Projet - Étudier le phénomène du passage de la barrière placentaire par de petits composés.

Analyse de données massives.

AMMICHE Naïma et BEL HAJ HASSINE Wajih

Introduction :

Le placenta est un organe éphémère qui est formé pendant la grossesse pour permettre le développement du fœtus. Il permet un échange entre la mère et le bébé et possède diverses fonctions. Il permet au bébé d'extraire du sang de la mère l'eau, les éléments nutritifs ainsi que l'oxygène nécessaires à son développement. Le placenta fait également office de barrière de protection pour bloquer certaines substances. La prise de médicaments est donc déconseillée lors de la grossesse car les molécules de médicaments risquent de franchir la membrane placentaire qui est une barrière entre le sang de la mère et celui du fœtus via les transferts passifs ou transports actifs, et causer une fausse couche ou encore des problèmes de croissance du fœtus. La clairance permet de mesurer la capacité qu'a ou non un composé à franchir la membrane placentaire. Pour mieux comprendre ce phénomène, nous disposons d'un jeu de données contenant 91 médicaments, leur clairance ainsi que 93 descripteurs physico-chimiques calculés à l'aide du logiciel MOE Molecular Operating Environment. L'objectif de ce projet est d'utiliser des méthodes d'analyse multivariée pour prédire la clairance des médicaments en évaluant si une relation quantitative ou qualitative entre la structure et l'activité peut prédire correctement le transport de médicament et produits chimiques à travers la barrière placentaire humaine, selon leurs propriétés moléculaire, physico-chimique et structurelles. Ces méthodes d'analyse multivariée permettent d'étudier plusieurs variables au même instant et de mettre en évidence des relations de dépendance. Il est donc intéressant de répondre à la problématique suivante. Existe-il une relation quantitative ou qualitative entre la structure et l'activité permettant de prédire la capacité d'un composé à passer la membrane placentaire ?

Présentation des données :

Nous disposons d'un jeu de données contenant 91 médicaments, leur clairance représentée sur deux variables Cl du type quantitatif correspondant à un entier compris entre 0 et 1. La variable Cl_prc du type qualitatif à deux modalités dont 0 si Cl est inférieur à 0.5 indiquant que le composé ne passe pas la membrane placentaire, 1 si Cl est supérieur à 0.5 indiquant que le composé franchit la membrane placentaire. Ce jeu de données comporte également 93 descripteurs physico-chimiques du type quantitatif calculés à l'aide du logiciel MOE. Le jeu de données est analysé via des méthodes d'analyse multivariées avec la programmation R. Le logiciel Rstudio et l'utilisation du langage de programmation R sont indispensables pour permettre l'analyse de ce jeu de données.

Méthodes utilisés :

Premièrement, un nettoyage des données est réalisé afin de retirer les valeurs manquantes et sélectionner les données d'intérêt. Le jeu de données nettoyé est ensuite divisé en deux jeux de données dont un jeu de données (Y) contenant les variables à prédire Cl et Cl_prc correspondant à la clairance décrite précédemment. Le second jeu de données (X) contenant les descripteurs physico-chimiques. Le nettoyage des données permet ensuite depuis le jeu de données X de retirer les descripteurs corrélés avec un seuil de 0.95 ainsi que les descripteurs colinéaires, car la méthode de régression linéaire multiple est sensible à la corrélation et colinéarité entre les descripteurs. Une visualisation des données permet de vérifier que le jeu de données nettoyé ne possède pas de corrélation et colinéarité entre les descripteurs.

Méthode non supervisées :
Une méthode non supervisée est une approche d'apprentissage automatique où l'algorithme sans données étiquetées. L'algorithme doit trouver des modèles et des structures dans les données par lui-même. Les techniques incluent donc la réduction de dimensionnalité et la segmentation de clusters. Nous avons choisi d'utiliser la classification hiérarchique et classification par K-means.

Classification par K-means, K-*means()*:
La méthode de classification K-means permet de classer des groupes de médicaments homogènes et qui différents des autres médicaments qui sont présents dans des groupes différents (cluster). La réalisation d'une classification K-means nécessite l'utilisation de la fonction K-means() de R qui prend en option le jeu de données des descripteurs mis à l'échelle (scale(X)) et l'option center correspondant au nombre de clusters.Le nombre de clusters optimaux est choisi en prenant en compte la variabilité intra-groupe, cette variabilité intra-groupe est obtenue avec la méthode withnss. Nous évaluons ensuite les différentes classifications présentes sur la table de comptage qui est obtenue avec la fonction table(), cette fonction prend en argument les clusters de la classification K-means. Les résultats sont ensuite visualisés via une projection du multidimensional scaling. La fonction cmdscale() prenant en argument la matrice de distance soit dist(scale(X)). La fonction plot() prend le cmdscale ainsi que les clusters du K-means et la visualisation via une projection du multidimensional scaling permettent de distinguer les divers clusters former.

