700152)

[4.3. Première normalisation des données 4](#_Toc98700153)

[4.4. Mélange des données 4](#_Toc98700154)

[4.5. Conception d’un modèle de prédiction du nombre de morts par régression 4](#_Toc98700155)

[4.5.1. Division en deux ensembles : données des prédictions, et du modèle 4](#_Toc98700156)

[4.5.2. Division en deux ensembles : variables explicatives, et variable résultat 4](#_Toc98700157)

[4.5.3. Division en deux ensembles : données des entrainements, et des tests 4](#_Toc98700158)

[4.5.4. Seconde normalisation des données 4](#_Toc98700159)

[4.5.5. Première configuration de modèle 4](#_Toc98700160)

[4.5.6. Deuxième configuration de modèle : plus de neurones 5](#_Toc98700161)

[4.5.7. Troisième configuration de modèle : plus de paquets 5](#_Toc98700162)

[4.5.8. Quatrième configuration de modèle : taux d’apprentissage de 5% 5](#_Toc98700163)

[4.5.9. Cinquième configuration de modèle : taux d’apprentissage de 0.1% 5](#_Toc98700164)

[4.5.10. Sixième configuration de modèle : taux d’apprentissage de 0.01% 5](#_Toc98700165)

[4.6. Conception d’un modèle de prédiction du nombre de morts par classification 5](#_Toc98700166)

[4.6.1. Classification des observations par classes de nombre de morts 5](#_Toc98700167)

[4.6.2. Création de l’architecture du modèle 5](#_Toc98700168)

[4.6.3. Test manuel de la justesse de prédiction du modèle 5](#_Toc98700169)

[4.6.4. Etude du modèle à partir de graphique 5](#_Toc98700170)

[5. Conclusion 5](#_Toc98700171)

# 1. Contexte

Dans le cadre de notre formation de Master 2 MIAGE SID, et pour le module de masse de données et de fouille de données, nous avions reçu la mission d’étudier un jeu de données.

Ce jeu de données comportait des **données quotidiennes portant sur l’épidémie de Covid-19 aux Etats-Unis**.

Afin de réaliser une étude approfondie de ce jeu données, nous avons réalisé une courte **pipeline BI**, qui était celle-ci :



Il s’agissait donc d’enchainer ces 3 étapes afin d’arriver au bout à une étude complète.

Nous avons réalisé cette étude avec le langage informatique **Python** dans des scripts sur **Jupyter Notebook**. Pour résumer au mieux notre travail : nous avons réalisé un script pour chaque étape de la pipeline citée précédemment, soit 3 scripts.

Nous avons fait en sorte de bien documenté notre travail, les scripts sont commentés en Markdown, et accessibles via le répertoire GitHub public de notre projet : <https://github.com/LordOfRaptor/Study-USA-Covid-19-Data>. Les scripts étant dans son répertoire « src/ ».

Ce rapport, dans son plan, suit la pipeline BI que nous avons exécuté, et d’ailleurs, son plan est globalement calqué sur les parties et sous-parties des scripts de notre étude. Nous vous invitons donc à les consulter en parallèle de votre lecture.

La finalité de notre pipeline BI était d’avoir en aval de cette dernière, des modèles de prédiction cohérents de l’évolution des morts de l’épidémie aux Etats-Unis. C ’est la problématique que nous avons choisi : **pouvoir prédire l’évolution des morts de l’épidémie** donc.

# 2. Analyse exploratoire des données

Nous avons donc commencé notre étude en faisant une analyse exploratoire des données du jeu de données.

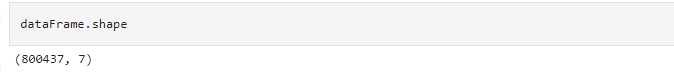
D’ailleurs voici le script de l’étape sur GitHub : <https://github.com/LordOfRaptor/Study-USA-Covid-19-Data/blob/main/src/script_exploratory_data_analysis.ipynb>.

Ce que nous entendons par analyse exploratoire des données, c’est en fait de regarder la **structure** des données, leur **volumétrie**, et surtout leurs problèmes : les **données manquantes et incohérentes**.

