Prédiction basée sur des symtoms.

Université Alioune Diop de Bambey

.

# Description du projet

Les besoins en matière d'information de santé modifient également le comportement de recherche d'information et cela peut être observé partout dans le monde. En effet, ces besoins, c'est-à-dire les types et la quantité d'informations sur la santé que les gens recherchent, évoluent. Ces changements influencent la manière dont les individus cherchent et accèdent à ces informations. Ce phénomène est global, affectant des populations à travers différents pays et cultures.

De plus, de nombreuses personnes rencontrent des défis lorsqu'elles recherchent en ligne des informations sur la santé concernant les maladies, les diagnostics et les différents traitements. Beaucoup de personnes rencontrent des difficultés lorsqu'elles cherchent des informations médicales sur Internet. Ces informations incluent des détails sur diverses maladies, leurs symptômes, les méthodes de diagnostic disponibles et les options de traitement. Les défis comprennent la véracité des sources, la compréhension des termes médicaux complexes et l'évaluation de la pertinence des informations trouvées.

Dans ce contexte, si un système de recommandation pouvait être créé pour les médecins et les médicaments en utilisant l'extraction de critiques, cela permettrait de gagner beaucoup de temps. La création d'un tel système, basé sur l'analyse des critiques et des avis disponibles en ligne, pourrait être très bénéfique. Ce système utiliserait des techniques d'exploration de données (data mining) pour analyser des avis et des critiques sur les médecins et les médicaments. Cela permettrait de proposer des recommandations pertinentes, facilitant ainsi les décisions des utilisateurs et réduisant le temps passé à chercher des informations.

Cependant, dans ce type de système, l'utilisateur rencontre des problèmes pour comprendre le vocabulaire médical hétérogène car les utilisateurs sont des profanes. Un des principaux défis de ces systèmes de recommandation est que les utilisateurs, souvent des personnes sans formation médicale, ont du mal à comprendre le vocabulaire médical varié et complexe. La diversité des termes et des concepts médicaux peut rendre la compréhension difficile pour les non-professionnels de la santé.

Par ailleurs, l'utilisateur est souvent confus car une grande quantité d'informations médicales sur différents supports est disponible. La disponibilité massive d'informations médicales sur de multiples plateformes (sites web, forums, applications mobiles, etc.) peut entraîner une confusion chez les utilisateurs. La surabondance d'informations rend difficile la distinction entre les sources fiables et non fiables, augmentant ainsi l'incertitude et la confusion lors de la prise de décisions liées à la santé.

L'idée derrière le système de recommandation est donc de s'adapter pour répondre aux exigences particulières du domaine de la santé en relation avec les utilisateurs. Le concept d'un système de téléconsultaion pour le domaine de la santé repose sur l'adaptation aux besoins spécifiques des utilisateurs. Cela inclut la personnalisation des recommandations en fonction des préférences et des conditions de santé individuelles, ainsi que l'amélioration de la compréhension et de l'accessibilité des informations médicales pour les utilisateurs profanes.

# Introduction

Avec l'augmentation du nombre de patients et de maladies chaque année, le système médical est surchargé et, avec le temps, est devenu hors de prix dans de nombreux pays. La plupart des maladies nécessitent une consultation avec des médecins pour être traitées. Toutefois, avec suffisamment de données, la prédiction des maladies par un algorithme peut être très facile et peu coûteuse. En effet, la prédiction des maladies en se basant sur les symptômes est une partie intégrante du traitement. Dans notre projet, nous avons tenté de prédire avec précision une maladie en examinant les symptômes du patient. Pour ce faire, nous avons utilisé quatre algorithmes différents et avons obtenu une précision de 92 à 95%.

Un tel système peut avoir un potentiel très important dans le traitement médical du futur. Nous avons également conçu une interface interactive pour faciliter l'interaction avec le système. En outre, nous avons essayé de montrer et de visualiser les résultats de notre étude et de ce projet.

