

Rapport de Stage Ingénieur

Effectué par

Firas Mansour

Classe : 4SE1

Réalisé à

Carthage solutions

De 8/2/2021 au 10/3/2021

Encadrant organisme d'accueil :

Oussama SOUAF

Année universitaire 2020/2021

esprit

18, rue de l'Usine - ZI Aéroport
Charguia II - 2035 Ariana
Tél. : +216 71 941 541 (LG)
Fax. : +216 71 941 889
e-mail : contact@esprit.ens.tn
www.esprit.ens.tn



Remerciements

Je tiens à exprimer mes remerciements aux personnes qui ont contribué à la réalisation de ce travail et plus particulièrement à :

M. SOUAF Oussama, mon encadreur à Carthage solutions pour m'avoir aimablement dirigé dans la réalisation de ce projet, pour leur disponibilité et leurs précieux conseils qui m'ont permis d'approfondir mon travail mais aussi pour la motivation qu'il m'a apportée tout au long de cette étude. Toute l'équipe de Carthage solutions pour son accueil et son esprit d'équipe.

Résumé

Ce document représente le compte-rendu de mon stage d'été, effectué au sein de la société Carthage Solution. Ce rapport de stage s'articule autour de trois chapitres :

Le premier chapitre comporte une brève présentation de l'organisme d'accueil. Il aborde la méthodologie appliquée pour assurer le bon déroulement du travail. Le reste est consacré au cadre du stage, les chronogrammes des tâches menées à ce dernier.

Table des matières

Remerciements.....	2
Résumé	3
Table des matières	4
Table des figures	6
Introduction générale	7
Chapitre 1 : Cadre générale du projet	8
1.1 Introduction	8
1.2 Présentation de l'organisme d'accueil	8
1.3 Présentation du projet	9
1.3.1 Cadre du projet.....	9
1.3.2 Contexte, problématique et solution propose	9
1.4 Conclusion	10
Chapitre 2 : Analyse statistique et descriptive du modèle	11
1.5 Introduction.....	11
1.6 Analyse statistique	11
1.7 Définitions	11
1.8 Distribution normale.....	13
1.9 Mesure de variabilité	13
1.10 Variance et écart type	14
1.11 Description détaillée du modèle	15
1.12 Lecture de données et importation de bibliothèques	15
1.13 Analyse exploratoire des données	18
1.14 Implémentation de model et "Feature selection"	21
1.15 Feature selection	21
1.16 Création de classificateur	22
1.17 Prédiction et Validation croisée de K-fold	22
1.18 Réglage des hyper paramètres et évaluation du modèle.....	23
1.19 Evaluation du modèle.....	24

1.20	Finalisons notre modèle avec Job lib	26
Chapitre 3 : Réalisation de l'application web		27
1.21	Présentation des outils.....	27
1.22	Django.....	27
1.23	Visual Studio Code.....	27
1.24	GitHub.....	28
1.25	React_Js	28
1.26	Création d'une application web	29
1.27	Conception	29
1.28	Déploiement de l'application sur Heroku	29
1.29	Réalisation	30
Conclusion générale		32
Référence		33

Table des figures

Figure 1: Statistique descriptive des données	12
Figure 2 :Informations générale sur les données	12
Figure 3 Distribution normale	13
Figure 4:Assemblage des fichiers.....	15
Figure 5:Dimensions des données	16
Figure 6:Types de données	16
Figure 7: Matrice de corrélation.....	17
Figure 8 Visualisation des données manquantes.....	18
Figure 9:supprimer les colonnes indésirable	18
Figure 10:Remplacer les colonnes supprimées	19
Figure 11:Gérer les données manquantes	19
Figure 12:Data preprocessing	20
Figure 13:Diviser la dataset	21
Figure 14:Test CHI2.....	21
Figure 15:Création d'un classificateur	22
Figure 16:Matrice de confusion	23
Figure 17:GridsearchCV.....	24
Figure 18:Courbe ROC	25
Figure 19:Enregistrement de modèle	26
Figure 20: Django.....	27
Figure 21:Visual Studio Code.....	27
Figure 22:Github	28
Figure 23: React _Js.....	28
Figure 24:heroku.....	29
Figure 25: Notre application Web.....	31

Introduction générale

De nos jours, la scène technologique connaît une évolution gigantesque en raison des demandes croissantes vers l'intelligence artificielle ,qui devrait occuper le marché mondial dans les prochaines années.

Comme nous le savons ,la situation sanitaire actuelle en Tunisie souffre de la troisième vague de la pandémie de SARS-cov2 .Etant donné la croissance exponentielle des infectés et le manque de vaccins ,nous cherchons une solution qui nous permette de réduire le taux d'infection en fournissant une plateforme virtuelle accessible à tous les Tunisiens qui nous permette de prédire si une personne est affectée par le virus .Pour cela, nous voulons développer un modèle de classification précis qui sera le cœur d'une application web.

La suite de ce document sera principalement structurée en trois chapitres qui exposent en détail l'approche et l'évolution de mon projet de développement .Dans le premier chapitre, je présente d'abord l'environnement du stage et l'organisation ainsi que la structure ,les objectifs et la méthodologie utilisés pour la réalisation de mon application .Le deuxième chapitre considère une analyse descriptive du modèle et le dernier chapitre sera consacré à la conception et réalisation du site web .

Les points majeurs soulevés, ainsi que les principaux résultats obtenus au cours de ce projet seront enfin rappelés en conclusion, et je termine en citant les améliorations possibles de mon travail et ses perspectives.

Chapitre 1 : Cadre générale du projet

1.1 Introduction

Dans ce chapitre, je présente l'organisation accueillante, le problème, la solution proposée et les objectifs. Dans une première partie, je présenterai l'entreprise Carthage Solution. Ensuite, je écrirai le cadre de mon projet. puis je vais faire une étude de l'existant pour identifier leurs lacunes et proposer l'enrichissement qu'apporte ma solution.

1.2 présentation de l'organisme d'accueil

Historiquement, Carthage solutions est née de l'expérience du département web de Carthage Publicité, une expérience de 6 ans dans le développement des sites et application web et mobile.

Carthage Publicité, l'Agence mère, est une agence de communication et publicité, référence dans le domaine de la création graphique, l'impression et les objets et supports publicitaires. Elle est aujourd'hui leader dans le service de la société civile, elle a réussi à mener de gros projet vers le succès, on cite parmi ces projets et à titre d'exemple le FSM dans ses deux session 2013 et 2015.

Carthage Solutions est une agence web implantée en Tunisie, Elle adopte les technologies de l'information et de la communication pour tirer et anticiper le meilleur parti des nouvelles tendances du monde de web et de l'informatique. Dotée d'une solide expérience et d'une équipe compétente aux profils variés, Carthage Solutions propose à sa clientèle un rapport d'accompagnement pour une réalisation réussie et bien enrichie de leurs projets.

Sa mission est de vous aider à être un modèle d'entreprise moderne, bien connectée et accessible, en évolution permanente et bénéficiant de toute la veille technologique pour une meilleure stratégie de développement. Carthage Solutions a comme objectif d'entrer le monde de l'intelligence artificiel. Comme initiation notre encadrant a divisé ses stagiaires en groupes pour partager les différentes tâches. Carthage Solutions offre à sa clientèle une variété de services, Ci-dessous les services principaux :

- *développement web

Sites vitrines, responsive design, Solutions clé en main (basé sur des CMS), développements spécifiques. Réalisation et conception d'une application web de gestion des recrutements

- * développement mobile

Application Mobile adapté sur la majorité des plateformes du marché (Android, iOS, Windows phone, Black Berry, ...)

- *Web design

Identités digitales : Conception de logo, maquette graphique, Bannière publicitaire.

- *Maintenance évolutive et corrective

Sauvegarde mensuelles, mise à jour régulière du CMS et de ces plugins/module, changement des mots de passe (pour des raisons de sécurité).

- *Hébergement et Support

Nom de domaine, espace d'hébergement, Certificat SSL, création de comptes mail, configuration serveur VPS, infogérance.