Cette étape avait en fait essentiellement pour but de **« diagnostiquer » les problèmes du jeu de données**, qui seraient à colmater dans l’étape suivante de nettoyage.

## 2.1. Volumétrie des données

En commençant à travailler sur les données, nous avons pu voir leur volumétrie :



Le jeu de données de base, non nettoyé donc, comporte 7 colonnes, et 800 437 lignes / observations.

L’essentiel à voir ici, est que la volumétrie des données est donc réductible à cet **ordre de grandeur : 100 000, soit 5 zéros** donc.

Cela nous a servi d’indicateur sur la complexité algorithmique qui serait à l'œuvre au moment de faire des opérations lourdes sur les données.

On peut ajouter que dans le monde actuel de la Big Data, des masses de données donc, cela reste une volumétrie correcte, comparée aux échelles existantes de données actuellement.

## 2.2. Structure des données

Ensuite, c’est bien de connaitre la taille des données mais l’essentiel pour nous était surtout de comprendre sa structure, à savoir ses colonnes, les dépendances entre celles-ci, etc.



Ce jeu de données **associe à une date et à un comté d'un état (colonnes "date", "county", "fips" et "state"), un nombre de cas et un nombre de morts du Covid-19 aux Etats-Unis (colonnes "cases", et "deaths")**.

Les comtés des états aux Etats-Unis ont des fips, il s’agit de sorte de code postal : <https://fr.wikipedia.org/wiki/Federal_Information_Processing_Standard>.

En termes de dépendance, on voit bien que pour un comté, on trouve toujours le même état, et également comté et fips sont des données qui font doublon, elles sont équivalentes.

## 2.3. Analyse exploratoire introductive

Par la suite, nous avons enchainer sur les étapes de diagnostic des problèmes dans les données. Afin de faire cela, nous avons commencé par une analyse exploratoire introductive, de recherche des valeurs manquantes au sens de cellules vides. Pour faire cela, nous avons utilisé deux fonctions, dont nous devons expliciter le fonctionnement.

La fonction len() appliquée à une colonne retourne le nombre de cellules total de cette dernière, elle n’omet pas les cellules vide. La fonction count() quant à elle renvoie sur une colonne le nombre de cellules non vides, soit le nombre de cellules qui contienne une valeur. La différence de ces fonctions retourne le nombre de données manquantes (données manquantes au sens cellules vides ici, on le rappelle) .

Une image contenant texte

Description générée automatiquement

Ainsi, avec cette première analyse introductive, on voyait déjà que toutes les colonnes, hormis la colonne d’identifiant, comportaient des données manquantes. Ce qui n’est pas une bonne nouvelle.

Ensuite, nous avons enchainé sur des analyses exploratoires individuelles de chaque colonne, dont celle de l’identifiant. Etant donné que cette colonne après analyse a été diagnostiquée sans problème, et étant donné qu’elle pas spécialement intéressante dans le cadre du rapport, nous n’évoquerons pas son analyse ici.

## 

## 2.5. Analyse exploratoire de la colonne « date »

Pour chaque colonne, nous avons effectué des analyses pour déterminer l’existence ou non de données manquantes et incohérentes. Ca a été à chaque fois le même protocole, homis pour la partie données incohérentes ou à il fallait analysé en fonction à chaque fois de ce que représente la colonne.

La colonne « date » comportait des données manquantes, 138 pour être exact. Une image contenant texte

Description générée automatiquement

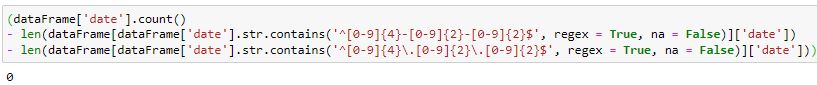
Pour l’étude la cohérence de ses valeurs, comme pour les autres colonnes, nous avons utilisé des expressions régulières. Et nous nous sommes rendus compte d’un certain problème : les dates étaient enregistrés sous deux formats différents dans la colonne.

