# Analyse de Données

Abdoulaye Baradji et Gabriel Danelian

Janvier 2021

## 1 Introduction

Grace au développement de la technologie et dans un monde qui n'a jamais autant produit de données qu'aujourd'hui, l'analyse de données est devenu un vecteur déterminant dans l'amélioration et la connaissance de l'activité globale d'une entreprise.
D'où en fonction de l'objectif de l'étude, différentes méthodes pourront être utilisées afin d'extraire des informations plus pertinentes etc ...

Ainsi le but de ce projet consiste à appliquer les méthodes de classification sur différentes espèces d'animales, qui permettent d'identifier des sous-groupes dans les données mais aussi à partir des méthodes d'analyses factorielles nous allons réduire la dimension de notre jeu donnnées en utilisant l'analyse en composantes principales (ACP).

La première partie de ce projet consistera a utiliser les méthodes de classifications supervisées dans le sens où nous sommes en dispositions de nos classes et l'objectif sera de prédire l'appartenance à une classe d'un animal en fonction des caractéristiques de cet animal.

À noter que pour jauger la qualité predictive d'un modèle il est important de mesurer sa performance sur plusieurs jeu de données. C'est pour ça que nous allons d'abord subdiviser notre base de données initiales en un échantillon d'apprentissage qui fait 75% de l'échantillon initial sur lequel nous allons entrainer notre modèle de prédiction et en échantillon de test qui fait alors 25% de l'échantilon initial sur lequel nous allons mesurer la performance de ce modèle en calculant notamment le taux d'erreur commis en classant un individu dans une classe alors qu'il ne devrait pas y être.

La deuxième partie de ce projet consistera cependant à appliquer des méthodes de classifications non supervisées dans le sens où nous ne sommes pas en possession de nos classes et le but sera de regrouper en groupes les individus en fonction de photographies d'animaux. Plusieurs modèles seront appliqués afin de choisir le plus performant.
Ainsi pour mener à bien tout ceci nous allons d'abord faire une analyse descriptive de notre base de données afin de mieux preparer les données.

## 2 Partie 1 : Méthodes de classification supervisées

### 2.1 Analyse descriptive de notre base de données

Notre base de données contient en tout 101 individus sur 18 variables. la variable d'interêt c'est à dire la variable à expliquer dans notre cas est la variable nommée *class_type* qui contient les différentes classes possibles pour nos animaux. Cette variable est qualitative et contient plusiseurs modalités. Nous avons en tout 7 classes qui sont les suivantes :

- Mammal (mammifère)

- Bird (oiseau)

- Reptile (Reptile)

- Fish (poisson)

- Amphibian (amphibien)

- Bug (insecte)

- Invertebrate (invertébré)

Parmis les variables restantes appelées variables explicatives toutes les autres sont des variables booléennes à part la variable *animal_name* qui est un caractère et qui designe le nom des animaux mais aussi la variable qualitative *legs* qui designe le *nombre de pattes* chez un animal qui n'est pas binaire mais possède plusieurs modalités (0, 2, 4, 5, 6 ou 8 pattes).

L'objectif des méthodes de classifications supervisées étant de créer des modèles qui permettront de prédire l'appartenance à une classe en se servant des variables explicatives en études, il est impératif de controler les dépendances entre ces variables explicatives en supprimant celles qui ont une forte corrélation entre elles afin de permettre à notre base de données d'entrées d'être inversible. Nous allons de ce fait presenter le graphique de corrélation de notre base de données afin d'identifier en fonction de l'objectif de notre étude les variables à supprimer.

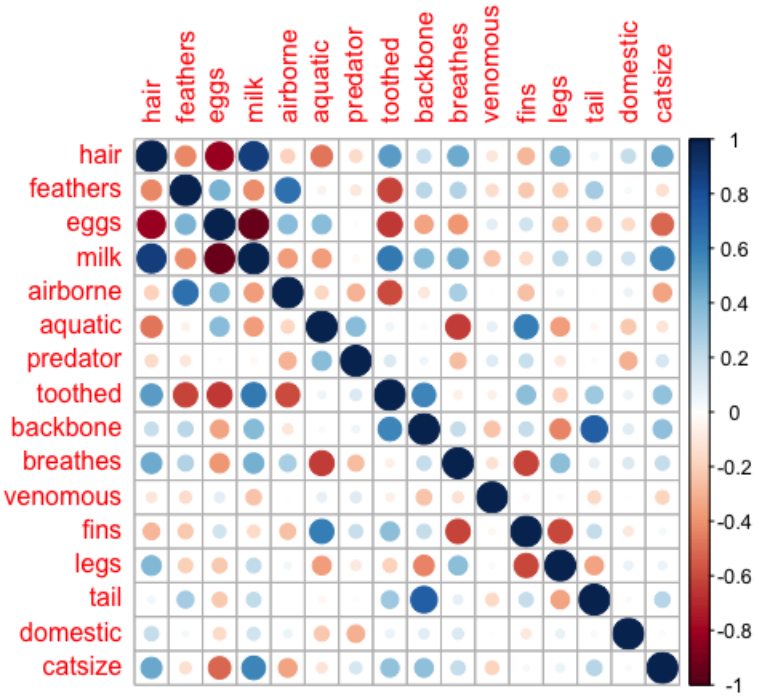


FIGURE 1: Corrplot matrice de corrélation entre les variables explicatives

Nous observons sur ce graphique que beaucoup de nos variables explicatives sont corrélées entre elles. En effet, le coéfficient de corrélation dépend de la profondeur de la couleur. Nous voyons que plus nos cercles sont de couleurs verts foncés et orange foncés plus les corrélations sont importantes. Nous allons cependant fixer un coefficient de corrélation de l'ordre de 0.60 dont nous avons decidé en fonction du nombre de variable d'étude. En effet si la corrélation entre deux variables depasse ce seuil 0.60 nous supprimerons l'une des variables. Ainsi nous voyons que la variable *hair* qui indique la présence de poils chez un animal a une forte corrélation avec les variables *eggs* et *milk* qui designent respectivement si l'animal pond des oeufs et si l'animal produit du lait. De même nous observons que la variable *eggs* a une forte corrélation avec la variable *milk*. De ce fait et en fonction de notre étude nous voyons que ces corrélations ont de sens car un animal qui a la présence de poils est susceptible de pondre des oeufs ou de produire du lait. Nous décidons donc de supprimer les variables *hair* et *eggs*.

De même nous observons une forte corrélation entre les variables *backbone* et *tail* qui designent respectivement la présence d'une colonne vertebrale et la présence de queue. Cependant, comparé à la presence de colonne vertebrale chez un animal il est plutôt rare d'avoir un animal qui a une queue. Donc nous decidons de garder la variable *tail* dans la suite de l'analyse et d'enlever la variable *backbone*.

On observe aussi que la variable $feathers$ qui indique la présence de plumes chez un animal a une forte corrélation avec les variables *airbonne* et $Toothed$ qui désignent respectivement si l'animal a la possibilité de voler et si l'animal a des dents. Une forte corrélation existe aussi entre cette variable *toothed* et la variable *milk*. Nous decidons donc d'enlever la variable $feathers$ de la suite de l'analyse ce qui était évident car elle dépend très fortement de deux autres variables mais aussi nous avons fait le choix de supprimer la variable *toothed* dans la suite de l'analyse au lieu de la variable *milk* car peu d'animaux ont cette capacité de produire du lait cela peut être un facteur important pour le but de notre étude qui est d'expliquer l'appartenance des animaux dans une classe.

Pour finir nous observons aussi que la variable $fins$ qui indique la présence de palme a une forte corrélation avec les variables *aquatic*, *breathes* et *legs* qui designent respectivement si c'est un animal aquatique, si l'animal à la possibilité de respirer et le nombre de pattes de l'animal mais aussi cette même variable *breathes* est corrélée très fortement avec la variable *aquatic*. Cependant nous supprimons de la suite de l'analyse la variable $fins$ mais aussi la variable *breathes* car en fonction des animaux qui ont la capacité de respirer peu d'animaux cependant sont acquatiques.

D'où nous pouvons observer sur la figure 2 qu'après suppression de nos variables les corrélations ne sont plus importantes.

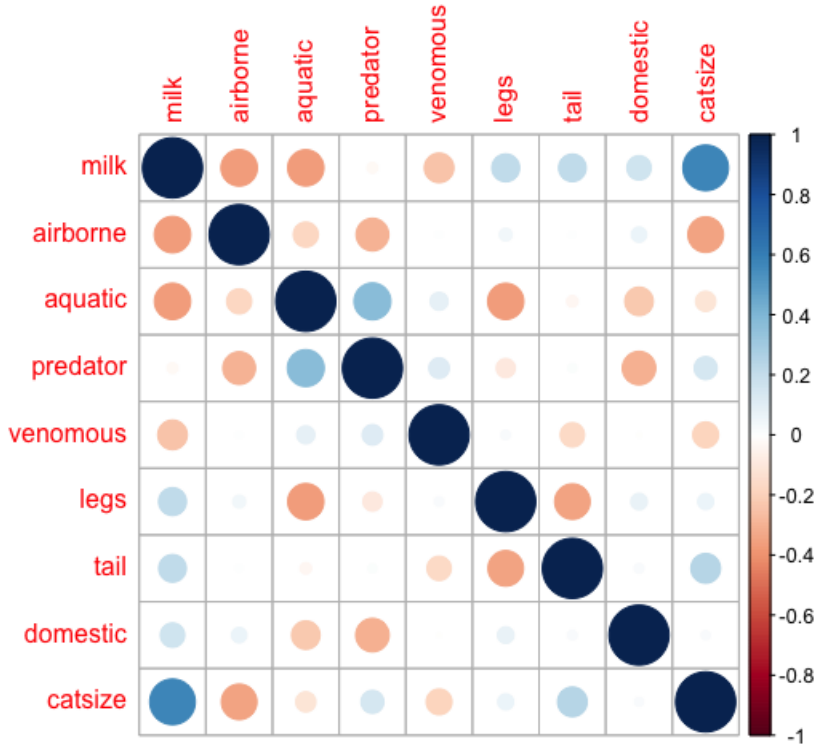


FIGURE 2: Corrplot après élimination des corrélations fortes

Ainsi nous disposons maintenant de notre matrice de données qui est linéairement independante, ce qui est très important pour la suite de notre analyse. Nous avons maintenant en tout 10 variables explicatives dont 9 seulement sont utiles pour l'analyse car nous enlèvons la variable *animal_name* qui n'indique seulement le nom des animaux. Ces variables sont toutes qualitatives.

