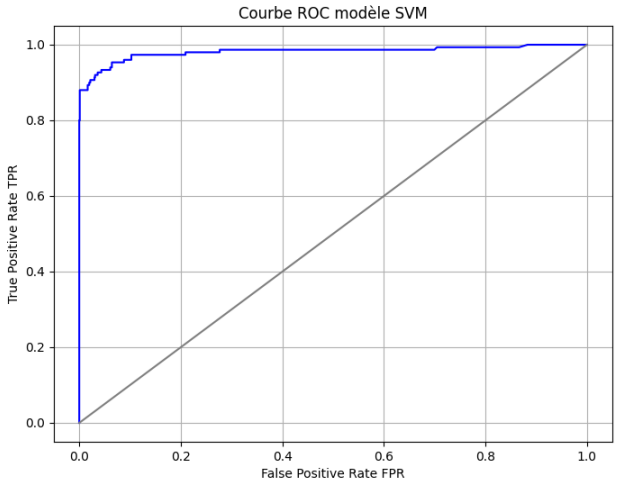
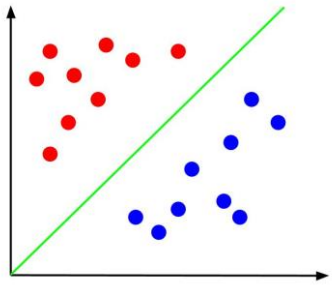
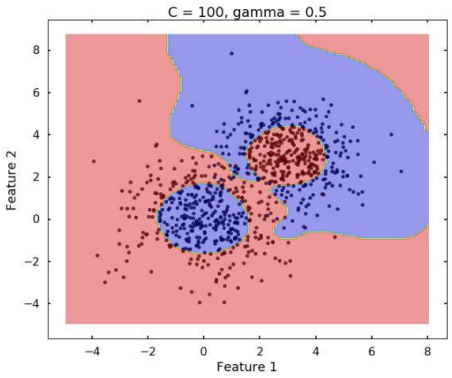
Rapport TP 5 ATDN

Dara PAK Master 1 OIVM

Exercice 1 :







SVM Noyau RBF

Matrice de confusion :

		Réalité	
		Négative : 0	Positive : 1
Prédiction	Négative : 0 (ham)	964(VN)	1(FN)
	Positive : 1 (spam)	25(FP)	125(VP)

Rapport de classification:

	Metric precision	recall	f1-score	support
0	0.97	1	1	965
1	0.99	0.85	0.91	150
Accuracy			0.98	1115
Macro avg	0.98	0.92	0.95	1115
weighted avg	0.98	0.98	0.98	1115

SVM Lineaire

Matrice de confusion :

		Réalité	
		Négative : 0	Positive : 1
Prédiction	Négative : 0	960(VN)	5(FN)
	Positive : 1	19(FP)	131(VP)

Rapport de classification

	Metric precision	recall	f1-score	support
0 (ham)	0.98	0.99	0.99	965
1 (spam)	0.96	0.87	0.92	150
Accuracy			0.98	1115
Macro avg	0.97	0.93	0.95	1115
weighted avg	0.98	0.98	0.98	1115

Question 2 :

Le choix du ratio 70/30 (standard) permet de maintenir un équilibre entre les données disponibles pour l'entrainement (70%) et le test (30%). Elle offre en général un bon compromis

- Un ratio trop bas pour les données d'entrainement le modèle risque d'être sous entrainé
- Un Ratio trop bas pour les données de test les performances risque d'être faussé.

Question 4 :

Rapport de classification SVM RBF :

- **Précision :** On peut voir que le score de précision de ham et spam sont très bon ce qui veut dire qu'il y a très peu de faux positif.
- **Recall :** Le modèle capte parfaitement la totalité des ham, cependant elle laisse passer quelques spam.
- **F1-score :** le score est assez équilibré, ce qui signifie qu'il y a un bon équilibre entre le score de précision et de recall

Comparaison avec un SVM Linéaire :

- **Précision :** On peut observer que le SVM RBF est nettement meilleur en termes de précision (SVM RBF 0.99 vs 0.98 SVM Lineaire)
- **Recall :** Concernant le recall on observe également un avantage au SVM RBF
- **F1-score :** Enfin le f1-score résumer en effet bien les conclusions faites au différentes metric. Le score-f1 du SVM RBF est légèrement meilleur que celle du SVM linéaire. (0.91 vs 0.90)

Conclusion : Le SVM avec noyau RBF semble le plus adapté ici lorsque l'on a des données non linéaires à l'inverse du SVM linéaire qui lui est le plus adapté lors de datasheet plus linéaire.

