Etude des facteurs de risque de mortalité chez les chiots

Centre NeoCare, Ecole Nationale Vétérinaire de Toulouse,
23 Chemin des Capelles, 31300 Toulouse

Rapport de projet tuteuré de 4ème année

Julia VILAS, Alex GODFRIN, Léandre GARRIGA, Imen MEGDICHE
Promotion 2023 Amélie MUGNIER

FICHE SIGNALETIQUE

Centre NeoCare, Ecole Nationale Vétérinaire de Toulouse

23 Chemin des Capelles, 31300 Toulouse

Etablissement public national à caractère administratif

Directeur : Pr Pierre Sans

05 61 19 38 00

https://envt.fr

Contacts :

Maîtrise d'ouvrage :

♦ Dr Amélie MUGNIER, maîtresse d'ouvrage, doctorante travaillant sur le petit poids de naissance chez les chiots et chatons

♦ Pr Sylvie CHASTANT-MAILLARD, responsable du centre NeoCare, Professeur en Reproduction

♦ Dr Hanna MILA, Maître de conférence en élevage des carnivores

♦ Dr Patricia RONSIN, Patricien hospitalier au Centre Hospitalier Universitaire Vétérinaire des Animaux de Compagnie

♦ Dr Pétra ROUCH-BUCK, ingénieur de recherche

♦ Quentin GARRIGUES, doctorant

♦ Betty DUC, assistante

Sommaire

1. Glossaire **3**

2. Introduction **3**

3. Parties prenantes **4**

4. Présentation du sujet **5**

5. Analyse des données complètes **5**
5.1 Feature Selection 6
5.2 ACP 7
5.3 Conclusion 11

6. Résolution des données incomplètes **12**
6.1 Clustering individuel des données complètes 12
6.2 Distances de Hamming et ET logique 12
6.3 Imputation des données 13

7. Analyse des données du dataset complet **15**
7.1 Feature Selection 15
7.2 ACP 17

8. Gestion de projet **18**

9. Conclusion **19**

1. GLOSSAIRE

- ✔ ACP (Analyse en Composantes Principales) : méthode d'analyse de données permettant de réduire la dimension des caractéristiques
- ✔ Feature : caractéristique du jeu de données utilisée comme critère servant à la prédiction dans un modèle prédictif.
- ✔ Ig G : immunoglobulines G, principaux anticorps.
- ✔ KNN (K Nearest Neighbors) : méthode d'apprentissage supervisée des k plus proches voisins
- ✔ MOA (Maîtrise d'ouvrage) : client d'un projet.
- ✔ MOE (Maîtrise d'œuvre) : groupe qui apporte les solutions technologiques demandées par la maîtrise d'ouvrage.
- ✔ Parvovirus : maladie infectieuse du chien.
- ✔ PCR (*Polymerase Chain Reaction*) : méthode de prélèvement.

2. Introduction

Nous avons choisi le sujet “Etude des facteurs de risque de mortalité chez le chiot” car la thématique d’analyse des données nous intéressait particulièrement et s’inscrivait dans la spécialité Big Data que nous suivons tous les trois. L’objectif de ce projet tuteuré, en tant que maîtrise d'œuvre, est d’apporter des solutions technologiques demandées par la maîtrise d’ouvrage et, notamment, des graphiques et interprétations de ces derniers pour apporter une réponse à la problématique posée.

Pour ce deuxième et dernier semestre, nous avons pour objectifs de réaliser une analyse des données sans valeurs manquantes puis de compléter les données manquantes afin de pouvoir, au final, analyser nos données dans leur globalité.

3. Parties prenantes

La maîtrise d’ouvrage est l’institut de recherche NeoCare - Néonatalogie des Carnivores, Reproduction et Élevages de l’Ecole Nationale Vétérinaire de Toulouse (ENVT). Ce centre vise à améliorer la santé des chiots et des chatons, avec une expertise sur leurs deux premiers mois de vie. Amélie Mugnier, doctorante travaillant sur le petit poids de naissance chez les chiots et chatons à NeoCare, est la maîtresse d’ouvrage de ce projet.

La maîtrise d'œuvre est composée de Julia Vilas, cheffe de projet, Alex Godfrin et Léandre Garriga. Les tuteurs de notre école pour nous aider sur le projet sont Imen Megdiche, Rémi Bastide et Lotfi Chaari.

Nous avons, pour ce premier semestre, déjà entrepris la compréhension des données à notre disposition ainsi que de la problématique et nous nous lançons sur une analyse diagnostique pour comprendre pourquoi les chiots de l’élevage sont morts. La préparation des données a rapidement été abordée, seulement l’étape d’inspection pour le moment, mais nous montre déjà plusieurs sous-problématiques à gérer comme les données manquantes ou mal formatées. Notre travail consiste alors à poursuivre l’inspection et le nettoyage des données. Cela nous permettra de commencer l’analyse et d’en tirer les premiers résultats que nous pourrons présenter à notre MOA. Si le temps nous le permet, nous pourrons également nous concentrer sur la création d’un modèle prédictif du problème, incluant son évaluation et son déploiement.

4. Présentation du sujet

La mortalité post-natale est un problème pour la filière canine avec près d'un chiot né-vivant sur 10 qui n'atteindra pas l'âge de l'adoption. Pour réduire le taux de mortalité chez les chiots, il faut trouver quels sont les facteurs qui augmentent ce phénomène. Cela permettra de mieux cibler ces facteurs et ainsi augmenter l'espérance de vie des chiots. L'étude concerne un élevage de 169 chiots observés dont les données, qui semblaient pertinentes pour notre MOA, ont déjà été extraites.

Nous avons reçu un jeu de données sous la forme d'un fichier Excel. Les lignes correspondent aux 169 chiots de l'élevage et les 80 colonnes aux différentes mesures et observations effectuées sur les chiots ou leur mère durant les 56 premiers jours de leur vie. Les principales difficultés de l'analyse de ces données résident dans le fait que les données biologiques sont complexes et que le phénomène à étudier est rare.

Il nous a d'abord fallu comprendre la totalité des données avant d'aborder la mise en place d'un plan adéquat. On retrouve dans le tableau des données "classiques" comme le sexe, la portée à laquelle le chiot appartient, la race et un identifiant pour le chiot et la portée en question. La colonne qui nous intéresse le plus, indiquant si les chiots sont morts ou non, est représentée par une variable binaire : 1 pour mort et 0 pour vivant.

