



Université Cadi Ayyad

Faculté des Sciences Semlalia Marrakech
Département d'Informatique

Master spécialisé Sciences des Données

Projet de module Machine Learning

Thème

PRÉVISIONS ET SIMULATIONS DE L'ÉPIDÉMIE COVID-19 AU MAROC AVEC MACHINE LEARNING



Encadré par :

Mme. Hajar MOUSANNIF

Réalisé Par :

GARBAZ Anass, OUKDACH Yassine, BOUH Youssef, HAKEM Adnane

Remerciement

Il nous est agréable de nous acquitter d'une dette de reconnaissance auprès de toutes les personnes, dont l'intervention au cours de ce projet, a favorisé son aboutissement.

Nous remercions nos chers parents pour leur amour et les efforts qu'ils ont fait pour notre éducation et notre scolarisation.

Nos sincères gratitudee à notre professeur, Mme. Hajar MOUSANNIF, pour ses conseils, ses directives tout au long de ce travail, ainsi que son intérêt incontestable qu'il porte à tous les étudiants.

Ainsi Tous les enseignants du département d'Informatique de la faculté des Sciences Semlalia après cette année d'études.

On souhaite que ce travail soit à la hauteur du niveau estimé.

Table des matières :

Remerciement :	1
Introduction Générale:	3
Chapitre 1 : Machine Learning et Covid-19	4
Introduction :	5
1- Machine Learning :	6
Définition :	6
Comment ça marche ?	6
2- Prophet :	6
Définition :	6
Exemple :	7
Pourquoi le modèle Prophet :	9
Équation du prophète :	10
3 - Le Machine Learning en Covid-19 :	11
Conclusion :	11
Chapitre 2 : Analyse	
Introduction :	13
1 - Analyse de Système :	13
Présentation générale de l'application :	13
Analyse de l'application :	13
Conclusion :	14
Chapitre 3 : L'environnement technique	
Introduction :	16
1- Le choix d'une application avec python :	16
Simple :	16
fonction :	17
2- les outils techniques :	19
pygame :	19
Google colab :	20
VScode :	21
Conclusion :	21
Chapitre 4 : L'implémentation	
Introduction :	23

Application :	23
Conclusion :	38
Conclusion générale	43
Bibliographie	44

Introduction Générale

Le projet d'un module apparaît comme un élément très important dans la mesure où il permet de faire adapter l'étudiant avec son futur environnement (le marché du travail), de lui offrir une possibilité d'élargir ses connaissances théoriques et de mettre en pratique son savoir. Au sein de la faculté des sciences semlalia en semestre 2, on a effectué notre projet dans le département d'informatique. Durant la période de s2, notre objectif était la réalisation d'une application de Machine Learning pour analyser les données de COVID-19 en effectuant des prédictions des futurs cas. Et qui permet aussi de faire des simulations pour des différents cas (sans quarantaine, avec quarantaine, ...). Se tirant de ce qui précède, ce rapport met en valeur la problématique de COVID-19. Or selon la majorité des communiqués et surtout de la Direction générale de la sûreté nationale (DGSN), Les cas infectés par le virus sont en évolution de jour en jour et varie d'une collectivité à l'autre. Au-delà des risques de se propager, le virus est également exposé à des risques physiques, posturaux, psychosociaux. Ce rapport présente l'une des solutions les plus intéressantes pour avoir comment le virus se propage et quand et prédire des cas de future. Ces choses vont nous aider d'avoir une idée pour éviter ce risque.

Ce rapport comporte 3 chapitres :

- Chapitre 1 : sera consacré à la présentation de Machine Learning en parlant de Modèle Prophet qui utilisé pour la prédiction des cas et le Covid-19.
- Chapitre 2 : contient l'analyse de l'application.
- Chapitre 3 : est pour les outils techniques utilisés.
- Chapitre 4 : contient l'architecture générale et des captures d'écran de l'application.

Chapitre 1 : Machine Learning et Covid-19

Introduction

Le Machine Learning est une étude systématique qui fonctionne dans de nombreux territoires logiques au fil des ans. Le Covid-19 est l'un de ces territoires, et de nombreux Data scientists ont bien compris l'importance de l'exploitation des informations dans la gestion d'une masse d'informations sur le Covid-19, et ils ont constaté les avantages des algorithmes de ML pour récupérer ou examiner ces informations.

Ce chapitre est composé de 3 grandes parties, La 1ere partie est pour définir le Machine Learning, En parlant de Modèle Prophet avec un exemple. Tandis que, la partie 3 décrit le virus en donnant les statistiques données par l'organisation mondiale de la santé (OMS). Finalement, on parle sur le Machine Learning en Covid-19, c'est-à-dire, l'application de Modèle de Machine Learning en Covid-19 qui est grâce au Prophet.

1- Machine Learning

Définition :

Le Machine Learning est une technologie d'intelligence artificielle permettant aux ordinateurs d'apprendre sans avoir été programmés explicitement à cet effet. Pour apprendre et se développer, les ordinateurs ont toutefois besoin de données à analyser et sur lesquelles s'entraîner.

Comment ça marche ?

Le Machine Learning est idéal pour exploiter les opportunités cachées du Big Data. Cette technologie permet d'extraire de la valeur en provenance de sources de données massives et variées sans avoir besoin de compter sur un humain. Elle est dirigée par les données, et convient à la complexité des immenses sources de données du Big Data. Contrairement aux outils analytiques traditionnels, il peut également être appliqué aux ensembles de données croissants. Plus les données injectées à un système Machine Learning sont nombreuses, plus ce système peut apprendre et appliquer les résultats à des insights de qualité supérieure. Le Machine Learning permet ainsi de découvrir les patterns enfouies dans les données avec plus d'efficacité que l'intelligence humaine.

2- Prophet :

Définition :

Prophet est une procédure de prévision des données de séries chronologiques basée sur un modèle additif où les tendances non linéaires sont adaptées à la saisonnalité annuelle, hebdomadaire et quotidienne, ainsi qu'aux effets des vacances. Cela fonctionne mieux avec des séries chronologiques qui ont de forts effets saisonniers et plusieurs saisons de données historiques. Prophet est robuste aux données manquantes et aux changements de tendance, et gère généralement bien les valeurs aberrantes.

L'entrée de Prophet est toujours une trame de données avec deux colonnes: ds et y. La ds colonne (datestamp) doit être d'un format attendu par les Pandas, idéalement AAAA-MM-JJ pour une date ou AAAA-MM-JJ HH: MM: SS pour un horodatage. La y colonne doit être numérique et représente la mesure que nous souhaitons prévoir.

Exemple :

Il faut d'abord importer le Modèle Prophet et notre data

```
import pandas as pd
from fbprophet import Prophet

df = pd.read_csv('MA-times_series - MA-times_series.csv')
df = df.groupby('Dates / التواريخ').sum()['Cases / الحالات'].reset_index()
df.head()
```

	Dates / التواريخ	Cases / الحالات
0	03/13/2020	8
1	03/14/2020	18
2	03/15/2020	29
3	03/16/2020	38
4	03/18/2020	54

On change les noms des colonnes si on veut, et il faut réformer la date si c'est nécessaire

```
[ ] df.columns = ['ds','y']
    df['ds'] = pd.to_datetime(df['ds'])
    df.head()
```

	ds	y
0	2020-03-13	8
1	2020-03-14	18
2	2020-03-15	29
3	2020-03-16	38
4	2020-03-18	54

Nous ajustons le modèle en instanciant un nouvel Prophet objet. Tous les paramètres de la procédure de prévision sont transmis au constructeur. Ensuite, vous appelez sa fit méthode et transmettez la trame de données historique. Le montage devrait prendre 1 à 5 secondes.

```
[5] m = Prophet()
    m.fit(df)
```

```
INFO:numexpr.utils:NumExpr defaulting to 2 threads.
INFO:fbprophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True to override this.
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
<fbprophet.forecaster.Prophet at 0x7f1ee4f6cc18>
```

Les prédictions sont ensuite effectuées sur une trame de données avec une colonne ds contenant les dates pour lesquelles une prédiction doit être faite. Vous pouvez obtenir une trame de données appropriée qui s'étend dans le futur sur un nombre de jours spécifié à l'aide de la méthode d'assistance Prophet.make_future_dataframe. Par défaut, il comprendra également les dates de l'historique, nous verrons donc également l'ajustement du modèle.

