REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre : SIOD14/M2/2021

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Système d'Information Optimisation et Décision (SIOD)**

Méthode Bio-informatique pour l'analyse de protéine

Par :
DJABALLAH BACHIR

Soutenu le …/07/2021 devant le jury composé de :

	Grade	Président
Belounnar Saliha	MAA	Rapporteur
	Grade	Examinateur

Année universitaire 2020-2021

Dédicases

Je tiens premièrement à prosterner remerciant Allah le tout puissant de m'avoir donné le courage et la patience pour terminer ce travail. Je voudrais dans un premier temps remercier mon Encadreur, Madame Saliha Bellounar, de m'avoir encadré, orienté, aidé et conseillé. Je tiens à remercier spécialement mon ami Messaoud pour ses conseils et ses critiques, Je remercie mes amis Abdelali et Ayoub , Yasine, et Youcef Rayeh. Je remercie mes très chers parents, et mes frères et ma soeur yakine, pour leur encouragements

Table de Matieres

Liste des Figures

Liste des tables

1. Introduction Général

La famille des Coronaviridae comprend des virus à ARN simple brin de sens positif qui ont une taille de 27 à 32 kb. Il comporte les catégories α, β, et γ. Comme son nom l'indique, la protéine de pointe externe sphérique a une forme de couronne. Le virus s'est avéré infecter un large éventail d'hôtes.

Le coronavirus 2 du syndrome respiratoire aigu sévère (SRAS-CoV-2) est un nouveau type de coronavirus qui provoque une maladie respiratoire sévère avec plusieurs autres manifestations et ayant un taux de mortalité de 4%. Il appartient au genre β-coronavirus de la famille des Coronaviridae et est identique à 96 % sur le plan gnomique avec un coronavirus de chauve-souris de type SRAS (BAT-CoV) précédemment détecté. Il a été identifié pour la première fois dans la province chinoise de Wuhan en décembre 2019. Il a progressé rapidement via l'interaction d'homme à homme et se propager dans tous les grands pays du monde.

Au cours de la transmission et de la réplication au sein de l'hôte, les virus acquièrent des mutations génétiques dans le génome. Le séquençage génomique rapide nous a permis de trouver et d'analyser (classifier) la génétique dans des milliers de séquences du génome viral. Pour l'identification de cibles vaccinales potentielles du virus, il est nécessaire d'identifier la région dans laquelle le virus est fortement muté [1].

Dans le premier chapitre de cette thèse, nous nous concentrerons sur les concepts généraux sur la famille des virus Corona et la biologie de l'analyse de ses puces à ADN, puis dans le deuxième chapitre nous discuterons de l'apprentissage automatique et de son rôle important dans l'analyse et la classification de certains types de virus Corona. Quant au troisième chapitre, il s'agira d'une implémentation des concepts précédents afin de classer parmi les différents types de virus Corona.

Chapitre 1 : Bio-Informatique

2.1 Introduction

L'étude du génome humain connait actuellement un énorme regain d'attention de la part du public, même si le nom de génomique est peu connu. Le séquençage complet du génome humain, relativement récent, a fait la une des journaux. En effet, les gènes sont à la base de toute vie terrestre. Il est donc normal qu'en tant qu'êtres vivants, nous soyons fascinés par cette sorte de programme qui nous construit. Il est tout aussi normal que sa modification pose des problèmes éthiques profonds et suscite des oppositions farouches. Si la bio-informatique est très liée à l'étude des génomes, elle ne s'y limite pas. Elle concerne aussi l'étude des protéines, brique de base du vivant dans son ensemble. Elle connait également des développements moins directement liés au vivant, la programmation biologique, utilisation d'ADN pour résoudre des problèmes informatiques, en est un bon exemple [2].

2.2 Éléments de biologie

2.2.1 La cellule

Unité structurale et fonctionnelle de tous les êtres vivants (animaux, plantes, champignons et micro-organismes), la cellule est une entité biologique d'une très grande complexité, en dépit de ses dimensions microscopiques, elle représente un espace clos, séparé du milieu environnant par une membrane périphérique, la membrane plasmique. Une enveloppe protectrice de structure variable, la paroi cellulaire, entoure la membrane plasmique. Les cellules possèdent toutes une membrane plasmique qui définissent la frontière entre le milieu interne et le milieu externe de la cellule. Le milieu interne est une substance riche en eau, appelée cytoplasme. La membrane plasmique Toutes les cellules contiennent par ailleurs des informations héréditaires (gènes) portées par des filaments d'ADN. L'ADN contrôle les activités de la cellule et lui permettent de se reproduire en transmettant ses caractéristiques à ses descendantes, issues des divisions cellulaires, il est le support de l'hérédité [3].

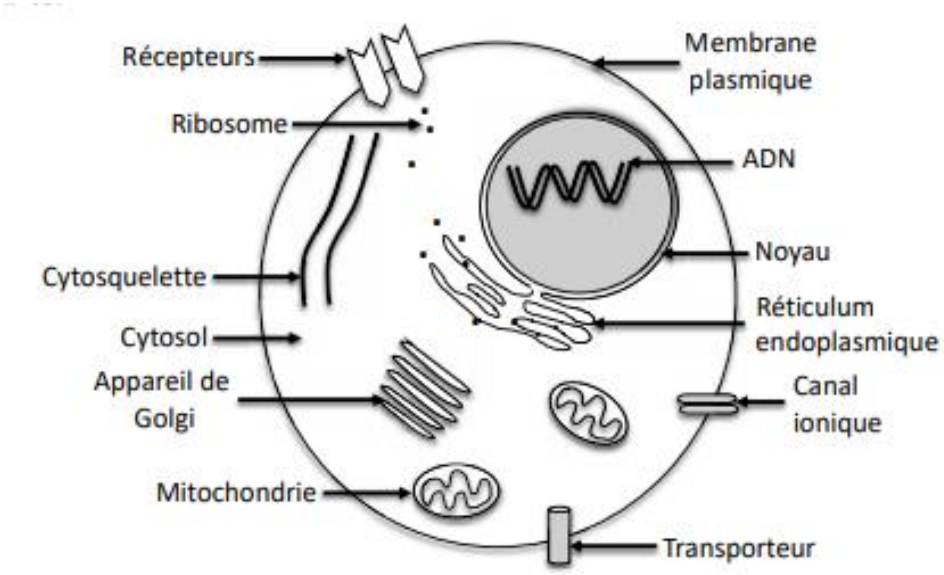


Figure 1. 1 : Schéma simplifié d'une cellule eucaryote.

2.2.2 Les virus

Un virus est une particule microscopique infectieuse qui ne peut se répliquer qu'en pénétrant dans une cellule et en utilisant sa machinerie cellulaire [4]. Les virus existent sous une forme extracellulaire ou intracellulaire [5]. Elles sont des éléments génétiques qui peuvent se répliquer de façon indépendante par rapport au chromosome, mais non indépendamment de la cellule hôte.

C'est en 1953 qu'André LWOFF a énoncé les trois caractères fondamentaux faisant des virus des entités originales [4]:

- Les virus ne contiennent qu'un seul type d'acide nucléique (ADN ou ARN) qui constitue le génome viral.
- Les virus se reproduisent à partir de leur matériel génétique et par réplication.
- Les virus sont doués de parasitisme intracellulaire absolu.

Le génome

Un virus comporte toujours un génome qui est du DNA ou du RNA, de sorte que dans la classification des virus on distingue en premier lieu virus à DNA et virus à RNA. Ce génome peut être monocaténaire (simple brin) ou bicaténaire (double brin) [4].

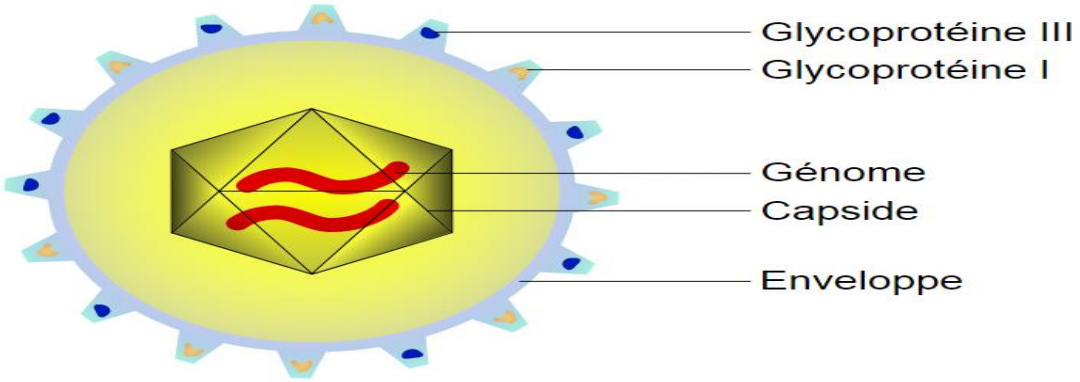


Figure 1. 2 : Schéma simplifié d'une cellule de virus.

2.2.3 Coronavirus

Virus	Désignation *	Hôte	Maladies
Virus de la bronchite infectieuse	IBV	Poulet	Maladies respiratoires - néphrite
Virus de l'hépatite murine	MHV	Souris	Hépatite - encéphalomyélite - entérite - pneumonie
Coronavirus bovin	BCV	Veau	Entérite
Coronavirus humain	HCV	Homme	Maladies respiratoires
Virus de la gastroentérite transmissible	TGEV	Porc	Entérite
Virus hémagglutinant de l'encéphalomyélite	HEV	Porc	Encéphalomyélite - entérite
Coronavirus du rat	RCV	Rat	Pneumonie - rhinotrachéite
Virus de la sialodacryoadénite	SDAV	Rat	Sialodacryoadénite
Coronavirus canin	CCV	Chien	Entérite
Virus de la péritonite infectieuse féline	FIPV	Chat	Péritonite
Coronavirus de l'entérite du chat	FECV	Chat	Entérite
Coronavirus de la dinde	TCV	Dinde	Entérite
Coronavirus du lapin	RbCV	Lapin	Entérite
Membres possibles			
Coronavirus entérique humain	HECV	Homme	Entérite
Coronavirus porcin CV-777	CV-777	Porc	Entérite
Coronavirus de l'entérite du poulain	FECV	Cheval	Entérite
Isolats SD-SK	SD, SK	Souris-Homme	Encéphalomyélite chez la souris

Table 1. 1 : Coronavirus : nom - hôte naturel et maladies prédominantes.

Définition

Les membres de la famille monogénique des Coronaviridae sont répandus dans le monde entier et infectent tous des vertébrés, oiseaux et mammifères ainsi que l'homme. Ces coronavirus sont répertoriés dans le tableau 1 avec mention de leur hôte naturel et des principales maladies qu'ils provoquent chez cet hôte. D'autres coronavirus ont été décrits qui

peut être ajoutés à cette liste comme membres possibles mais n'ont pas encore été officiellement classés dans la famille des Coronaviridae [6].

Les coronavirus sont des virus à ARN fréquents, de la famille des Coronaviridae, qui sont responsables d'infections digestives et respiratoires chez l'Homme et l'animal. Le virus doit son nom à l'apparence de ses particules virales, portant des excroissances qui évoquent une couronne. Les virions, qui sont constitués d'une capside recouverte d'une enveloppe, mesurent 80 à 150 nm de diamètre. Les petites sphères contiennent un acide ribonucléique (ARN) monocaténaire (avec une seule chaîne), linéaire et positif, comptabilisant 27 à 32 kilo bases. Cet ARN se réplique dans le cytoplasme de la cellule infectée [7].