Classification hiérarchique, *hclust()* :
La méthode de classification hclust permet de répertorier de façon hiérarchique (hierarchy d'où h) des composants individuels en groupes de composants similaires (clusters d'où clust). Ces clusters sont déterminés initialement pour être ensuite combinés entre eux jusqu'à obtenir potentiellement un seul cluster. Ces combinaisons sont faites itérativement selon les dissimilarités entre les clusters avec la formule de Lance—Williams et selon les paramètres spécifiés. La classification hclust se fait par la fonction hclust() de R, et ses paramètres sont : d, une matrice de distances obtenue par la fonction dist() appliquée sur le jeu de donnés. Method, la méthode de classification spécifiée. Members, qui est NULL par défaut mais peut être spécifié dans certains cas spécifiques que nous ne verrons pas. Plusieurs fonctions de R peuvent être utilisées pour appréhender les résultats de hclust(), dont print(), plot() et identify.

Méthodes supervisées :
Une méthode supervisée est une technique d'apprentissage automatique où un algorithme est entraîné sur un ensemble de données étiquetées pour prédire une variable cible dans notre cas il s'agit de prédire la clairance pour de nouvelles données.
Les méthodes supervisées sélectionnées sont la régression linéaire multiple et arbres de classification. Un échantillon trin contenant ⅔ médicaments et un échantillon test contenant le ⅓ restant est créé, un apprentissage du modèle sur l'échantillon train suivi d'une validation de ce dernier sur un échantillon de validation est réalisé.

Régression linéaire multiple, *lm()* :
La régression linéaire multiple permet de prédire la clairance à partir des descripteurs connus. Plusieurs étapes sont nécessaires pour la construction du modèle. Des modèles de régression linéaire bivariés entre la variable Cl et chaque descripteur physico-chimique avec une boucle for() permettent de récupérer les p-value qui sont associées à chaque modèle. Les descripteurs

dont le modèle possède une p-value significative au seuil 5% sont conservés. Un histogramme des p-value avec la fonction hist() permet de choisir un seuil. Les descripteurs physico-chimiques pertinents pour prédire la clairance possèdent un seuil significatif inférieur à 5% et sont sélectionnés. Ensuite, la fonction lm() permet de calculer le modèle et prend en arguments l'équation donc la variable à prédire en fonction de tous les descripteurs (Cl ~.), et le data frame utilisé qui est l'échantillon train. Le modèle est visualisé avec la fonction summary(), la qualité du modèle est ensuite vérifiée à partir d'un graphe du modèle, un bon modèle possède des résidus qui suivent une loi normale, qui sont indépendants, possède une même variance (homoscédasticité). Enfin, la fonction predict() permet de calculer les valeurs prédites sont calculées sur l'échantillon de validation. Les performances du modèle sont calculées avec la fonction postResample() qui prend en argument les clairances prédites et clairances observées de l'échantillon test et calcul R^2, R^2_{adj} et RMSE. Une visualisation de la clairance observée en fonction de la clairance prédite pour les échantillons de validation et d'apprentissage est réalisée afin de les comparer, avec la fonction plot().

Arbres de classification, *rpart()*:

La méthode de classification rpart, ou Recursive Partitioning, comme son nom l'indique, partitionne des composants de façon récursive. Elle est ainsi utilisée pour générer des arbres de classification. La classification rpart utilise une librairie de R du même nom, qui contient la fonction rpart(). Les paramètres à spécifier sont formula, la formule qu'on prédit en fonction des autres paramètres ; data, notre jeu de données ; method qu'on spécifie « class » pour classification. Il nous est pertinent d'utiliser également le paramètre minsplit, qui donne le nombre d'observations minimum nécessaires pour que rpart découpe une feuille de l'arbre. Ceci a pour but d'éviter les sur- et sous-apprentissages. Pour choisir le bon niveau de simplification, ou encore le bon nombre de feuilles, nous procédons par validation croisée que rpart() effectue automatiquement. Nous pouvons ensuite évaluer le nombre de mauvaises classifications avec la fonction plotcp(), et visualiser l'arbre obtenu avec la fonction prp().