# Collectes de données

L'objectif de cette collecte de données était de créer un ensemble de données complet et détaillé qui pourrait être utilisé pour analyser des tendances médicales, développer des modèles de prédiction de maladies, et améliorer les méthodes de traitement. Les chercheurs ont probablement suivi des protocoles stricts pour garantir que les données collectées étaient précises, complètes et anonymisées afin de protéger la confidentialité des patients.   
   
mon enquête sur la pertinance des données m’a révélés que ces chercheurs étaient tous des doctorants et que les recherchent sont efféctuées aux sein de laboratoirs et ayant eu une durée de 5 à 7 mois.  
  
Ma mémoire étant donc universelle et des enquêtes aux prêt de medecins mon expliqués leur logique de consulataion sur les patients en se basant sur leurs symtômes auras aussi conforté quant à la pertinance de ces données.<http://people.dbmi.columbia.edu/~friedma/Projects/DiseaseSymptomKB/index.html>

# Bibliothêque à utilisé

In this project standard libraries for database analysis and model creation are used. The following are the libraries used in this project.

1. **Numpy** est la bibliothèque principale de calcul scientifique en Python. Elle fournit des outils puissants pour manipuler diverses matrices multidimensionnelles en Python. Il s'agit d'un package de traitement de tableaux à usage général. L'objectif principal de Numpy est de gérer des tableaux homogènes multidimensionnels. Elle offre des outils allant de la création de tableaux à leur manipulation. Par exemple, il est facile de créer un tableau n-dimensionnel en utilisant np.zeros() ou de gérer son contenu avec diverses autres méthodes telles que replace, arrange, random, save, load. Numpy aide également au traitement des tableaux en utilisant des méthodes comme sum, mean, std, max, min, all, etc.

Les tableaux créés avec Numpy se comportent différemment des tableaux créés normalement lorsqu'ils sont manipulés avec des opérateurs tels que +, -, \*, /. Toutes ces qualités et services offerts par les tableaux Numpy les rendent très adaptés à notre objectif de gestion des données. La manipulation des données dans les tableaux lors de l'exécution de diverses opérations doit donner les résultats souhaités, ce qui nécessite de telles capacités opérationnelles élevées.

**pandas** : est la bibliothèque Python la plus populaire utilisée pour l'analyse de données. Elle offre des performances hautement optimisées grâce à un code source backend écrit entièrement en C ou en Python. Les données en Python peuvent être analysées de deux manières :

* Séries
* DataFrames

Une Série est un tableau unidimensionnel défini dans Pandas, utilisé pour stocker tout type de données. Les DataFrames sont une structure de données bidimensionnelle utilisée en Python pour stocker des données constituées de lignes et de colonnes. Les DataFrames de Pandas sont utilisés de manière extensive dans ce projet pour manipuler les jeux de données nécessaires à l'entraînement et au test des algorithmes. Les DataFrames facilitent le travail avec les attributs et les résultats.

Plusieurs de ses fonctions intégrées, telles que replace, ont été utilisées dans notre projet pour la manipulation et le prétraitement des données. Par exemple, la fonction replace permet de remplacer des valeurs spécifiques dans le DataFrame, ce qui est utile pour nettoyer les données et standardiser les valeurs avant de les utiliser dans des modèles de machine learning.

Les DataFrames offrent une interface conviviale pour effectuer des opérations complexes sur les données. Ils permettent de filtrer, trier et agréger les données facilement. Grâce à leur structure en lignes et colonnes, ils ressemblent à des tableaux de bases de données ou à des feuilles de calcul Excel, ce qui rend leur utilisation intuitive pour ceux qui sont habitués à ces formats.

En résumé, Pandas est un outil essentiel pour l'analyse de données en Python, permettant une manipulation efficace et optimisée des ensembles de données. Son utilisation dans notre projet a grandement facilité le traitement et la préparation des données pour les algorithmes de machine learning.

sklearn: est une bibliothèque Python open-source qui implémente une vaste gamme d'algorithmes de machine learning, de pré-traitement, de validation croisée et de visualisation. Elle propose divers outils simples et efficaces pour le data mining et le traitement des données. Elle comprend divers algorithmes de classification, de régression et de clustering tels que les machines à vecteurs de support (Support Vector Machine, SVM), les forêts aléatoires (Random Forest Classifier), les arbres de décision (Decision Tree), le Naïve Bayes gaussien, le KNN (k-Nearest Neighbors), pour n'en nommer que quelques-uns.

