***Rapport projet nlp***

# Introduction

*L’extraction de caractéristiques (Features extraction) est un domaine d'application majeur pour les algorithmes machine learning.*

*Cette application sert à analyser les données qui sont soit sous forme texte ou image.*

*L’extraction de caractéristiques commence à partir d'un ensemble initial de données mesurées et construit des valeurs dérivées (caractéristiques) destinées à être informatives et non redondantes.*

# Les bibliothèques utilisées

1. *Nltk (Natural langage toolkit ).*

*Est une* [*bibliothèque logicielle*](https://fr.wikipedia.org/wiki/Biblioth%C3%A8que_logicielle) *en* [*Python*](https://fr.wikipedia.org/wiki/Python_(langage)) *permettant un* [*traitement automatique des langues*](https://fr.wikipedia.org/wiki/Traitement_automatique_des_langues)*, En plus de la bibliothèque, NLTK fournit des démonstrations graphiques, des données-échantillon, des tutoriels, ainsi que la documentation de l'*[*interface de programmation*](https://fr.wikipedia.org/wiki/Interface_de_programmation) *(API).*

# Feature extraction of a text

🡪 *Afin de résoudre ce problème, nltk fournit des utilitaires pour les moyens les plus courants d’extraire des caractéristiques numériques du contenu textuel, à savoir :*

* ***Tokenization***

*C’est la segmentation d’un texte en des mots.*

* ***Counting frequency***

*Compter les fréquences de chaque mot dans le texte.*

* ***Normalizing***

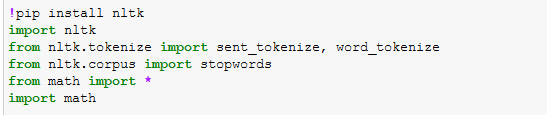
***Tf idf*** *(term-frequency time* ***inverse document-frequency):***

***Tf idf (d,t)=Tf(t,d)\* idf(t)***

***Tf (t) = (Number of times term t appears in a document) / (Total number of terms in the document)***

***idf*** *(t) = log\_e(Total number of documents / Number of documents with term t in it)*

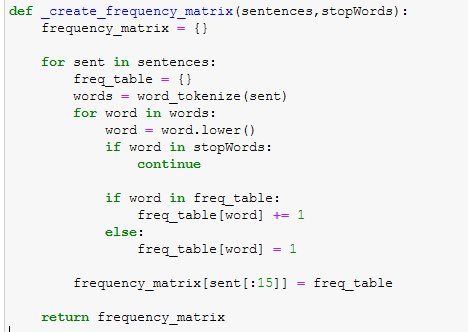
***->pour commencer il faut importer les bibliothèques demandé.***

******

***->ensuite on segmente le texte en des phrases.***

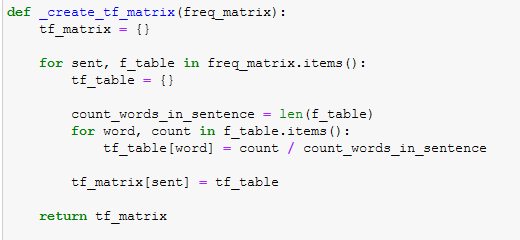
******

***->maintenant pour chaque phrase on extrait les mots et on calcule la fréquence de chaque mot dans le texte.***

******

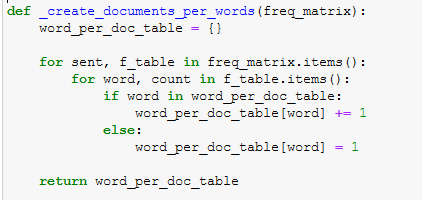
***->******Ici, chaque phrase est la clé et la valeur est un dictionnaire de fréquence de mots.***