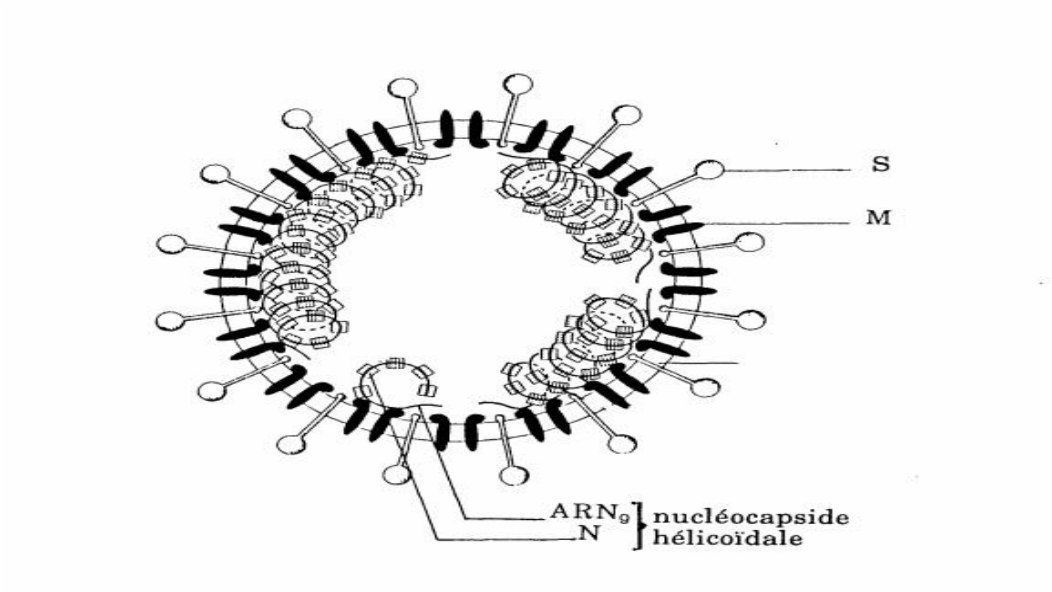


Figure 1. 3 : Structure du coronavirus(N) protéine de la nucléocapside (M) protéine de la matrice(S) protéine du péplomère.

ORGANISATION DES CORONAVIRUS

Les nombreuses études réalisées, et concernant surtout le MHV et l'IBV, ont permis d'établir un modèle structural identique pour tous les coronavirus.il comprend [6]:

- Une enveloppe virale constituée de deux types de protéines M et S, associées à des glyco-aminoglycanes, et implantées dans une double couche de lipides dont la composition reflète le type de cellules dans lequel le virus s'est répliqué.
- Une nucléocapside interne qui associe une molécule d'ARN génomique (ARNg) et la protéine de la capside N.

Structure	- virion sphérique 100 nm de diamètre - enveloppé, présence de spicules - nucléocapside hélicoïdale 10-20 nm de diamètre
Génome	- ARN monobrin de polarité positive - Mr : $5,5.10^6$ en moyenne - polyadénylé, structure cap en 5' - fonctionne comme un ARNm
Protéines structurales	- glycoprotéine du péplomère S (180 à 200 k) - glycoprotéine de la matrice M (20 à 30 k) - phosphoprotéine de la nucléocapside N (50 à 60 k)

Table 1. 2 : Caractères généraux et propriétés physico-chimiques Des coronavirus.

2.3 Qu'est-ce que la Bio-informatique ?

Domaine interdisciplinaire, situé au carrefour de l'informatique, des mathématiques et de la biologie, qui traite de l'application de l'informatique aux sciences biologiques.

La bio-informatique est un vaste domaine qui recouvre l'ensemble des utilisations de l'informatique pour la gestion, l'entreposage, l'analyse, le traitement, l'organisation, la comparaison et la diffusion de données relatives à l'ensemble des sciences biologiques (physiologie, écologie, biochimie, biologie moléculaire et, dans une large mesure génétique et génomique) [8].

2.3.1 Histoire du terme « Bio-informatique »

Le terme de bio-informatique date du début des années 80. Cependant, le concept sous-jacent de traitement de l'information biologique est bien plus vieux. Durant les années 60, la biologie moléculaire a eu besoin de modélisation formelle, ce qui a mené à la création des biomathématiques. L'apparition de la bio-informatique n'est donc pas une conséquence de la génomique (séquençage d'un génome et son interprétation), mais plutôt une de ses fondations [2].

2.3.2 Buts

Chapitre 1 : bio-informatique

La bio-informatique est l'étude de l'information biologique. Ce n'est pas simplement l'application à la biologie de l'informatique ; c'est une branche à part entière de la biologie. La bio-informatique actuelle se concentre surtout sur l'étude des séquences d'ADN et sur le repliement des protéines, donc travaille surtout au niveau moléculaire. De nombreux bios informaticiens travaillent également à l'élaboration d'outils biologiques permettant de résoudre des problèmes de l'informatique classique [2].

Chapitre 2 : Text Mining et l'apprentissage automatique

2.1 Introduction

Les textes expriment un grand nombre d'informations de natures diverses mais la manière dont cette information est représentée rend difficile l'analyse automatique. L'information n'est donc pas structurée (texte libre). Cette absence de structure n'autorise pas un accès direct aux informations. Le volume de données est très important rendant impossible toute analyse par un humain.

L'apprentissage automatique est au cœur de la science des données et de l'intelligence artificielle. Que l'on parle de transformation numérique des entreprises, de Big Data ou de stratégie nationale, l'apprentissage automatique est devenu incontournable. Ses applications sont nombreuses et variées, allant des moteurs de recherche et de la reconnaissance de caractères à la recherche en génomique, l'analyse des réseaux sociaux, la publicité ciblée, la vision par ordinateur, la traduction automatique ou encore le trading algorithmique. Ce chapitre se veut une introduction aux concepts de <<text mining>> et algorithmes qui fondent les l'apprentissage automatique.

2.2 Qu'est-ce que text mining ?

Le text mining est un processus **d'extraction de structures** (connaissances) **inconnues**, **valides** et potentiellement exploitables dans les **documents textuels**, à travers la mise en œuvre de techniques statistiques ou d'apprentissage automatique <<machine learning>>. Mais d'autres applications spécifiques aux textes sont possibles : résumé automatique, extraction d'information, etc. [9].

2.3Processus de text mining

Les étapes nécessaires pour effectuer le processus de text mining sont :

1. **Recherche de documents :** La première étape consiste donc à effectuer une simple recherche au sein des ressources disponibles, en général à partir du Web et de bases de données bibliographiques ou textuelles, pour trouver les documents ayant cette caractéristique. Le résultat de la recherche constitue le contexte de fouille [10].
2. **Structuration des données :** L'ensemble de documents à fouiller n'est pas structuré (du moins au sens informatique, car le contenu des documents possède bien sur une structure sémantique). Mais, de nombreuses méthodes, essentiellement issues du

domaine du traitement automatique des langages, permettent de pallier `à ce problème. La plus couramment utilisée consiste à rechercher, et si nécessaire à filtrer, les mots-clés ou les phrases-clés contenus dans les documents, et éventuellement les relations existantes entre ces divers éléments clés [10].

3. **Exploration des données structurées :** Cette étape repose sur une forte interaction entre l'utilisateur et le système. La machine apporte la puissance de calcul et les capacités de mémorisation. L'utilisateur est le seul à pouvoir dominer l'aspect sémantique des résultats. N'néanmoins, le système agit comme un assistant, dans le sens o u il est capable de faire des suggestions a l'utilisateur et de prendre des initiatives, tout en justifiant ses choix [10].

2.4 Applications de text mining

Cette section est consacrée à un bref exposé des applications dans différents domaines dans lesquels les méthodes du text Mining ont été une réussite.

- **Catégorisation de textes :** Une application phare est le filtrage auto de documents (ex. e-mail, pages web, etc.), mais il y en a d'autres (analyse des sentiments, opinion mining, etc.)
- **Clustering de textes :** une organisation des documents pour faciliter la consultation et la recherche, disposé d'un résumé du corpus (une forme de réduction de la dimensionnalité), etc.
- **Recherche d'information :** L'objectif de la recherche d'information est de mettre en place les stratégies permettant d'identifier, dans un corpus, les documents pertinents relatifs à un document requêtent. Il s'agit d'une recherche par le contenu, le texte est concerné, mais elle peut s'étendre à l'image, la vidéo, le son, etc.
- **Extraction d'information :** L'extraction d'information consiste recherche des champs prédéfinis dans un texte plus ou moins rédigé en langage naturel. On s'appuie plus sur l'analyse lexicale et morphosyntaxique pour identifier les zones d'intérêts [9].

2.5 text mining et bio-informatiques

La reconnaissance d'une **bio-entité** cherche à identifier et à classifier les termes techniques dans le domaine de la biologie moléculaire qui correspondent à des concepts marquant l'intérêt des biologistes. Des exemples de pareilles entités incluent les noms des protéines ou des gènes et leurs locations comme les cellules ou les noms d'organismes. La reconnaissance

d'entité est devenue de plus en plus importante à cause de la croissance massive des résultats rapportés par forte cadence de méthodes expérimentales. Elle peut être utilisée dans plusieurs fonctions **d'extraction d'informations** comme l'extraction de relation ou la synthétisation.

2.6 Qu'est-ce que l'apprentissage ?

L'apprentissage automatique (AA) ("Machine Learning") est à la croisée de plusieurs disciplines [11]:

- **Les statistiques :** pour l'inférence de modèles à partir de données.
- **Les probabilités :** pour modéliser l'aspect aléatoire inhérent aux données et au problème d'apprentissage.
- **L'intelligence artificielle :** étudier les taches simples de reconnaissance de formes que font les humains (comme la reconnaissance de chiffres par exemple), et parce qu'elle fonde une branche de l'AA dite symbolique qui repose sur la logique et la représentation des connaissances.
- **L'optimisation :** optimiser un critère de performance afin, soit d'estimer des paramètres d'un modèle, soit de déterminer la meilleure décision à prendre étant donné une instance d'un problème.
- **L'informatique :** puisqu'il s'agit de programmer des algorithmes et qu'en AA ceux-ci peuvent être de grande complexité et gourmands en termes de ressources de calcul et de mémoire.

2.6.1 Données d'Apprentissage

Les données d'entraînement sont divisées en 3 groupes, ces groupes sont [12]:

1. **L'ensemble d'apprentissage ou population d'entrainement :** constitue l'ensemble des candidats ou exemples (images, attributs, DB, …) utilisés pour générer le modèle d'apprentissage.
2. **L'ensemble de Test :** est constitué des candidats sur lesquels sera appliqué le modèle d'apprentissage (pour tester et corriger l'algorithme).
3. **L'ensemble de validation :** peut être utilisé lors de l'apprentissage (comme sous population de l'ensemble d'apprentissage) afin de valider (intégrer) le modèle et d'éviter le sur-apprentissage.

2.6.2 Types d'apprentissage

En apprentissage automatique, les modèles prédictifs utilisent divers algorithmes sous-jacents pour déduire des relations mathématiques à partir des données d'entraînement. Il existe principalement trois types de méthodes d'apprentissage, à savoir : l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage semi-supervisé. Dans la section ci-dessous, nous discuterons de chaque méthode plus en détail [13].

A) Apprentissage supervisé

Dans l'apprentissage supervisé, le modèle est alimenté par un ensemble de données d'apprentissage contenant à la fois des observations (données d'entrée) ainsi que leurs résultats correspondants (données de sortie). Le modèle déduit ensuite le mappage mathématique des entrées aux sorties qu'il peut utiliser pour classer les futurs points de données de test d'entrée [13].

B) Apprentissage non supervisé

Dans l'apprentissage non supervisé, le modèle est alimenté par des données d'entraînement non classifiées (données d'entrée uniquement). Ensuite, le modèle classe les points de données de test dans différentes classes en trouvant des points communs entre eux.

