Université des Sciences et de la Technologie Houari Boumedien
Faculté d'Electronique et d'Informatique
Département d'informatique

Master I Système Informatique Intelligents

Module : Apprentissage Automatique et Réseaux de Neurones

Projet TP

Implémentation d'un détecteur de spam

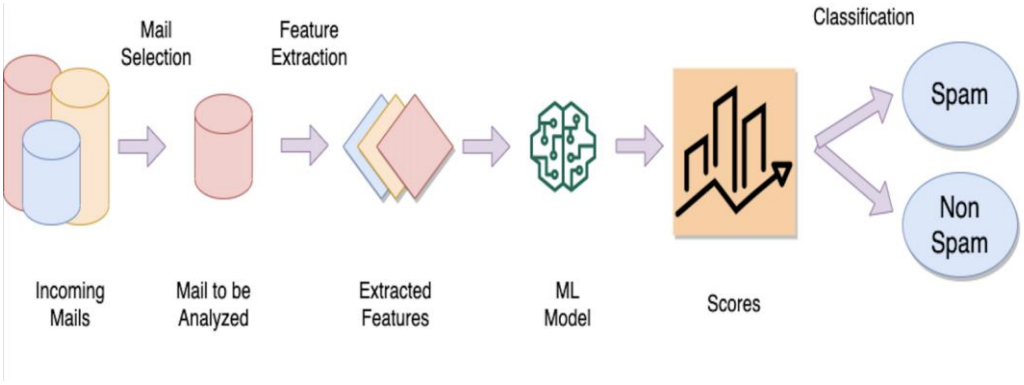
Réalisé par

LAMINI Narimene 181831043933

Année universitaire : 2021-2022

Introduction

Le courrier indésirable (ou spam emails en anglais) est devenu un problème très sérieux. L'envoi de messages inappropriés à un grand nombre de destinataires sans discernement a provoqué la colère des utilisateurs, mais de gros profits pour les spammeurs. Dans ce projet nous allons travailler tout en apprenant des concepts de science des données et d'apprentissage automatique. Le but est d'identifier si un e-mail est un spam ou non spam. Nous commençons en premier lieu par la préparation de notre ensemble de données et ceci en appliquant plusieurs techniques d'analyse du langage naturel NLP, par la suite, nous prendrons l'ensemble de données étiquetés et appliquerons des techniques de classification. Nous pourrons ensuite tester la précision et les performances du modèle sur les e-mails non classifiés.



Approches utilisées

1.1 Libraires

Afin de réaliser ce projet, nous avons implémenté plusieurs libraires décrites dans le tableau ci-dessous avec une justification du choix de chacune.

Libraire	justification
sklearn	Implémentation des différents modèles d'apprentissage automatique comme les svm, logisticRegression,..etc
classification_report	Evaluation des performances telle que la précision, le rappel, le F1 score et le support du modèle entrainé.
loadmat	Téléchargement les données dans un fichier.mat
numpy	Manipulation et utilisation des vecteurs et des matrices
matplotlib.pyplot	Visualisation et les traçages des graphiques
confusion_matrix	Mesurer les performances des modèles utilisés (l'exactitude de leur précision)
plot_confusion_matrix	Traçage de la matrice de confusion
roc_curve, auc	Traçage de la courbe ROC pour chaque classifieur

1.2 Classifieurs

Pour la détection des emails spam, nous avons implémenté les classifieurs qu'on a vus durant ce semestre, à savoir :

- La regression linéaire
- La regression logistique
- Les réseaux de neurones
- Machine à vecteurs de support

- L'algorithme des k moyennes
- L'algorithme des k plus proches voisins
- Bayes naif

Nous avons de plus exploiter d'autres classifieurs qui ne sont pas appris durant ce semsetre, tel que:

- Les arbres de descision
- La foret aléatoire

Le tableau suivant résume tous les classifieurs susmentionnés en précisant l'utilisation de chacun d'eux. Il est à noter qu'on a implémenté les modèles de la libraire **sklearn**.