Résultats :

Le nettoyage des données a permis d'avoir un jeu de données contenant les descripteurs d'intérêt pour réaliser les différentes méthodes permettant d'analyser ce jeu de données et répondre à la problématique. Notre jeu de données contient zéro valeur manquante, de nombreux descripteurs corrélés et colinéaires. Le jeu de données nettoyé contient l'ensemble des molécules-médicaments (91 lignes) et 59 descripteurs physico-chimiques ainsi que deux variables à prédire. Le nettoyage a permis de retirer 34 descripteurs physico-chimiques. Lorsqu'une corrélation est réalisée sur le jeu de données nettoyé, nous n'observons plus de valeurs corrélées ce qui témoigne que le jeu de données est correctement nettoyé.

Classification par K-means, K-means():

La classification K-means requiert le nombre de clusters nécessaires qui correspond au nombre de groupes de médicaments obtenus à l'issue de cette classification.

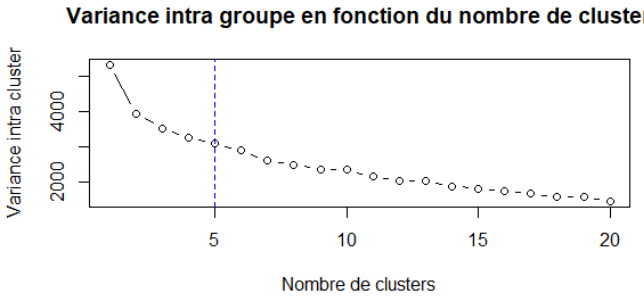


Figure 1: Le graphique représente la variance intra groupe en fonction du nombre optimal de cluster pour une classification K-means.

La figure 1 permet de déterminer visuellement le nombre de clusters nécessaire, nous observons la variance intra groupe en fonction du nombre optimal de clusters. Lorsque le nombre de clusters est faible, la variance intra groupe est élevée et inversement. Le point d'inflexion de la courbe indique le nombre de clusters optimaux, en effet, une augmentation du nombre de clusters au-delà de ce cluster optimal possède une faible variance intra-groupe. Le nombre optimal de cluster est donc 5. La classification K-means est réalisée avec un nombre de clusters de 5, nous analysons ensuite le nombre de médicaments contenus par groupe.

```
 1  2  3  4  5
16 32  9  8 26
```

Figure 2: Table de comptage représentant la taille des clusters pour la méthode K-means.

L'utilisation de l'option nstart pour la méthode K-means() permet d'obtenir une taille du groupe stable et garder le meilleur résultat. Nous observons trois groupes qui de grande taille entre 16 et 32 molécules-médicaments par groupe et deux groupes de petit effectif avec 9 et 8 médicaments. Une visualisation des 5 clusters K-means en multidimensional scaling est ensuite réalisée.

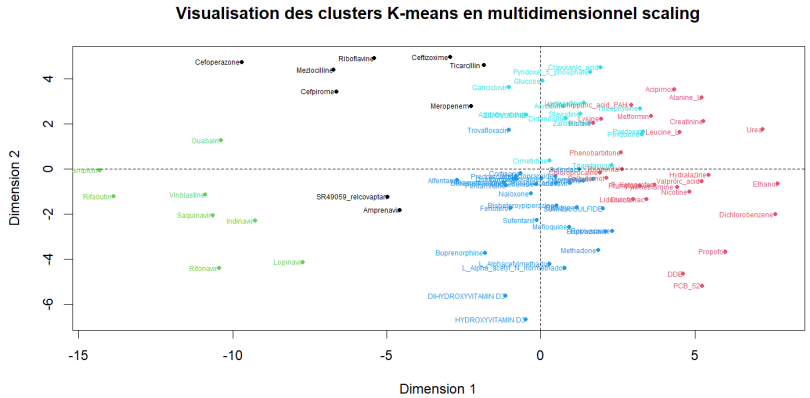


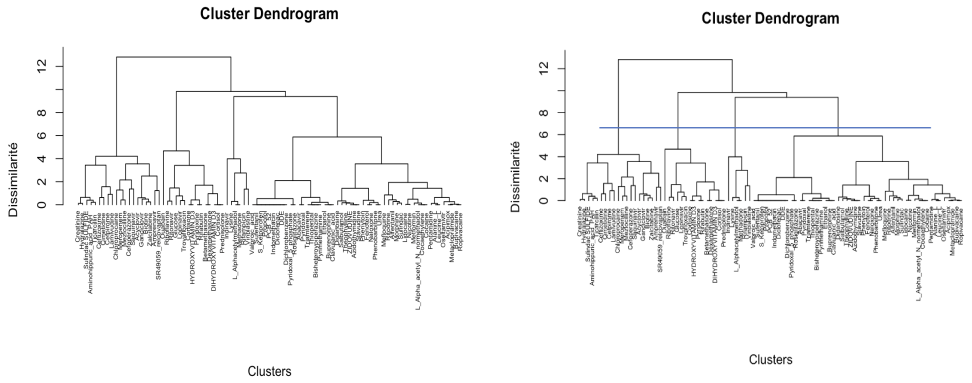
Figure 3: Graphique représentant la visualisation des clusters K-means avec une projection en multidimensional scaling. Cinq clusters avec cluster 1, cluster 2, cluster 3, cluster 4 et cluster 5 respectivement en vert, rouge, bleu, noir et rouge.

