Centrale
CASABLANCA

RAPPORT

Travaux pratiques du module Deep Learning

Réalise par :

- **Ange-Marie EXAUCE KPEDETIN GOUNADON**
- **Sourou Alfred SODJI**
- **Adam Ez-zahir**

Encadré par :

- **M. Vincent Lefieux**

22 décembre 2023

I. Table des matières

Liste des figures :

II. Exercice 1 : Jeu de données ozone complet

A. Introduction :

La qualité de l'air est un pilier essentiel de la santé publique et de la préservation de l'environnement.Le pic d'ozone journalier représente un indicateur clé de cette qualité atmosphérique et a des répercussions significatives sur la santé des individus ainsi que sur l'écosystème dans son ensemble. En effet, une exposition prolongée à des niveaux élevés d'ozone peut entraîner des problèmes respiratoires et cardiovasculaires, affectant particulièrement les personnes sensibles telles que les enfants et les personnes âgées.

Ce rapport s'attache à explorer différentes approches de modélisation pour anticiper le pic d'ozone du jour suivant. Pour ce faire, plusieurs modèles ont été considérés, notamment la régression linéaire, le random forest, le gradient boosting, et les réseaux de neurones profonds (DNN). L'objectif est d'évaluer la capacité de chaque modèle à prédire avec précision le niveau de pollution par l'ozone et d'identifier le modèle le plus performant pour cette tâche spécifique.

L'une des facettes cruciales de cette étude réside dans l'évaluation du nombre de caractéristiques (features) utilisées pour chaque modèle. Cette sélection méticuleuse des features impacte directement la capacité prédictive des modèles. Ainsi, cette analyse se concentre sur la pertinence et l'impact de ces caractéristiques dans la prédiction du pic d'ozone. Les conclusions tirées de cette exploration permettront de mieux appréhender l'importance de la sélection des features dans la modélisation des problèmes environnementaux complexes tels que la prédiction du pic d'ozone.

1. Contexte du problème :

L'ozone, un polluant atmosphérique majeur, représente un défi continu pour la qualité de l'air. Il est formé par des réactions chimiques complexes entre les oxydes d'azote (NOx) et les composés organiques volatils (COV) en présence de lumière du soleil. Les concentrations élevées d'ozone dans l'air sont associées à divers impacts néfastes sur la santé, tels que l'aggravation des problèmes respiratoires existants, l'irritation des voies respiratoires et des poumons, voire des effets à long termesur le système respiratoire.

Le pic d'ozone journalier est particulièrement significatif car il représente la concentration la plus élevée d'ozone enregistrée au cours d'une journée. Prédire avec précision ces pics est crucial pour prendre des mesures préventives, informer le public et mettre en place des politiques de gestion de la qualité de l'air.

L'environnement urbain, avec ses diverses sources de pollution, est souvent le théâtre de concentrations élevées d'ozone. La variabilité des conditions météorologiques, des émissions anthropiques et des caractéristiques géographiques contribue à la complexité de ce problème. Par conséquent, la modélisation du pic d'ozone demeure un défi scientifique et technique important, nécessitant une approche multidisciplinaire et des modèles de prédiction précis.

Dans ce contexte, cette étude s'inscrit dans une démarche visant à améliorer la capacité de prédiction du pic d'ozone du jour suivant en utilisant différentes techniques de modélisation. La précision de ces prédictions pourrait avoir un impact significatif sur la santé publique en permettant des avertissements précoces et des recommandations pour réduire l'exposition à des niveaux élevés d'ozone.

2. Objectif de la modélisation :

L'objectif principal est d'évaluer et de comparer l'efficacité de plusieurs modèles de prédiction pour anticiper le pic d'ozone journalier. Cette analyse vise à identifier le modèle le plus performant et à évaluer l'impact du nombre de caractéristiques sur la qualité des prédictions.

3. Importance du Choix du Nombre de Features :

La sélection des caractéristiques (features) joue un rôle fondamental dans la construction de modèles de prédiction précis et robustes. Choisir les bonnes caractéristiques peut améliorer la capacité du modèle à généraliser les tendances et les relations sous-jacentes présentes dans les données.

Cependant, la sélection de features n'est pas simplement une question de quantité. Une multitude de caractéristiques peut entraîner du bruit et de la redondance, compliquant ainsi la tâche du modèle pour extraire des informations significatives. À l'inverse, un nombre insuffisant de features peut limiter la capacité du modèle à capturer la complexité des relations entre les variables.

Ce rapport explore différentes configurations de features pour évaluer leur impact sur les performances des modèles de prédiction du pic d'ozone. En ajustant le nombre de features utilisées dans chaque modèle, nous examinons comment ces variations influencent la capacité des modèles à généraliser et à produire des prédictions précises.

Cette analyse approfondie de la sélection des features permet de comprendre comment la complexité des modèles varie en fonction du nombre de caractéristiques utilisées. En identifiant le juste équilibre entre le nombre de features et la capacité prédictive du modèle, cette étude offre des perspectives essentielles pour optimiser la performance des modèles de prédiction du pic d'ozone et, plus généralement, pour la construction de modèles prédictifs dans des contextes environnementaux complexes.

B. Exploration des données :

1. Brève description du jeu de données :

Le jeu de données se concentre sur la prédiction des pics d'ozone journaliers et comprend 1464 observations avec 24 variables. Ces variables englobent des mesures environnementales et météorologiques, telles que la température à différents moments de la journée (T6, T9, T12, T15, T18), la nébulosité (Ne6, Ne9, Ne12, Ne15, Ne18), la direction du vent (Vdir6, Vdir9, Vdir12, Vdir15, Vdir18) et sa vitesse (Vvit6, Vvit9, Vvit12, Vvit15, Vvit18), ainsi que les niveaux maximaux d'ozone observés (maxO3). Parmi ces données, certaines variables sont de type object et nécessitent une conversion en format numérique pour une analyse appropriée. Le jeu de données présente également des valeurs manquantes, avec un total de 175 variables manquantes, nécessitant une attention particulière pour le nettoyage et la préparation des données avant l'application des modèles de machine learning. Ce jeu de données offre une base riche pour l'exploration et la modélisation, permettant une analyse approfondie des facteurs influençant les niveaux d'ozone.

2. Analyse Exploratoire des données (EDA) :

L'analyse exploratoire des données dans notre projet vise à comprendre et à interpréter les caractéristiques du jeu de données sur les pics d'ozone. Cette phase est cruciale pour identifier les tendances, les modèles et les anomalies dans les données, et pour éclairer les étapes ultérieures de modélisation.

a) Distribution des Pics d'Ozone et Autres Variables Clés :