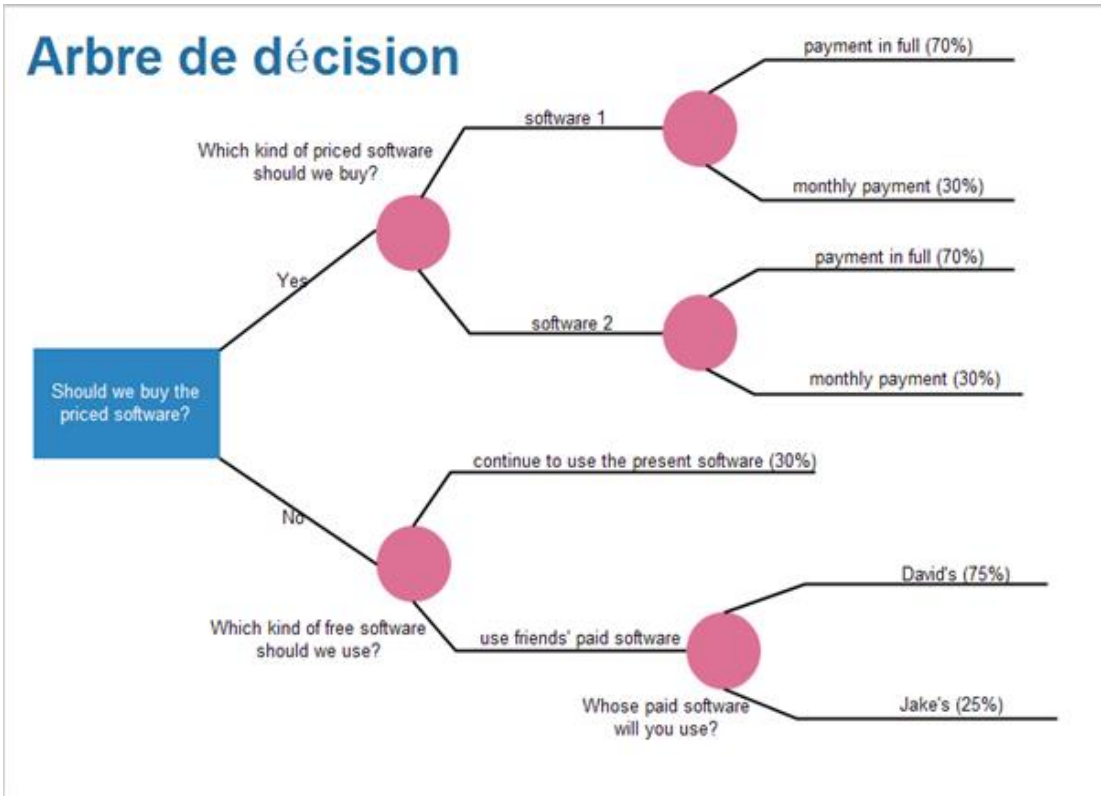
Classifieur	**Utilisation**
linear_model.linearRegression	Prédire la classe des nouvelles données en utilisant la régression linéaire
linear_model.LogisticRegression	Prédire la classe des nouvelles données en utilisant la régression logistique
neural_network.MLPClassifier	Prédire la classe des nouvelles données en utilisant les réseaux de neurones perceptrons
svm	Prédire la classe des nouvelles données en utilisant les machines à vecteur de support
Cluster.KMeans	Prédire la classe des nouvelles données en utilisant l'algorithme des k-moyennes
neighbors.KNeighborsClassifier	Prédire la classe des nouvelles données en utilisant l'algorithme des k-plus proces voisins
naive_bayes.GaussianNB	Prédire la classe des nouvelles données en utilisant l'algorithme de Baies Naif
tree.DecisionTreeClassifier	Prédire la classe des nouvelles données en utilisant les arbres de décision
Ensemble.RandomForestClassifier	Prédire la classe des nouvelles données en utilisant l'algorithme de forêt aléatoire

Nous allons passer en revue une définition des algorithmes non appris durant ce semestre pour bien comprendre le principe de leur fonctionnement

Les arbres de décision (Descision Tree)

