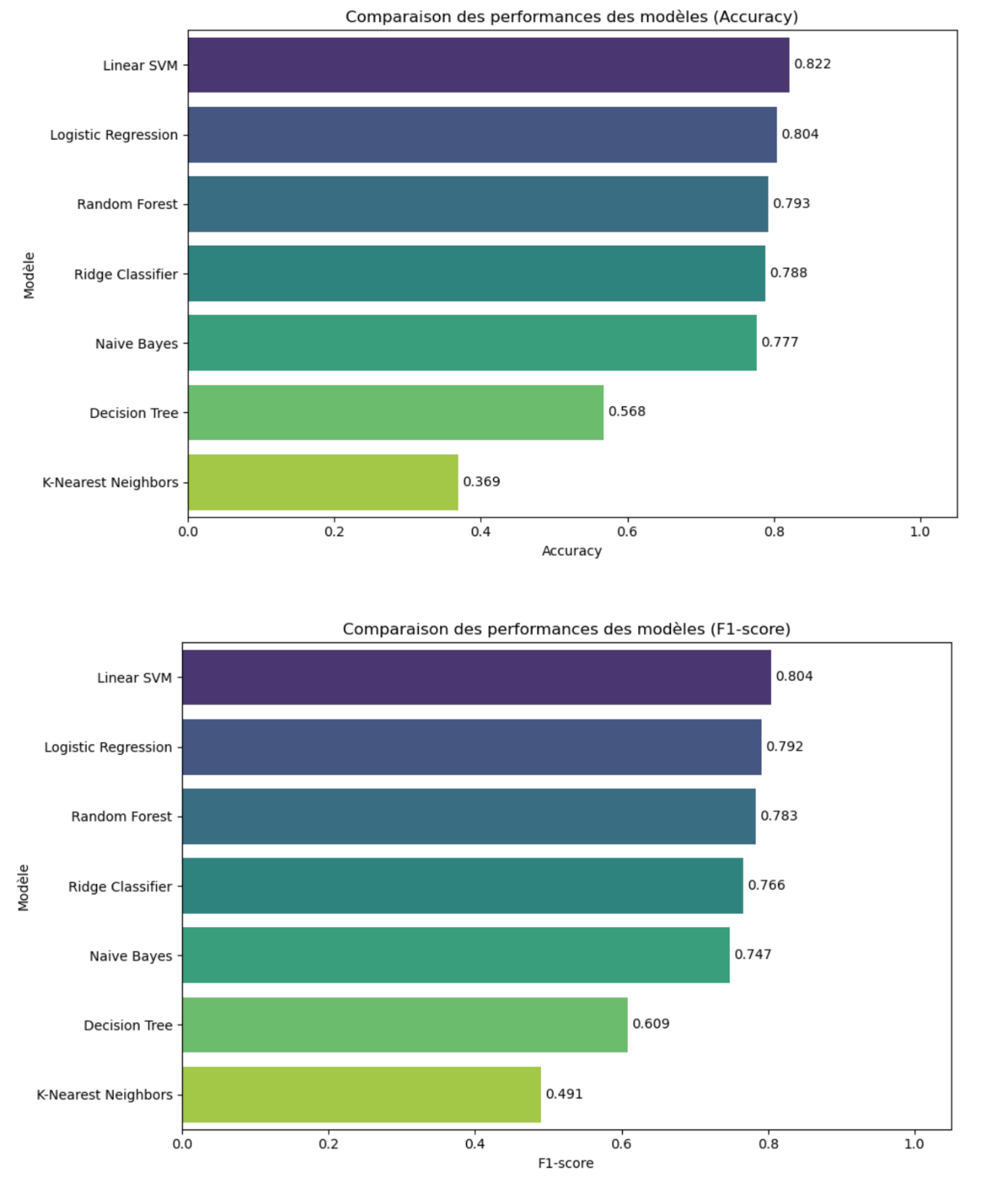
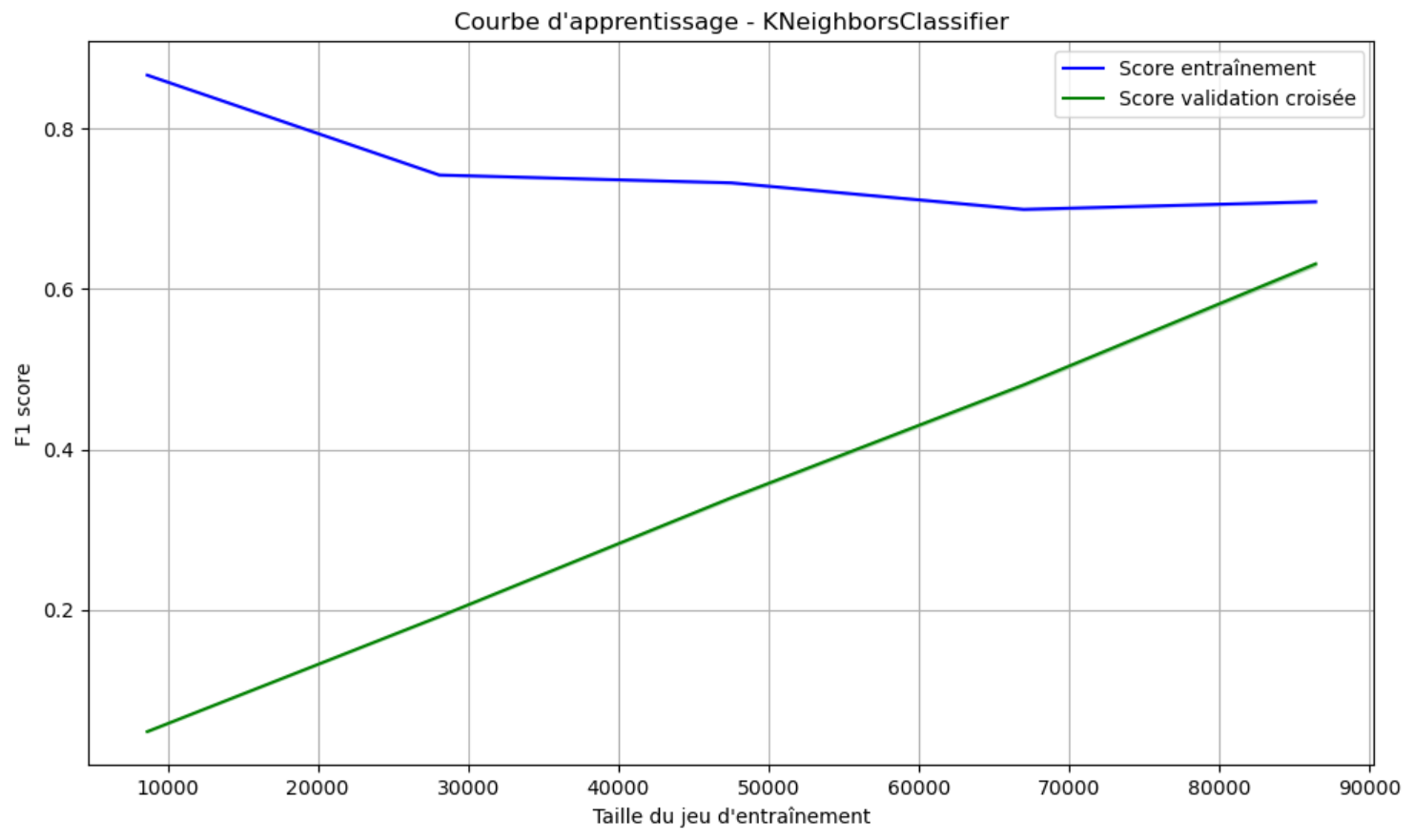
# Modèles simples



## Modélisation KNN

### Courbe d’apprentissage



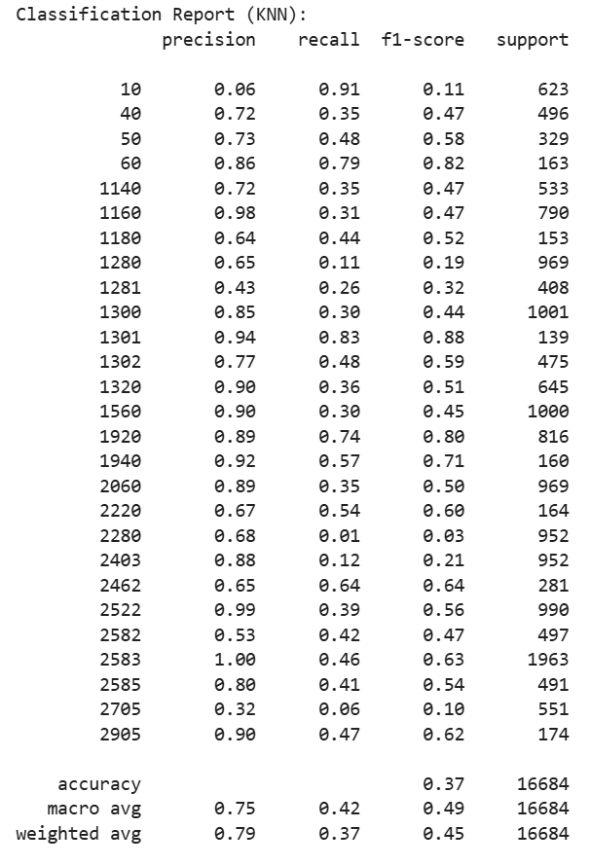
**Analyse de la courbe d’apprentissage** :

* Le score d’entrainement est élevé / bon au départ, puis chute à mesure que la taille de l’échantillon augmente en passant de 0.87 à 0.70. Le modèle s’adapte relativement bien aux données d’entrainement même si son score final n’est pas exceptionnel.
* Le score de validation commence très faible (0.05) et croît de pair avec l’augmentation de la taille de l’échantillon jusqu’à atteindre 0.62.

**Conclusion** :

Finalement, on se retrouve avec un écart non négligeable entre le score d’entrainement et le score de validation (70% vs 62%). Le score final de validation est faible et non satisfaisant. Ces deux informations traduisent que le modèle généralise mal et qu’il y a probablement un problème d’overfitting.

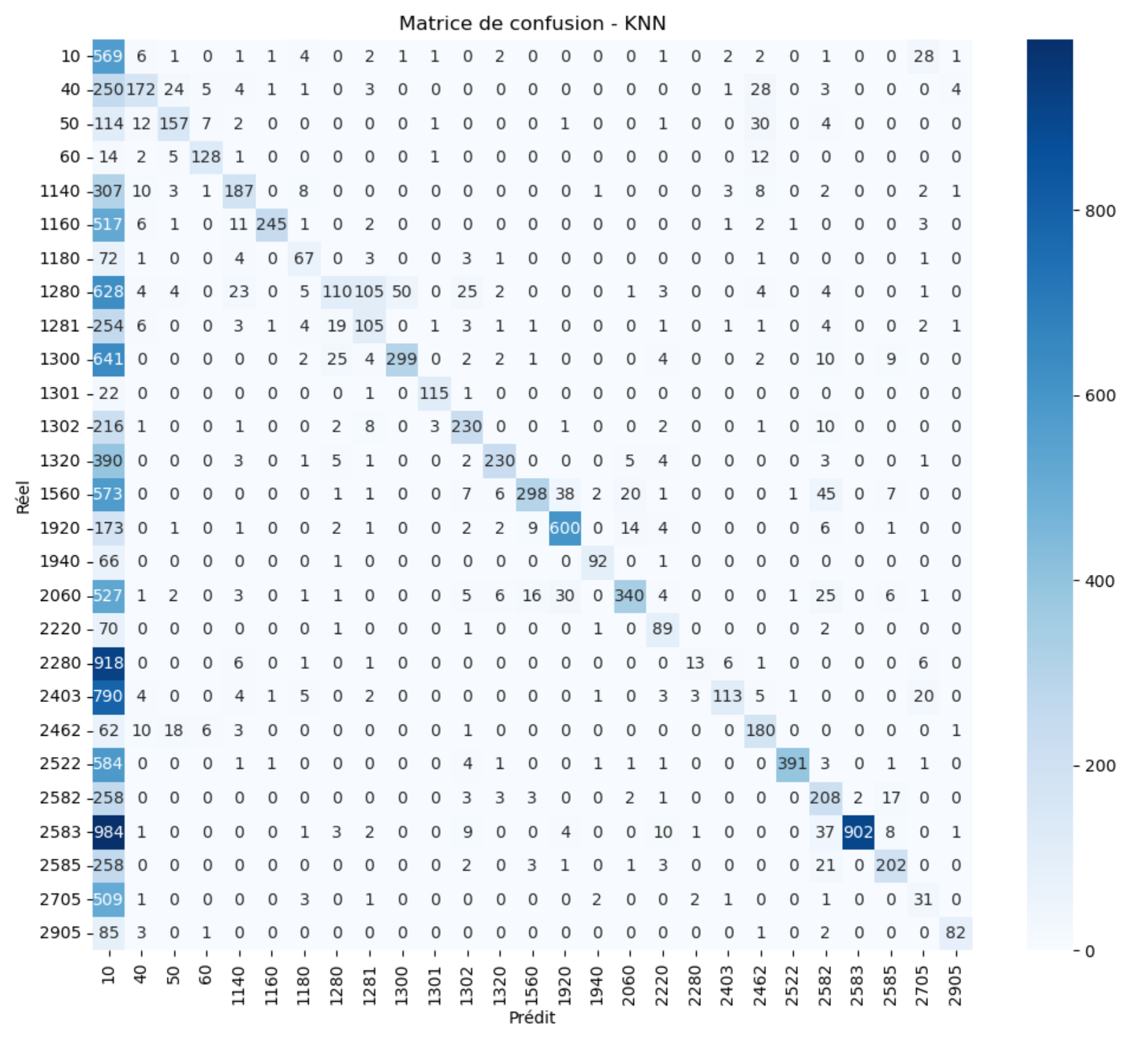
### Rapport de classification



Le rapport de classification nous permet d’observer une forte disparité entre les scores des différentes classes, allant de seulement **0.10** et **0.11** pour les classes 2705 et 10, jusqu’à **0.88** pour la classe 1301.

Globalement, le modèle n’est pas satisfaisant avec un F1-score moyen de seulement **0.49**.