Dans ce projet, nous avons utilisé Sklearn pour tirer parti des algorithmes de classification intégrés comme l'arbre de décision (Decision Tree), le classificateur de forêt aléatoire (Random Forest Classifier), le KNN (k-Nearest Neighbors) et le Naïve Bayes. Ces algorithmes ont été choisis pour leur efficacité et leur capacité à gérer différents types de données et de problèmes de classification.

Nous avons également utilisé les fonctionnalités de validation croisée et de visualisation intégrées, telles que le rapport de classification (Classification Report), la matrice de confusion (Confusion Matrix) et le score de précision (Accuracy Score). La validation croisée est une technique essentielle pour évaluer la performance des modèles de machine learning. Elle permet de diviser les données en plusieurs sous-ensembles pour garantir que le modèle est testé de manière exhaustive et que ses performances sont généralisables à des données non vues.

Le rapport de classification fournit des métriques détaillées sur la performance du modèle, telles que la précision, le rappel et le score F1 pour chaque classe. La matrice de confusion offre une vue d'ensemble des prédictions correctes et incorrectes du modèle, ce qui permet d'identifier les classes qui posent des problèmes. Le score de précision, quant à lui, donne une mesure globale de l'exactitude du modèle.

En résumé, Sklearn est une bibliothèque puissante et polyvalente pour le machine learning en Python. Elle offre une gamme d'outils et d'algorithmes qui facilitent le développement, l'évaluation et la visualisation des modèles de machine learning. Son utilisation dans notre projet a permis d'améliorer l'efficacité et la précision de nos prédictions.

1. **Models**

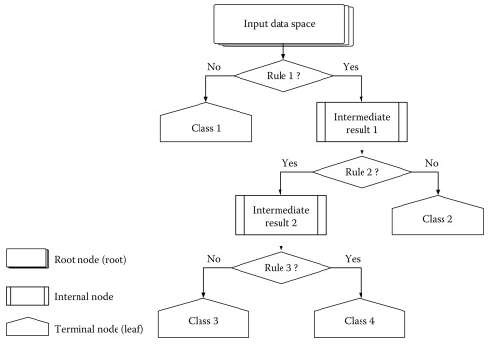
Il y a quatre types de modèles différents présents dans notre projet pour prédire la maladie :

* Arbre de décision
* Forêt aléatoire (Random Forest)
* Naïve Bayes gaussien
* KNN (k-Nearest Neighbors)

L'arbre de décision est classé comme une technique de classification très efficace et polyvalente. Il est utilisé en reconnaissance de formes et en classification pour les images. Il est utilisé pour la classification dans des problèmes très complexes en raison de sa grande adaptabilité. Il est également capable de résoudre des problèmes de dimensionnalité plus élevée. Il se compose principalement de trois parties : la racine, les nœuds et les feuilles.

La racine contient l'attribut qui a le plus d'effet sur le résultat, les nœuds testent la valeur de certains attributs, et les feuilles donnent le résultat de l'arbre. En d'autres termes, l'arbre de décision est une structure d'arbre où chaque nœud représente un test sur un attribut, chaque branche représente le résultat de ce test, et chaque feuille représente une classe de sortie ou une décision.

L'avantage de l'arbre de décision réside dans sa capacité à représenter des règles de décision complexes de manière intuitive et facilement interprétable. Cela en fait un outil précieux pour la classification dans de nombreux domaines, y compris la médecine, où les décisions doivent souvent être prises en fonction de plusieurs facteurs.



Decision tree is the first prediction method we have used in our project. It gives us an accuracy of ~95%.

**Random Forest Algorithm** is a supervised learning algorithm used for both classification and regression. This algorithm works on 4 basic steps –

1. It chooses random data samples from dataset.
2. It constructs decision trees for every sample dataset chosen.
3. At this step every predicted result will be compiled and voted on.
4. At last most voted prediction will be selected and be presented as result of classification.

In this project we have used random forest classifier with 100 random samples and the result given is ~95% accuracy.

**K Nearest Neighbour** is a supervised learning algorithm. It is a basic yet essential algorithm. It finds extensive use in pattern finding and data mining.

