BOUGUERRA Mhamed

2MP-SD

Étude de projet Machine Learning

I) Analyse de besoins

1) Définition du programme

Note projet et définir comme un système d’apprentissage (Analyse et prédiction) qui permet de découvrir le nombre des personnes qui se font vacciner avec une data bien déterminer aussi prédire le nombre de personnes qui se font vacciner à cette date ainsi qu'à un autre moment.

2) Exigences

* Le logiciel doit être capable de s'entraîner avec des données pour atteindre le taux de prédiction le plus élevé.
* Le logiciel doit être recyclé à l'aide des données.
* Le logiciel doit avoir une interface graphique facile à comprendre pour l'utilisateur final.
* La précision doit atteindre au moins 70%+.

3) Solution proposée

Le but de l'application (Tunisiens Vac) est découvrir le nombre de personnes qui se font vacciner à la date que nous avons choisie (lorsque nous sommes à la date et le nombre de personnes inscrites, cela nous prédire le nombre de personnes qui se font vacciner à cette date, même la date est dans le futur).

Pour ce faire, l’application doit permettre les fonctionnalités suivantes :

* Charger les données.
* Démarrez l'analyse.
* Train le model.

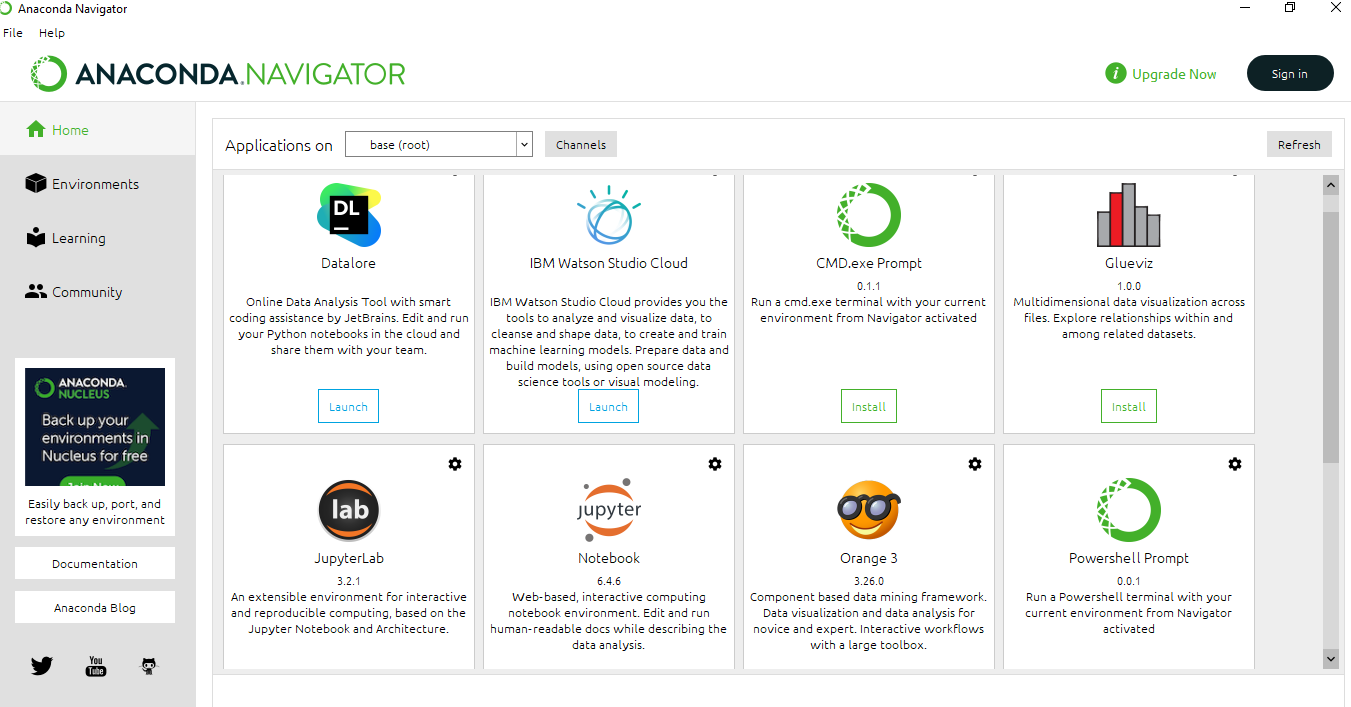
II) Méthode de conception

1. Environnement logiciel

Dans cette phase, nous avons montré les différents outils utilisé tout au long de ce projet pour la mise en place de notre application.