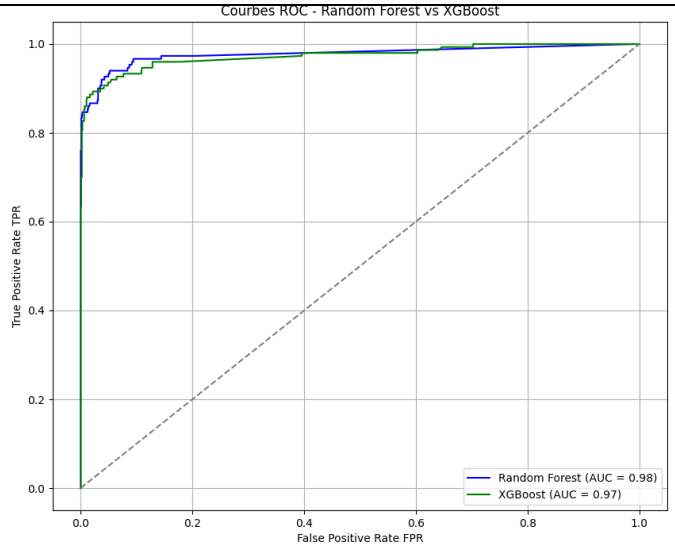
Question 5 :

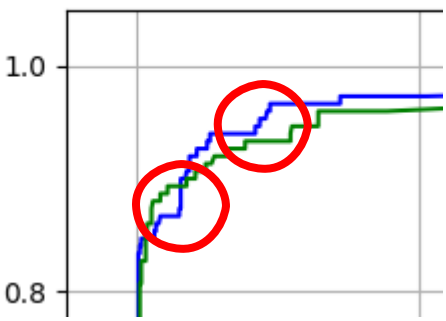
Matrice de confusion :

- **VN :** le modèle a correctement détecté 964 'ham' (0) ce qui est très bon vu le nombre de ham réels !
- **FN :** le modèle à seulement eu 1 erreur lors de la détection d'un ham le classant en tant que spam.
- **VP :** le modèle a correctement détecté 125 'spam' (1)
- **FP :** Dans le cas de la détection des spam le modèle a un peu plus de mal et 25 spam n'ont pas été détectés.

Conclusion : On peut dire que le modèle est très performant pour la détection de ham, mais rencontre quelques difficultés pour la détection de spam.

Exercice 2 :





XGBoost :

Matrice de confusion :

		Réalité	
		Négative : 0	Positive : 1
Prédiction	Négative : 0	959	6
	Positive : 1	23	127

Rapport de classification:

	Metric precision	recall	f1-score	support
0	0.98	0.99	0.99	965
1	0.95	0.85	0.90	150
Accuracy			0.97	1115
Macro avg	0.97	0.92	0.94	1115
weighted avg	0.97	0.97	0.97	1115

Random Forest :

Matrice de confusion :

		Réalité	
		Négative : 0	Positive : 1
Prédiction	Négative : 0	965	0
	Positive : 1	97	53

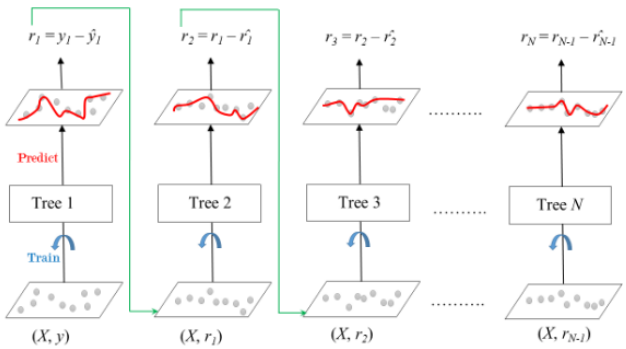
Rapport de classification:

	Metric precision	recall	f1-score	support
0	0.91	1	0.95	965
1	1	0.35	0.52	150
Accuracy			0.91	1115
Macro avg	0.95	0.68	0.74	1115
weighted avg	0.92	0.91	0.89	1115

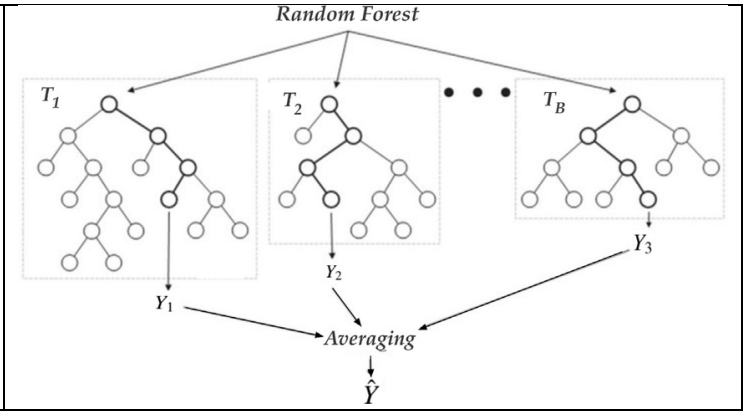
Question 1 :

Les hyperparamètres les plus importantes à ajuster dans notre modèle Random Forest sont :

- **Le n_estimators :** C'est le nombre d'arbre dans le foret, cependant un trop grand nombre d'arbre augmente le temps de calcul
- **Max_depth :** Elle limite la profondeur des arbres afin d'éviter le surapprentissage



Gradient Boosting



Question 2 :

- La complexité du Gradient boosting est plus complexe, car elle reconstruit de manière séquentielle les arbres en corrigeant les erreurs précédentes.
- La complexité du random forest est moindre car chaque arbre est indépendant.

Question 3 :

Comparaison Gradient Boosting & Random Forest :

- Précision :
  Gradient Boosting : On observe que la précision est plutôt équilibrée entre les spam et ham (il n'y pas trop de disparité selon la classe)
  Random Forest : On peut voir que le score de précision est plutôt bon avec un score de 100 pour les spam
- Recall :
  Gradient Boosting : A la différence du score de précision on observe ici que le score de recall n'est pas très équilibré. En effet on peut observer un plus grand nombre de recall pour les spam.
  Random Forest : Malgré le très bon score de précision pour les spam. Le score de recall est catastrophique c'est-à-dire que plus de 75% des spams ne sont pas bien détecté. En comparaison des ham qui eut ont un bon score.
- F1-score :
  Gradient Boosting : Le score f1 résume bien ce que nous avons observé précédemment avec un score plutôt équilibré
  Random Forest : Le score f1 des spam est très mauvais reflétant un mauvais équilibre lors de l'apprentissage du modèle.

Question 4 :

Sur notre courbe ROC-AUC on peut observer :

- **Le Random Forest** est généralement plus performante. Cependant on peut observer une certaine non-homogénéité de la courbe, ce qui peut refléter des difficultés lors de l'apprentissage du modèle sur nos datasets.
- **Le Gradient Boosting** est généralement plus homogène est plus stable. Même si elle peut être moins performante par moment.

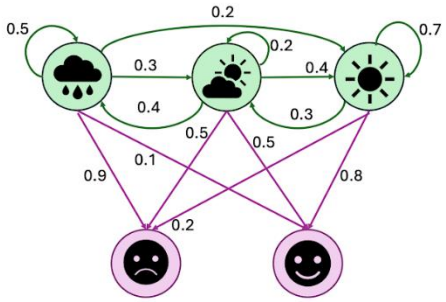
Question 5 :

Avantage modèles d'ensemble : (Random Forest et Gradient Boosting)

- **Auto-Correction :**
  - **Le Gradient Boosting** s'auto corrige après chaque itération.
  - **Le Random Forest** diminue le surapprentissage
- **Robustesse :** Le système de tree permet d'avoir un modèle plus robuste, donc moins sensible au déséquilibre de classes.