1.3 Présentation du projet

1.3.1 Cadre du projet

Ce projet intitulé "Le développement d'un système de prédiction d'infection SARS- COV2" est réalisé dans le cadre du stage d'ingénieur qui a lieu après la quatrième année du cycle d'ingénieur en informatique à l'école Supérieure Privée d'Ingénierie et de Technologies durant l'année académique 2020/2021 et qui a pour but de permettre à l'élève ingénieur de confronter ses études théoriques à la réalité de l'entreprise, De découvrir les multiples aspects du fonctionnement d'une entreprise ou d'une organisation, De témoigner de son expérience pratique en milieu professionnel et de démontrer sa compétence dans des situations réelles d'ingénierie. Il a été réalisé au sein de la société Carthage solutions.

1.3.2 Contexte, problématique et solution proposé

L'analyse de l'état actuel est une étape primordiale pour étudier nos besoins dans le futur et savoir comment améliorer l'existant.

La récente épidémie de troubles respiratoires COVID-19 provoquée par le nouveau coronavirus SARS-Cov2 constitue une préoccupation mondiale sévère et urgente. En l'absence de traitements efficaces, la meilleure stratégie de confinement consiste à réduire la contagion en isolant les personnes infectées, mais l'isolement des personnes non contaminées est fortement indésirable. Pour aider à une prise de décision rapide sur les besoins en ce qui concerne le traitement et l'isolement, il serait très utile de déterminer quelles sont les caractéristiques présentées par les cas suspects d'infection qui sont les principaux prédicteurs d'un diagnostic favorable. Cela peut se faire en étudiant les particularités du patient, la progression du cas, les facteurs de risques, les symptômes, le diagnostic et les résultats. J'ai développé un modèle qui utilise des algorithmes d'apprentissage automatique supervisé pour identifier les particularités de présentation permettant de prédire avec une grande précision les diagnostics de la maladie COVID-19.

Les aspects examinés comprenaient des détails sur les individus concernés, par exemple, l'âge, le sexe, la fièvre observée, un historique des voyages, et des informations cliniques telles que la gravité de la toux et la fréquence de l'infection pulmonaire. J'ai mis en œuvre

et appliqué différents algorithmes d'apprentissage automatique aux données recueillies et j'ai constaté que l'algorithme naïve Bayes était le plus précis (>90%) pour prédire et sélectionner les critères qui indiqueraient correctement le statut COVID-19 pour tous les groupes d'âge.

Mon modèle prédictif pourrait améliorer significativement la prédiction du statut COVID- 19, y compris aux premiers stades de l'infection.

Ma solution est alors une plateforme que tout le monde peut utiliser gratuitement et il suffit de suivre et répondre correctement aux questions pour avoir la prédiction au résultat finale.

1.4 Conclusion

Dans ce chapitre, il a été question de présenter mon organisme d'accueil. De plus, je présenterai le contexte dans lequel se situe ce projet et la problématique qu'il vient de résoudre. Finalement j'exposerai ma solution proposée.

En se basant sur cette étude, je spécifierai dans le chapitre suivant une analyse statistique et descriptive du modèle développé dans mon projet.

Chapitre 2 : Analyse statistique et descriptive du modèle

2.1 Introduction

L'analyse statistique descriptive nous aide à comprendre nos données et constitue une partie très importante de l'apprentissage automatique. Ceci est du fait que l'apprentissage automatique consiste à faire des prédictions. Les statistiques, quant à elles, consistent à tirer des conclusions à partir des données, ce qui constitue une étape initiale nécessaire.

Dans ce chapitre, nous allons découvrir les concepts statistiques descriptifs les plus importants. Ils nous aident à mieux comprendre ce que nos données essaient de nous dire pour obtenir un meilleur modèle et une meilleure compréhension de l'apprentissage automatique.

2.2 Analyse statistique

2.2.1 Définitions

Il est absolument crucial de procéder à une analyse statistique descriptive de notre ensemble de données. De nombreuses personnes sautent cette partie et perdent ainsi beaucoup d'informations précieuses sur leurs données, ce qui conduit souvent à des conclusions erronées.

Prenez votre temps et exécutez soigneusement les statistiques descriptives et assurez-vous que les données répondent aux exigences d'une analyse plus poussée. Mais d'abord, nous devrions considérer ce que sont réellement les statistiques : "Les statistiques sont une branche des mathématiques qui traite de la collecte, l'organisation et l'interprétation des données."

Au sein des statistiques, il existe deux catégories principales :

2.2.1.1 Statistique descriptives

Avec les statistiques descriptives, nous décrivons, nous présentons, nous résumons et nous organisons nos données (population), soit par des calculs mathématiques, soit par des graphes ou des tables.

```
dataset.describe(include=object)
```

	batch_date	test_name	swab_type	covid19_test_results	high_risk_exposure_occupation	high_risk_interactions	rapid_flu_results	rapid_strep_results
count	11169	11169	11169	11169	11000	9668	165	86
unique	11	5	5	2	2	2	2	2
top	2020-05-05	Rapid COVID-19 Test	Nasopharyngeal	Negative	False	False	Negative	Negative
freq	1601	3714	6248	10854	8933	6763	164	80