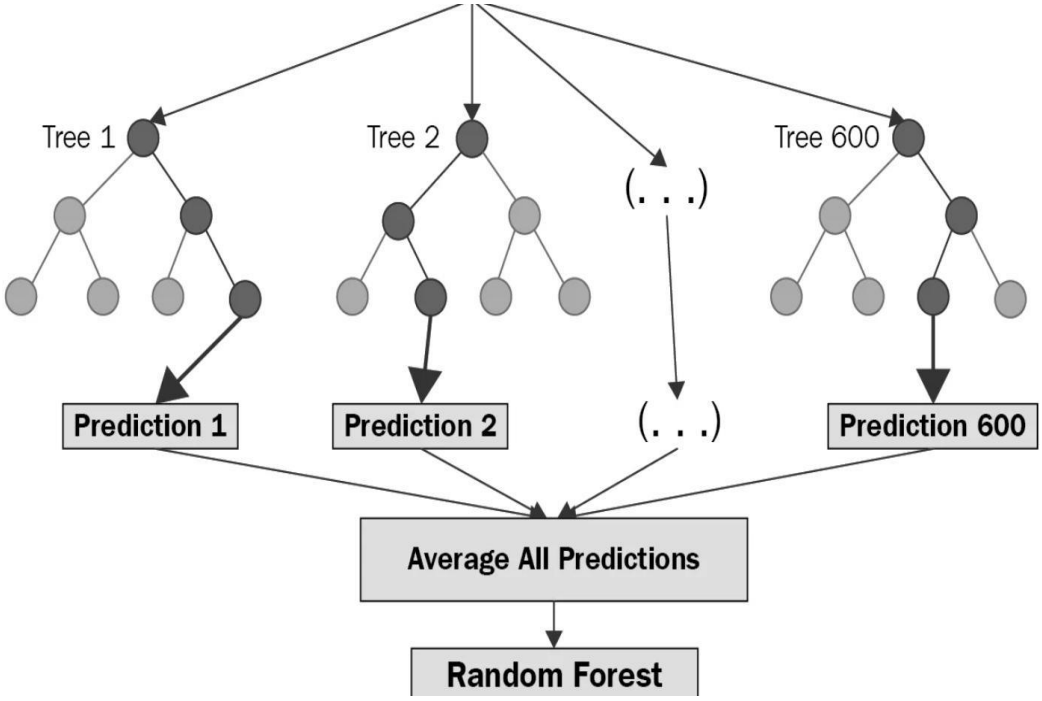
Un arbre de décision est un schéma représentant les résultats possibles d'une série de choix interconnectés. Il permet à une personne ou une organisation d'évaluer différentes actions possibles en fonction de leur coût, leur probabilités et leurs bénéfices. Il peut être utilisé pour alimenter une discussion informelle ou pour générer un algorithme qui détermine le meilleur choix de façon mathématique.

Un arbre de décision commence généralement par un nœud d'où découlent plusieurs résultats possibles. Chacun de ces résultats mène à d'autres nœuds, d'où émanent d'autres possibilités. Le schéma ainsi obtenu rappelle la forme d'un arbre.



La forêt aléatoire (Random Forest)

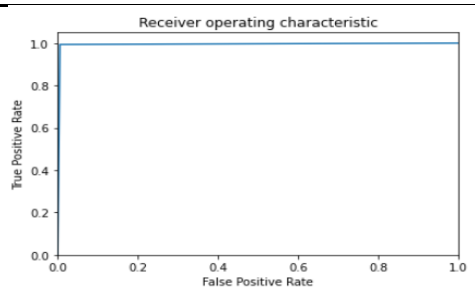
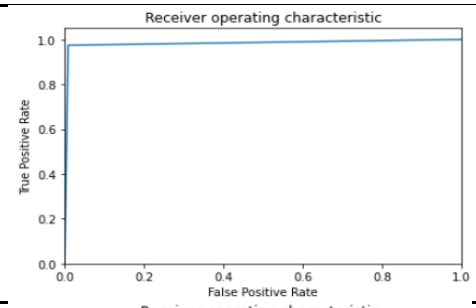
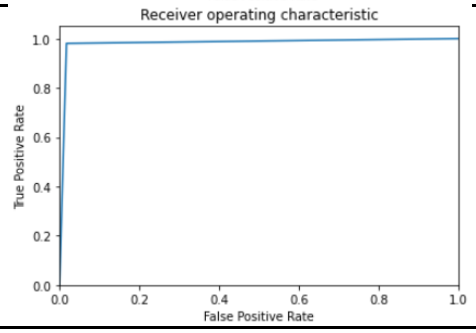
Le random forest ou forêt aléatoire est uilisé en machine learning, il est composé de plusieurs arbres de décision, entrainés de manière indépendante sur des sous-ensembles du data set d'apprentissage (méthode de bagging). Chacun produit une estimation, et c'est la combinaison des résultats qui va donner la prédiction finale qui se traduit par une variance réduite. En somme, il s'agit de s'inspirer de différents avis, traitant un même problème, pour mieux l'appréhender. Chaque modèle est distribué de façon aléatoire en sous-ensembles d'arbres décisionnels.