Nous avons recodé toutes nos 9 variables explicatives en plus de la variable d'interêt qui est *class_type* en facteur car elles sont toutes qualitatives. Cependant nous avons remarqué que la variable *legs* présente

des modalités qui sont mal répresentées ce qui peut biaiser nos modèles de classification supervisées. Ainsi nous avons decidé de regrouper certaines modalités entre elles à savoir les animaux qui ont respectivement 5 et 8 pattes avec les animaux qui ont 6 pattes.

Nous allons maintenant proposer un modèle de classification supervisée pour notre matrice de données à l'étude. En raison de la nature binaire ou qualitative de nos variables, une classification avec une méthode géométrique n'est pas appropriée et nous nous concentrer sur les méthodes utilisant des arbres de décision. Pour commencer nous décidons d'appliquer les modèles de $RandomForest$.

## 2.2 Modèle 1 : RandomForest

En classification supervisée, les forêts aléatoires font partie de méthodes de bagging qui elles font partie de la famille des méthodes d'ensemble dont le principe est basé sur l'utilisation de plusieurs échantillons boootstrap pour construire la famille de classifieurs. Cependant comme nous l'avons vu, ils sont developpés dans le cadre spécifique des arbres de décision $CART$ dans le sens où ces derniers, de part leur instabilité, sont de bons candidats aux méthodes d'ensemble.
À noter aussi que cet algorithme fait d'une part de la classification mais aussi de la regression. Notre objectif ici est de faire de la classification d'où le travail effectué pour le recodage de nos variables explicatives en facteur.

En appliquant cet algorithme de classification supervisée sur notre échantillon d'apprentissage et en regardant la performance préditive sur l'échantillon test avec $ntree = 20$ qui indique le nombre d'arbre à inclure dans le modèle. Cependant il doit être elevé pour garantir que chaque ligne d'entrée soit prédite au moins une fois et $mtry = 1$ qui indique le nombre de variables échantillonnées au hasard comme candidats à chaque division de l'arbre. Nous obtenons ainsi avec ces paramètres un taux de mauvais classement de l'ordre de 30% environ. En effet cet taux d'erreur est relativement fort pour le choix de ces paramètres. Nous allons cependant essayer de voir l'influence du nombre d'arbres ou du nombre de variables sur le taux d'erreur pour pouvoir finalement voir si en réduisant ou augmentant ces paramètres nous pouvons avoir un taux de mauvais classement plus faible afin de jauger au mieux notre modèle de prédiction.

### 2.2.1 Effet du nombre de variables dans le modèle

Nous remarquons sur la figure 3 que plus le nombre de variable dans notre modèle est élévé plus le taux de mauvais classement a tendance à diminuer. On peut cepedant noter que pour 1 et 2 variables le taux de mauvais classement est passé d'environ 34% à 17%.
pour mieux apprécier la décroissance et comment elle se stabilise, nous allons tracer le graphe suivant :

Ainsi nous observons sur la figure 4 que le taux d'erreur du mauvais classement reste stable autours de 8 variables dans le modèle avec un taux d'erreur de l'ordre de 10%.

Pour la suite nous allons donc prendre $mtry = 8$.

### 2.2.2 Effet du nombre d'arbres dans la forêt sur le taux de mauvais de classement

Comme précédemment, traçons le boxplot du taux d'erreur de mauvais classement en fonction du nombre d'arbre dans la forêt.

Nous pouvons ainsi observer sur la figure 5 que plus le nombre d'arbres est élévé plus le taux d'erreur de mauvais classement à tendance à diminuer. Il passe de 30% environ à 13% pour $ntree = 1$ et $ntree = 500$. Comme tout à l'heure pour une meilleure visualisation regardons le graphe 6 suivant :

Ainsi nous allons maintenant reprendre notre modèle avec ces paramètres ajustés à savoir $ntree = 500$ et $mtry = 8$.

Nous obtenons finalement un taux d'erreur de mauvais classement de l'ordre 4% environ, ce qui correspond à un individu mal classé.

Il serait maintenant important de voir les variables qui rentrent le plus dans la construction de notre modèle.

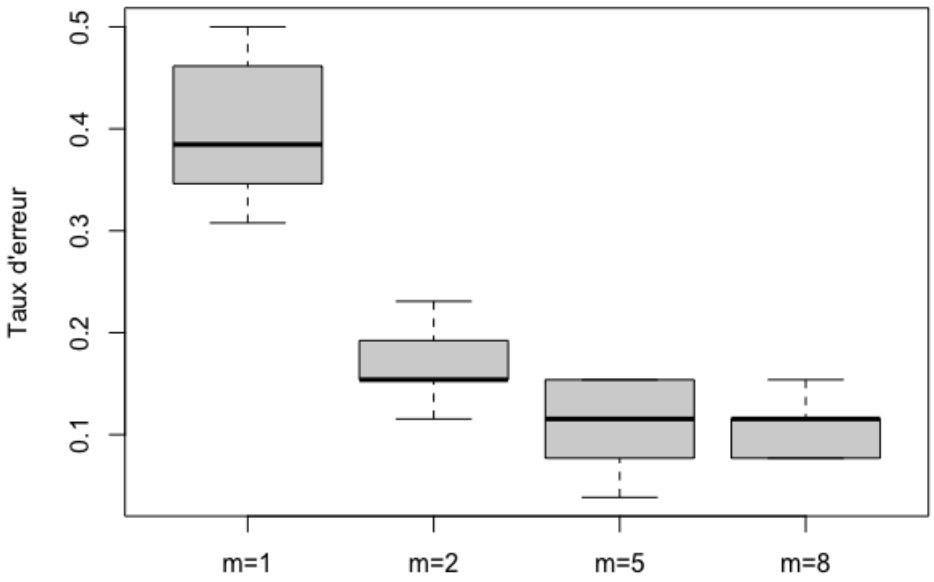


FIGURE 3: Boxplot du Taux d'erreur de mauvais classement en fonction du nombre de variable

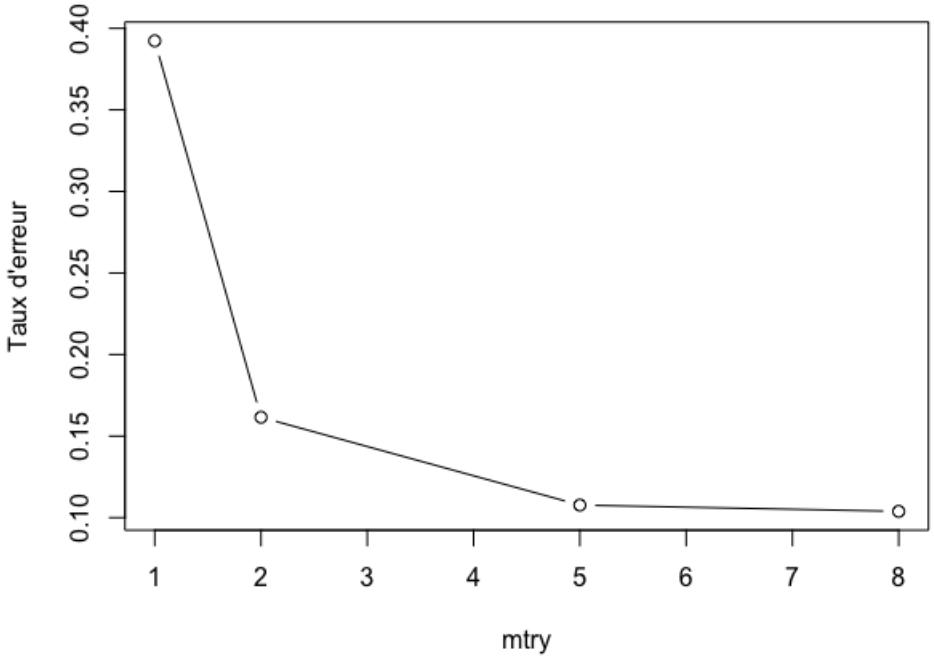


FIGURE 4: Taux de mauvais classement en fonction du nombre de variable

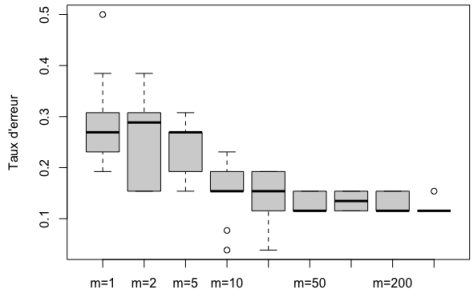


FIGURE 5: Boxplot du taux d'erreur de mauvais classement en fonction du nombre d'arbres

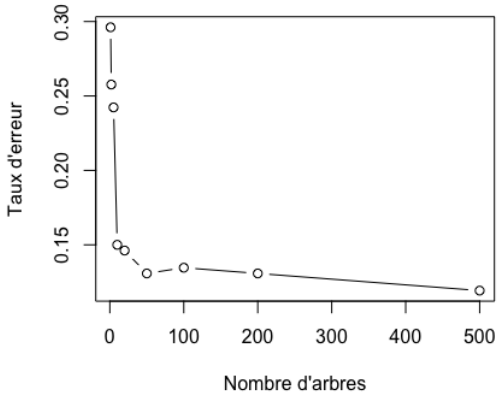


FIGURE 6: Taux d'erreur de mauvais classement en fonction du nombre d'arbre

### 2.2.3 Caractéristiques qui influencent le plus dans la création du modèle

Dans la sortie de notre fonction $randomforest$, la fonction $ImportanceSD$ permet d'obtenir le classement des variables qui participent à une division de l'arbre et $VarImpPlot$ permet d'avoir le graphique. En effet cette fonction indique la diminution moyenne de la précision sur toutes les prédictions.

