Notes générales SY09 :

Test de shapiro : on récupère les quantiles empiriques sur lesquels on émet deux hypthèse : H0 qu’ils suivent une loi normale, H1 qu’ils la suivent pas.  La p-valeur est obtenue en comparant la statistique WWW à la distribution théorique de WWW sous l'hypothèse de normalité.

Une p-valeur élevée (généralement > 0.05) indique que les données ne s'écartent pas significativement de la normalité.

Une p-valeur faible (≤ 0.05) suggère que les données s'écartent significativement de la normalité.

Pour la LDA : résultats nuls car carbone organique a une super faible corrélation avec la variable à prédire, et aussi avec l’autre variable suivant une loi normale : la conductivité

Si on accepte l’hypothèse cependant que toutes nos variables suivent une loi normale, on obtient de meilleur résultat car on détient bien plus d’information.  
Si on faisait un test de shapiro moins « severe » donc qui accepterait davantage des variables qui suivent moins une loi normale, alors dans ce cas là on aurait réaliser notre analyse sur davantage de données qui du coup vu qu’elles détiennent davantage d’information, donne de meilleurs résultats.

Régression logistique est sensible aux points abberants, c’est pourquoi dans le dataset corrigé elle n’est pas très efficace (voir courbe pour bien comprendre)

Un F1-score de 57,6% pour les échantillons d'eau potable signifie que le modèle a un équilibre modéré entre la précision et le rappel, mais pas optimal, car ni la précision ni le rappel ne sont particulièrement élevés.

Donc le modèle n’est pas très bon en réalité :

Il réussit mieux à éviter de fausses identifications d'échantillons non potables comme potables, mais il échoue souvent à reconnaître les échantillons qui sont effectivement potables. Un rappel de 48.5% (Seulement 48,5% des échantillons réellement potables sont correctement identifiés comme tels ) est quand même très faible, ce qui fait que ce classifieur n’est pas à considérer.

**Évaluation des Modèles sur d'Autres Jeux de Données**

* **ACP** : Ne pas afficher les résultats basés sur l'Analyse en Composantes Principales (ACP) car ils ne sont pas très représentatifs.
* **Approche Alternée** :
  + **Comparaison** : Évaluer les modèles sur d'autres datasets ayant les mêmes colonnes.
  + **Observation** : Bien que nombreux datasets Kaggle existent pour la prédiction de la potabilité, ils proviennent tous du même jeu de données initial, mais scindés différemment (train/test).
  + **Impact** : Nos modèles de prédiction sur les données corrigées sont moins performants car évalués sur les mêmes variables d'apprentissage.

**2. Comparaison des Classificateurs**

* **Tableau de Comparaison** : Créer un tableau pour comparer les performances des classificateurs sur les données corrigées et non corrigées.

**3. Différences entre Régression et Classification**

* **Régression** : Prédit une valeur continue.
* **Classification** : Prédit l'appartenance à une classe (0 ou 1).

**4. Apprentissage Supervisé de Classification**

* **Arbres de Décision** :
  + **Type** : Apprentissage supervisé.
  + **Usage** : Classification.
* **Régression Logistique** :
  + **Type** : Apprentissage supervisé.
  + **Usage** : Classification binaire.
* **K-Nearest Neighbors (KNN)** :
  + **Type** : Apprentissage supervisé.
  + **Usage** : Classification et régression.
  + **Principe** : Classe les points en fonction de la majorité des k voisins les plus proches.
* **Décision de Bayes** :
  + **Type** : Apprentissage supervisé.
  + **Usage** : Classification.
  + **Principe** : Utilise la règle de Bayes pour classer les instances en maximisant la probabilité a posteriori.
* **Analyse Discriminante Linéaire (LDA)** :
  + **Type** : Apprentissage supervisé.
  + **Usage** : Classification.
  + **Principe** : Trouve une combinaison linéaire de caractéristiques qui sépare le mieux les classes.

**Régression Multiple**:

**Usage**: Régression (on peut pas nous car on a une prédiction binaire)

**2. Apprentissage Non Supervisé**

* **K-means** :
  + **Type** : Apprentissage non supervisé.
  + **Usage** : Clustering.
  + **Principe** : Partitionne les données en k clusters en minimisant la variance intra-cluster.
* **Théorie Bayésienne** :
  + **Type** : Peut être supervisé ou non supervisé.
  + **Usage** : Classification et régression.
  + **Principe** : Basé sur le théorème de Bayes, utilisé pour faire des inférences statistiques sur les données.