On fait une prédiction pour la semaine prochaine

```
future = m.make_future_dataframe(periods=7)
future.tail()
```

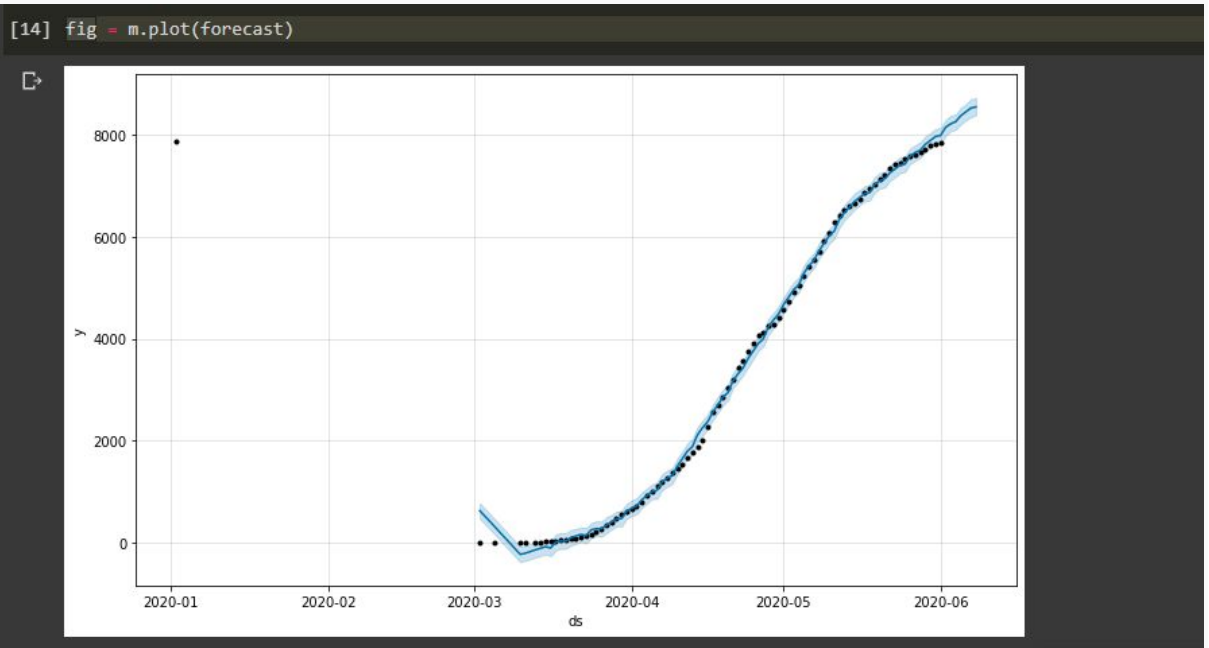
	ds
88	2020-06-04
89	2020-06-05
90	2020-06-06
91	2020-06-07
92	2020-06-08