It works by finding a pattern in data which links data to results and it improves upon the patter recognition with every iteration.

We have used K Nearest Neighbour to classify our dataset and achieved ~92% accuracy.

**Naïve Bayes** algorithm is a family of algorithms based on naïve bayes theorem. They share a common principle that is every pair of prediction is independent of each other. It also makes an assumption that features make an independent and equal contribution to the prediction.

In our project we have used naïve bayes algorithm to gain a ~95% accurate prediction.

**GUI**

GUI made for this project is a simple tkinter GUI consisting of labels, messagebox, button, text, title and option menu

Root.title() is used to set the the title as Smart Disease Predictor System



Label is used to add heading and contributors section.

Labels are further used for different sections

OptionMenu is used to create drop down menu

Buttons are used to give functionalities and predict the out come of models also two utility buttons namely exit and rest are also created.

Text is used to show output of the prediction using blank space.

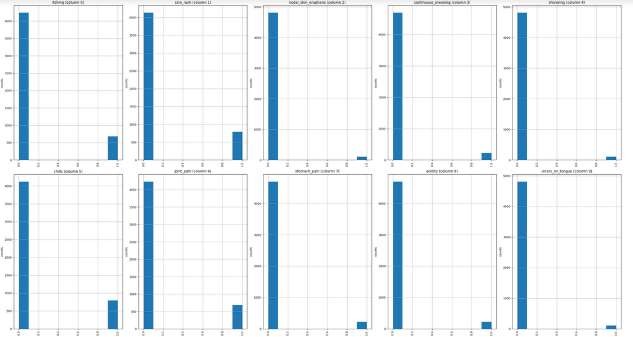
Messagebox are used at three different places, one- to restrain the to enter name

two- to ask for at least two symptoms,

three- to confirm to exit system.

# Modules

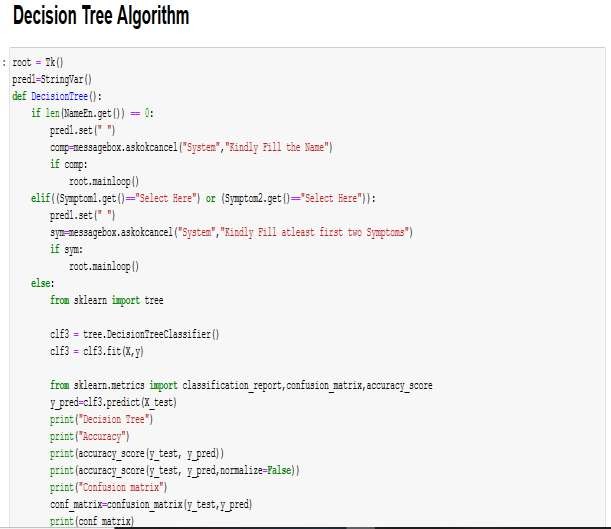


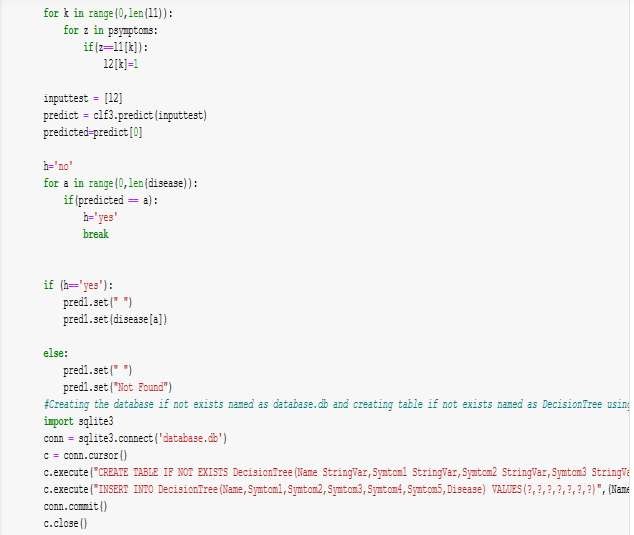


Functions like plotpercolumbdistribution() plotScatterMatrix() is used to viaualize the data.

