Description et justification de la méthodologie :

ZAOUAM Sirageddine Farhi Abdellah Hadj-Said Mohand

I. INTRODUCTION :

Dans la première partie on s'est intéressé principalement à l'analyse de données, on a étudié les relations possibles entre les paramètres du problème, pour cela on a utilisé la matrice de corrélation et on a pu découvrir qu'il ya une forte corrélation entre les variables explicatives et la variable expliqué , donc on a transformé notre problème en un problème de classification binaire dont la variable Class est la variable à prédire. Dans cette partie on va choisir 3 algorithmes de classification qu'on va étudier par la suite, le but est donc de décrire le protocole qu'on va utiliser pour les comparer (sélection des hyper-paramètres , évaluation des erreurs , comparaison des précisions à l'aide de la matrice de confusion)

II . CHOIX ET JUSTIFICATION DES MODÈLES D'APPRENTISSAGE :

Nous avons un problème de classification, en plus que ça nos données sont étiquités : i.e pour chaque produit chimique on connaît sa classe correspondante (1, 2 ou3) donc pour résoudre ce problème , il existe plusieurs méthodes de classification supervisé. Dans ce projet on va etre limité à trois modèles, en se basant sur notre cours nous avons choisis de tester les algorithmes suivants : Le K-Nearest-Neighbors (KNN), Decision Tree (DT) et le Support Vector Machines (SVM)

1) Le K-Nearest Neighbors :

L'algorithme des K plus proches voisins est un algorithme d'apprentissage supervisé, le principe de ce modèle consiste en effet à choisir les K données les plus proche du point étudié afin d'en prédire sa classe. Le choix de cet algorithme se justifie sur la base de notre analyse de donnée puisqu'il est très efficace pour la classification binaire (Si le produit chimique est de la classe 1 ou pas) , ainsi ce modèle est très rapide , simple et facile à mettre en œuvre (il prend en entrée un seul paramètre k)

Ce dernier reçoit un ensemble de données étiquetés avec des valeurs de sorties correspondantes sur lequel il va pouvoir s'entraîner et définir un modèle de prédiction. Cet algorithme pourra par la suite être utilisé sur de nouvelles données afin de prédire leurs valeurs de sorties correspondantes.

- un ensemble de données D .
- une fonction de définition distance d : la distance euclidienne
- Un nombre entier K

2) L'arbre de décision :

L' arbre de décision fait partie des méthodes d'apprentissage supervisé, et fait à ce titre partie de la boîte à outils du parfait petit dataminer. elle vise à prédire les valeurs prises par une variable en fonction d'un jeu de variables d'entrée. Cette prédiction se fait à travers la construction d'un arbre dont chaque nœud correspond à une décision quant à la valeur de la variable à prédire.

Le choix de cet algorithme se justifie de sa efficacité et sa capacité de sélectionner les meilleurs caractéristiques, dans notre dataset on a bien remarqué qu'il pas mal de variables qui ont une très faible corrélation avec notre variable expliqué donc cet algorithme va nous permettre de diminuer la dimension en sélectionnant que les meilleures caractéristiques calculés à partir d'une fonction .

Plus on a de nœuds, plus notre arbre décisionnel sera précis (en général) mais les arbres décisionnels ils font un peu défaut lorsqu'on parle de précision ou d'exactitude.

Données en entrée : un ensemble de données D , la profondeur maximale de l'arbre (max_depth) et le nombre minimale d'échantillons dans les feuilles (min_samples_leaf)

3) Support Vector Machines (SVM) :

L'algorithme Support Vector Machines est un algorithme d'apprentissage supervisé qui est connu pour être efficace pour des problèmes de classification binaire. Lors de notre description de donnée on a pu remarquer qu'il existe pas mal de règles qui peuvent être définies entre les attributs et qui permettent

de les distinguer donc il est possible de trouver un hyperplan qui nous permettra de distinguer entre les classes.

III. CHOIX ET JUSTIFICATION DES MÉTHODES D'ÉVALUATION :

Dans cette section, on va introduire notre méthodologie qu'on va suivre pour l'évaluation et la comparaison des différents modèles.