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement



Il y avait ces deux formats : « YYYY-MM-DD » et « YYYY.MM.DD ».

La colonne « date » comportait des donc des données manquantes et des données incohérentes.

## 2.6. Analyse exploratoire de la colonne « county »

Pour la colonne « county », nous avons trouvé deux formes de données manquantes : des données manquantes sous forme de cellules vides (130 lignes), et des données manquantes renseignées explicitement avec le mot clé « Unknown » (6870 lignes).

Une image contenant texte

Description générée automatiquement

Une image contenant texte

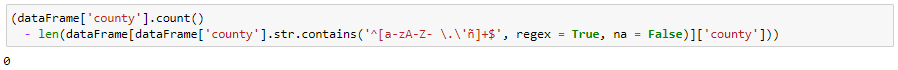
Description générée automatiquement

En revanche, nous n’avons pas trouvé de données incohérentes, l’ensemble des comtés matchaient avec une expression régulière cohérente avec la manière dont pourrait être nommé un comté, pas de chiffre, de moins devant les noms, etc.

Une image contenant texte

Description générée automatiquement





La colonne « county » comportait donc des valeurs manquantes de deux types, mais ne contenaient pas de valeurs incohérentes.

## 2.7. Analyse exploratoire de la colonne « state »

La colonne « state » comportait des valeurs manquantes, d’un seul type : des cellules vides (134 lignes pour être exact).

Une image contenant texte

Description générée automatiquement

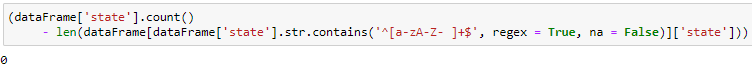
Une image contenant texte

Description générée automatiquement

De même que pour la colonne « county », la colonne « state » ne comportait pas de valeurs incohérentes, ses valeurs matchant avec une expression là aussi cohérente pour le nom qui peut être attendu pour un état.

Une image contenant texte

Description générée automatiquement



La colonne « state » ne colportait donc pas de valeur

## 2.8. Analyse exploratoire de la colonne « fips »

## 2.9. Analyse exploratoire de la colonne « cases »

## 2.10. Analyse exploratoire de la colonne « deaths »

## 2.11. Vérification de l’unicité du couple (date, comté)

# 3. Nettoyage des données

## 3.1. Suppression des lignes avec des valeurs manquantes

## 3.2. Conversion au même format pour les dates

## 3.3. Suppression des "-" dans les colonnes quantitatives

## 3.4. Transformation des colonnes quantitatives en entiers

## 4. Fouille des données

## 4.1. Complétion des données

## 4.2. Sélection des données

## 4.3. Première normalisation des données

## 4.4. Mélange des données

## 4.5. Conception d’un modèle de prédiction du nombre de morts par régression

### 4.5.1. Division en deux ensembles : données des prédictions, et du modèle

### 4.5.2. Division en deux ensembles : variables explicatives, et variable résultat

### 4.5.3. Division en deux ensembles : données des entrainements, et des tests

### 4.5.4. Seconde normalisation des données

### 4.5.5. Première configuration de modèle

#### 4.5.5.1. Création de l’architecture du modèle 4.5.5.2. Test manuel de la justesse de prédiction du modèle

#### 4.5.5.3. Etude du modèle à partir de graphique

### 4.5.6. Deuxième configuration de modèle : plus de neurones

### 4.5.7. Troisième configuration de modèle : plus de paquets

### 4.5.8. Quatrième configuration de modèle : taux d’apprentissage de 5%

### 4.5.9. Cinquième configuration de modèle : taux d’apprentissage de 0.1%

### 4.5.10. Sixième configuration de modèle : taux d’apprentissage de 0.01%

## 4.6. Conception d’un modèle de prédiction du nombre de morts par classification

### 4.6.1. Classification des observations par classes de nombre de morts

### 4.6.2. Création de l’architecture du modèle

### 4.6.3. Test manuel de la justesse de prédiction du modèle

### 4.6.4. Etude du modèle à partir de graphique

# 5. Conclusion