L'analyse de la figure 7 nous permet de voir qu'en fonction du choix de nos paramètres pour ce modèle la variable *milk* qui indique si l'animal produit du lait est la plus importante dans la prédiction de nos classes. Puis viennent ensuite les variables *legs*, *tail*, *aquatic* etc ...
À noter qu'elle nous permet juste de voir l'ordre d'importances des variables qui ont été selectionnées pour participer à une division mais ne nous permet pas de voir cependant les variables qui participent effectivement à la création d'une classe precise. Néamoins elle nous permet d'avoir un aperçu global sur ceux qui sont beaucoup utilisées et très importantes.

Nous allons maintenant presenter ce tableau suivant qui indique les prédictions obtenues par le modèle $RandomForest$ sur notre échantillon test contenant 26 individus soit 25% de l'échantillon initiale avec les vraies classes.

L'analyse de la table 1 nous montre effectivement que la précision de la prédiction pour notre modèle $RandomForest$ est meilleure sur notre échantillon test. Ainsi nous pouvons voir que tous sauf un individus dans l'échantillon test ont bien été prédits. En effet, l'individu 73 *scorpion* a été prédit comme étant un reptile plutôt qu'un invertébré. Nous allons donc étudier l'arbre de décision et les paramètres qui y sont considérés pour pouvoir comprendre les caractéristiques du scorpion qui font qu'il est affecté à la mauvaisse classe.

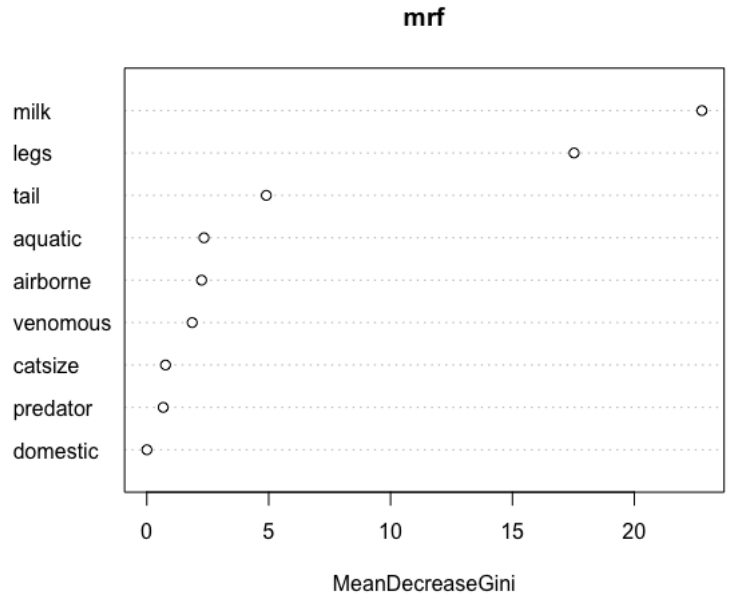


FIGURE 7

| Animal Number | Animal Name | Predictions | True Class |
|---|---|---|---|
| 3 | Bass | Fish | Fish |
| 6 | Buffalo | Mammal | Mammal |
| 9 | Catfish | Fish | Fish |
| 10 | Cavy | Mammal | Mammal |
| 12 | Chicken | Bird | Bird |
| 13 | Chub | Fish | Fish |
| 32 | Goat | Mammal | Mammal |
| 37 | Hare | Mammal | Mammal |
| 41 | Housefly | Bug | Bug |
| 47 | Lobster | Invertebrate | Invertebrate |
| 54 | Octopus | Invertebrate | Invertebrate |
| 62 | Piranha | Fish | Fish |
| 65 | Polecat | Mammal | Mammal |
| 66 | Pony | Mammal | Mammal |
| 70 | Raccoon | Mammal | Mammal |
| 73 | Scorpion | Reptile | Invertebrate |
| 78 | Seawasp | Invertebrate | Invertebrate |
| 80 | Skua | Bird | Bird |
| 81 | Slowworm | Reptile | Reptile |
| 83 | Sole | Fish | Fish |
| 89 | Termite | Bug | Bug |
| 90 | Toad | Amphibian | Amphibian |
| 92 | Tuatara | Reptile | Reptile |
| 94 | Vampire | Mammal | Mammal |
| 98 | Wasp | Bug | Bug |
| 100 | Worm | Invertebrate | Invertebrate |

TABLE 1: Tableau comparant les classes prédites par le modèle randomforest avec les classes initiales sur l'échantillon test

## 2.3 Etude de l'arbre de décision et des autres algorithmes

La base de test donnée ci-dessus donne un taux d'erreur similaire entre les méthodes de forêts aléatoires et de bagging pour les meilleurs paramètres de chacunes de ces méthodes, le scorpion étant toujours l'individu mal classé. Le package *randomForest* ne se prêtant pas bien à la visualisation, nous allons présenter un des arbres de décision du bagging (figure 8) car il aboutit à la même classification et l'algorithme est similaire au forêt aléatoire, en enlevant l'aléa sur le choix des variables. Ces deux algorithmes ont néanmoins le problème d'effectuer un vote majoritaire sur un grand nombre d'arbres pour aboutir à une classification et ne donnent donc pas un arbre de décision définitif. L'arbre CART en revanche ne donne qu'un seul arbre mais donne un taux d'erreur plus élevé que le bagging pour notre échantillon. Cependant, l'arbre CART en ne regroupant pas les animaux ayant plus de cinq pattes donnait un taux d'erreur de 4% avec un arbre identique à celui présenté, mis-à-part que l'on effectuait la division selon $legs = 6$ plutot que $legs = 5+$. Il est donc raisonnable de considérer que l'arbre 8 donne la meilleure classification pour nos données (même si d'autres arbres données par le bagging aboutissent à cette classification avec des critères différents). Le paramètre masqué par une boîte est $legs = 5+$, et le côté gauche d'un noeud indique que le trait est présent chez l'animal, et le côté droit indique qu'il ne l'est pas (par exemple le côté droit de $milk = 0$ dit qu'il n'y a pas une absence de lait, et donc que l'animal produit du lait). Nous cher-

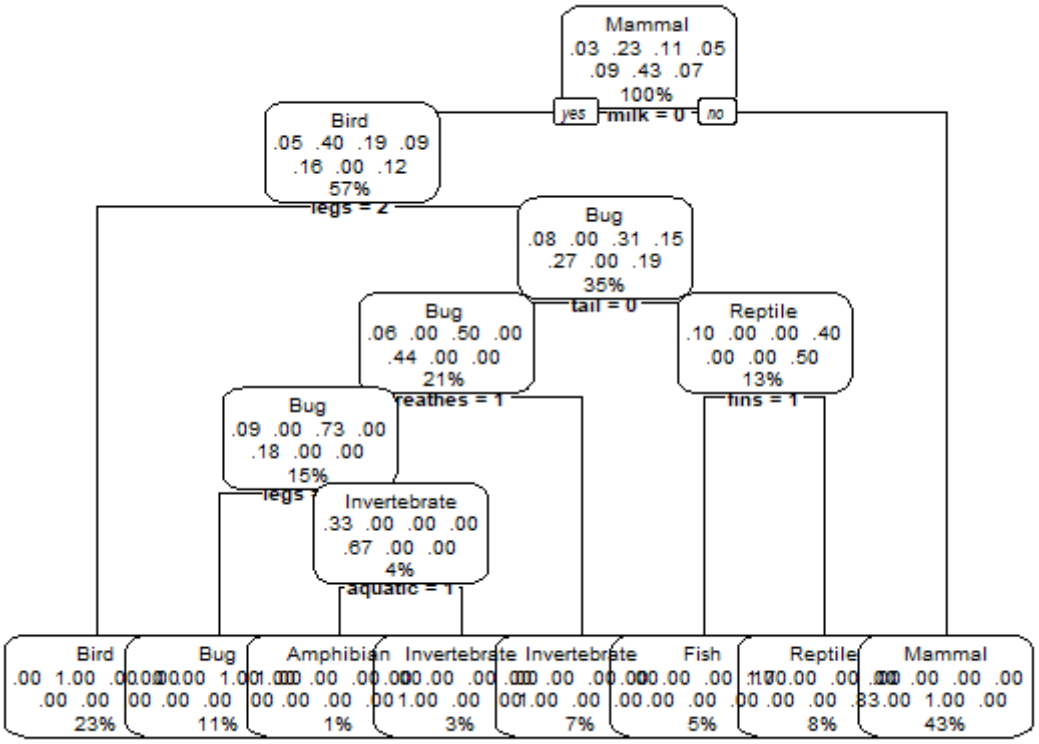


FIGURE 8: Arbre de décision CART, le paramètre caché étant $legs = 5+$

chons à savoir pourquoi le scorpion a été mal classifié en tant que reptile plutôt qu'en tant qu'invertébré, et voyons sur l'arbre que l'erreur s'est produite car les reptiles sont différentiés des invertebrés en ayant une queue, ce qui est le cas du scorpion. En comparant les caractéristiques du scorpion aux reptiles, nous voyons un des inconvénients que peut avoir le retrait des variables corrélés. En effet,la queue et la colonne vertébrale sont les deux seuls paramètres présents chez tous les reptiles, ce qui fait explique leur forte corrélation, mais la queue est présente chez le scorpion tandis que la colonne ne l'est pas. Ainsi, n'ayant que la queue comme variable présente chez tous les reptiles, les algorithmes ont raisonnablement placé le scorpion dans cette catégorie.

