



Master : Cryptologie Et Sécurité Informatique

Rapport



Réalisée par :

ELMNAJJA DINA

ELM'RABTI WAFAE

Encadré par :

Pr. SOUFI ADIL

Année universitaire :2024/2025

TABLE DES MATIÈRES

Introduction	5
1 Base de Donnée	6
1.1 Caractéristiques des Champignons	6
1.2 Quelque Visualisation des Caractéristiques	8
2 Algorithmes de Classification Utilisées	11
2.1 Support Vector Machine(SVM)	11
2.2 K-Nearest Neighbors	12
2.3 Régression Logistique	13
3 Étapes de Prétraitement pour l'Analyse des Champignons	15
3.1 Chargement des Données	15
3.2 Prétraitement des Données	16
3.2.1 Identification des Valeurs Manquantes	16
3.3 Conversion des Données	16
3.4 Division des Données	17
3.5 Modèles d'Entrainement	18
3.5.1 Support Vector Machine	18
3.5.2 K-Nearest Neighbors	18
3.5.3 Logistic Regression	19
3.6 Matrice de Confusion	19
4 Implémentation	20
4.1 Outils Pour le Développement D'applications Web	20
4.1.1 Flask	20
4.1.2 Bibliothèques Cruciales	20
Scikit-learn	20
Pandas	21
Numpy	21
4.2 Interface Web	22
Conclusion	24

TABLE DES FIGURES

1.1	Extraire de Data	6
1.2	Cap-Color	8
1.3	Habitat	9
1.4	Cap-surface	9
1.5	Odor	10
1.6	Cap-Shape	10
2.1	Exemple de Fonctionnement de SVM	12
2.2	Exemple de Fonctionnement de KNN	13
2.3	Exemple de Fonctionnement de Régression Logistique	14
3.1	Code de Chargement des Données	15
3.2	Code d'identifier les Valeures Manquantes	16
3.3	Code de Conversion des Données	17
3.4	Code de Division des Données	18
3.5	Code d'Entrainement de SVM	18
3.6	Code d'Entrainement de KNN	18
3.7	Code d'Entraînement de Régression Logistique	19
3.8	Métriques de Confusion	19
4.1	Résultat de Mushroom : "Toxic"	22
4.2	Résultat de Mushroom : "Edible"	23

INTRODUCTION

Les champignons sont des organismes fascinants, présents dans une grande variété d'environnements à travers le monde. Cependant, leur diversité pose un défi lorsqu'il s'agit de les identifier et de distinguer les espèces comestibles des espèces toxiques. La classification manuelle des champignons peut être complexe et risquée, car de nombreuses espèces non comestibles ou toxiques peuvent ressembler à des variétés comestibles.

Dans le domaine de l'apprentissage automatique, la classification des champignons est un problème intéressant qui peut être résolu à l'aide de techniques de machine learning. En utilisant des ensembles de données contenant des informations sur les caractéristiques morphologiques des champignons, telles que la forme du chapeau, la couleur, la présence de lamelles et d'autres attributs, il est possible de développer des modèles prédictifs capables de différencier les champignons comestibles des non comestibles.

Dans ce rapport, nous explorons l'application de divers algorithmes de classification de machine learning pour résoudre le problème de classification des champignons. Nous examinons en détail les données utilisées, les techniques de prétraitement, la sélection des caractéristiques et l'évaluation des modèles. En fin de compte, l'objectif est de développer un modèle robuste et précis capable d'aider à identifier les champignons comestibles et à réduire les risques associés à la consommation de champignons toxiques.

CHAPITRE 1

BASE DE DONNÉE

Ce jeu de données sur les champignons comprend des descriptions de spécimens hypothétiques correspondant à 23 espèces de champignons à lamelles de la famille Agaricus et Lepiota, tirées du Guide des champignons nord-américains de la Société Audubon (1981). Chaque espèce est identifiée comme définitivement comestible, définitivement toxique, ou d'édibilité inconnue et non recommandée. Cette dernière classe a été combinée avec la classe des champignons toxiques. Le guide indique clairement qu'il n'existe pas de règle simple pour déterminer l'édibilité d'un champignon ; aucune règle comme "trois folioles, laisse-le tranquille" pour le chêne et le lierre véneneux.

class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	...	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color	ring-number	ring-type	spore-print-color	population
0	p	x	s	n	t	p	f	c	n	k	...	s	w	w	p	w	o	p	k
1	e	x	s	y	t	a	f	c	b	k	...	s	w	w	p	w	o	p	n
2	e	b	s	w	t	I	f	c	b	n	...	s	w	w	p	w	o	p	n
3	p	x	y	w	t	p	f	c	n	n	...	s	w	w	p	w	o	p	k
4	e	x	s	g	f	n	f	w	b	k	...	s	w	w	p	w	o	e	n
...	
8119	e	k	s	n	f	n	a	c	b	y	...	s	o	o	p	o	o	p	b
8120	e	x	s	n	f	n	a	c	b	y	...	s	o	o	p	n	o	p	b
8121	e	f	s	n	f	n	a	c	b	n	...	s	o	o	p	o	o	p	b
8122	p	k	y	n	f	y	f	c	n	b	...	k	w	w	p	w	o	e	w
8123	e	x	s	n	f	n	a	c	b	y	...	s	o	o	p	o	o	p	o

8124 rows × 23 columns

FIGURE 1.1 – Extraire de Data

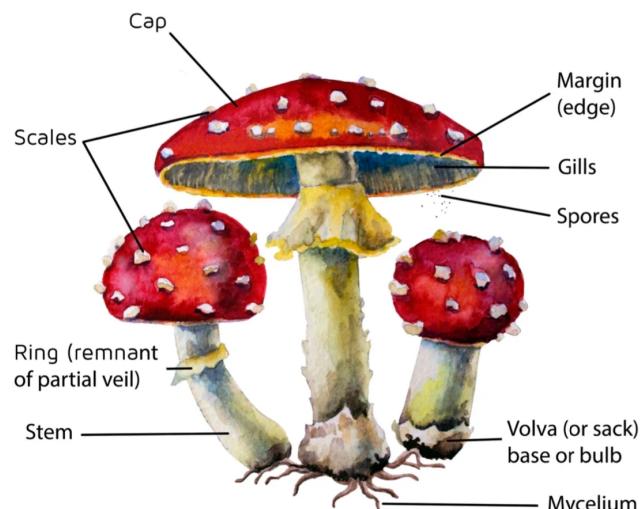
1.1 Caractéristiques des Champignons

Les caractéristiques des champignons sont décrites par les attributs suivants :

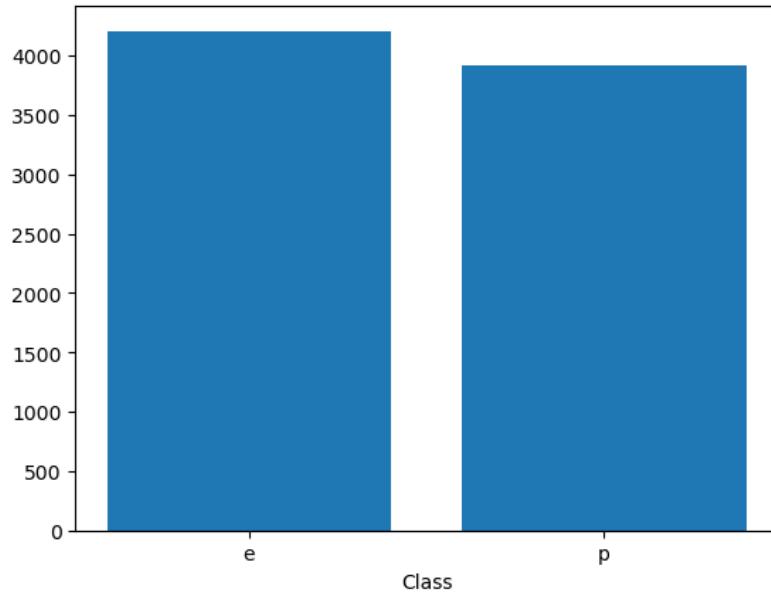
- **cap-shape** : bell=**b**,conical=**c**,convex=**x**,flat=**f**, knobbed=**k**,sunken=**s**.
- **cap-surface** : fibrous=**f**,grooves=**g**,scaly=**y**,smooth=**s**
- **cap-color** : brown=**n**,buff=**b**,cinnamon=**c**,gray=**g**,green=**r**,pink=**p**,purple=**u**, red=**e**,white=**w**,yellow=**y**