La predict méthode attribuera à chaque ligne future une valeur prédite qu'elle nomme yhat. Si vous transmettez des dates historiques, cela fournira un ajustement dans l'échantillon. L' forecast objet ici est une nouvelle base de données qui comprend une colonne yhat avec la prévision, ainsi que des colonnes pour les composants et les intervalles d'incertitude.

```
[8] forecast = m.predict(future)
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

	ds	yhat	yhat_lower	yhat_upper
88	2020-06-04	8251.124859	8091.225437	8399.942694
89	2020-06-05	8366.405826	8210.949132	8524.003878
90	2020-06-06	8441.732669	8292.797029	8608.621025
91	2020-06-07	8518.386087	8348.511863	8685.555191
92	2020-06-08	8539.315564	8364.195680	8706.621729

Vous pouvez tracer la prévision en appelant la Prophet.plot méthode et en transmettant votre cadre de données de prévision.



Pourquoi le modèle Prophet :

- **Précis et rapide** : Prophet est utilisé dans de nombreuses applications sur Facebook pour produire des prévisions fiables pour la planification et l'établissement d'objectifs. Nous avons constaté qu'elle fonctionnait mieux que toute autre approche dans la majorité des cas. Nous installons des modèles dans Stan afin que vous obteniez des prévisions en quelques secondes.
- **Entièrement automatique** : Obtenez une prévision raisonnable sur des données désordonnées sans effort manuel. Prophet est robuste aux valeurs aberrantes, aux données manquantes et aux changements dramatiques dans votre série chronologique.
- **Prévisions réglables** : La procédure Prophet comprend de nombreuses possibilités pour les utilisateurs d'ajuster et d'ajuster les prévisions. Vous pouvez utiliser des paramètres interprétables par l'homme pour améliorer vos prévisions en ajoutant vos connaissances de domaine.
- **Disponible en R ou Python** : Nous avons implémenté la procédure Prophet en R et Python, mais ils partagent le même code Stan sous-jacent pour

l'ajustement. Utilisez la langue avec laquelle vous êtes à l'aise pour obtenir des prévisions.

Équation du prophète :

Prophet ajuste plusieurs fonctions linéaires et non linéaires du temps en tant que composantes. Dans sa forme la plus simple :

$$y(t) = g(t) + s(t) + h(t) + e(t)$$

où :

$g(t)$: modèles de tendance changements non périodiques (c.-à-d. croissance au fil du temps)

$s(t)$: la saisonnalité présente des changements périodiques (c.-à-d. hebdomadaire, mensuelle, annuelle)

$h(t)$: liens dans les effets des **vacances** (sur des horaires potentiellement irréguliers ≥ 1 jour (s))

$e(t)$: couvre les changements idiosyncratiques non pris en compte par le modèle

3 - Le Machine Learning en Covid-19 :

De nos jours, Covid-19 est l'une des régions essentielles de la recherche dans le monde et particulièrement au Maroc. Un assortiment de recherches a été effectué sur les informations recueillies dans les dossiers de police couvrant une partie restreinte de la propagation de ce virus dans les villes. Dans cette partie, nous avons utilisé des stratégies