Si nous considérons désormais l'ensemble de l'arbre, nous voyons que l'algorithme a trié les mammifères comme ayant du lait et les oiseaux comme ayant deux pattes, qui sont des critères relativement logiques. L'arbre distingue ensuite les reptiles des poissons par la présence de nageoires, mais les critères pour classer un individu comme invertebré, amphibiens ou insectes sont moins clairs, avec notamment deux

branchements qui mènent aux invertébrés. Ceci est sans doute du à la variété des animaux qui constituent les invertebrés, avec des animaux marins et terrestres, avec et sans pattes, et aucun attribut qui est possédé par tous les membres du groupe. Notre arbre reste tout de même assez intuitif, avec un attribut distinctif qui émerge pour la plupart des groupes, et pourrait correspondre à un arbre phylogénétique que l'on utiliserait en biologie.

Rappelons finalement que les résultats que nous avons présenté précédemment sont issus d'une coupe aléatoire de nos données en base d'apprentissage et en base de test, et que donc les taux d'erreurs entre les méthodes vont varier à chaque découpe. Nous avons choisi de présenter la méthode des forêts aléatoires car elle requiert un travail plus important dans le choix des paramètres, mais les méthodes de bagging avec un arbre maximal ont globalement le même taux d'erreur peu importe les échantillons. Il est tout de même important de prendre un arbre maximal car nous sommes dans un contexte où les classes sont déjà assez générales et donc réduire le nombre de classes davantage en prenant un arbre élagué agrandit grandement le taux d'erreur. De manière générale, effectuer une classification avec un nombre de classes élevé encourt le risque du sur-apprentissage, c'est-à-dire de trop adapter notre classification à nos données actuelles, qui fonctionnera alors mal avec des données nouvelles. Réduire le nombre de classes en sortie de notre modèle réduit le nombre de paramètres qui entrent en compte et permettent une meilleure généralisation du modèle. Cependant, nous effectuons ici une classification de tout le monde animal et les classes et paramètres que nous étudions sont très larges dans le contexte biologique, et pourraient même gagner à être affinés. On ne risque pas le sur-apprentissage car des nouvelles données vont être définies selon les mêmes catégories que nos données, et réduire le nombre de classes entraîne automatiquement une augmentation de l'erreur par une mauvaise classification des classes qui n'ont pas été retenues. Par exemple, un élagage de l'arbre CART précédent ne contient plus d'amphibiens, d'invertebrés ou de reptiles, en les classant tous comme des insectes. Au delà du taux d'erreur clairement insuffisant, une telle classification n'a pas de sens d'un point de vue biologique car les amphibiens, invertebrés, insectes et reptiles sont fondamentalement différents les uns des autres. Les méthodes de bagging ne fonctionnent guère mieux avec un arbre élagué, avec le bagging qui choisit d'enlever les mêmes classes que l'arbre CART. Les méthodes de bagging avec un arbre maximal sont des méthodes très performantes sur nos données, avec des taux d'erreur similaires aux forêts alétoires, et un arbre CART maximal fonctionne bien également, avec parfois un individu mal classé supplémentaire, en étant une méthode bien plus simple et intuitive. Nous avons passé sous silence le boosting car il donne une classification légèrement différente de celle présentée ci-dessus, mais cette méthode fonctionne aussi bien que le bagging sur plusieurs découpages de nos données avec le même taux d'erreur en général, et donc est également appropriée pour notre étude.

# 3 Partie 2 : Méthodes de classification non supervisée

Nous allons désormais utiliser des méthodes de classification non supervisée dont le but sera de prédire les classes d'animaux à partir de photographies d'animaux que nous avons à notre disposition dans la base de données.

Pour commencer nous allons dans une première partie traiter notre base de données en essayant de faire ressortir les variables les plus importantes en perdant le minimum d'informations sur notre jeu de données. Pour cela nous allons utiliser l'analyse en composante principale (ACP) qui permet de grandement réduire la dimension de nos données en gardant le maximum d'informations, ici représenté par la qualité de l'image.

Nous allons d'abord présenter la matrice de données que nous utilisons pour notre analyse avant de préciser la différence avec une ACP habituelle. Nos données sont constituées de 4738 images d'animaux sauvages, la majorité étant des grands chats (lions, tigres et léopards principalement) et le restant étant des canins (loups et renards). Contrairement à la première partie, nous ne donnons pas à nos algorithmes une catégorisation des images et choisissons notre méthode de classification en fonction de sa cohérence et si l'on arrive à distinguer un critère selon laquelle elle a été effectuée. Chacune de nos images est constitué de trois carrés de 128 par 128 pixels, en couleurs rouge, vert et bleu. On va linéariser chaque image en formant un vecteur constitué de 128 fois 128 pixels pour chacune des trois couleurs que l'on concatène ensemble. On peut désormais construire la matrice de nos données en mettant une image à chaque ligne, ce qui donne une matrice de taille 4738 x 49152 (128 fois 128 fois 3). L'interêt de l'ACP pour réduire la dimension de la matrice est désormais évident, car le temps de calcul pour une matrice de cette taille est extrêmement prohibitif. La taille de la matrice était d'ailleurs trop importante pour pouvoir être stockée en mémoire, ce qui nous a forcé à n'utiliser que les 2500 premières images de notre base de données. Ceci n'est pas problématique car les images sont ordonnées arbitrairement et le nombre d'images retenu est suffisamment large pour que les conclusions soient similaires à celles que l'on aurait obtenu avec l'échantillon complet.

On a donc une matrice avec $n = 2500$ individus et $p = 49152$ variables pour laquelle nous voulons réduire la dimension (le nombre de variables). Cependant, lors d'une ACP classique, on a un nombre de variables qui est inférieur à celui d'individus ce qui nous donne des règles empiriques pour le choix du nombre d'axes factoriels à conserver, tels que chercher un coude dans le graphe des valeurs propres ou atteindre 80% de l'inertie totale. On a ici un nombre $p$ de variables bien supérieur au nombre d'individus ce qui fait que ces critères ne sont plus valables et l'on devra regarder les images produites pour différentes valeurs propres et voir si les animaux sont reconnaissables. Remarquons que l'ACP nous renvoie une matrice de taille 2500 par 2500 (2500 composantes principales de longueur 2500) et donc qu'utiliser cette matrice serait déjà un gain considérable en mémoire et en temps de calcul. Les propriétés de l'ACP nous indiquent cependant que l'on pourrait ne conserver qu'un nombre réduit des premiers axes factoriels et avoir une qualité d'image proche de l'image originale. On joint à la fin du rapport (figures 11 à 15) une comparaison des images en conservant 200, 500, 1000 et tous les axes factoriels (en allant de gauche à droite et de haut en bas).

Dans l'ensemble, 200 axes sont insuffisants avec les images 8 et 9 qui sont méconnaissables, l'image 10 est visible mais de piètre qualité. 500 axes permettent de reconnaître l'animal car on commence à distinguer les motifs du pelage (notamment l'image 9 du léopard des neiges que l'on reconnaît malgré la mauvaise qualité de l'image). On va cependant choisir de conserver 1000 axes car les images avec 500 axes sont de trop mauvaise qualité tandis que les images avec 1000 axes sont presque indistinguables de l'image originale, avec simplement une petite perte de définition entre les deux.

Introduisons les divers algorithmes que nous allons utiliser pour notre classification. Les valeurs de notre matrice originale sont des réels compris entre 0 et 1 qui représentent la proportion de chaque couleur rouge, vert et bleu qui sont utilisés. Ainsi, des différences et des similarités de couleurs se traduisent par des écarts ou des rapprochements entre les valeurs de la matrice. Les valeurs de la matrice réduite donnée par l'ACP ne sont plus comprises entre 0 et 1 mais ces différences de valeurs reflètent toujours des différences de couleurs. On va donc utiliser des algorithmes qui calculent les distances entre les individus et qui classifient ensemble les individus qui sont proches les uns des autres. Par la nature géométrique de ces méthodes, nous appelerons parfois les individus des points.

L'algorithme des k-means consiste à choisir $k$ points (centres) au hasard, à construire $k$ classes en assignant chaque point au centre dont il est le plus proche (au sens de la distance euclidienne généralement), à calculer les nouveaux centres des classes, à réassigner chaque point au centre dont il est le plus proche, et effectuer ces dernières étapes jusqu'à que les classes se stabilisent.

La classification ascendante hierarchique (CAH) place chaque point dans sa propre classe, et calcule

la matrice des distances entre tous les individus. On regroupe les deux individus les plus proches en une classe, on définit le centre (l'inertie de ces deux points) de cette classe comme un point puis on recalcule les distances entre tous les points et on regroupe les deux points les plus proches en une classe, et l'on répète ces manoeuvres jusqu'à avoir une seule classe. On peut représenter le regroupement des classes sous la forme d'un arbre appelé dendogramme où l'on peut voir la distance entre chaque classe qui est représentée par la longueur de chaque branche du dendogramme.

L'algorithme DBSCAN choisit un point au hasard, et s'il y a $m$ points dans un rayon $\epsilon$ de ce point, regroupe tous ces points dans un même groupe, et si ce point n'a pas $m$ points dans son $\epsilon$-voisinage il est considéré comme un point isolé. On répète ce procédé pour tous les points.