### Matrice de confusion

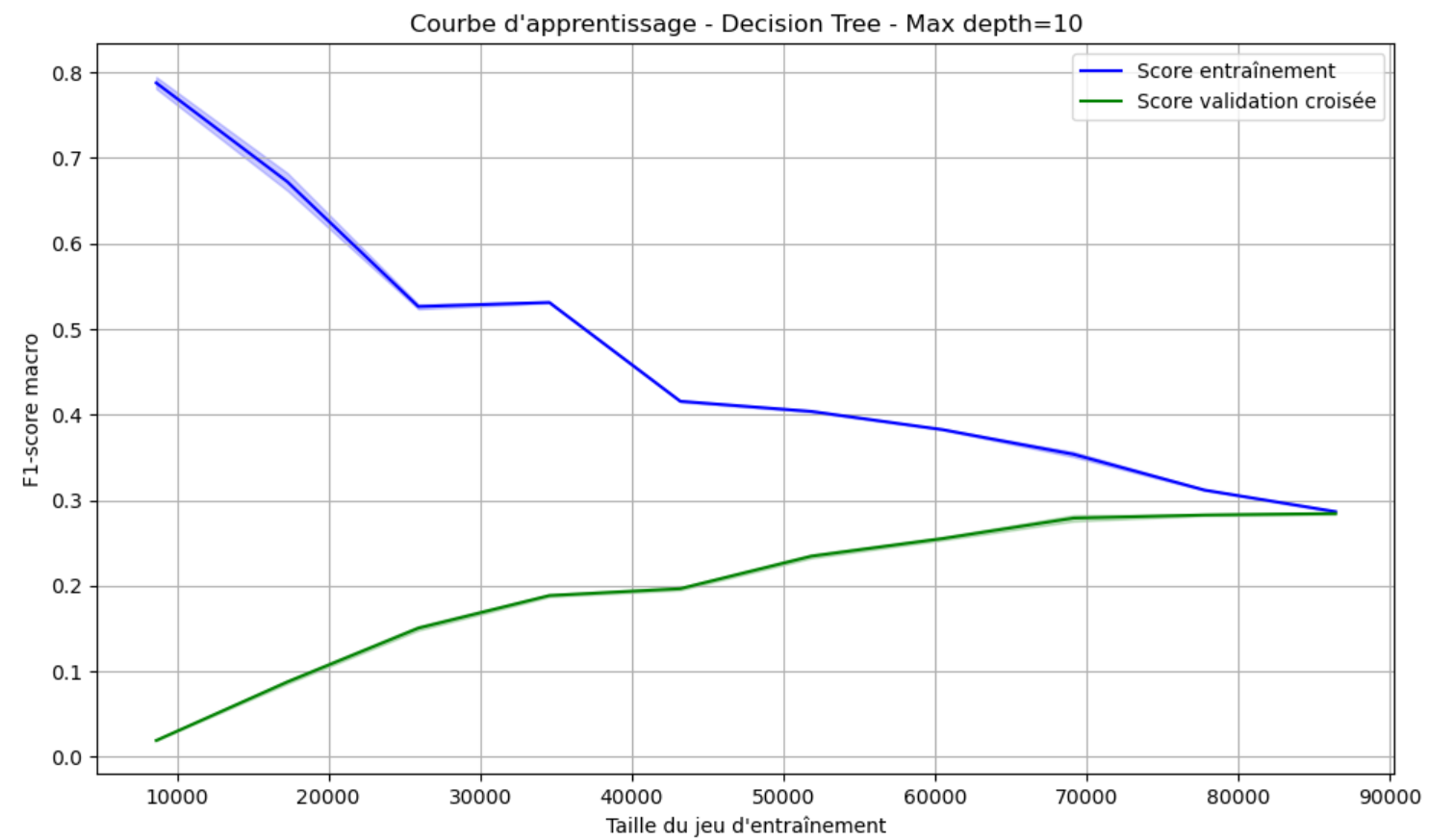


La matrice de confusion met en évidence le fait que le modèle KNN a tendance à sur-prédire **la classe 10** dans quasiment tous les cas. Par exemple : pour les 952 occurences la classe 2280, seuls 13 ont été correctement prédits et 918 ont été prédits en classe 10.

## Modélisation Decision Tree

### Courbes d’apprentissages

Nous commençons en ajustant le paramètre **max\_depth = 10**, puis nous l’augmenterons pour voir comment réagit le modèle.



**Analyse de la courbe d’apprentissage pour max\_depth = 10**

* Le score entraînement très rapidement passant de ~0.80 à ~0.28. La chute de performance sur ses propres données indique que le modèle est trop contraint / sous-ajusté.
* Le score validation croisée croît légèrement, mais reste très faible. Il finit par rejoindre le score d’entrainement en passant de 0.05 à ~0.28.

**Conclusion** : Le modèle est trop simple pour apprendre (pas assez de profondeur), même avec plus de données. Les deux courbes finissent par se rapprocher à un faible niveau de performance. On conclut donc qu’il y a un fort under-fitting.



**Analyse de la courbe d’apprentissage pour max\_depth = 50**

* Le score entraînement décroît mais reste élevé en passant de 0.85 à 0.68. La chute est beaucoup moins prononcée que pour le modèle avec max\_depth = 10. Le modèle mémorise bien les données d'entraînement, même avec de grands volumes.
* Le score validation croisée croît progressivement en passant de 0.02 à 0.65. À mesure que l'on ajoute plus de données, le modèle généralise de mieux en mieux.

**Conclusion** : les deux courbes se rejoignent quasiment, et le score final de 0.65 est bien meilleur que pour le modèle précédent qui convergeait à 0.29. Il ne semble pas y avoir d’overfitting ou d’underfitting prononcé. Cependant, même si le score est meilleur, 0.65 reste tout de même faible.

Une image contenant ligne, texte, Tracé, diagramme

Le contenu généré par l’IA peut être incorrect.

**Analyse de la courbe d’apprentissage sans max\_depth**

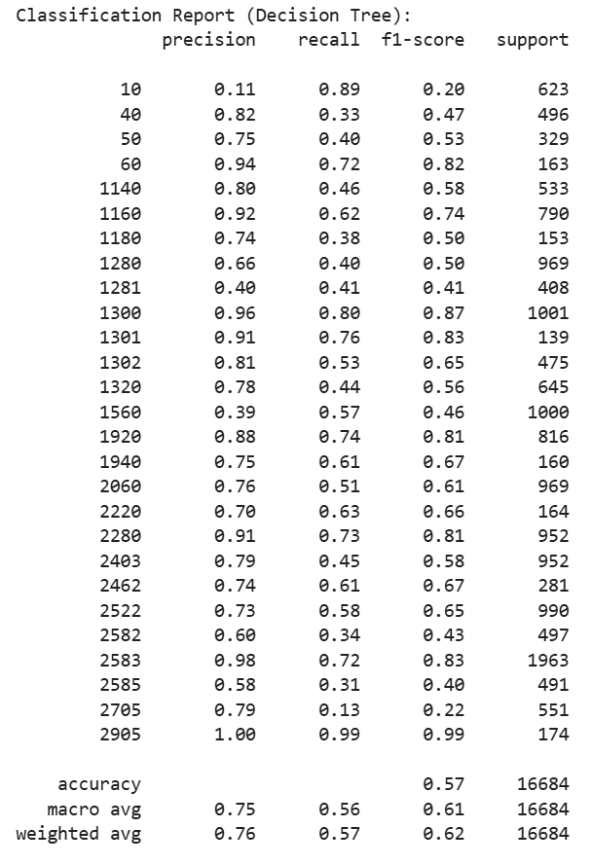
Pour finir, nous tester un modèle sans spécifier de profondeur maximale.

* Comme nous pouvions nous y attendre, le score d'entraînement est toujours égal à 1. L'arbre de décision s'adapte exactement aux données d'entrainement jusqu'à mémoriser les données. Il y a un problème fort de surapprentissage.
* Le score de validation est très faible avec peu de données et croit avec l'augmentation de la taille des données. Malheureusement, il ne parvient par à rejoindre le score d’entrainement et plafonne à 0.81.

**Conclusion** : l’écart entre le score d’entrainement et le score de validation est significatif (0.19). Il y a un overfitting clair. Pour la suite, nous retiendrons donc le modèle avec max\_depth = 50 qui semble être le meilleur compromis.

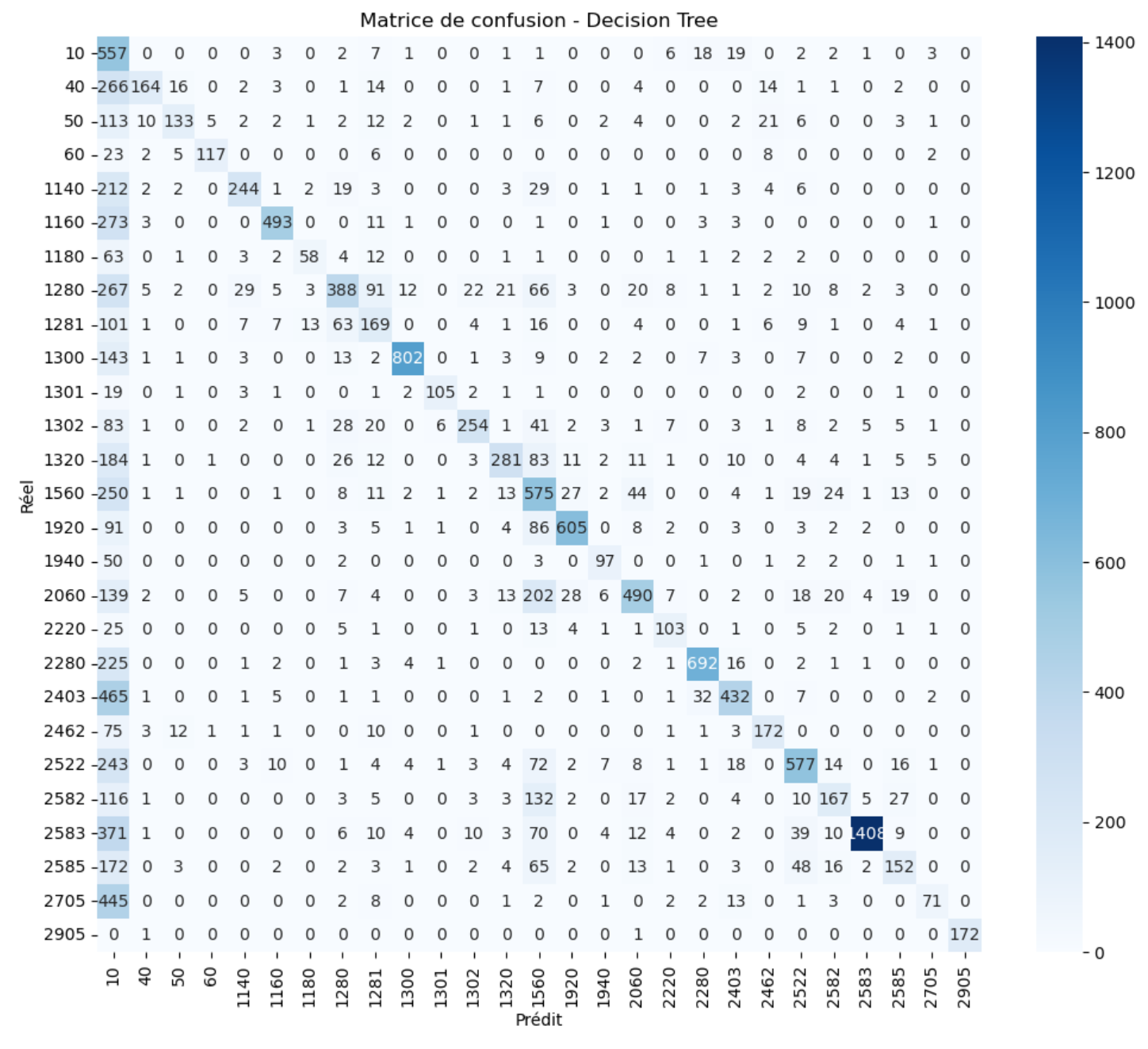
### Rapport de classification

Rapport de classification du modèle Decision Tree (max\_depth = 50)



Comme pour le modèle KNN précédent, nous observons une forte disparité entre les classes, allant de 0.20 pour la classe 10, jusqu’à 0.99 pour la classe 2905. Globalement, les scores du modèle Decision Tree est meilleur que ceux du modèle KNN avec un **F1-score moyen de 0.61**. Cependant ce score reste faible et non satisfaisant.