Le but de tels problèmes d'apprentissage non supervisé peut être de découvrir des groupes d'exemples similaires dans les données, où cela s'appelle le regroupement, ou de déterminer comment les données sont distribuées dans l'espace, connu sous le nom d'estimation de densité [13].

C) Apprentissage semi-supervisé

L'apprentissage semi-supervisé in hérité des propriétés de l'apprentissage supervisé et de l'apprentissage non supervisé. Un ensemble de données semi-supervisé contient principalement des points de données d'apprentissage non classifiés ainsi que de petites quantités de données classifiées [13].

Les modèles semi-supervisés présentent deux avantages importants [13]:

- Premièrement, ils sont nettement plus précis que les modèles non supervisés avec l'ajout de quelques points de données classifiés.

- Deuxièmement, ils sont nettement moins laborieux et chronophages que l'apprentissage supervisé.

2.7 Apprentissage et classification supervisés

2.7.1 Introduction

Dans cette section ci-dessous, nous discuterons de l'apprentissage automatique supervisé et expliquerons certains de ses algorithmes sous-jacents (algorithmes SVM et bayes). Un aperçu à jour sera présenté pour chacun de ces deux algorithmes.

2.7.2 Classification supervisée

La classification supervisée est l'une des techniques les plus utilisées dans l'analyse des bases de données. Elle permet d'apprendre des modèles de décision qui permettent de prédire le comportement des exemples futurs. La classification est un processus à deux étapes : une **étape d'apprentissage** (entrainement) et une **étape de classification** (utilisation) [14]:

- **Dans l'étape d'apprentissage,** un classiffieur (une fonction, un ensemble de règles, ...) est construit en analysant une base de données d'exemples d'entrainement avec leurs classes respectives.$\boldsymbol{X} = (\boldsymbol{x1}, \boldsymbol{x2}, .., \boldsymbol{xm})$ est représenté par un vecteur d'attributs de dimension $\boldsymbol{m}$. Chaque exemple est supposé appartenir à une classe prédéfinie représentée dans un attribut particulier de la base de données appelé attribut de classe. Puisque la classe de chaque exemple est donnée, cette étape est aussi connue par l'apprentissage supervisé.
- **Dans l'étape de classification**, le modèle construit dans la première étape est utilisé pour classer les nouvelles données. Mais avant de passer à l'utilisation, le modèle doit être testé pour s'assurer de sa capacité de généralisation sur les données non utilisées dans la phase d'entrainement. Le modèle obtenu peut être testé sur les données d'entrainement elles-mêmes, la précision (le taux de reconnaissance) est généralement élevée mais ne garantit pas automatiquement une bonne précision sur les nouvelles données.

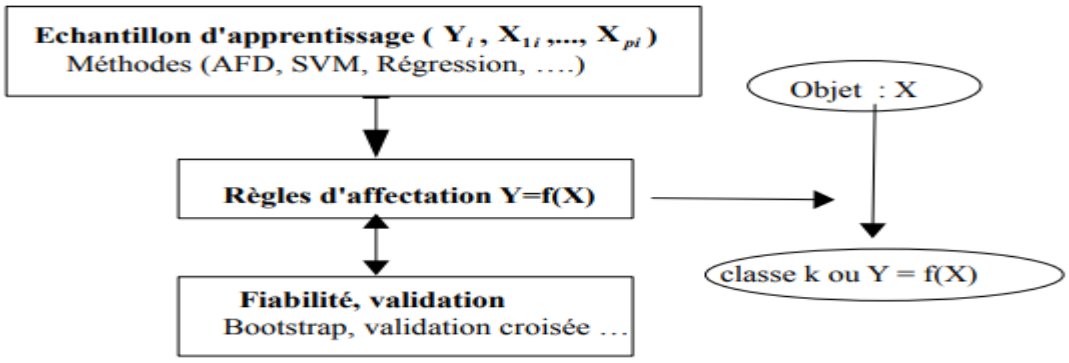


Figure 2. 1 : les étapes de Classification supervisée.

2.7.3 Problème Linéaire et Non-Linéaire

Les méthodes de classification supervisée peuvent être basées sur [14]:

- ***Hypothèses probabilistes*** (cas du classifieur naïf bayésien).
- ***Notions de proximité*** (***k*** plus proches voisins).
- ***Espaces d'hypothèses*** (arbres de décisions).

En fonction du problème, il faut pouvoir choisir le classifieur approprié, c'est-à-dire celui qui sera à même de séparer au mieux les données d'apprentissage. On dit qu'un problème est **linéairement séparable** si les exemples de classes différentes sont complètement séparables par un **hyperplan** (appelé hyperplan séparateur, ou séparatrice). Ce genre de problème se résout par des classifieurs assez simples, qui ont pour but de trouver l'équation de l'hyperplan séparateur. Mais, le problème peut également être **non séparable** de manière linéaire comme illustré dans la **figure 2.2.** Dans ce cas, il faut utiliser d'autres types de classifieurs, souvent plus longs à paramétrer, mais qui obtiennent des résultats plus précis [12].

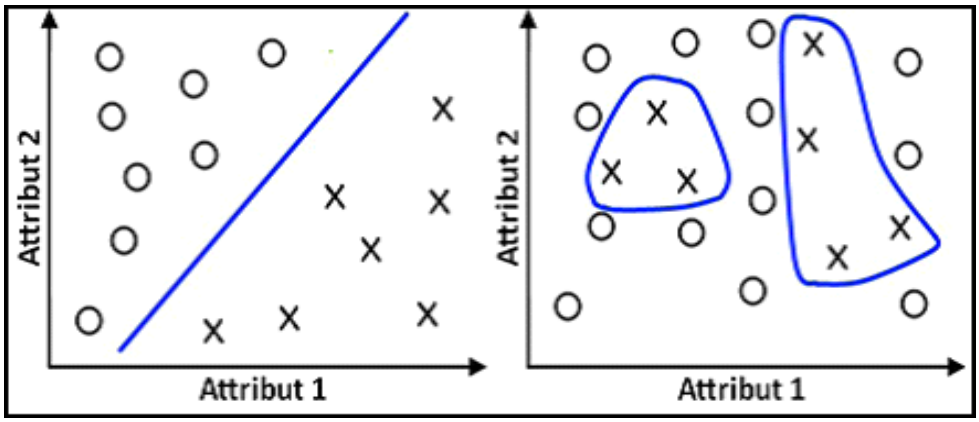


Figure 2. 2: Problème linéairement séparable (Frontière linéaire). A Droite : Problème non linéairement séparable.

2.7.4 Machine à Vecteurs de Support

B) Définition :

Les Support Vecteurs Machines souvent traduit par l'appellation de Séparateur à Vaste Marge (SVM) sont une classe d'algorithmes d'apprentissage initialement définis pour la discrimination c'est-à-dire la prévision d'une variable qualitative binaire. Ils ont été ensuite généralisés à la prévision d'une variable quantitative [15].

Le but des SVM est de trouver un classificateur qui va séparer les données et maximiser la distance entre ces deux classes. Avec SVM, ce classificateur est linéaire appelé « hyperplan » [16].

Dans le schéma qui suit, on détermine un hyperplan qui sépare les deux ensembles de points.

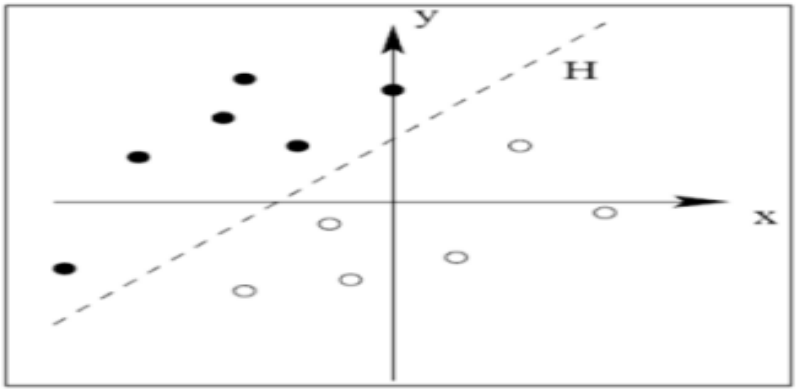


Figure 2. 3 : Séparation de deux ensembles de points par un hyperplan h.

Les points les plus proches, qui seuls sont utilisés pour la détermination de l'hyperplan, sont appelés vecteurs de support.

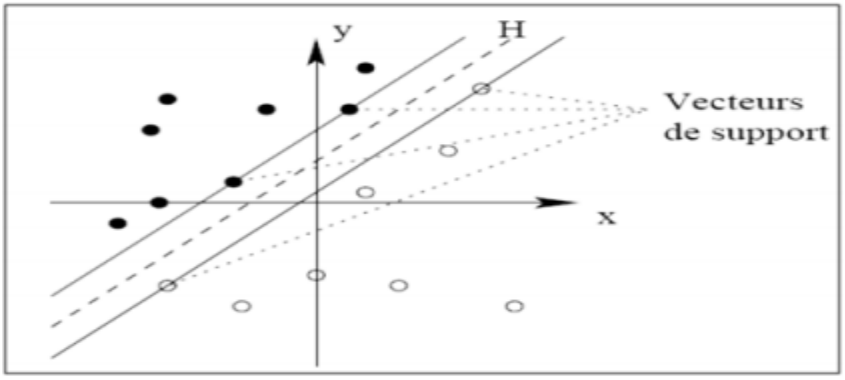


Figure 2. 4 : Vecteurs de support.

Revient à chercher un hyperplan dont la distance minimale aux exemples d'apprentissage est maximale. On appelle cette distance « **marge** » entre l'hyperplan et les exemples. L'hyperplan

séparateur optimal est celui qui **maximise la marge**. Comme on cherche à maximiser cette marge, on parlera de séparateurs à vaste marge [16].

Dans le schéma qui suit, on détermine un hyperplan qui sépare les deux ensembles de points.

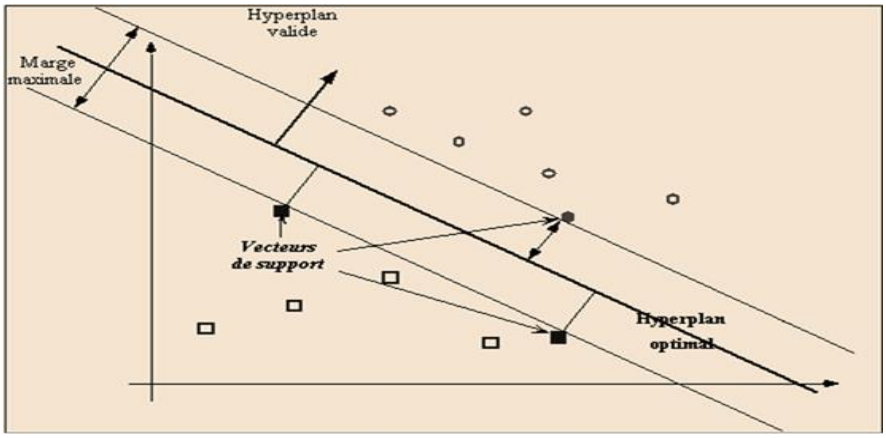


Figure 2. 5 : Hyperplan optimal, marge et vecteurs de support.

C) Pourquoi maximiser la marge ?