Bien que ces algorithmes soient facilement implémentables sous R, ils sont sujets à plusieurs paramètres dont le choix influence grandement la classification, ce qui requiert de lancer plusieurs fois chaque algorithme en faisant varier les paramètres. L'algorithme k-means requiert le choix d'un nombre de classes, qui n'est pas immédiat dans notre cas. Etant donné les images que nous avons, on pourrait songer à une division entre les animaux avec un pelage uniforme, comme les lions ou les loups, et les animaux avec un pelage à motifs, comme les tigres ou les léopards. Une classification idéale parviendrait à distinguer chaque espèce d'animal , mais il est sans doute ambitieux de s'attendre à une classification aussi fine. Cela nous donnerait de cinq classes (si l'on considère les cinq espèces principales de lion, tigre, léopard, loup et renard) à neuf ou dix classes (si l'on sépare les lions mâles des femelles et les autres animaux en fonction de leur couleur). Il serait également possible que l'on obtienne une classification basée sur les couleurs de l'animal, comme par exemple une classe jaune et une classe blanche.

Essayons donc l'algorithme k-means pour un nombre de classes égal à 2 car nous avons deux qualificatifs, les motifs de pelage et la couleur des animaux, qui se divisent en deux groupes. Malheureusement, le fait que nous n'ayons pas d'étiquettes pour chaque image fait que nous ne pouvons pas savoir le nombre d'animaux de chaque espèce qui ont été placés dans les groupes. Nous allons donc sauvegarder dans un fichier les images pour chaque classe donnée par l'algorithme et voir si un motif devient apparent. Les k-means donnent une répartition avec 1231 individus dans une classe et 1269 individus dans l'autre, mais il n'émerge pas de distinction claire entre ces groupes, avec un mélange d'animaux de toutes les espèces et de toutes les couleurs. La classification k-means n'est donc pas satisfaisante pour k=2.

Plutôt que de tester toutes les valeurs de k jusqu'à 10, il serait préférable de pouvoir estimer le nombre de classes autrement que par notre observation visuelle. L'algorithme CAH permet cela, car comme nous avons mentionné précédemment celui-ci produit un arbre dont la hauteur des branches représente l'écart entre les classes qui se trouvent à ces noeuds. On représente en figure 9 le dendogramme pour nos données avec comme mesure de distance la distance de Ward, qui est la plus usuelle. Il pousse également à choisir deux classes, avec un très grand écart avant d'avoir une nouvelle division, même si une division en trois ou cinq classes semble raisonnable. On rappelle également que l'algorithme fonctionne en regroupant des classes deux à deux à chaque itération. Nous effectuons le raisonnement inverse, en partant du haut de l'arbre pour voir si nous observons un effet et en agrandissant le nombre de classes, ce qui signifie qu'à chaque étape une classe va se diviser en deux, les autres classes restant inchangées. Cela implique que nous connaissons dans l'ensemble l'effet qu'aura l'augmentation par un du nombre de classes. En effet, si nous avons par exemple une très bonne partition en deux classes selon un critère, nous pouvons regarder les groupes d'une partition en trois classes en sachant que l'un des groupes initiaux restera inchangé et que l'autre groupe initial sera divisé en deux, ce qui préservera la distinction selon le critère initial mais permettra possiblement de faire apparaître une classification selon un autre critère. A l'inverse, si nous avons une partition sans critère évident, une nouvelle division avec une mesure de dissimilarité faible entre les deux groupes risque de ne pas être fructueuse.

Nous pouvons nous servir de ce nombre de classes pour nous aider à choisir $k$ dans notre algorithme k-means, mais nous allons d'abord regarder la classification que fournit l'algorithme CAH. Une division en deux classes par la CAH donne 1176 individus dans un groupe et 1324 dans le second et semble effectuer une distinction selon la couleur, avec le deuxième groupe qui contient la majorité des animaux au pelage blanc. Bien que ce soit une amélioration considérable par rapport à l'algorithme k-means, cette distinction est imparfaite car elle n'est pas très fine, avec quelques animaux blancs qui font partie du groupe 1, et car la majorité de animaux de nos données sont jaunes/oranges et sont répartis sans critère évident entre les deux groupes. Ainsi, la classification par la CAH n'est pas vraiment appropriée pour nos données car elle répartit en groupes de tailles similaires selon un critère qui ne s'applique qu'à une petite portion des individus de notre échatillon, mais pourrait mieux fonctionner pour des données constituées d'un nombre plus important d'animaux blancs. On augmente le nombre de classes pour voir si l'algorithme va séparer notre second groupe selon la couleur du pelage, ce qui nous permettrait d'affiner notre classification. Une CAH avec 3 classes divise le second groupe en deux groupes de taille 357 et de taille 967 que

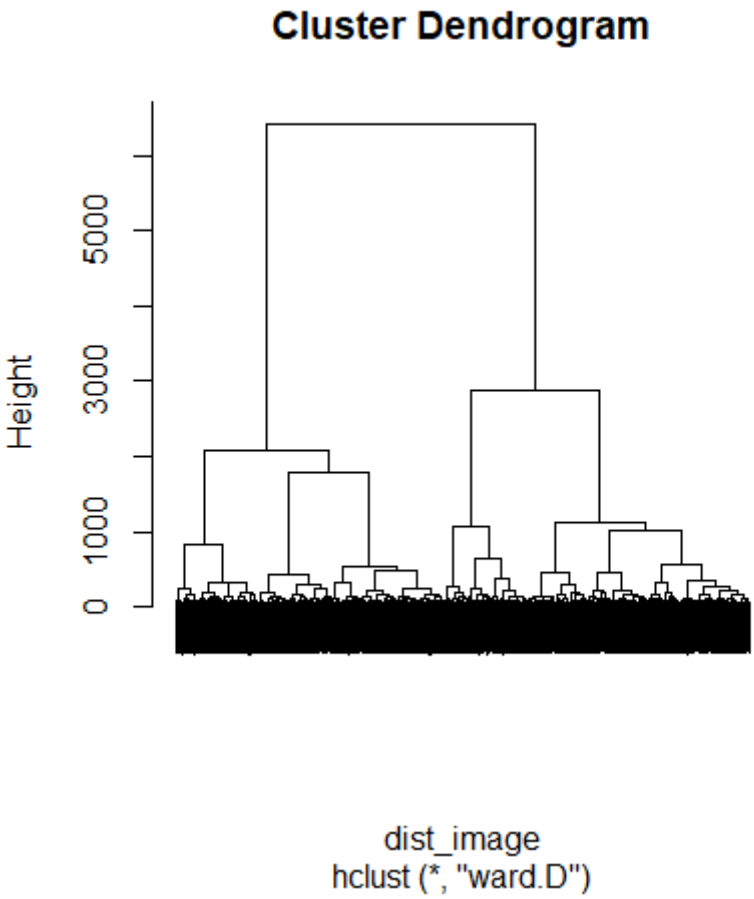


FIGURE 9

nous nommerons groupe 2 et groupe 3 respectivement. Il n'est pas vraiment clair si cette subdivision est fructueuse car le groupe 2 possède une grande proportion d'individus blancs (environ un-tiers), bien supérieure à celle dans l'échantillon original, mais le groupe 3 en possède toujours un nombre significatif également, ce qui montre que cette subdivision est très grossière. La classification en trois groupes est intéressante pour faire valeur d'illustration car la présence des animaux blancs est visuellement frappante dans le groupe 2 par rapport à la classification en deux groupes et aux données originelles, et pourrait être utile si notre objectif était de trouver les animaux blancs dans notre échantillon. En effet, le faible nombre d'animaux blancs dans l'échantillon total fait que la taille similaire des groupes dans la CAH à deux classes donnerait toujours une majorité d'animaux jaunes dans le groupe, même si on avait une discrimination parfaite pour mettre tous les animaux blancs dans un seul groupe, alors que l'introduction d'un troisième groupe plus petit qui contient un grand nombre d'animaux blancs réduit automatiquement le nombre d'animaux jaunes dans ce groupe. Par contre, si notre objectif est d'obtenir le maximum d'animaux blancs sans nous soucier du nombre d'animaux jaunes dans la classe, une classification en deux groupes reste préférable car la subdivision du second groupe n'est pas très fine et laisse un nombre non négligeable d'animaux blancs dans le groupe 3 de la CAH à trois classes.

Les divisions en 4 et 5 classes s'effectuent sur le premier groupe de la CAH à deux classes (les deux branchements à une hauteur de 1500 sur le côté gauche du dendogramme) ce qui signifie qu'elles ne vont probablement pas créer de divisions intéressantes car le premier groupe n'avait pas vraiment de propriété distinguante. En effet, une CAH à quatre classes divise le premier groupe en un groupe de taille 365 et un de taille 811, la CAH à cinq classes divisant ce dernier groupe en deux de taille 316 et 495, mais il ne semble pas y avoir de critère évident pour ces classifications. Ainsi, la CAH donne une classification raisonnable pour une division en deux ou trois groupes, mais l'ajout de groupes supplémentaires n'est pas approprié.

On peut également se servir de ces résultats pour l'algorithme k-means. En effet, étant donné la faible différence entre les divisions au-delà de cinq classes montrées par la CAH, on peut limiter nos essais de

l'algorithme k-means à cinq classes ou moins. Similairement à k=2, l'algorithme des k-means pour d'autre valeurs de k ne donne pas une classification satisfaisante. Il semblerait donc que cet algorithme ne soit pas approprié en général pour nos données, peu importe le nombre de classes $k$ que nous cherchons à trouver.

L'algorithme EM (Espérance-Maximisation) est une méthode qui s'inspire des méthodes de mélange en classification supervisée, mais qui requiert donc l'estimation des classes car celles-ci sont inconnues dans notre cadre. L'algorithme est assez technique donc nous allons nous contenter de donner ses résultats sur nos données en passant les explications de la théorie derrière. Cet algorithme requiert le choix du nombre de classes et, similairement à l'algorithme k-means, on ne distingue pas de critères évidents selon laquelle la classification s'effectue, siginifiant que l'algorithme n'est pas adapté à nos données, et nous n'allons donc pas nous attarder dessus.