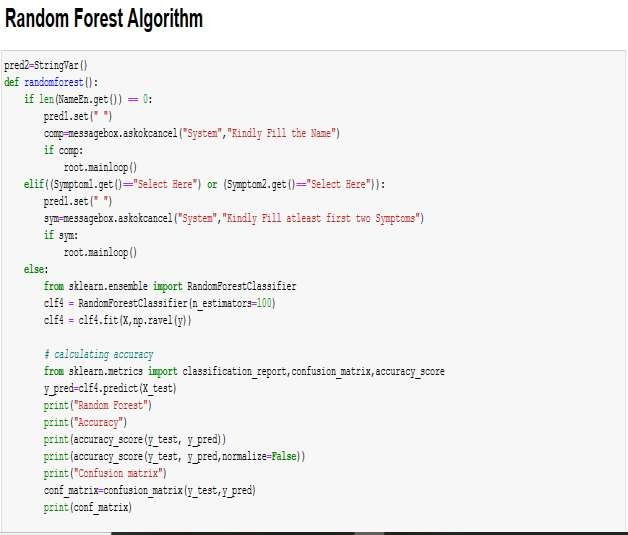


Function like scatterplt and scatterinp are used to compare input to training data.

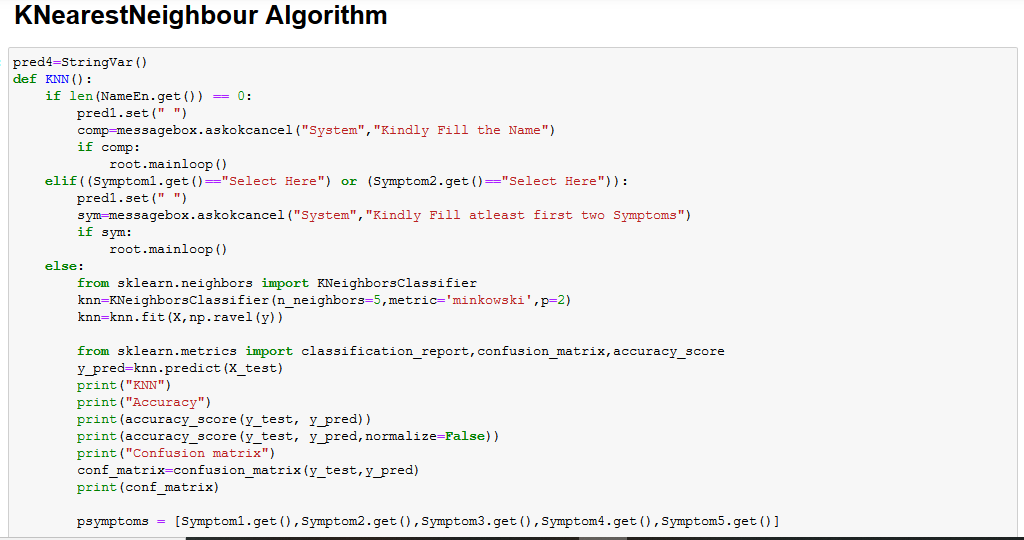




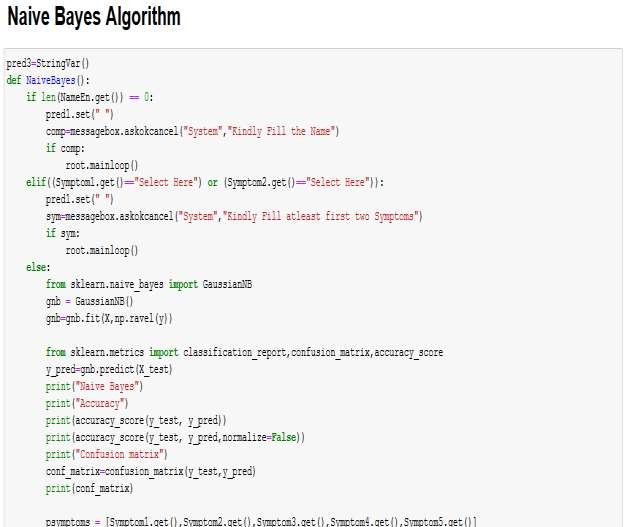
Algorithm of decision tree and database storage.