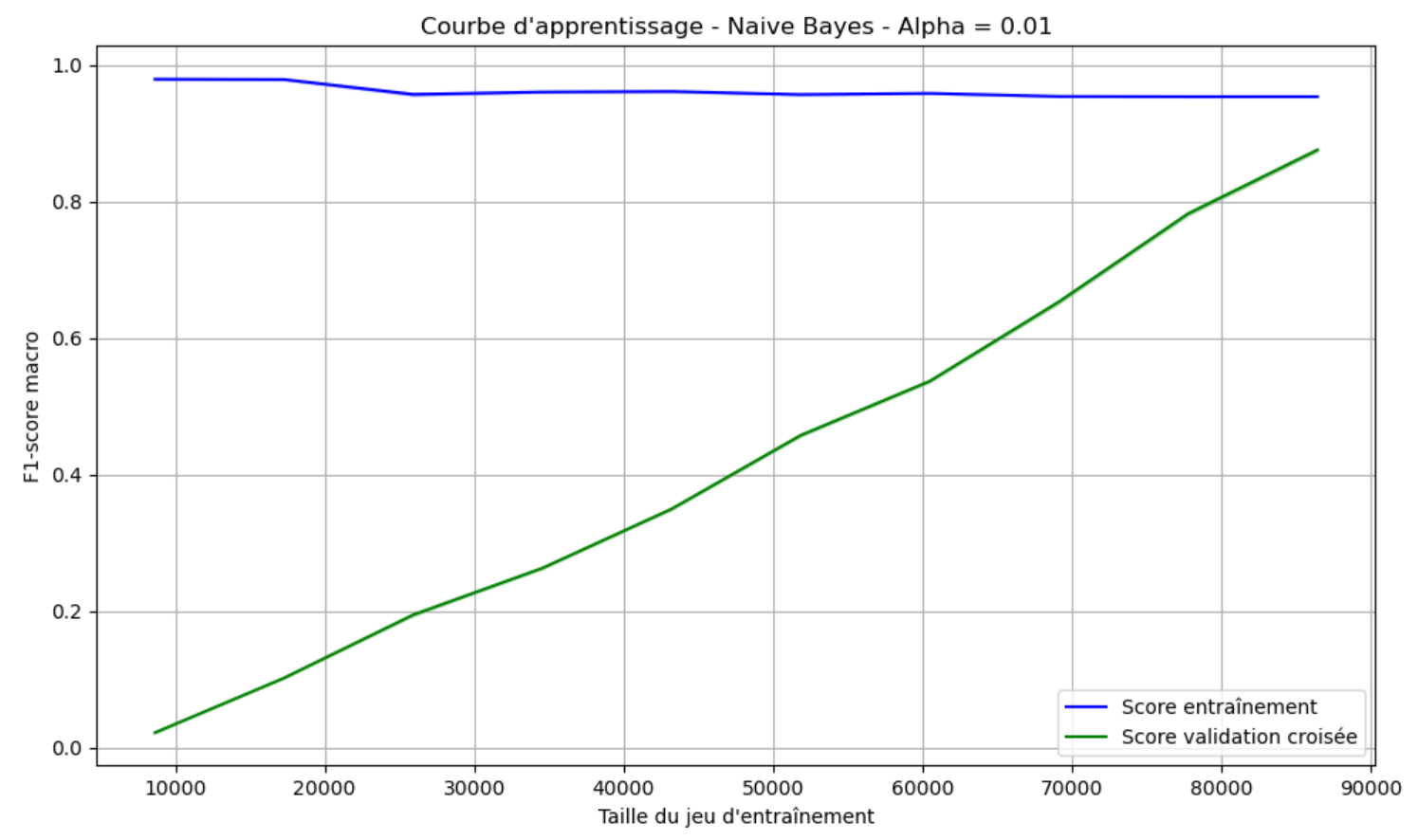
### Matrice de confusion

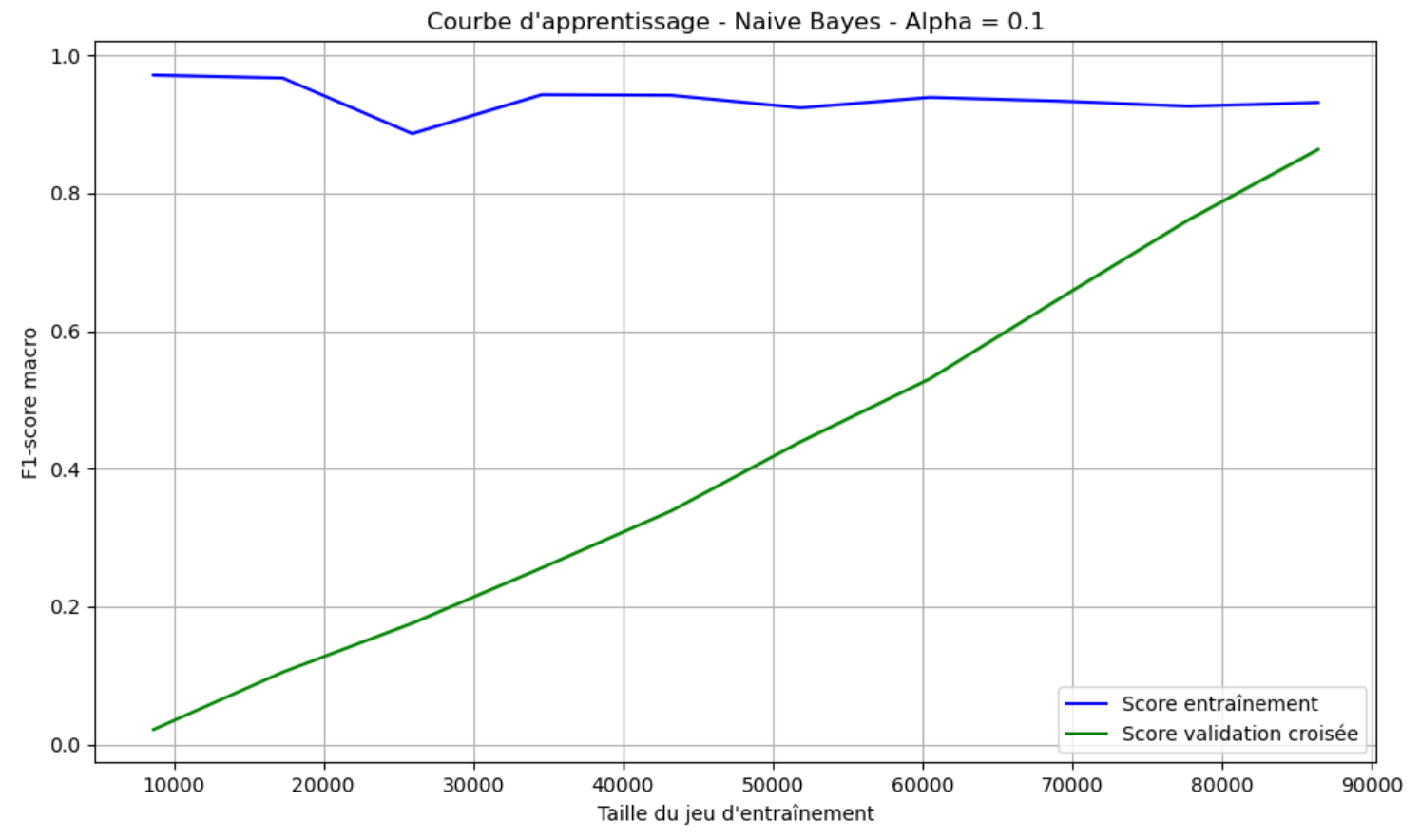


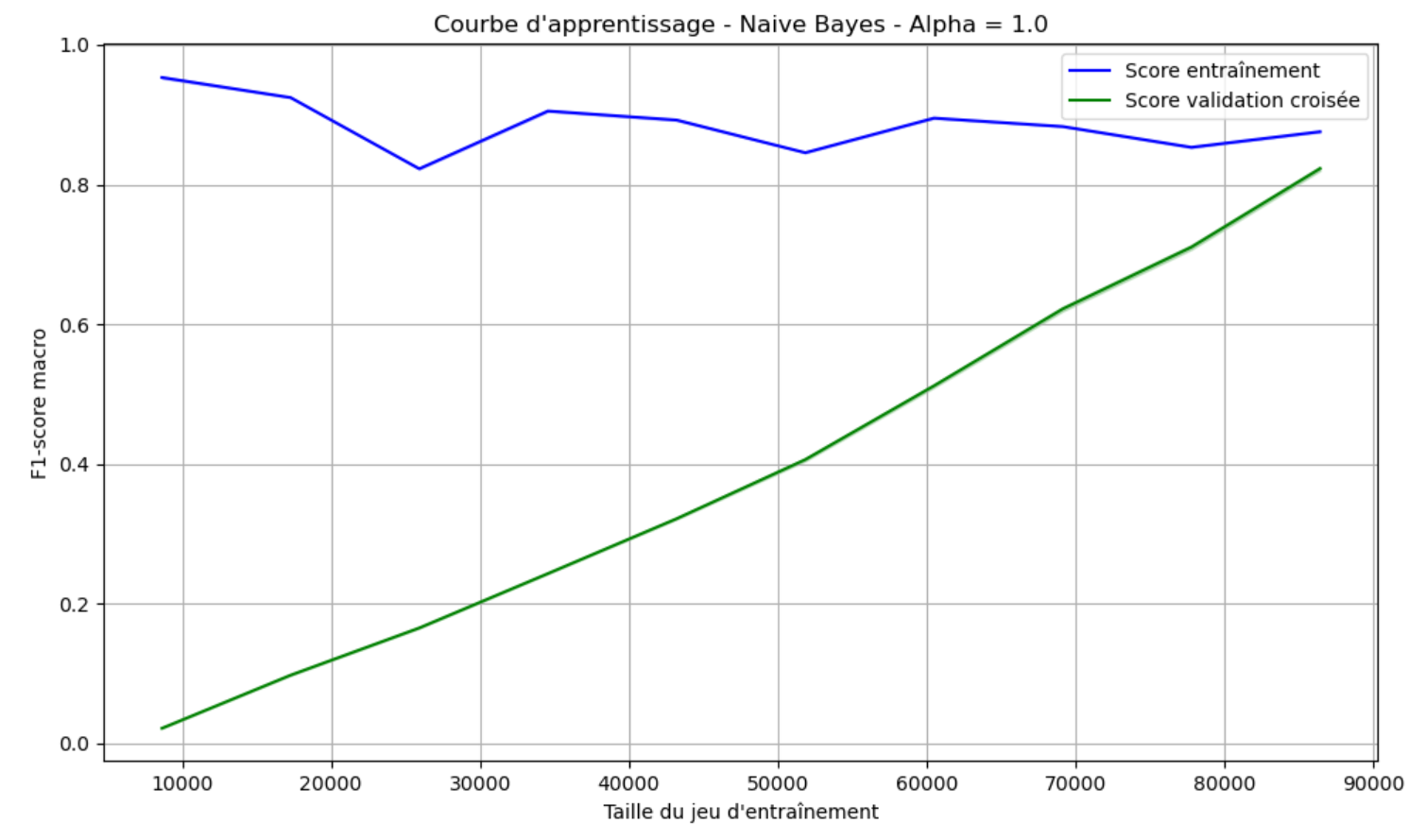
La matrice de confusion met en évidence le même problème que celui observé avec le modèle KNN : il y a une forte confusion avec la classe 10 qui est sur-prédite.