- **bruises** : bruises=t,no=f
- **odor** : almond=a,anise=l,creosote=c,fishy=y,foul=f,musty=m,none=n,pungent=p,spicy=s
- **gill-attachment** : attached=a,descending=d,free=f,notched=n
- **gill-spacing** : close=c,crowded=w,distant=d
- **gill-size** : broad=b,narrow=n
- **gill-color** : black=k,brown=n,buff=b,chocolate=h,gray=g,green=r,orange=o,pink=p,purple=u,red=e,white=w,yellow=y
- **stalk-shape** : enlarging=e,tapering=t
- **stalk-root** : bulbous=b,club=c,cup=u,equal=e,rhizomorphs=z,rooted=r,missing=?
- **stalk-surface-above-ring** : fibrous=f,scaly=y,silky=k,smooth=s
- **stalk-surface-below-ring** : fibrous=f,scaly=y,silky=k,smooth=s
- **stalk-color-above-ring** : brown=n,buff=b,cinnamon=c,gray=g,orange=o,pink=p,red=e,white=w,yellow=y
- **stalk-color-below-ring** : brown=n,buff=b,cinnamon=c,gray=g,orange=o,pink=p,red=e,white=w,yellow=y
- **veil-type** : partial=p,universal=u
- **veil-color** : brown=n,orange=o,white=w,yellow=y
- **ring-number** : none=n,one=o,two=t
- **ring-type** : cobwebby=c,evanescent=e,flaring=f,large=l,none=n,pendant=p,sheathing=s,zone=z
- **spore-print-color** : black=k,brown=n,buff=b,chocolate=h,green=r,orange=o,purple=u,white=w,yellow=y
- **population** : abundant=a,clustered=c,numerous=n,scattered=s,several=v,solitary=y
- **habitat** : grasses=g,leaves=l,meadows=m,paths=p,urban=u,waste=w,woods=d

Le but de l'analyse de ce jeu de données est de déterminer quelles caractéristiques sont les plus indicatives d'un champignon toxique et d'entraîner des modèles d'apprentissage automatique pour prédire si un champignon est comestible ou toxique en fonction de ses caractéristiques.



La figure ci-dessous présente la visualisation de l'édibilité et de la toxicité des champignons dans notre ensemble de données de classification. Cette représentation graphique permet d'observer clairement la répartition des champignons entre les catégories comestibles et toxiques. En examinant cette visualisation, nous pouvons mieux comprendre la répartition des classes dans notre ensemble de données, ce qui est essentiel pour élaborer un modèle de classification précis et fiable.