Figure 1: Statistique descriptive des données

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 11169 entries, 0 to 1445
Data columns (total 46 columns):
 #   Column                                Non-Null Count  Dtype
---  -
 0   batch_date                            11169 non-null  object
 1   test_name                             11169 non-null  object
 2   swab_type                             11169 non-null  object
 3   covid19_test_results                  11169 non-null  object
 4   age                                   11169 non-null  int64
 5   high_risk_exposure_occupation         11000 non-null  object
 6   high_risk_interactions                9668 non-null   object
 7   diabetes                              11169 non-null   bool
 8   chd                                    11169 non-null   bool
 9   htn                                    11169 non-null   bool
10  cancer                                11169 non-null   bool
11  asthma                                11169 non-null   bool
12  copd                                   11169 non-null   bool
13  autoimmune_dis                        11169 non-null   bool
14  smoker                                11169 non-null   bool
15  temperature                           6542 non-null   float64
16  pulse                                 6525 non-null   float64
17  sys                                    6551 non-null   float64
18  dia                                    6551 non-null   float64
19  rr                                     5762 non-null   float64
20  sats                                   6386 non-null   float64
21  rapid_flu_results                     165 non-null    object
22  rapid_strep_results                   86 non-null     object
23  ctab                                   5876 non-null   object
24  labored_respiration                   7023 non-null   object
25  rhonchi                               4511 non-null   object
```

Figure 2 :Informations générale sur les données

2.2.1.2 Statistique inférentielle

Les statistiques inférentielles sont générées par des calculs numériques plus complexes et nous permettent de déduire des tendances et de faire des hypothèses et des prédictions sur une population sur la base de l'étude d'un échantillon prélevé dans celle-ci.

2.2.2 Distribution normale

La distribution normale est l'un des concepts les plus essentiels en statistique car presque tous les tests statistiques exigent des données normalement distribuées. Elle décrit essentiellement l'aspect de grands échantillons de données lorsqu'ils sont représentés graphiquement. On l'appelle parfois la "courbe en cloche" ou la "courbe gaussienne". Les statistiques inférentielles et les calculs de probabilité requièrent qu'une distribution normale soit donnée. Cela signifie en gros que si vos données ne sont pas distribuées normalement, vous devez faire très attention aux tests statistiques que vous appliquez car ils pourraient conduire à des conclusions erronées. Dans une distribution normale parfaite, chaque côté est un miroir exact de l'autre.

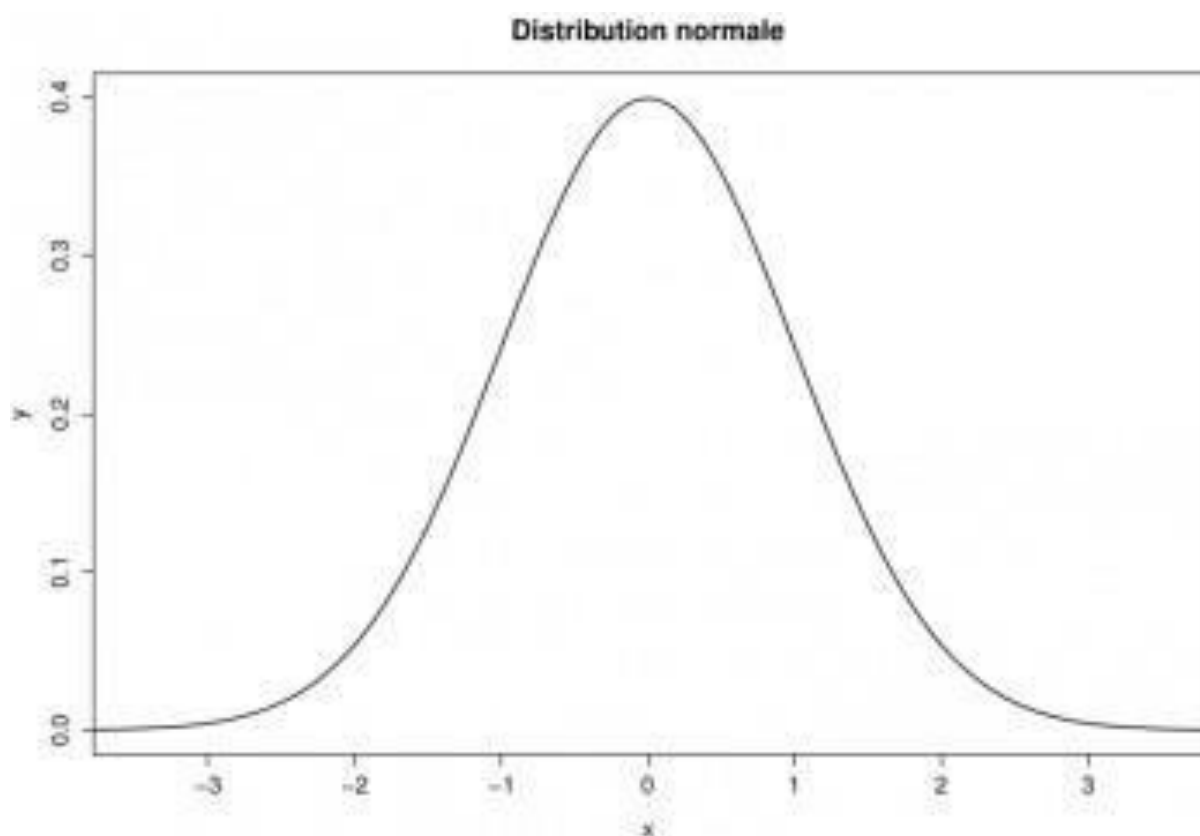


Figure 3 Distribution normale

2.2.3 Mesure de variabilité

Les mesures de variabilité les plus courantes sont l'intervalle, l'intervalle interquartile (IQR), la variance et l'écart type.

Ceux-ci sont utilisés pour mesurer la quantité de propagation ou de variabilité au sein de vos données. La plage décrit la différence entre le plus grand et le plus petit point de vos données. L'intervalle interquartile (IQR) est une mesure de la dispersion statistique entre les quartiles supérieurs (75 e) et inférieurs (25 e).

Alors que la plage mesure le début et la fin de votre point de données, la plage inter-quartile est une mesure de l'emplacement de la majorité des valeurs.

Ci-dessous je vais continuer par vous expliquerai la différence entre l'écart type et la variance.

2.2.4 Variance et écart type

L'écart type et la variance mesurent également, comme la plage et l'IQR, la dispersion de nos données. Par conséquent, ils sont tous deux dérivés de la moyenne.

La variance est calculée en trouvant la différence entre chaque point de données et la moyenne, en les mettant au carré, puis en les additionnant et finalement en prenant la moyenne de ces nombres. Les carrés sont utilisés lors du calcul car ils pondèrent les valeurs aberrantes plus fortement que les points proches de la moyenne. Cela évite le fait que les différences au-dessus de la moyenne neutralisent celles au-dessous de la moyenne.

Le problème avec la variance est qu'en raison de la mise au carré, elle n'est pas dans la même unité de mesure que les données d'origine. Supposons que vous ayez affaire à un ensemble de données contenant des valeurs en centimètres. Votre variance serait en centimètres carrés et donc pas la meilleure mesure.

C'est pourquoi l'écart type est utilisé plus souvent car il se trouve dans l'unité d'origine. Il s'agit tout simplement de la racine carrée de la variance et de ce fait, elle est renvoyée à l'unité de mesure d'origine.

Examinons un exemple qui illustre bien la différence entre la variance et l'écart type : Imaginez un ensemble de données contenant des valeurs en centimètres comprises entre 1 et 15, ce qui donne une moyenne de 8.

La quadrature de la différence entre chaque point de données et la moyenne et la moyenne des carrés donne une variance de 18,67 (centimètres carrés), tandis que l'écart type est 4,3 centimètres. Lorsque vous avez un faible écart type, vos points de données ont tendance à être proches de la moyenne.

Un écart type élevé signifie que vos points de données sont repartis sur une large plage. L'écart type est mieux utilisé lorsque les données sont unimodales.

Dans une distribution normale, environ 34 % des points de données se situent entre la moyenne et un écart type au-dessus ou au-dessous de la moyenne. Puisqu'une distribution normale est symétrique, 68% des points de données se situent entre un écart-type au-dessus et un écart-type en dessous de la moyenne.

Environ 95% se situent entre deux écarts-types en dessous de la moyenne et deux écarts-types au-dessus de la moyenne. Et environ 99,7% se situent entre trois écarts types au-dessus et trois écarts types en dessous de la moyenne.

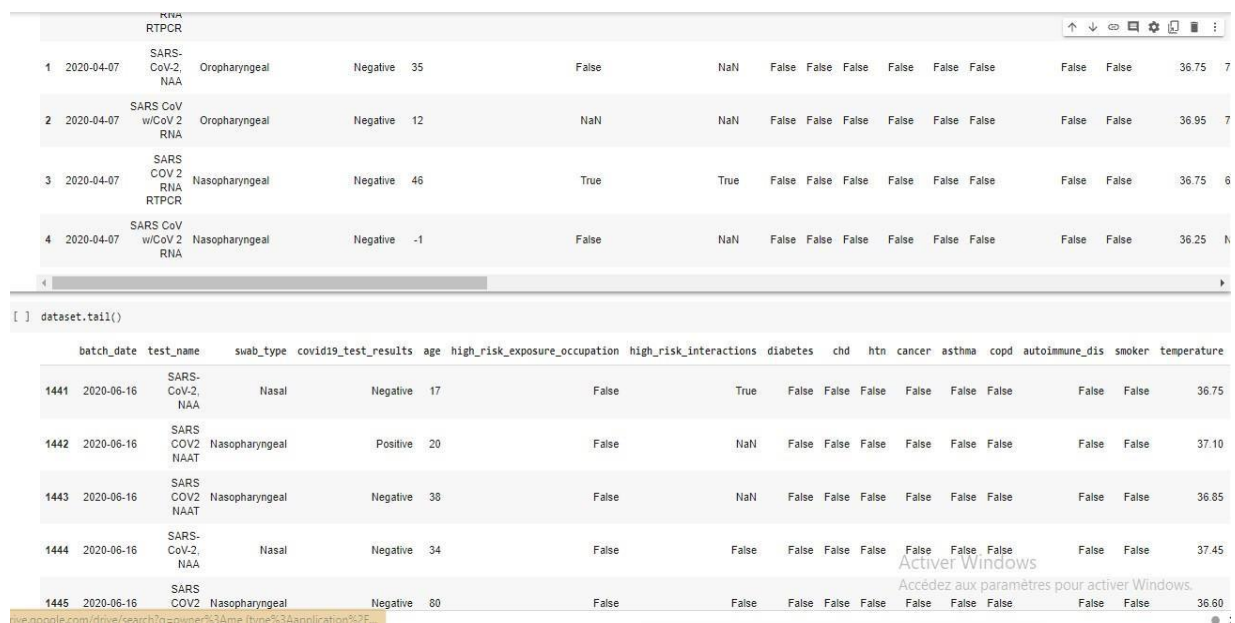
2.3 Description détaillée du modèle

2.3.1 Lecture de données et importation de bibliothèques

2.3.1.1 Chargement et assemblage des fichiers CSV

Après la partie de recherche dans le but d'obtenir une base de données pertinente, nous passons ensuite à la première étape de la mise en œuvre de notre modèle qui est l'étape d'importation des bibliothèques nécessaires qui sont dans ce cas : matplotlib, pandas, seaborn et numpy afin de charger et lire par la suite les fichiers csv et ensuite nous faisons l'assemblage de ces fichiers dans un seul ensemble de données pour qu'on puisse à la fin le visualiser et le manipuler.

Des captures qui existent dans l'annexe prouvent cette étape.



	batch_date	test_name	swab_type	covid19_test_results	age	high_risk_exposure_occupation	high_risk_interactions	diabetes	chd	htn	cancer	asthma	copd	autoimmune_dis	smoker	temperature
1	2020-04-07	SARS-CoV-2, NAA	Oropharyngeal	Negative	35	False	NaN	False	False	False	False	False	False	False	False	36.75
2	2020-04-07	SARS CoV w/CoV 2 RNA	Oropharyngeal	Negative	12	NaN	NaN	False	False	False	False	False	False	False	False	36.95
3	2020-04-07	SARS COV 2 RNA RTPCR	Nasopharyngeal	Negative	46	True	True	False	False	False	False	False	False	False	False	36.75
4	2020-04-07	SARS CoV w/CoV 2 RNA	Nasopharyngeal	Negative	-1	False	NaN	False	False	False	False	False	False	False	False	36.25
1441	2020-06-16	SARS-CoV-2, NAA	Nasal	Negative	17	False	True	False	False	False	False	False	False	False	False	36.75
1442	2020-06-16	SARS COV2 NAAT	Nasopharyngeal	Positive	20	False	NaN	False	False	False	False	False	False	False	False	37.10
1443	2020-06-16	SARS COV2 NAAT	Nasopharyngeal	Negative	38	False	NaN	False	False	False	False	False	False	False	False	36.85
1444	2020-06-16	SARS-CoV-2, NAA	Nasal	Negative	34	False	False	False	False	False	False	False	False	False	False	37.45
1445	2020-06-16	SARS COV2	Nasopharyngeal	Negative	80	False	False	False	False	False	False	False	False	False	False	36.60

Figure 4: Assemblage des fichiers

2.3.1.2 Dimensions et type de données

Vous devez avoir une véritable vision de la quantité de données dont vous disposez, à la fois en termes de lignes et de colonnes.

Également, le type de chaque attribut est important. Les chaînes de caractères peuvent avoir à être converties en valeurs à virgule flottante ou en nombres entiers pour représenter des valeurs catégoriques ou ordinales. Vous pouvez avoir une idée des types de données en regardant les données brutes, comme ci-dessus. Vous pouvez également dresser la liste des types de données utilisés par le DataFrame pour caractériser chaque attribut à l'aide de la propriété "dtypes".