* 1. Environnement de développements

Dans ce cadre-là on va utiliser « Anaconda » comme une plateforme de distribution des langages de programmation Python et R pour le calcul scientifique (science des données, applications d'apprentissage automatique, traitement de données à grande échelle, analyse prédictive, etc.), qui vise à simplifier la gestion et le déploiement des packages.



1. Base de Données

Notre dataset est fondamentalement un fichier du type csv. La source de ce dernier est la plateforme sanitaire EVAX.

1. Environnement matériel

Nous mentionnons les caractéristiques de notre ordinateur sur lesquelles nous avons développé l'application. Cette dernière a été développée sur une machine avec les caractéristiques suivantes :

- Marque : ACER Aspire F5-573G

- Processeur : Intel(R) Coré (TM) i7-7500U CPU @ 2.70GHz 2.90 GHz

- Mémoire RAM : 12.00 Gb.

- Système d’exploitation : Windows 10 Professionnel.

III) Simulation

1. Description de l’application
   1. Présentation des interfaces de l’application
2. Interface de Bienvenue

Dans notre première interface nous avons une interface de bienvenue contient le nom de projet avec le name de réalisateur et le choix pour entrer ou bien quitté l’application

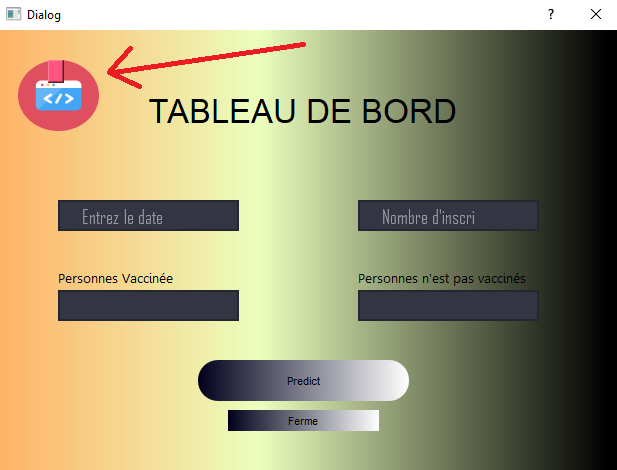


1. Interface du Tableau de bord

Dans cette interface nous permettent de faire des prédictions sur notre dataset. On va utiliser l’algorithme de modèle d'apprentissage automatique supervisé : Linear Régression.

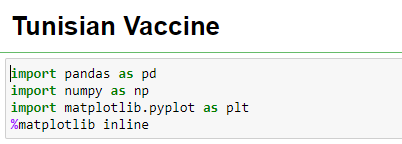


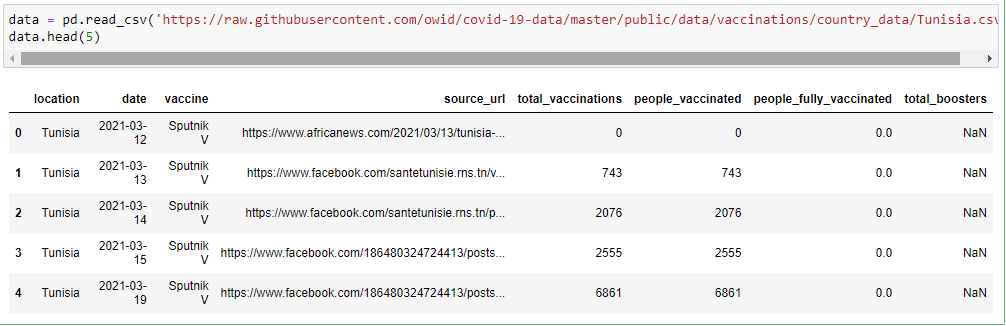
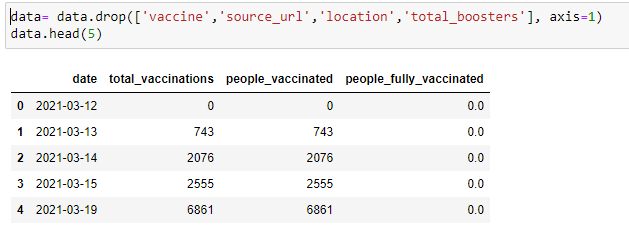
1. Interface du lien Source code