1.2 Quelque Visualisation des Caractéristiques

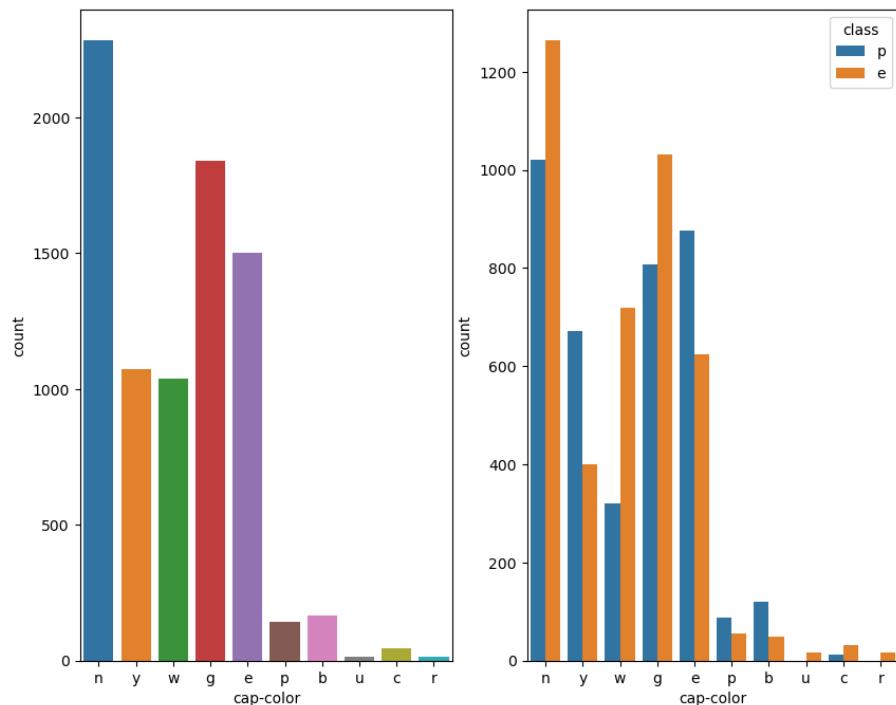


FIGURE 1.2 – Cap-Color

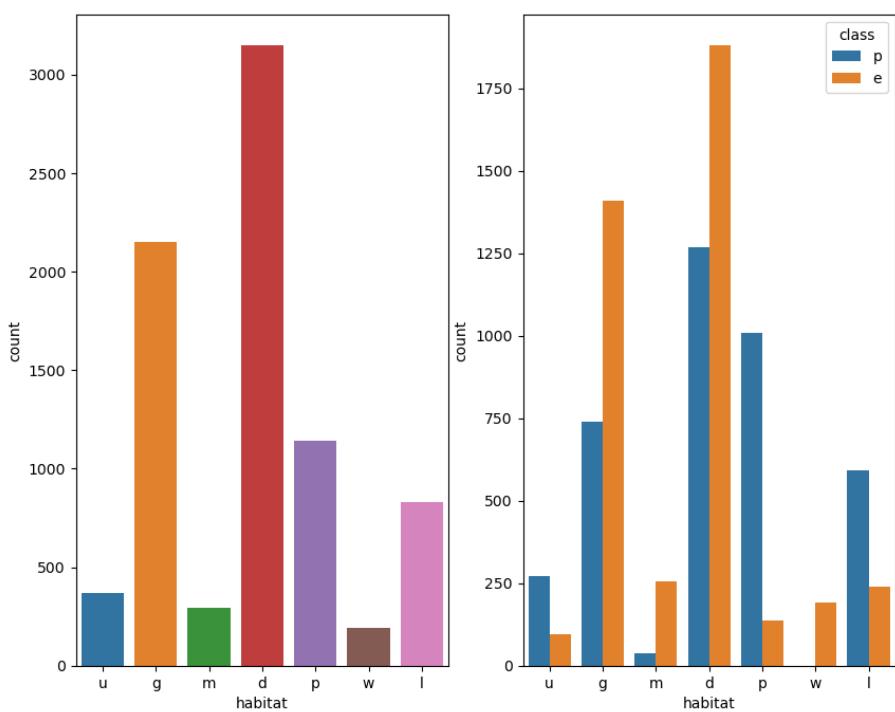


FIGURE 1.3 – Habitat

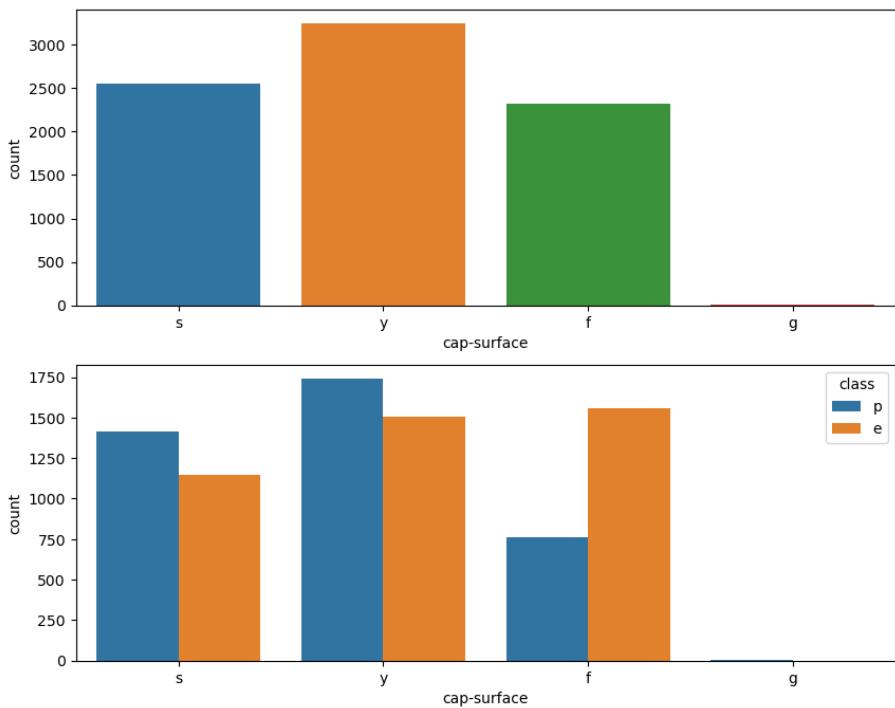


FIGURE 1.4 – Cap-surface

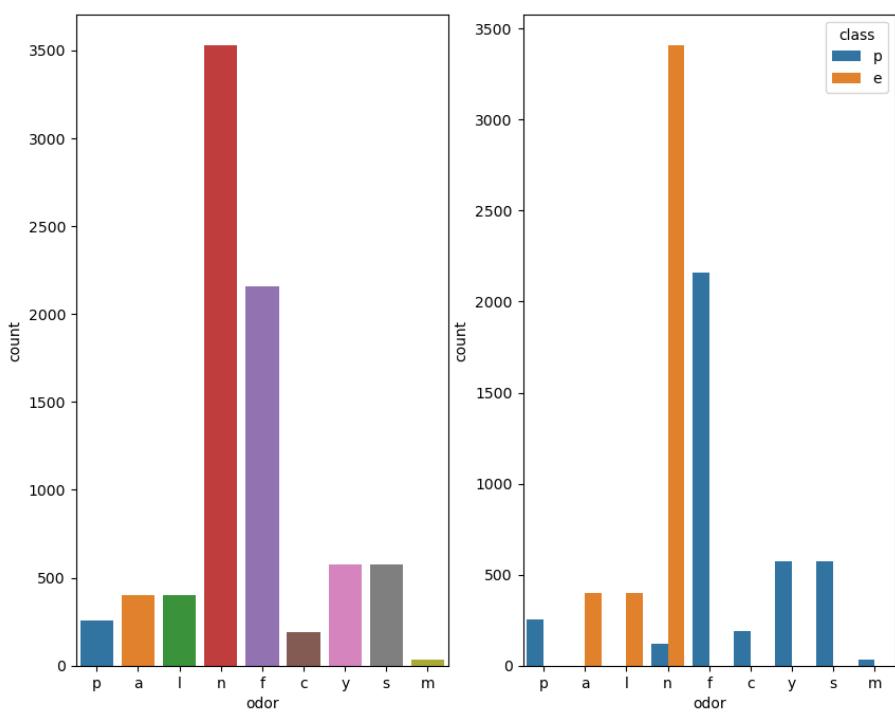


FIGURE 1.5 – Odor

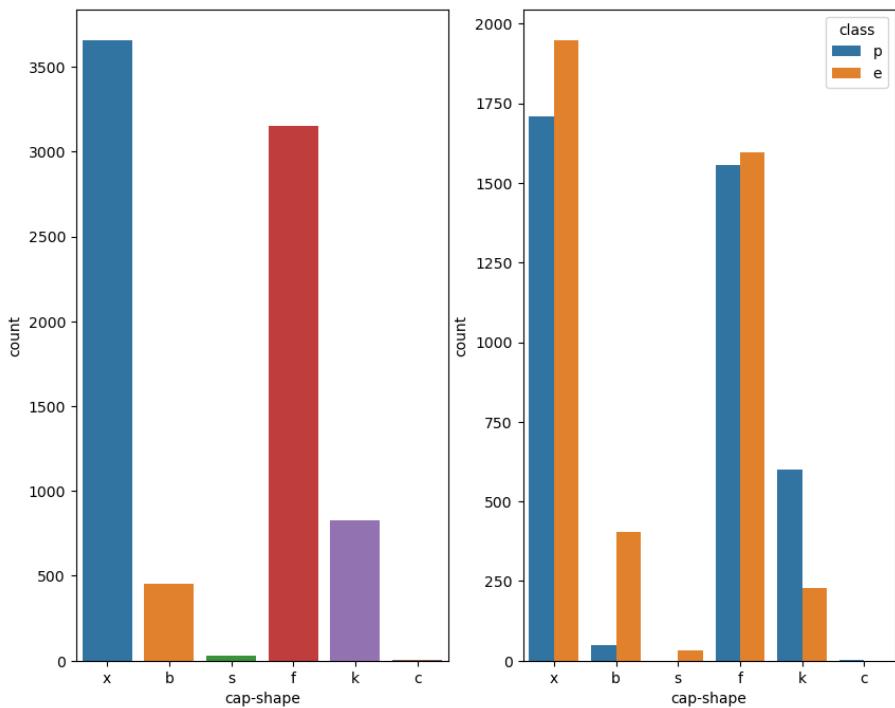


FIGURE 1.6 – Cap-Shape

CHAPITRE 2

ALGORITHMES DE CLASSIFICATION UTILISÉES

2.1 Support Vector Machine(SVM)

Les machines à vecteurs de support (SVM) sont des modèles d'apprentissage automatique puissants utilisés pour la classification et la régression. L'objectif principal de SVM en classification est de trouver l'hyperplan optimal qui sépare les données en deux classes tout en maximisant la marge entre les deux classes. Cette marge est la distance entre l'hyperplan et les points de données les plus proches de chaque classe, appelés vecteurs de support. SVM peut effectuer une séparation linéaire ou non linéaire des données en utilisant des fonctions de noyau pour projeter les données dans un espace de dimension supérieure où elles sont potentiellement linéairement séparables. Ensuite, SVM résout un problème d'optimisation quadratique pour trouver l'hyperplan optimal en minimisant la norme du vecteur de poids de l'hyperplan, tout en maximisant la marge et en garantissant que tous les points de données sont classés correctement. Une fois l'hyperplan optimal trouvé, SVM peut être utilisé pour prédire la classe d'un nouveau point de données en le classant en fonction de quel côté de l'hyperplan il se trouve.

Alors dans notre projet,nous avons choisi d'utiliser cet algorithme pour les raisons suivantes :

- **Efficace dans les espaces de grandes dimensions :** SVM fonctionne bien dans les espaces de grande dimension, comme dans le cas où le nombre de caractéristiques des champignons est élevé.(Dans notre cas, nos données contiennent 23 colonnes, correspondant à 23 caractéristiques de champignons.)
- **Capacité à traiter des données non linéaires :** Avec l'utilisation de noyaux, les SVM peuvent gérer efficacement des données qui ne sont pas linéairement séparables.
- **Bon contrôle de l'overfitting :** En utilisant le paramètre de régularisation, les SVM peuvent bien contrôler le surajustement, ce qui est important lorsque les données sont limitées.

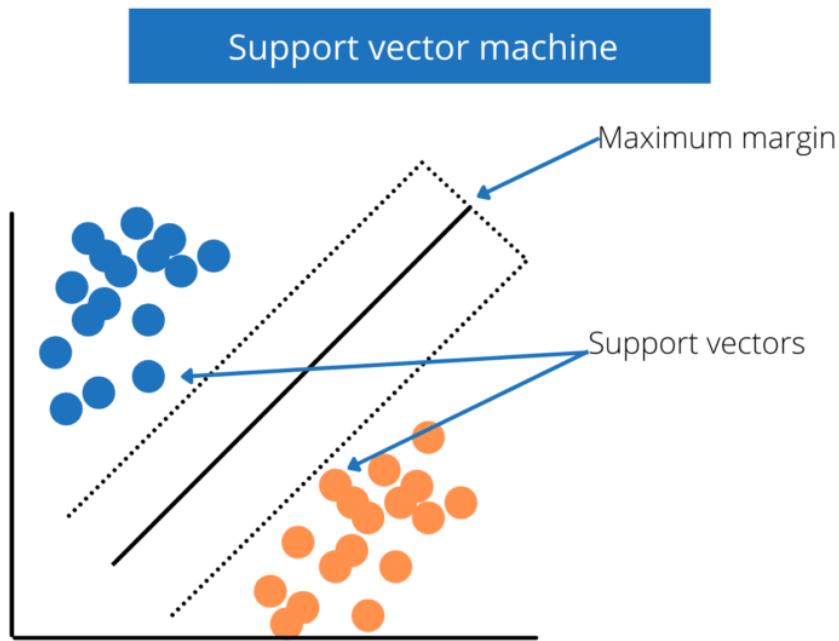


FIGURE 2.1 – Exemple de Fonctionnement de SVM

2.2 K-Nearest Neighbors

L'algorithme K-NN (K-nearest neighbors) est une méthode d'apprentissage supervisé. Il peut être utilisé aussi bien pour la régression que pour la classification. Son fonctionnement peut être assimilé à l'analogie suivante “**dis moi qui sont tes voisins, je te dirais qui tu es...**”.

En régression, K-NN prédit la valeur d'une variable cible en prenant la moyenne (ou la médiane) des valeurs de cette variable pour les K observations les plus proches dans l'espace des caractéristiques. Par exemple, si nous voulons prédire le prix d'une maison en fonction de ses caractéristiques, nous pourrions utiliser K-NN pour trouver les K maisons les plus similaires en termes de caractéristiques et utiliser la moyenne de leurs prix comme prédiction.

En classification, K-NN prédit la classe d'une observation en prenant le mode (la valeur la plus fréquente) des classes des K observations les plus proches. Par exemple, si nous voulons classer un champignon comme comestible ou toxique en fonction de ses caractéristiques, nous pourrions utiliser K-NN pour trouver les K champignons les plus similaires et attribuer à notre champignon la classe la plus fréquente parmi ces K voisins.

Pour effectuer une prédiction, l'algorithme K-NN ne va pas calculer un modèle prédictif à partir d'un Training Set comme c'est le cas pour la régression logistique ou la régression linéaire. En effet, K-NN n'a pas besoin de construire un modèle prédictif. Ainsi, pour K-NN il n'existe pas de phase d'apprentissage proprement dite. C'est pour cela qu'on le catégorise parfois dans le Lazy Learning. Pour pouvoir effectuer une prédiction, K-NN se base sur le jeu de données pour produire un résultat.

L'algorithme K-NN va se baser sur le jeu de données en entier. En effet, pour une observation, qui ne fait pas parti du jeu de données, qu'on souhaite prédire, l'algorithme va chercher les K instances du jeu de données les plus proches de notre observation. Ensuite pour ces K voisins, l'algorithme se basera sur leurs variables de sortie (output variable) y pour calculer la valeur de la variable y de l'observation qu'on souhaite prédire.