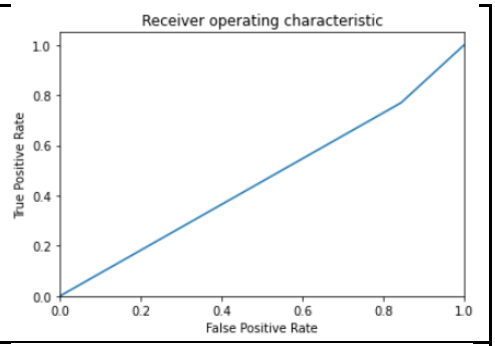
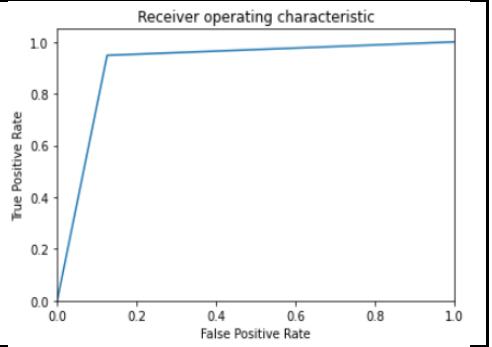
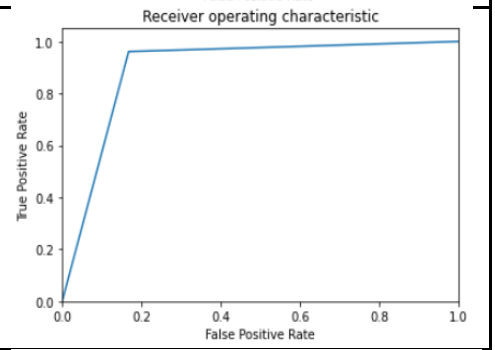
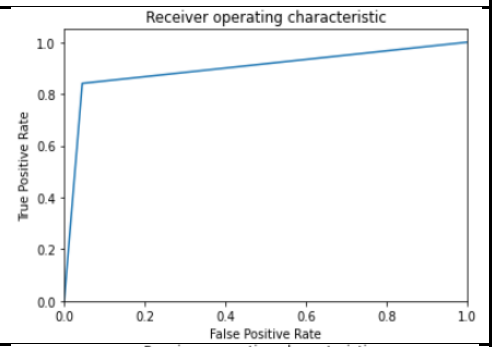
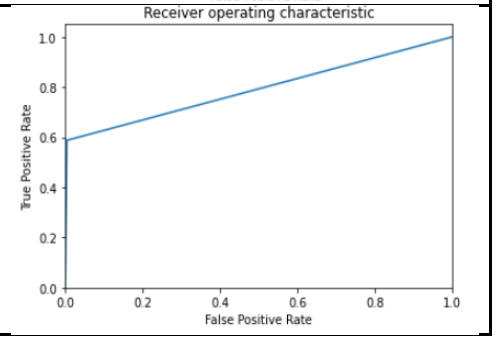


Synthèse des résultats

Après avoir testé tous les modèles décrits ci-dessous sur l'ensemble des emails, nous allons présenter un tableau récapitulatif des différents résultats obtenus suivi d'une analyse et comparaison de leurs qualités et leur efficacité de prédiction.

- L'ensemble d'entrainement utilisé contient 4000 exemples
- L'ensemble d'entrainement utilisé contient 1000 exemples

Modèle	Accracy	TF	TN	FP	FN	Courbe ROC
linearRegression	0%	-	-	-	-	-
LogisticRegression	99.4 %	688	306	4	2	
MLPClassifier	98.6 %	686	300	6	8	
svm	98.2 %	680	302	12	6	

KMeans	34.5 %	-	-	-	-	
KNeighborsClassifier	89.7 %	605	292	87	16	
GaussianNB	87.2 %	576	296	116	12	
DecisionTreeClassifier	92.0 %	661	259	31	49	
RandomForestClassifier	87.0 %	689	181	3	127	

Analyse et comparaison des résultats

En analysant les résultats de test des différents modèles, on remarque que :

- L'algorithme de régression linéaire n'a pas réussi à classifier les nouvelles données, il a pour précision 0%, et ceci revient au fait que cet algorithme est dédié aux problèmes continus et non discret.
- L'algorithme des k-moyenne a marqué une faible précision de 34%, étant donné que c'est un algorithme d'apprentissage non supervisé, traitant des données non libellées (unlabeled data), et ce n'est pas le cas pour l'ensemble de données qu'on a traité dans ce projet.
- Pour les algorithmes **KNeighborsClassifier, GaussianNB, DecisionTreeClassifier et RandomForestClassifier**, la précision est jugée acceptable (entre 87% et 92%) avec un taux de FP estimé entre 3 et 116 et FN entre 16 et 127.
- Les algorithmes **LogisticRegression, MLPClassifier et svm** sont jujés meilleurs avec une excellente précision entre 98,2% et 99.4% et un taux de FP et FN très bas (minimum 2, maximum 12).

Conclusion

La prédiction est en général un problème difficile et il est naturel de se focaliser sur la précision, ou le taux d'erreur. Ceci explique les nombreux travaux en machine learning qui visent à développer des classifieurs maximisant cette métrique.