Intuitivement, le fait d'avoir une marge plus large procure plus de sécurité lorsque l'on classe un nouvel exemple. De plus, si l'on trouve le classificateur qui se comporte le mieux vis- a-vis des données d'apprentissage, il est clair qu'il sera aussi celui qui permettra au mieux de classer les nouveaux exemples. Dans le schéma qui suit, la partie droite nous montre qu'avec un hyperplan optimal, un nouvel exemple reste bien classé alors qu'il tombe dans la marge. On constate sur la partie gauche qu'avec une plus petite marge, l'exemple se voit mal classé [16]. La classification d'un nouvel exemple inconnu est donnée par sa position par rapport

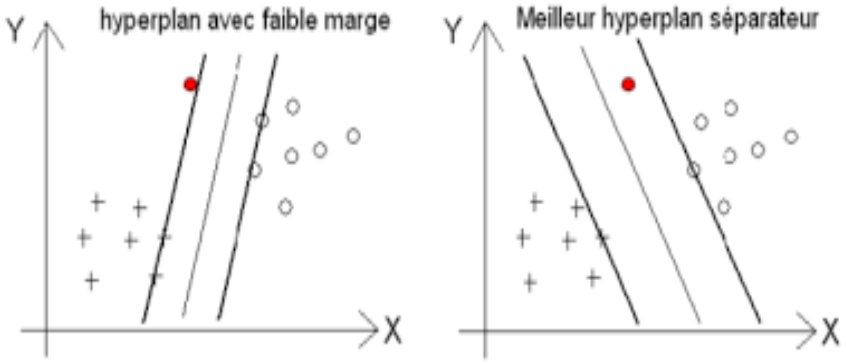


Figure 2. 6 : Maximisation de la marge.

D) données linéairement séparables :

Un problème de discrimination est dit linéairement séparable lorsqu'il existe une **fonction de décision linéaire** (appelé aussi séparateur linéaire), de la forme $D(x) = (signe\, f(x))avec\, f(x) = v^T x + a,\ v \in R^P et\ a \in R$, classant correctement toutes les observations de l'ensemble d'apprentissage($D(Xi) = Yi, i \in [1,n]$) [17].

La fonction $\boldsymbol{f}$ est appelée fonction caractéristique qui permet d'introduire de manière pédagogique les principaux principes des SVM : marge, programmation quadratique, vecteur support, formulation duale et matrice de gram. Nous allons ensuite généraliser au cas des observations non séparables et non linéaires par l'introduction de variables d'écart et de noyaux. A toute fonction de décision et donc aux fonctions de décision linéaire on peut associer une frontière de décision [17]:

$$\Delta(\boldsymbol{v}, \boldsymbol{a}) = \{\boldsymbol{x} \in \boldsymbol{R}^{\boldsymbol{p}} | \boldsymbol{v}^{\boldsymbol{T}}\boldsymbol{x} + \boldsymbol{a} = \boldsymbol{0}\}$$

E) Données non séparables linéairement :

Souvent il arrive que même si le problème est linéaire, les données sont affectées par un bruit, et les deux classes se retrouvent mélangées autour de l'hyperplan de séparation. Pour gérer ce type de problème on utilise une technique dite de marge souple, qui tolère les mauvais classements [18]:

- Rajouter des variables de relâchement des contraintes ξi.
- Pénaliser ces relâchements dans la fonction objective.

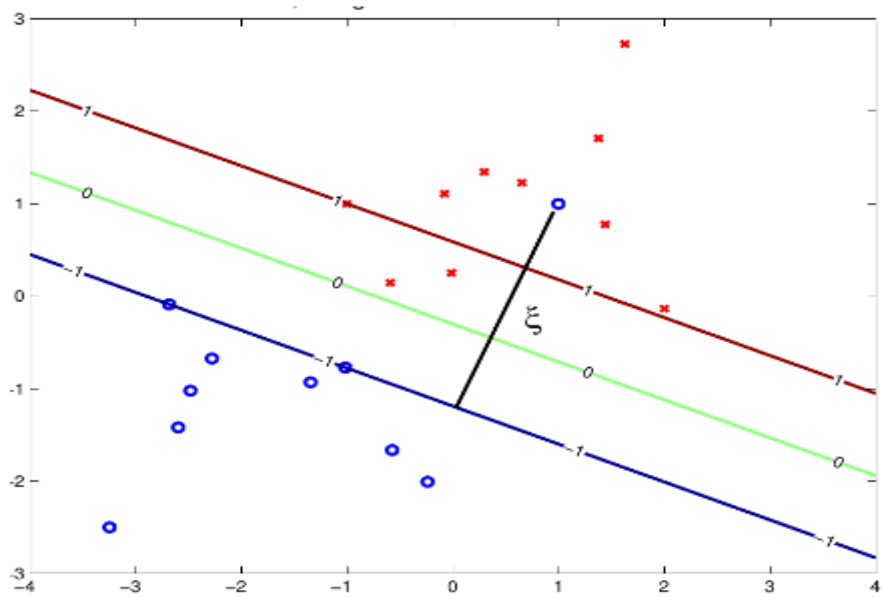


Figure 2. 7 : Plus en exemple est éloigné du mauvais côté du séparateur (point bleu), plus la variable de relâchement ξi a une valeur importante.

L'idée est de modéliser les erreurs potentielles par des variables d'écart positives ξi associées aux observations$(xi, yi), i = 1, \dots n.$ Si un point (xi, yi) vérifie la contrainte de marge $yiw^T xi + b \geq 1$ alors la variable d'écart (qui est une mesure du coût de l'erreur) est nulle [18].

Nous avons donc deux situations [18] :

- Pas d'erreur :$\boldsymbol{Y_i(W^T X_i + b) \geq 1 => \xi i = 0}$.
- Erreur : $\boldsymbol{Y_i(W^T X_i + b) < 1 => \xi i = 1 - Y_i(W^T X_i + b) > 0}$.

2.7.5 Applications des SVMs :

Plusieurs applications de classificateur SVM, parmi lesquelles [12] :

- Classification de données biologiques/physiques, Classification de documents numériques.
- Reconnaissance d'expressions faciales, Classification de textures, E-learning.
- 3) Détection d'intrusion, Reconnaissance de la parole.
- Reconnaissance d'Image Basée Contenu (CBIR : content based image retrieval).
- Bio-informatique : Comprend la classification des protéines et la classification du cancer, du virus, etc. Nous utilisons SVM pour identifier la classification des gènes, des patients sur la base de gènes et d'autres problèmes biologiques.

2.7.6 Classifieur bayésien

A) Définition :

La classification naïve bayésien est un type de classification bayésien probabiliste simple basée sur le théorème de Bayes avec une forte indépendance (dite naïve) des hypothèses. Les classifieurs bayésien sont très répandus dans le domaine de l'apprentissage machine. Nous en présentons deux parmi les mieux connus : les classifieurs bayésien naïfs et les réseaux bayésiens. Commençons par celui qui en est à la base, à savoir le classifieurs bayésien naïf. Nous présentons les réseaux bayésiens dans la section qui suit.

L'idée derrière un classifieurs bayésien naïf est de considérer que chaque réponse à une influence égale sur la **classe** que l'on cherche à prédire et que les **attributs** est indépendante entre eux. Ce n'est pas exact car les questions ne sont pas indépendantes mais cela permet en

pratique d'avoir un schéma simple qui fonctionne bien dans la plupart des cas (voir Domingos et Pazzani (1997)) [19].

Cette méthode est basée sur la règle de Bayes de probabilité conditionnelle. En considérant **l'hypothèse H (classes)** et les observations **E (attributs)**, on a [19]:

$$P_r(H|E) = \frac{P_r(E|H).P_r(H)}{P_r(E)}.$$

Pour nos besoins, **l'hypothèse H** sera le succès ou l'échec à un item dont on n'a pas encore observé la réponse, et **l'évidence E** représentera les succès ou échecs observés pour d'autres items.

B) Construire un classifieur bayésien :

Le processus de construction se déroule selon les étapes suivantes [20] :

1. On sépare l'ensemble d'entraînement en M sous-ensembles contenants chacun tous les **points x** d'une même **classe c.**
2. On entraîne un estimateur de densité sur chacun : $c \in \{1, .., m\}$. $\widehat{P_c}(x) \approx P(X = x|Y = c)$.
3. On détermine les probabilités à priori de chaque **classe C :** $\widehat{P_c} = \frac{N_c}{n} \approx P(Y = c)$ [20].
4. On applique la règle de Bayes pour obtenir la probabilité à postériori des classes au **point x.**

$$P_r(Y = c|X = x) = \frac{P_r(X = x|Y = c).P_r(Y=c)}{P_r(X=x)}.$$

$$P_r(Y = c|X = x) = \frac{P_r(X = x|Y = c).P_r(Y = c)}{\sum_{c=1}^{m} P_r(X = x \mid Y = c).P_r(Y = c)}.$$

5. On choisit la **classe c** la plus probable.

2.7.7 Applications classifieur bayésien

Comme vous avez dû le remarquer, cet algorithme offre de nombreux avantages à ses utilisateurs. C'est pourquoi il a également de nombreuses applications dans divers secteurs. Voici quelques applications de l'algorithme Naïve Bayes [21] :

- Comme cet algorithme est rapide et efficace, vous pouvez l'utiliser pour faire des prédictions en temps réel. Cet algorithme est populaire pour les prédictions multi-classes. Vous pouvez facilement trouver la probabilité de plusieurs classes cibles en utilisant cet algorithme.
- Les services de messagerie (comme Gmail) utilisent cet algorithme pour déterminer si un e-mail est un spam ou non. Cet algorithme est excellent pour le filtrage du spam.
- Son hypothèse d'indépendance des fonctionnalités et son efficacité dans la résolution de problèmes multi-classes le rendent parfait pour effectuer une analyse des sentiments. L'analyse des sentiments fait référence à l'identification des sentiments positifs ou négatifs d'un groupe cible (clients, public, etc).
- Le filtrage collaboratif et l'algorithme Naive Bayes travaillent ensemble pour créer des systèmes de recommandation. Ces systèmes utilisent l'exploration de données et l'apprentissage automatique pour prédire si l'utilisateur souhaite ou non une ressource particulière.

Chapitre 3 : Conception de système

3.1 Introduction

Dans ce chapitre nous introduire une description générale de notre système, en mettant en évidence son côté conceptuel du prétraitement qui constitue une étape fondamentale avant la réalisation de ce système. Ensuite nous détaillerons chaque étape en citant les principaux algorithmes et techniques utilisées dans chacune des étapes.

3.2 Conception général

Notre système se base sur l'utilisation de l'apprentigase automatique pour classifier entre déffirents types de coronavirus. Le système prend en entrée plusueirs bases des déffirents coronavirus types la transforme en une base de caractéristiques utilisable pour la preparation phase d'apprentissage automatique. Cette transformation est appelée prétraitement, elle efectue une série d'opérations telles que étiquettes de données, et l'encodage. La base pré-traitée est subdivisée en deux partie, une pour l'entraînement et l'autre pour la validation.