Dans notre projet, nous avons choisi d'utiliser cet algorithme pour les raisons suivantes :

- **Simplicité conceptuelle** : KNN est intuitif et facile à comprendre, car il se base sur la similarité entre les données.
- **Pas de suppositions sur la distribution des données** : Contrairement à certains autres algorithmes, KNN n'impose pas d'hypothèses sur la distribution des données, ce qui le rend plus flexible.
- **Robuste aux données bruitées** : KNN peut bien fonctionner même avec des données bruitées ou des frontières de décision complexes.

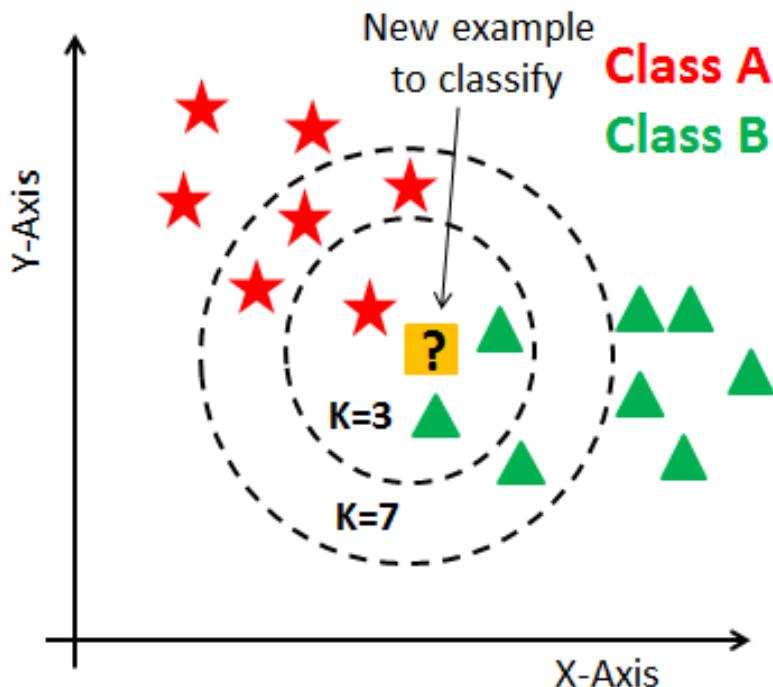


FIGURE 2.2 – Exemple de Fonctionnement de KNN

2.3 Régression Logistique

La régression logistique est une approche statistique qui peut être employée pour évaluer et caractériser les relations entre une variable réponse de type binaire (par exemple : Vivant / Mort, Malade / Non malade, succès / échec), et une, ou plusieurs,

variables explicatives, qui peuvent être de type catégoriel (le sexe par exemple), ou numérique continu (l'âge par exemple).

La régression logistique multinomiale est une extension de la régression logistique aux variables qualitatives à trois modalités ou plus, la régression logistique ordinaire aux variables qualitatives à trois modalités ou plus qui sont ordonnées hiérarchiquement.

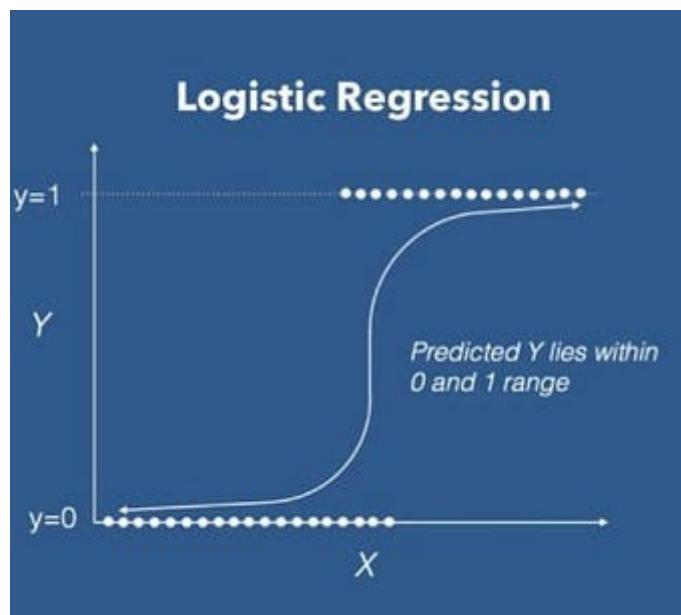


FIGURE 2.3 – Exemple de Fonctionnement de Régression Logistique

Nous avons choisi d'utiliser cet algorithme dans notre projet pour les raisons suivantes :

- **Interprétabilité des coefficients** : La régression logistique fournit des coefficients pour chaque caractéristique, ce qui permet d'identifier les caractéristiques les plus importantes pour la classification des champignons.
- **Bonnes performances avec des classes binaires** : Si votre problème de classification se réduit à une classification binaire (comestible(edible) ou non comestible(poisonous)), la régression logistique peut être efficace et rapide à entraîner.

CHAPITRE 3

ÉTAPES DE PRÉTRAITEMENT POUR L'ANALYSE DES CHAMPIGNONS

3.1 Chargement des Données

La collecte de données est une étape cruciale dans le processus de classification des champignons, car nous avons besoin de données pour entraîner notre modèle. Nous avons utilisé une base de données comprenant des informations sur différentes espèces de champignons, telles que leur apparence, leur habitat et leur comestibilité. Ces données ont été collectées à partir de diverses sources fiables, y compris des guides de champignons et des bases de données spécialisées. En utilisant cette base de données, nous avons pu fournir à notre algorithme d'apprentissage automatique un ensemble varié de caractéristiques de champignons, ce qui lui a permis de mieux généraliser et de prendre des décisions plus précises lors de la classification des champignons.

Loading data

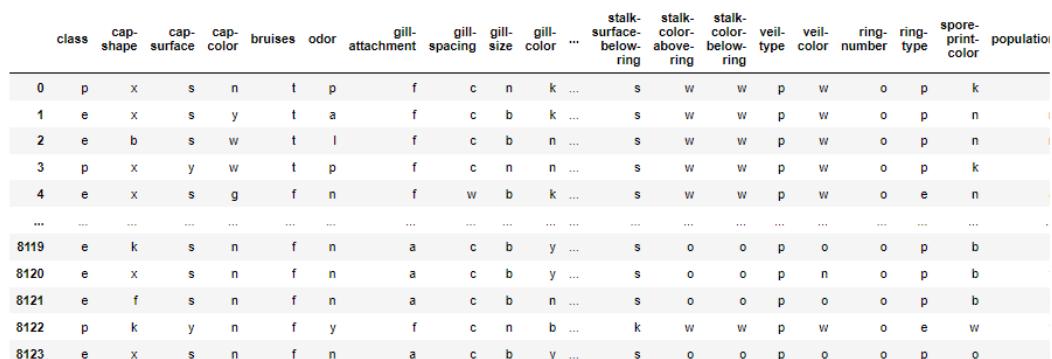
In [164]:	dataset=pd.read_csv('C:/Users/pc/Desktop/Mushroom_project/mushroom_data/mushrooms.csv')
In [165]:	dataset
Out[165]:	
	
8124 rows × 23 columns	

FIGURE 3.1 – Code de Chargement des Données

3.2 Prétraitement des Données

La principale cible du prétraitement des données en apprentissage automatique est d'améliorer la qualité des données d'entrée afin d'optimiser les performances et l'efficacité des modèles d'apprentissage automatique. Le prétraitement vise à rendre les données plus adaptées aux algorithmes utilisés en traitant divers problèmes tels que les valeurs manquantes, les données bruyantes, la mise à l'échelle, la normalisation, l'ingénierie des fonctionnalités, et plus encore. En prétraitant correctement les données, les modèles d'apprentissage automatique peuvent mieux capturer les motifs et les relations au sein des données, ce qui conduit à des prédictions et des insights plus précis.