```
dataset.shape
```

```
(11169, 33)
```

Figure 5: Dimensions des données



dataset.dtypes	
covid19_test_results	uint8
age	int64
high_risk_exposure_occupation	uint8
high_risk_interactions	uint8
diabetes	uint8
chd	uint8
htn	uint8
cancer	uint8
asthma	uint8
copd	uint8
autoimmune_dis	uint8
smoker	uint8
temperature	float64
pulse	float64
sys	float64
dia	float64
rr	float64
sats	float64
labored_respiration	uint8
wheezes	uint8
days_since_symptom_onset	float64
cough	uint8
fever	uint8
sob	uint8
diarrhea	uint8
fatigue	uint8
headache	uint8
loss_of_smell	uint8
loss_of_taste	uint8
runny_nose	uint8
muscle_sore	uint8
sore_throat	uint8

Figure 6: Types de données

2.3.1.3 Corrélations entre les attributs

La corrélation fait référence à la relation entre deux variables et à la façon dont elles peuvent ou non évoluer ensemble. La méthode la plus courante pour calculer la corrélation est le coefficient de corrélation de Pearson, qui suppose une distribution normale des attributs concernés. A corrélation de -1 ou 1 indique respectivement une corrélation négative ou positive complète. Alors qu'une valeur de 0 n'indique aucune corrélation du tout.

Certains algorithmes d'apprentissage automatique tels que la régression linéaire et logistique peuvent souffrir de performances médiocres s'il existe des attributs fortement corrélés dans votre ensemble de données. En tant que tel, c'est une bonne idée de passer en revue toutes les corrélations par paires des attributs de votre ensemble de données. Vous pouvez utiliser la fonction `corr()` sur le DataFrame de Pandas pour calculer une matrice de corrélation.

La figure ,ci-dessous illustrent bien la corrélation entre les attributs de ma base de données.

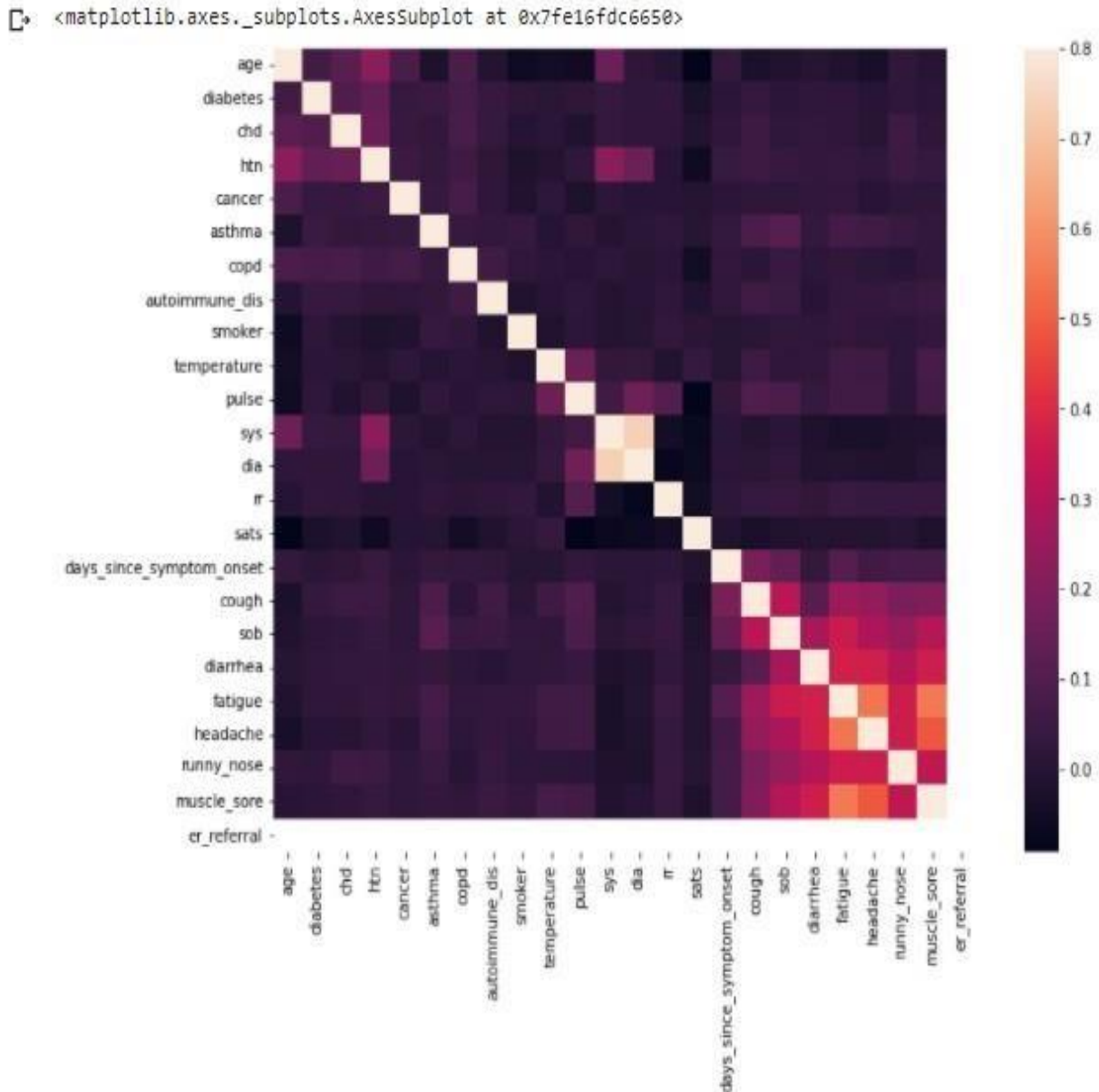


Figure 7: Matrice de corrélation

2.3.2 Analyse exploratoire des données

2.3.2.1 Visualiser et Supprimer les données manquantes

En fait, l'ensemble de données a toujours besoin de ce que nous appelons le nettoyage des données dans lequel nous devons supprimer les colonnes contenant plus de 50% de données manquantes ou "NaN" mais il faut visualiser d'abord ces données pour qu'on puisse de faire ce traitement.

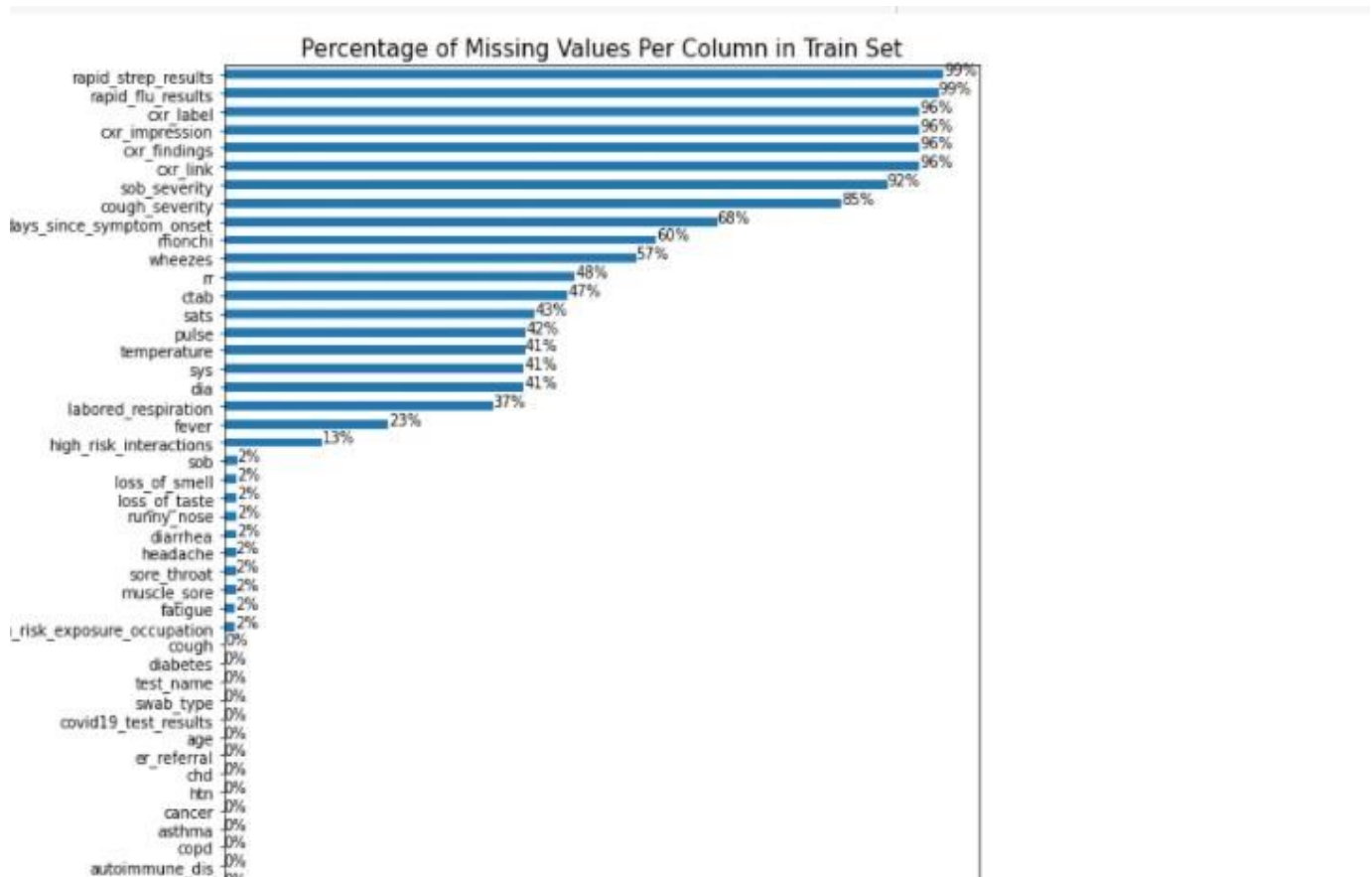


Figure 8 Visualisation des données manquantes