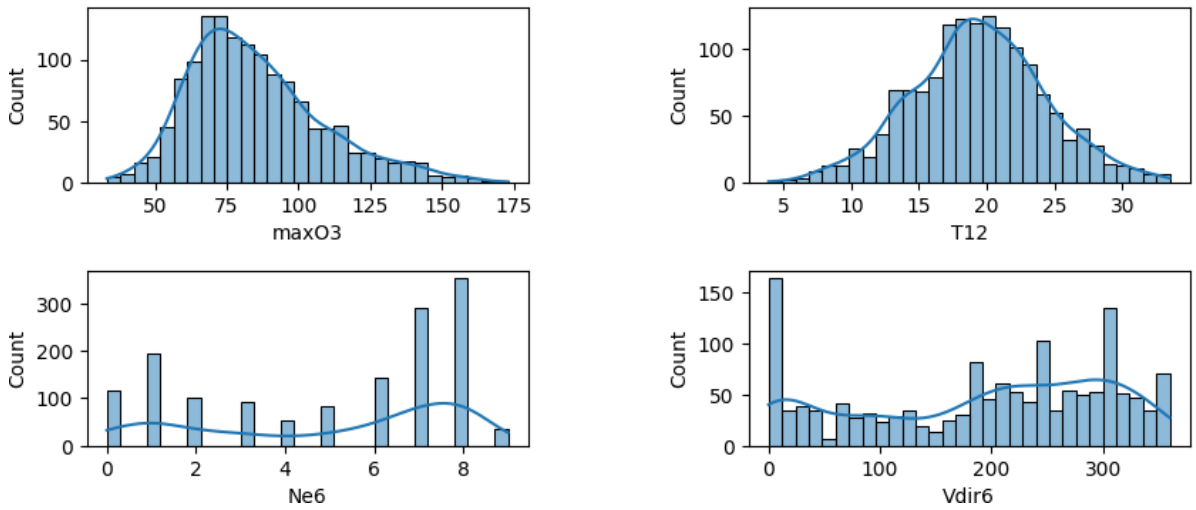

Figure 1 : Distribution des Pics d'ozone et des variables clés

Le premier ensemble de graphiques que nous avons examiné illustre la distribution des pics d'ozone (maxO3), de la température à différents moments de la journée (T12), de la nébulosité observée à 06h (Ne6), et de la direction du vent observée à 6h (Vdir6). Ces histogrammes, réalisés avec Seaborn, montrent que la distribution des pics d'ozone présente une légère asymétrie vers la droite, indiquant une tendance à avoir plus de jours avec des niveaux d'ozone modérés à élevés. La température et les données de vent présentent également des distributions variées, suggérant des relations potentiellement intéressantes avec les niveaux d'ozone.

Nous avons aussi examiné la distribution des variables clés à travers des graphiques de dispersion pour évaluer leurs relations avec les pics d'ozone ('max03'). Ces graphiques sont particulièrement utiles pour visualiser les tendances et détecter des comportements non linéaires ou des anomalies.

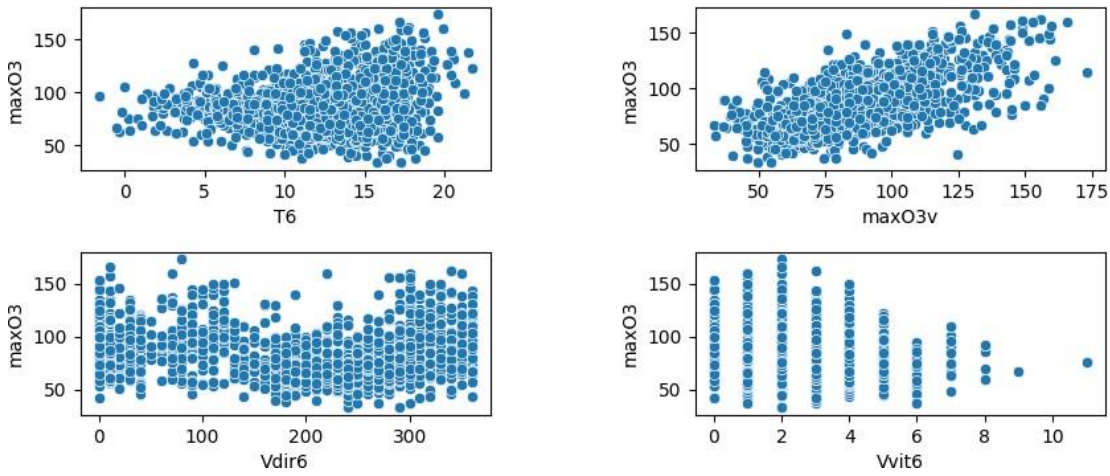

Figure 2 : pics d'ozone en fonctions de variables clés

b) Température à 6h (T6) vs Pics d'Ozone (max03) :

- Le premier graphique montre une dispersion des valeurs de température matinale (T6) par rapport aux niveaux maximaux d'ozone. On observe que les points sont dispersés de manière assez large, ce qui indique qu'il n'y a pas de corrélation linéaire évidente entre ces deux

variables. Cependant, il semble y avoir une concentration de jours avec des niveaux d'ozone plus élevés lorsque les températures du matin sont modérées à élevées.

c) Pics d'Ozone la Veille (max03v) vs Pics d'Ozone (max03) :

- Le deuxième graphique met en évidence la relation entre les niveaux d'ozone observés la veille (maxO3v) et les niveaux actuels d'ozone. Ici, nous pouvons voir une tendance à une corrélation positive ; des niveaux plus élevés d'ozone la veille semblent être associés à des niveaux plus élevés d'ozone le jour actuel. Cette relation suggère que les conditions qui ont conduit à un pic d'ozone élevé pourraient persister d'un jour à l'autre.

d) Direction du Vent à 6h (Vdir6) vs Pics d'Ozone (max03) :

- Dans le troisième graphique, la direction du vent matinale est comparée aux pics d'ozone. Il ne semble pas y avoir de modèle distinct ou de corrélation visible, indiquant que la direction du vent à elle seule n'est pas un prédicteur fiable des niveaux d'ozone.

e) Vitesse du vent à 6h (Vvit6) vs Pics d'Ozone (max03) :

- Le quatrième graphique compare la vitesse du vent matinal avec les pics d'ozone. Les données montrent une certaine concentration de niveaux d'ozone inférieurs pour les vitesses de vent élevées, ce qui pourrait indiquer que des vents plus forts dispersent les polluants, entraînant potentiellement des niveaux d'ozone plus bas.

Ces graphiques de dispersion sont un outil exploratoire fondamental pour identifier les variables qui méritent une attention supplémentaire dans notre modélisation. La température et les niveaux d'ozone précédents sont particulièrement pertinents et seront donc pris en compte dans la sélection des features pour la modélisation. La direction et la vitesse du vent, bien que moins clairement liées aux pics d'ozone dans ces visualisations, pourraient interagir avec d'autres variables de manière à affecter les niveaux d'ozone et mériteront une analyse plus poussée.

Après l'examination des différentes distributions des variables clés à travers des graphiques de dispersion, nous avons utilisé les boxplots pour avoir un aperçu sur la répartition des niveaux de concentration d'ozone (max03), ainsi que d'autres variables météorologiques telles que les températures à différents moments de la journée (T6, T9, T12, T15, T18), la vitesse du vent (Vx) et les niveaux d'ozone de la veille (maxO3v).

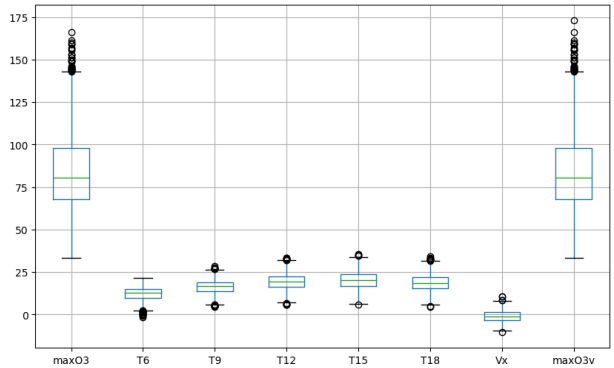

Figure 3 : visualisation à l'aide des boxplots

Le boxplot pour les pics d'ozone (maxO3) montre une médiane inférieure à 100 µg/m³ avec un nombre relativement faible d'outliers, indiquant que la plupart des niveaux de concentration d'ozone sont modérés, mais il y a des jours où ces niveaux dépassent significativement la normale. Ces outliers pourraient indiquer des événements de pollution exceptionnels ou des erreurs de mesure qui nécessitent une vérification supplémentaire.

Les boxplots pour les températures à différents moments de la journée montrent des médianes et des étendues différentes, ce qui suggère que la température varie comme prévu tout au long de la journée, avec des températures plus élevées en milieu de journée (T12, T15) et plus basses le matin et le soir (T6, T18). Il est intéressant de noter que les températures du matin (T6) présentent quelques valeurs aberrantes basses, tandis que les températures de l'après-midi (T15, T18) semblent plus stables avec très peu de valeurs aberrantes.

La variable Vx, qui pourrait représenter la vitesse du vent, montre une médiane près de zéro et quelques valeurs aberrantes. Ces données pourraient indiquer des jours de vent calme avec quelques occurrences de rafales significatives.