3.2.1 Identification des Valeurs Manquantes

Lors de l'identification des valeurs manquantes dans un ensemble de données en machine learning, il est important de noter que les valeurs manquantes ne sont pas toujours représentées par **NAN** (Not a Number). Elles peuvent également être indiquées par d'autres symboles ou conventions, comme des points d'interrogation (?), des valeurs spéciales (comme -999 ou 999), des chaînes vides, ou d'autres indicateurs spécifiques choisis par le créateur du jeu de données.

```
In [184]: sns.heatmap(dataset.isin(["?", "NaN"]), cmap='Blues')# if we have "NaN" or "?" values
Out[184]: <Axes: >
```

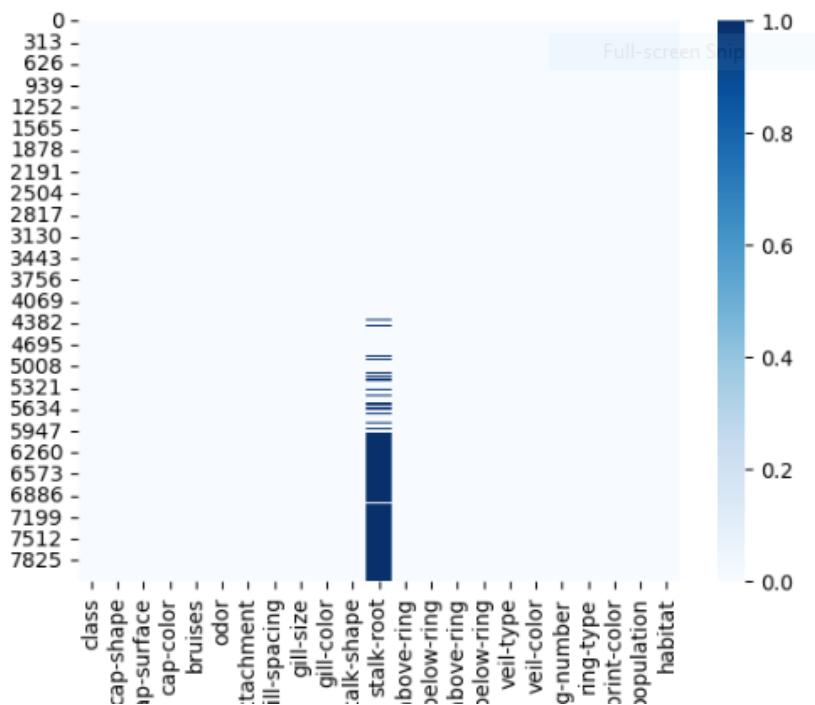


FIGURE 3.2 – Code d'identifier les Valeures Manquantes

3.3 Conversion des Données

Avant de former nos données et d'appliquer l'algorithme, nous devons convertir toutes les variables en valeurs numériques. Cela garantit que notre algorithme puisse

traiter les données de manière appropriée et effectuer des calculs cohérents pour générer des prédictions précises. La conversion des variables en valeurs numériques est une étape essentielle dans le processus d'apprentissage automatique, car elle permet de standardiser le format des données et de faciliter la manipulation par les algorithmes.