La figure suivant représente l'architerture générale pour realisation de système :

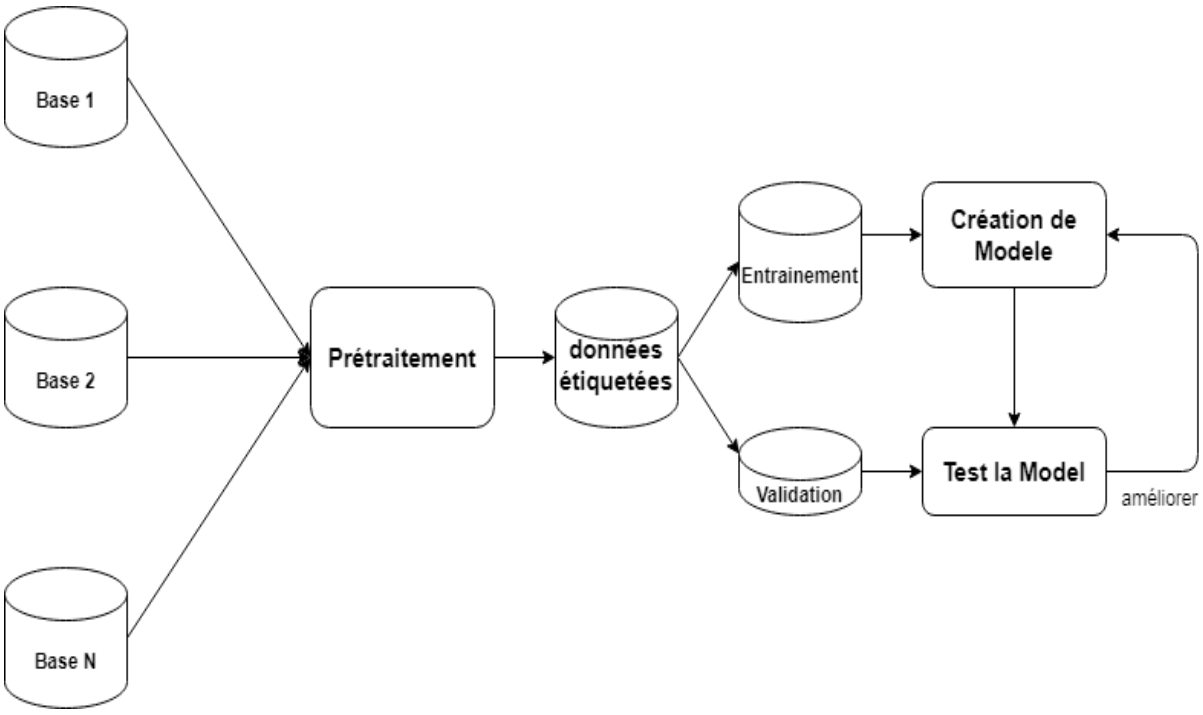


Figure 3. 1 : Architecture générale de système.

3.3 Conception détaillée

Dans cette partie, nous expliquerons les étapes les plus importantes de la mise en œuvre du système.

03.3.1 Prétraitement

Notre objectif est d'extraire les meilleures caractéristiques permettant de classifier entre défirent coronavirus types. On commence par le prétraitement des données des datasets brutes. Ces datasets qui contiennent des données textuelles (séquences ADN) seront étiquetés par une colonne appelée **Class ou label** Manuellement.

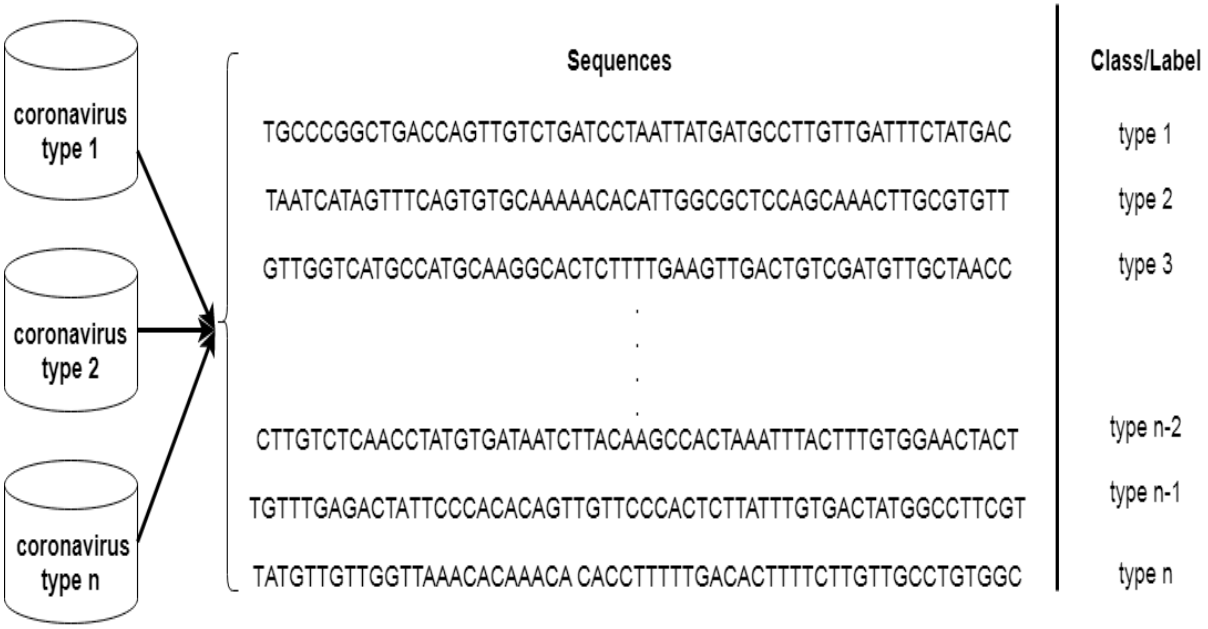


Figure 3. 2 : étiquette manuellement des différents coronavirus types.

Après avoir étiquette manuellement les données pour chaque type de coronavirus, nous passons aux étapes suivantes :

A) Méthode k-mer

Une expérience de séquençage peut être représentée par une collection de k-mers, c'est à dire l'ensemble des sous-séquences de longueur k contenues dans l'expérience. Cette approche est utilisée dans de nombreuses branches de la bio-informatique [22].

L'approche k-mer est, par exemple, très efficace dans le cadre de l'assemblage avec l'utilisation de graphes de De Bruijn pour L'assemblage de novo12-14. On la retrouve aussi dans l'alignement de séquences ou encore la détection de variantes. Par exemple, la séquence **ATCGATCAC** avec les 3-mères suivants (k-mer de taille 3) [23]:

Numéro	**0**	**1**	**2**	**3**	**4**	**5**	**6**
3-mer	**ATC**	**TCG**	**CGA**	**GAT**	**ATC**	**TCA**	**CAC**

Table 3. 1 : une représentation simple de la méthode k-mer.

Dans cette partie nous avons utilisé cette méthode pour gérer toutes les possibilités des sous-séquences. L'objectif donc est préparé ces sous-séquences pour l'étape de l'encodage.

B) Encodage

Dans cette étape, nous avons transformé les sous-séquences gérées par k-mer en un vecteur numérique, en utilisant la méthode TF-IDF. Comme suivant [24]:

1. TF-IDF (Term Frequency-Inverse Document Frequency) : est une combinaison de deux mots différents, à savoir la fréquence du terme et la fréquence du document inverse.
2. TF : Term Frequency, qui mesure la fréquence à laquelle un terme apparaît dans un document. Étant donné que chaque document est de longueur différente, il est possible qu'un terme apparaisse beaucoup plus de fois dans les documents longs que dans les documents plus courts. Ainsi, la fréquence des termes est souvent divisée par la longueur du document (c'est-à-dire le nombre total de termes dans le document) comme moyen de normalisation :

$$\boldsymbol{TF} = (\boldsymbol{Nombre\ de\ répétitions\ de\ mot\ dans\ un\ document}) / (\boldsymbol{de\ mots\ dans\ un\ document}).$$

3. IDF : Inverse Document Frequency, qui mesure l'importance d'un terme. Lors du calcul de la TF, tous les termes sont considérés d'égale importance. Cependant, il est connu que certains termes, tels que « est », « de » et « que », peuvent apparaître

beaucoup de fois mais ont peu d'importance. Ainsi, nous devons alourdir les termes fréquents tout en augmentant les rares, en calculant ce qui suit :

$$\mathbf{IDF} = \mathbf{Log}[(\mathbf{Number\ of\ documents}) / (\mathbf{Number\ of\ documents\ containing\ the\ word})]$$

4. **TF-IDF** n'est rien, mais juste la multiplication de la fréquence de terme (TF) et de la fréquence de document inverse (IDF).

L'objectif de cette technique est d'extraire les termes (sous-séquences) les plus fréquentés de chaque séquence génomique des types de coronavirus que nous avons utilisés et. Puis alimentez les sous-séquences extraites de notre modèle d'entraînement à l'étape suivante (apprentissage automatique).

3.3.2 Apprentissage Automatique

Dans cette étape, nous utilisons un ensemble d'entraînement pour construire un modèle décisionnel. L'ensemble de données a été séparé en deux sous-ensembles de données. Le premier sous-ensemble de données utilisé pour fournir le modèle de décision et le deuxième sous-ensemble de données utilisé pour évaluer la performance de ce modèle de décision.

A) Processus d'entrainement : pour former votre modèle, nous avons déjà utilisé deux algorithmes que nous avons mentionnés dans le chapitre précédent, les classifieurs bayésien multinomial et SVM.

Bayésien multinomial : nous avons utilisé ce classificateur, car il est compatible avec le terme fréquence (le classificateur multinomial utilise le terme fréquence), ce qui nous donne le meilleur score dans notre problème de classification. Ce classificateur utilisé pour les problèmes de classification binaire et multi, comme nous l'avons dans notre cas.

B) Performances du modèle : Après avoir construit notre modèle de décision, nous évaluerons les performances de ce modèle construit de prise de décision de classification à l'aide de différentes métriques d'évaluation. Parmi ces métriques :

Le tableau suivant montre un exemple de format de matrice de confusion avec **N** classes (c'est-à-dire que **N** représente le nombre de types de coronavirus) :

		Predicted Number			
		Class 1	Class 2	…	Class *n*
Actual Number	Class 1	x_{11}	x_{12}	…	x_{1n}
	Class 2	x_{21}	x_{22}	…	x_{2n}
	. . .	. . .	. . .	. . .	. . .
	Class *n*	x_{n1}	x_{n2}	…	x_{nn}

Table 3. 2 : Une représentation simple de la matrice de confusion.

Pour chaque classe $\boldsymbol{i}$ représentée danscette matrice de confusion on a :

$$\boldsymbol{FN_{de\ classe(i)} = \sum_{j=1}^{N} X_{ij} \quad et\ j \neq i} \qquad (1)$$

$$\boldsymbol{FP_{de\ classe(i)} = \sum_{j=1}^{N} X_{ji} \quad et\ j \neq i} \qquad (2)$$

$$\boldsymbol{VN_{de\ classe(i)} = \sum_{j=1}^{N} \sum_{k=1}^{N} X_{jk} \quad et\ j \neq i\ \ et\ k \neq i} \qquad (3)$$

$$\boldsymbol{VP_{tous\ classes} = \sum_{j=1}^{N} x_{jj}} \ (4)$$

Tel que :

-VP (Vrai Positive) : les exemples de classe **i en Actual Number** classés correctement dans classe **i en Predicted Number**.

-VN (Vrai Négative) : les exemples sont hors classe **i en Actual Number** classés correctement dans le hors classe **i en Predicted Number**.

- FP (Faux Positive) : les exemples de classe **i en Actual Number** classés incorrectement dans le hors classe **i en Predicted Number.**

- FN (Faux Négative) : les exemples sont hors classe **i en Actual Number** classés incorrectement dans le classe **i en Predicted Number.**

–Précision : proportion des exemples positifs correctement classés dans l'ensemble des exemples.

$$Précision_{classe(i)} = \frac{\boldsymbol{VP_{tous\ classes}}}{\boldsymbol{VP_{tous\ classes} + FP_{de\ classe(i)}}}$$

–Rappel : proportion des exemples de classe **i en Actual Number** correctement classés dans classe **i en Predicted Number** par rapport les exemples classés incorrectement hors classe **i en Predicted Number**.

$$Précision_{classe(i)} = \frac{\boldsymbol{VP_{tous\ classes}}}{\boldsymbol{VP_{tous\ classes} + FN_{de\ classe(i)}}}$$

– F1 score : proportion des exemples hors de classe **i en Actual Number** correctement classés hors classe **i en Predicted Number** par rapport les exemples classés incorrectement dans classe **i en Predicted Number**.

$$Précision_{classe(i)} = \frac{\boldsymbol{VN_{tous\ classes}}}{\boldsymbol{VN_{tous\ classes} + FP_{de\ classe(i)}}}$$

– Exactitude (précision globale) : proportion des exemples dans chaque de classe **i en Actual Number** correctement classés dans chaque classe **i en Predicted Number** par rapport les tous les exemples donnés.

$$Précision_{classe(i)} = \frac{\boldsymbol{VP_{tous\ classes}}}{\boldsymbol{nombre\ total\ d'entrées\ de\ test}}$$

3.3. Validation

Une méthode générale pour estimer le risque espéré est celle des données de test ou de l'échantillon-test (appelée aussi estimation out-of-sample) [25]:

- L'ensemble de données disponibles DN est partitionné en deux ensembles mutuellement exclusifs par sélection aléatoire, les données d'apprentissage A (par ex.

env. 70% du nombre total) et les données de validation $\boldsymbol{V}$ (par ex. 30% du nombre total) [25].