Le reste des données sont des mesures scientifiques vétérinaires complexes. Bien que les noms des colonnes soient expliqués dans le mémo donné par la MOA, nous avons dû rechercher une explication des termes biologiques utilisés pour mieux assimiler le sujet, et plus précisément, les facteurs à considérer.

Le parvovirus est un virus mortel qui se transmet par contact direct avec les selles d'un chien malade. On peut déterminer sa charge virale (le nombre de copies du virus issus de réplications) en analysant les selles des chiots. Ce virus est neutralisé par les immunoglobulines G dont le taux a été mesuré. Les vétérinaires ont aussi étudié plus globalement les selles par le score fécal, de 1 à 13 chez les chiots, qui est une mesure de la qualité des selles produites. Un bon score se situe aux alentours de 7, tandis qu'un score proche de 1 ou 13 est mauvais. L'étude des selles permet aussi de détecter la présence d'œufs de parasites comme les giardias, les coccidies et les toxocaras. Ces trois parasites se transmettent tous par les fèces canin d'où l'importance considérable portée à ce facteur. En outre, la toxocarose peut être contractée initialement par l'ingestion de larves enkystées provenant d'une alimentation carnée mal cuite, par le lait maternel ou encore par voie intra-utérine chez le chiot.

5. Analyse des données complètes

À la réception de notre jeu de données, nous avons tout d'abord évalué le taux de données manquant par facteurs. Ce taux varie de 10 à 90% de données manquantes. Cela nous a permis de mettre en évidence 10 facteurs sans aucune donnée manquante et sur lesquels nous avons fait une première analyse.

5.1 Feature Selection

La Feature Selection est un procédé qui consiste à trouver des facteurs liés à une variable précise dans notre cas : la mort des chiots. Comme nous l'avons vu au premier semestre, il existe trois méthodes : Filter, Wrapper et Embedded.

Au premier semestre, nous n'avions implémenté que la méthode Filter et seulement sur le dataset complet. Or, nous avons appris entre temps que certaines features dépendent de la taille de la race, ce qui changent les corrélations que l'on peut obtenir avec la mort.

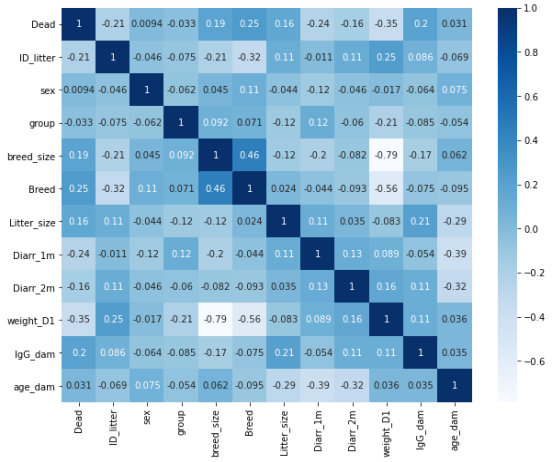


Figure 1 : Résultats de la méthode Filter sur le dataset complet

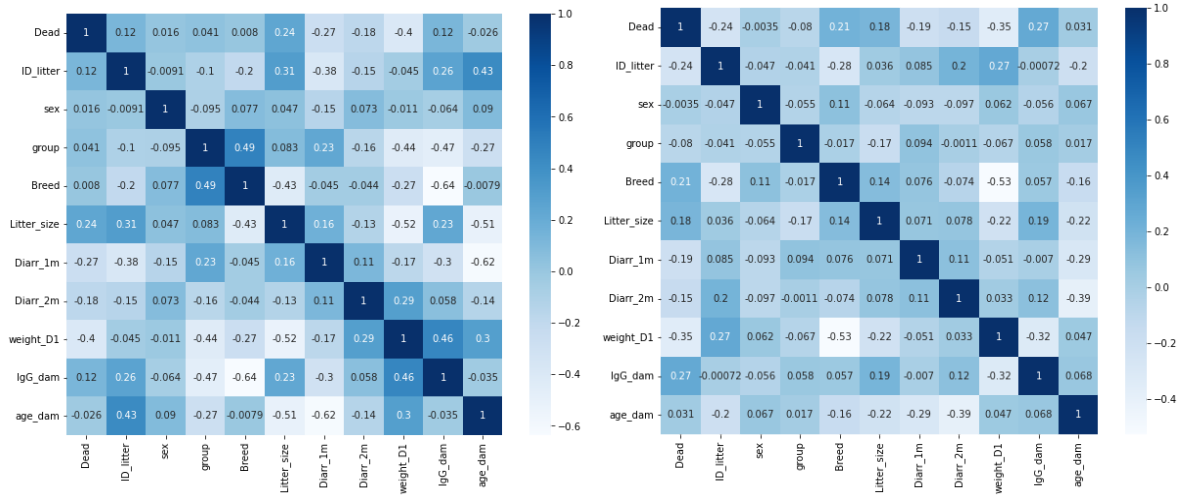


Figure 2 : Résultats de la méthode Filter sur les races de taille L et de taille S

Pour la méthode Wrapper, nous n'avons pas pu tracer de graphique.