Enfin, les niveaux d'ozone de la veille (maxO3v) présentent une répartition similaire à celle des niveaux d'ozone actuels avec une gamme d'outliers similaire, ce qui suggère que des conditions jour après jour peuvent être liées à des concentrations élevées d'ozone.

La présence d'outliers dans plusieurs variables est notable. Les valeurs aberrantes peuvent indiquer des conditions extrêmes réelles qui sont importantes pour la prévision des pics d'ozone, ou elles pourraient être le résultat d'erreurs de mesure. Le traitement de ces outliers est crucial car ils peuvent influencer la performance des modèles de prédiction. Une approche serait de les examiner de plus près pour déterminer s'il faut les conserver, les ajuster ou les supprimer de l'analyse pour éviter de pénaliser la performance des modèles.

Ensuite nous avons dessiné une table de corrélation représentant les coefficients de corrélation entre les diverses variables météorologique et les niveaux de concentration d'ozone dans le jeu de données.

Figure 4 : Table de corrélation

f) Corrélation entre les températures et les niveaux d'ozone :

- Les températures à différents moments de la journée (T6, T9, T12, T15, T18) présentent une forte corrélation positive avec elles-mêmes au fil de la journée, ce qui est attendu puisque les températures ont tendance à augmenter ou à diminuer de manière assez uniforme. En particulier, la température à midi (T12) et celle de l'après-midi (T15) sont fortement corrélées aux niveaux d'ozone (maxO3), indiquant que des températures plus élevées pourraient être associées à des niveaux plus élevés d'ozone.

g) Corrélation entre les niveaux de nébulosité et les niveaux d'ozone :

- Les variables de nébulosité (Ne6, Ne9, Ne12, Ne15, Ne18) présentent des corrélations positives modérées entre elles, ce qui suggère un comportement cohérent dans la couverture nuageuse tout au long de la journée. Cependant, ces variables ne montrent pas de fortes corrélations avec les niveaux d'ozone, ce qui pourrait indiquer que la nébulosité n'est pas un prédicteur dominant des niveaux d'ozone dans ce jeu de données.

h) Corrélation entre la direction et la vitesse du vent et les niveaux d'ozone :

- La direction (Vdir6, Vdir9, Vdir12, Vdir15, Vdir18) et la vitesse du vent (Vvit6, Vvit9, Vvit12, Vvit15, Vvit18) montrent certaines corrélations avec les niveaux d'ozone, mais elles ne sont pas particulièrement fortes. Cela suggère que bien que le vent puisse jouer un rôle dans la dispersion des polluants, d'autres facteurs tels que la température peuvent être plus influents.

i) Corrélation entre les niveaux d'Ozone précédents et actuels :

- Il est intéressant de noter que la variable maxO3v, qui représente les niveaux d'ozone de la veille, est raisonnablement corrélée avec les niveaux actuels d'ozone (maxO3). Cela implique une certaine persistance dans les conditions qui affectent les niveaux d'ozone d'un jour à l'autre.

C. Test des modèles sur différents nombres de covariantes :

1. Choix des features :

Nous avons essayé d'afficher les features selon leur pertinence avec la fonction mutual_info_regression de sklearn.

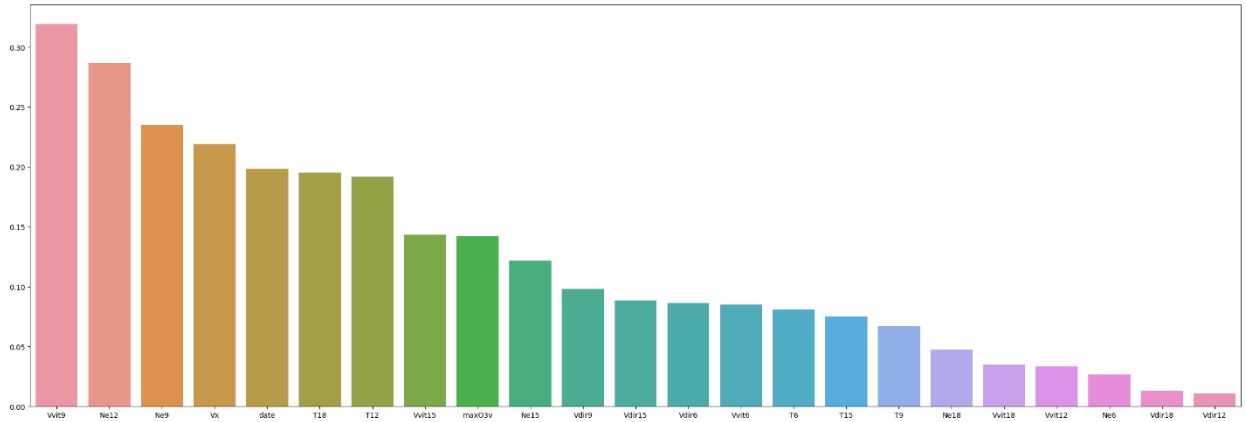

Figure 5 : Histogramme montrant la contribution de chaque covariable

2. Préparation de la data pour l'entrainement :

Afin de tester nos modèles avec un nombre restreint de covariables, la fonction database a été conçue pour sélectionner les K premières variables les plus pertinentes. La variable 'date' ne sera pas intégrée dans notre modèle, car elle ne doit pas être traitée comme une valeur numérique. Pour la répartition de nos données, nous allouerons 30% de celles-ci à l'étape d'entraînement et 70% à celle de test. En vue de normaliser nos données, nous utiliserons le 'scaler' pour garantir une mise à l'échelle adéquate des variables.

3. Création et sélection du meilleur modèle parmi la régression linéaire, random forest, gradient boosting en variant le nombre de covariables choisi dans k= [10, 15, 20, 22] :

Ici, tous les modèles sont soumis à des tests sur différentes bases de données afin de justifier le choix des covariables. Après cette série d'évaluations, les résultats de chaque modèle sont visualisés indépendamment les uns des autres, en utilisant le nombre de covariables retenu. Certains modèles sont également ajustés avec l'ajout de paramètres pour observer leur impact. Nous avons sélectionné un ensemble de 10, 15, 20 et 22 covariables pour ces analyses.
Les résultats obtenus sont les suivants :

```
Nombre features choisis 10

LinearRegression R_MSE: 14.362344547941463 || MAE: 11.275072632568282 || R_squared : 0.7135505378233851
RandomForestRegressor R_MSE: 13.850339498876782 || MAE: 10.570595121951218 || R_squared : 0.7135505378233851
GradientBoostingRegressor R_MSE: 13.290653820259582 || MAE: 9.919332732778317 || R_squared : 0.7135505378233851


Nombre features choisis 15

LinearRegression R_MSE: 14.484348234154059 || MAE: 11.36033604294707 || R_squared : 0.7135505378233851
RandomForestRegressor R_MSE: 13.560604807362518 || MAE: 10.197878048780488 || R_squared : 0.7135505378233851
GradientBoostingRegressor R_MSE: 13.243523875372377 || MAE: 9.923137422939089 || R_squared : 0.7135505378233851


Nombre features choisis 20

LinearRegression R_MSE: 14.005608211168578 || MAE: 10.85458209010294 || R_squared : 0.7135505378233851
RandomForestRegressor R_MSE: 13.490412063713265 || MAE: 10.154999999999998 || R_squared : 0.7135505378233851
GradientBoostingRegressor R_MSE: 13.029401000693364 || MAE: 9.742204131600312 || R_squared : 0.7135505378233851


Nombre features choisis 22

LinearRegression R_MSE: 13.99953285023314 || MAE: 10.834243434807451 || R_squared : 0.7135505378233851
RandomForestRegressor R_MSE: 13.360728935496176 || MAE: 9.999575609756095 || R_squared : 0.7135505378233851
GradientBoostingRegressor R_MSE: 13.017781639218793 || MAE: 9.786365731095206 || R_squared : 0.7135505378233851
```

Figure 6 : comparaison des modèles sur différents nombres de covariables

Parmi ces modèles, le GradientBoostingRegressor avec 22 covariables se démarque comme le plus performant, affichant une racine carrée de l'erreur quadratique moyenne (Root Mean Squared Error) de 13.08, une erreur moyenne absolue de 9,77 et un coefficient de détermination (R_square) de 0.71."