Le graphique du multidimensional scaling comporte deux axes avec en abscisse la première dimension et en ordonner la seconde dimension. Chaque point correspond à un médicament et ces médicaments sont divisés en clusters représentés par les différentes couleurs. Les médicaments des clusters 1 et 2 sont correctement séparés selon la première dimension avec un groupe entre -15 et -5 et le second entre 0 et 10. Le cluster 3 est correctement séparé, les

clusters 2, 5 et 6 ne sont pas correctement séparés et se superposent donc les dimensions choisies ne permettent pas d'expliquer la différence entre ces groupes de médicaments 2, 5 et 6.

Classification hiérarchique, hclust() :

Dans la classification hiérarchique, il est essentiel de déterminer d'abord le nombre de clusters qu'on veut obtenir. Pour ce faire, nous devons analyser le dendrogramme qui regroupe des formes dont les similarités sont prononcées. Une fois un dendrogramme produit, nous traçons la ligne de subdivision qui permet de déterminer le nombre de clusters optimal. Cette ligne est au milieu de l'axe des ordonnées, qui est la dissimilitude :



Figures 4 et 5 : Dendrogrammes de classification hiérarchique respectivement avec et sans la ligne de subdivision qui détermine le nombre de clusters

Nous obtenons donc 4 clusters, quoique la ligne de subdivision soit assez proche de 5 clusters, qui serait potentiellement une alternative acceptable. Nous utilisons donc la fonction cutree() pour déterminer le nombre de composants de chaque cluster suivi d'un affichage avec table() : Un nombre relativement faible de composants dans un cluster ne permet pas une analyse pertinente, et un nombre relativement grand peut indiquer un besoin de plus de clusters.

```
## X_cut
##  1  2  3  4
## 50 20 15  6
```

Figure 6: Table de comptage de 4 clusters.

```
## X_cut
##  1  2  3  4  5
## 20 30 20 15  6
```

Figure 7: Table de comptage de 5 clusters.

Nous observons un groupe de très grande taille, à 50 composants, par rapport aux autres groupes dans la figure 6, surtout une disparité notable avec le groupe à 6 composants. Ce cluster de faible taille a beaucoup moins de médicaments, et analyser des clusters avec une trop grande disparité n'est pas souvent recommandé. La figure 7 règle ce souci en séparant le groupe à 50 médicaments en deux groupes à 20 et 30 composants. Le nombre de clusters optimal serait donc bien 5 et non 4, ce qui prouve la pertinence des tables de comptage. Nous pouvons à présent visualiser les clusters hclust() en multi-dimensional scaling:

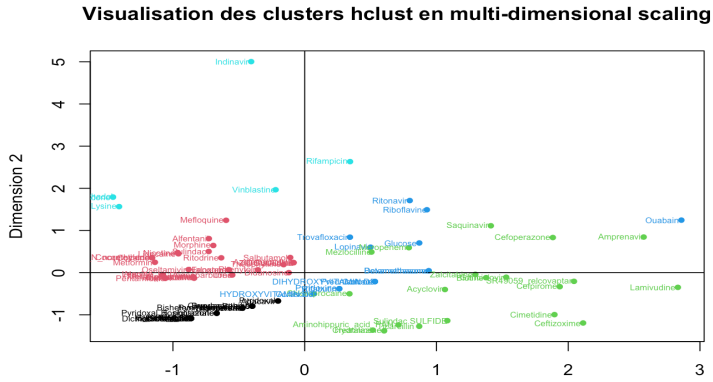


Figure 8: *Graphique représentant la visualisation des clusters hclust avec une projection en multidimensional scaling. Cinq clusters avec cluster 1, cluster 2, cluster 3, cluster 4 et cluster 5 respectivement en noir, rouge, vert, bleu et cyan.*

L'analyse en multi-dimensional scaling indique une séparation assez pertinente entre les clusters, mais nous apercevons néanmoins des outliers tel que le composant Ouabain. Un plus grand nombre de clusters est une solution pour mieux catégoriser ces outliers.