***->après Calculer TermFrequency et générer une matrice***

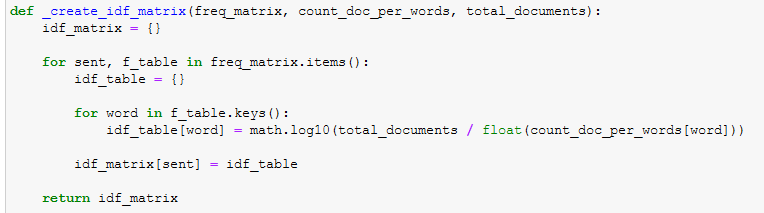
******

***->******Nous trouverons le TermFrequency pour chaque mot dans un paragraphe.***

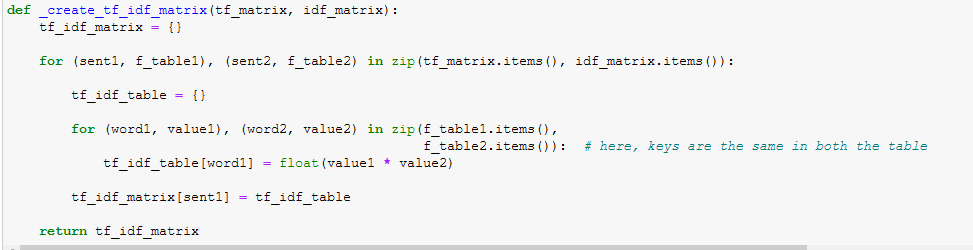
***-> Ensuite nous calculons, combien de phrases contiennent un mot cela nous aide pour calculer idf***

******

***->maintenant idf***

******

***->et enfin tf idf***

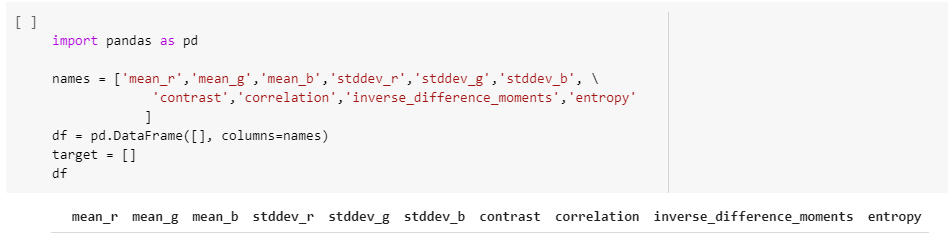
******

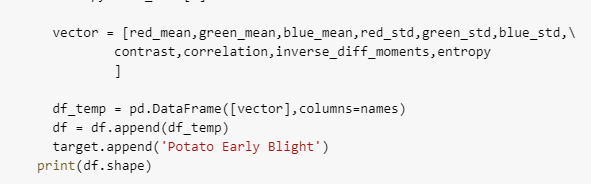
***->après la génération de la matrice qui contient les valeurs de tf idf de chaque mot on pourrait extraire les caractéristiques du texte***

# Development d’un machine learning “Disease detection(optional)”

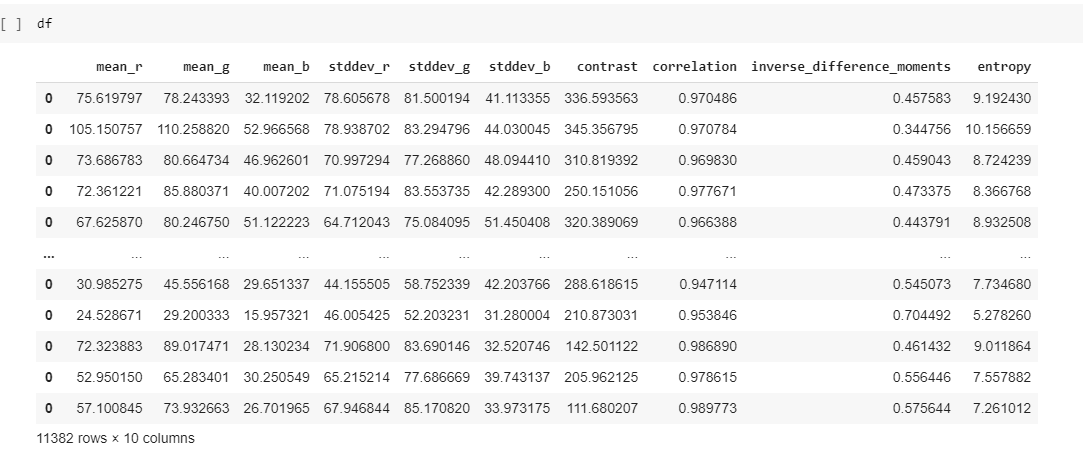
1. *Clonage du base de donnée qu’on va utiliser pour l’entrainement :  
   *
2. *Installation des librairies besoin pour le projet :  
   *