Conclusion : Pour une analyse rapide on choisira le random forest. Pour une meilleure précision on choisira le Gradient boosting. Cependant dans notre cas le random forest à des difficultés avec la classe mineure dans notre datasheet, le modèle privilégiera la détection de la classe majeur.

Exercice 3 :



HMM :

Matrice de confusion :

		Réalité	
		Négative : 0	Positive : 1
Prédiction	Négative : 0	2219	192
	Positive : 1	74	301

Rapport de classification:

	Metric precision	recall	f1-score	support
0	0.97	0.92	0.94	2411
1	0.61	0.80	0.69	375
Accuracy			0.90	2786
Macro avg	0.79	0.86	0.82	2786
weighted avg	0.92	0.90	0.91	2786

Question 1 :

Les modèles HMM sont des modèles statistiques qui modélisent des séquences ou des états inobservables qui influencent le visible. Exemple dans notre cas la séquence cachés qui influence le visible du non visible sont les spam et ham.

Question 4 :

- **Précision :** On peut voir que le score de précision n'est pas très homogène on observe de grande disparité entre les spams et hams.
- **Recall :** Le score de recall reflète bien le score de précision. Elle montre que le modèle a un recall de spam de 80% c'est-à-dire qu'elle classe beaucoup de spam étant comme ham.
- **F1-score :** Le score f1 résume bien ce que nous avons observé précédemment avec notamment des difficultés lors de la détection de spam

Question 5 :

Pour ce type de données le modèle HMM à de grande difficulté lorsque les classes ne sont pas équilibré, notamment dans notre cas où nous avons plus de ham que de spam.

Exercice 4 :

Question 1 :

Avec la bibliothèque SHAP, on peut interpréter comment les décisions d'un modèle sont influencées par différentes caractéristiques.

Question 2 :

Les mots les plus influents pour prédire un spam ou ham sont :
['____' 'aah' 'aathi' ... 'ïï' 'û_' 'ûò']

Question 4 :

Pour des modèles complexes cela peut être long et couteux. Pour l'analyse et l'interprétation elle peut être difficile. Par exemple dans notre cas nous avons des mots assez étranges pour la prédiction de la classe ham et spam.

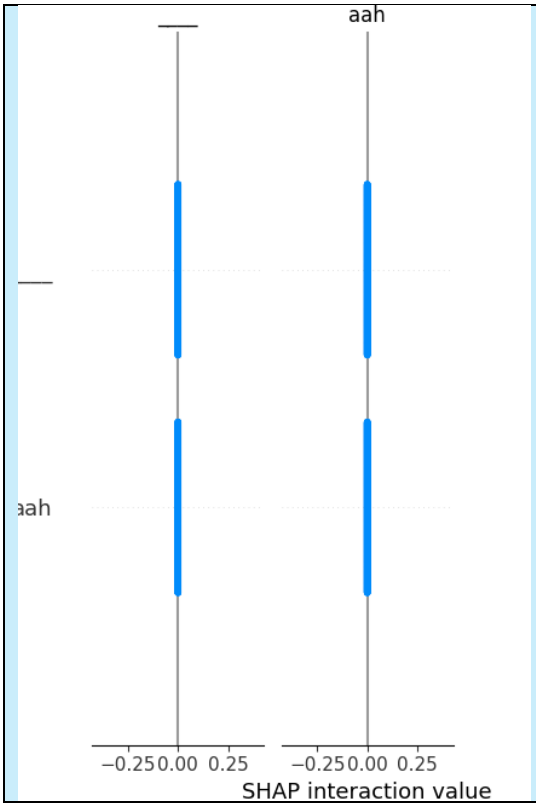
SHAP avec Random Forest :

Matrice de confusion :

		Réalité	
		Négative : 0	Positive : 1
Prédiction	Négative : 0	962	3
	Positive : 1	25	125

Rapport de classification:

	Metric precision	recall	f1-score	support
0	0.97	1	0.99	965
1	0.98	0.83	0.90	150
Accuracy			0.97	1115
Macro avg	0.98	0.92	0.94	1115
weighted avg	0.97	0.97	0.97	1115