**Arbres de décision / Random forest :**

**1. Overfitting et Arbres de Décision**

* **Problème** : Les arbres de décision sont très sujets à l'overfitting. Car Cette flexibilité permet aux arbres de décision de capturer parfaitement les variations des données d'entraînement, y compris le bruit et les anomalies, plutôt que de se concentrer uniquement sur les tendances générales. Cela conduit à une faible généralisation sur les données de test ou nouvelles données.
* Lorsque les arbres de décision sont autorisés à croître sans restriction, ils deviennent très profonds et complexes, avec de nombreux nœuds et branches. Chaque nœud de l'arbre est une décision basée sur une caractéristique des données d'entraînement, ce qui signifie que l'arbre peut devenir extrêmement spécifique aux données d'entraînement. En conséquence, l'arbre peut sur-apprendre les détails spécifiques de l'ensemble de formation, ce qui ne se traduit pas bien sur des données non vues.
* **Solutions** :
  + **Limiter le nombre de feuilles** : Réduire la profondeur de l'arbre.
  + **Bagging/Forêt d'arbres** : Combiner plusieurs arbres pour réduire l'overfitting.

**Elagage**: L'objectif principal de l'élagage est de retirer les branches de l'arbre qui sont peu utiles ou qui peuvent avoir été construites en raison de bruit dans les données d'entraînement. En supprimant ces branches, l'arbre devient moins complexe et est moins susceptible de sur-apprendre les détails spécifiques des données d'entraînement (réduisant ainsi l'overfitting).

**2. Bagging**

* **Définition** : Bagging (Bootstrap Aggregating) consiste à entraîner plusieurs modèles sur des sous-échantillons différents du jeu de données original, créés par resampling (tirage avec remise).
* **Processus** :
  + Tirer aléatoirement des sous-échantillons du jeu de données.
  + Construire des arbres de décision indépendants sur chaque sous-échantillon.
  + Moyenne ou vote des prédictions des arbres pour obtenir une prédiction finale.

**3. RandomForest**

* **Définition** : Une extension du bagging où chaque arbre est construit en utilisant un sous-ensemble aléatoire de caractéristiques.
* **Processus** :
  + **Sampling des données** : Tirage avec remise pour créer des jeux de données différents pour chaque arbre.
  + **Sélection des caractéristiques** : Pour chaque arbre, sélectionner aléatoirement un sous-ensemble de caractéristiques. Par exemple, si l'arbre original utilise 10 caractéristiques, on peut en sélectionner 5 différentes pour chaque arbre.
* **Avantages** :
  + **Diminution de l'invariabilité** : Diversité des arbres réduit la variance du modèle final.
  + **Performance accrue** : Généralement meilleur qu'un seul arbre de décision.
  + **Réduction de l'overfitting**.
* **Inconvénient** :
  + **Interprétabilité réduite** : Plus difficile d'interpréter une forêt de plusieurs arbres qu'un seul arbre de décision.

**4. Critère de Division : Indice de Gini**

* **Objectif** : Trouver la division qui maximise le gain de Gini, c'est-à-dire celle qui réduit le plus l'impureté des nœuds.

**5. RandomBagging**

* **Définition** : Pour chaque arbre, prendre différentes caractéristiques.
* **Caractéristiques** :
  + **Caractéristiques aléatoires** : Par exemple, utiliser les 2 premières caractéristiques pour un arbre, puis les deux dernières pour un autre, etc.

**6. Élagage (Pruning)**

* **Objectif** : Réduire la taille des arbres de décision en supprimant des parties de l'arbre qui sont inutiles ou qui contribuent peu à la décision des classes.
* **Méthodes** :
  + **Élagage par coût-complexité** : Réduire la complexité de l'arbre tout en maintenant un bon niveau de performance.
  + **Validation croisée** : Utiliser pour déterminer le niveau optimal de complexité de l'arbre.

**7. Compromis Complexité vs Performance**

* **Arbre complexe et performant** : Peut surapprendre et être moins généralisable.
* **Arbre simple et moins performant** : Moins sujet à l'overfitting mais peut manquer de précision.
* **Objectif** : Minimiser le coût tout en ayant une complexité suffisante pour capturer les relations importantes dans les données.

Indicie de Gini : https://www.youtube.com/watch?v=l-wr6EOypas

**Régression Logistique :**

Classification binaire sur des données discrètes. Comme régression linéaire sauf qu'au lieu d'une droite on aura une sigmoïde. But trouver une fonction sigmoïde qui classe les deux groupes deux points. On va trouver un seuil, par exemple pour tous les pnts au dessus de 0.5, x appartient à y =0, et à y=1 sinon

* **Utilisation** :
  + **Classification** : Utilisée pour prédire la probabilité d'appartenance à une classe (0 ou 1).
  + **Pas appropriée pour les variables très corrélées** : Les variables fortement corrélées peuvent biaiser les coefficients et réduire l'efficacité du modèle.
* **Différence avec la régression linéaire** :
  + **Régression linéaire** : Prédit une valeur continue.
  + **Régression logistique** : Prédit une probabilité, donc une classe.

**2. Transformation des Données**

* **Transformation linéaire** : Applique une transformation linéaire sur les données pour mieux séparer les points entre différentes classes.

**4. Régression Logistique Quadratique**

* **Objectif** : Capturer des relations non linéaires en ajoutant des termes quadratiques et d'interaction.
* **Avantage** : Améliore la capacité du modèle à capturer des relations complexes entre les variables.