<u>Comparaison des méthodes Hclust et K-means :</u>

Afin de comparer à présent les deux méthodes de classification, nous observons leurs visualisations en multi-dimensional scaling côte à côte, ainsi qu'une table de comptage contrastant les deux.

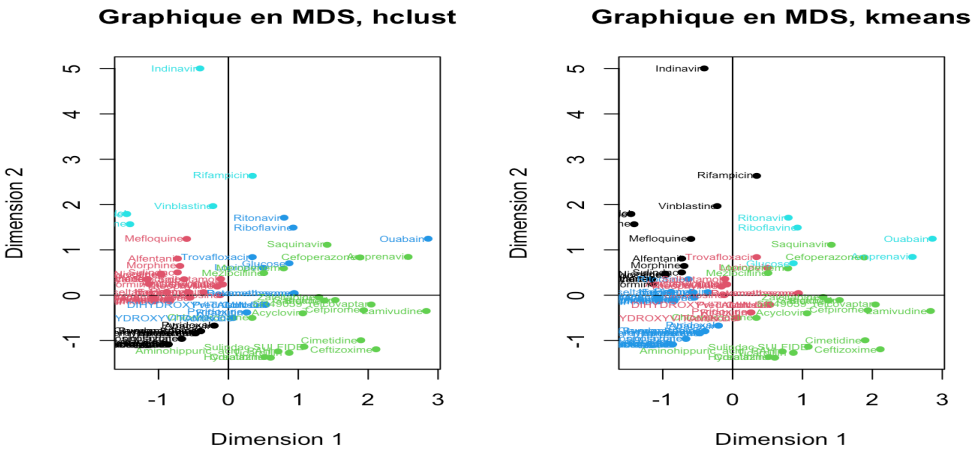


Figure 9: Comparaison des visualisations en Multi-Dimensional Scaling des deux méthodes de classification hclust et K-means.

```
##
## X_cut  1  2  3  4  5
##     1  0  0  0 20  0
##     2 10  6  0 14  0
##     3  0  0 19  0  1
##     4  0 11  0  0  4
##     5  6  0  0  0  0
```

Figure 10: Table de comptage des médicaments par cluster (hclust en lignes, K-means en colonnes)

Les deux méthodes ont le même nombre de clusters optimal, à 5 clusters. Le nombre de médicaments par cluster est également similaire entre les deux méthodes. Nous observons cependant que la répartition des médicaments n'est pas identique : Le cluster 1 K-means contient le cluster 5 hclust et 10 autres composants du cluster 2 hclust. Nous observons donc une différence assez importante dans la méthode et les critères de classification.

Régression linéaire multiple, lm() :

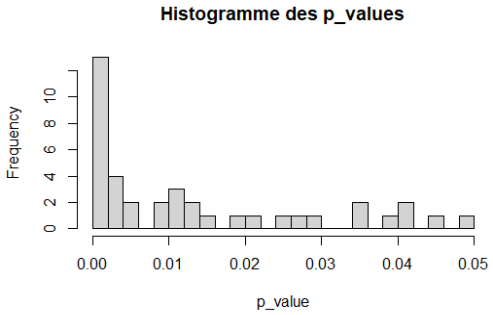


Figure 11: Histogramme des p-values des modèles de régression linéaire construit pour chaque descripteur physico-chimique pour prédire la clairance.