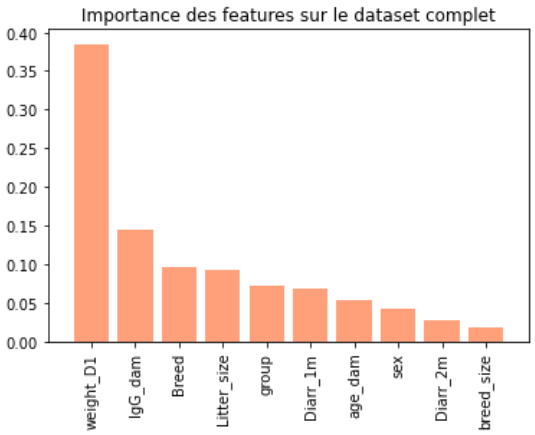


Figure 3 : Résultats de la méthode Embedded sur le dataset complet

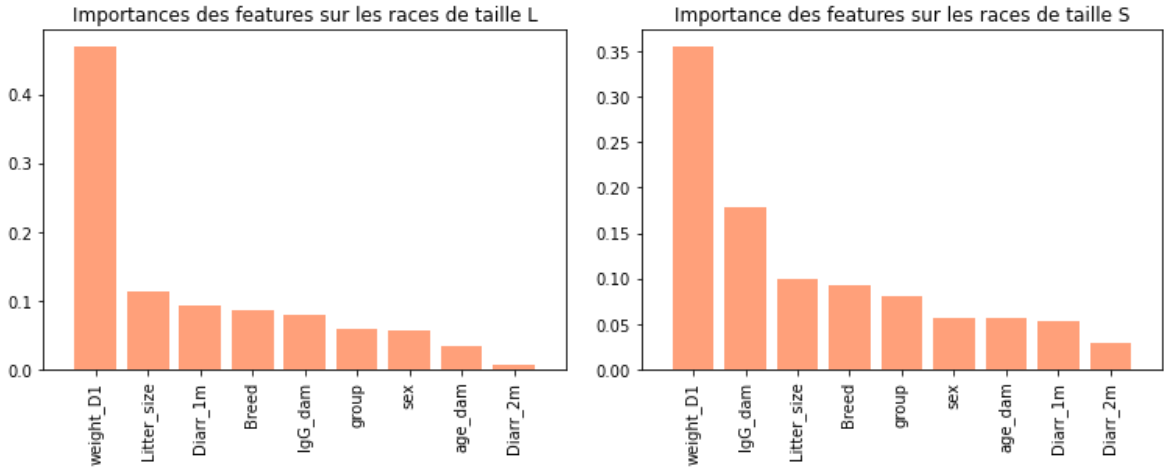


Figure 4 : Résultats de la méthode Embedded sur les races de taille L et S

Nous avons présenté ces résultats à notre cliente lors d'une réunion début février. Pour chaque méthode, nous avons obtenu des résultats avec des ressemblances mais globalement différents. Le poids des chiots au jour 1 revient dans chaque méthode, mais le reste des résultats est différent. On a pu obtenir par exemple, le taux d'immunoglobuline G de la mère, la présence de signe de diarrhée au 1er mois ou au 2ème mois, ou encore l'âge de la mère.

5.2 ACP

Comme la Feature Selection n'a pas suffi à extraire des facteurs, nous avons utilisé une autre méthode appelée ACP (Analyse en Composantes Principales) pour affiner la recherche sur ces 10 features. L'ACP est un outil très utile lorsque le jeu de données contient un nombre important de données quantitatives à traiter et interpréter. Nous avons ici 10 variables ce qui est relativement important. L'ACP utilise une matrice indiquant la corrélation entre les variables pour réduire la dimension des variables et les projeter dans un nouvel espace plus petit. Ces

nouvelles variables sont appelées composantes principales. L'ACP permet aussi de visualiser graphiquement les distances entre les individus.

Les 169 individus décrits par les p composantes principales peuvent être représentés comme un nuage de points dans un espace à p dimensions appelé espace des individus. Des groupes d'individus peuvent être identifiés au sein de ce nuage.

Il faut trouver le bon nombre p de composantes principales à utiliser. Il existe des méthodes calculatoires et des méthodes graphiques. L'analyse des valeurs propres ou eigenvalues est la base de différentes techniques calculatoires pour déterminer le nombre optimal de composantes principales.

	Val.Propre	Seuils
0	2.688317	2.828968
1	2.099705	1.828968
2	1.383937	1.328968
3	1.072756	0.995635
4	0.642925	0.745635
5	0.557957	0.545635
6	0.372018	0.378968
7	0.133610	0.236111
8	0.048774	0.111111

Figure 5 : Valeurs propres issues de l'ACP des 10 features

La règle de Kaiser-Guttman repose sur le fait que lorsque l'ACP est normalisée, la somme des valeurs propres étant égale au nombre de variables, leur moyenne vaut 1. Par conséquent, cette règle stipule qu'un axe est intéressant si sa valeur propre est supérieure à 1. Selon cette méthode, on garderait quatre composantes.
Une deuxième approche mathématique avec les valeurs propres est le test des bâtons brisés. Il consiste à comparer la valeur propre à son seuil. Un axe est prometteur si la valeur propre est supérieure à son seuil. Pour nos données, on garderait les axes 1, 2, 3 et 5 d'après cette technique. Ce résultat est étrange car ce test permet généralement de garder les k premières composantes.

Il existe aussi des méthodes graphiques pour déterminer le nombre de composantes principales. La première consiste à tracer un scree plot. C'est un graphique qui affiche la décroissance des valeurs propres en fonction du nombre de features. On utilise ensuite la méthode du coude : lorsque la courbe se casse, formant ainsi un coude, la valeur en abscisse correspond au nombre de composantes principales à conserver. En règle générale, le coude est très marqué lorsque les variables sont fortement corrélées.

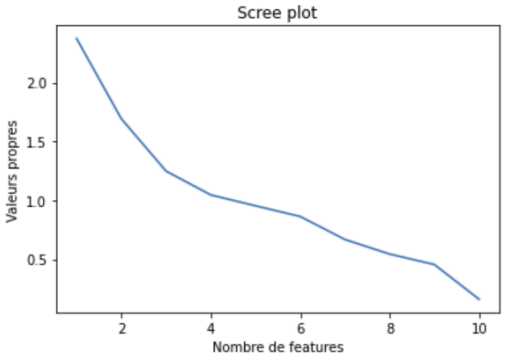
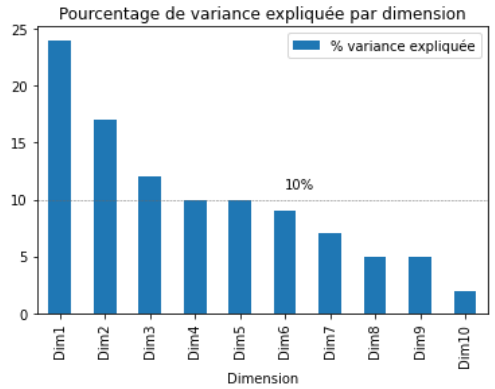


Figure 6 : Scree plot de l'ACP sur les 10 features

Dans notre cas, le coude n'est pas bien défini mais il semble se situer vers 3-4 features.

Les méthodes calculatoires et graphiques ne donnent pas de résultats clairs mais semblent indiquer quatre composantes principales.

On peut aussi tracer le pourcentage de variance qu'explique chaque dimension pour en être sûr.