```
Entrée [ ]: data.drop(['cxr_findings', 'cxr_impression', 'cxr_label', 'cxr_link', 'sob_severity', 'rapid_strep_results', 'rapid_flu_results', 'rhonchi', 'lays_since_symptom_onset', 'cough_severity', 'wheezes', 'rr', 'ctab', 'sats', 'pulse', 'temperature', 'sys', 'dia', 'labored_respiration', 'fever', 'high_risk_interactions', 'sob', 'loss_of_smell', 'loss_of_taste', 'runny_nose', 'diarrhea', 'headache', 'sore_throat', 'muscle_sore', 'fatigue', '_risk_exposure_occupation', 'cough', 'diabetes', 'test_name', 'swab_type', 'covid19_test_results', 'age', 'er_referral', 'chd', 'htn', 'cancer', 'asthma', 'copd', 'autoimmune_dis'])
```

Figure 9:supprimer les colonnes indésirable

2.3.2.2 Remplacer les colonnes supprimées

Nous devons remplir les lignes booléennes évidentes avec les meilleures informations appropriées, par exemple si un patient a un résultat de test positif et que sa colonne "toux" est NaN alors il vaut mieux le considérer comme Vrai pour une meilleure classification et pour cela nous incluons un pour la boucle pour traiter cela comme indiqué sur la figure 10

```
:hi','wheezes','days_since_symptom_onset','cough_severity','batch_date','test_name','swab_type','sob','ctab'],axis=1,inplace=True)
```

Figure 10: Remplacer les colonnes supprimées

2.3.2.3 Gérer les données manquantes

Nous devons traiter les colonnes numériques NaN en remplaçant les valeurs NaN par des valeurs médianes

et enfin nous nous fierons à la fonction Fillna() pour le reste

```
Entrée [ ]: m=data['temperature'].median()
            s=data['sys'].median()
            d=data['dia'].median()
            r=data['rr'].median()
            p=data['pulse'].median()
            sa=data['sats'].median()

            data['temperature'].replace(np.nan,m,inplace=True)
            data['sats'].replace(np.nan,sa,inplace=True)
            data['rr'].replace(np.nan,r,inplace=True)
            data['dia'].replace(np.nan,d,inplace=True)
            data['sys'].replace(np.nan,s,inplace=True)
            data['pulse'].replace(np.nan,p,inplace=True)
            np.any(np.isnan(data['rr']))
```

```
Entrée [ ]: data=data[data['age']>0]
```

```
Entrée [ ]: data.fillna(method='bfill',inplace=True)
```

Figure 11: Gérer les données manquantes

2.3.2.4 La normalisation des données

La normalisation est une technique utile pour transformer des attributs avec une distribution gaussienne et moyennes différentes et écarts types par rapport à une distribution gaussienne standard avec une moyenne de 0 et un écart-type de 1.

Il est plus approprié pour les techniques qui supposent une gaussienne distribution dans les variables d'entrée et fonctionne mieux avec des données redimensionnées, telles que la régression linéaire, régression logistique et analyse discriminante linéaire. Vous pouvez standardiser les données à l'aide de scikit-learn avec la classe `StandardScaler` et classe `Normalizer`.

```
Entrée [ ]: data['high_risk_exposure_occupation']=data['high_risk_exposure_occupation'].astype('bool')
data['high_risk_interactions']=data['high_risk_interactions'].astype('bool')
data['labored_respiration']=data['labored_respiration'].astype('bool')
data['loss_of_smell']=data['loss_of_smell'].astype('bool')
data['loss_of_taste']=data['loss_of_taste'].astype('bool')
data['sore_throat']=data['sore_throat'].astype('bool')
```

```
Entrée [ ]: from numpy import set_printoptions
from sklearn.preprocessing import Normalizer
scaler=Normalizer().fit(X)
normalizedX=scaler.transform(X)
set_printoptions(precision=3)
print(normalizedX[0:5,:])
```

```
Entrée [ ]: from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Figure 12: Data preprocessing

2.3.3 Implémentation de model et “Feature selection”:

2.3.3.1 Fractionnement de nos données en ensemble de d’entrainement et ensemble de test :

Afin d'entraîner notre modèle à prédire les personnes malades, nous lui donnons 75% de nos données et 25% pour le tester et ensuite améliorer ses performances.

```
Entrée [16]: from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X_new,y,test_size=0.25,random_state=0)
```

Figure 13:Diviser la dataset

2.3.3.2 Feature selection

Les tests statistiques peuvent être utilisés pour sélectionner les fonctionnalités qui ont la relation la plus forte avec la variable de sortie. La bibliothèque scikit-learn fournit la classe SelectKBest qui peut être utilisé avec une suite de différents tests statistiques pour sélectionner un nombre spécifique de fonctionnalités.