de prédiction des cas et des simulations pour examiner les informations fournies par METLM dans lesquelles nous regroupons d'abord les informations sur Covid-19. Une procédure supplémentaire de prédiction en utilisant Prophet est également utilisée pour distinguer les futures cas dans lesquelles un incident peut survenir pour chaque groupe, et pour la simulation on utilise pygame pour avec comment le virus se propage.

Les résultats peuvent être utilisés pour mettre en place des tentatives d'aversion pour les mésaventures dans les territoires reconnus pour diverses classifications de virus afin de battre le nombre des affectées.

Conclusion :

Au bout du compte, le Machine Learning joue un rôle très importants en sécurité routière, d'après la fameuse techniques qui a changé le danger de risque de la propagation de virus, à la faveur de Prophet pour prédire des futures cas et de voir comment le virus se propage dans une simulation, à fin de connaître les principales causes de propagation pour les éviter.

Chapitre 2 : Analyse

Introduction :

Dans ce chapitre nous essayerons de donner une vue générale de l'application en spécifiant l'ensemble des fonctionnalités qu'elle doit apporter.

1 - Analyse de Système :

Présentation générale de l'application :

L'application qu'on va réaliser permet de prendre une grande masse de données de Covid-19 stockées dans notre environnement. Ces données seront traitées par un modèle pour obtenir les prédictions des cas.

L'application utilise le modèle Prophet pour analyser et générer les prédictions à partir des données de virus. Ces données sont stockées dans un environnement de travail.

Pour les simulations, on va les faire en utilisant pygame pour voir dans de figures comment le virus se propage selon les cas.

Analyse de l'application :

Le but principal de notre application est de connaître les causes de propagation de virus selon les simulations et de prévoir les cas (affectées, non affectées, rétablis...). Donc les notes les plus intéressants qu'elle doit contenir sont :

- l'application contient une analyse de données pour le virus en se basant sur des plots ...
- L'application va faire de prédictions à l'aide de modèle Prophet.
- Les prédictions vont générer des connaissances sur les cas de future pour aider le gouvernement de faire des précautions.
- Toutes les données utilisées seront stockées dans des différents fichiers CSV selon le type d'information.
- les simulations sont faites séparément aux ces dernières.

- Il y aura 3 simulations, une est simple dans la vie normale, l'autre dans la quarantaine, la dernière est pour la quarantaine en arrêtant les cas affectés s'ils sont détectés.

Conclusion :