Converting

```
In [189]: features={}
for feature in new_columns:
    Q=dataset[feature].unique()
    Q.sort()
    l=[i+1 for i in range(len(Q))]
    features[feature]={0:Q,1:l}

In [190]: features

Out[190]: {'class': {0: array(['e', 'p'], dtype=object), 1: [1, 2]},
 'cap-shape': {0: array(['b', 'c', 'f', 'k', 's', 'x'], dtype=object),
 1: [1, 2, 3, 4, 5, 6]},
 'cap-surface': {0: array(['f', 'g', 's', 'y'], dtype=object),
 1: [1, 2, 3, 4]}, 
 'cap-color': {0: array(['b', 'c', 'e', 'g', 'n', 'p', 'r', 'u', 'w', 'y'], dtype=object),
 1: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]}, 
 'bruises': {0: array(['f', 't'], dtype=object), 1: [1, 2]},
 'odor': {0: array(['a', 'c', 'f', 'l', 'm', 'n', 'p', 's', 'y'], dtype=object),
 1: [1, 2, 3, 4, 5, 6, 7, 8, 9]}, 
 'gill-attachment': {0: array(['a', 'f'], dtype=object), 1: [1, 2]},
 'gill-spacing': {0: array(['c', 'w'], dtype=object), 1: [1, 2]},
 'gill-size': {0: array(['b', 'n'], dtype=object), 1: [1, 2]}, 
 'gill-color': {0: array(['b', 'e', 'g', 'h', 'k', 'n', 'o', 'p', 'r', 'u', 'w', 'y'],
  dtype=object),
 1: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]}, 
 'stalk-shape': {0: array(['e', 't'], dtype=object), 1: [1, 2]},
 'stalk-surface-above-ring': {0: array(['f', 'k', 's', 'y'], dtype=object),
 1: [1, 2, 3, 4]}, 
 'stalk-surface-below-ring': {0: array(['f', 'k', 's', 'y'], dtype=object),
 1: [1, 2, 3, 4]}, 
 'stalk-color-above-ring': {0: array(['b', 'c', 'e', 'g', 'n', 'o', 'p', 'w', 'y'], dtype=object),
 1: [1, 2, 3, 4, 5, 6, 7, 8, 9]}}

In [191]: for column in new_columns:
    dataset[column].replace(features[column][0],features[column][1],inplace=True)

In [192]: dataset

Out[192]:
   class cap_shape cap_surface cap_color bruises odor gill_attachment gill_spacing gill_size gill_color ... stalk_surface_above_ring stalk_surface_below_ring stalk_color_above_ring stalk_color_below_ring veil_color ring_number ring_type spore_print_color popul
0      2         6           3       5     2    7          2        1     2     5 ...          3          3        8       8     3      2      5      3
1      1         6           3       10    2   1          2        1     1     5 ...          3          3        8       8     3      2      5      4
2      1         1           3       9     2    4          2        1     1     6 ...          3          3        8       8     3      2      5      4
3      2         6           4       9     2    7          2        1     2     6 ...          3          3        8       8     3      2      5      3
4      1         6           3       4     1    6          2        2     1     5 ...          3          3        8       8     3      2      1      4
...   ...
8119   1         4           3       5     1    6          1        1     1    12 ...          3          3        6       6     2      2      5      1
8120   1         6           3       5     1    6          1        1     1    12 ...          3          3        6       6     1      2      5      1
8121   1         3           3       5     1    6          1        1     1     6 ...          3          3        6       6     2      2      5      1
8122   2         4           4       5     1    9          2        1     2     1 ...          3          2        8       8     3      2      1      8
8123   1         6           3       5     1    6          1        1     1    12 ...          3          3        6       6     2      2      5      5
8124   1         2           2       2     1    2          1        1     1    12 ...          3          3        6       6     2      2      5      5
8125   1         3           3       3     1    3          1        1     1     3 ...          3          3        6       6     1      1      3      1
8126   1         4           4       4     1    4          1        1     1     4 ...          3          3        6       6     1      1      4      1
8127   1         5           5       5     1    5          1        1     1     5 ...          3          3        6       6     1      1      5      1
8128   1         6           6       6     1    6          1        1     1     6 ...          3          3        6       6     1      1      6      1
8129   1         7           7       7     1    7          1        1     1     7 ...          3          3        6       6     1      1      7      1
8130   1         8           8       8     1    8          1        1     1     8 ...          3          3        6       6     1      1      8      1
8131   1         9           9       9     1    9          1        1     1     9 ...          3          3        6       6     1      1      9      1
8132   1         1           1       1     1    1          1        1     1     1 ...          3          3        6       6     1      1      1      1
8133   1         2           2       2     1    2          1        1     1     2 ...          3          3        6       6     1      1      2      1
8134   1         3           3       3     1    3          1        1     1     3 ...          3          3        6       6     1      1      3      1
8135   1         4           4       4     1    4          1        1     1     4 ...          3          3        6       6     1      1      4      1
8136   1         5           5       5     1    5          1        1     1     5 ...          3          3        6       6     1      1      5      1
8137   1         6           6       6     1    6          1        1     1     6 ...          3          3        6       6     1      1      6      1
8138   1         7           7       7     1    7          1        1     1     7 ...          3          3        6       6     1      1      7      1
8139   1         8           8       8     1    8          1        1     1     8 ...          3          3        6       6     1      1      8      1
8140   1         9           9       9     1    9          1        1     1     9 ...          3          3        6       6     1      1      9      1
8141   1         1           1       1     1    1          1        1     1     1 ...          3          3        6       6     1      1      1      1
8142   1         2           2       2     1    2          1        1     1     2 ...          3          3        6       6     1      1      2      1
8143   1         3           3       3     1    3          1        1     1     3 ...          3          3        6       6     1      1      3      1
8144   1         4           4       4     1    4          1        1     1     4 ...          3          3        6       6     1      1      4      1
8145   1         5           5       5     1    5          1        1     1     5 ...          3          3        6       6     1      1      5      1
8146   1         6           6       6     1    6          1        1     1     6 ...          3          3        6       6     1      1      6      1
8147   1         7           7       7     1    7          1        1     1     7 ...          3          3        6       6     1      1      7      1
8148   1         8           8       8     1    8          1        1     1     8 ...          3          3        6       6     1      1      8      1
8149   1         9           9       9     1    9          1        1     1     9 ...          3          3        6       6     1      1      9      1
8150   1         1           1       1     1    1          1        1     1     1 ...          3          3        6       6     1      1      1      1
8151   1         2           2       2     1    2          1        1     1     2 ...          3          3        6       6     1      1      2      1
8152   1         3           3       3     1    3          1        1     1     3 ...          3          3        6       6     1      1      3      1
8153   1         4           4       4     1    4          1        1     1     4 ...          3          3        6       6     1      1      4      1
8154   1         5           5       5     1    5          1        1     1     5 ...          3          3        6       6     1      1      5      1
8155   1         6           6       6     1    6          1        1     1     6 ...          3          3        6       6     1      1      6      1
8156   1         7           7       7     1    7          1        1     1     7 ...          3          3        6       6     1      1      7      1
8157   1         8           8       8     1    8          1        1     1     8 ...          3          3        6       6     1      1      8      1
8158   1         9           9       9     1    9          1        1     1     9 ...          3          3        6       6     1      1      9      1
8159   1         1           1       1     1    1          1        1     1     1 ...          3          3        6       6     1      1      1      1
8160   1         2           2       2     1    2          1        1     1     2 ...          3          3        6       6     1      1      2      1
8161   1         3           3       3     1    3          1        1     1     3 ...          3          3        6       6     1      1      3      1
8162   1         4           4       4     1    4          1        1     1     4 ...          3          3        6       6     1      1      4      1
8163   1         5           5       5     1    5          1        1     1     5 ...          3          3        6       6     1      1      5      1
8164   1         6           6       6     1    6          1        1     1     6 ...          3          3        6       6     1      1      6      1
8165   1         7           7       7     1    7          1        1     1     7 ...          3          3        6       6     1      1      7      1
8166   1         8           8       8     1    8          1        1     1     8 ...          3          3        6       6     1      1      8      1
8167   1         9           9       9     1    9          1        1     1     9 ...          3          3        6       6     1      1      9      1
8168   1         1           1       1     1    1          1        1     1     1 ...          3          3        6       6     1      1      1      1
8169   1         2           2       2     1    2          1        1     1     2 ...          3          3        6       6     1      1      2      1
8170   1         3           3       3     1    3          1        1     1     3 ...          3          3        6       6     1      1      3      1
8171   1         4           4       4     1    4          1        1     1     4 ...          3          3        6       6     1      1      4      1
8172   1         5           5       5     1    5          1        1     1     5 ...          3          3        6       6     1      1      5      1
8173   1         6           6       6     1    6          1        1     1     6 ...          3          3        6       6     1      1      6      1
8174   1         7           7       7     1    7          1        1     1     7 ...          3          3        6       6     1      1      7      1
8175   1         8           8       8     1    8          1        1     1     8 ...          3          3        6       6     1      1      8      1
8176   1         9           9       9     1    9          1        1     1     9 ...          3          3        6       6     1      1      9      1
8177   1         1           1       1     1    1          1        1     1     1 ...          3          3        6       6     1      1      1      1
8178   1         2           2       2     1    2          1        1     1     2 ...          3          3        6       6     1      1      2      1
8179   1         3           3       3     1    3          1        1     1     3 ...          3          3        6       6     1      1      3      1
8180   1         4           4       4     1    4          1        1     1     4 ...          3          3        6       6     1      1      4      1
8181   1         5           5       5     1    5          1        1     1     5 ...          3          3        6       6     1      1      5      1
8182   1         6           6       6     1    6          1        1     1     6 ...          3          3        6       6     1      1      6      1
8183   1         7           7       7     1    7          1        1     1     7 ...          3          3        6       6     1      1      7      1
8184   1         8           8       8     1    8          1        1     1     8 ...          3          3        6       6     1      1      8      1
8185   1         9           9       9     1    9          1        1     1     9 ...          3          3        6       6     1      1      9      1
8186   1         1           1       1     1    1          1        1     1     1 ...          3          3        6       6     1      1      1      1
8187   1         2           2       2     1    2          1        1     1     2 ...          3          3        6       6     1      1      2      1
8188   1         3           3       3     1    3          1        1     1     3 ...          3          3        6       6     1      1      3      1
8189   1         4           4       4     1    4          1        1     1     4 ...          3          3        6       6     1      1      4      1
8190   1         5           5       5     1    5          1        1     1     5 ...          3          3        6       6     1      1      5      1
8191   1         6           6       6     1    6          1        1     1     6 ...          3          3        6       6     1      1      6      1
8192   1         7           7       7     1    7          1        1     1     7 ...          3          3        6       6     1      1      7      1
8193   1         8           8       8     1    8          1        1     1     8 ...          3          3        6       6     1      1      8      1
8194   1         9           9       9     1    9          1        1     1     9 ...          3          3        6       6     1      1      9      1
8195   1         1           1       1     1    1          1        1     1     1 ...          3          3        6       6     1      1      1      1
8196   1         2           2       2     1    2          1        1     1     2 ...          3          3        6       6     1      1      2      1
8197   1         3           3       3     1    3          1        1     1     3 ...          3          3        6       6     1      1      3      1
8198   1         4           4       4     1    4          1        1     1     4 ...          3          3        6       6     1      1      4      1
8199   1         5           5       5     1    5          1        1     1     5 ...          3          3        6       6     1      1      5      1
8200   1         6           6       6     1    6          1        1     1     6 ...          3          3        6       6     1      1      6      1
8201   1         7           7       7     1    7          1        1     1     7 ...          3          3        6       6     1      1      7      1
8202   1         8           8       8     1    8          1        1     1     8 ...          3          3        6       6     1      1      8      1
8203   1         9           9       9     1    9          1        1     1     9 ...          3          3        6       6     1      1      9      1
8204   1         1           1       1     1    1          1        1     1     1 ...          3          3        6       6     1      1      1      1
8205   1         2           2       2     1    2          1        1     1     2 ...          3          3        6       6     1      1      2      1
8206   1         3           3       3     1    3          1        1     1     3 ...          3          3        6       6     1      1      3      1
8207   1         4           4       4     1    4          1        1     1     4 ...          3          3        6       6     1      1      4      1
8208   1         5           5       5     1    5          1        1     1     5 ...          3          3        6       6     1      1      5      1
8209   1         6           6       6     1    6          1        1     1     6 ...          3          3        6       6     1      1      6      1
8210   1         7           7       7     1    7          1        1     1     7 ...          3          3        6       6     1      1      7      1
8211   1         8           8       8     1    8          1        1     1     8 ...          3          3        6       6     1      1      8      1
8212   1         9           9       9     1    9          1        1     1     9 ...          3          3        6       6     1      1      9      1
8213   1         1           1       1     1    1          1        1     1     1 ...          3          3        6       6     1      1      1      1
8214   1         2           2       2     1    2          1        1     1     2 ...          3          3        6       6     1      1      2      1
8215   1         3           3       3     1    3          1        1     1     3 ...          3          3        6       6     1      1      3      1
8216   1         4           4       4     1    4          1        1     1     4 ...          3          3        6       6     1      1      4      1
8217   1         5           5       5     1    5          1        1     1     5 ...          3          3        6       6     1      1      5      1
8218   1         6           6       6     1    6          1        1     1     6 ...          3          3        6       6     1      1      6      1
8219   1         7           7       7     1    7          1        1     1     7 ...          3          3        6       6     1      1      7      1
8220   1         8           8       8     1    8          1        1     1     8 ...          3          3        6       6     1      1      8      1
8221   1         9           9       9     1    9          1        1     1     9 ...          3          3        6       6     1      1      9      1
8222   1         1           1       1     1    1          1        1     1     1 ...          3          3        6       6     1      1      1      1
8223   1         2           2       2     1    2          1        1     1     2 ...          3          3        6       6     1      1      2      1
8224   1         3           3       3     1    3          1        1     1     3 ...          3          3        6       6     1      1      3      1
8225   1         4           4       4     1    4          1        1     1     4 ...          3          3        6       6     1      1      4      1
8226   1         5           5       5     1    5          1        1     1     5 ...          3          3        6       6     1      1      5      1
8227   1         6           6       6     1    6          1        1     1     6 ...          3          3        6       6     1      1      6      1
8228   1         7           7       7     1    7          1        1     1     7 ...          3          3        6       6     1      1      7      1
8229   1         8           8       8     1    8          1        1     1     8 ...          3          3        6       6     1      1      8      1
8230   1         9           9       9     1    9          1        1     1     9 ...          3          3        6       6     1      1      9      1
8231   1         1           1       1     1    1          1        1     1     1 ...          3          3        6       6     1      1      1      1
8232   1         2           2       2     1    2          1        1     1     2 ...          3          3        6       6     1      1      2      1
8233   1         3           3       3     1    3          1        1     1     3 ...          3          3        6       6     1      1      3      1
8234   1         4           4       4     1    4          1        1     1     4 ...          3          3        6       6     1      1      4      1
8235   1         5           5       5     1    5          1        1     1     5 ...          3          3        6       6     1      1      5      1
8236   1         6           6       6     1    6          1        1     1     6 ...          3          3        6       6     1      1      6      1
8237   1         7           7       7     1    7          1        1     1     7 ...          3          3        6       6     1      1      7      1
8238   1         8           8       8     1    8          1        1     1     8 ...          3          3        6       6     1      1      8      1
8239   1         9           9       9     1    9          1        1     1     9 ...          3          3        6       6     1      1      9      1
8240   1         1           1       1     1    1          1        1     1     1 ...          3          3        6       6     1      1      1      1
8241   1         2           2       2     1    2          1        1     1     2 ...          3          3        6       6     1      1      2      1
8242   1         3           3       3     1    3          1        1     1     3 ...          3          3        6       6     1      1      3      1
8243   1         4           4       4     1    4          1        1     1     4 ...          3          3        6       6     1      1      4      1
8244   1         5           5       5     1    5          1        1     1     5 ...          3          3        6       6     1      1      5      1
8245   1         6           6       6     1    6          1        1     1     6 ...          3          3        6       6     1      1      6      1
8246   1         7           7       7     1    7          1        1     1     7 ...          3          3        6       6     1      1      7      1
8247   1         8           8       8     1    8          1        1     1     8 ...          3          3        6       6     1      1      8      1
8248   1         9           9       9     1    9          1        1     1     9 ...          3          3        6       6     1      1      9      1
8249   1         1           1       1     1    1          1        1     1     1 ...          3          3        6       6     1      1      1      1
8250   1         2           2       2     1    2          1        1     1     2 ...          3          3        6       6     1      1      2      1
8251   1         3           3       3     1    3          1        1     1     3 ...          3          3        6       6     1      1      3      1
8252   1         4           4       4     1    4          1        1     1     4 ...          3          3        6       6     1      1      4      1
8253   1         5           5       5     1    5          1        1     1     5 ...          3          3        6       6     1      1      5      1
8254   1         6           6       6     1    6          1        1     1     6 ...          3          3        6       6     1      1      6      1
8255   1         7           7       7     1    7          1        1     1     7 ...          3          3        6       6     1      1      7      1
8256   1         8           8       8     1    8          1        1     1     8 ...          3          3        6       6     1      1      8      1
8257   1         9           9       9     1    9          1        1     1     9 ...          3          3        6       6     1      1      9      1
8258   1         1           1       1     1    1          1        1     1     1 ...          3          3        6       6     1      1      1      1
8259   1         2           2       2     1    2          1        1     1     2 ...          3          3        6       6     1      1      2      1
8260   1         3           3       3     1    3          1        1     1     3 ...          3          3        6       6     1      1      3      1
8261   1         4           4       4     1    4          1        1     1     4 ...          3          3        6       6     1      1      4      1
8262   1         5           5       5     1    5          1        1     1     5 ...          3          3        6       6     1      1      5      1
8263   1         6           6       6     1    6          1        1     1     6 ...          3          3        6       6     1      1      6      1
8264   1         7           7       7     1    7          1        1     1     7 ...          3          3        6       6     1      1      7      1
8265   1         8           8       8     1    8          1        1     1     8 ...          3          3        6       6     1      1      8      1
8266   1         9           9       9     1    9          1        1     1     9 ...          3          3        6       6     1      1      9      1
8267   1         1           1       1     1    1          1        1     1     1 ...          3          3        6       6     1      1      1      1
8268   1         2           2       2     1    2          1        1     1     2 ...          3          3        6       6     1      1      2      1
8269   1         3           3       3     1    3          1        1     1     3 ...          3          3        6       6     1      1      3      1
8270   1         4           4       4     1    4          1        1     1     4 ...          3          3        6       6     1      1      4      1
8271   1         5           5       5     1    5          1        1     1     5 ...          3          3        6       6     1      1      5      1
8272   1         6           6       6     1    6          1        1     1     6 ...          3          3        6       6     1      1      6      1
8273   1         7           7       7     1    7          1        1     1     7 ...          3          3        6       6     1      1      7      1
8274   1         8           8       8     1    8          1        1     1     8 ...          3          3        6       6     1      1      8      1
8275   1         9           9       9     1    9          1        1     1     9 ...          3          3        6       6     1      1      9      1
8276   1         1           1       1     1    1          1        1     1     1 ...          3          3        6       6     1      1      1      1
8277   1         2           2       2     1    2          1        1     1     2 ...          3          3        6       6     1      1      2      1
8278   1         3           3       3     1    3          1        1     1     3 ...          3          3        6       6     1      1      3      1
8279   1         4           4       4     1    4          1        1     1     4 ...          3          3        6       6     1      1      4      1
8280   1         5           5       5     1    5          1        1     1     5 ...          3          3        6       6     1      1      5      1
8281   1         6           6       6     1    6          1        1     1     6 ...          3          3        6       6     1      1      6      1
8282   1         7           7       7     1    7          1        1     1     7 ...          3          3        6       6     1      1      7      1
8283   1         8           8       8     1    8          1        1     1     8 ...          3          3        6       6     1      
```