Dans notre cas on utilise le chi carré (chi2) test statistique des caractéristiques non négatives pour sélectionner 4 des meilleurs caractéristiques de l'ensemble de données.

```
Entrée [15]: from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2

X_new = SelectKBest(chi2, k=10).fit_transform(X,y)
fit=SelectKBest(chi2, k=10).fit(X,y)
print(fit.scores_)

[1.35496755e+02 8.56072619e+00 2.61476050e+00 1.41523774e+00
 1.51327075e+00 8.35843084e-02 1.18353224e-01 5.03406406e+00
 8.71440650e-01 6.52058476e-01 7.84794648e+00 1.61968624e-01
 8.52705778e+01 1.36803642e+00 3.80049927e+00 2.32886381e+00
 2.27394453e-01 3.09724038e+01 3.97452944e+01 8.06404929e+01
 6.29232487e-01 1.23621920e-01 7.96597570e+00 1.15654467e+02
 1.05510166e+02 8.73263198e-01 1.61975668e+01 1.39677594e-01
 nan]
```

```
Entrée [19]: SelectKBest(chi2, k=10).fit(X,y).get_support(indices=True)
```

```
Out[19]: array([ 0,  1, 12, 17, 18, 19, 22, 23, 24, 26], dtype=int64)
```

Figure 14:Test CHI2

2.3.3.3 Création de classificateur :

C'est le moment de commencer à construire notre classificateur dans notre cas, nous utiliserons le classificateur 'Naïve bayes'

```
Entrée [ ]: from sklearn.naive_bayes import GaussianNB  
            classifieur = GaussianNB()  
            classifieur.fit(X_train, y_train)
```

Figure 15:Création d'un classificateur

2.3.3.4 Prédiction et Validation croisée de K-fold

La validation croisée est une approche que vous pouvez utiliser pour estimer les performances d'une machine algorithmique d'apprentissage avec moins de variance qu'une seule division de train-test. Cela fonctionne en divisant l'ensemble de données en k-parties (par exemple, k = 5 ou k = 10). Chaque division des données est appelée un pli. L'algorithme est entraîné sur k - 1 plis avec un repli retenu et testé sur le pli retenu.

C'est répété afin que chaque pli de l'ensemble de données ait une chance d'être l'ensemble de test retenu. Après en exécutant la validation croisée, vous vous retrouvez avec k scores de performance différents que vous pouvez résumer en utilisant une moyenne et un écart type. Le résultat est une estimation plus fiable des performances de l'algorithme sur de nouvelles données. Il est plus précis car l'algorithme est entraîné et évalué plusieurs fois sur différentes données. Le choix de k doit permettre à la taille de chaque partition de test d'être suffisamment grande pour être un raisonnable échantillon du problème, tout en permettant suffisamment de répétitions de l'évaluation du train-test de l'algorithme pour fournir une estimation juste des performances des algorithmes sur des données invisibles.

Pour modestes ensembles de données de taille en milliers ou dizaines de milliers d'enregistrements, k valeurs de 3, 5 et 10 sont communes. Dans l'exemple ci-dessous, nous utilisons la validation croisée par 10.

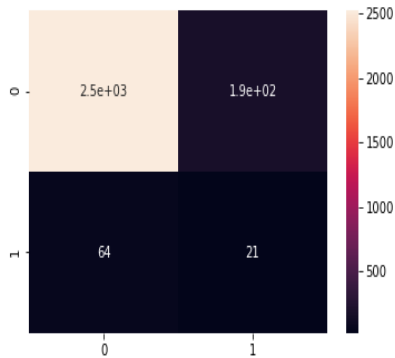
```

Intr  e [21]: y_pred = classifier.predict(X_test)

# Making the Confusion Matrix
import seaborn as sns
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm,annot=True)

```

Out[21]: <AxesSubplot:>



```

Intr  e [22]: print(cm)

```

```

[[2519  186]
 [   64   21]]

```

```

Intr  e [23]: from sklearn import metrics
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))

```

Accuracy: 0.910394265232975

Figure 16:Matrice de confusion

2.3.3.4 R  glage des hyper param  tres et   valuation du mod  le :

1)R  glage des hyper param  tres :

Les mod  les d'apprentissage automatique sont param  tr  s afin que leur comportement puisse   tre ajust   pour un probl  me. Les mod  les peuvent avoir de nombreux param  tres et trouver la meilleure combinaison de param  tres peut   tre trait   comme un probl  me de recherche. Dans ce chapitre, vous d  couvrerez comment r  gler les param  tres d'algorithmes d'apprentissage automatique en Python    l'aide de scikit-learn.

GridsearchCV est une approche du r  glage des param  tres qui construira et   valuera m  thodiquement un mod  le pour chaque combinaison de param  tres d'algorithme sp  cifi  s dans une grille. Vous pouvez effectuer une recherche de grille en utilisant la classe GridSearchCV1.


```
Entrée [26]: from sklearn.model_selection import RepeatedStratifiedKFold
cv_method = RepeatedStratifiedKFold(n_splits=5, n_repeats=3, random_state=999)
```

```
Entrée [ ]: from sklearn.model_selection import GridSearchCV
from sklearn.preprocessing import PowerTransformer

params_NB = {'var_smoothing': np.logspace(0, -9, num=100)}

gs_NB = GridSearchCV(estimator=classifier,
                     param_grid=params_NB,
                     cv=cv_method,
                     verbose=1,
                     scoring='accuracy')

Data_transformed = PowerTransformer().fit_transform(X_test)

gs_NB.fit(Data_transformed, y_test)
```

Figure 17:GridsearchCV

2.3.3.5 Evaluation du modèle

L'une des mesures les plus couramment utilisées de nos jours est la courbe AUC-ROC (Area Under Curve - Receiver Operating Characteristics). Les courbes ROC sont assez faciles à comprendre et à évaluer une fois qu'il y a une bonne compréhension de la matrice de confusion et des différents types d'erreurs.

```
from sklearn.metrics import classification_report
model_pred = model.predict(X_test)
print(classification_report(y_test, model_pred))
```

	precision	recall	f1-score	support
0	0.97	1.00	0.98	2162
1	0.00	0.00	0.00	72
accuracy			0.97	2234
macro avg	0.48	0.50	0.49	2234
weighted avg	0.94	0.97	0.95	2234

```
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1272: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.
_warn_prf(average, modifier, msg_start, len(result))
```



```

Entrée [27]: from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
classifier_roc_auc = roc_auc_score(y_test, classifier.predict(X_test))
fpr, tpr, thresholds = roc_curve(y_test, classifier.predict_proba(X_test)[:,-1])
pyplot.figure()
pyplot.plot(fpr, tpr, label='classifier (area = %0.2f)' % classifier_roc_auc)
pyplot.plot([0, 1], [0, 1], 'r--')
pyplot.xlim([0.0, 1.0])
pyplot.ylim([0.0, 1.05])
pyplot.xlabel('Taux de faux positifs')
pyplot.ylabel('Taux de vrais positifs')
pyplot.title('ROC')
pyplot.legend(loc="lower right")
pyplot.show()

```

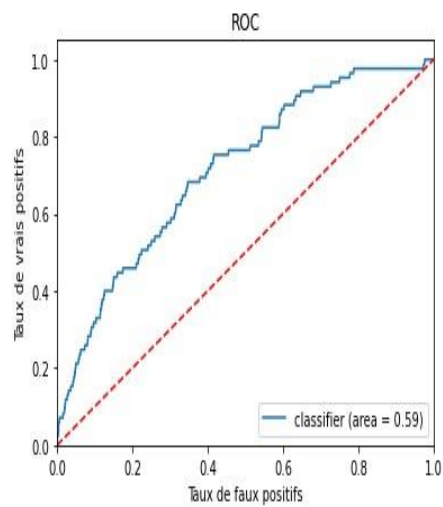


Figure 18: Courbe ROC

2.3.3.6 Finalisons notre modèle avec Job lib :

Joblib est le moyen standard de sérialiser des objets en Python. Vous pouvez utiliser l'opération pickle¹ pour sérialiser vos algorithmes d'apprentissage automatique et enregistrer le format sérialisé dans un fichier. Plus tard vous pouvez charger ce fichier pour désérialiser votre modèle et l'utiliser pour faire de nouvelles prédictions.

```
[ ] from joblib import dump

dump(classifier , 'sars_cov2_pred_gaussien_nb.joblib')

['sars_cov2_pred_gaussien_nb.joblib']
```

Figure 19:Enregistrement de modèle

Chapitre 3 : Réalisation de l'application web

3.1 Présentation des outils

3.1.1 Django

Django est un cadre de développement web open source en Python. Il a pour but de rendre le développement web 2.0 simple et rapide. Pour cette raison, le projet a pour slogan « Le Framework pour les perfectionnistes avec des deadlines. ». Développé en 2003 pour le journal local de Lawrence, Django a été publié sous licence BSD à partir de juillet 2005.