	Dimension	Variance expliquée	% variance expliquée	% cum. var. expliquée
0	Dim1	2.387349	24.0	24.0
1	Dim2	1.702286	17.0	41.0
2	Dim3	1.255418	12.0	53.0
3	Dim4	1.051615	10.0	64.0
4	Dim5	0.958899	10.0	73.0
5	Dim6	0.867579	9.0	82.0
6	Dim7	0.671149	7.0	88.0
7	Dim8	0.546747	5.0	94.0
8	Dim9	0.457302	5.0	98.0
9	Dim10	0.161177	2.0	100.0

Figure 7 : Variance expliquée par les dimensions de l'ACP sur les 10 features

Les quatre premières dimensions expliquent 64% de la variance ce qui est relativement peu. On essaie généralement de choisir le nombre de composantes principales tel qu'au moins 80% de la variance soit expliquée. Si on choisissait d'avoir au moins 80% de variance expliquée, on devrait choisir 6 dimensions. Or, passer de 10 dimensions à 6 n'est pas très optimal.

L'ACP est aussi une méthode qui permet de projeter les observations dans un espace réduit. Par souci d'affichage, on peut seulement représenter jusqu'à la dimension 2 tel qu'un maximum d'information soit conservé. Représenter les enregistrements permet d'identifier des groupes homogènes ou au contraire des observations atypiques. Dans notre cas, on aimerait obtenir deux clusters.

On représente les chiots selon leurs coordonnées factorielles des deux dimensions expliquant le plus de variance. Un axe correspond à la première composante principale qui explique 24% de la variance et l'autre à la deuxième composante principale qui explique 17% de la variance. Le plan formé par ces deux axes représente donc 41% de la variance expliquée, ce qui est très faible.

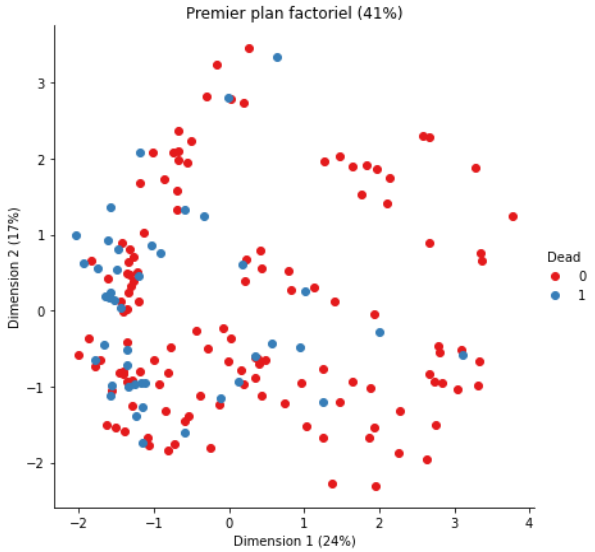


Figure 8 : Représentation des chiots avec l'ACP des 10 features suivant la mort

Le premier graphe affiche les chiots selon s'ils sont morts ou vivants afin d'observer d'éventuels clusters. Dans notre cas, aucun cluster n'est clairement défini car des points rouges et des points bleus se superposent. On remarque tout de même que la plupart des chiots morts se situent là où la première composante est la plus faible, tout à gauche. Cependant, cela ne suffit pas pour dire que c'est un cluster. On définit la validité d'un cluster par le rapport de la variance inter cluster par la variance intra cluster. En effet, les clusters doivent être séparés les uns des autres : la variance inter cluster doit être élevée. A l'inverse, la variance intra cluster doit être faible car les points du cluster doivent être proches pour former un ensemble. Finalement, ce graphique nous confirme que l'ACP n'est pas performante.
Le deuxième affiche l'indice du chiot afin de repérer des valeurs aberrantes. Il n'y en a pas dans nos données.

Nous avons calculé deux indicateurs relatifs aux variables : le COS^2 et le CTR.
Le cosinus carré des variables détermine la qualité de représentation des individus sur les axes donc par rapport aux composantes.
Le CTR est la contribution des variables dans la définition des composantes principales. Les contributions permettent d'identifier les individus qui représentent le mieux les composantes principales et donc par opposition les valeurs aberrantes.

Nous avons obtenu des résultats non significatifs.

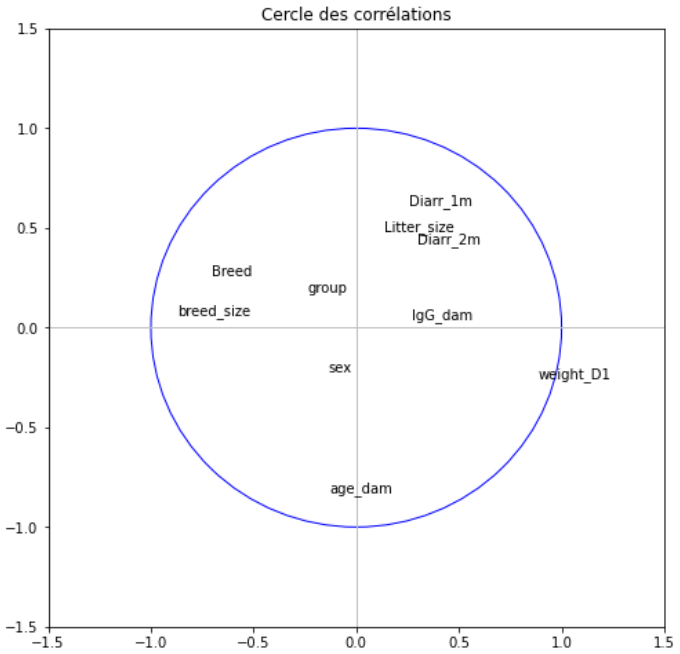


Figure 9 : Cercle des corrélations de l'ACP des 10 features

Les features qui sont du même côté sur le cercle sont corrélées. Au contraire, celles qui forment un angle de 90° ne sont pas corrélées. On voit par exemple que Diarr_1m et Diarr_2m sont du même côté du cercle donc que ces deux variables sont corrélées ce qui est logique puisqu'elles représentent la même chose à un temps différent.

5.3 Conclusion

Suite aux différentes analyses présentées précédemment, nous avons pu faire diverses constatations, certaines affirmant ce que nous savions déjà (l'importance du poids des chiots à la naissance par exemple), d'autres apportant de nouveaux éléments sur les possibles causes de décès chez les chiots. Parmi les observations que nous avons pu faire trois caractéristiques se détacher du lot de neuf, "IgGDam" correspondant au taux d'immunoglobuline G de la mère. "LitterSize" représentant le nombre de chiot de la portée. Et finalement, "Diarr_1m" représentant à la présence de signe de diarrhée avant un mois.