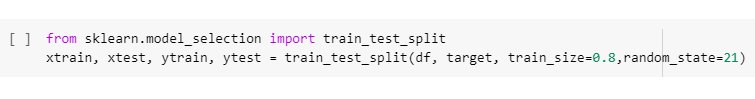
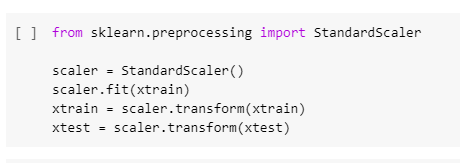
* *Mahotas et Opencv-python po*  *ur le traitement d’image*
* *sklearn pour entrainement des algorithmes*

1. *Segmentation des images   
   *
2. *Création de la trame de donnée :* **
3. *Traitement d’image :  
   *

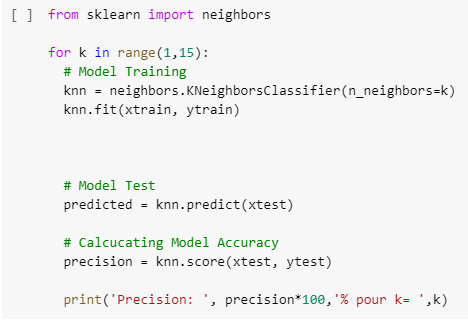
**

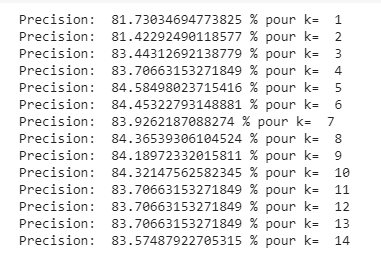
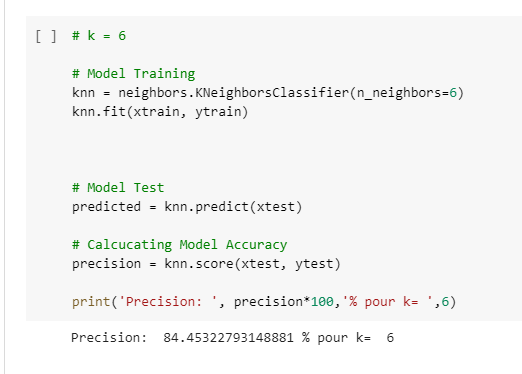
* *Répéter de même pour tous les maladies de plant !!!*

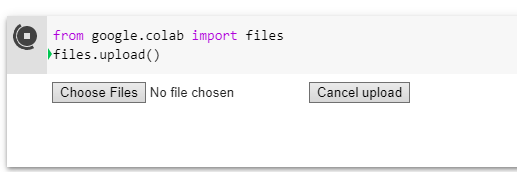
1. *Affichage de la trame df ou on a mis les résultats du traitement d’image :* **
2. *Fractionnement et mise en échelle des données :*

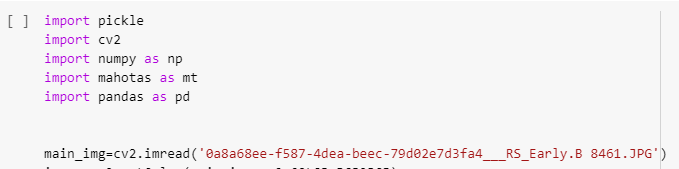
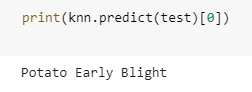
** **

Ect … le reste et dans le ficheir .ipynb github !!!!!!

1. *La modélisation :  
   *

* *Résultat   
  *
* *Le choix de k est 6 pour meilleur precision :  
  *

1. *Test :  
   *

* *choix de image d’après la machine d’un feuille de plante « tomate ou bien pomme de terre ».*
* *Changer le nom d’image a l’intérieur de imread avec le nom d’image choisi avant.*
* *On aura un résultat qui ressemble a celui la*
* *Si non vous pouvez tester avec le fichier   
  Diseace\_Detection.sav !!!!!!!!!!!!*