Figure 20: Django

3.1.2 Visual Studio Code

VS code 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.



Figure 21: Visual Studio Code

3.1.3 GitHub

C'est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git (illustration 18). Ce site est développé en Ruby on Rails et Erlang par Chris Wanstrath, PJ Hyett et Tom Preston-Werner.

GitHub propose des comptes professionnels payants, ainsi que des comptes gratuits pour les projets de logiciels libres. Le site assure également un contrôle d'accès et des fonctionnalités destinées à la collaboration comme le suivi des bugs, les demandes de fonctionnalités, la gestion de tâches et un wiki pour chaque projet [9].



Figure 22: Github

3.1.4 React_Js

C'est une bibliothèque JavaScript libre développée par Facebook depuis 2013.

Le but principal de celle-ci est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page (ou portion) HTML à chaque changement d'état

React est une bibliothèque qui ne gère que l'interface de l'application, considéré comme la vue dans le modèle MVC. Elle peut ainsi être utilisée avec une autre bibliothèque ou un Framework MVC comme AngularJS. La bibliothèque se démarque de ses concurrents par sa flexibilité et ses performances, en travaillant avec un DOM virtuel et en ne mettant à jour le rendu dans le navigateur qu'en cas de nécessité

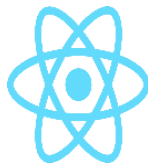


Figure 23: React _Js

3.2 Création d'une application web

3.2.1 Conception

La création et la conception de site web ou web design est la conception de l'interface web : l'architecture interactionnelle, l'organisation des pages, l'arborescence et la navigation dans un site web. La conception d'un design web tient compte des contraintes spécifiques du support Internet, notamment en termes d'ergonomie, d'utilisabilité et d'accessibilité.

Dans notre répertoire Template « / », créez un nouveau fichier appelé home.html.

Ce sera la page qui s'affichera lorsque nous tenterons d'accéder à notre application Web. Pour voir à quoi ressemble le code HTML rendu, vous pouvez ouvrir le fichier dans un navigateur Web.

ajoutons un peu de formatage ! Tout d'abord, choisissons une police différente pour le texte. J'aime aller sur Google Fonts, qui propose un large éventail de polices gratuites que vous pouvez utiliser.

Pour cette application, j'ai choisi une police appelée Rubik. À côté de chacun des poids de police, il y a un bouton qui dit "Sélectionnez ce style" - cliquer sur au moins l'un d'entre eux ouvrira une fenêtre sur la droite qui contient du code que vous pouvez copier/coller dans notre code HTML pour y avoir accès Police de caractère! Je viens de choisir le poids de police Regular 400 et j'ai obtenu le code suivant que j'ai ajouté en haut de mon fichier home.html.

3.2.2 Déploiement de l'application sur Heroku

Heroku est un service qui vous permet de déployer cette application web Python afin que toute personne disposant du lien puisse l'utiliser. La première chose à faire est de créer un compte gratuit sur Heroku. Une fois que vous avez créé un compte, à partir de votre tableau de bord, cliquez sur le bouton

« Créer une nouvelle application ». À partir de là, donnez un nom à votre application et cliquez sur « Créer une application ».

Avant de pouvoir déployer notre application Web, nous devons ajouter quelques fichiers supplémentaires pour qu'Heroku les reconnaisse. Le premier est un fichier appelé requirements.txt et est essentiellement une liste de dépendances Python que le serveur Heroku doit installer pour exécuter votre application. Heureusement, en travaillant dans un environnement virtuel, vous aviez déjà installé les dépendances nécessaires.

Désormais, depuis notre tableau de bord d'application, nous pouvons choisir de nous connecter à GitHub en appuyant sur le bouton associé. Nous serons ensuite redirigés pour nous connecter à GitHub et autoriser l'utilisation par Heroku. Nous pouvons alors choisir le référentiel qui est associé à notre projet. Désormais, depuis notre tableau de bord d'application, nous pouvons choisir de nous connecter à GitHub en appuyant sur le bouton associé. Nous serons ensuite redirigés pour nous connecter à GitHub et autoriser l'utilisation par Heroku.