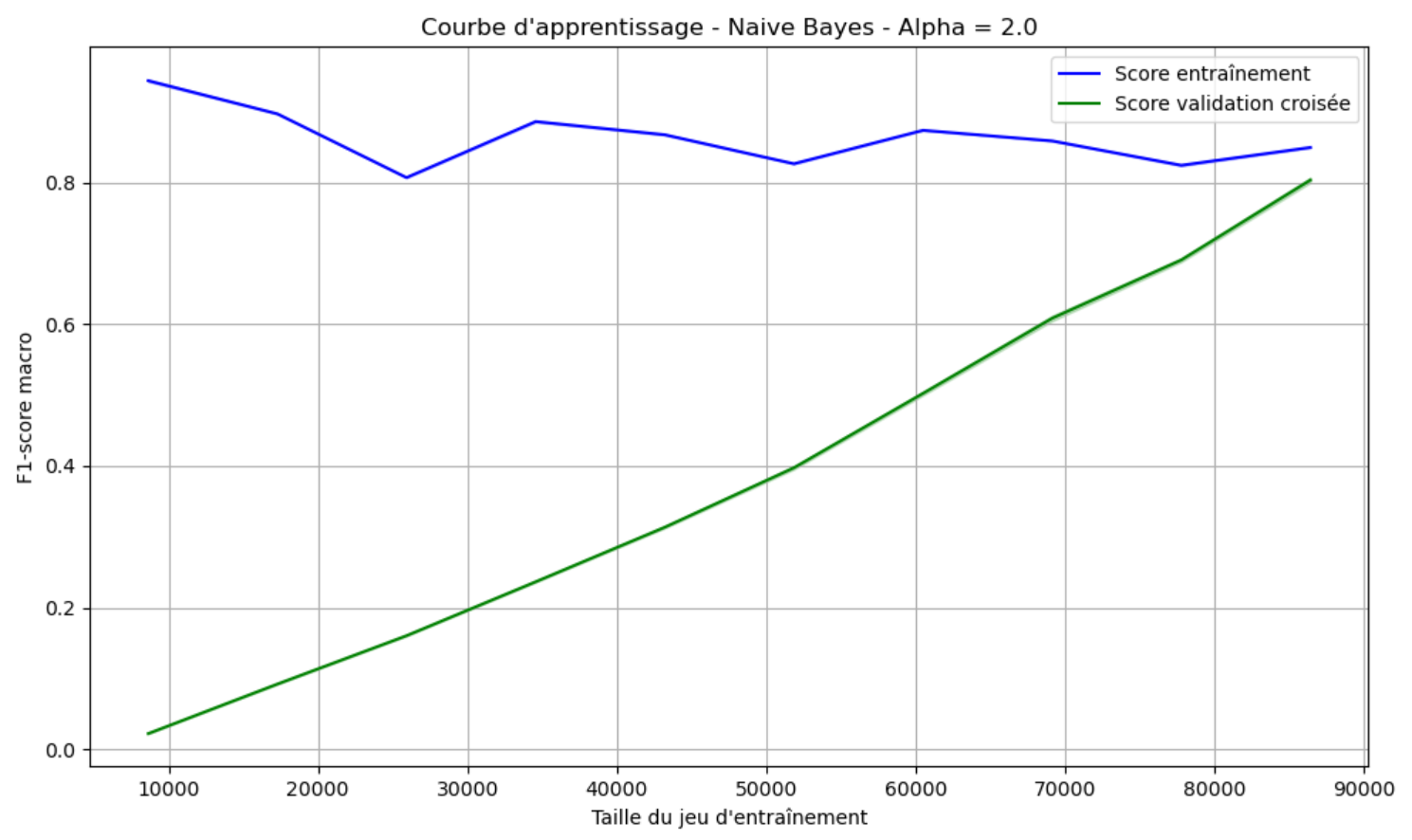
## Modélisation Naïve Bayes

### Courbes d’apprentissages



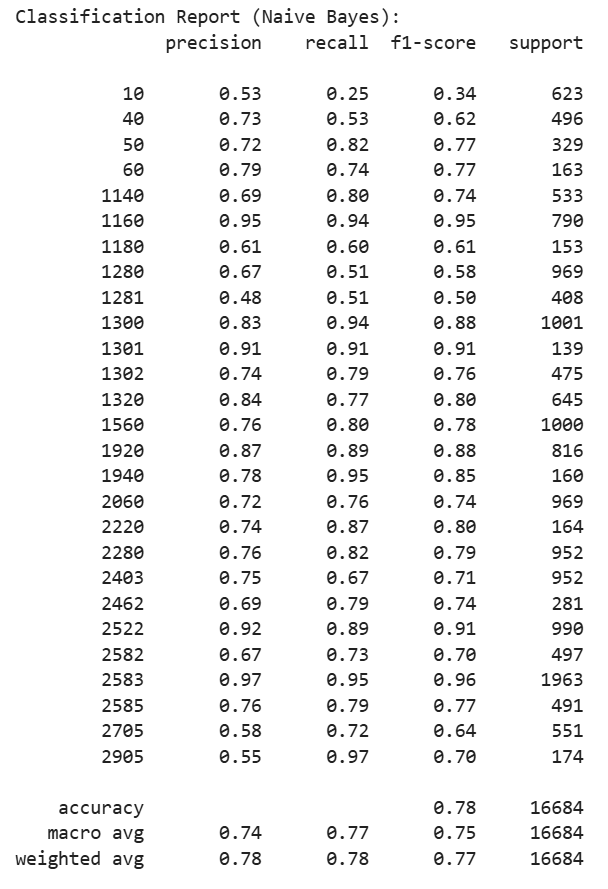




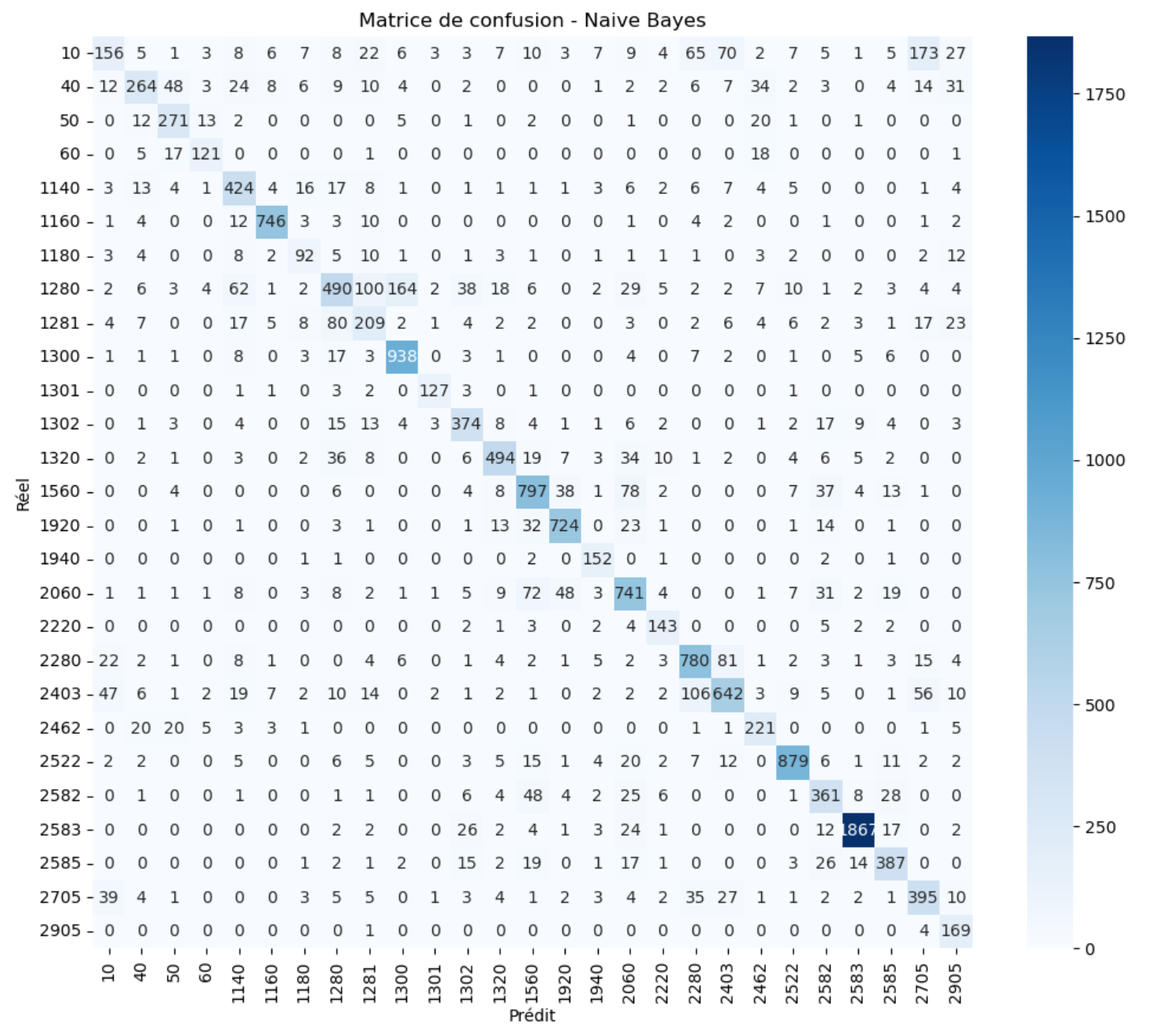


### Rapport de classification

Rapport de classification du modèle Naïve Bayes avec Alpha = 0.1

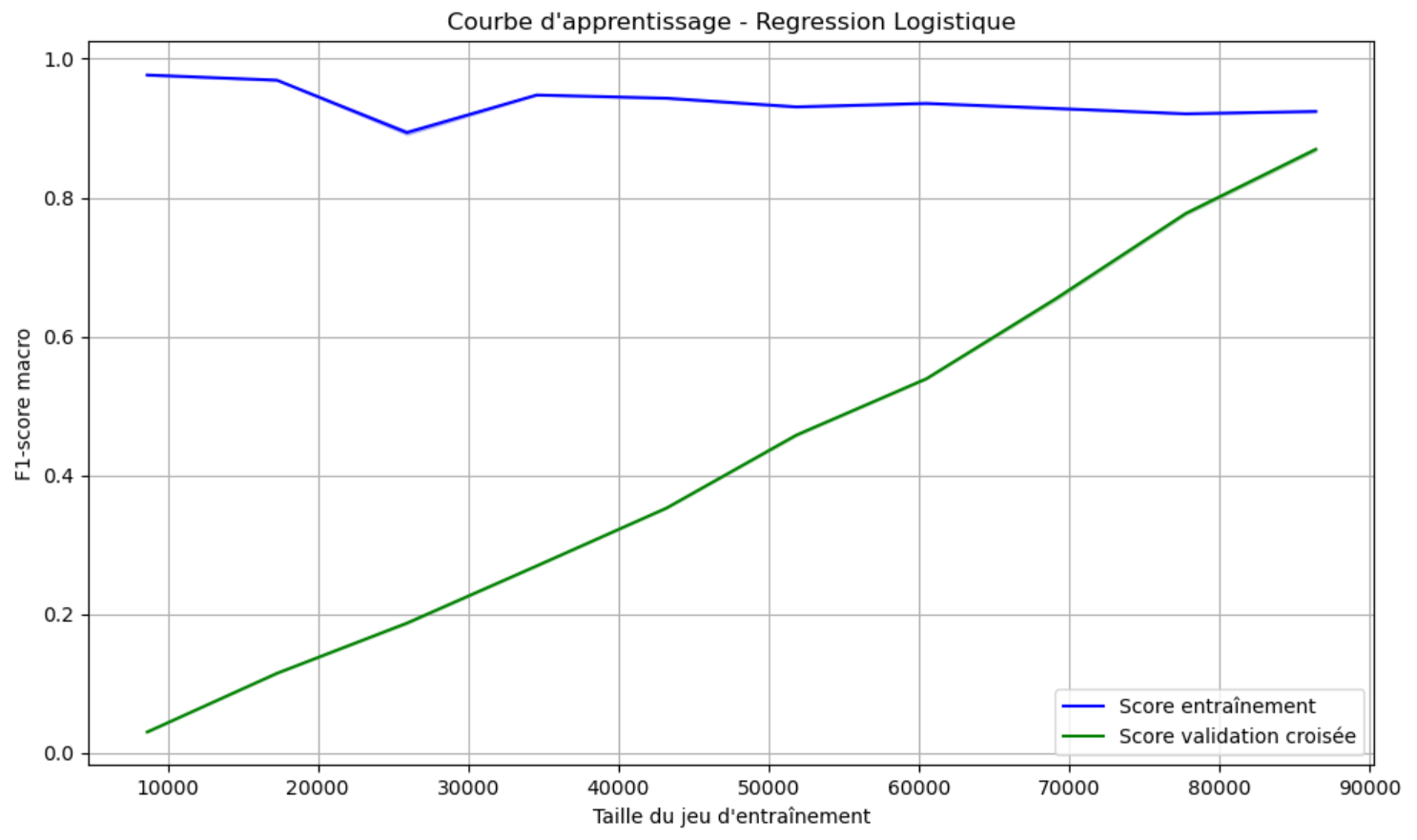


### Matrice de confusion

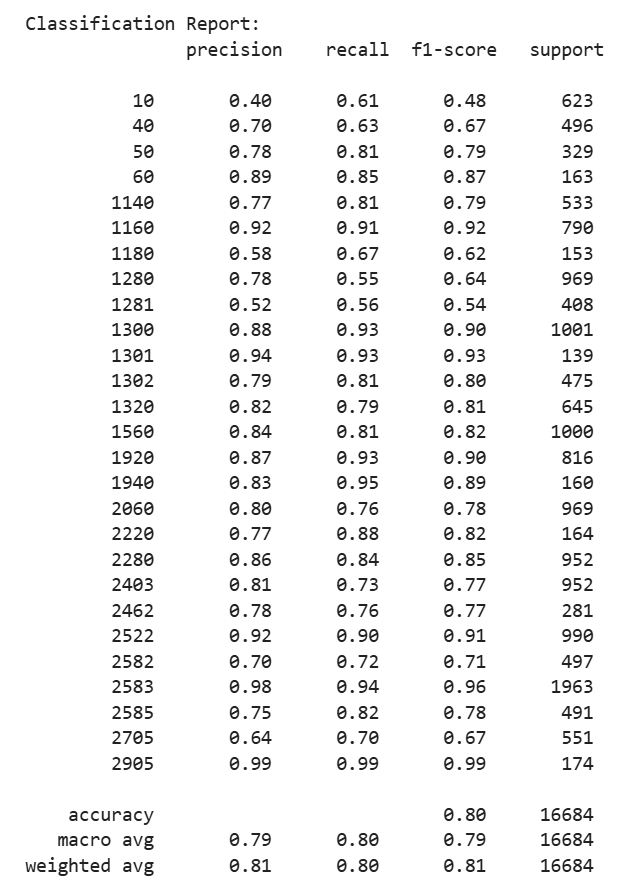


## Modélisation Logistic Regression

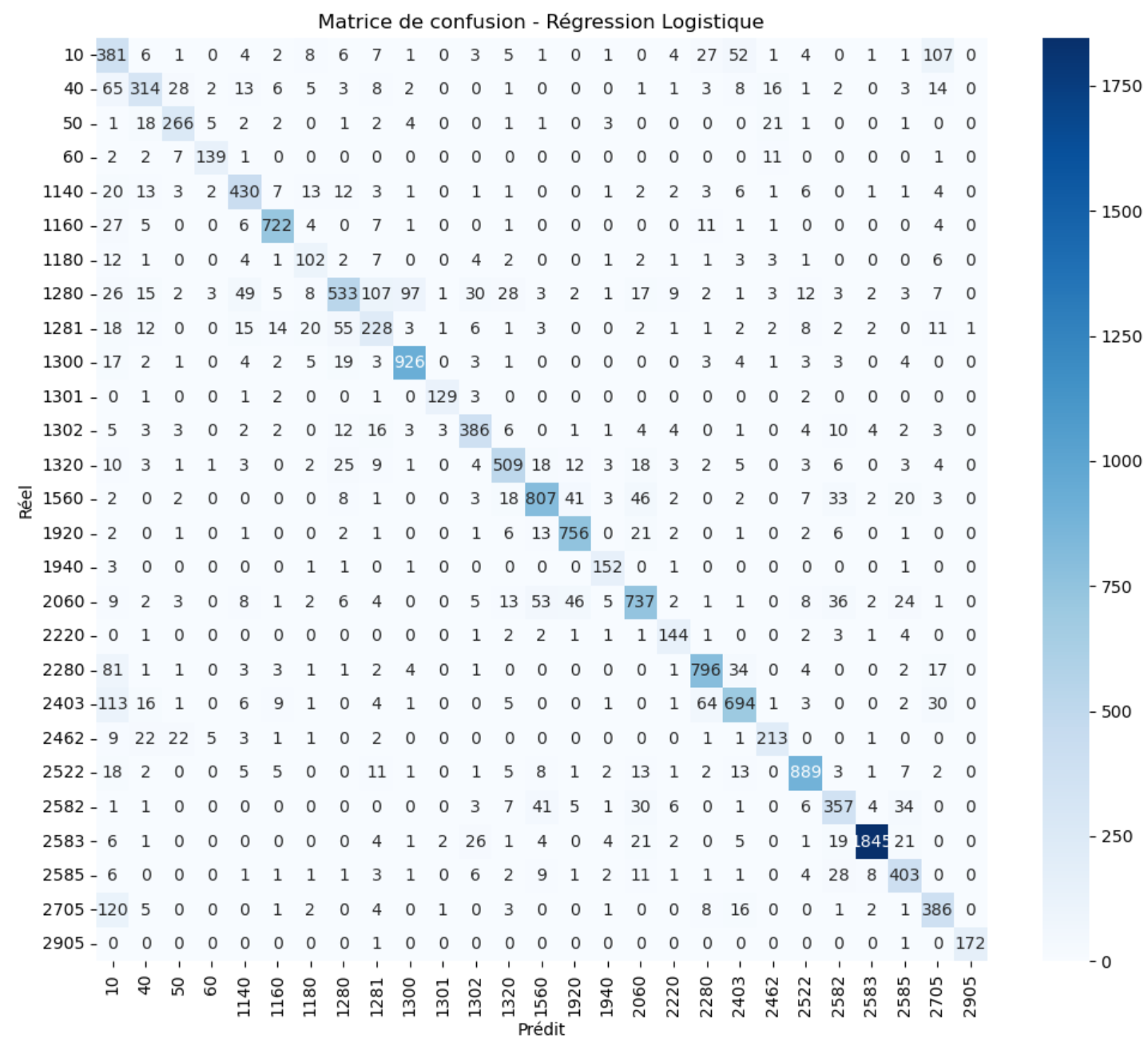
### Courbe d’apprentissage



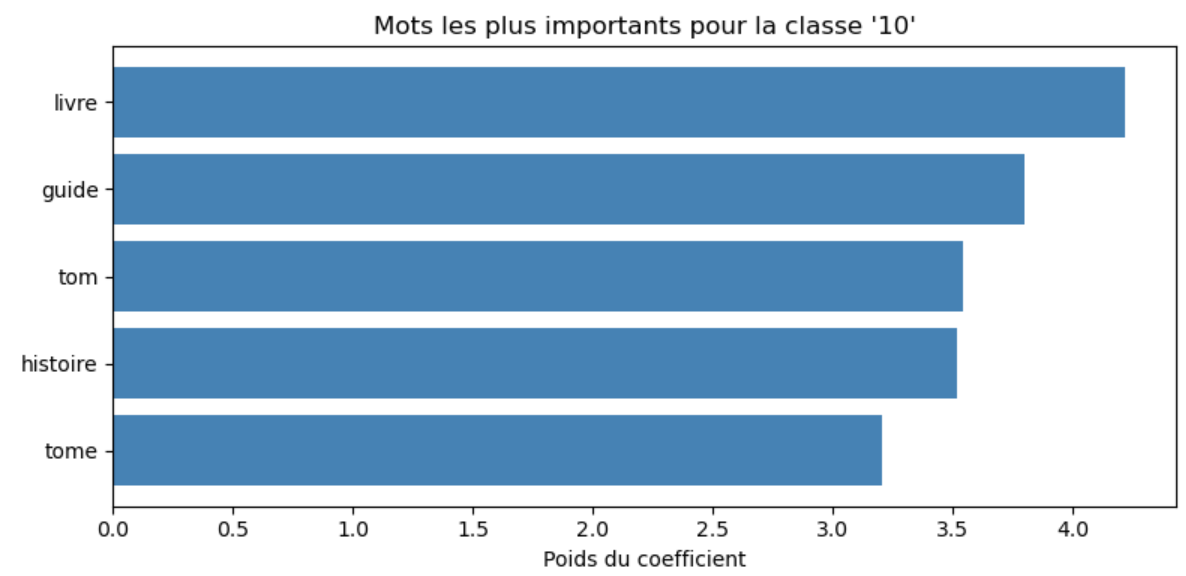
### Rapport de classification

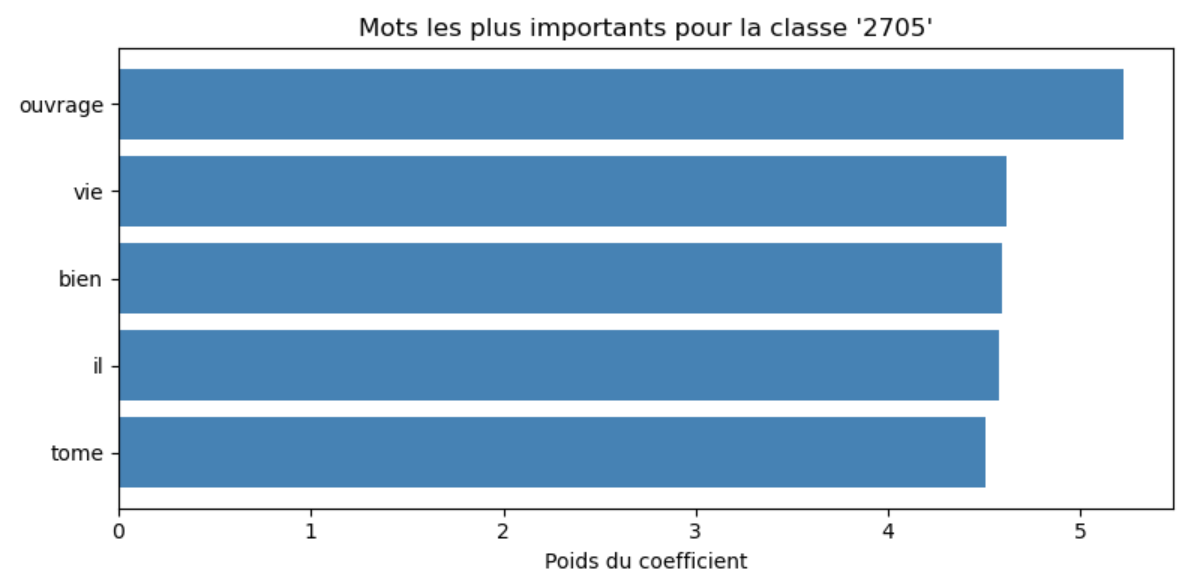


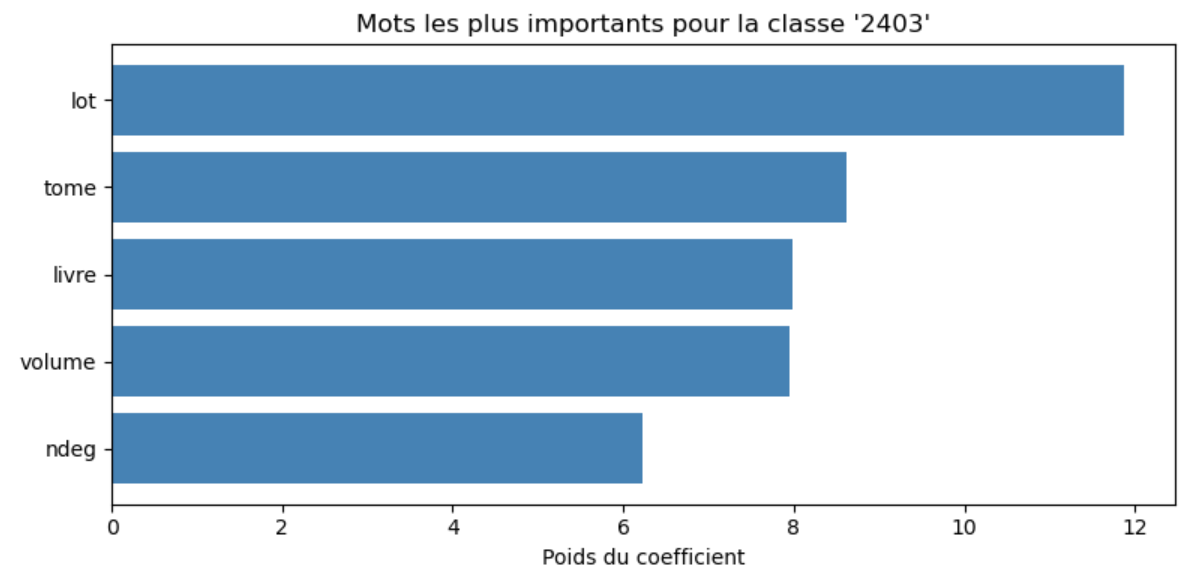
### Matrice de confusion



### Importance des mots







## Modélisation Random Forest

### Courbe d’apprentissage

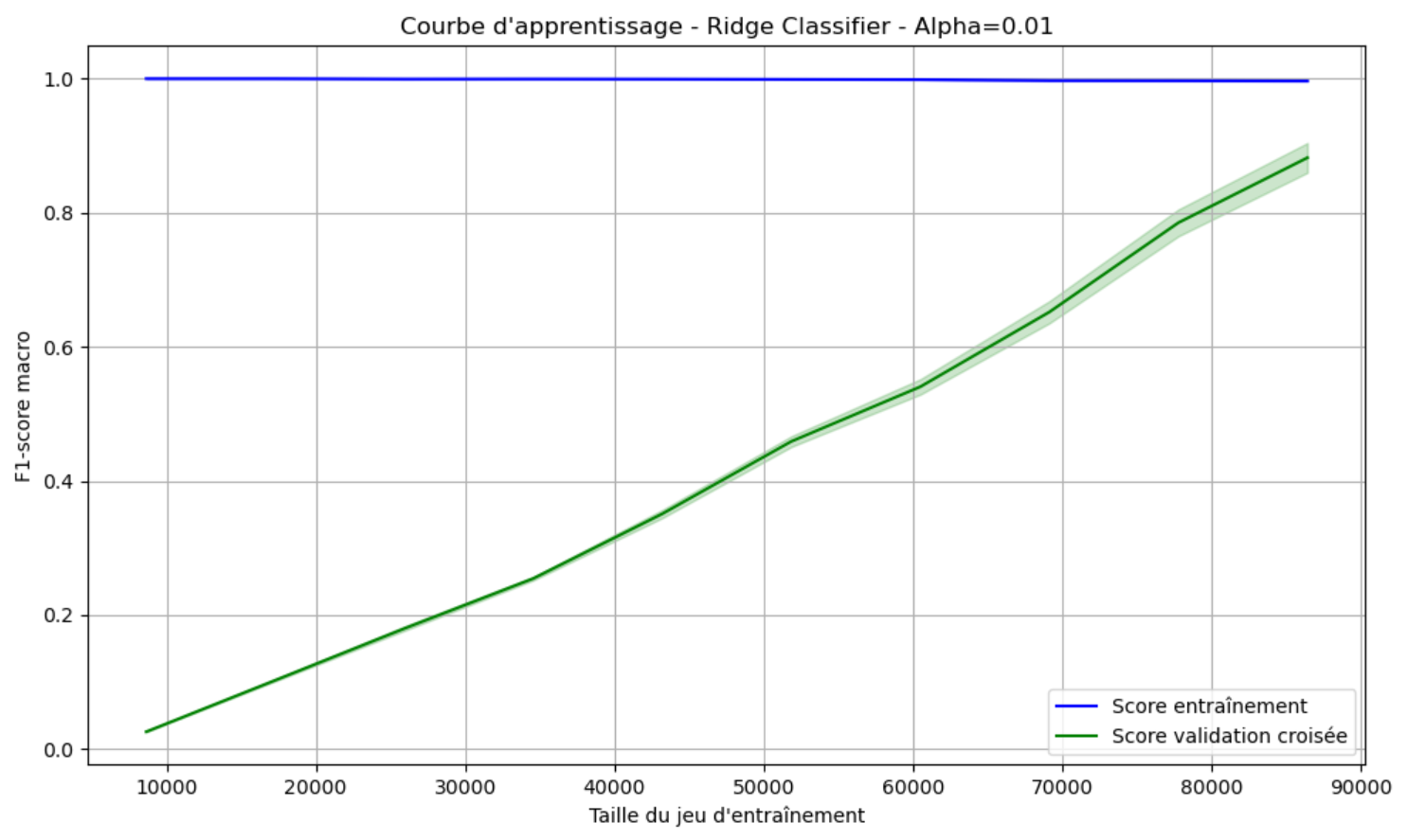