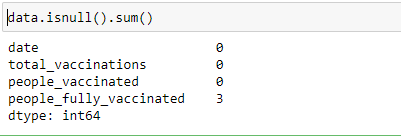
Avec cette button notre interface en va nous transfert dans le lien de code source de notre application.

Captures d’écran du code source

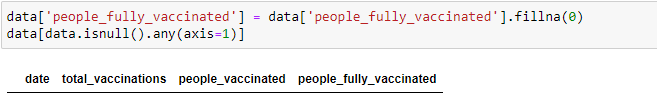
* Importer des bibliothèques qui on va travailler avec

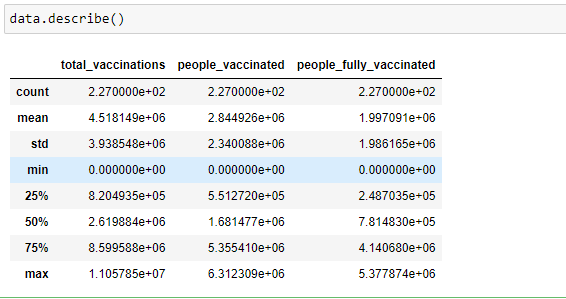


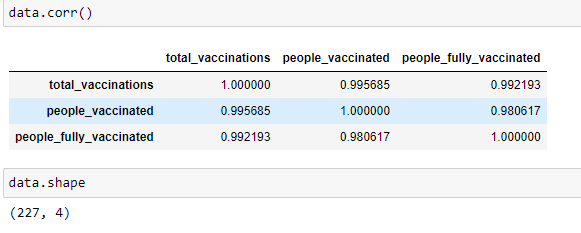
* Importer de la base de données et afficher les 5 premières lignes
* Supprimer les colonnes que nous n'en avons pas besoin et afficher la résultat
* Afficher les données disparus (NULL)



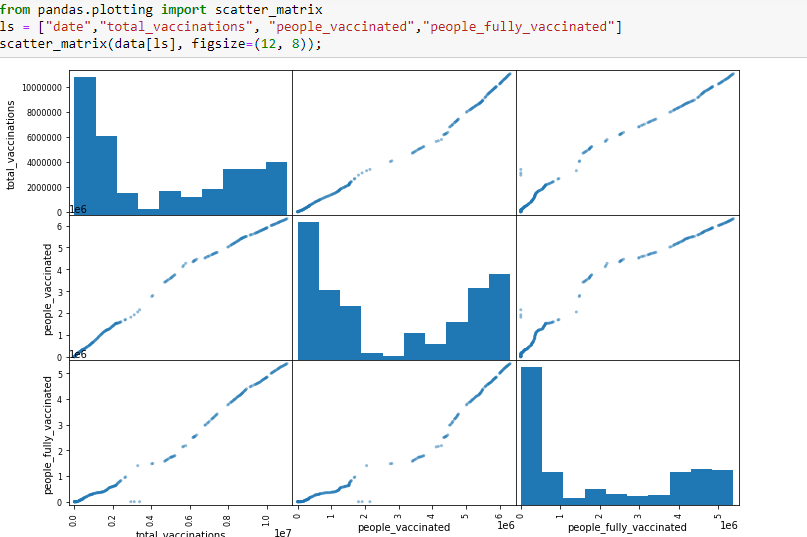
* Replacer tous les cas nuls avec 0 pour éviter les problèmes dans la partie prédiction