L'algorithme DBSCAN détermine le nombre de classes présentes dans les données, mais nécessite un choix judicieux du rayon $\epsilon$ et du nombre de points $m$ qui devraient se trouver dans ce rayon, des paramètres auxquels les résultats de l'algorithme sont très sensibles. En pratique nous choisissons d'abord le nombre de points proches $m$ et calculons pour chaque individu de notre base la distance qui le sépare du $m$-ième plus proche individu. On trace ensuite cette distance pour tous les individus et on cherche un coude dans les valeurs ce qui indique que la majorité des individus ont $m$ voisins à cette distance. La courbe des distances prend la forme d'une sigmoïde ce qui fait que nous obtenons plutôt deux coudes et l'on peut justifier toute valeur de $\epsilon$ entre ces deux coudes. DBSCAN est un algorithme rapide et très sensible au rayon $\epsilon$ ce qui fait qu'il est judicieux de tester plusieurs valeurs de $\epsilon$ raisonnables, et d'effectuer notre choix en fonction de l'interprétabilité des résultats. Dans le cas classique $p << n$, on prend généralement $m = p + 1$ ou $m = 2p$. Etant donné que nous sommes dans un cas où $n << p$, nous allons prendre $m = \sqrt{p} = 50$ et ajuster cette valeur si les résultats ne sont pas satisfaisants.

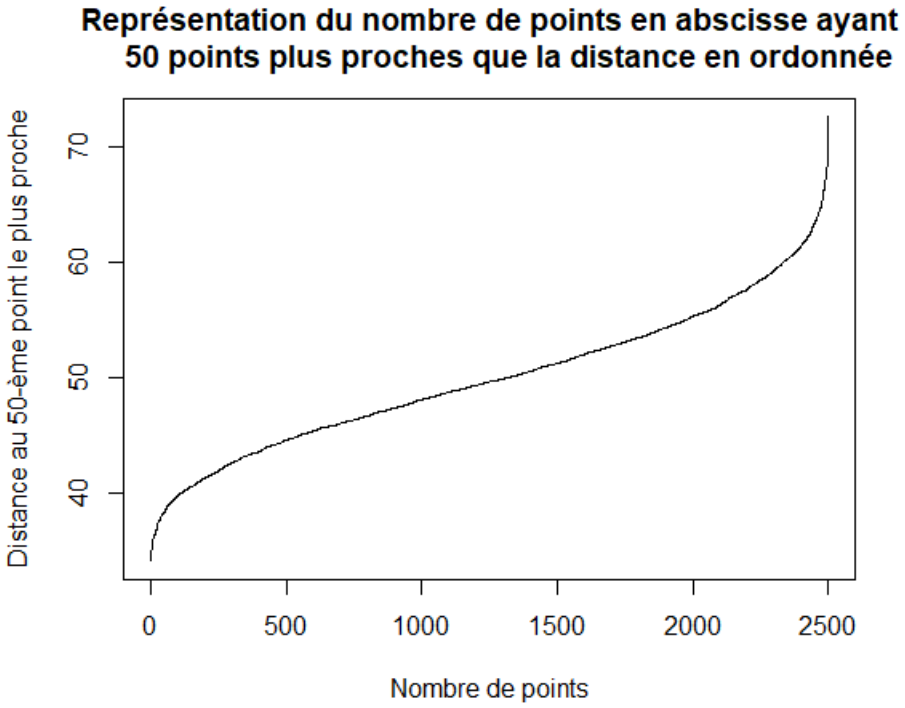


FIGURE 10

La figure 10 nous donne en ordonnée une distance allant de 35 à 75 et en abscisse le nombre de points qui ont au moins cinquante points dans un rayon égal à cette distance. Pour nos données, la distance minimale pour avoir cinquante points dans un rayon est de 35 environ et au delà de 70 tous les points de l'échantillon ont cinquante points dans leur voisinage. On voit ici pourquoi le choix de $\epsilon$ à $m$ fixé est essentiel car si nous prenons un $\epsilon$ inférieur à 30, l'algorithme va considérer tous les points comme des outliers car aucun point n'aura 50 points dans un voisinage de taille $\epsilon$ tandis que pour une valeur de 70 tous les points seront placés dans le même groupe car ils auront tous au moins cinquante points dans leurs voisinages. Ainsi, ce graphique nous donne une idée du nombre d'individus qui seront dans chaque

groupe pour des $\epsilon$ différents et montre la raison pour laquelle on sélectionne une valeur de $\epsilon$ entre les deux coudes du graphe, pour avoir un plus grand nombre d'individus placés dans le second groupe. On va donc tester différentes distances entre 35 et 60 et regarder la classification donnée pour voir si elle paraît raisonnable. Un $\epsilon$ de 60 est mauvais car la classification place 2489 individus dans un groupe et en considère 11 comme des outliers. Par contre, un $\epsilon$ de 35 donne un groupe de 105 individus avec les 2395 non classés, ce qui ne semble pas être très bon car la grande majorité des individus n'ont pas été classés, mais les individus qui composent le groupe sont presque tous des lions, avec quelques renards. Il reste tout de même quelques léopards mal classés, mais la séparation est très bonne et nous donne espoir qu'un $\epsilon$ plus grand donnera une classification de qualité pour un nombre plus important d'individus. Malheureusement, la classification avec $\epsilon = 40$ donne un groupe de 734 individus qui est beaucoup plus hétérogène avec désormais autant de léopards que de lions et même quelques tigres, et qui n'a donc pas de critère distinctif.

Pour $\epsilon = 45$, on obtient un groupe de 1498 individus mais ce sont les 1002 outliers qui nous intéressent car ils sont composés majoritairement de tigres et de léopards, donc des animaux au pelage à motif. Il reste quand même des lions dans ce groupe mais la séparation est très bonne étant donné la taille du groupe. Un $\epsilon$ de 50 donne une encore meilleure séparation, avec un groupe composé presque entièrement de tigres, de léopards et d'animaux au pelage blanc, mais qui ne contient que 406 individus. Cependant, la qualité de la discrimination entre tigres/léopards et autres animaux pour un $\epsilon$ de 50 fait que nous n'avons pas grand intérêt à réduire la valeur de $\epsilon$ davantage, car on réduit la taille du groupe sans vraiment améliorer notre classification (un $\epsilon$ de 55 donne un groupe de 99 individus).

Une idée pour améliorer notre classification est de subdiviser notre échantillon en enlevant les individus du groupe donné par la classification avec $\epsilon = 35$ (principalement des lions), et de refaire l'algorithme DBSCAN sur l'échantillon réduit avec $\epsilon$ égal à 45 ou 50 pour essayer de distinguer les tigres/léopards. Malheureusement, la classification est bien moins bonne en mettant plus de lions dans le groupe de tigres, ce qui fait que nous préférons la classification originale sur l'échantillon complet.

En conclusion, il est possible d'obtenir un petit groupe de lions avec un $\epsilon$ faible de 35, mais qui se dégrade rapidement en augmentant $\epsilon$, avec beacoup de léopards pour $\epsilon \geq 37$. La classification la plus intéressante est donnée par un $\epsilon$ de 45-50 qui permet de séparer les individus au pelage à motifs (les tigres et léopards) des autres, avec un groupe de taille considérable de 1000 individus pour $\epsilon = 45$.

## 3.1 Conclusion

Après avoir examiné quatres méthodes géométriques de classification non supervisée, nous en avons trouvé deux, les k-means et l'Espérance-Maximisation qui ne semblent pas appropriées à nos données car les groupes qui en ressortent n'ont pas de critère distinctif. La Classification Ascendante Hiérarchique donne des résultats meilleurs car elle effectue une séparation en fonction des couleurs en plaçant la plupart des animaux au pelage blanc dans un même groupe. Cependant, notre échantillon contient une minorité d'animaux blancs et le groupe créé par la CAH contient beaucoup d'autres couleurs. Cette classification n'est donc pas idéale pour nos données mais pourrait produire de très bons résultats avec d'autres données ayant une répartition plus équilibrée entre les couleurs des animaux.

L'algorithme le plus performant sur nos données est de loin DBSCAN qui arrive à créer un petit groupe composé de lions pour une faible valeur de $\epsilon$ de 35, et crée un grand groupe de tigres et de léopards pour des valeurs plus élevées de $\epsilon$ entre 45 et 55. Les très bons résultats de DBSCAN sur nos données nous permettent de dire avec confiance que cet algorithme fonctionnera bien pour des jeux de données différents, notamment pour différencier les animaux avec des motifs différents sur leur pelage.

Une manière potentielle d'améliorer la classification sur nos données serait d'effectuer plus algorithmes à la suite, par exemple en ne gardant qu'un seul groupe issu de la classification DBSCAN et de réeffectuer l'algorithme DBSCAN ou CAH dessus. Nous avons essayé cela pour le groupe donné par l'algorithme DBSCAN avec $\epsilon = 35$, ce qui n'a pas donné de meilleurs résultats, mais nous n'avons pas pu essayé toutes les combinaisons possibles de paramètres. En effet, il y a également les valeurs de $\epsilon$ de 45, 50 et 55 qui donnaient des classifications de qualité, et l'on devrait tester plusieurs paramètres de l'algorithme DBSCAN et CAH sur l'un des groupes issus de ces classifications. On pourrait également faire varier le paramètre $m$, mais des essais avec $m = 20$ semblaient donner des résultats assez similaires. Néanmoins, on peut s'épargner du temps car les mauvais résultats des algorithmes k-means et EM font qu'il ne semble pas intéressant de tester ces algorithmes plus loin.