- L'apprentissage du modèle est réalisé sur les données de l'ensemble A, en utilisant une des approches mentionnées (Bayésien ou SVM).
- Le risque espéré du modèle résultant est estimé sur les données de $\boldsymbol{V}$.

Nous avons Proposé holdout méthode pour la validation de ce modèle.

Holdout méthode : la base de données de taille N est subdivisé en deux partie, la première généralement de 70% ou plus utilisé pour l'apprentissage, et la deuxième de 30% ou moins utiliser pour le test.

3.4 Conclusion

Dans Ce chapitre, nous avons introduire la conception de notre système. On a présenté la démarche suivie dans ses différentes étapes. Dans le chapitre suivant, nous allons implémenter le fonctionnement de notre application mettant en œuvre le système proposé.

Chapitre 4 : Implémentation du Système

4.1 Introduction

Dans ce chapitre, nous présenterons notre interface graphique pour la classification des types de coronavirus multiples. L'objectif de ce chapitre est de présenter les outils utilisés (par exemple source des données, langage de programmation, etc.) pour développer cette interface et d'évaluer nos modèles décisionnels créés après le processus d'entrainement.

4.2 Les outils utilisés

Dans cette section, nous donnons tous les outils que nous avons utilisés pour réaliser notre projet. Parmi ceux-ci :

4.2.1 Source des Données

Nous avons utilisé trois bases de données qui sont les suivantes :

Type 1 : Human coronavirus NL63 isolate HCoV NL63/Haiti-1/2015, complete genome.

Les données : sont disponibles ici: https://www.ncbi.nlm.nih.gov/nuccore/KT266906.1

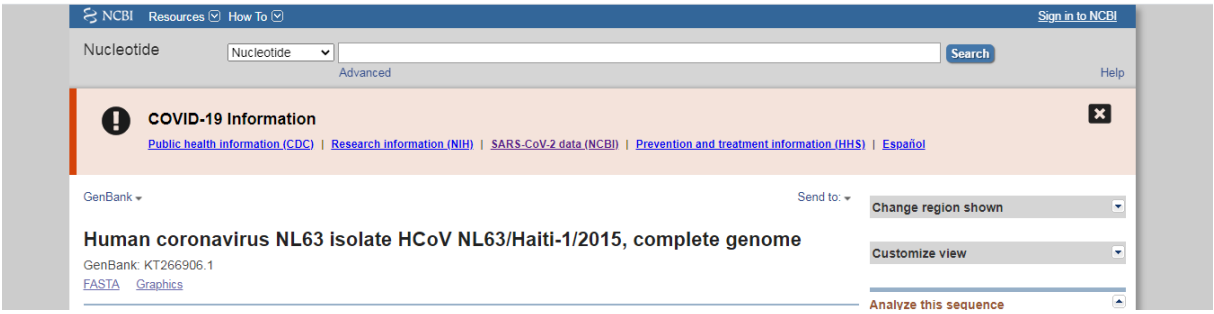


Figure 4. 1 : Le coronavirus humain isole la source des données du HCoV.

Type 2 : Middle East respiratory syndrome-related coronavirus isolate HCoV-EMC/2012, complete genome.

Les données : sont disponibles ici: https://www.ncbi.nlm.nih.gov/nuccore/667489388

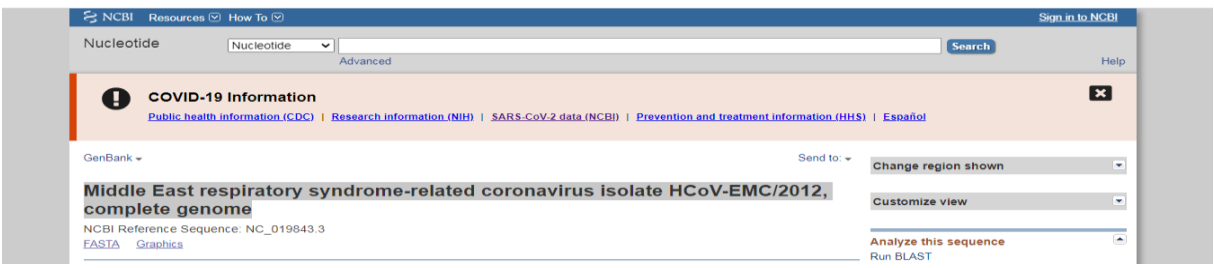


Figure 4. 2 : Le coronavirus humain isole la source des données du MERS

Type 3 : Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1, complete genome.

Les données : sont disponibles ici: https://www.ncbi.nlm.nih.gov/nuccore/NC_045512

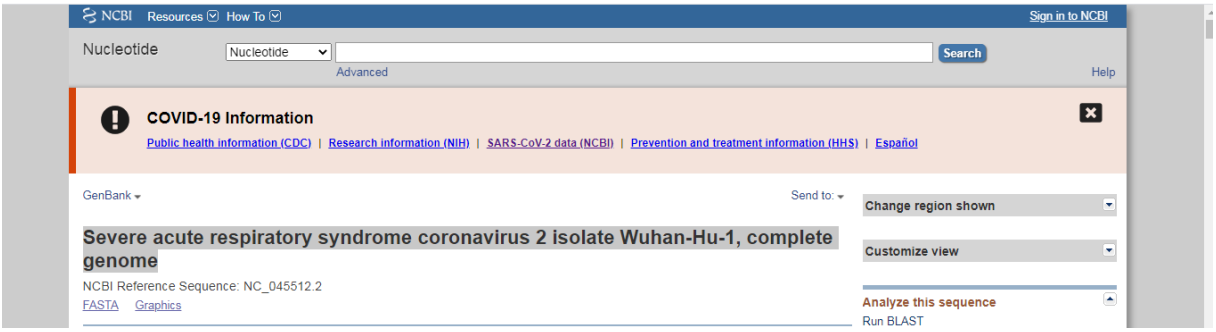


Figure 4. 3 : Le coronavirus humain isole la source des données du SARS_COV.

4.2.2 Langage de Programmation

Comme langage de développement, nous utilisons **python version 3.8** avec **pycharm** comme éditeur de code python et **anaconda navigator** comme fournisseur de bibliothèques python.

Python :

Python est un langage de programmation polyvalent, et il est applicable à peu près partout qui utilise des données, des calculs mathématiques ou des lignes de code. Cela signifie que contrairement à Java, par exemple, Python ne se limite pas à être utilisé pour le développement Web. Comme la plupart des langages de programmation, Python fonctionne en tandem avec un interpréteur qui exécute les lignes de codes finalisées. Il existe de nombreuses ressources gratuites pour apprendre le langage de codage Python, qui, avec sa base dans la syntaxe anglaise, est considéré comme l'un des langages de codage les moins difficiles et les plus simples à apprendre et à lire [26].

Python fournit de nombreuses bibliothèques prenant en charge le domaine bio-informatique tel que Bio-python, Pybio, etc. Cet avantage rend notre travail beaucoup plus facile pour réaliser notre projet.

Pycharm :

L'éditeur de code prend en charge le langage python pour construire notre projet.

Anaconda Navigator :

Anaconda navigator est un interpréteur python. Cet interpréteur permet de télécharger et de lier des bibliothèques python dans l'éditeur de code pycharm pour développer notre projet.

4.2.2 Les bibliothèques utilisées

Dans cette section, nous présentons uniquement les bibliothèques principales que nous avons utilisées dans notre projet, telles que :

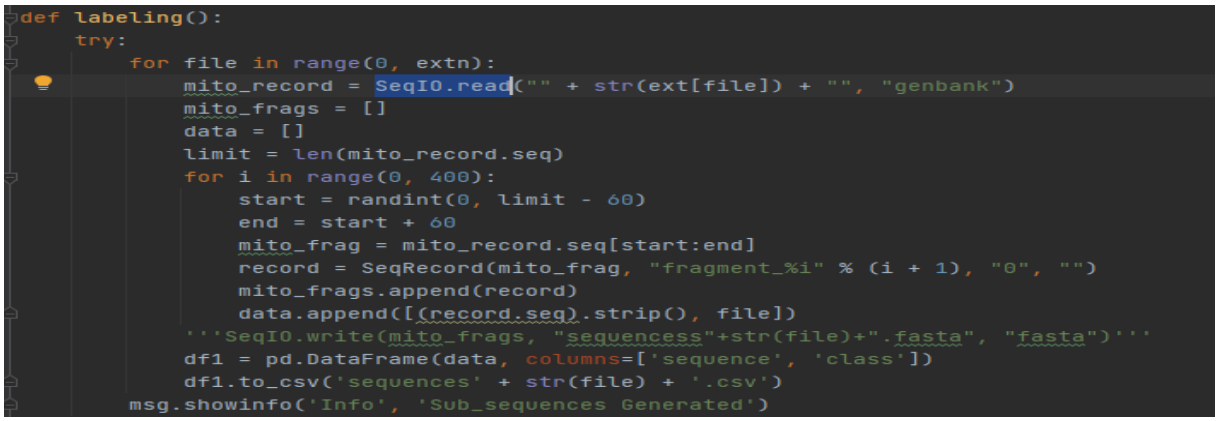
A) Bibliothèques OS :

L'objectif de cette bibliothèque est de charger nos trois bases de données (trois types du coronavirus sars_cov2, mers, et hcov).

B) Bibliothèques BIO :

L'objectif de cette bibliothèque est de lire les séquences génomiques de trois ensembles de données puis de générer des sous-séquences à partir de chaque ensemble de données et de les enregistrer sous forme de fichier ***.csv** (étiquetage manuel à notre ensemble de données) contenant la sous-séquence et la classe à laquelle elle appartient.