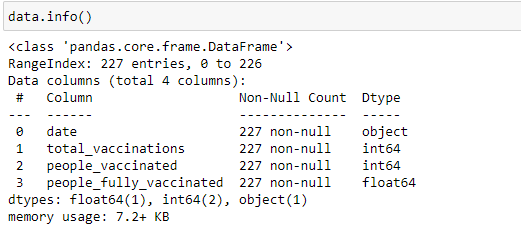
* Afficher la description de notre dataset 
* Afficher la corr de notre dataset
* Afficher le nombre du colonne et du ligne de notre dataset



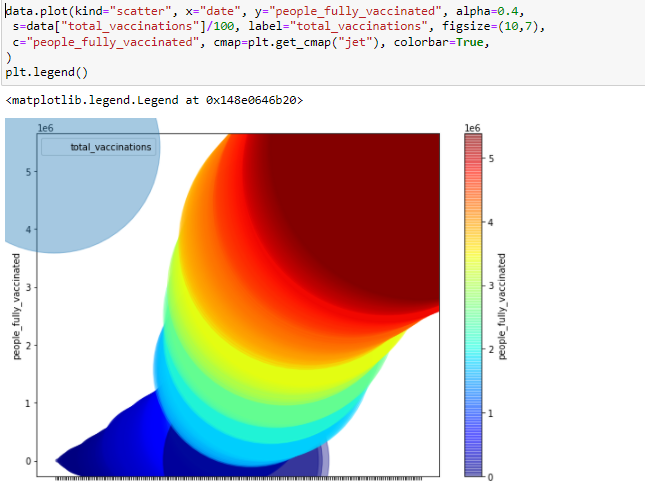
* Importer bibliothèque scatter\_matrix et représenter la distribution des données

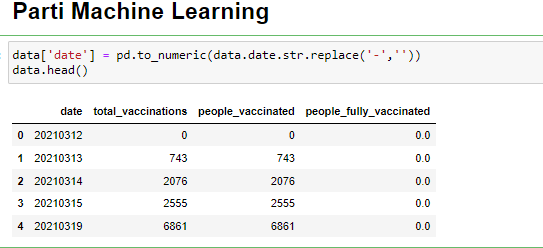


* Afficher les informations et les types des colonnes



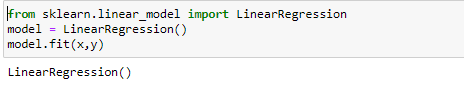
* Afficher une courbe du nombre des personnes vaccinés faible jusqu'au devient plus nombreux



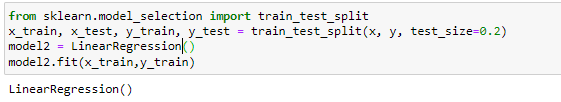
* Changer le type de date en numérique pour éviter les problèmes de train de model 
* Diviser notre dataset en deux parties x et y



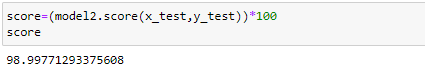
* Importer la bibliothèque sklearn et fit le dataset avec le model LinearRegression

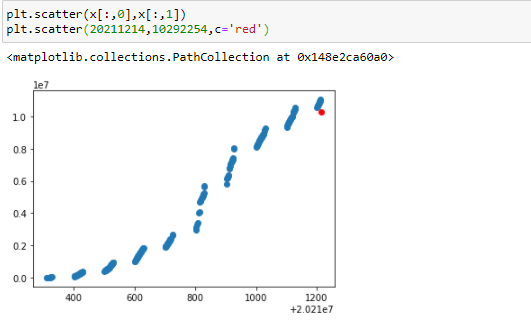
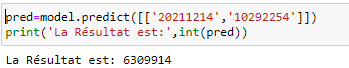


* Importer la bibliothèque train\_test\_split et fit un deuxième model de test avec le model LinearRegression



* Afficher le score du notre test dataset qui est égale a 98,99%



* Pointer dans le modèle scatter un point que nous voulons savoir leur prédiction
* Faire la prédiction d’une random saisir (dans la future)
* Parti interface qui été créé avec Qt\_Designer