L'analyse de la figure 11 montre que les p-values des modèles construits pour chaque descripteur physico-chimique sont majoritairement inférieurs à 0.05 ce qui permet de choisir un seuil de 0.05. Uniquement les p-values inférieur à ce seuil sont conservés et un modèle de régression linéaire est construit avec les descripteurs conservés. Le résumé (summary) de ce modèle permet de déterminer l'équation du modèle.

```
Call:
lm(formula = Cl ~ SMR_VSA1 + PEOE_VSA.5.1 + PEOE_VSA.3 + Kier2 +
    PEOE_RPC..1 + a_nCl + opr_violation + Kier3 + SlogP_VSA2 +
    SMR_VSA3 + chiral + SlogP_VSA0 + lip_don + ast_violation +
    PEOE_VSA_FHYD + a_don + PEOE_PC. + a_nS + PEOE_RPC. + SMR_VSA0 +
    ast_fraglike + PEOE_VSA.4, data = XY_app)

Residuals:
     Min       1Q   Median       3Q      Max
-0.30799 -0.10207 -0.00194  0.07306  0.38976

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   -0.673034   0.333746  -2.017 0.051038 .
SMR_VSA1      -0.005384   0.002624  -2.052 0.047311 *
PEOE_VSA.5.1  -0.012773   0.005120  -2.495 0.017197 *
PEOE_VSA.3    -0.002340   0.001424  -1.643 0.108840
Kier2          0.046962   0.036667   1.281 0.208240
PEOE_RPC..1    0.874582   0.259053   3.376 0.001740 **
a_nCl          0.098557   0.049890   1.975 0.055708 .
opr_violation -0.133493   0.059268  -2.252 0.030319 *
Kier3         -0.102191   0.048361  -2.113 0.041402 *
SlogP_VSA2     0.007520   0.002773   2.712 0.010088 *
SMR_VSA3      -0.015131   0.005053  -2.995 0.004877 **
chiral        -0.051781   0.024119  -2.147 0.038426 *
SlogP_VSA0     0.006648   0.002965   2.242 0.031017 *
lip_don       -0.226257   0.066141  -3.421 0.001536 **
ast_violation  0.206018   0.076197   2.704 0.010296 *
PEOE_VSA_FHYD  0.900668   0.341708   2.636 0.012193 *
a_don          0.178063   0.050155   3.550 0.001069 **
PEOE_PC.       0.456185   0.141922   3.214 0.002711 **
a_nS          -0.360443   0.082923  -4.347 0.000104 ***
PEOE_RPC.     -0.568231   0.448564  -1.267 0.213151
SMR_VSA0      -0.003299   0.002353  -1.402 0.169262
ast_fraglike   0.428284   0.121886   3.514 0.001184 **
PEOE_VSA.4     0.004540   0.002616   1.735 0.091003 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1854 on 37 degrees of freedom
Multiple R-squared:  0.7524,    Adjusted R-squared:  0.6052
F-statistic: 5.111 on 22 and 37 DF,  p-value: 6.874e-06
```

Figure 13: Résumé (summary) du modèle de régression linéaire pour prédire la clairance avec les descripteurs les plus appropriés. Avec les p-value significatives au risque 5% encadré en rouge. Le R^2 en bleu et R^2_{adj} en vert.

Nous observons que ce modèle conserve les descripteurs physico-chimiques les plus informatifs. Les descripteurs ont des coefficients significativement différents de 0. Ce modèle

possède un R^2 égale à 0.7524 qui correspond au coefficient de détermination, étant proche de 1 il s'agit d'un bon modèle car la régression permet de déterminer un pourcentage important de la distribution des points. Le R^2adj est égal à 0.6052 soit supérieur à 0.5, ce qui indique que plus de la moitié de la variance dépendante peut être expliquée par la variable indépendante dans le modèle, mais d'autres facteurs doivent être pris en compte pour évaluer la qualité du modèle. L'équation de ce modèle est *Cl= -0.673034 + -0.005384*SMR_VSA1 - 0.012773*PEOE_VSA.5.1 + 0.874582*PEOE_RPC..1 - 0.133493*opr_violation - 0.102191*Kier3 + 0.007520*SlogP_VSA2 - 0.015131*SMR_VSA3 - 0.051781*chiral + 0.006648*SlogP_VSA0 - 0.226257*lip_don + 0.206018*ast_violation + 0.900668*PEOE_VSA_FHYD + 0.178063*a_don + 0.456185*PEOE_PC. - 0.360443*a_nS + 0.428284*ast_fraglike.* Les descripteurs utilisés dans l'équation de ce modèle permettent de prédire la clairance d'un médicament.