de validation est utilisé pour ajuster les hyperparamètres du modèle et l'ensemble de test est utilisé pour évaluer les performances finales du modèle sur des données invisibles pendant l'entraînement. Cette division garantit que le modèle est évalué de manière impartiale et permet d'estimer sa capacité à généraliser à de nouvelles données.

Data Splitting

```
In [119]: #feature  
X=dataset.drop("class",axis='columns')  
#target  
Y=dataset["class"]  
  
In [120]: from sklearn.model_selection import train_test_split  
  
In [121]: X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.3)
```

FIGURE 3.4 – Code de Division des Données

3.5 Modèles d’Entrainement

3.5.1 Support Vector Machine

```
In [125]: from sklearn.svm import SVC  
svc=SVC()  
  
In [126]: #training  
svc.fit(X_train,Y_train)  
  
Out[126]: SVC()
```

FIGURE 3.5 – Code d’Entrainement de SVM

3.5.2 K-Nearest Neighbors

```
In [50]: from sklearn.neighbors import KNeighborsClassifier  
  
In [51]: knn= KNeighborsClassifier(n_neighbors=90)#sqrt(8124)=90  
  
In [52]: knn.fit(X_train,Y_train)  
  
Out[52]: KNeighborsClassifier(n_neighbors=90)
```

FIGURE 3.6 – Code d’Entrainement de KNN

3.5.3 Logistic Regression

```
In [67]: from sklearn.linear_model import LogisticRegression
In [68]: log_reg = LogisticRegression()
In [69]: log_reg.fit(X_train, Y_train)
C:\Users\pc\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
    n_iter_i = _check_optimize_result()

Out[69]: LogisticRegression()
LogisticRegression()
```

FIGURE 3.7 – Code d’Entraînement de Régression Logistique

3.6 Matrice de Confusion

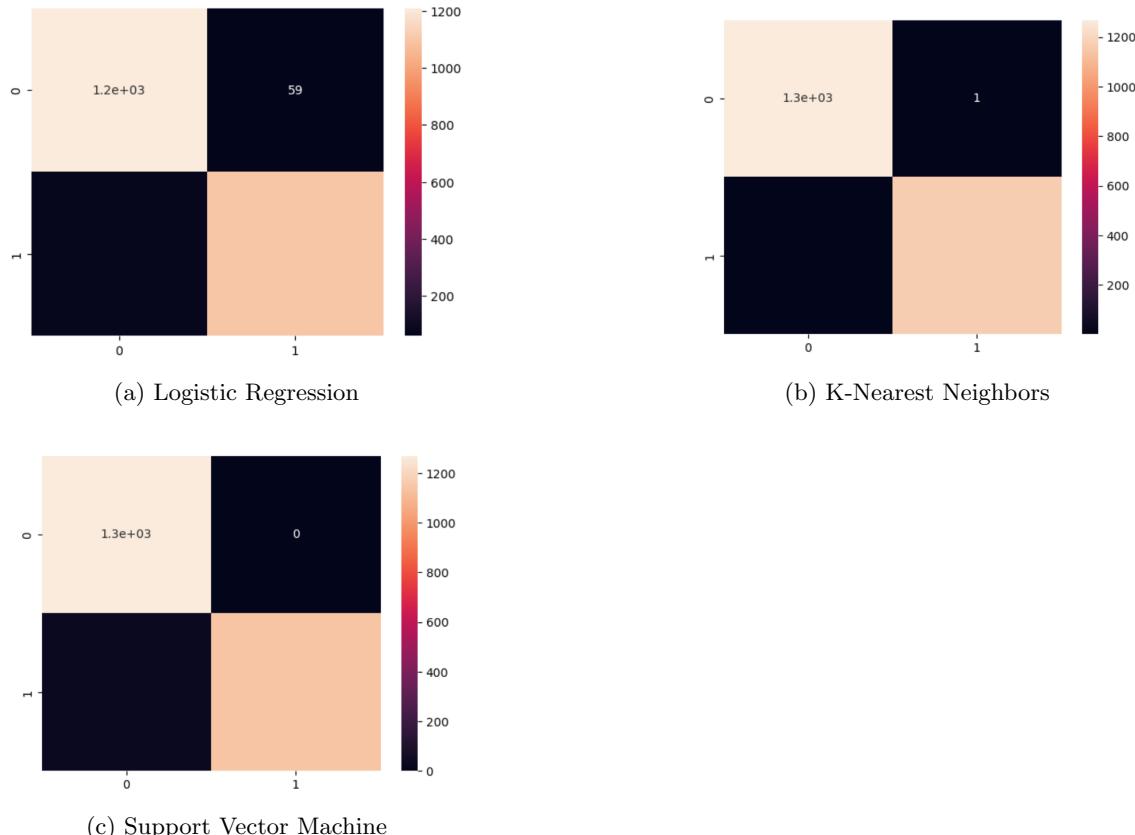


FIGURE 3.8 – Métriques de Confusion

CHAPITRE 4

IMPLÉMENTATION

4.1 Outils Pour le Développement D'applications Web

4.1.1 Flask



Flask est un framework web léger et flexible écrit en Python. Il est conçu pour faciliter le développement rapide d'applications web, en offrant une structure simple et modulaire tout en étant extensible selon les besoins du projet. Flask fournit les outils de base nécessaires pour créer des applications web, tout en permettant aux développeurs de choisir et d'intégrer les extensions et les bibliothèques qui conviennent le mieux à leur projet. Avec sa conception minimalist mais puissante, Flask est largement utilisé pour construire une variété d'applications web, des plus simples aux plus complexes, et reste populaire dans la communauté Python pour sa simplicité d'utilisation et sa flexibilité.