# 4 Annexe : Comparaison de qualité d'images pour différentes réductions de dimension

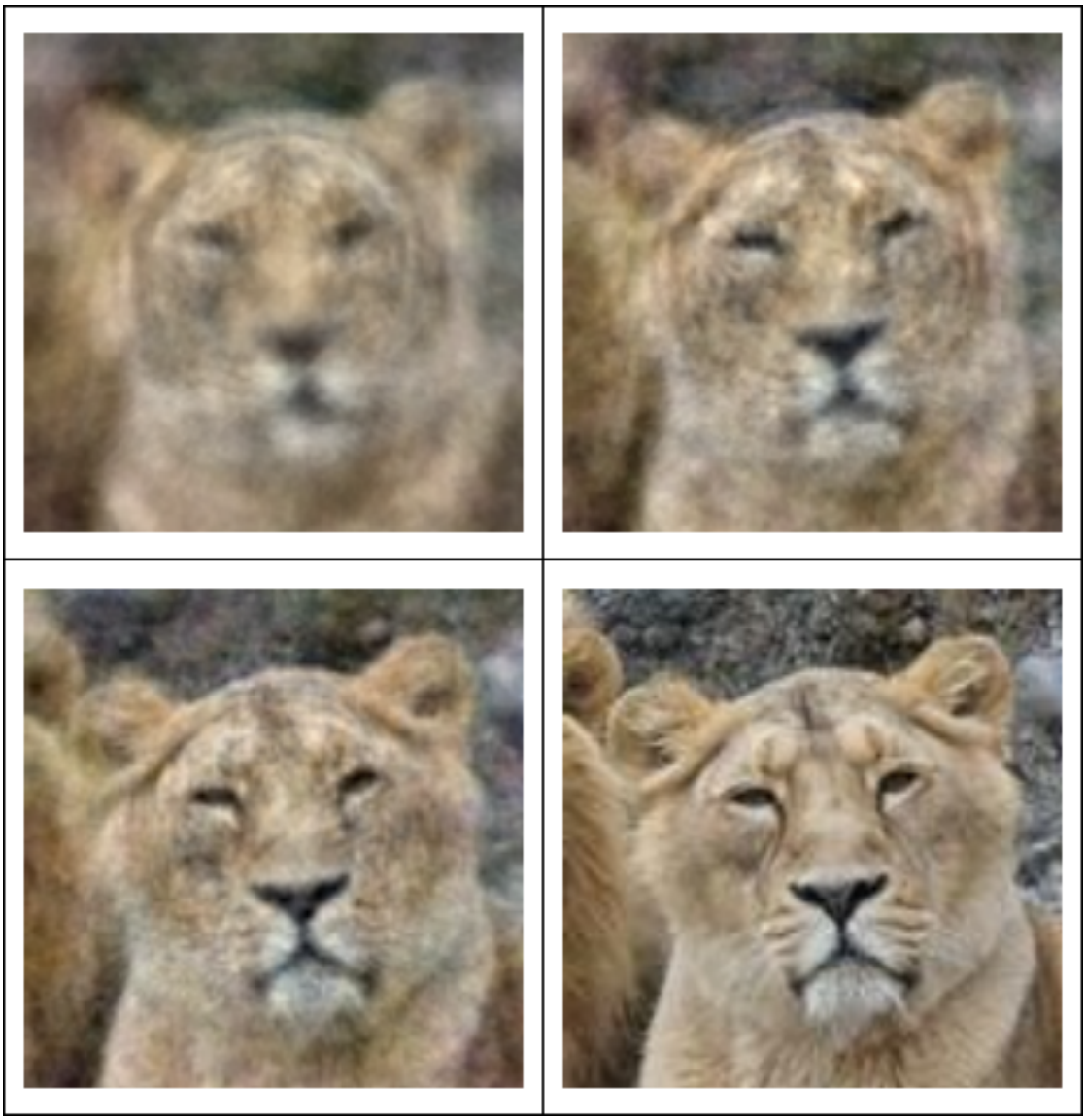

Figure 11

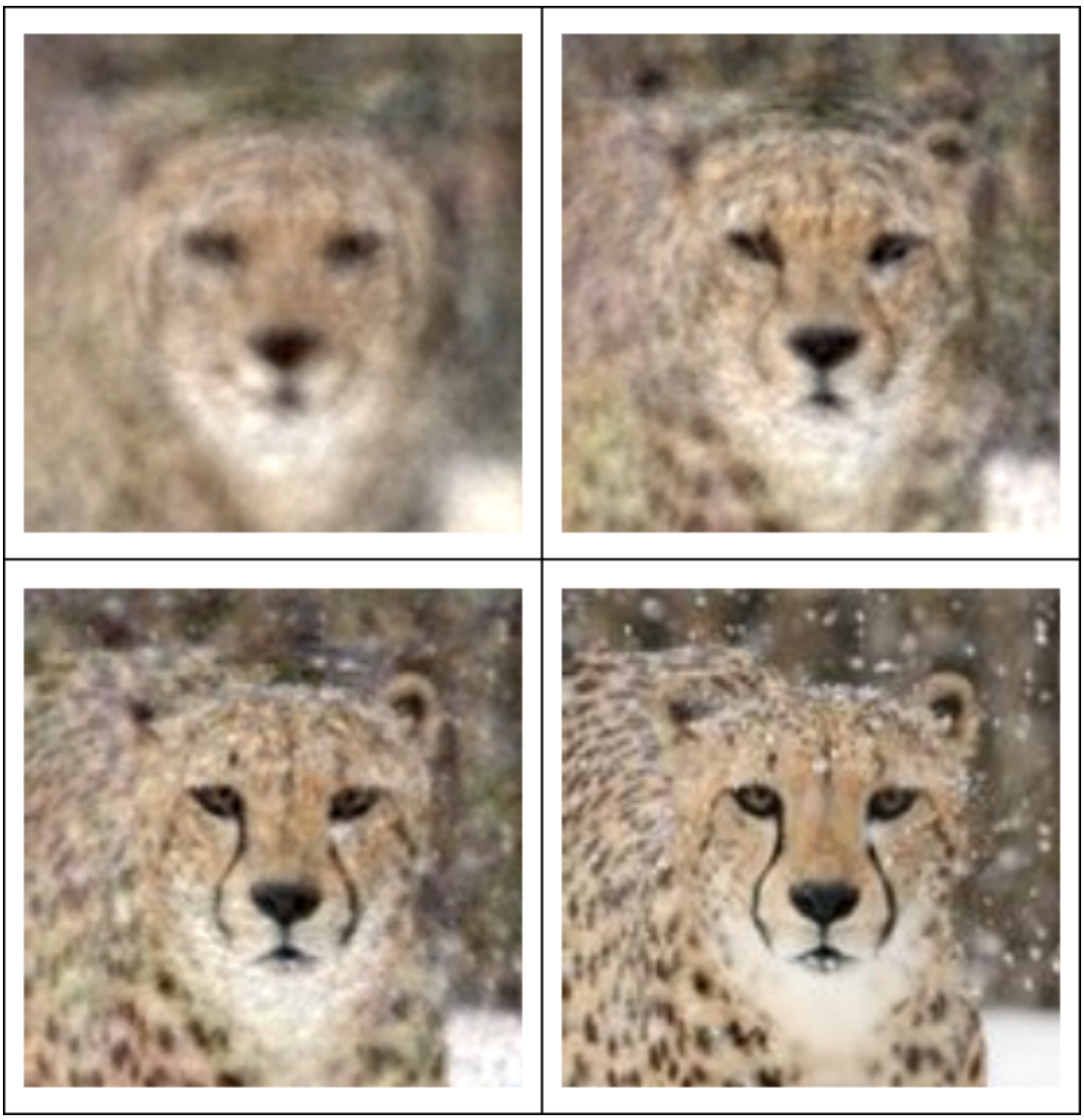