1) Sélection des modèles :

Pour choisir l'algorithme le plus performant il faudra d'abord étudier les hyper-paramètres pour chacun de ces modèles, pour cela on va utiliser la grille d'hyper-paramètres (**Grid search)** qui nous permet de tester plusieurs valeurs pour chaque hyper-paramètre respectivement pour chaque modèle. On essayera de trouver le meilleur nombre de voisin K pour le **KNN**, ainsi que la meilleure profondeur maximale (**max_depth**) et le nombre minimale d 'échantillons dans les feuilles pour l'**arbre de décision** et finalement on essaiera de tester plusieurs types de noyaux pour l'algorithme **SVM**.
Une deuxième technique est la **méthode de validation croisé** (Cross validation) afin de mieux évaluer les modèles , on peut séparer les données uniquement en deux parties : un jeu d'entraînement et un jeu de test . On fera ensuite une validation croisé sur le jeu d'entraînement pour évaluer les trois modèles et déduire lequel est plus performant sur l'ensemble de jeu d'entraînement qu'on va tester ensuite sur l'ensemble de jeu de test. Par exemple pour le KNN on peut s'intéresser à un moyen de choisir le K pour lequel la classification sera la meilleure. Une façon de le trouver consiste à utiliser la méthode de validation croisée avec **GridSearchCV** pour tracer le graphique de la valeur K et le taux d'erreur correspondant pour l'ensemble de données. De même pour l'arbre de décision en traçant le graphe de la variation de la profondeur de l'arbre en fonction du score ou de l'erreur.

2) Évaluation et comparaison des modèles :

Dans cette partie on va étudier les différentes manières d'évaluer un modèle de classification :

2.1 La matrice de confusion et les indicateurs de performance :

La matrice de confusion peut être utilisée pour évaluer un classificateur, sur la base d'un ensemble de données de test pour lesquelles les vraies valeurs sont connues. C'est un outil simple, qui aide à donner un bon aperçu visuel des performances de l'algorithme utilisé.
Une matrice de confusion est représentée sous forme de tableau. Dans cet exemple, nous allons examiner une matrice de confusion pour un classificateur binaire .

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

On peut extraire de cette matrice différentes mesures de performance, comme le **rappel** ou la sensibilité (taux de vrais positifs) la **précision**, la **spécificité** (taux de vrais négatifs) et le **F-score**, qui reflètent différents aspects du modèle.

Figure 1 : Matrice de confusion d'une classification binaire

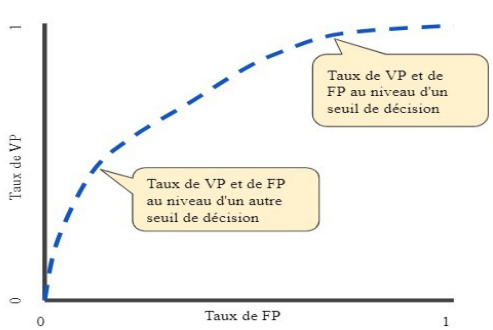
$$Précision = \frac{TP}{TP + FP} \qquad Rappel = \frac{TP}{TP + FN} \qquad Spécificité = \frac{TN}{FP + TN}$$

On peut donc calculer pour chacun des trois modèles (KNN, SVM et l'arbre de décision) leur matrice de confusion, ainsi les indicateurs de performances afin d'en déduire le modèle qui nous garantit plus la précision (la proportion des prédictions correctes parmis les points qu'on a prédit positifs) , la sensibilité ou bien celui qui donne le moins d'erreurs.

2.2 La Courbe ROC et AUC :

Une courbe ROC est un graphique représentant les performances d'un modèle de classification Cette dernière trace le taux devrais positifs en fonction du taux de faut négatifs :
La courbe ROC dans la **figure 2** nous permettre de trouver le seuil optimale de classifications pour chaque modèle.

Figure 2 : Courbe ROC



Références :

Precision, Recall, Confusion and Matrices in Machine Learning (Bmc blogs)
Evaluer les performances d'un modèle de machine learning (OPENCLASSROOMS)
Classification ROC et AUC https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc?hl=f