En conclusion ce chapitre présente une analyse globale de l'application tout en spécifiant les besoins fonctionnels et les contraintes que notre travail doit satisfaire et respecter. Etant donné les besoins des utilisateurs de cette application. Nous avons aussi décrit le travail que nous avons réalisé pour concevoir notre application.

Chapitre 3 : l'environnement technique

Introduction :

La prédiction et la simulation de Covid-19 nécessite une interaction rapide entre le système et la dataset pour obtenir les résultats en le moins de temps possible.

Pour cela, il faut choisir des outils technique rapide et avec un temps d'exécution très performant.

- La justification de choix d'une application avec python.
- les différents outils techniques que nous avons utilisé dans L'application avec une petite explication pour chaque outils.

1- Le choix d'une application avec python :

Il y a de nombreux critères qui nous ont tirés à choisir une application avec python et ne pas une application avec d'autre language. Parmi ces critères on a :

Simple :

- **Une syntaxe archi-simple**

Pas besoin de se prendre la tête avec les parenthèses, accolades,

crochets, points-virgules, etc...

La syntaxe du code utilise l'indentation. Ça veut dire quoi ?

Par exemple si tu veux définir une condition « if » ou une boucle « while » ou même une fonction, pas besoin d'accolades. Juste une indentation. Ce qui est simple à mettre en place.

- **Les types sont dynamiques :**

Tu n'as pas à écrire le type de tes variables : nombres, strings, arrays, t'as pas besoin de le préciser. Il saura les identifier directement en fonction des opérations que tu fais avec.

- **Il ressemble à du langage parlé**

Multi fonction :

Déjà il faut savoir que Python fonctionne sur tous les OS, que tu sois sur Windows, Mac ou Linux, tu pourras l'utiliser.

Et surtout Python peut être utilisé dans des tas de domaines... même des domaines auxquels on s'attend moins comme le développement de jeux vidéo ou les applis de bureau.

Mais bon, ça ne fait pas quand même pas partie de ses utilisations premières. Python est principalement utilisé dans 3 domaines :

- **Développement Web**

Eh oui, Python est beaucoup utilisé dans le Web. Il y a même des frameworks très évolués et très populaires comme Django pour bien structurer ton travail en Python.

Alors une petite précision tout de même, pour ceux qui pensent pouvoir devenir développeur Full-Stack avec Python.

Aujourd'hui, il n'y a pas vraiment de Front-End avec Python.

Les navigateurs fonctionnent avec un moteur Javascript et pas un moteur Python.

Certains ont essayé d'utiliser Python même en Front-End comme le projet Brython: mais il y a beaucoup de problèmes de légèreté, rapidité, etc... qui font que ce n'est pas vraiment un succès.

Tu devras toujours apprendre le Javascript si tu veux maîtriser le développement Front-End.

- **Data Science**