### Rapport de classification

### Matrice de confusion

## Modélisation Ridge Classifier

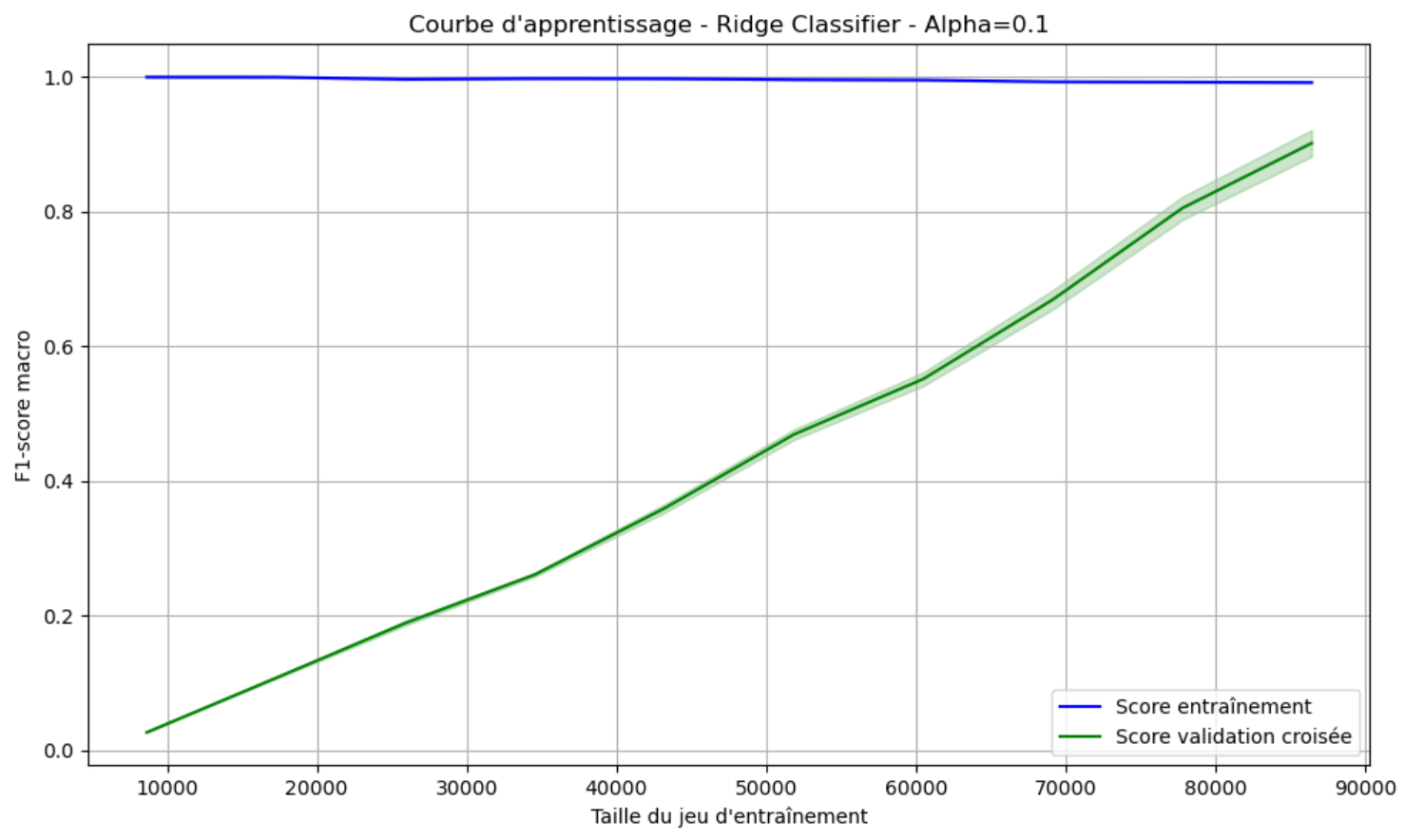
### Courbes d’apprentissages

Nous allons jouer sur le paramètre Alpha (0.01, 0.1, 1, 10, 100) pour trouver le meilleur modèle possible.



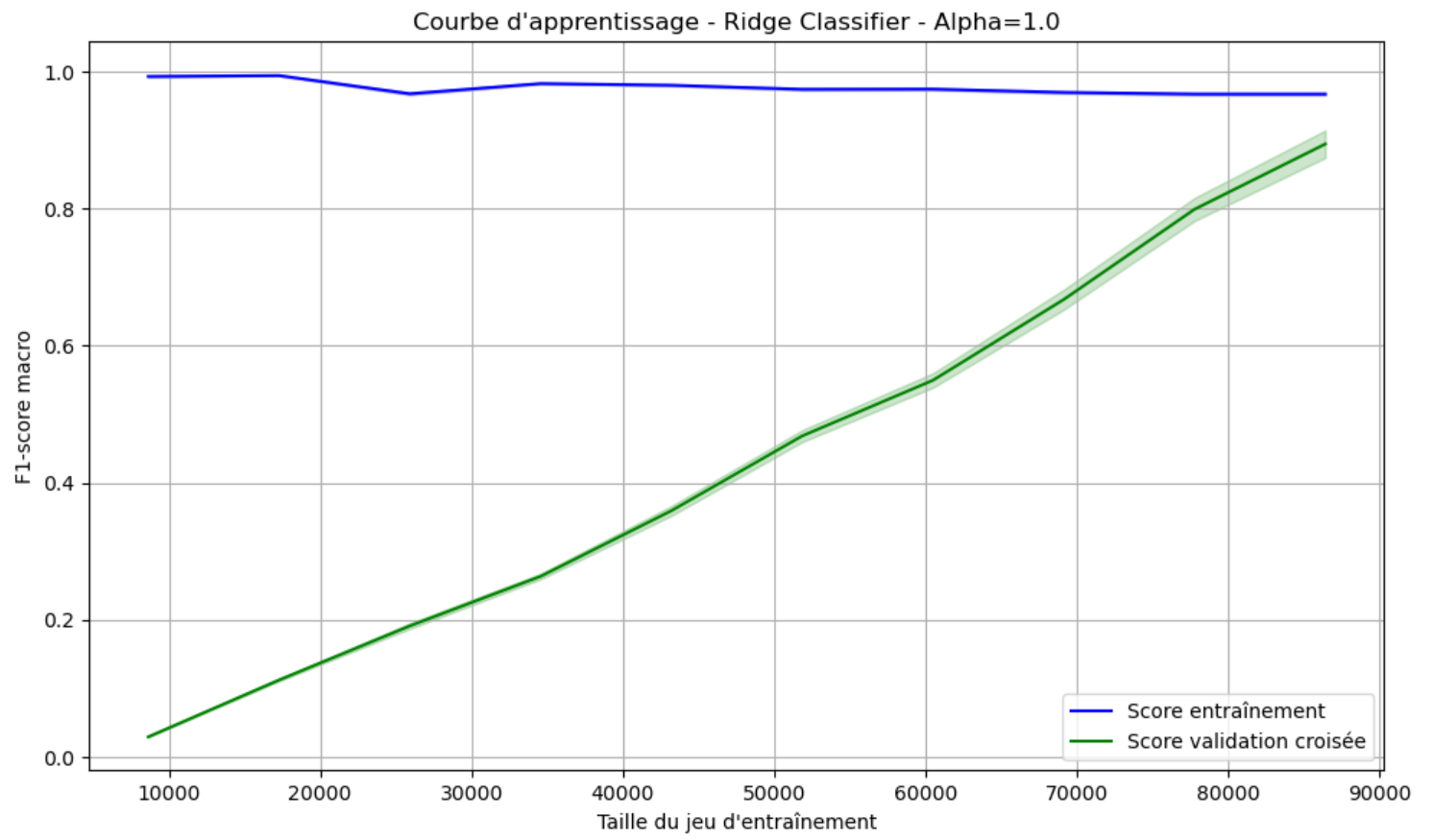
**Analyse de la courbe d’apprentissage pour Alpha = 0.01**

* Avec le paramètre Alpha = 0.01, la régularisation est trop forte. Le score d’entrainement ne montre aucune variation tout au long de l’apprentissage et reste strictement égal à 1. Il y a donc un overfitting fort.
* Le score validation croisée démarre bas, puis monte progressivement jusqu'à atteindre environ 0.87. Cependant, l’écart avec le score d’entrainement reste significatif. Le modèle n'arrive pas à généraliser.