4. Visualisation de la prédiction par rapport à la valeur réelle (y réel) pour les modèles *RandomForestRegressor* et *GradientBoostingRegressor :*

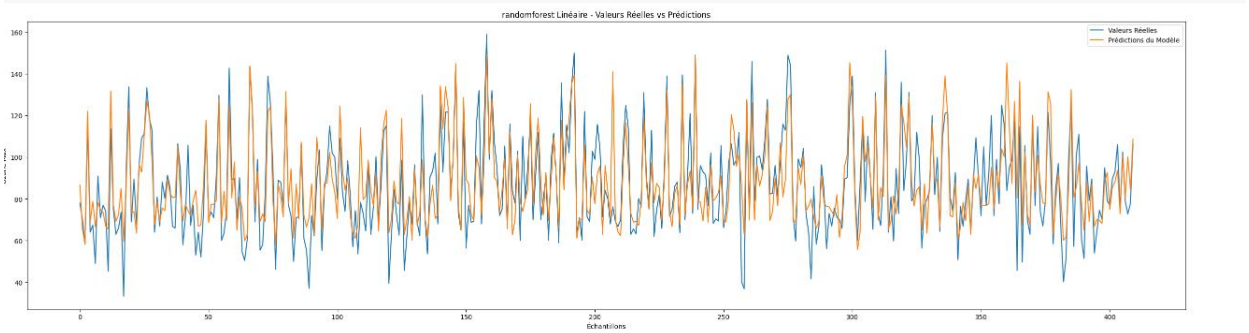

Figure 7 : visualisation des valeurs réelles vs prédites pour Randomforest linéaire

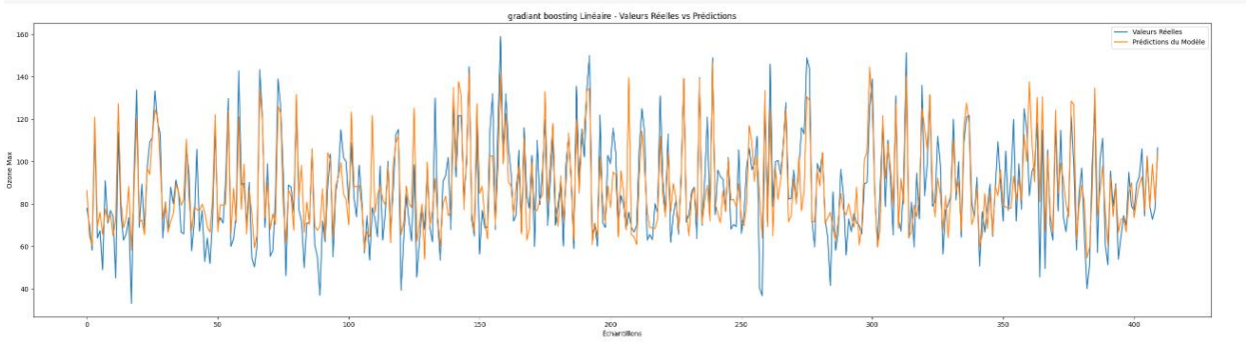

Figure 8 : Visualisation des valeurs réelles vs prédites pour gradient boosting linéaire

5. Implémentation du modèle DNN :

Notre modèle DNN est construit avec deux couches cachées : la première comprenant 32 neurones activés par la fonction ReLU, suivie d'une seconde couche avec 64 neurones. La couche de sortie est composée d'un unique neurone activé de manière linéaire. Pour l'entraînement du modèle, le critère d'erreur retenu est le mean_square_error, optimisé à l'aide de l'algorithme Adam. Les paramètres choisis incluent un batch de 80 et 32 epochs, avec une division de 80% pour l'ensemble d'entraînement (X_train, Y_train) et 20% pour la validation.

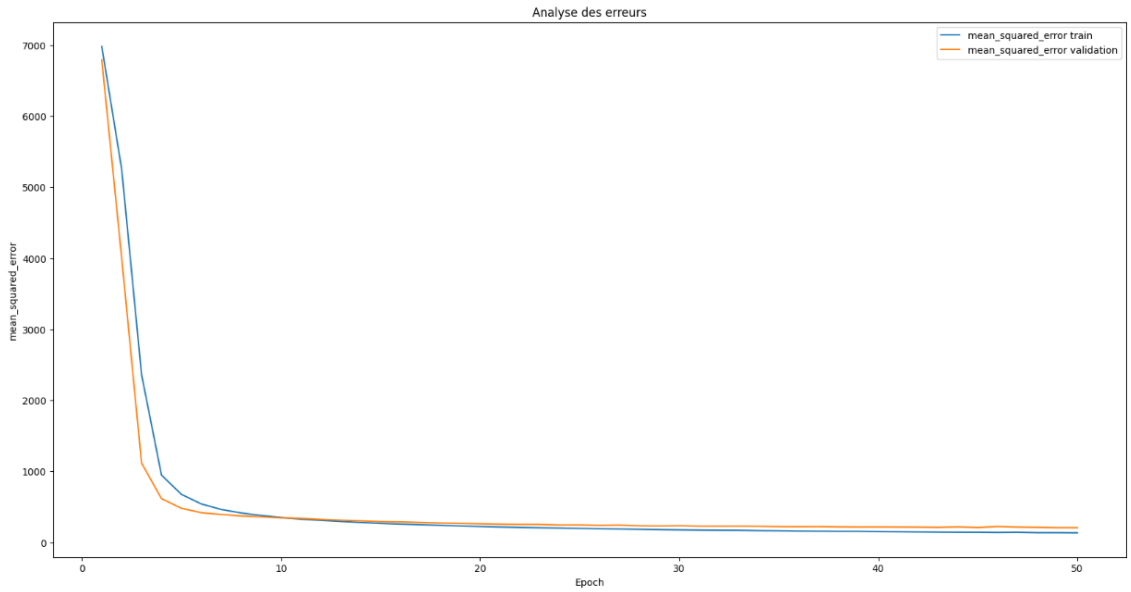

Figure 9 : Analyse des erreurs

La structure du modèle est présentée de manière exhaustive, détaillant chaque couche et le nombre de paramètres utilisés. L'évaluation de ce modèle a abouti à des métriques importantes pour l'évaluation de la performance : un R_Mean Squared Error de 14.63 et un coefficient de détermination (R_square) de 0.63 après l'exécution de 13 epochs. Ces résultats offrent un aperçu de l'efficacité du modèle et serviront de base pour des ajustements futurs dans notre démarche d'optimisation et de sélection du modèle adéquat pour notre analyse.

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 Hidden_layer1 (Dense)       (None, 64)                1472

 Hidden_layer2 (Dense)       (None, 256)               16640

 Output_layer (Dense)        (None, 1)                 257

=================================================================
Total params: 18369 (71.75 KB)
Trainable params: 18369 (71.75 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 10 : Structure du modèle DNN