Malgré ces découvertes/révélations, il n'est malheureusement pas possible de faire de réelle conclusion sur les causes de décès c'est pourquoi nous devons réaliser une imputation de données sur la totalité de notre jeu de données. Nous effectuons ensuite les mêmes étapes que pour le jeu de données incomplet, feature selection et ACP.

6. Résolution des données incomplètes

Le travail réalisé jusque là nous a apporté quelques premiers éléments de réponse concernant les facteurs à risque pour les chiots. Cependant, l'utilisation d'une dizaine de caractéristiques seulement ne nous permet pas d'affirmer avec certitude que ces facteurs soient les seuls importants. Il nous faut donc procéder de manière à ce que les autres facteurs, ne comportant pas assez de données, puissent néanmoins être utilisés. L'utilisation de toutes les données nous permettra de confirmer (ou non) nos résultats précédents et de tirer de nouvelles conclusions. Dans cette partie nous nous intéresserons donc à la résolution des données, qui nous permettra ensuite de les analyser dans leur globalité (ou presque).

6.1 Clustering individuel des données complètes

Le clustering est un algorithme permettant de diviser une série de données en 2 ou plusieurs classes. Le seuil de division est définie selon la méthode de clustering utilisée, comme par exemple la maximisation de l'espérance, la partition de graphe ou, comme nous l'avons utilisé, KMeans. Cette dernière permet de calculer les centroïdes (points centraux) de chaque classe. Avec les coordonnées de ces centroïdes, la méthode calcule ensuite la distance entre chaque point de notre série et chaque centroïde calculé pour définir la sous-classe à laquelle appartient chaque point.
Dans notre cas, nous avons divisé chaque feature parmis les 10 features complètes en 2 classes.

6.2 Distances de Hamming et ET logique

Nous avons calculé les distances de Hamming et effectué un ET logique entre chaque feature deux à deux afin de déterminer quelles features correspondaient le mieux pour faire l'imputation des données manquantes.

Ensuite, nous avons pu calculer la variance intra-cluster (variance de chaque sous-classe) de chaque caractéristique étudiée pour évaluer l'écart des données à la moyenne intra-cluster.

```
Variance intra cluster :

{'sex': [0.0, 0.0],
 'group': [0.2525536261491318, 0.253006329113924],
 'breed_size': [0.0, 0.0],
 'Breed': [3.033894343151005, 2.2531578947368422],
 'Litter_size': [1.120979020979021, 1.5454026270702457],
 'Diarr_1m': [0.0, 0.0],
 'Diarr_2m': [0.0, 0.0],
 'weight_D1': [2302.76914503016, 6904.8],
 'IgG_dam': [2.042612438026801, 0.6886470221368511],
 'age_dam': [0.39254658385093166, 0.5419501133786848]}
```

Figure 10 : Variance intra cluster des 10 features

De cette variance, nous avons pu définir 4 différents groupes parmi nos 10 features qui nous permettront ensuite de réaliser l'imputation de données.

On a un premier groupe qui comporte sex, breed_size, diarr_1m et diarr_2m dont la variance est nulle. C'est le cas car les variables sont binaires : 0 ou 1. On a un deuxième groupe où la variance est comprise entre 0,25 et 2 comportant group, IgG_dam et age_dam. Un troisième groupe contient breed et litter_size dont la variance intra cluster se situe entre 1,12 et 3. Enfin, le dernier groupe a une variance extrêmement élevée puisque la variance est de 2302.

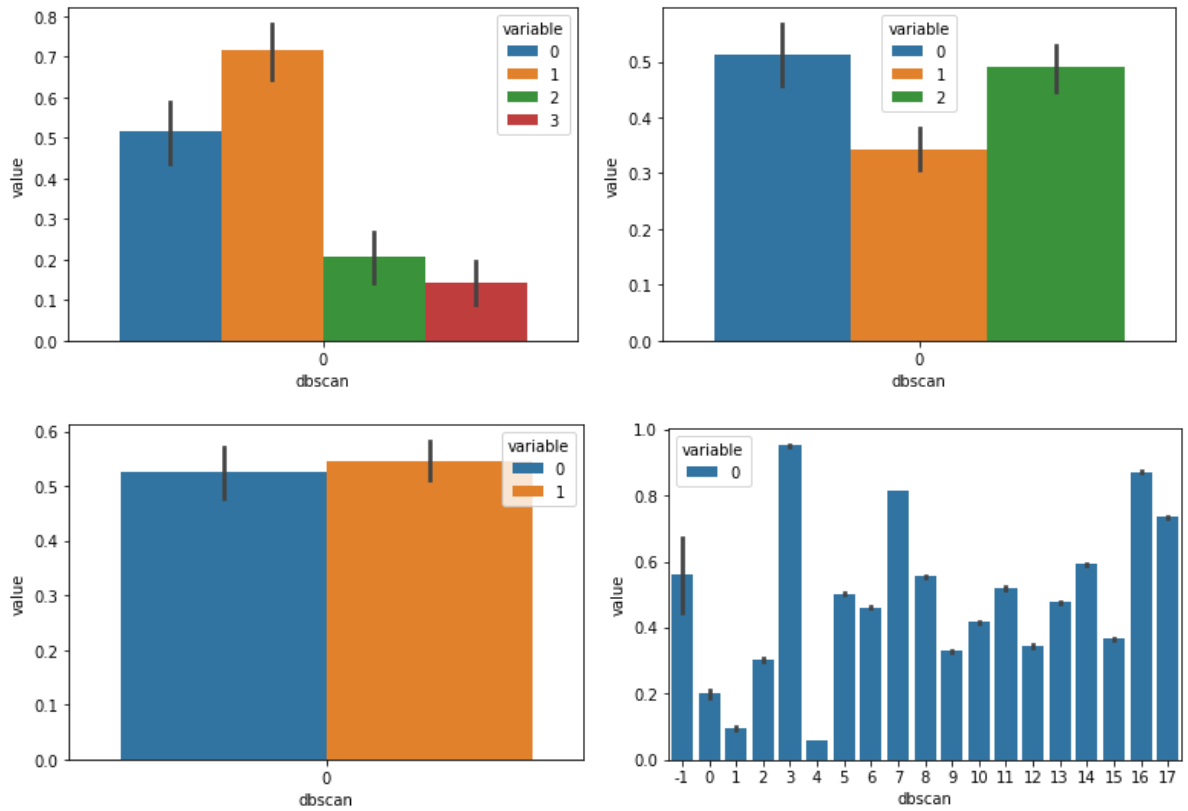


Figure 11 : DbScan des quatre groupes de features

Nous n'avons pas pu continuer cette méthode d'imputation par manque de temps et nous n'avons pas compris comment procéder ensuite. L'idée était de créer des clusters individuels avec nos 10 features par rapport à la variable cible Dead.