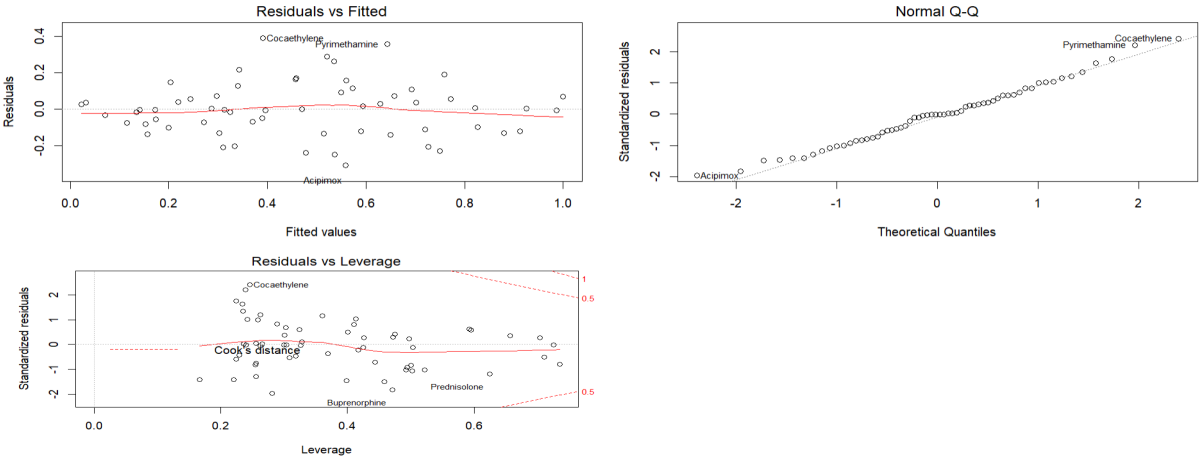


Figure 14: Analyse graphique de la qualité du modèle de régression linéaire, avec Residual vs Fitted, Normal Q-Q et Residuals vs Leverage.

La qualité du modèle est ensuite vérifiée en analysant ces graphiques. Le graphique residuals vs Fitted montre la relation entre les valeurs résiduelles et les valeurs ajustées du modèle. Les résidus sont aléatoires, ce qui indique une homoscédasticité. Les points suivent une ligne droite sur le graphique normal Q-Q, ce qui indique que les résidus suivent une distribution normale. Le graphique leverage évalue l'influence de chaque observation sur le modèle de régression linéaire. Les points éloignés de la ligne rouge (Cook) peuvent avoir une influence.

```
      RMSE   Rsquared        MAE
0.41484567 0.06744903 0.27052008
```

Figure 15 : Mesures de précision pour évaluer la performance du modèle de régression.

Le RMSE correspondant à l'écart-type des résidus autrement dit erreur de prévision qui est de 0.41484567 donc éloigné de 0. Cela suggère que le modèle a une bonne précision. Le R^2=0.06744903 seulement 6,74% de la variance totale de la variable dépendante peut être expliquée par le modèle. Le modèle a une faible qualité d'ajustement. La MAE correspond à l'erreur absolue moyenne est faible avec une valeur de 0.27052008. Le modèle a une bonne prédiction. Ces résultats indiquent que le modèle a une bonne précision mais une qualité d'ajustement relativement faible.

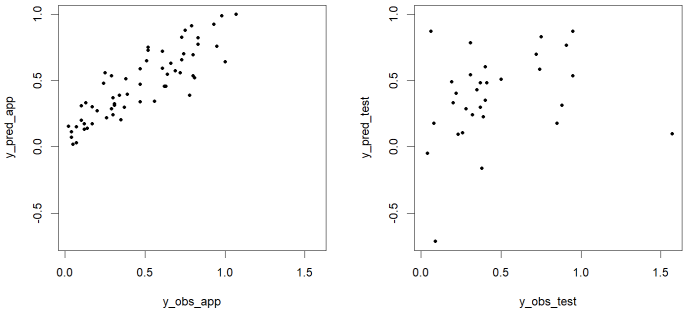


Figure 16 : Comparaison des prédictions entre l'échantillon d'apprentissage (à gauche) et de validation (à droite) dans le modèle de régression.

Nous observons l'existence d'une relation linéaire entre la clairance prédite par le modèle et celle observée pour l'échantillon d'apprentissage mais pas pour l'échantillon de validation. Il s'agit d'un bon modèle d'apprentissage et mauvais modèle de validation.

Arbres de classification rpart():

Nous utilisons la fonction R rpart() en régression pour obtenir un arbre de partitionnement de notre jeu de données (séparé en ⅔ apprentissage et ⅓ test), avec la méthode anova.

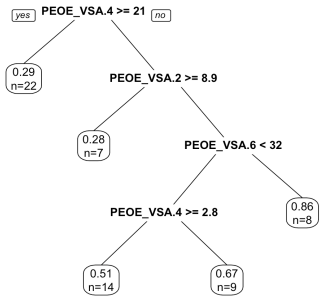


Figure 17: Arbre de partitionnement

Nous obtenons un arbre de partitionnement très lisible grâce à la librairie rpart.plot. Chaque nœud représente une question, et l'arbre donne les pourcentages et nombre de médicaments qui répondent ou non(respectivement yes et no) à la question.

```
##
## Regression tree:
## rpart(formula = Cl ~ ., data = XY_app, method = "anova")
##
## Variables actually used in tree construction:
## [1] PEOE_VSA.2.1 PEOE_VSA.4.1 PEOE_VSA.6
##
## Root node error: 6.0483/60 = 0.1008
##
## n= 60
##
##         CP nsplit rel error xerror    xstd
## 1 0.188817      0   1.00000 1.0275 0.21617
## 2 0.129674      1   0.81118 1.0850 0.26989
## 3 0.082222      2   0.68151 1.0444 0.27649
## 4 0.024588      3   0.59929 1.0366 0.27867
## 5 0.010000      4   0.57470 1.0267 0.26286
```