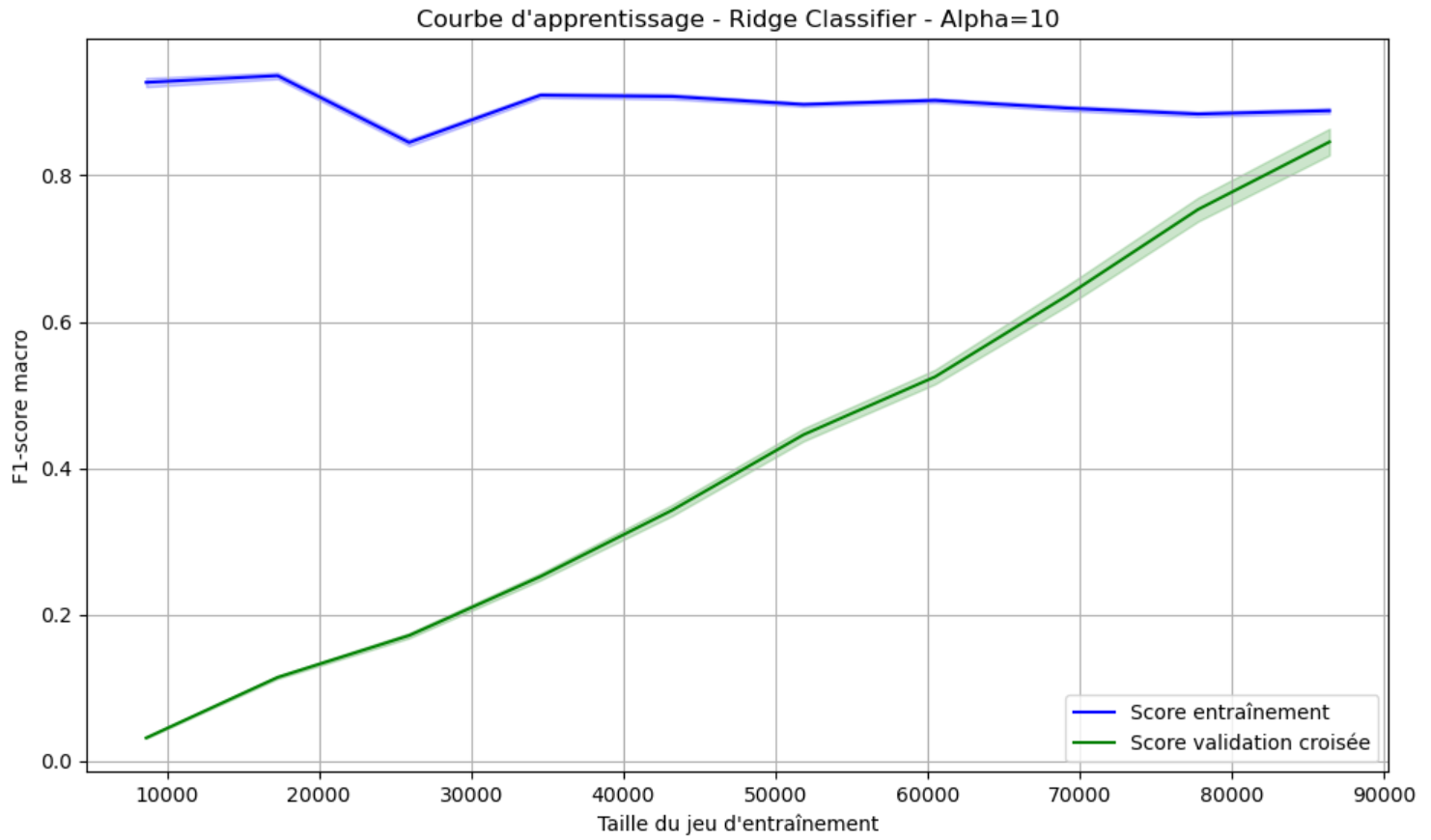
**Analyse de la courbe d’apprentissage pour Alpha = 0.1**

* Les observations sont sensiblement les mêmes avec le paramètre Alpha = 0.1 qu’avec le paramètre Alpha = 0.01. On constate que le score d’entrainement diminue très légèrement sur la fin pour tomber à 0.99. L’overfitting reste cependant trop prononcé.
* Le score validation croisée démarre bas, puis monte progressivement jusqu'à atteindre environ 0.88. Cependant, l’écart avec le score d’entrainement reste significatif. Le modèle n'arrive pas à généraliser.



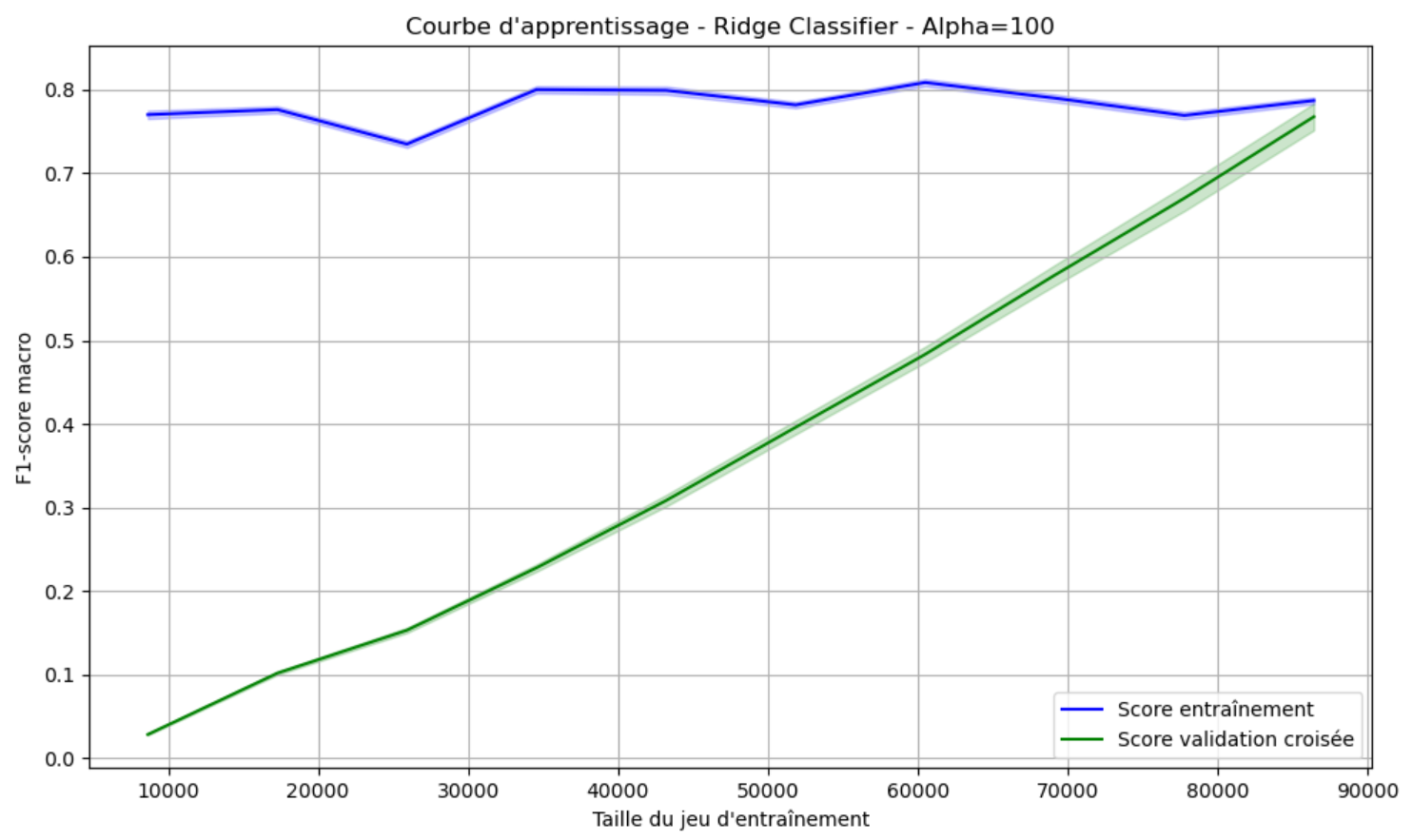
**Analyse de la courbe d’apprentissage pour Alpha = 1**

* Le score d'entrainement reste élevé, mais est plus réaliste que les 2 modèles précédent, en terminant à 0.95
* Le score de validation est également bon : 0.90
* Le modèle est meilleur que les 2 précédents. Il reste toutefois un écart non négligeable entre entrainement et validation.



**Analyse de la courbe d’apprentissage pour Alpha = 10**

* Le score d'entrainement diminue encore un peu par rapport au modèle précédent, mais reste très bon : 0.90
* Le score de validation se rapproche très fortement de la courbe d'apprentissage 0.85. L’écart entre le score d’apprentissage et le score de validation est encore réduit par rapport aux modèles précédents
* Il y a moins d'overfitting que pour les courbes précédentes.



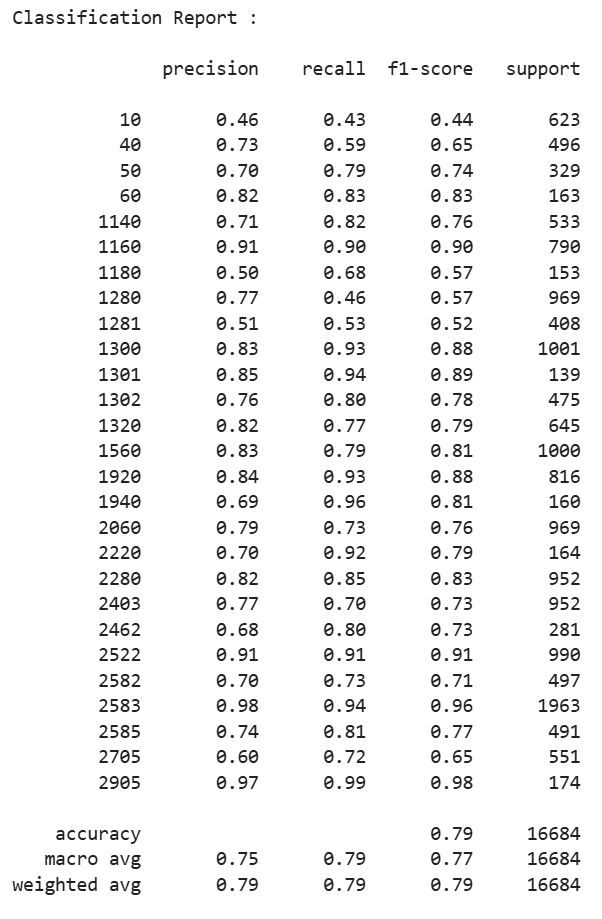
**Analyse de la courbe d’apprentissage pour Alpha = 100**

* Le score d'apprentissage est trop faible par rapport aux tests précédents (~ 0.78). Il y a sans doute un sous-apprentissage
* Idem pour le score de validation. Bien qu'il rejoigne le score d'apprentissage, le score atteint est trop faible par rapport aux tests précédents (~0.76)

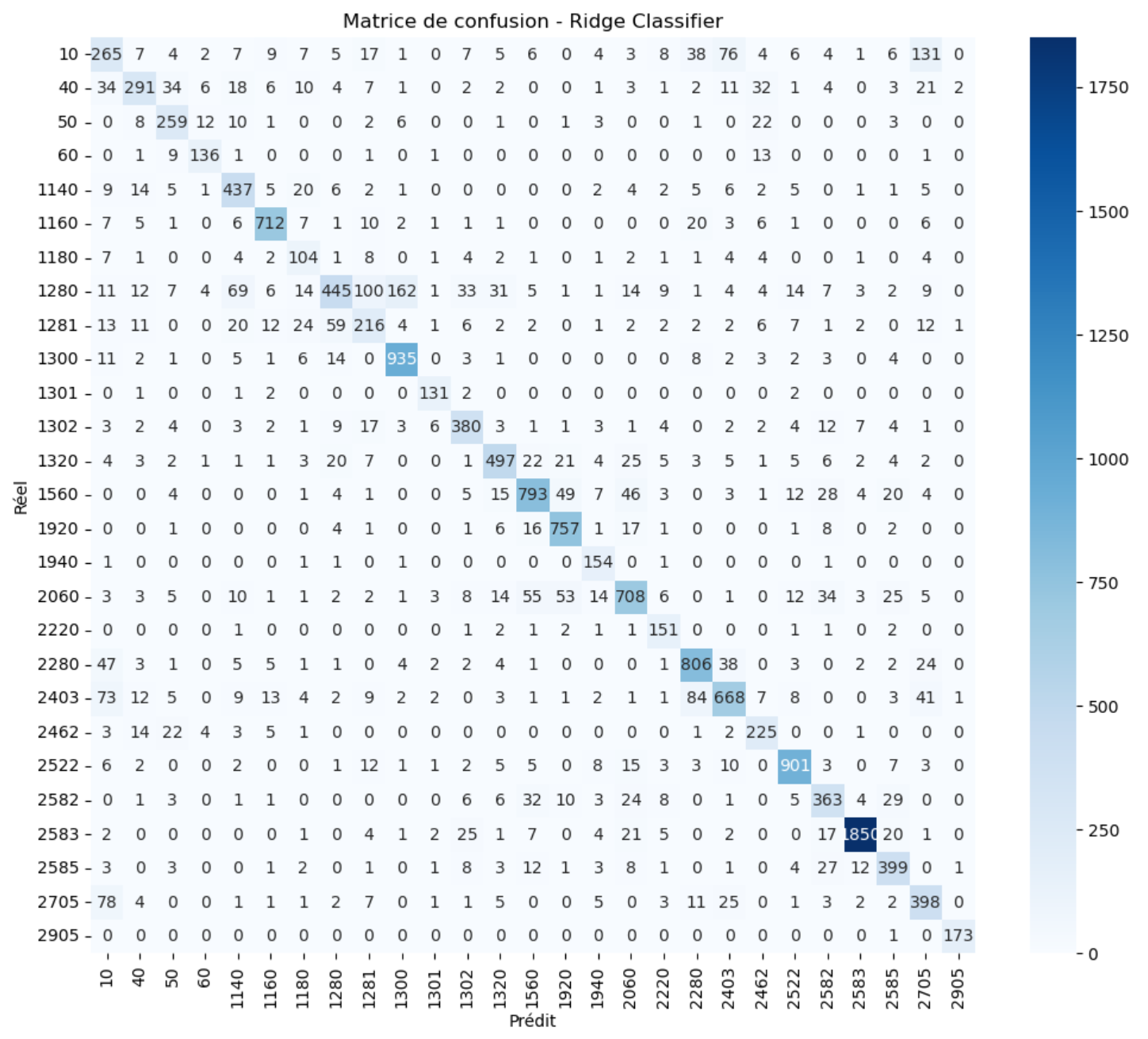
**Conclusion** : Nous en concluons que le meilleur compromis est celui avec le paramètre alpha = 10 qui a une courbe d'entrainement élevée et écart avec le score de validation relativement faible.