6.3 Imputation des données

Nous avons premièrement attaqué le problème de résolution des données par des imputations simples, comme par exemple KMeans, KNN ou RandomForest. Les imputations simples sont des algorithmes d'imputations qui permettent de compléter les valeurs manquantes d'une feature sans utiliser d'autres features du dataset.

Nous n'étions pas sur de la démarche à suivre pour réaliser une imputation de données à partir de certaines features de notre dataset, d'où cette première démarche d'imputation simple.
L'algorithme KMeans permet de réaliser une imputation d'une feature par la moyenne des données déjà présentes. Elle représente un algorithme assez simple, mais qui ne correspond pas vraiment à notre dataset.
Celui du KNN (K-Nearest Neighbors) permet également de réaliser une imputation par la moyenne, mais basé sur un nombre définis (K) de voisins les plus proches. Si on sélectionne par exemple un K égal à 5, pour une donnée manquante, la valeur imputée sera la moyenne des valeurs des 5 voisins les plus proches du point manquant. Cet algorithme propose de meilleurs résultats en général que celui de KMeans.
Enfin, l'algorithme RandomForest réalise une imputation des données par la moyenne, avant d'entraîner un RandomForest, d'utiliser la matrice de proximité du dataset qui sera utilisée pour définir le poids de chaque donnée. Les valeurs imputées sont alors remplacées par la moyenne des poids des données non-manquantes.

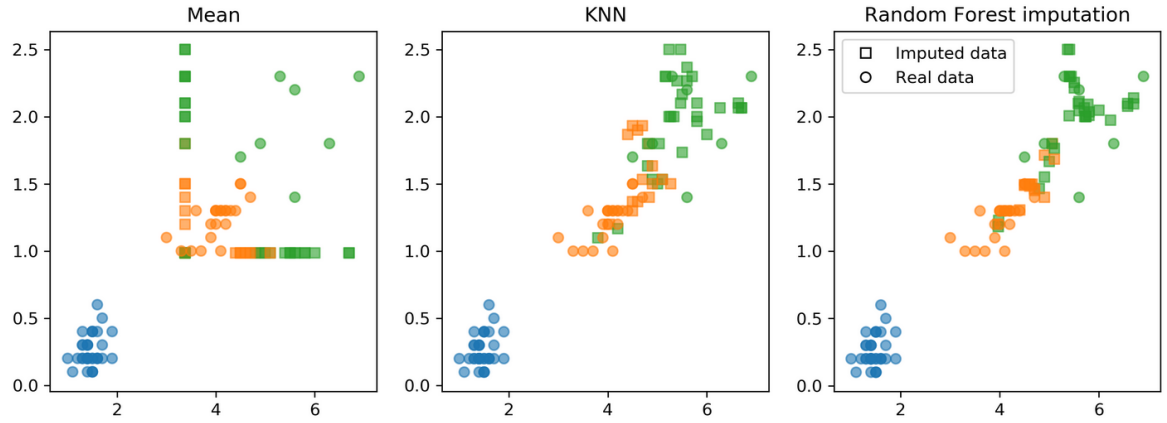


Figure 12 : Comparaison théorique des résultats de méthodes d'imputation

Ces algorithmes d'imputations simples fournissent des résultats exploitables mais pas aussi convenables qu'une imputation utilisant les features complètes de notre dataset.
Il nous a donc fallu réaliser un ET logique entre les vecteurs de chaque features pour déterminer quelles features seraient les plus appropriées pour imputer celles contenant des données manquantes. Ce travail est pour l'instant en suspens, mais réussit, il pourra nous permettre d'avoir un jeu de données beaucoup plus clair avec des données plus précises qu'avec une des imputations simples citées plus haut. Avec ces nouvelles données, nous pourrons refaire une feature selection et une ACP avec un dataset beaucoup plus complet pour obtenir des résultats plus convenables.

7. Analyse des données du dataset complet

Pour cette dernière partie, nous avons finalement décidé d'utiliser l'algorithme IterativeImputer de la librairie Sklearn. Ce dernier permet d'imputer des données des features en utilisant la corrélation entre la feature à imputer et les autres features de notre dataset. Cela nous a permis d'éviter de potentielles erreurs sur nos calculs de ET logique entre les différents vecteurs, et donc permis de choisir les bonnes features à utiliser pour imputer nos features avec des données manquantes.

7.1 Feature Selection

Pour effectuer à nouveau la feature selection de notre dataset, maintenant complété, nous avons appliqué chaque méthodes (à savoir, Filter, Wrapper et Embedded) sur le dataset complet, le dataset ne comportant que les chiots de race de grande taille et le dataset ne comportant que les chiots de race de petite taille.
Contrairement à notre première feature selection, nous n'allons pas mettre les matrices de corrélations car ces dernières sont peu lisibles. Cependant, nous avons retenu pour chaque méthode et chaque application les 10 features les plus corrélées positivement avec la mort.
Pour la méthode Filter, nous avons donc :

Dataset Complet		Races de taille S		Races de taille L	
Breed	0.254188	PCR_CPV2_D24	0.896003	PCR_CPV2_D31	0.873356
IgG_dam	0.201954	PCR_CPV2_D17	0.710273	weight_D56	0.618438
breed_size	0.194271	dam_sep	0.268713	weight_D49	0.607277
Litter_size	0.163666	Litter_size	0.239756	weight_D42	0.554458
IgG_milk_mean	0.072328	PCR_CPV2_D45	0.237149	weight_D35	0.494499
PCR_CPV2_D45	0.069218	IgG_puppy_D28	0.174831	weight_D28	0.412601
dam_sep	0.062400	IgG_dam	0.122159	weight_D21	0.373835
IgG_puppy_D28	0.045540	PCR_CPV2_D38	0.118215	weight_D14	0.325685
age_dam	0.031035	fec_score_D49_56	0.048819	IgG_dam	0.268857
Abs_CPV2_D42	0.022135	group	0.041158	Breed	0.210283

Figure 13 : Résultats de la méthode Filter sur les 10 features

Avec la méthode Wrapper, nous avons obtenu les résultats suivants.