C'est le langage le plus populaire dans le Big Data, qui est lui-même un domaine en pleine expansion. Si tu veux te lancer dans les statistiques et analyser des données et même pouvoir visualiser ces données, Python est le maître. Je vais te donner un exemple : Imagine-toi que tu travailles pour un centre commercial et que ton but, c'est de vendre le plus de machines à café car t'en as trop en stock et que le nouveau modèle va bientôt sortir.

Et disons que tu veux mettre en place une opération sur les machines à café en offrant un petit cadeau pour chaque machine à café achetée.

Alors grâce à Python, tu pourras analyser les données des achats des années précédentes pour savoir quel est le produit que les gens achetaient en général avec leur machine à café. Et tu pourras leur

offrir ce produit pour l'achat d'une machine à café.

Sinon, il y a un autre domaine relié à l'intelligence artificielle qui cartonne de plus en plus, c'est ce qu'on appelle le machine learning. Je ne vais pas te faire un cours là-dessus, mais en gros c'est lorsqu'on essaie d'apprendre aux machines à réfléchir en analysant des tas de données.

Et là aussi Python mène la course ! Donc si c'est ce que tu veux faire alors mets-toi à Python !

- **Script**

Pour t'expliquer ce que j'entends par script, le mieux c'est que je te donne directement des exemples.

Premier exemple tout bête. Imagine que t'as plein de fichiers sur ton ordi et que tu cherches une info précise mais tu ne te rappelles pas dans lequel de ces fichiers c'est écrit. Alors tu pourras écrire un script qui passe sur chacun de tes fichiers en cherchant le mot clé que tu veux et en quelques secondes, tu trouveras dans quel fichier ton info se cache.

Un autre exemple, imagine que tu veux afficher dans ton site des infos sur les indices boursiers et que chaque jour ces infos doivent se mettre à jour avec les valeurs réelles du marché, alors tu pourras utiliser Python pour faire ce qu'on appelle du Web Scrapping.

C'est-à-dire qu'il ira récupérer ces données directement depuis un site web et les enregistrera dans ta base de données pour que tu puisses y avoir accès.

2- les outils techniques :

pygame :



Pygame est une bibliothèque libre multiplate-forme qui facilite le développement de jeux vidéo temps réel avec le langage de programmation Python.

Elle est distribuée selon les termes de la licence GNU LGPL.

Google colab :



Google Colab ou Colaboratory est un service cloud, offert par Google (gratuit), basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud. Sans donc avoir besoin d'installer quoi que ce soit sur notre ordinateur à l'exception d'un navigateur. Cool, n'est ce pas ? Avant de présenter ce magnifique service, nous rappellerons ce qu'est un Jupyter Notebook.



Jupyter Notebook est une application Web Open Source permettant de créer et de partager des documents contenant du code (exécutable directement dans le document), des équations, des images et du texte. Avec cette application il est possible de faire du traitement de données, de la modélisation statistique, de la visualisation de données, du Machine Learning, etc. Elle est disponible par défaut dans la distribution [Anaconda](#) (suivre [ce lien](#) pour savoir comment l'installer).