### Rapport de classification

Rapport de classification du Ridge Classifier avec Alpha = 10



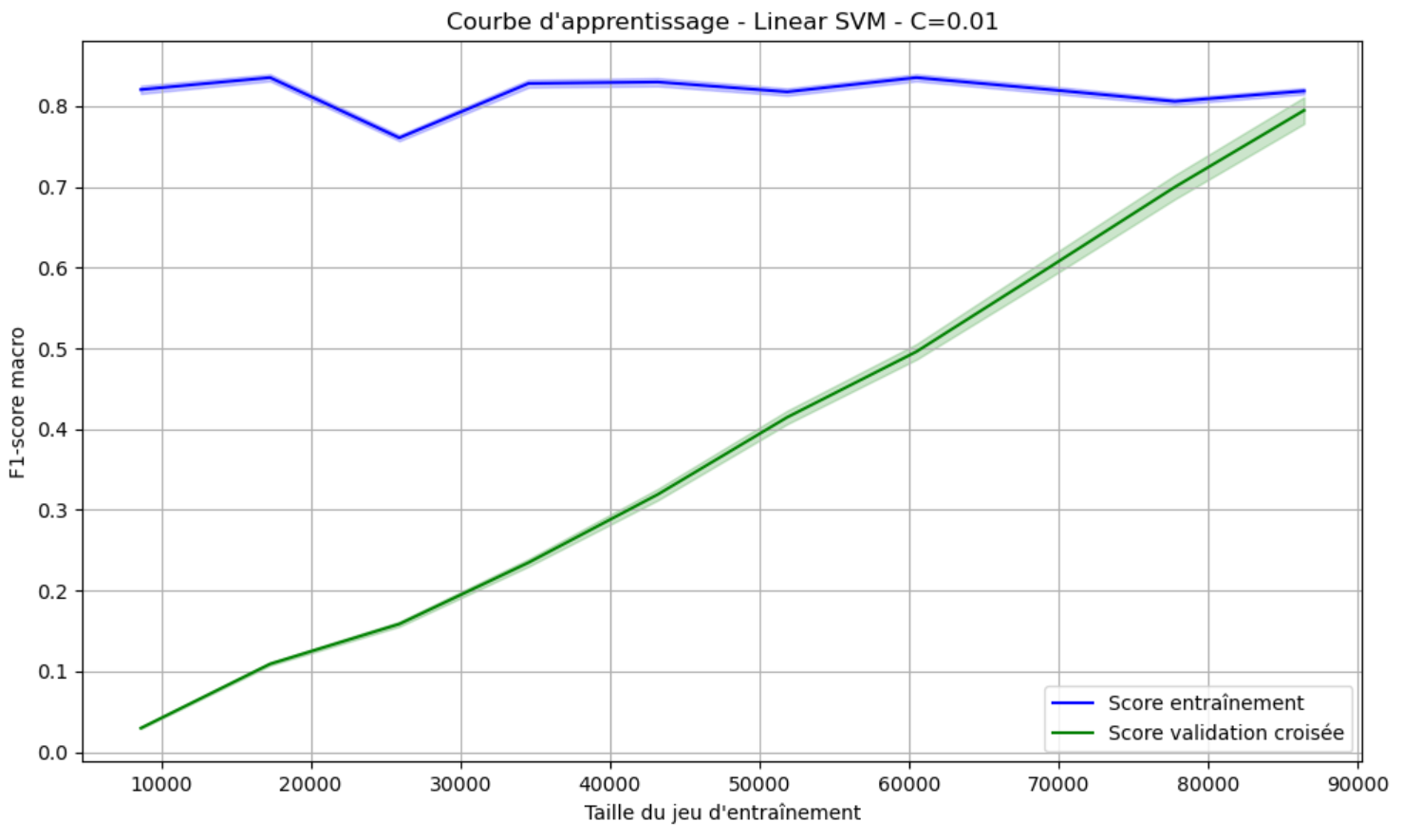
### Matrice de confusion



## Modélisation Linear SVM

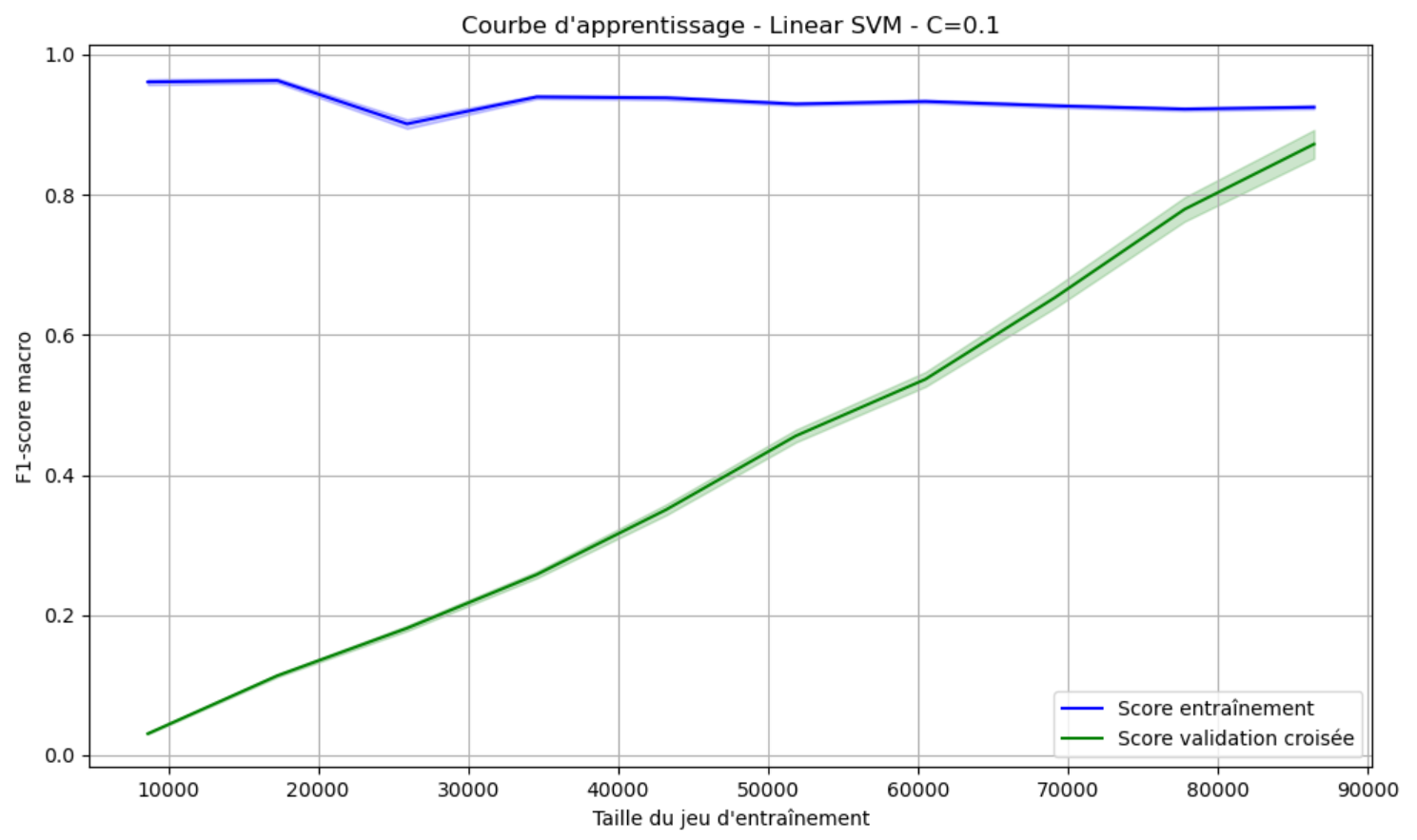
### Courbes d’apprentissages

Nous allons jouer sur le paramètre C (0.01, 0.1, 1, 10, 100) pour trouver le meilleur modèle possible.



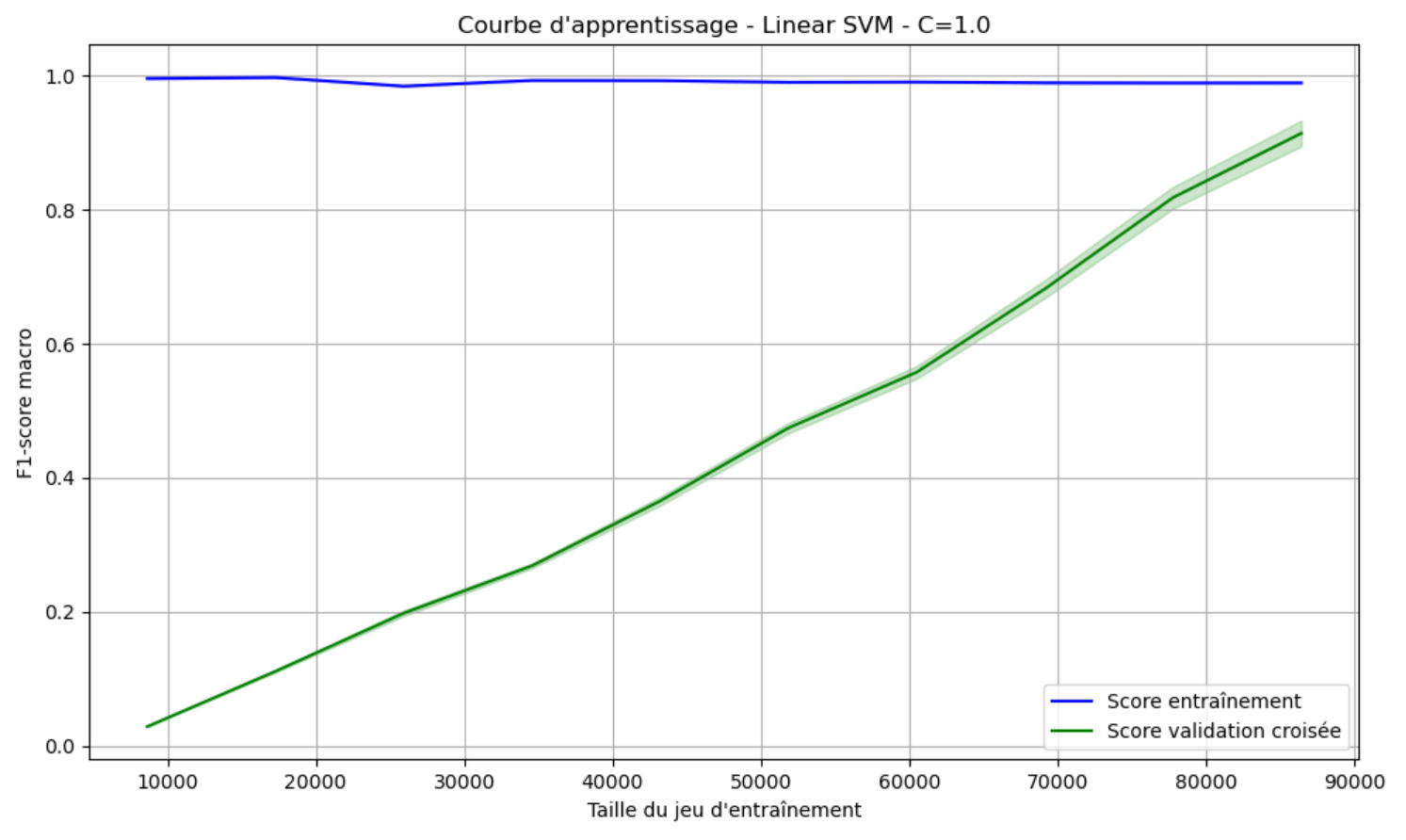
**Analyse de la courbe d’apprentissage pour C = 0.01**

* Avec le paramètre C = 0.01, l’apprentissage est assez permissif et accepte les erreurs sur les données d’entrainement. Le score d’entrainement est donc relativement éloigné d’un score de 1 qui représenterait un surapprentissage. Le score d’entraînement varie autour des 0.80 et termine à 0.82.
* Le score validation croisée démarre bas mais remonte rapidement vers la courbe d’entrainement pour atteindre les 0.80.



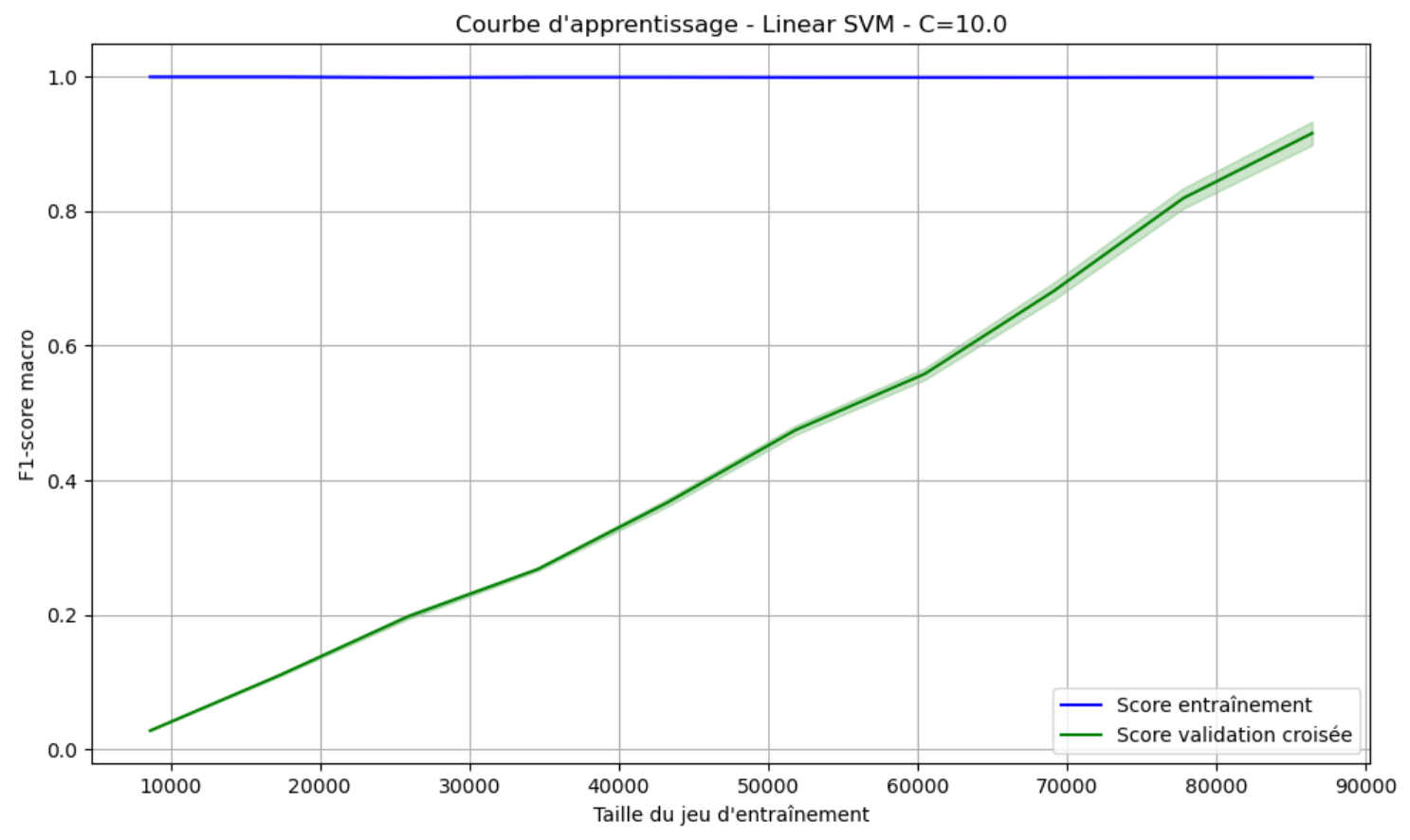
**Analyse de la courbe d’apprentissage pour C = 0.1**