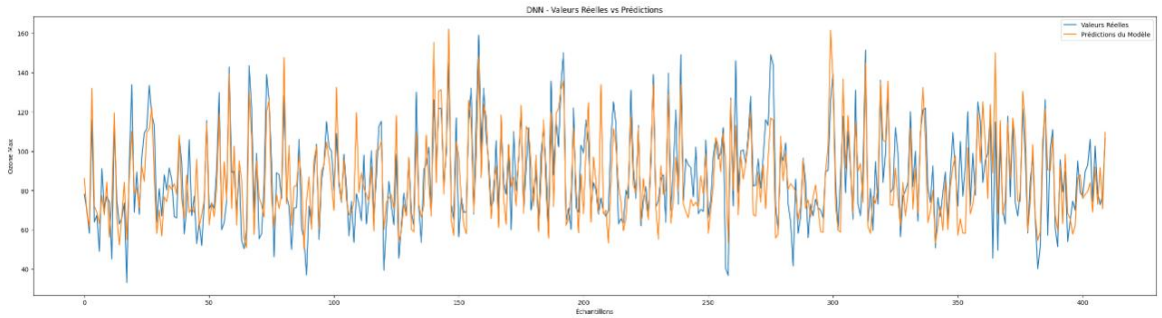

Figure 11 : Valeurs Réelles vs Valeurs prédites

II. Exercice 2 : Jeu de données SPAM :

A. Introduction :

La problématique de la détection de spam demeure une préoccupation majeure dans le domaine de la sécurité informatique. Dans le cadre de ce travail pratique, nous nous sommes attaqués à cette question en explorant divers modèles de machine learning, tels que le Deep Neural Network (DNN), la régression logistique, le Random Forest et le Gradient Boosting. Notre objectif était d'évaluer la performance de ces modèles dans la classification efficace des courriers indésirables. L'organisation de notre travail s'est articulée autour d'une approche systématique, débutant par une analyse exploratoire approfondie des données. Cette phase préliminaire nous a permis de comprendre la nature du jeu de données, d'identifier des tendances et des caractéristiques importantes, ainsi que de mettre en évidence des éventuelles anomalies. Cette démarche a jeté les bases nécessaires pour une modélisation adaptée et a servi de socle à notre prise de décision tout au long du processus. Au cours de cette étude, nous avons implémenté différents modèles de machine learning, chacun avec ses spécificités. Une attention particulière a été accordée à l'optimisation des hyperparamètres de chaque modèle, une tâche délicate, compte tenu de la complexité des algorithmes et de la diversité des paramètres à ajuster.

Ce travail a pour ambition de fournir une compréhension approfondie des performances relatives de ces modèles dans la détection de spam, tout en mettant en lumière les défis inhérents à la recherche des meilleurs paramètres et à la gestion des variables. Cette approche holistique offre des perspectives significatives pour une optimisation continue et l'exploration de nouvelles avenues pour renforcer l'efficacité des systèmes de détection de spam.

B. Analyse exploratoire des données :

1. Exploration du jeu de donnés :

Le jeu de données sur lequel nous avons travaillé est un jeu de données de spam provenant de la source : https://archive.ics.uci.edu/ml/datasets/spambase. Il s'agit d'un jeu de données de comportant des 4601 lignes représentant chacune un mail. Notre dataset est également composé de 58 colonnes. Sur ce total, 57 représentent les variables explicatives. Il s'agit des variables modélisant le mail. Elles sont chacune la fréquence d'occurrence de mots ou de caractères donnés dans les mails. Mais aussi des variables relatives à la présence de majuscules dans les mails. Enfin quant à la variable cible, elle renseigne si le mail est un spam ('1') ou non ('0').

index	word_freq_make	word_freq_address	word_freq_all	word_freq_3d	word_freq_our	word_freq_over	word_freq_remove	word_freq_internet	word_freq_order	word_freq_mail
0	0.0	0.64	0.64	0.0	0.32	0.0	0.0	0.0	0.0	0.0
1	0.21	0.28	0.5	0.0	0.14	0.28	0.21	0.07	0.0	0.94
2	0.06	0.0	0.71	0.0	1.23	0.19	0.19	0.12	0.64	0.25
3	0.0	0.0	0.0	0.0	0.63	0.0	0.31	0.63	0.31	0.63
4	0.0	0.0	0.0	0.0	0.63	0.0	0.31	0.63	0.31	0.63
5	0.0	0.0	0.0	0.0	1.85	0.0	0.0	1.85	0.0	0.0
6	0.0	0.0	0.0	0.0	1.92	0.0	0.0	0.0	0.0	0.64
7	0.0	0.0	0.0	0.0	1.88	0.0	0.0	1.88	0.0	0.0
8	0.15	0.0	0.46	0.0	0.61	0.0	0.3	0.0	0.92	0.76
9	0.06	0.12	0.77	0.0	0.19	0.32	0.38	0.0	0.06	0.0

Figure 12 : Aperçu des données

La figure ci-dessus montre un aperçu des données. Après cette visualisation, nous nous sommes assuré qu'il n'y avait pas de données manquantes dans le jeu de données. Cela s'est fait grâce à la méthode « .info() » de pandas.
Les conclusions sont qu'à l'exception de 3 variables qui elles, sont des variables entières, toutes les autres variables sont des nombres décimaux. Par ailleurs, parmi ces trois variables, figure notre variable cible qui en effet peut être considérées comme une variable booléenne. Par ailleurs, aucune valeur manquante n'est à signaler dans notre jeu de données. Ensuite, dans notre démarche exploratoire, nous nous assurons que certaines variables ne se répètent pas. Cela passe par une analyse de cardinalité. Et cette dernière confirme que les variables ne se répètent pas.

2. Visualisation du jeu de données et statistiques descriptives :

Seules 3 variables sont entières dans notre jeu de données. Et parmi ces 3, seule la variable cible est en effet une variable catégorielle. Notre analyse descriptive des données commencera par cette dernière.

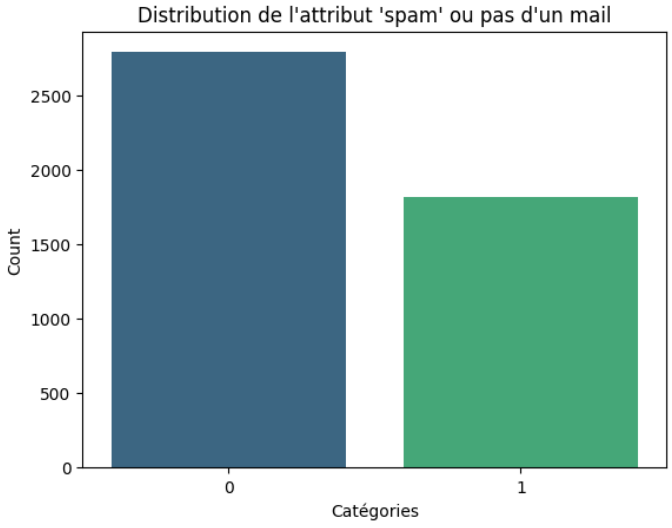

Figure 13 : Distribution des deux attributs 'Spam' et non 'Spam'

La figure ci-dessus reflète la distribution de l'attribut spam ou pas d'un mail. Sur l'ensemble des 4601 mails, notre jeu de données contient 2788 lignes qui ne sont pas des spams (0) et 1813 qui sont spams (1). Comme le montre la figure. Notre base de données n'est pas réellement équilibrée et contient donc

plus d'exemples de mails qui ne sont pas des spams que de mails qui en sont. La classe "0" est majoritaire même si cela n'est pas excessif. Ce déséquilibre des classes dans la base de données peut avoir un impact sur l'entraînement d'un modèle de machine learning. En effet, on peut rencontrer un souci de généralisation. Et cela peut éventuellement expliquer plus tard les performances de nos modèles. Passons à présent à une brève description des variables quantitatives. Elles sont au nombre de 57. Le schéma ci-dessous affiche les statistiques descriptives de ces dernières.

	capital_run_length_average	capital_run_length_longest	capital_run_length_total
count	4601.000000	4601.000000	4601.000000
mean	5.191515	52.172789	283.289285
std	31.729449	194.891310	606.347851
min	1.000000	1.000000	1.000000
25%	1.588000	6.000000	35.000000
50%	2.276000	15.000000	95.000000
75%	3.706000	43.000000	266.000000
max	1102.500000	9989.000000	15841.000000

8 rows × 57 columns

Figure 14 : Statistiques descriptives des variables quantitatives

Dans le cas par exemple de la variable ‘’capital_run_length_total’’ (longueur totale de majuscules), les statistiques décrivent une distribution avec une moyenne d'environ 283. Par rapport à celle-ci, le minimum de 1 indique qu'il y a des mails avec un nombre très court de majuscules. Le maximum de 15841, suggère aussi qu'il existe des mails avec un nombre de majuscules exceptionnellement long. L’écart type de 606 environ justifie donc bien ce constat de variabilité importante dans le nombre de majuscule et une grande dispersion par rapport à la moyenne. De plus, le second quartile (médiane) est de 95. Cette médiane relativement basse par rapport à la moyenne indique une certaine asymétrie dans la distribution, avec quelques mails avec un très grand nombre de majuscules faisant augmenter la moyenne. D’ailleurs 75% des mails ont une longueur de mots en dessous de 266.
Globalement, toutes les variables sont d’ailleurs très déséquilibrées en termes de distribution. En témoignent les schémas ci-dessous.