Dataset complet

```
Régression logistique :
('sex', 'breed_size', 'Breed', 'Litter_size', 'Diarr_2m', 'weight_D0', 'weight_D49', 'age_dam', 'IgG_milk_mean', 'IgG_puppy
_D2')
```

Races de taille S

```
Régression logistique :

('group',
 'Breed',
 'Diarr_1m',
 'Diarr_2m',
 'dam_sep',
 'weight_D0',
 'weight_D49',
 'IgG_dam',
 'Abs_CPV2_D2',
 'Abs_CPV2_D14')
```

Races de taille L

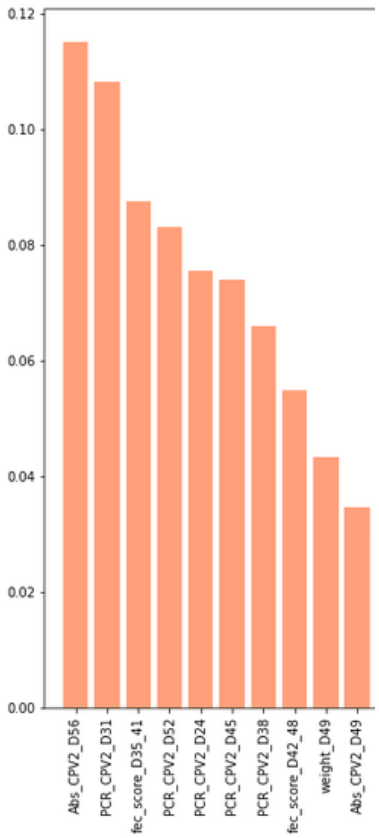
```
Régression logistique :

('sex',
 'group',
 'Breed',
 'Litter_size',
 'Diarr_1m',
 'Diarr_2m',
 'dam_sep',
 'weight_D2',
 'weight_D14',
 'weight_D49')
```

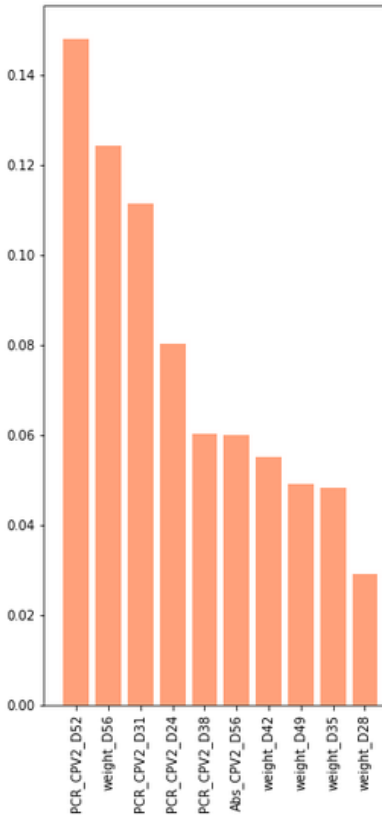
Figure 14 : Résultats de la méthode Wrapper sur les 10 features

Enfin, pour Embedded, nous avons obtenu les résultats suivants.

Embedded Dataset Complet



Embedded race petite taille



Embedded race grande taille

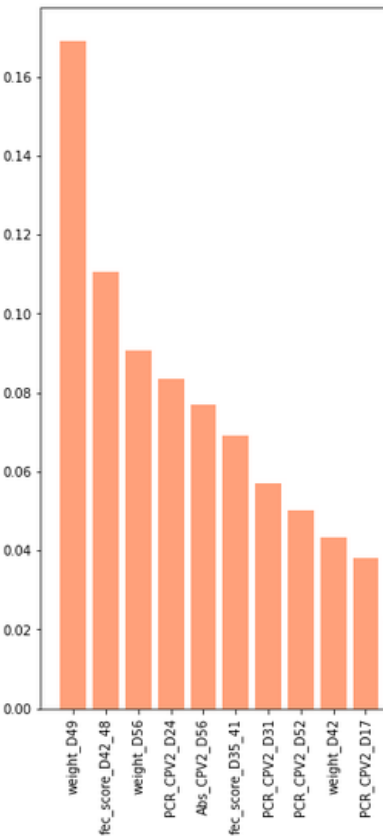


Figure 15 : Résultats de la méthode Embedded sur les 11 features

7.2 ACP

Reprenons la méthode utilisée pour le dataset comprenant les données complètes. Commençons par l'analyse des valeurs propres pour déterminer le nombre optimal de composantes principales.

	Val.Propre	Seuils
0	13.899134	4.394948
1	4.364138	3.394948
2	3.762999	2.894948
3	2.472914	2.561615
4	2.009125	2.311615
5	1.934039	2.111615
6	1.777265	1.944948
7	1.559741	1.802091
8	1.257115	1.677091
9	1.237449	1.565980
10	1.096438	1.465980
11	1.050827	1.375071
12	1.002724	1.291737
13	0.953529	1.214814
14	0.862913	1.143386
15	0.701499	1.076719
16	0.634574	1.014219
17	0.570973	0.955396

Figure 16 : Valeurs propres issues de l'ACP sur toutes les features

Selon la règle de Kaiser-Guttman, qui pour rappel dit qu'un axe est intéressant si sa valeur propre est supérieure 1. Ainsi, on retiendrait dix-sept composantes. En revanche, selon le test des bâtons brisés qui considère qu'on garde une composante seulement si sa valeur propre est supérieure à son seuil, on aurait seulement trois composantes. La différence entre les deux méthodes est considérable.

Tracer le screeplot nous éclairera sur cette différence. D'après la règle du coude, le nombre de composantes principales semble se situer autour de 5 features.