4.1.2 Bibliothèques Cruciales

Scikit-learn



Scikit-Learn est une bibliothèque open-source de machine learning en Python qui offre une grande variété d'algorithmes d'apprentissage supervisé et non supervisé.

Elle fournit également des outils pour la préparation des données, l'évaluation des modèles, et le réglage des hyperparamètres. Dans notre projet de classification de mushroom, Scikit-Learn serait utile pour l'entraînement et l'évaluation des modèles de classification, en utilisant des algorithmes tels que les SVM, ou les méthodes ensemblistes.

Pandas



Pandas est une bibliothèque open-source en Python destinée à la manipulation et à l'analyse de données. Elle offre des structures de données puissantes, notamment les DataFrame, qui permettent de travailler efficacement avec des données tabulaires. Dans notre projet, Pandas serait utilisé pour charger, nettoyer, et préparer les données sur les champignons à partir de sources de données brutes, ainsi que pour effectuer des opérations de transformation et d'agrégation sur les données.

Numpy



NumPy est une bibliothèque fondamentale en Python pour le calcul numérique. Elle fournit des structures de données efficaces pour travailler avec des tableaux multidimensionnels, ainsi qu'une grande variété de fonctions mathématiques pour effectuer des opérations sur ces tableaux. Dans notre projet, NumPy serait utilisé pour représenter les données sous forme de tableaux, ainsi que pour effectuer des opérations de manipulation de données et de calculs statistiques nécessaires à la préparation des données et à l'entraînement des modèles de classification.

4.2 Interface Web



Nous avons développé une interface web intuitive pour permettre la prédiction et la classification des champignons en utilisant cinq caractéristiques clés : les "bruises" (lésions), l'espacement des lames ("gill spacing"), la couleur des lames ("gill color"), la taille des lames ("gill size"), et le type d'anneau ("ring type"). L'utilisateur peut sélectionner les valeurs de ces caractéristiques à partir d'une liste déroulante correspondante, fournissant ainsi les données nécessaires pour l'analyse. Une fois les paramètres choisis, notre interface utilise un modèle pré-entraîné pour prédire ou classifier le champignon en question. Cette approche simplifiée rend l'outil accessible même pour les utilisateurs non techniques, tout en offrant une manière efficace et fiable de tirer des conclusions sur la nature des champignons.

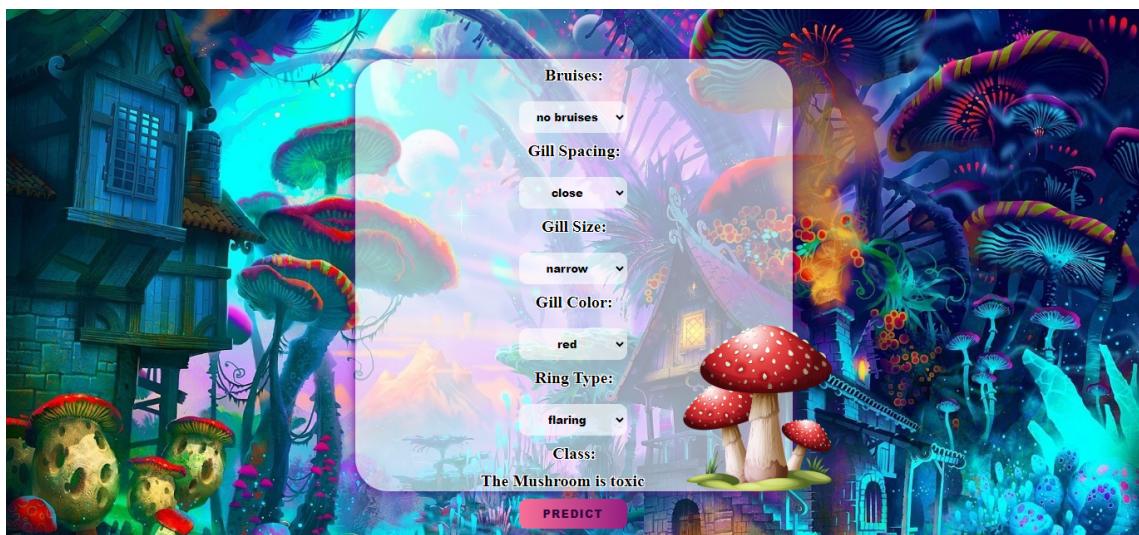


FIGURE 4.1 – Résultat de Mushroom : "Toxic"

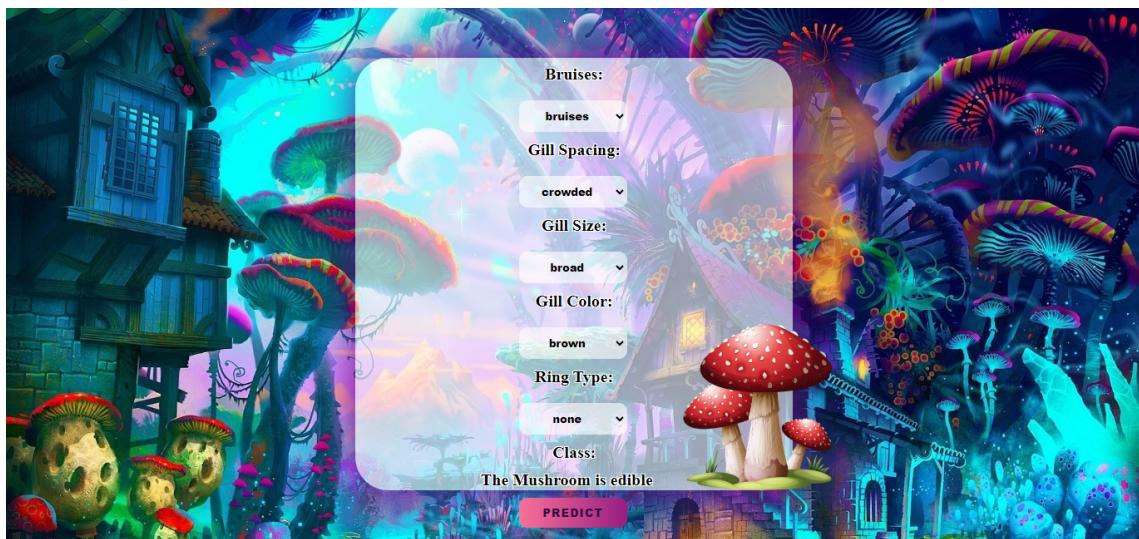


FIGURE 4.2 – Résultat de Mushroom : "Edible"

CONCLUSION

En conclusion, ce projet de classification de champignons a été une expérience enrichissante dans le domaine de l'apprentissage automatique. Il nous a offert l'opportunité de plonger dans le vaste univers de l'intelligence artificielle et de comprendre les tenants et aboutissants de l'apprentissage automatique appliqué. Tout au long du processus, nous avons été confrontés à divers défis, de la sélection des entrées et des sorties à la collecte et au nettoyage des données, en passant par le choix des algorithmes et leur mise en œuvre.

En parallèle, nous avons développé une application web simple pour mettre en pratique les prédictions de classification de champignons. Cette application nous a permis de visualiser de manière concrète les résultats de nos modèles et de les tester avec différentes combinaisons de caractéristiques.

Ce projet nous a également permis de développer de nombreuses compétences techniques, telles que la recherche, la programmation en Python et la manipulation de données. Nous avons acquis une compréhension approfondie des principes fondamentaux du machine learning et de leur application pratique. De plus, nous avons appris à utiliser des outils et des bibliothèques populaires comme scikit-learn pour construire et évaluer nos modèles de classification.

BIBLIOGRAPHIE

- [1] Rahma Nabil ,Said Ahmed Sallam, Eman A.Shehab ,Sara A.Shehab -The classification of mushroom using ML [[Lien](#)]
- [2] Balika J. Chelliah- Classification of Mushrooms using Supervised Learning Models [[Lien](#)]
- [3] Mohammad Ashraf Ottom , Noor Aldeen Alawad, Khalid M.O .Nahar- Classification of Mushroom Fungi Using Machine Learning Techniques [[Lien](#)]
- [4] JAYAPUNITHA,JEMMY JASMIN - MUSHROOM CLASSIFICATION USING LOGISTICREGRESSION ALGORITHM [[Lien](#)]