La figure suivante représente l'étiquetage manuel pour nos trois ensembles de données. La fonction **SeqIO.read()** est importée de la bibliothèque Bio et permet de lire chaque jeu de données pour générer des sous-séquences.



```
def labeling():
    try:
        for file in range(0, extn):
            mito_record = SeqIO.read("" + str(ext[file]) + "", "genbank")
            mito_frags = []
            data = []
            limit = len(mito_record.seq)
            for i in range(0, 400):
                start = randint(0, limit - 60)
                end = start + 60
                mito_frag = mito_record.seq[start:end]
                record = SeqRecord(mito_frag, "fragment_%i" % (i + 1), "0", "")
                mito_frags.append(record)
                data.append([(record.seq).strip(), file])
            '''SeqIO.write(mito_frags, "sequencess"+str(file)+".fasta", "fasta")'''
            df1 = pd.DataFrame(data, columns=['sequence', 'class'])
            df1.to_csv('sequences' + str(file) + '.csv')
        msg.showinfo('Info', 'Sub_sequences Generated')
```

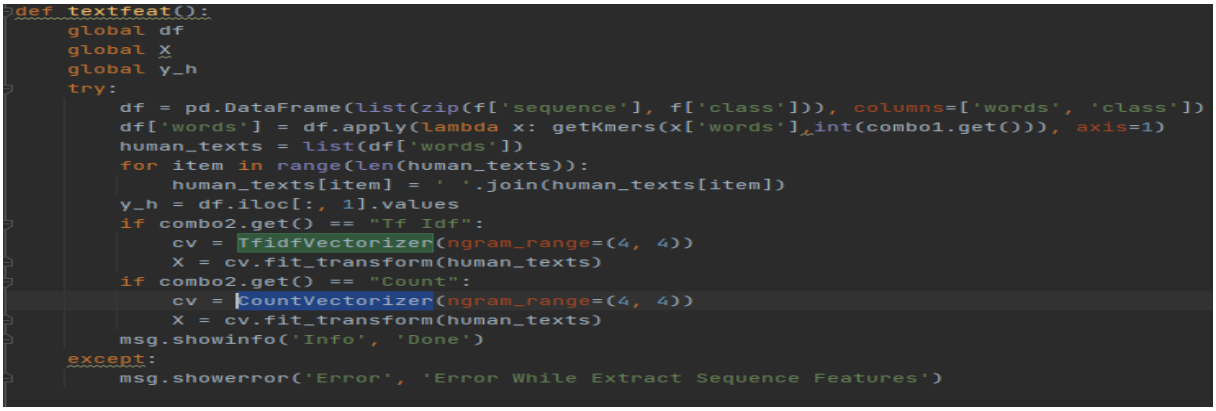
Figure 4. 4 : fonction d'étiqueter nos trois ensembles de données (générer des sous-séquences à chaque type (classe) lui appartiennent).

C) Bibliothèques SICKIT-LEARN :

C'est la bibliothèque principale qui représente le code majeur. Nous utilisons cette bibliothèque dans :

1. **Prétraitement :** nous avons utilisé cette bibliothèque pour extraire les caractéristiques de l'ADN des sous-séquences en définissant les fonctions **TfidfVectorizer** et **CountVectorizer** (tf-idf sans normalisation), comme nous l'avons mentionné dans le chapitre précédent.
2. **Apprentissage :** nous avons utilisé cette bibliothèque pour construire votre modèle de décision en utilisant l'un des algorithmes **multinomiaux bayes** ou **SVM**.

Les figures suivantes représentent la méthode d'extraction des caractéristiques de l'ADN et les algorithmes du modèle de décision :



```
def textfeat():
    global df
    global X
    global y_h
    try:
        df = pd.DataFrame(list(zip(f['sequence'], f['class'])), columns=['words', 'class'])
        df['words'] = df.apply(lambda x: getKmers(x['words'],int(combo1.get())), axis=1)
        human_texts = list(df['words'])
        for item in range(len(human_texts)):
            human_texts[item] = ' '.join(human_texts[item])
        y_h = df.iloc[:, 1].values
        if combo2.get() == "Tf Idf":
            cv = TfidfVectorizer(ngram_range=(4, 4))
            X = cv.fit_transform(human_texts)
        if combo2.get() == "Count":
            cv = CountVectorizer(ngram_range=(4, 4))
            X = cv.fit_transform(human_texts)
        msg.showinfo('Info', 'Done')
    except:
        msg.showerror('Error', 'Error While Extract Sequence Features')
```

Figure 4. 5 : le code python représente la technique d'extraction des caractéristiques des sous-séquences d'ADN.

```
        if combo3.get() == "SVM RBF":
            svm_kernel = svm.SVC(kernel='rbf')
            svm_kernel.fit(X_train, y_train)
            y_pred = svm_kernel.predict(X_test)
            accuracy = accuracy_score(y_test, y_pred)
            label.config(text=str(accuracy*100)[:5])
            y_preds = svm_kernel.predict(X_train)
            label1.config(text=str(accuracy_score(y_train, y_preds))[:5])
            f1 = f1_score(y_test, y_pred, average='macro')
            label2.config(text=str(f1 * 100)[:5])
            recall = recall_score(y_test, y_pred, average='macro')
            label3.config(text=str(recall * 100)[:5])
        if combo3.get()=="NAIVE BAYS":
            classifier = MultinomialNB(alpha=0.1)
            classifier.fit(X_train, y_train)
            y_pred = classifier.predict(X_test)
            accuracy = accuracy_score(y_test, y_pred)
            label.config(text=str(accuracy*100)[:5])
            y_preds = classifier.predict(X_train)
            label1.config(text=str(accuracy_score(y_train,y_preds))[:5])
            f1 = f1_score(y_test, y_pred, average='macro')
            label2.config(text=str(f1 * 100)[:5])
```

Figure 4. 6 : Le code python représente le modèle de décision de construction à l'aide de l'algorithme Multinomial Bayes ou SVM (avec noyau rbf).

D) Bibliothèques TKINTER :

Cette bibliothèque utilisée pour développer notre projet d'interface utilisateur graphique et pour contrôler les différents paramètres du modèle de décision.

4.3 Application

Notre projet contenant une interface utilisateur graphique principale. Cette interface graphique principale contenant toutes les étapes du modèle de construction (prétraitement, Apprentissage, etc.).

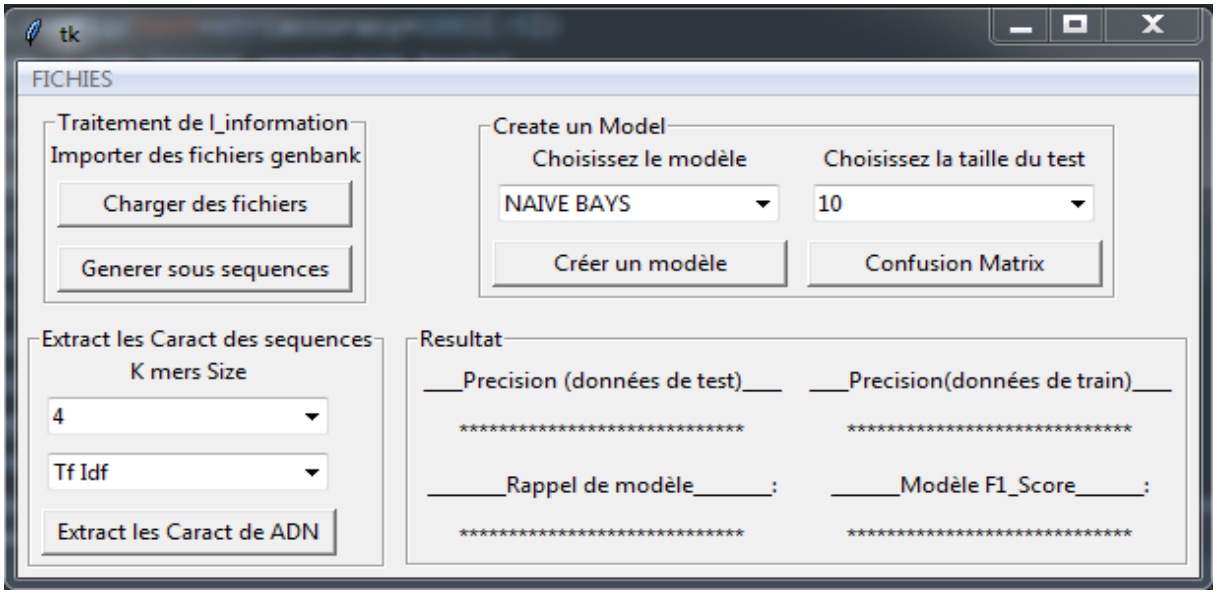
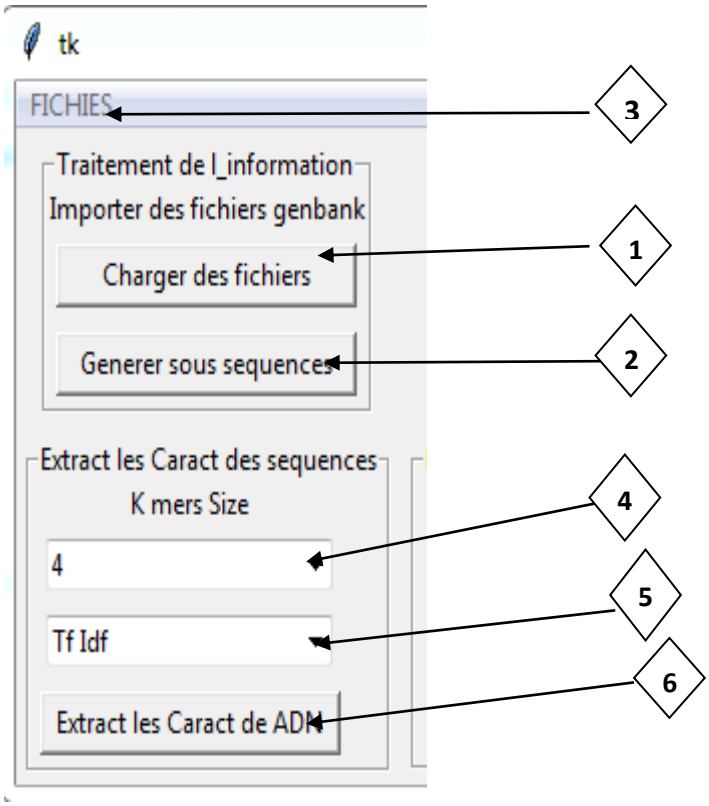


Figure 4. 7 : interface principale du projet.

4.3.1 Prétraitement

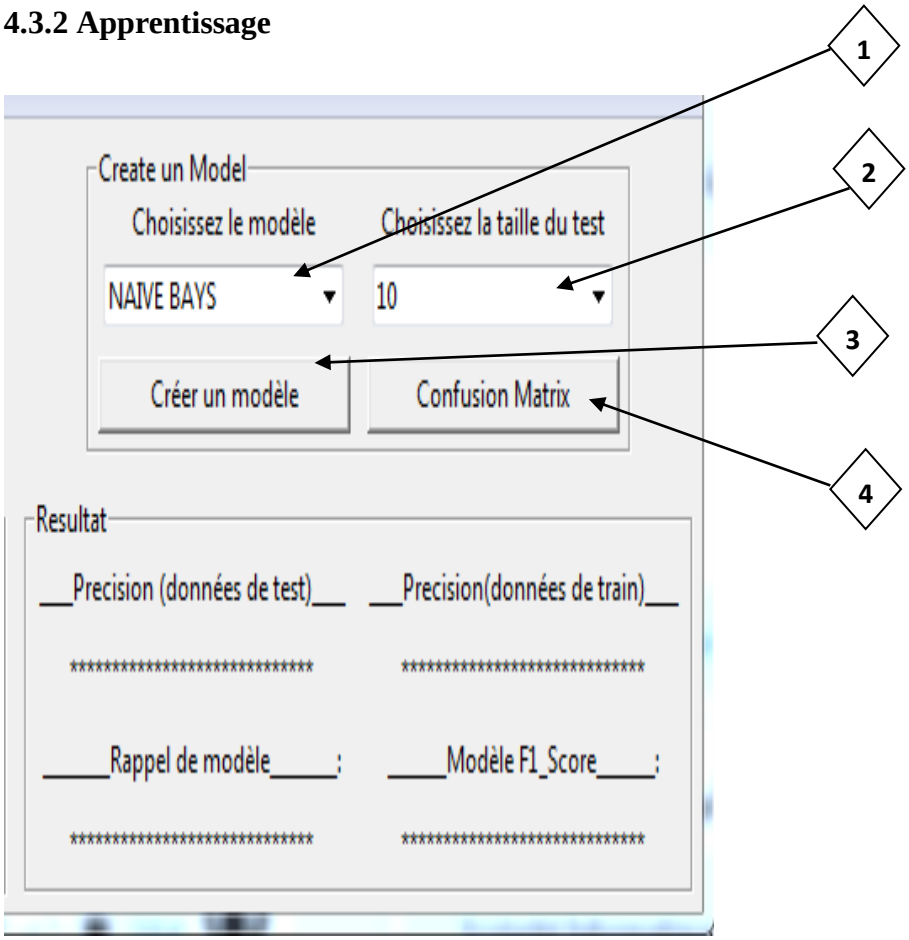


Pour le prétraitement des données, nous respectons les étapes suivantes qui sont :

1. Chargez les fichiers genbank (***.gb**) que nous utiliserons pour classer (dans notre cas trois fichiers sont chargés sars-cov 2, mers, h-cov).
2. Cliquez sur le bouton générer sous-séquences.
3. Après cela, il créera un fichier de type csv (***.csv**). Nous cliquons dans les fichiers (numéro 3) puis import des fichiers csv pour charger notre jeu de données généré.
4. Choisissez la taille k-mer.
5. Choisissez **Tf Idf** ou **Count** dans la boîte de dépôt.
6. Le clic sur **Extract les Caract d'ADN** bouton pour la dernière étape du prétraitement des données.