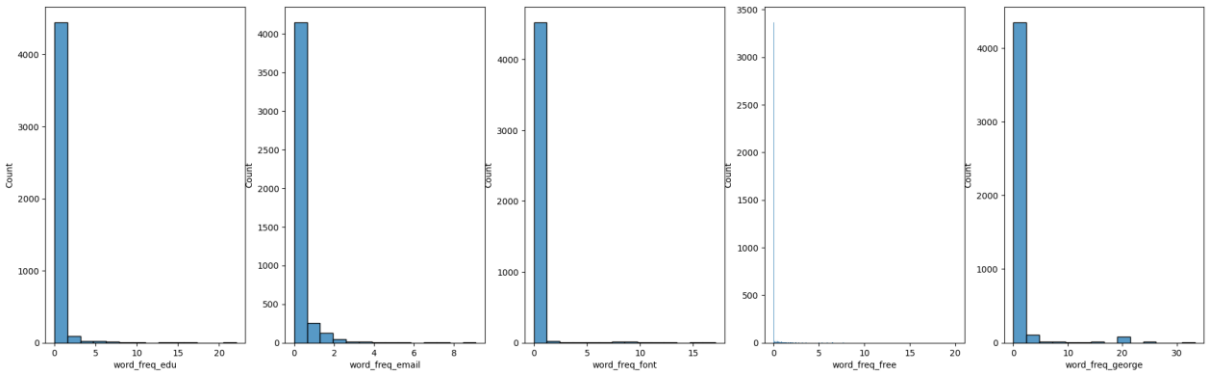

Figure 15 : Exemples de distribution de quelques variables explicatives

La dernière étape de cette section concerne l'analyse de corrélation. En effet, ayant compris le comportement global de chaque variable, nous sommes passés à l'analyse du comportement des variables en relation entre elles et avec la variable cible. Déjà, la grande conclusion de cette section est que les variables ne sont pas grandement corrélées à la variable cible. Aucune d'elle n'atteint en valeur absolue le seuil de corrélation de 0.5 avec la variable cible. La plus forte corrélation en valeur absolue est de 0.38 environs et seules 4 variables dépassent le seuil de 0.3. Cela rendra probablement difficile l'explicabilité des résultats des modèles.

	word_freq_you	word_freq_business	capital_run_length_total	char_freq_dollar	spam
word_freq_you	1.000000	0.084983	-0.007307	0.091470	0.273651
word_freq_business	0.084983	1.000000	0.064261	0.098323	0.263204
capital_run_length_total	-0.007307	0.064261	1.000000	0.201948	0.249164
char_freq_dollar	0.091470	0.098323	0.201948	1.000000	0.323629
spam	0.273651	0.263204	0.249164	0.323629	1.000000

Figure 16 : Matrice de corrélation entre quelques variables et la variable cible

Par ailleurs, certaines variables sont cependant bien corrélées entre elles et 7 paires de variables dépassent une corrélation de 0.7 entre elles en valeur absolue par exemple. Cela introduira par conséquent du bruit dans les modèles qui seront développés. Une approche serait par exemple de supprimer une variable dans chaque paire de variables fortement corrélées afin d'accroitre la précision de nos modèles plus tard. Mais cela dans les tests n'a pas significativement amélioré la prédiction. Ainsi nous garderons toutes les variables dans la section suivante.

C. Modèles de Machine Learning et de Deep Learning :

Après cette phase d'exploration des données, nous passons au déploiement des modèles retenus. Notre problématique ici concerne la prédiction du caractère spam ou pas de mails. Il s'agit d'un problème de classification donc. Le choix du modèle a été fait sur cette base et conformément aux consignes. Dans la suite, nous décrirons leur configuration, leur processus d'entraînement, et les résultats obtenus.

1. Présentation des Modèles et mise en œuvre :

a) Deep Neural Network :

Le DNN est réputé pour sa capacité à capturer des relations complexes dans les données. Nous détaillerons l'architecture du réseau, le nombre de couches, et les fonctions d'activation utilisées. Avant tout, la première étape consiste à diviser le jeu de données en données d'entrainement et de test. Mais avant cela, nous normalisons les données via une normalisation « MinMax ». 80% de ces dernières sera utilisé pour l'entrainement et 20% pour le test. Ainsi, la taille des données d'entrainement est de 3680 et celles des données de test est de 921. Pour un premier usage, nous avons déployé un DNN à 2 couches avec 100 neurones par couche. La démarche ensuite consiste à créer un modèle séquentiel et à convertir les données en un vecteur unidimensionnel. Une fonction d'activation ReLU (Rectified Linear Unit) est utilisée sur les couches cachées pour introduire de la non-linéarité. On ajoute une régularisation en désactivant aléatoirement 20% des neurones de chaque couche pendant l'entraînement afin de prévenir le surajustement. Enfin, la couche de sortie a 1 neurone avec une fonction d'activation sigmoïde. Le choix des fonctions d'activation est typique de notre problème : classification binaire. La fonction de perte correspond à l'entropie binaire, la métrique à suivre est l'« accuracy » » et enfin l'algorithme d'optimisation est celui d'Adam (Adaptive Moment Estimation : combine les idées de l'optimiseur de descente de gradient stochastique (SGD) avec des améliorations telles que l'ajustement adaptatif des taux d'apprentissage pour chaque paramètre du réseau neuronal). Nous chercherons pendant l'entrainement à surveiller la perte sur l'ensemble de validation pendant l'entraînement du modèle. Si cette perte ne s'améliore pas pendant 20 époques consécutives, l'entraînement est arrêté et les poids du modèle sont

restaurés à ceux du meilleur point. Enfin l'entrainement est effectué, sur 50 époques avec des tailles de mini-lots de 368 et le modèle est testé sur les données de test. Les résultats d'entrainement et de test seront présentés plus bas. Après cette première phase, nous procédons à une optimisation pour la recherche des meilleurs paramètres. Nous chercherons à améliorer les résultats obtenus lors du premier déploiement en jouant sur le nombre de neurones par couche, sur le nombre d'époques, sur la taille des mini-lots et sur les paramètres d'activation et de désactivation des neurones par couche. Cette optimisation se fera grâce au module keras-turner. Nous commençons par définir les valeurs possibles pour les hyperparamètres ainsi que la configuration de notre DNN. "RandomSearch" est utilisé pour une recherche aléatoire des hyperparamètres. Dans notre recherche, l'objectif est d'optimiser la précision sur les données de validation et la recherche effectuera au maximum 100 essais.

b) Régression Logistique :

La régression logistique, en tant que modèle linéaire, constitue une référence. La démarche ici est moins complexe de même que dans la suite. La première étape consiste à rendre numérique la variable cible conformément aux exigences de la régression logistique. Cela peut être fait grâce à la classe *LabelEncoder* de scikit-learn. Ensuite, afin de nous assurer que toutes les caractéristiques contribuent de manière égale à un modèle d'apprentissage automatique, nous normalisons leurs valeurs en les ramenant à une plage spécifiée (par défaut, [0, 1]) à l'aide de la mise à l'échelle Min-Max. Nous séparons par suite les données en données d'entrainement puis de test. Et enfin, nous utilisons la bibliothèque scikit-learn en Python pour créer, entraîner et évaluer un modèle de régression logistique. Les paramètres spécifiés sont max_iter qui contrôle le nombre maximal d'itérations pour la convergence de l'optimisation (fixé à 100), et random_state qui fixe la graine pour la reproductibilité des résultats. De même que dans le cas précédent, nous procédons par suite à ce premier apprentissage à une optimisation des paramètres grâce à une recherche aléatoire. Nous définissons un éventail de paramètres parmi lesquels la recherche aléatoire de meilleurs paramètres sera faite. Il s'agit de trois paramètres. Cependant, le jeu de données ne comportant pas de valeurs manquantes et pas de variables explicatives catégorielles, le seul paramètre pertinent est le paramètre c correspondant à l'inverse de la force de régularisation dans la régression logistique. Des valeurs plus élevées de c correspondent à une régularisation plus faible. Cela permet d'évaluer comment la performance du modèle varie en fonction du paramètre de régularisation.