FIGURE 12

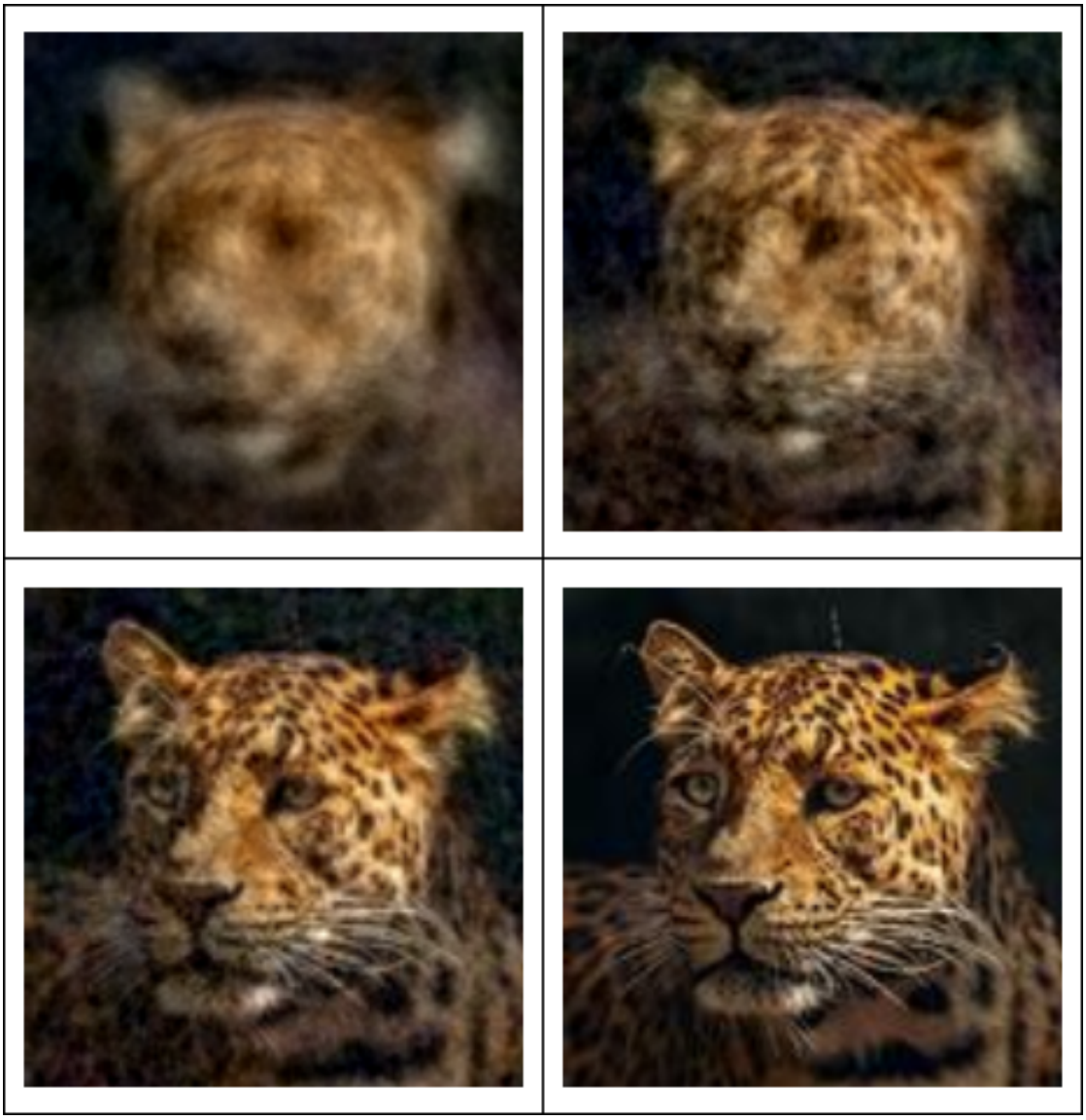

FIGURE 13

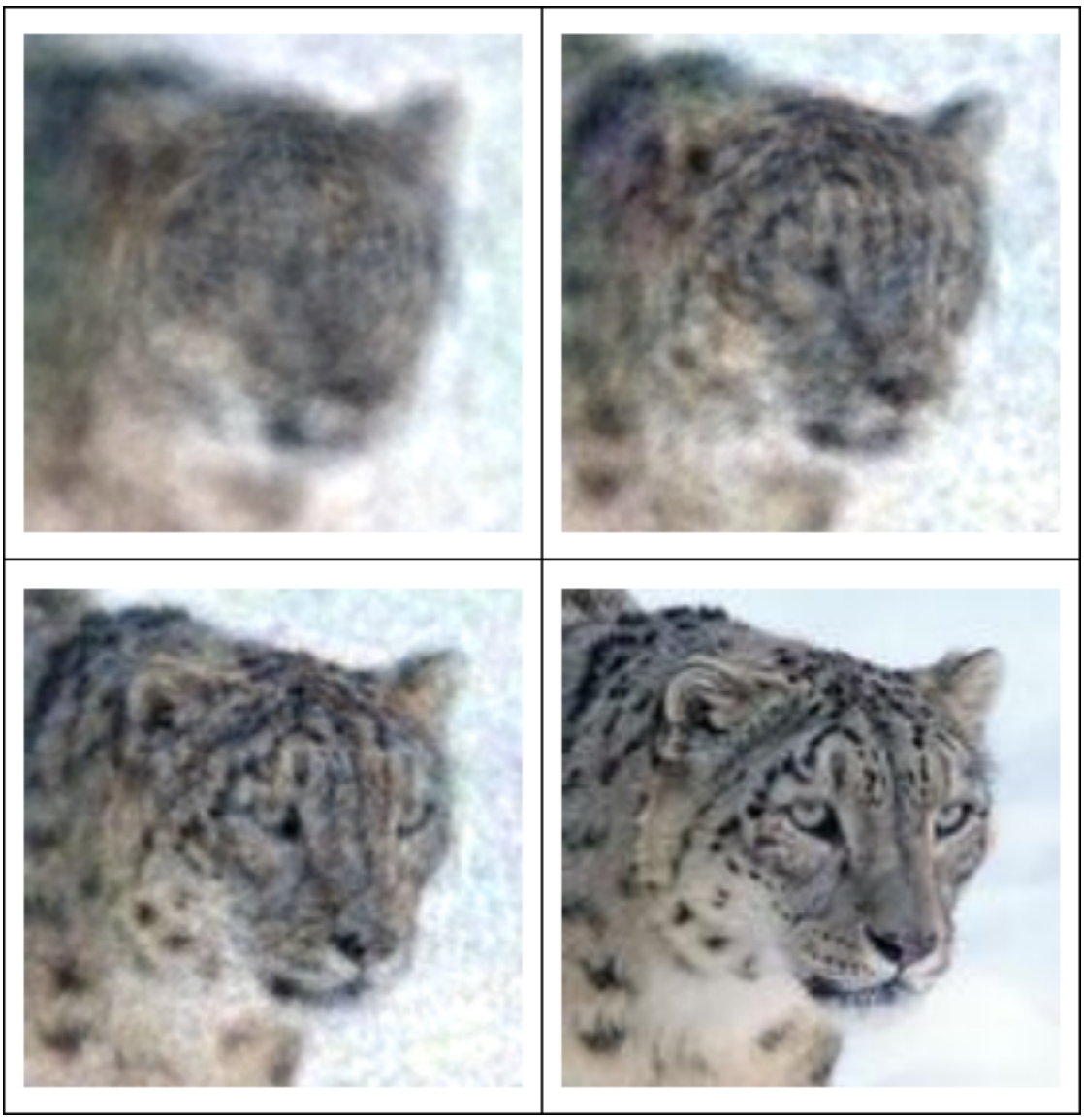

FIGURE 14

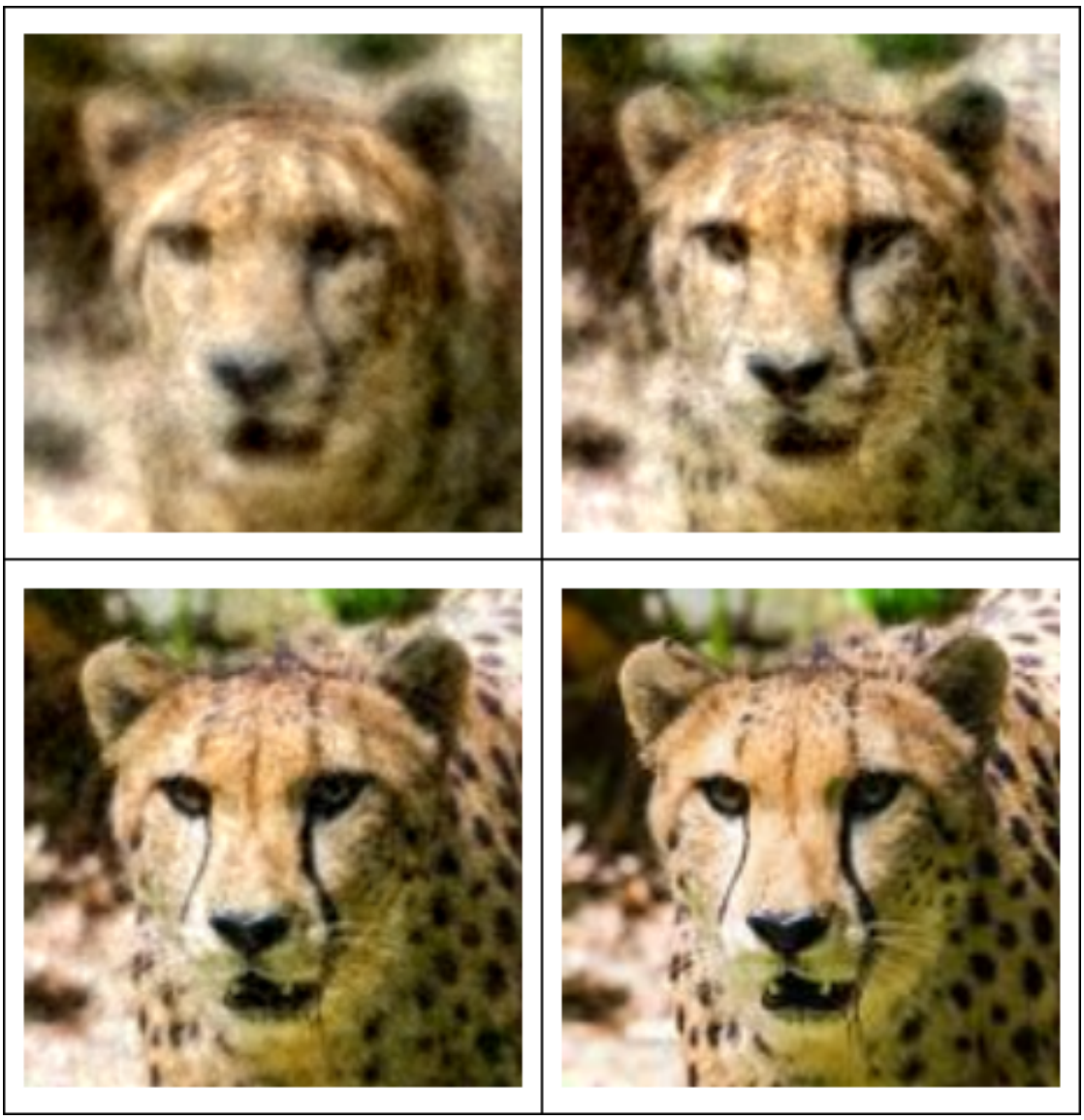

FIGURE 15