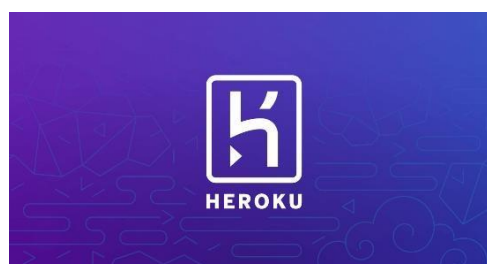
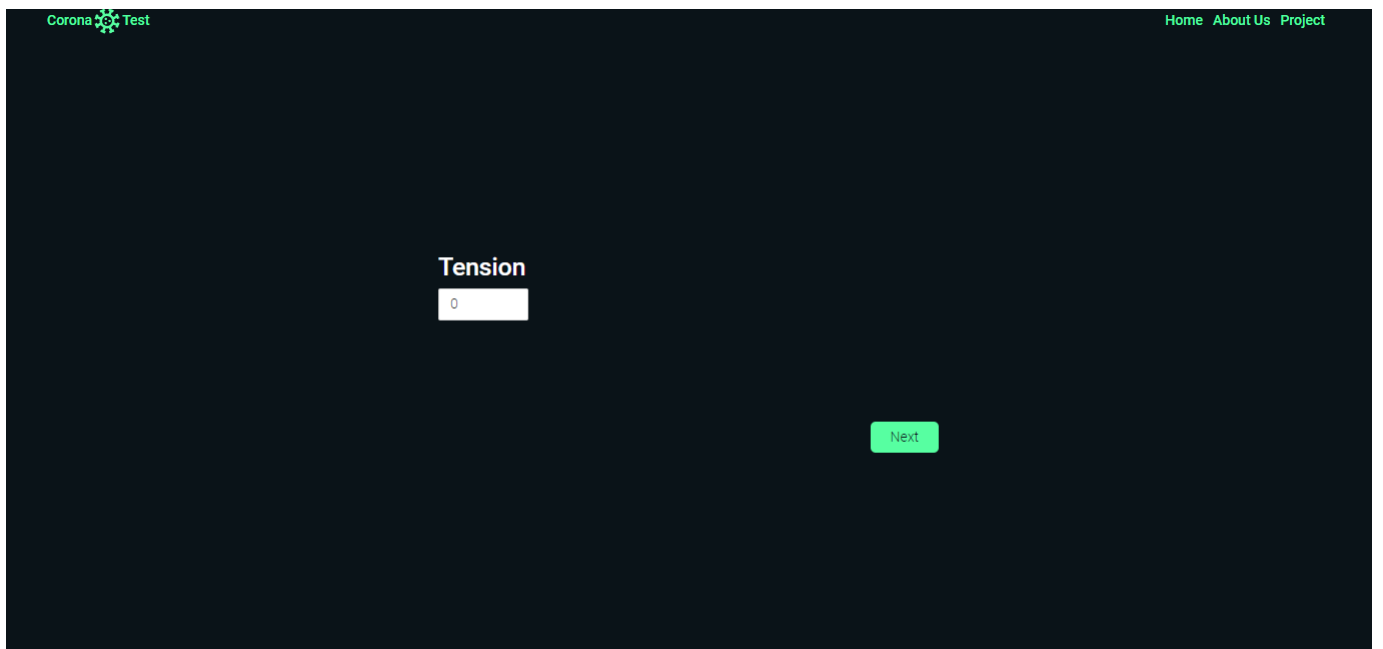
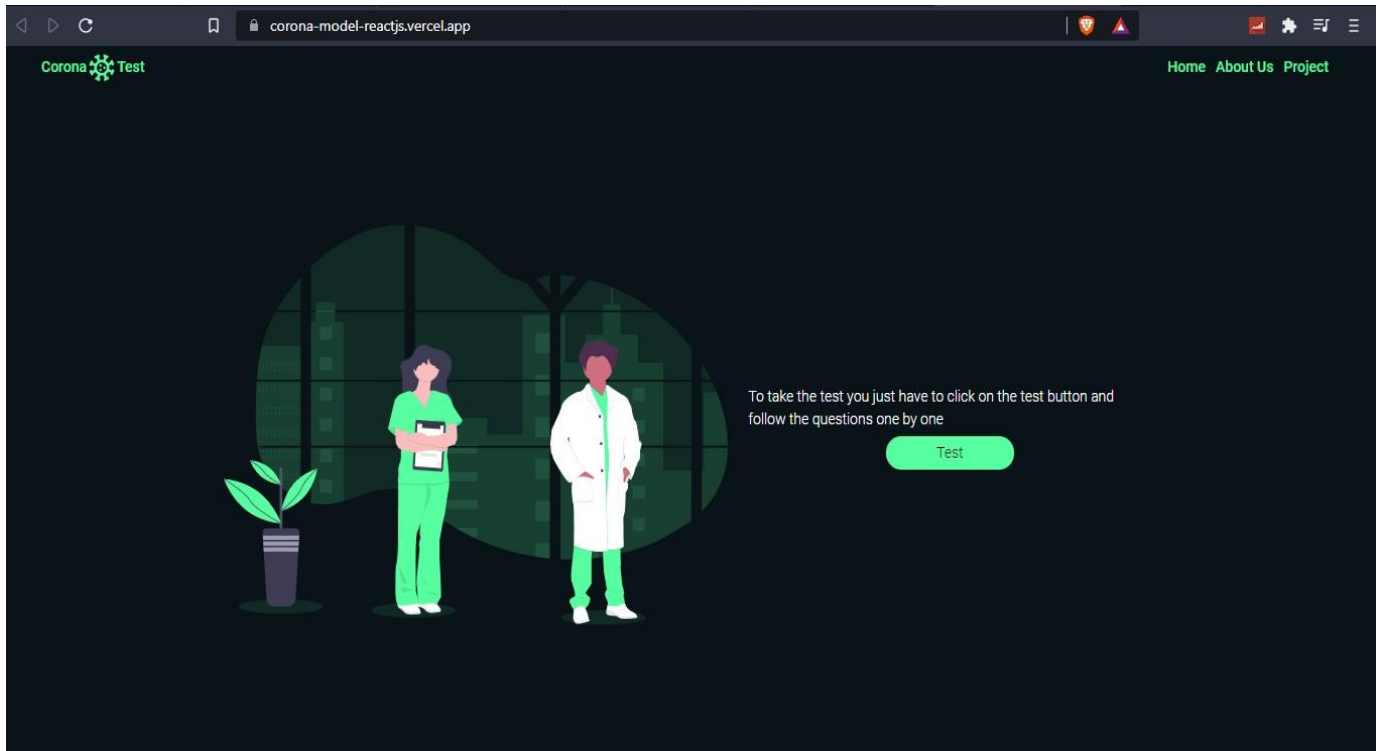


Figure 24:heroku

3.3 Réalisation



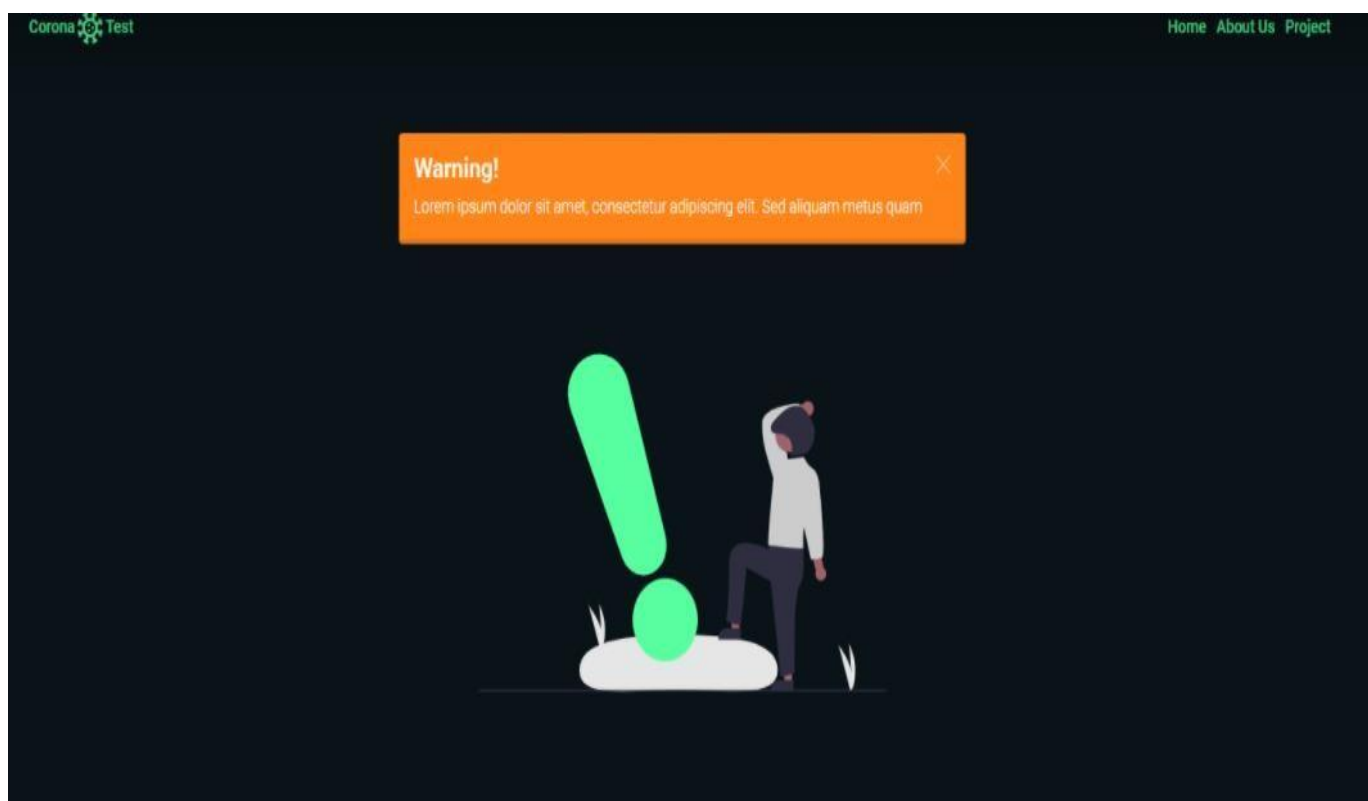


Figure 25: Notre application Web

Conclusion générale

Le développement de la pandémie de COVID-19 représente actuellement une sérieuse menace pour la santé mondiale. La clé pour arrêter cette propagation est le développement de méthodologies permettant d'identifier les individus infectés le plus tôt possible. Cela peut s'avérer difficile étant donné le délai de l'apparition des symptômes.

Cependant, les algorithmes d'apprentissage automatique fournissent une approche prometteuse pour résoudre ce problème et peuvent être appliqués rapidement et à moindre coût dans une situation de panique.

Dans cette étude, j'ai développé et testé une série d'approches d'apprentissage automatique et détermine que les caractéristiques prédictives cliniques COVID-19 les plus significatives étaient (par ordre décroissant) : infection pulmonaire, toux, fatigue, risque dans le travail, des problèmes respiratoires, antécédents de voyage, fièvre, isolement, âge, douleurs musculaires, diarrhée et sexe. mon modèle a pu prédire le stade de la COVID-19 à partir des informations de basés sur le patient (^âge et sexe), du voyage et de l'isolement, et des symptômes cliniques (notamment la fièvre, la toux, perte d'odorat et de goût).

Référence