c) Random Forest :

Le Random Forest a été choisi pour sa robustesse et sa capacité à gérer un grand nombre de variables. Le Random Forest est pratique du fait de sa polyvalence, sa capacité à gérer des données complexes, sa robustesse et son aptitude à fournir des performances élevées dans une variété de scénarios. Dans cette section nous décrirons son déploiement sur les données de spam dans le but de faire un meilleur apprentissage des relations entre les variables explicatives et notre variable cible. Une remise à l'échelle n'est pas importante dans ce cas. Nous divisons donc directement en données de test et d'entrainement. Nous importons le modèle et déployons en utilisant 50 arbres pour l'agrégation et en fixant à 2 le nombre d'échantillons minimum pour diviser un nœud sur chaque arbre. Ce modèle est ensuite évalué. Comme dans les autres cas, il fera l'objet d'une optimisation des paramètres. Les deux paramètres que nous optimisons sont ceux cités précédemment. Le nombre de partition pour la croise validation sera fixée à 5. Une comparaison rapide du modèle de random forest sera faite avec une SVM enfin.

d) Gradient Boosting :

Cette technique d'ensemble séquentielle : ensemble de modèles faibles a souvent montré de bonnes performances dans de nombres applications. Cette dernière pourrait nous offrir aussi une adaptabilité aux erreurs et une potentialité de performances supérieures au Random Forest. Comme pour le Random Forest, nous implémentons notre Gradient Boosting depuis scikit learn. La première étape comme souvent fut la division des données et leur normalisation. S'en est suivi l'entrainement avec pour seul paramètre le nombre maximal d'itérations ou d'étapes que l'algorithme effectuera lors de l'entraînement du modèle. Chaque itération correspond à l'ajout d'un nouvel arbre à l'ensemble. Après cet entrainement, une évaluation du modèle est faite et ici nous ne ferons pas d'optimisation de paramètres pour une raison qui sera précisée lors de la présentation des résultats.

2. Présentation des résultats :

Dans la section suivante, nous présenterons les résultats obtenus par chaque modèle lors de l'évaluation, en mettant en évidence les mesures de performance clés. Nous comparerons également les modèles pour identifier celui qui a surperformé dans la détection de spam.

a) Deep Neural Network (DNN) :

À la suite de la paramétrisation et au déploiement du modèle nous pouvons observer la fonction de perte suivante :

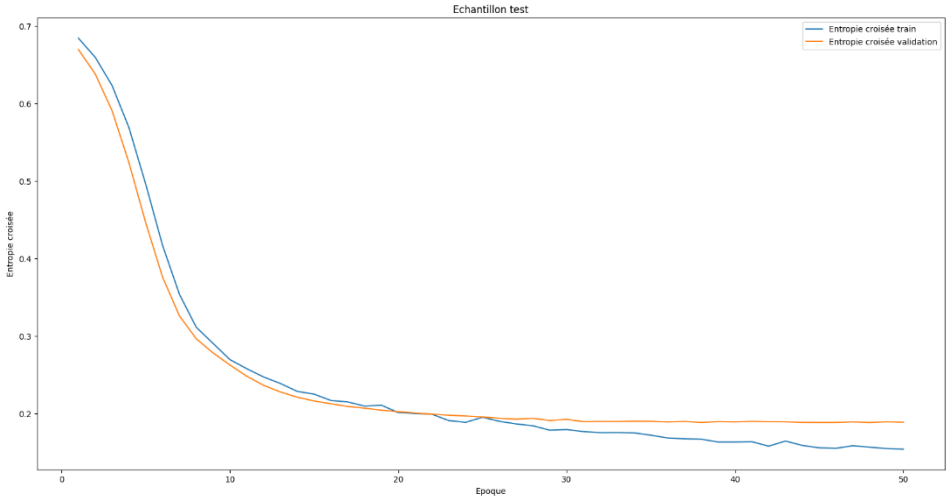

Figure 17 : Évolution précision pendant l'entraînement du DNN

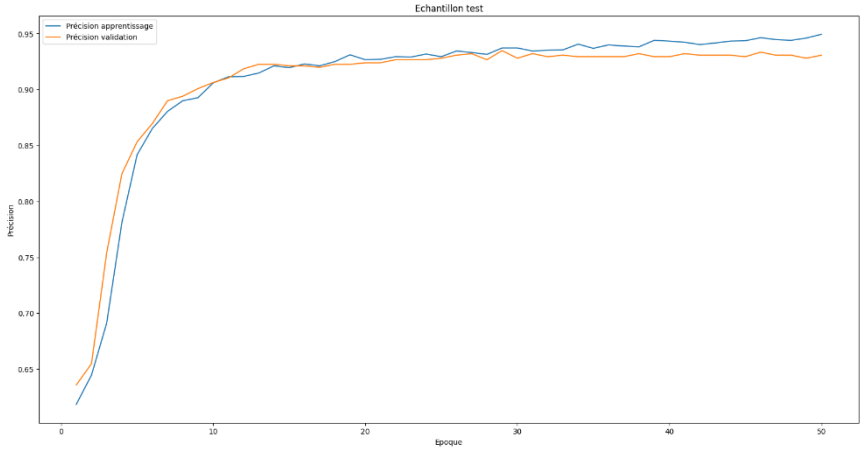

Figure 18 : Fonction de Perte

b) Régression logistique :

Dans le cas de la régression logistique, nous atteignons une performance de 88.71% en accuracy. Il s'agit d'une performance relativement bonne mais sur laquelle nous pouvons chercher plus de détails. Notamment en affichant la matrice de confusion.

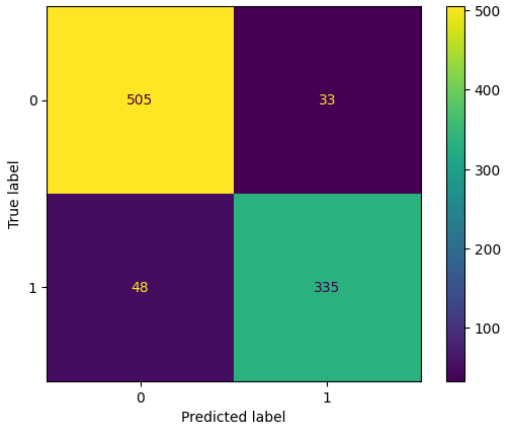

Figure 19 : Matrice de confusion du modèle de la régression linéaire

La précision est de 91%. Autrement 91% des instances prédites comme positives (1) par le modèle sont réellement positives. Cela indique une faible proportion de faux positifs. Les performances pour la classe 0 sont bonnes aussi et même meilleures que celle de la classe 1. Ce qui peut s'expliquer par le déséquilibre en faveur de de la classe 0. Par suite, lors de la recherche de meilleurs paramètres, nous atteignons un score de validation croisée de 93%. Cette performance est obtenue avec l'inverse de la force de régulation valant 100. Un modèle ainsi configuré permet d'obtenir une accuracy de 90.8%. Soit une amélioration de +2.35%.

c) Random Forest :

Les résultats obtenus dès la première utilisation sont remarquables. On atteint presque la même performance en accuracy que le DNN ré-entrainé. Plus exactement, on atteint 94.46%. Il s'agit d'une très bonne performance. Le score de cross validation quant à lui est par contre à 93.57%. Ce qui est aussi remarquable.