Notons que **Count** est **Tf-Idf** sans normalisation des valeurs acquises. Nous l'ajoutons car cela donne plus de précision que **Tf-Idf** dans nos scores de modèle.

4.3.2 Apprentissage



Pour l'apprentissage (créer un modèle de décision), nous respectons les étapes suivantes qui sont :

1. Choisissez l'algorithme que nous construisons pour notre modèle (NAIVE BAYES OU SVM RBF).
2. Choisissez la taille du test de validation (10%, 20%, ou 30%) pour évaluer les performances de notre modèle.
3. Cliquez sur **Créer un modèle**, après quoi votre modèle sera créé et évalué (les mesures d'évaluation s'afficheront dans le panneau de **résultat**).
4. Après cela, nous pouvons mettre votre matrice de confusion pour réaliser des extrapolations sur les résultats obtenus pour notre modèle de décision (cliquez sur **Confusion matrix** bouton).

4.3.3 Résultat

Nous avons déjà utilisé la classification binaire entre**sars-cov2** et **mers-cov**. Les résultats sont donnés dans le tableau suivant :

Modele utiliser	Vetorisation	Précision (base de test)	Rappel	F1-score
SVM RBF	CountVectorizer	81,87%	81,66%	82,38%
	CountVectorize	81,25%	80,81%	81,92%
SVM RBF	TF-IDFVectorizer	82,50%	82,27%	83,03%
	TF-IDFVectorize	81,25%	80,81%	81,92%
Multinomial	CountVectorizer	81%	80,56%	80,66%
	CountVectorize	81,87%	80,96%	80,16%
Multinomial	TF-IDFVectorizer	81%	80,56%	80,66%
	TF-IDFVectorize	81,87%	80,96%	80,16%

Table 4. 1 : Une comparaison entre divers résultats de métriques de classification binaire (Sars_Co2, Mers).

Nous avons utilisé une classification multiple entre sars-cov2, mers-cov, et hcov. Les résultats sont donnés dans le tableau suivant :

Modele utiliser	Vetorisation	Précision (base de test)	Rappel	F1-score
SVM RBF	CountVectorizer	75,41%	75,35%	74,31%
	CountVectorize	72,08%	72,02%	70,78%
SVM RBF	TF-IDFVectorizer	75,41%	75,35%	74,31%
	TF-IDFVectorize	72,08%	72,02%	70,78%
Multinomial	CountVectorizer	79%	79,08%	78,88%
	CountVectorize	72,91%	72,97%	71,70%
Multinomial	TF-IDFVectorizer	79%	79,08%	78,88%
	TF-IDFVectorize	72,91%	72,97%	71,70%

Table 4. 2 : Une comparaison entre divers résultats de métriques de classification multiple.

Les résultats montrent que les deux classificateurs (classificateur multinomial bayes et svm) sont très proches avec des différences insignifiantes dans la classification binaire et multiple. Les résultats montrent que les deux classificateurs sont plus performants en classification binaire que leurs performances en classification multiple.

4.5 Conclusion

Un nouveau cadre d'apprentissage automatique pour la classification des différents types de coronavirus à partir de séquences d'ADN génomiques a été signalé. Dans ce travail, nous avons implémenté Multinomial bayes et SVM à la pointe de la technologie pour classifier automatiquement tous les types de coronavirus de manière célèbre. Ces deux modèles reçoivent en entrée un texte contenant des sous-séquences pour chaque type de coronavirus et donnent une classification en sortie. Les résultats démontrent que les deux modèles fonctionnent très bien en classification binaire.

Conclusion Général

Dans ce travail, nous avons utilisé l'apprentissage automatique, en particulier les algorithmes bayes multinomiaux et SVM, pour classer automatiquement entre plusieurs types de coronavirus. Sur le plan théorique, nous avons introduit la technique du Machine Learning dans le domaine de la classification en général.

Nous avons expliqué les composants des algorithmes multinomiaux et SVM. Le rôle de chacun de ces composants dans le processus de classification Sur le plan expérimental, nous avons mis en œuvre ces deux algorithmes en utilisant diverses métriques d'évaluation de classification pour classer automatiquement entre plusieurs types de coronavirus.

Nous avons remarqué que la quantité de données utilisées pour construire la structure et certains paramètres jouaient un rôle important dans la modification des performances du modèle de décision pour la classification. Étant donné que les données utilisées dans ce travail sont largement disponibles, cela a permis à notre modèle construit d'obtenir des résultats satisfaisants en utilisant diverses métriques d'évaluation de la classification.

Les résultats obtenus dans ce travail représentent des perspectives prometteuses pour la possibilité d'utiliser l'apprentissage automatique pour aider à une classification objective de la maladie à coronavirus.

Bibliographie

[1] **F. R. Z. B. M. S. M. U. G. K. I. A. K. Hafiz Abdul Rehman, «Comprehensive comparative genomic and microsatellite analysis of SARS, MERS, BAT-SARS, and COVID-19 coronaviruses,» 30 Mar 2021. Available: https://onlinelibrary.wiley.com/doi/10.1002/jmv.26974.**

[2] **M. S. H. Damien Imbs, «Bioinformatique,» de Université de Nice Sophia Antipolis, Available: http://deptinfo.unice.fr/twiki/pub/Linfo/SuiviDesTE/rapport-imbsd-sayedham.pdf.**

[3] **«cellule latin cellula diminutif de cella chambre - LAROUSSE,» en LA CELLULE, UNITÉ DE BASE DES ÊTRES VIVANTS, Available: https://www.larousse.fr/encyclopedie/divers/cellule/31685.**

[4] **«QU'EST-CE QU'UN VIRUS ?,» en STRUCTURE ET DEFINITION DES VIRUS, Available: http://www.microbes-edu.org/etudiant/virus.html.**

[5] **françois jacob, «Virus - Définition et Explications,» en techno-science.net, Available: https://www.techno-science.net/glossaire-definition/Virus.html.**

[6] **Jocelyne Collomb, «Approche moléculaire et physico-chimique de la détection du coronavirus entérique bovin dans l'environnement,» 29 Mar 2018. Available: https://hal.univ-lorraine.fr/tel-01747168/document.**

[7] **«Coronavirus,» .en futura santé, Available: https://www.futura-sciences.com/sante/definitions/medecine-coronavirus-13502/.**

[8] **Office québécois de la langue française, 2005, «Bio-informatique,» 2005. Available: http://gdt.oqlf.gouv.qc.ca/ficheOqlf.aspx?Id_Fiche=8354234.**

[9] Ricco Rakotomalala, «Introduction à la fouille de textes,» Available: https://eric.univ-lyon2.fr/~ricco/cours/slides/TM.A%20-%20introduction%20text%20mining.pdf.

[10] D. D. K. D. N. G. M. T. M. Z. Jérome Champavere, «Text Mining,» 15 Nov 2005. Available: https://perso.liris.cnrs.fr/alain.mille/enseignements/master_ia/Alain/exposes_2005/text_mining.pdf.

[11] Julien Ah-Pine, «Apprentissage automatique,» 20019/2020. Available: https://eric.univ-lyon2.fr/~jahpine/cours/m2_dm-ml/cm.pdf.

[12] Mokhtar TAFFAR, «INITIATION A L APPRENTISSAGE AUTOMATIQUE,» Available: https://docplayer.fr/87678390-Initiation-a-l-apprentissage-automatique.html.

[13] S.-M. C. Witold Pedrycz, Deep Learning: Concepts and Architectures, 29 Oct 2019.

[14] Abdelhamid Djeffal, «Chapitre 3 Classification,» Available: http://www.abdelhamid-djeffal.net/web_documents/classification-conceptsdebase.pdf.

[15] Philippe Besse, «Machines à vecteurs supports,» Available: https://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-m-app-svm-old.pdf.

[16] F. B. Mohamadally Hasan, «SVM : Machines à Vecteurs de Support ou Séparateurs à Vastes Marges,» 16 Jan 2006. Available: http://georges.gardarin.free.fr/Surveys_DM/Survey_SVM.pdf.

[17] phillipe besse, «Machines à vecteurs supports(nouvel version),» Available:

https://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-m-app-svm.pdf.

[18] Lotfi CHAARI, «SVM,» Available: http://lotfi-chaari.net/ens/RCP209/svm.pdf.

[19] Alexandre Spaeth, «SÉLECTION DE QUESTION ET CHOIX DE CLASSIFICATEUR POUR QUESTIONNAIRES ADAPTATIFS,» Aout 2009. Available: https://publications.polymtl.ca/141/1/2009_AlexandreSpaeth.pdf.

[20] Pascal Vincent, «Régression multiple et classifieur multiclasse. Rappels de proba. Classifieur de Bayes. Classifieur de Bayes Naïf,» . en Fondements de l'apprentissage machine, Available: https://www.iro.umontreal.ca/~vincentp/ift3395/cours/5_classifieur_bayes.pdf.

[21] Pavan Vadapalli, «Naive Bayes Classifier: Pros & Cons, Applications & Types Explained Applications & Types Explained,» 11 Dec 2020. Available: https://www.upgrad.com/blog/naive-bayes-classifier/#Applications_of_Naive_Bayes_Algorithm.

[22] Téo Lemane, «Search engine for genomic sequencing data,» en Bio-informatique [q-bio.QM]., 13 Dec 2019. Available: https://hal.inria.fr/hal-02410102/document.

[23] Bernardo J. Clavijo, «k-mer counting, part I: Introduction,» 17 Sep 2018. Available: https://bioinfologics.github.io/post/2018/09/17/k-mer-counting-part-i-introduction/.

[24] R. A. Shahzad Qaiser, «Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents,» . en International Journal of Computer Applications, 01 Jul 2018. Available: https://www.researchgate.net/publication/326425709_Text_Mining_Use_of_TF-IDF_to_Examine_the_Relevance_of_Words_to_Documents.

[25] cedric.cnamfr, «Cours - Introduction à l'apprentissage supervisé,» Available: http://cedric.cnam.fr/vertigo/cours/ml2/coursIntroductionApprentissageSupervise.html.

[26] Emma Witman, «What is Python? The popular, scalable programming language, explained,» 25 Jun 2021. Available: https://www.businessinsider.fr/us/what-is-python.

[27] «Définition | Virus,» en futura santé, Available: https://www.futura-sciences.com/sante/definitions/medecine-virus-291/.

[28] Jamal Kharroubi, «Etude de techniques de classement "Machines à vecteurs supports" pour la vérification automatique du locuteur,» . pour la vérification automatique du locuteur, 2002. Available: https://pastel.archives-ouvertes.fr/pastel-00001124/document.