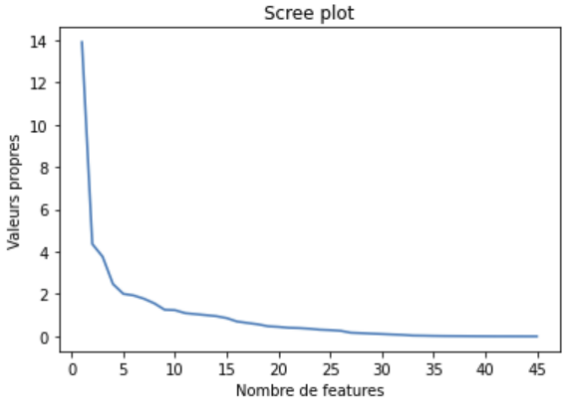
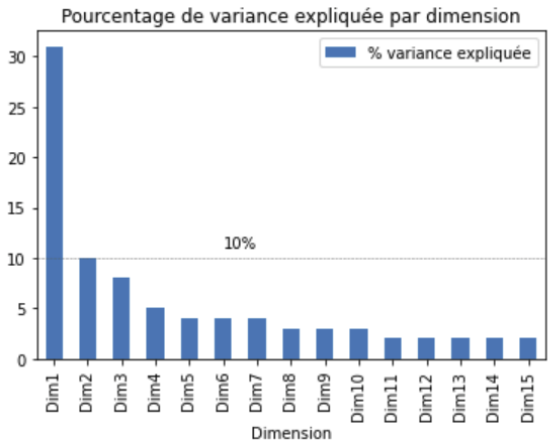


Figure 17 : Scree plot de l'ACP sur toutes les features

Pour nous aider dans notre décision, nous pouvons calculer le pourcentage de variance expliquée par chaque feature.



	Dimension	Variance expliquée	% variance expliquée	% cum. var. expliquée
0	Dim1	13.981867	31.0	31.0
1	Dim2	4.390115	10.0	41.0
2	Dim3	3.785398	8.0	49.0
3	Dim4	2.487634	5.0	54.0
4	Dim5	2.021085	4.0	59.0
5	Dim6	1.945551	4.0	63.0
6	Dim7	1.787844	4.0	67.0
7	Dim8	1.569025	3.0	71.0
8	Dim9	1.264598	3.0	73.0
9	Dim10	1.244815	3.0	76.0
10	Dim11	1.102965	2.0	79.0
11	Dim12	1.057081	2.0	81.0
12	Dim13	1.008693	2.0	83.0

Figure 18 : Variance expliquée par les dimensions de l'ACP sur les 10 features

On voit que c'est surtout la première dimension qui joue un grand rôle par rapport aux autres. Pour expliquer 80% de la variance, on devrait garder douze composantes. De même, les deux premières composantes expliquent seulement 41% de la variance.

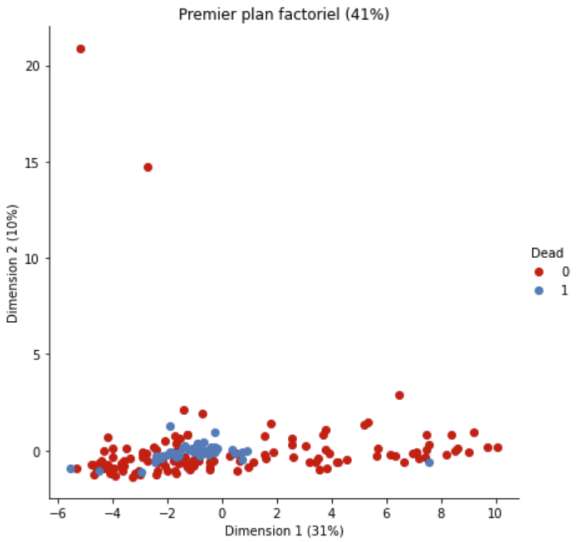


Figure 19 : Représentation des chiots avec l'ACP des 10 features suivant la mort

8. Gestion de projet

Nous n'avons pas pu compléter toutes les tâches du projet comme nous le voulions. Nous n'avons pas réussi à imputer les données avec la méthode utilisant le clustering individuel des features sans données manquantes. Nous avons à la place utilisé un Iterative Imputer qui était plus simple à mettre en place mais peut-être moins précis.

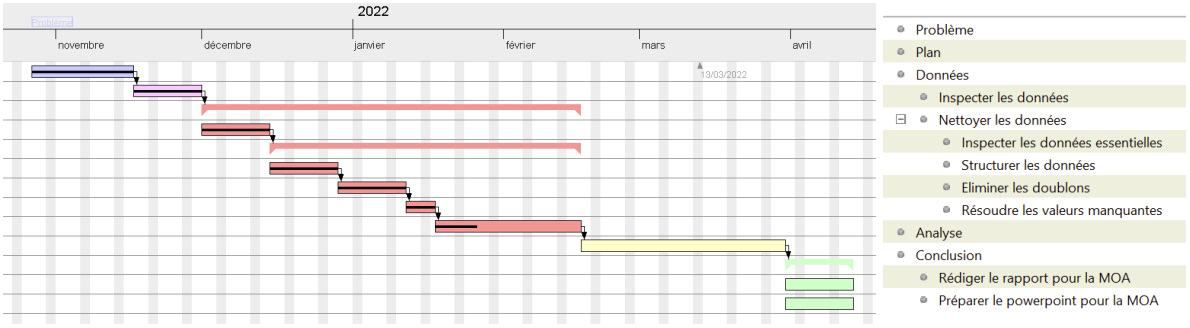


Figure 20 : Diagramme de Gantt du projet

Si un autre groupe venait à reprendre notre projet, il devrait refaire l'imputation des données en utilisant cette méthode comme nous voulions le faire initialement puis refaire la feature selection.

9. Conclusion

Nous avons imputé les données manquantes avec une méthode Iterative Imputer comme nous n'avons pas réussi à finaliser la méthode avec les clusters et le calcul des ET logique et distances de Hamming. Nous avons aussi essayé de réduire la dimension des features afin de faciliter leur interprétation mais cela n'a pas donné de résultats probants pour les features sans données manquantes. Quant à l'ACP sur les données une fois l'imputation faite, nous n'avons pas eu le temps de nous pencher sur la question. Nous avons seulement effectué la Feature Selection directement sur l'ensemble des données et non sur le résultat de l'ACP.
Finalement, nous avons trouvé que les features qui avaient le plus d'impact étaient les poids à la naissance, le taux d'immunoglobulines G de la mère, si le chiot a déjà eu la diarrhée ou encore le parvovirus.