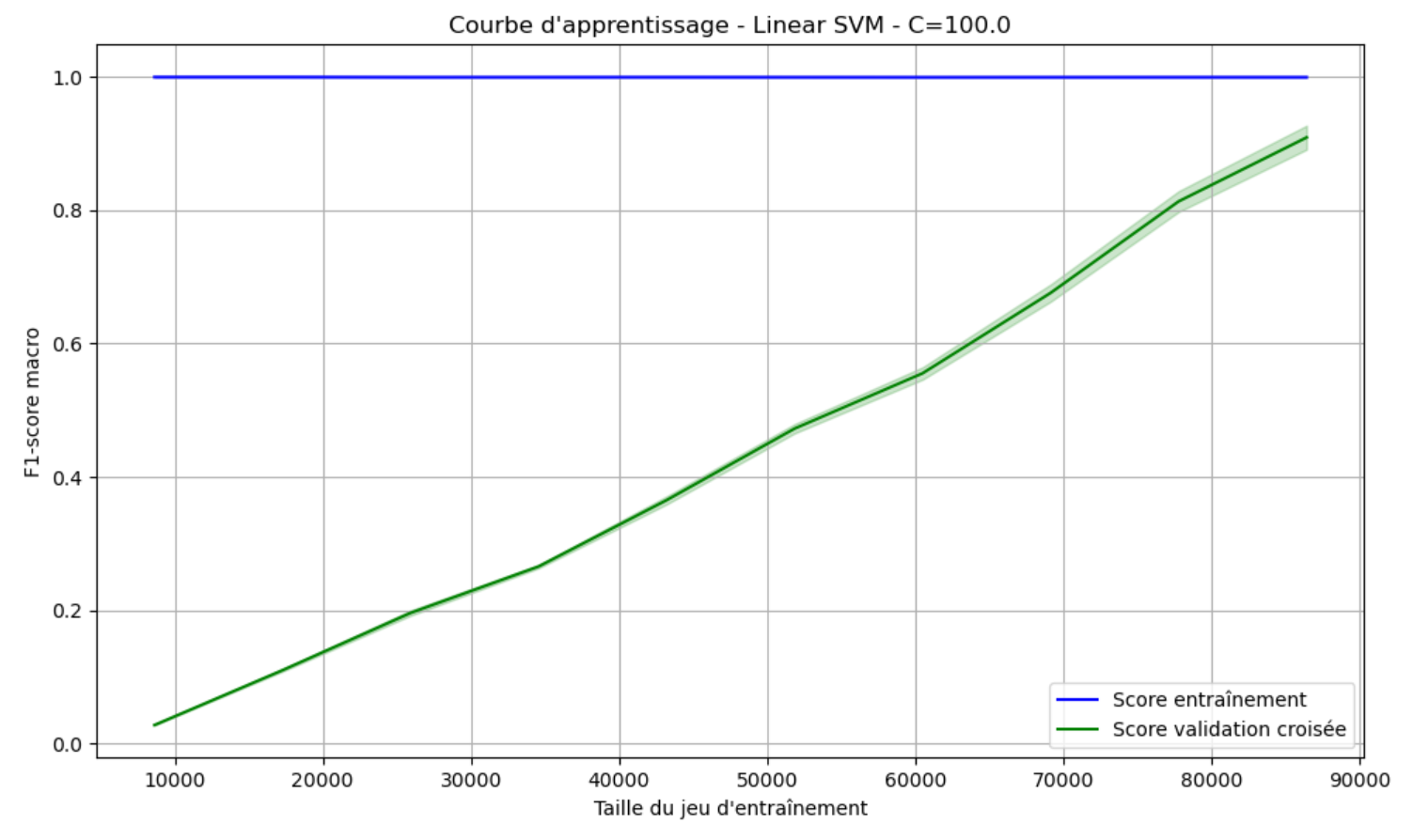
* Avec le paramètre C = 0.1, le poids des erreurs est augmenté. Le score d’entrainement est donc supérieur à celui vu précédemment. Le score d’entraînement débute à 0.95 et termine à 0.92.
* Le score validation croisée démarre bas mais remonte rapidement vers la courbe d’entrainement pour atteindre les 0.87.
* Il reste un léger écart entre le score d’apprentissage et le score de validation



**Analyse de la courbe d’apprentissage pour C = 1**

* Avec le paramètre C =1, le poids des erreurs force un entrainement presque (trop) parfait. Le score d’entrainement est très élevé : 0.99
* Le score validation croisée démarre bas mais remonte rapidement vers la courbe d’entrainement pour atteindre les 0.91
* L’écart entre le score d’apprentissage et le score de validation a augmenté par rapport à la modélisation précédente. On peut conclure à un surapprentissage léger mais le score d’apprentissage reste tout de même meilleur.





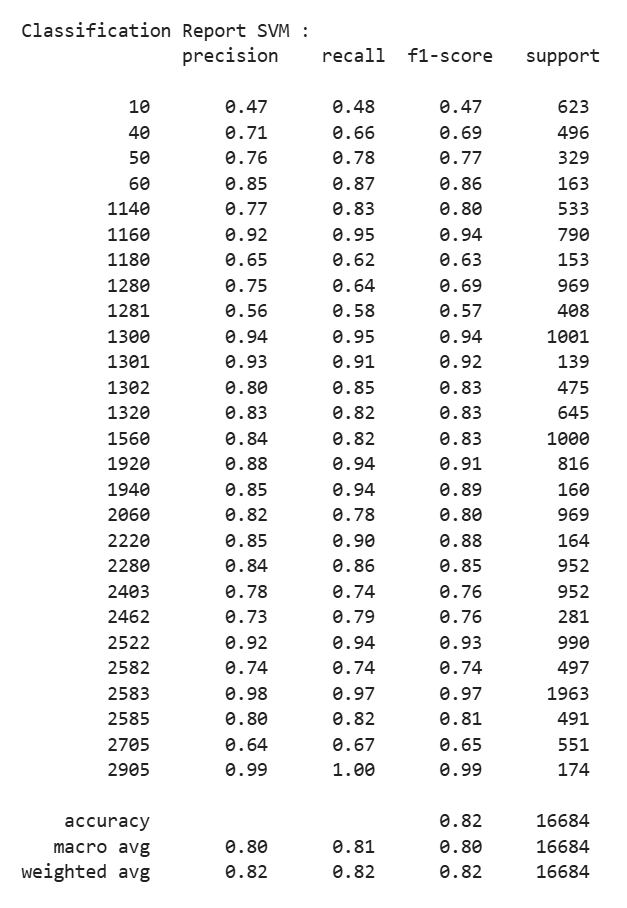
**Analyse de la courbe d’apprentissage pour C = 10 et C = 100**

* Lorsque C =10 ou C = 100, la courbe d’apprentissage est trop parfaite (=1). Il n’y a plus de variation. Ce score trop parfait nous indique qu’il y a un surapprentissage certain.
* Le score de validation est sensiblement le même que pour le modèle C = 1, voir légèrement inférieur pour C = 100.

**Conclusion** : le gain par rapport à C = 1 ne semble pas significatif. Parmi les 5 tests, le modèle avec C = 1 semble être le meilleur compromis.

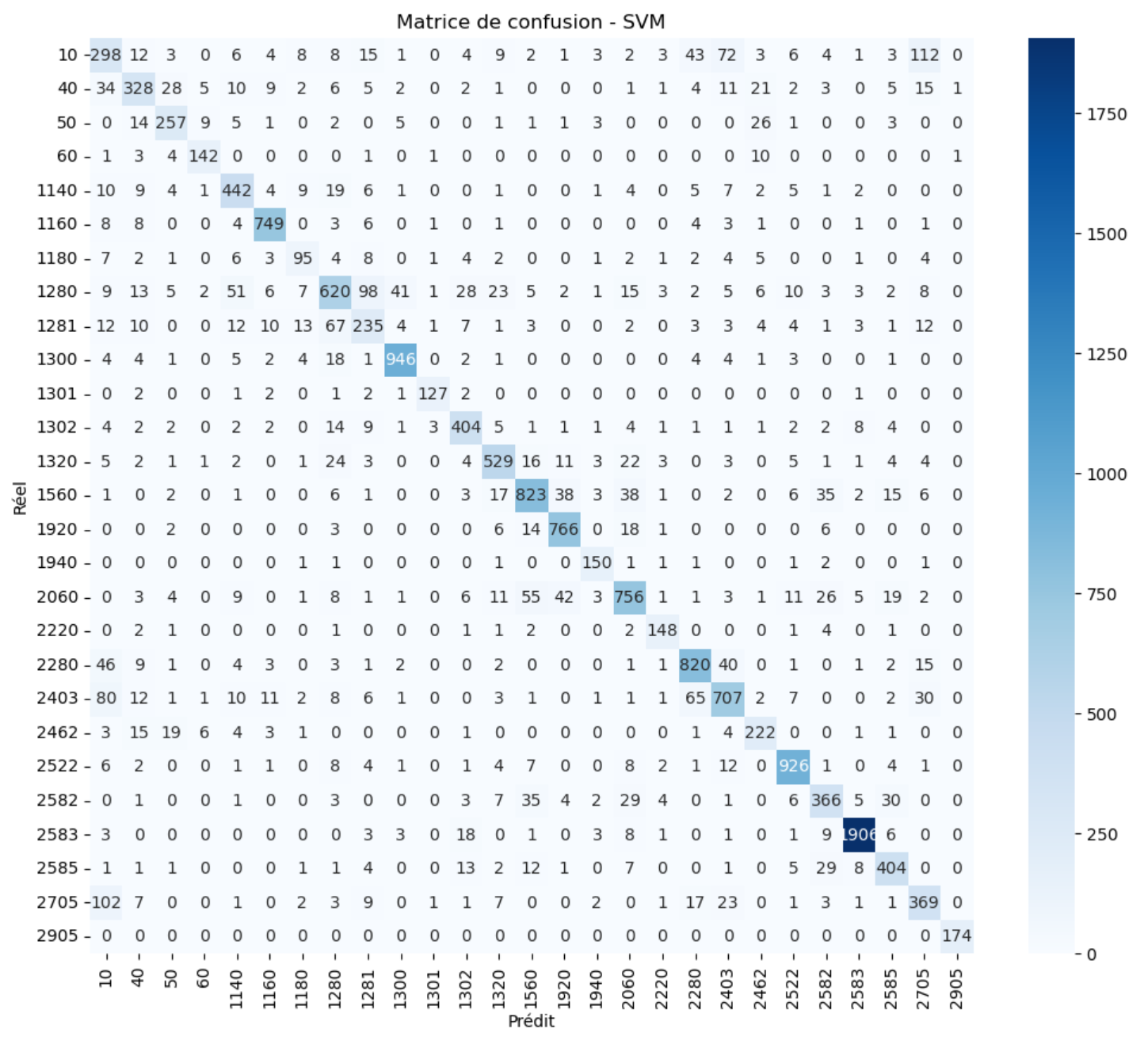
### Rapport de classification

Rapport de classification du modèle SVM avec C = 1



Le F1-score est le meilleur obtenu jusqu’à présent. Il est très bon avec une moyenne de 0.80. Comme pour les autres modèles, il y a toujours des difficultés avec la classe 10 pour laquelle le F1-score n’est que de 0.47.

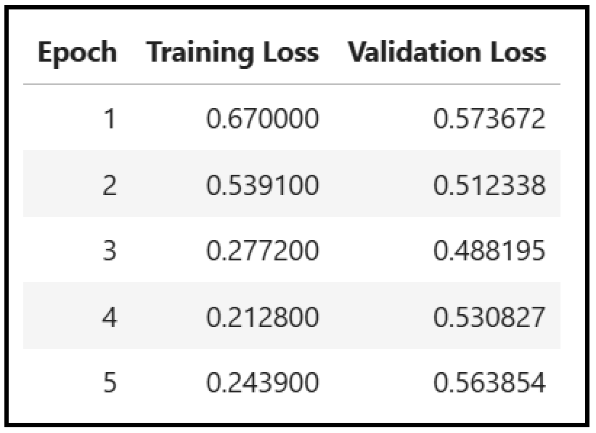
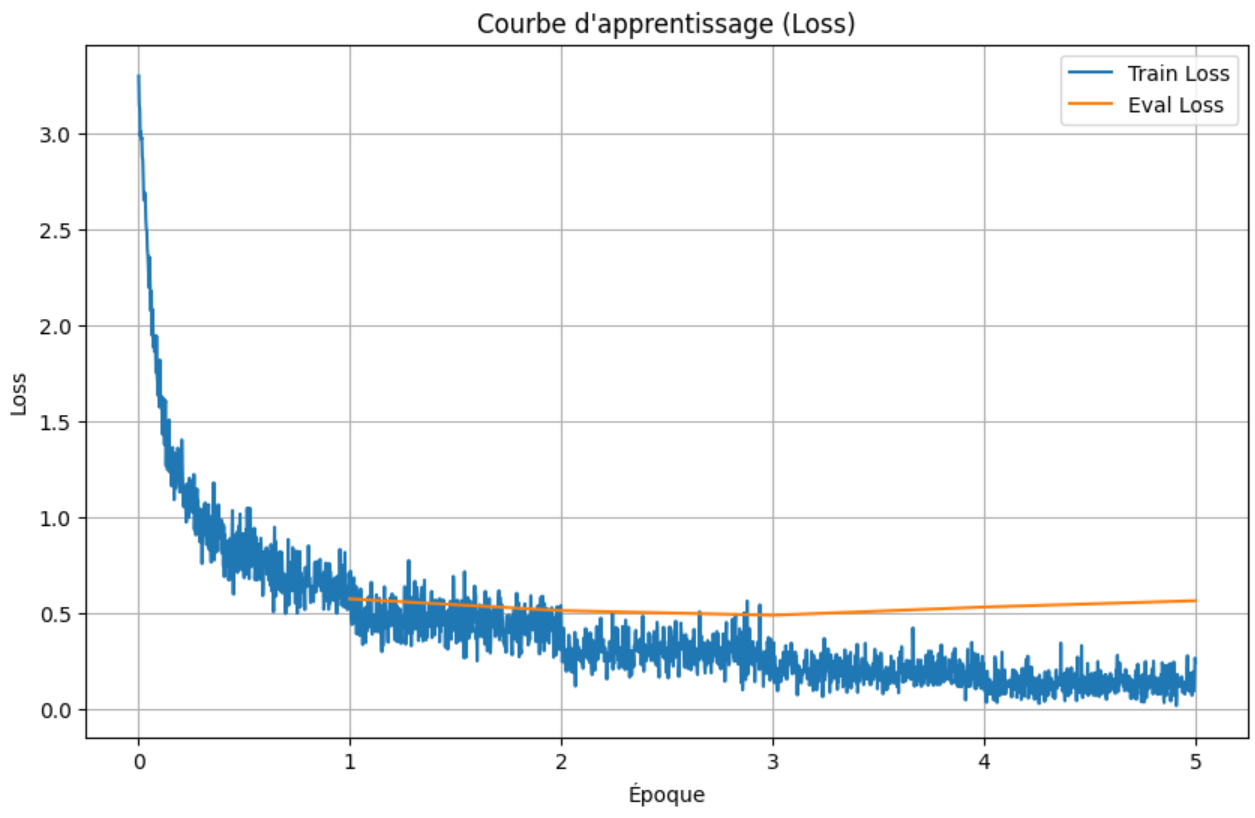
### Matrice de confusion



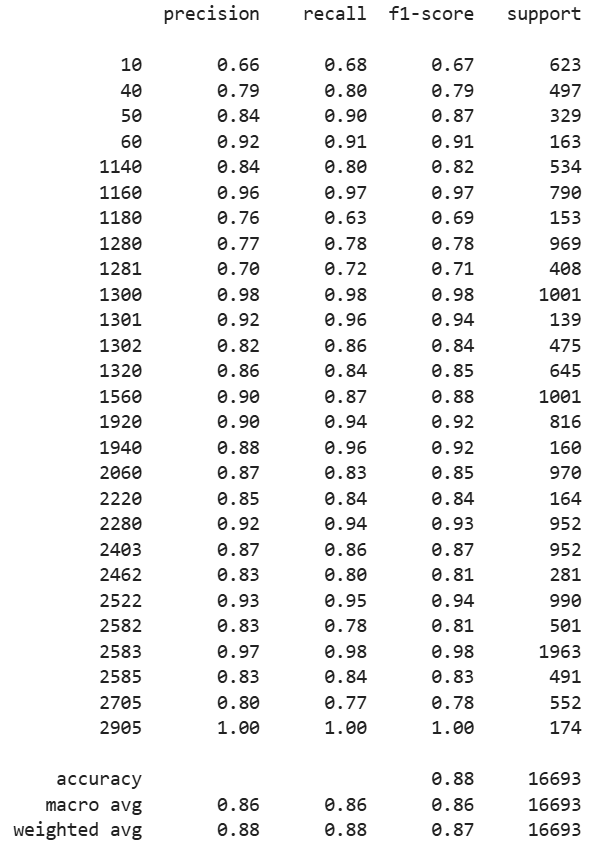
La matrice de confusion nous permet de voir plus en détail les erreurs de prédiction, notamment pour la classe 10 qui pose problème. Le modèle semble avoir du mal à différencier la classe 10 de la classe 2705. En effet, on peut voir que 102 occurrences de la classe 2705 ont été classé à tort dans la classe 10, et que 112 occurrences de la classe 10 ont été classé à tord dans la classes 2705.

# Modèle de BERT

### Courbe de Loss



### Rapport de classification



### Matrice de confusion