**5. Limitations des Clustering et ACP avec Régression Logistique**

* **K-means et ACP** :
  + **Problème** : K-means peut mal regrouper les données si les variables sont mal catégorisées.
  + **Résultat** : Utiliser k-means sur des données transformées avec ACP peut donner des résultats insatisfaisants, résultant en une mauvaise catégorisation.
* **Application de la régression logistique** :
  + **Inadéquation** : Appliquer une régression logistique sur des données mal catégorisées par k-means et ACP n'est pas efficace.

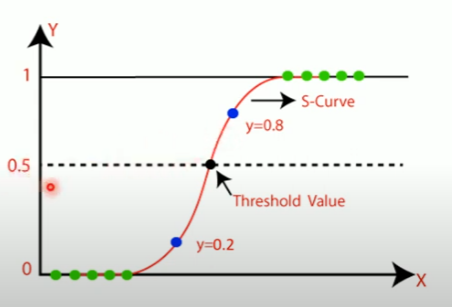
Pourquoi on a une meilleure précision en utilisant toutes les colonnes du dataset :  **Information additionnelle** : Ajouter des variables peut inclure des informations pertinentes, améliorant la précision des prédictions. Retirer des variables peut faire perdre des informations importantes.

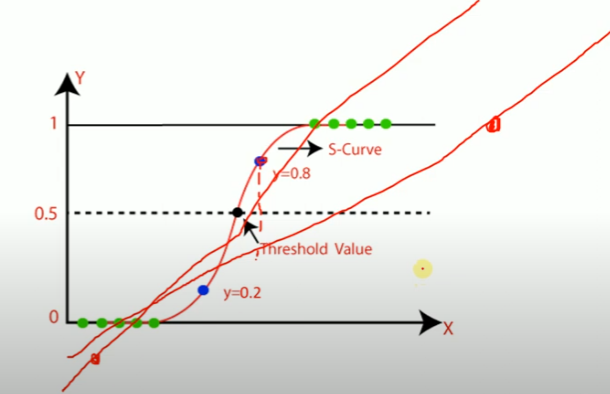
 **Multicolinéarité et redondance** : Même des variables corrélées peuvent fournir des informations utiles en combinaison, améliorant les prédictions.

 **Interactions non linéaires** : Des variables faiblement corrélées individuellement peuvent interagir de manière non linéaire avec d'autres variables pour améliorer les prédictions.

 **Sélection des variables basée sur la corrélation** : La corrélation seule peut ne pas suffire. Des variables peu corrélées linéairement peuvent être utiles dans un modèle plus complexe.

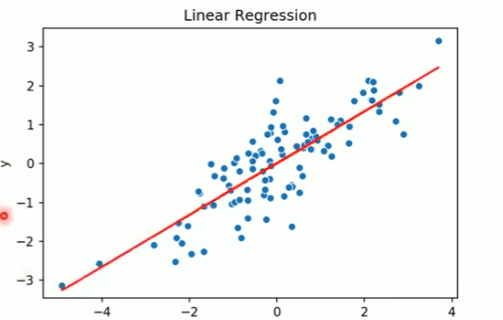
 **Overfitting et régularisation** : Utiliser toutes les variables peut causer du surajustement. La régularisation (Lasso, Ridge) aide à contrôler cela tout en utilisant toutes les informations disponibles.

**Régression logistique** : 

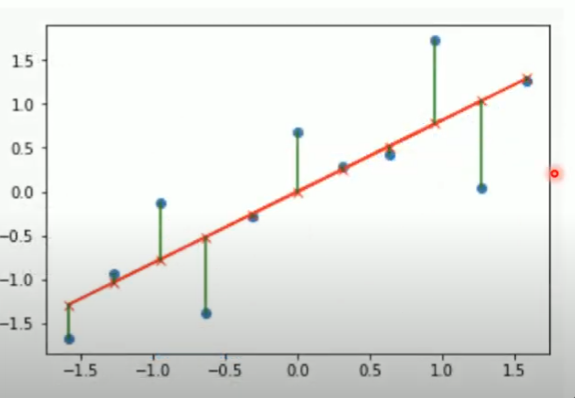
**La régression linéaire est sensible aux points aberrants et si on a des points négatifs on ne saura pas les classer**

Régression Logistique : On calcule le cout pour chaque point. Le but est d’optimiser le cout

**Régression linéaire** : algo supervisé de régression sur des données continues



On veut une droite qui passe par un maximum de points, qui suit la tendance du nuage de points. L’algo va détecter une dépendance entre les variables pour faire des prédictions.

L’algo va trouver les paramètres de la droite . On fait la somme des erreurs et on veut l’erreur la moins grande et ainsi on trouve la droite de régression.

**Décisions de Bayes :**

**Théorie bayésienne et analyse discriminante :**

**Texte supp :**

Nous avons fixé le seuil de p-value à 0,05, conformément aux standards, pour déterminer si nos échantillons suivent une distribution normale.

Nous avons commencé par une approche d'apprentissage non supervisé en utilisant un modèle basé sur les \textit{K-means}. Cependant, ce modèle n'a pas donné de résultats satisfaisants, avec un taux de prédiction ne dépassant pas 60\%.