VScode :



Pour la partie de simulation par Pygame on peut pas utiliser colab donc on utilise vscode.

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS2.

Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré. Les utilisateurs peuvent

modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires.

Conclusion :

Le choix des outils technique est généralement dépend aux besoins de client. Dans ce chapitre nous avons choisi les outils les plus adapter au notre projet, ainsi on a bien expliqué ces outils. Dans les chapitres suivant on va utiliser ces techniques pour réaliser notre application.

Chapitre 4 : Implémentation

Introduction :

Dans ce chapitre nous allons parler de l'architecture générale de l'application : les diverses fonctions. Nous allons commencer par une conception de l'application, Puis nous allons présenter leur espace à l'aide de capture écran.

Application :

Affichage de données :

Les cas enregistrés par régions :

Region / الجهة	Total Cases / إجمالي الحالات	Active Cases / الحالات النشطة	Total Deaths / إجمالي الوفيات	Total Recovered / إجمالي المعافين
0 Tétouan-Al Hoceïma	1109	777	29	305
1 Oriental	187	34	8	145
2 Fès-Meknès	1001	551	26	424
3 Rabat-Salé-Kénitra	705	362	11	332
4 Béni Mellal-Khénifra	120	59	7	54
5 Casablanca-Settat	2665	1963	46	656
6 Marrakech-Safi	1349	438	50	861
7 Drâa-Tafilalet	588	119	4	465
8 Souss-Massa	89	45	7	37
9 Guelmim-Oued Noun	45	26	0	29
10 Laayoune-Sakia El Hamra	5	1	0	4
11 Eddakhla-Oued Eddahab	5	3	0	2

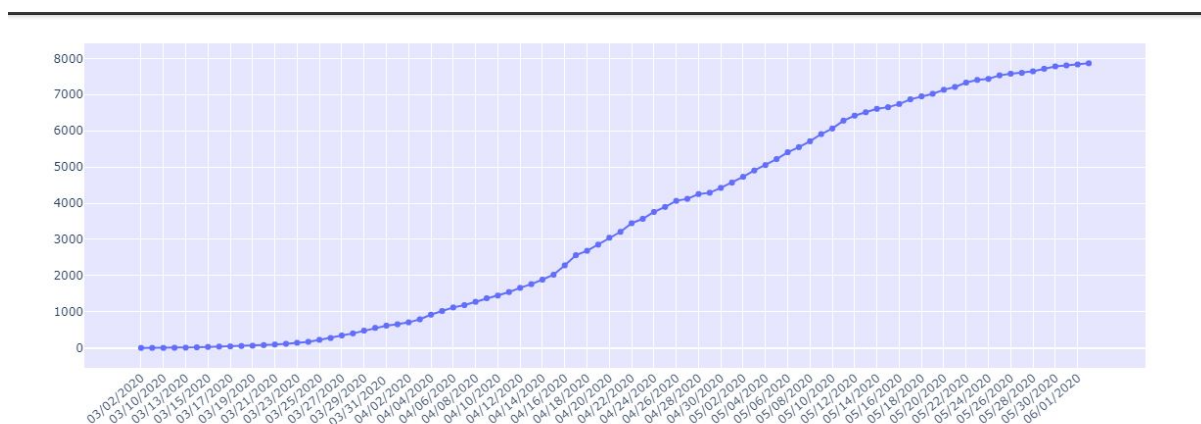
Les cas enregistrés par jour :

Dates / التواريخ	Cases / الحالات	Recovered / تعافى	Deaths / الوفيات
0 03/02/2020	1	0	0
1 03/05/2020	2	0	0
2 03/10/2020	2	0	1
3 03/11/2020	6	0	1
4 03/13/2020	8	1	1

Les cas enregistrés par bien :

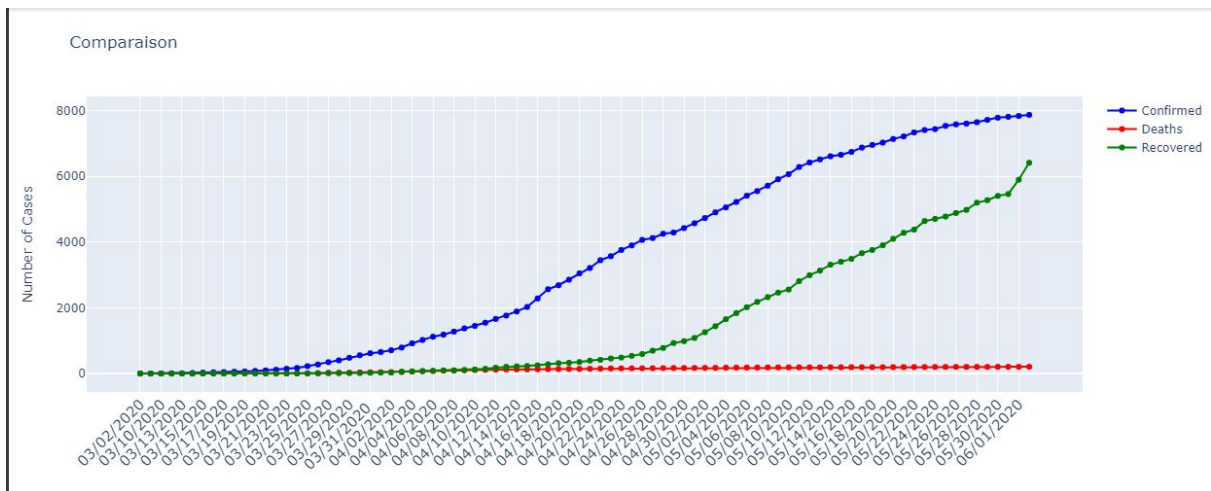
City / المدينة	Region / المنطقة	Active Cases / الحالات النشطة	Total Deaths / إجمالي الوفيات	Total Recovered / إجمالي المعافين
0 Aïn Harrouda	Casablanca-Settat	0	0	0
1 Ben Yakhlef	Casablanca-Settat	0	0	0
2 Bouskoura	Casablanca-Settat	0	0	0
3 Casablanca	Casablanca-Settat	130	1	1
4 Médiouna	Casablanca-Settat	0	0	0

visualisation des cas infectés chaque jour par coronavirus :

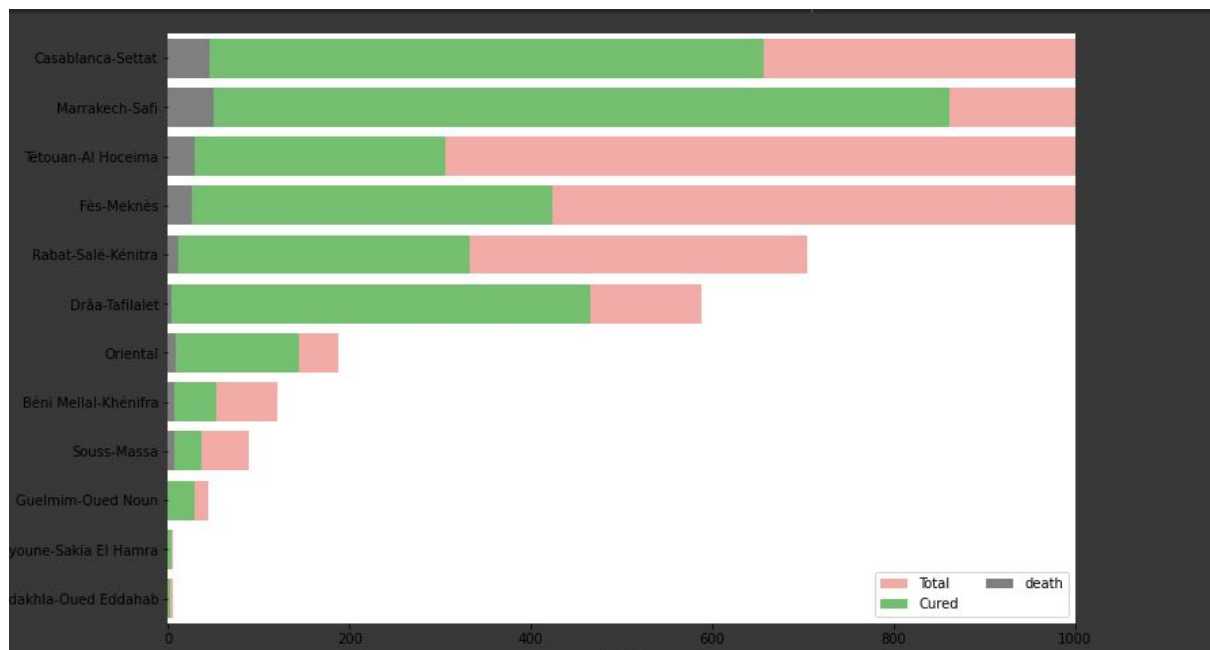


Comparaison entre les cas confirmés, les décès et les rétablis :

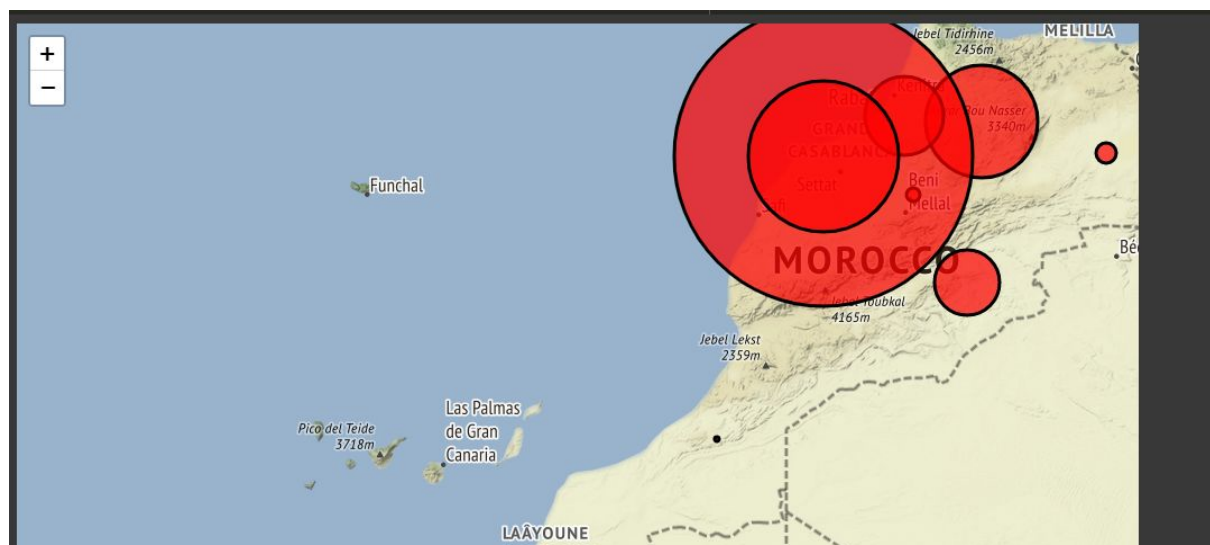
Comparaison des cas total :



Comparaison des cas total :



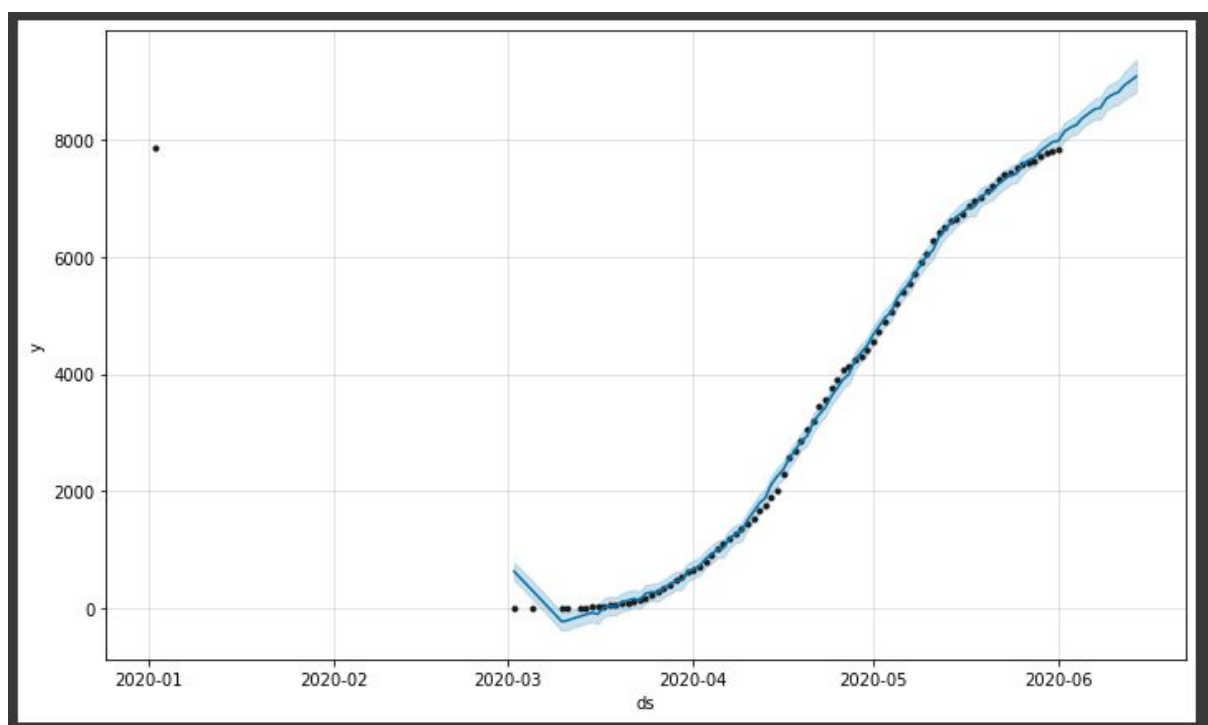
Visualisation de la répartition géographiquement :



Prédiction de cas confirmés de coronavirus au Maroc :

	ds	yhat	yhat_lower	yhat_upper
93	2020-06-10	8767.362445	8568.146742	8967.302203
94	2020-06-11	8810.533911	8601.419948	9016.792452
95	2020-06-12	8925.814877	8690.922678	9161.377201
96	2020-06-13	9001.141720	8750.152223	9261.471349
97	2020-06-14	9077.795138	8811.952781	9369.257359