Algorithm of random forest classifier.

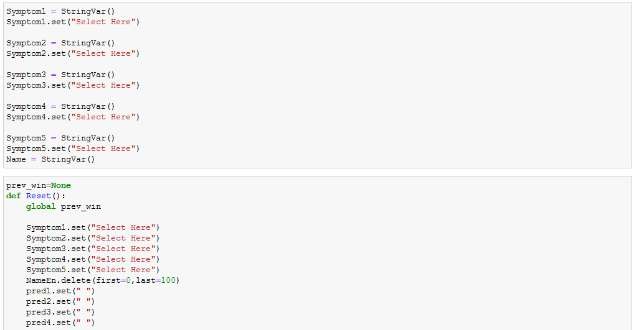


Algorithm of K nearest neighbour.

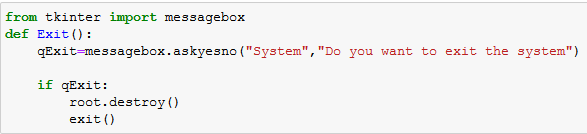


Algorithm of naïve bayes classifier

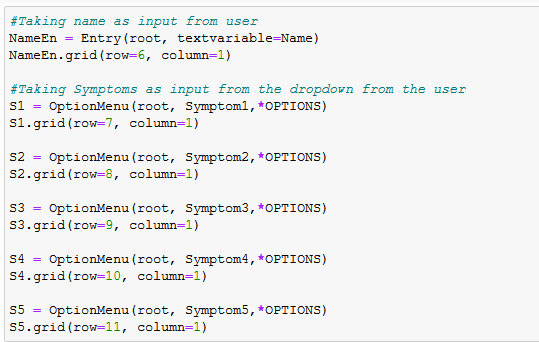
All these classifier is connected to database and GUI to function seamlessly.



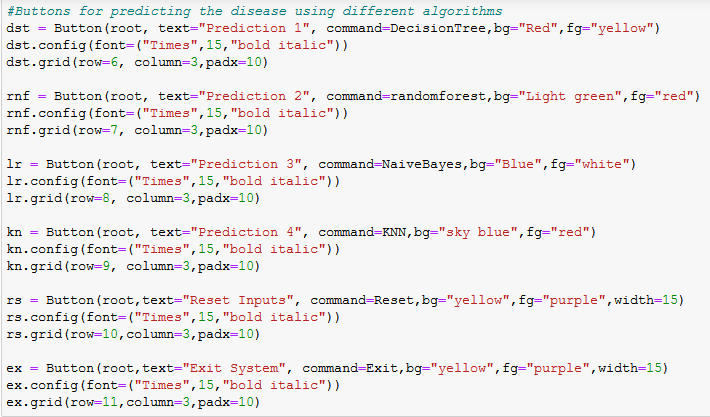
Code of GUI to set initial values of labels.



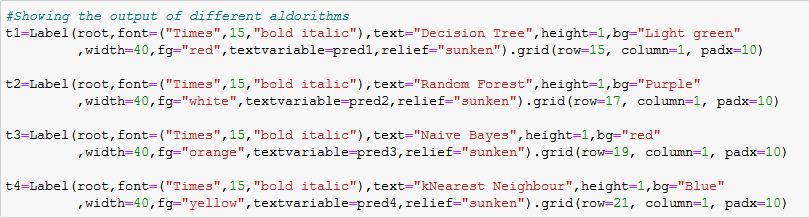
Code of message box.



Code of option menu



Code of buttons.



Code of result display.

# Conclusions

We set out to create a system which can predict disease on the basis of symptoms given to it. Such a system can decrease the rush at OPDs of hospitals and reduce the workload on medical staff. We were successful in creating such a system and use 4 different algorithm to do so. On an average we achieved accuracy of ~94%. Such a system can be largely reliable to do the job. Creating this system we also added a way to store the data entered by the user in the database which can be used in future to help in creating better version of such system. Our system also has an easy to use interface. It also has various visual representation of data collected and results achieved.

# References

a. <http://people.dbmi.columbia.edu/~friedma/Projects/DiseaseSymptomKB/index.html>