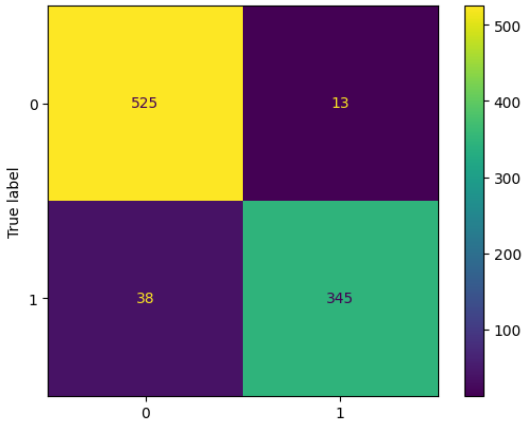

Figure 20 : Matrice de Confusion du modèle de random forest

Le modèle prédit bien environ 96% des instances positives (1). Cela indique une très faible proportion de faux positifs. Il en est de même pour les faux négatifs. Une fois la recherche de meilleurs paramètres effectuée, l'accuracy passe à 94.57%. Une légère amélioration (+0.12%) par rapport aux 50 arbres. Par ailleurs, la courbe ROC du random forest ci-dessous en orange (AUC = 0.99) témoigne de sa très bonne performance. En comparaison avec une SVM dont la courbe ROC est aussi dessiné ci-dessous en bleu (AUC = 0.81), il performe beaucoup mieux. Nous nous abstenons donc d'implémenter en dure forme une SVM.

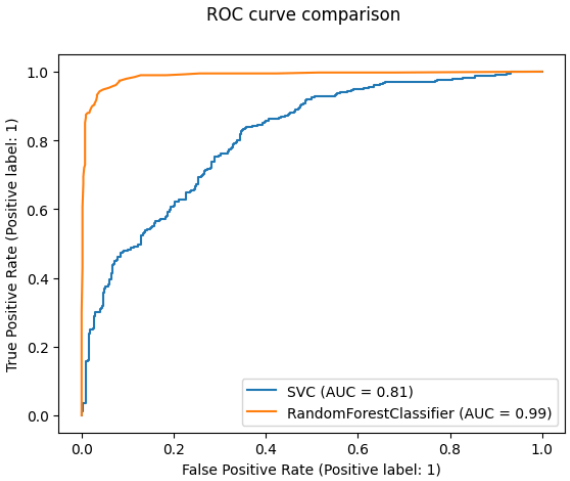

Figure 21 : Schèma comparatif des courbes ROC du random forest et de la SVM

d) Gradient Boosting :

Avec 100 itérations nous atteignons une performance de 96.20% environs en accuracy. Ce qui est bien meilleur que tous les modèles précédents. Aussi nous nous dispenserons d'une recherche de meilleurs paramètres. En effet, sur la base des valeurs choisies et du résultat précédent, il performera mieux que tous les autres. Voyons plus en détail les performances.

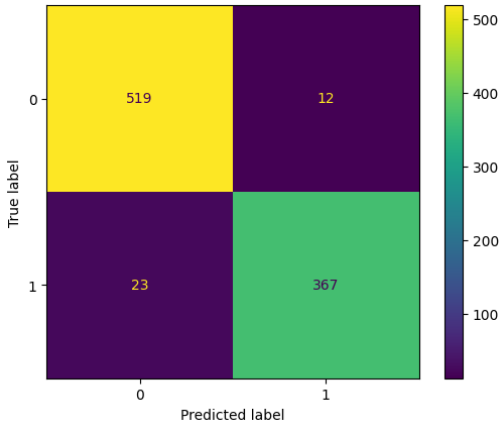

Figure 22 : Matrice de confusion du modèle de gradient boosting

La précision est ici très bonne. Elle est d'environ 97% et cela sans optimisation de paramètre.

3. Discussion et comparaison :

Commençons par remarquer que les performances du modèle de régression logistique sont relativement bonnes. Malgré sa simplicité, le modèle arrive toutefois à atteindre de bonnes performances en apprentissage et surtout généraliser sur les données non connues. Ces performances n'atteignent toutefois pas celles du modèle de deep learning premièrement développé. Pour le modèle de random forest, comparativement au modèle de la régression logistique, nous prédisons mieux les mails spam (une amélioration de +10 sur le même nombre). Il en va de même pour non-spam (amélioration de +20). Cela correspond à un gain de +5.81% en précision et de +3.51% en exactitude environs. Le modèle de gradient boosting montre une amélioration remarquable sur le label "1" (positif) avec moins de faux négatifs (FN) et plus de vrais positifs (TP) par rapport de random forest. Cependant, il montre une légère régression sur le label "0" (négatif) avec moins de vrais négatifs (TN). La progression ou la régression dépend du contexte spécifique de la tâche et des objectifs du modèle. En se plaçant dans un contexte où l'on a plus d'intérêt à ne pas rater les spams, le gradient boosting s'avère plus efficace. Il montre une amélioration d'environ 0.68% en précision par rapport au random forest et une amélioration d'environ 1.9% en exactitude par rapport à ce dernier. Enfin, quant au modèle de deep learning, sur l'ensemble de tous les essais, ses performances sont proches de celles du random forest. Mais lui restent légèrement supérieures. Potentiellement qu'une architecture avec une couche supplémentaire apporterait des améliorations. Il aurait donc fallu essayer plus de configurations. Mais nous n'avons aucune certitude du

fait que ces améliorations soient significatives. La complication liée au fait de trouver la meilleure configuration est à souligner ici car même après une recherche de meilleurs paramètres une exploration hasardeuse nous a fourni de meilleurs résultats. Cela est lié à la quantité de paramètres à optimiser ici qui est beaucoup plus grande que dans les autres cas. Et à un moment, il a fallu se contenter d'une configuration vue que les améliorations n'étaient pas trop grandes. Enfin, la taille des données, ne reconnaissons pas trop grande a probablement à avoir avec le fait que le DNN ne soit pas le meilleur modèle en fin de compte.

D. Conclusion de la section :

Dans le cadre de ce travail pratique sur la détection de spam, nous avons exploré plusieurs modèles de machine learning, notamment un Deep Neural Network (DNN), une régression logistique, un Random Forest et un Gradient Boosting. L'objectif était de comparer les performances de ces modèles et de déterminer celui qui offre la meilleure précision dans la détection des spams. Les résultats obtenus ont révélé que le Gradient Boosting a surperformé les autres modèles, se positionnant en tant que modèle le plus performant pour la détection de spam. Sa capacité à capturer des relations complexes dans les données a contribué à une amélioration significative des performances par rapport aux autres modèles. Le DNN a également démontré des performances robustes, le plaçant en deuxième position. Le DNN et le Random Forest ont montré des performances presque égales et comparables, bien qu'ils soient légèrement inférieurs au Gradient Boosting. Vient en dernier la régression logistique avec ses remarquables 90% environ. Cependant, au-delà des performances des modèles, la recherche des meilleurs paramètres a présenté des défis, en particulier en raison du grand nombre de variables dans le jeu de données. L'optimisation des hyperparamètres pour chaque modèle a demandé du temps et des ressources. La complexité des modèles et le grand nombre de paramètres, en particulier du DNN, a nécessité une recherche minutieuse pour éviter le surajustement et garantir une généralisation efficace. Par ailleurs, la gestion du grand nombre de variables dans le jeu de données a également été un défi. La sélection des caractéristiques aurait pu être une étape cruciale pour améliorer l'efficacité des modèles. Certaines variables peuvent ne pas contribuer de manière significative à la prédiction du spam et auraient pu être éliminées pour simplifier le modèle et accélérer l'entraînement. En conclusion, bien que le Gradient Boosting se soit avéré être le modèle de choix pour la détection de spam dans cette étude, il est important de noter les défis liés à la recherche des meilleurs paramètres et à la gestion des variables. Des étapes supplémentaires telles que la sélection de caractéristiques aurait pu être explorées pour optimiser davantage les modèles. Ce travail fournit des perspectives intéressantes pour des améliorations futures, notamment l'exploration de techniques avancées de Deep Learning et des méthodes plus sophistiquées de gestion des caractéristiques. Mais également la prise en compte du nombre de données pour le choix d'un modèle. En effet, plus de données aurait potentiellement influencé les performances et à notre avis propulser le DNN à la première place.