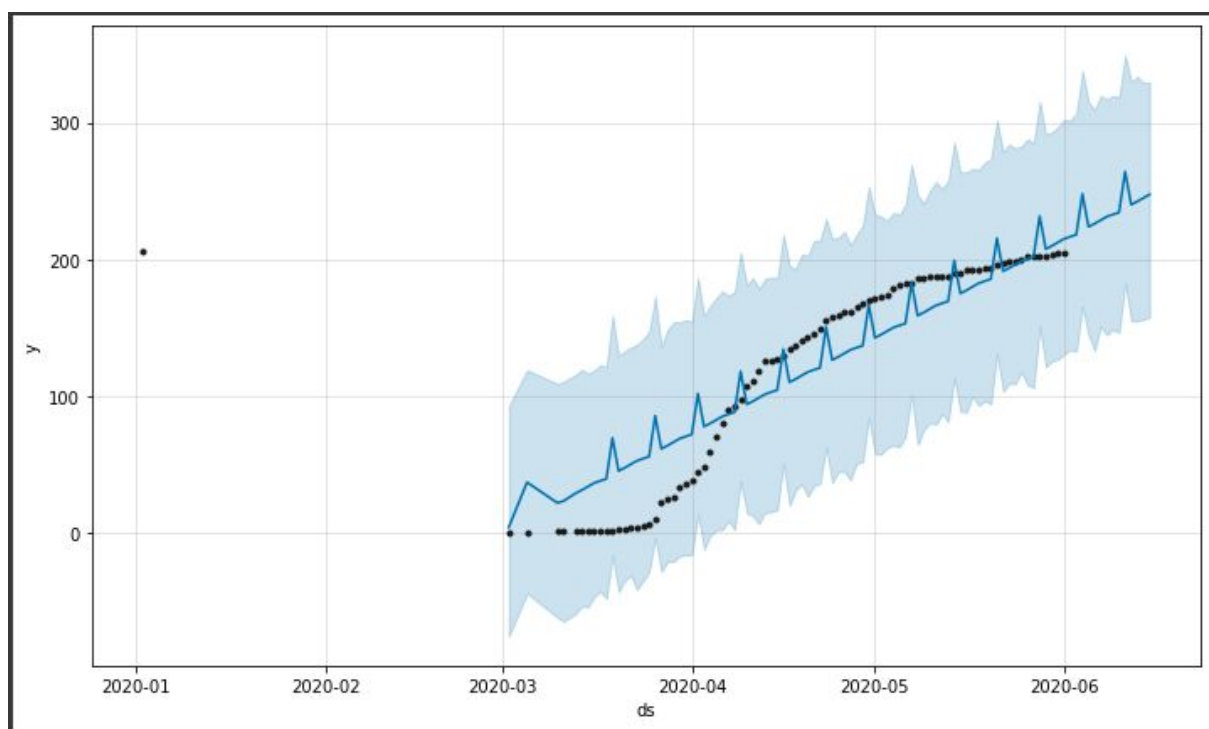
visualisation des cas confirmés :



Prédiction des décès au Maroc à l'aide de Prophet :

	ds	yhat	yhat_lower	yhat_upper
94	2020-06-11	264.062051	182.423820	349.132278
95	2020-06-12	239.991574	154.778115	329.771319
96	2020-06-13	242.240309	154.948359	333.556559
97	2020-06-14	244.822320	156.039176	329.001769
98	2020-06-15	247.386520	157.969995	329.034450

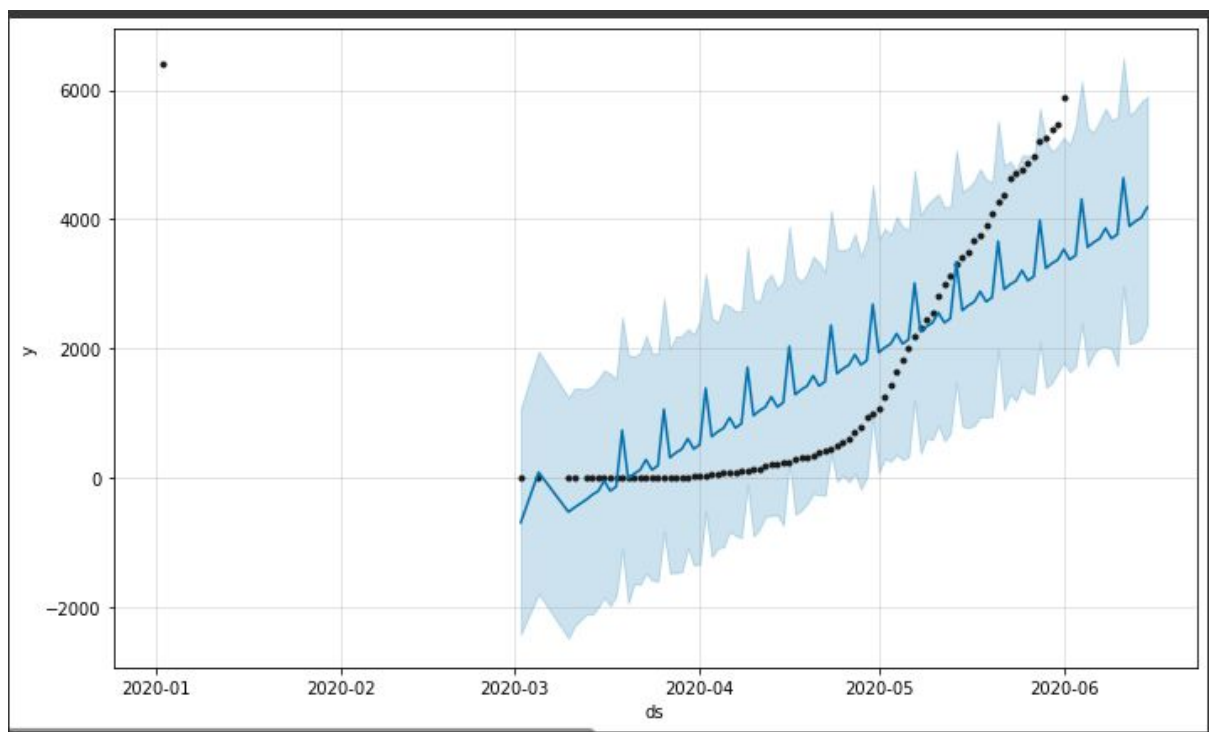
visualisation de la prédiction des décès :



Prédiction des rétablis :

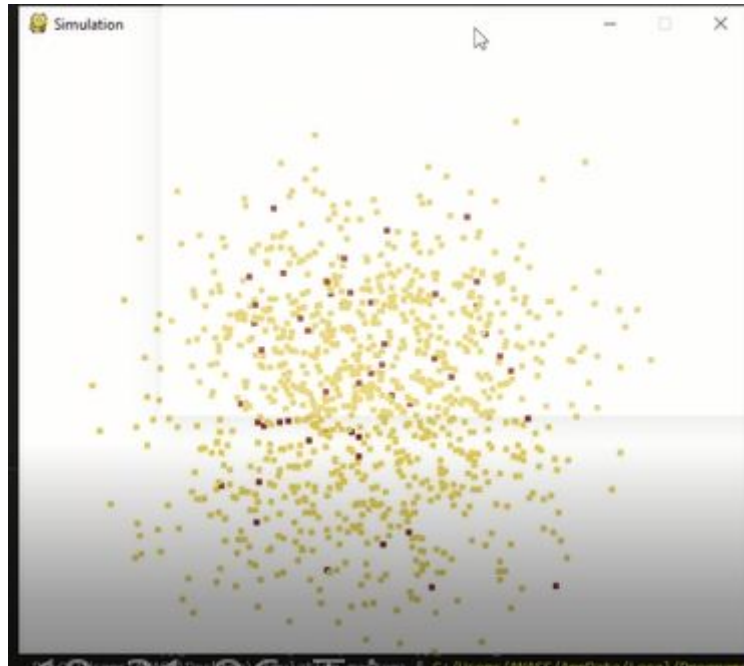
	ds	yhat	yhat_lower	yhat_upper
94	2020-06-11	4638.827042	2956.768261	6492.070060
95	2020-06-12	3892.491335	2077.807046	5610.407176
96	2020-06-13	3968.043743	2100.777592	5691.190032
97	2020-06-14	4025.009089	2150.906622	5821.556438
98	2020-06-15	4185.707530	2370.727024	5898.463044

visualisation des prédictions des rétablis :



Pour la simulation simple :

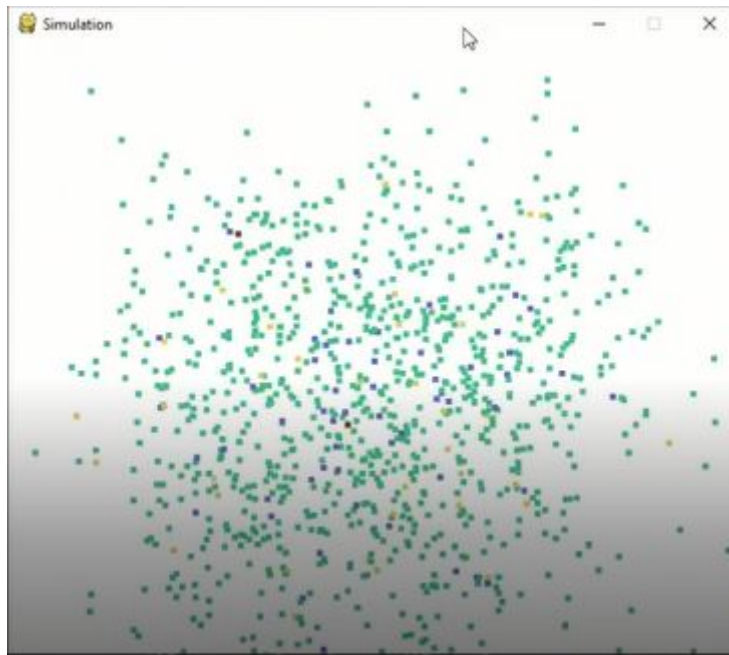
Au début :



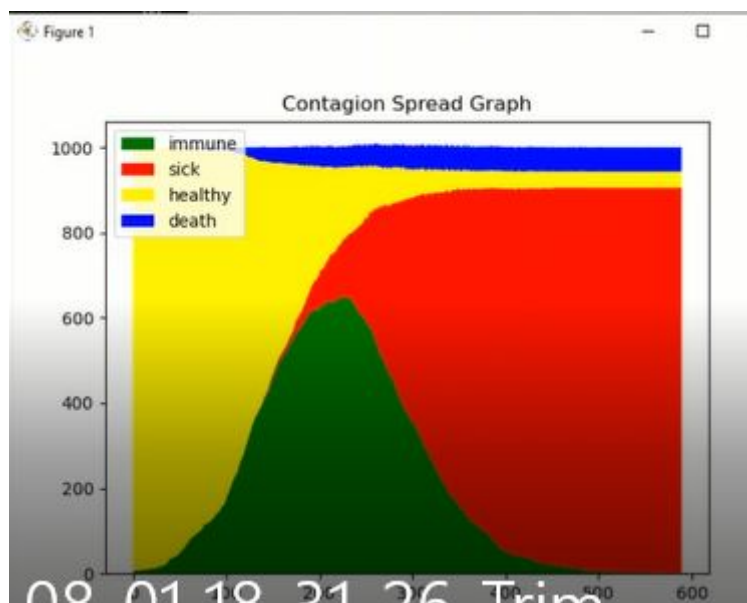
Au milieu :



À la fin :



Variation des cas :



Pour le lockdown :

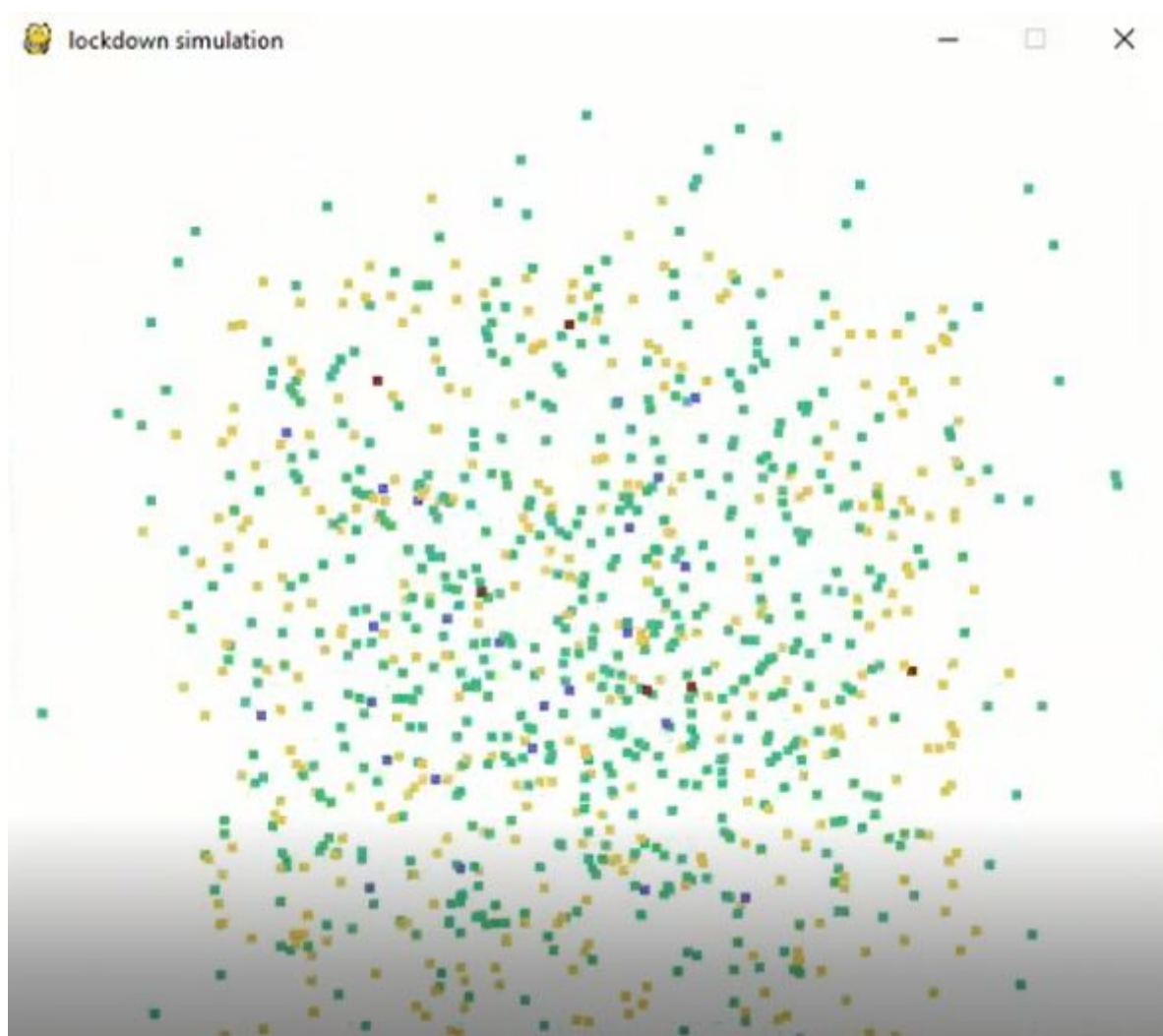
Au début :



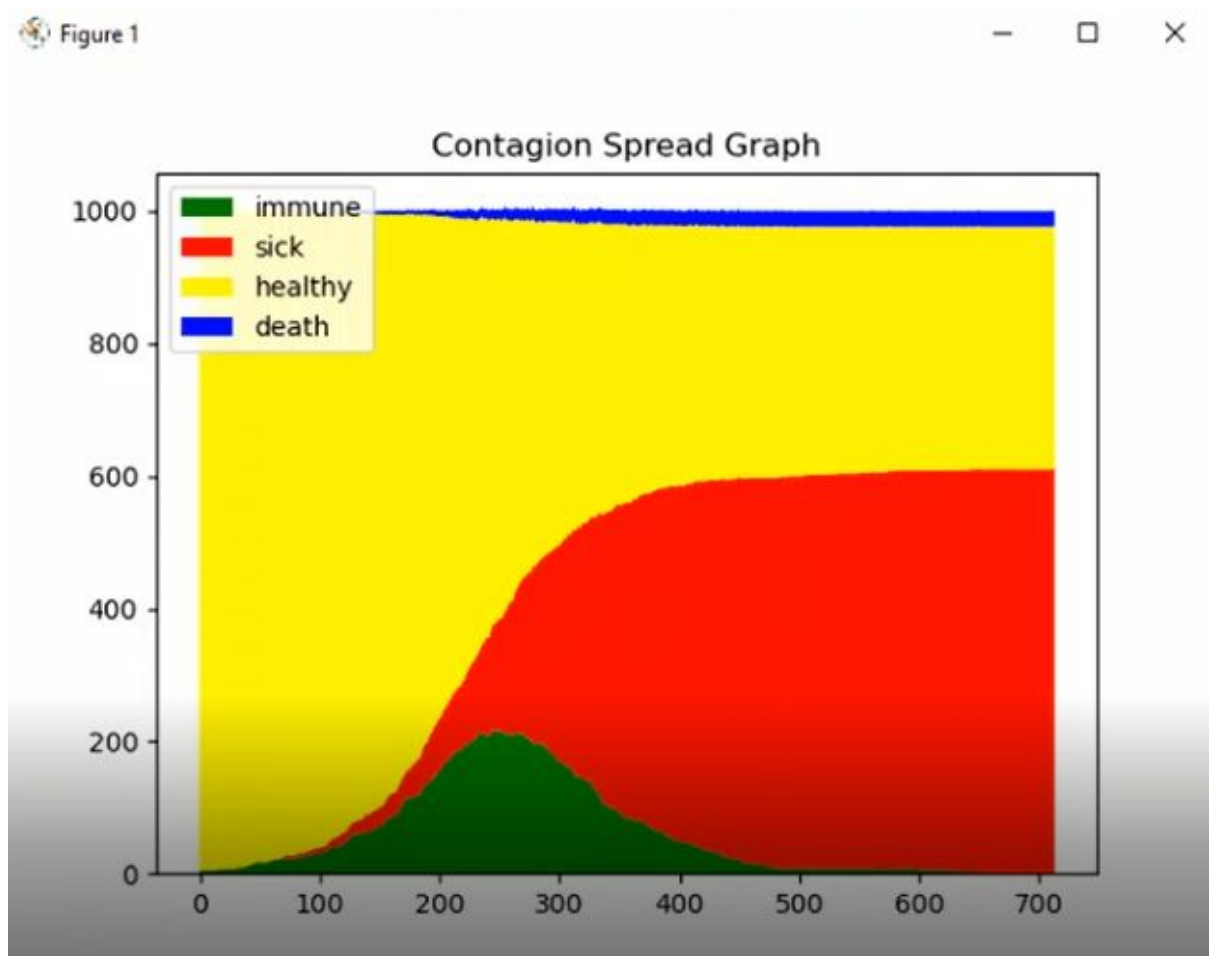
Au milieu :



À la fin :

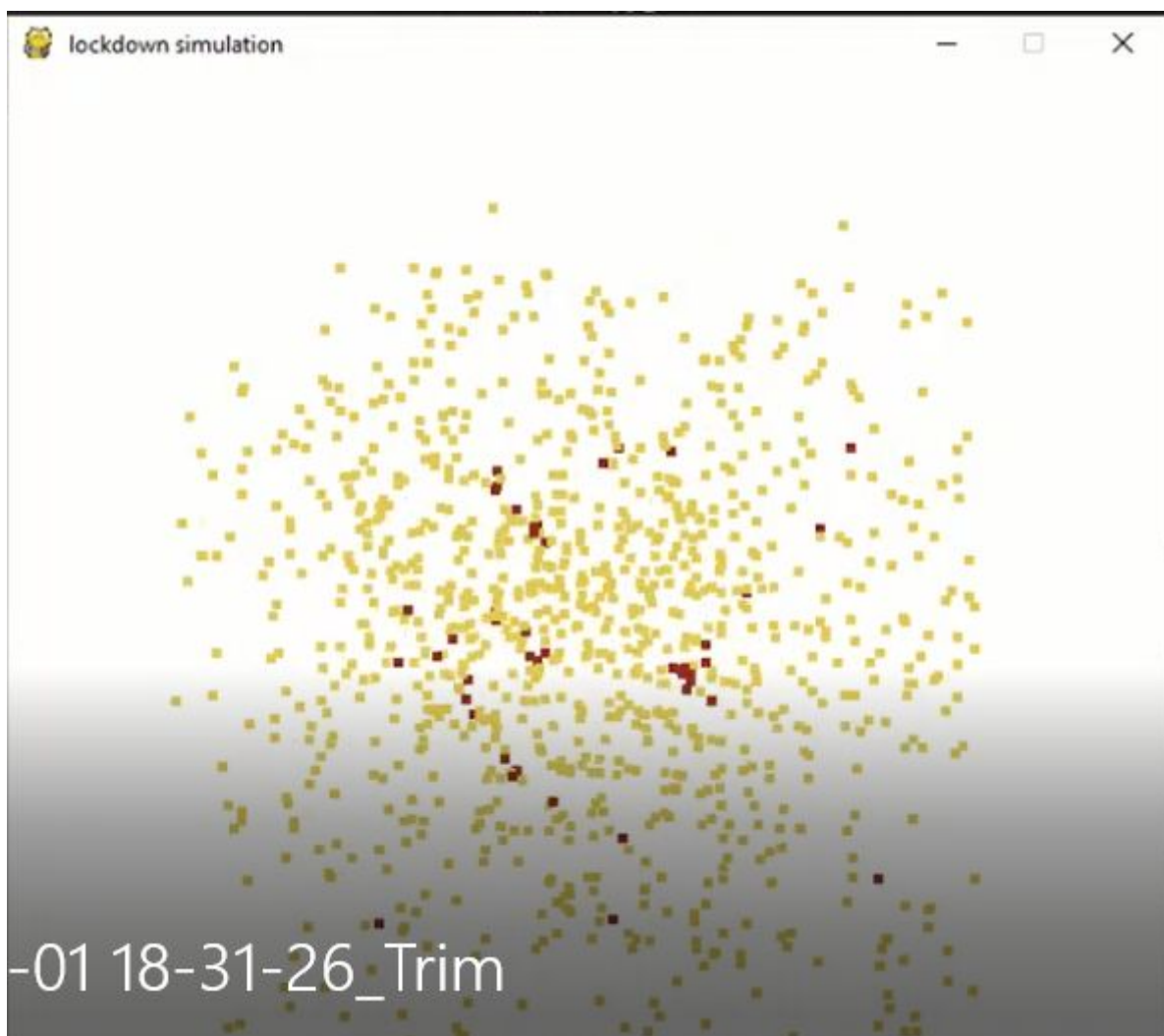


Variation des cas :

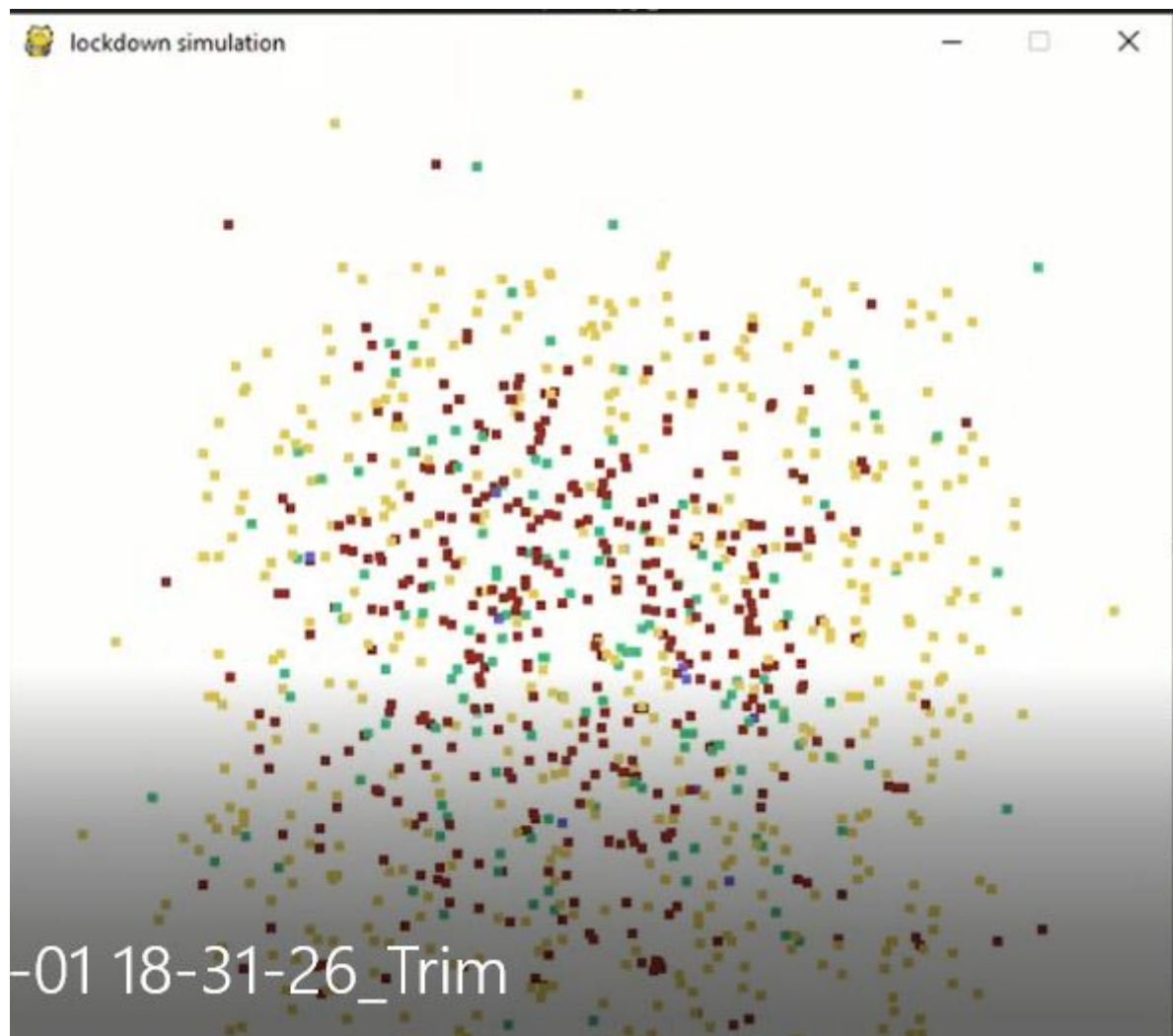


Pour le lockdown en arrêtant les cas affectés s'ils sont détectés :

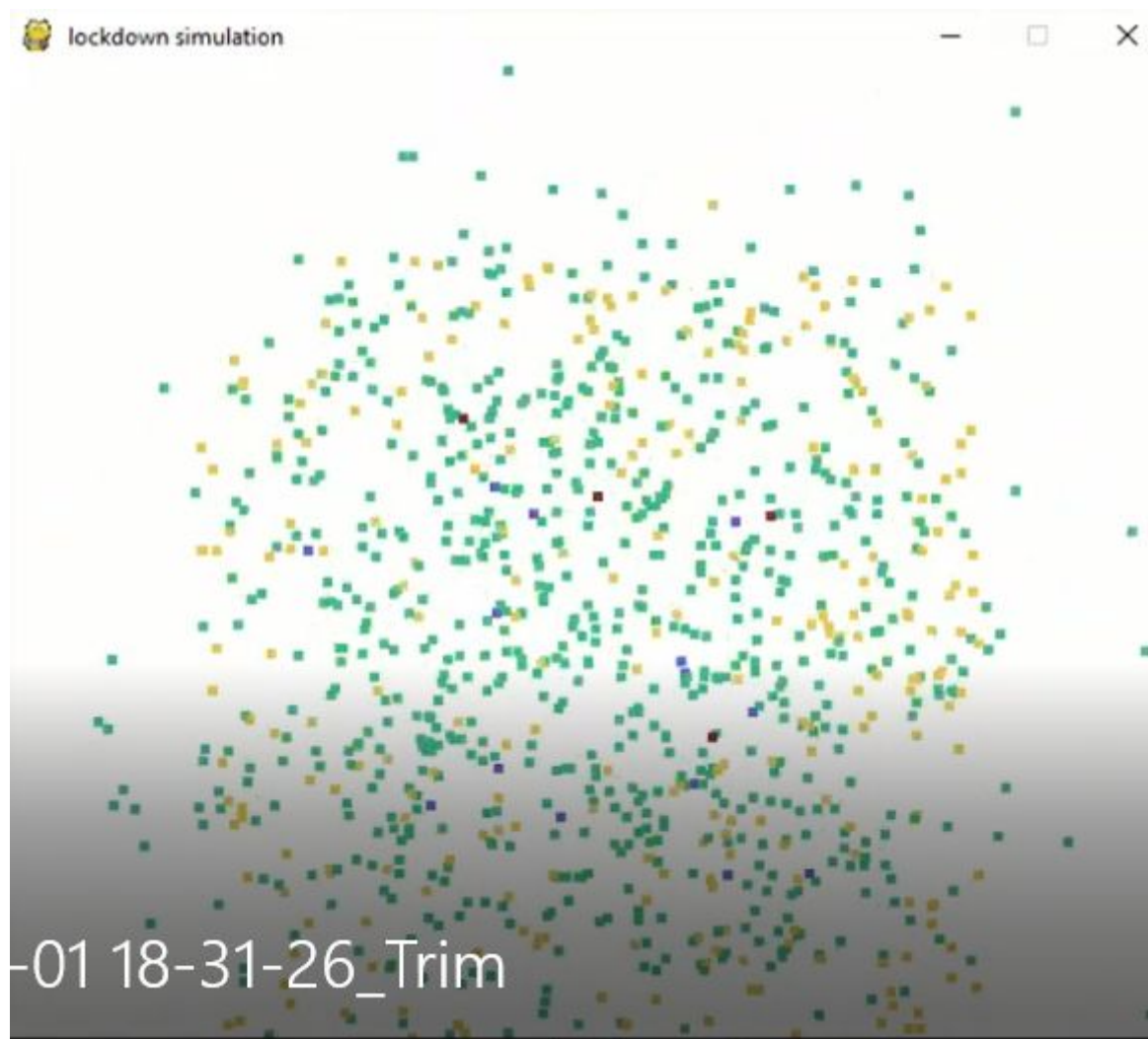
Au début :



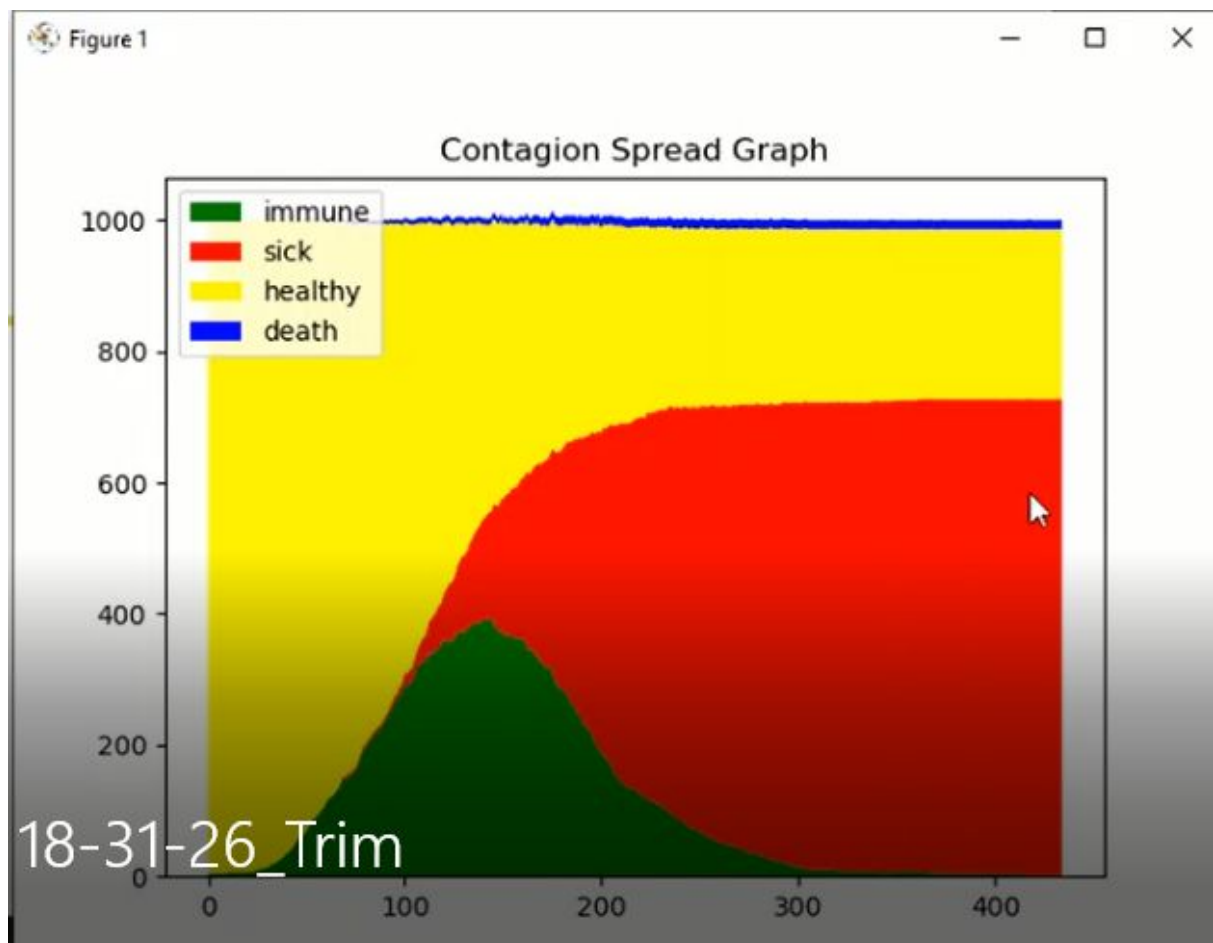
Au milieu :



À la fin :



Variation des cas :



Conclusion :

Dans ce chapitre nous avons fait une architecture générale de l'application. De plus nous avons donné quelque capture d'écran de notre application web.

Conclusion générale :

Notre projet consiste à mettre en œuvre une solution pour la prédiction et la simulation du virus Covid-19 à l'aide du Machine Learning. Cette solution permettra d'éviter les grands problèmes qui se déclenchent au niveau de la propagation de ce virus. En effet ce problème est la pandémie Covid-19 qui menace la vie d'un grand nombre de personnes. Ainsi, le projet va bien nous aider à prévoir les futurs cas et de voir la simulation afin de connaître le risque qui peut nous arriver.

Nous avons tiré un grand bénéfice de ce projet, aussi bien au niveau personnel qu'au niveau professionnel. Nous avons aussi pu raffiner nos capacités d'abstraction en termes d'analyse, et de développer l'esprit de recherche et améliorer notre méthodologie de travail. En outre, ce projet était l'occasion pour nous afin de plonger dans l'univers de domaine du Machine Learning dont rêverait tout passionné de nouvelles technologies.

Comme perspective, nous souhaitons pour les prochains projets d'élargir notre travail sur le Deep Learning.

Bibliographie

Sites web :

- <https://www.coursera.org/learn/machine-learning>
- https://facebook.github.io/prophet/docs/quick_start.html
- <https://matplotlib.org/tutorials/introductory/pyplot.html>