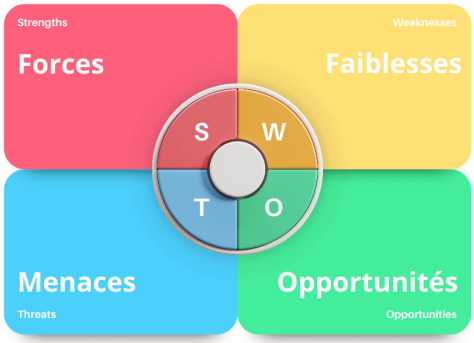
Exercice 5:

Question 1:

SVM RBF : • **Précision :** On peut voir que le score de précision de ham et spam sont très bon ce qui veut dire qu'il y a très peu de faux positif. • **Recall :** Le modèle capte parfaitement la totalité des ham, cependant elle laisse passer quelques spam. • **F1-score :** le score est assez équilibré, ce qui signifie qu'il y a un bon équilibre entre le score de précision et de recall.	**SVM Linéaire :** • **Précision :** On peut observer que le SVM RBF est nettement meilleur en termes de précision • **Recall :** Concernant le recall on observe également un avantage au SVM RBF • **F1-score :** Enfin le f1-score résumer en effet bien les conclusions faites au différentes metric. Le score-f1 du SVM RBF est légèrement meilleur que celle du SVM linéaire.

<table>
<tr>
<td>Gradient Boosting :
Précision : On observe que la précision est plutôt équilibrée entre les spam et ham (il n’y pas trop de disparité selon la classe)
Recall : A la différence du score de précision on observe ici que le score de recall n’est pas très équilibré. En effet on peut observer un plus grand nombre de recall pour les spam.
F1-score : Le score f1 des spam est très mauvais reflétant un mauvais équilibre lors de l’apprentissage du modèle.</td>
<td>Random Forest :
Précision : On peut voir que le score de précision est plutôt bon avec un score de 100 pour les spam
Recall : Malgré le très bon score de précision pour les spam. Le score de recall est catastrophique c’est-à-dire que plus de 75% des spams ne sont pas bien détecté. En comparaison des ham qui eut ont un bon score.
F1-score : Le score f1 résume bien ce que nous avons observé précédemment avec un score plutôt équilibré</td>
</tr>
<tr>
<td>HMM
Précision : On peut voir que le score de précision n’est pas très homogène on observe de grande disparité entre les spams et hams.
Recall : Le score de recall reflète bien le score de précision. Elle montre que le modèle a un recall de spam de 80% c’est-à-dire qu’elle classe beaucoup de spam étant comme ham.
F1-score : Le score f1 résume bien ce que nous avons observé précédemment avec notamment des difficultés lors de la détection de spam</td>
<td></td>
</tr>
<tr>
<td>Temps entrainement :
SVM RBF : 53.97 secondes
XGBoost : 0.68 secondes
Random Forest : 1.21 secondes
HMM : 0.78 secondes
SHAP : 7.19secondes</td>
<td>Temps de prédiction :
SVM RBF : 2.8 secondes
XGBoost : 0.019 secondes
Random Forest : 0.018 secondes
HMM : 0.02 secondes
SHAP : 0.04 secondes</td>
</tr>
</table>

Discussion



SVM RBF : **Force :** • Précision et rappel élevés, en particulier pour la classe "ham". **Faiblesses** : • Temps d'entraînement long de 55,87 secondes par rapport aux autres **Opportunités** : • Convient aux datasheets avec des séparations de classes non linéaires. **Menaces** : • Coût pouvant être très élever sur des ensembles de données plus grands.	**Gradient Boosting :** **Force :** • Performance équilibrée pour les spams et les hams. **Faiblesses** : • Précision légèrement inférieure comparée aux autres modèles. **Opportunités** : • Performances améliorables avec un réglage plus précis. • **Menaces** : • Risque de surapprentissage si mal paramétré.
Random Forest : **Force :** • Temps d'entraînement et de prédiction rapides. **Faiblesses** : • Performances diminuées sur des ensembles de données déséquilibrés. **Menaces** : • Problèmes sur les ensembles de données déséquilibrés sans prétraitement.	**HMM:** **Force :** • Bon rappel pour les messages hams. **Faiblesses** : • Difficultés avec la détection des spams. **Menaces** : Pas performant pour des ensembles de données déséquilibrés comme les spams et ham