Figure 18: Validation croisée et estimation des erreurs de classification de l'arbre

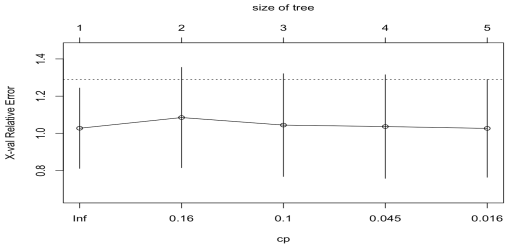


Figure 19: Erreur relative en fonction de la taille de l'arbre en nombre de feuilles

Nous choisissons la taille de l'arbre qui minimise l'erreur donc ici 5 feuilles. Pour ne pas surentrainer l'arbre aux données d'apprentissage, nous pouvons choisir de le tailler ou "prune", donc diminuer sa taille et remplacer les feuilles par la moyenne de leurs valeurs. Nous utilisons donc la fonction prune() avec un cp seuil qu'on choisit (0.025) et nous comparons les deux arbres:

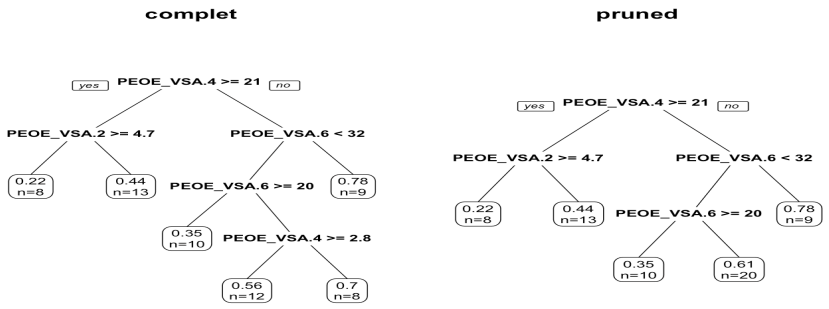


Figure 20: L'arbre "taillé" a bien combiné deux feuilles et éliminé un nœud.

```
     RMSE  Rsquared       MAE
0.2505499 0.3421271 0.1886917
```

Figure 21: Qualité de l'arbre en apprentissage

```
     RMSE  Rsquared       MAE
0.2941511 0.1860696 0.2458306
```

Figure 22: Qualité de l'arbre en test

L'arbre a une RMSE éloignée de 0, donc une bonne précision. R^2 est bien en apprentissage mais moins bon en test, ce qui signifie une bonne qualité d'ajustement mais améliorable, et une MAE faible donc une prédiction acceptable.

Conclusion:

Pour conclure, les modèles utilisés étaient plus ou moins performants selon les paramètres utilisés, et ont donné des résultats intéressants. Les méthodes non supervisées K-means et Hclust ont produit des résultats similaires et ont permis d'identifier des groupes de composés ayant des propriétés similaires, ce qui est attendu. Les méthodes supervisées ont montré des indices de performance encourageants mais potentiellement améliorables. Le jeu de données, surtout une fois nettoyé, était de relativement faible taille, et un jeu de données plus grand aurait potentiellement de meilleurs résultats. Enfin, nous avons réussi à répondre à la question biologique : les descripteurs physico-chimiques ont bien un effet sur la clairance des médicaments, comme vu dans les résultats d'analyse. SMR_VSA1, PEOE_VSA.5.1, PEOE_RPC..1, opr_violation, Kier3, SMR_VSA3, SlogP_VSA2, SlogP_VSA0, ast_violation, PEOE_VSA_FHYD, PEOE_PC, a_nS et ast_fraglike sont tous descripteurs de propriétés physico-chimiques de la molécule tels que la taille, la polarité et la solubilité. Chiral caractérise la présence de centres stéréogéniques. Lip_don et a_don caractérisent les propriétés d'accepteur et donneur de liaisons hydrogène de la molécule. En combinant ces descripteurs, il est possible de prédire la clairance. Ces méthodes ont permis de mieux comprendre le phénomène du passage de la barrière placentaire et d'identifier des composés avec des propriétés nécessaires pour traverser la barrière placentaire.