Tâche 1 - Point 3

Nous avons 50000 points de données dans l'ensemble de données "Critiques de films IMDb".

La colonne cible "review" contient 49582 points uniques et aucune valeur NaN ou vide. Cependant nous observons 418 critiques en double.

Tâche 3

La distribution des valeurs cibles sont presque équilibré, avec un peu plus de revue positive que négative:

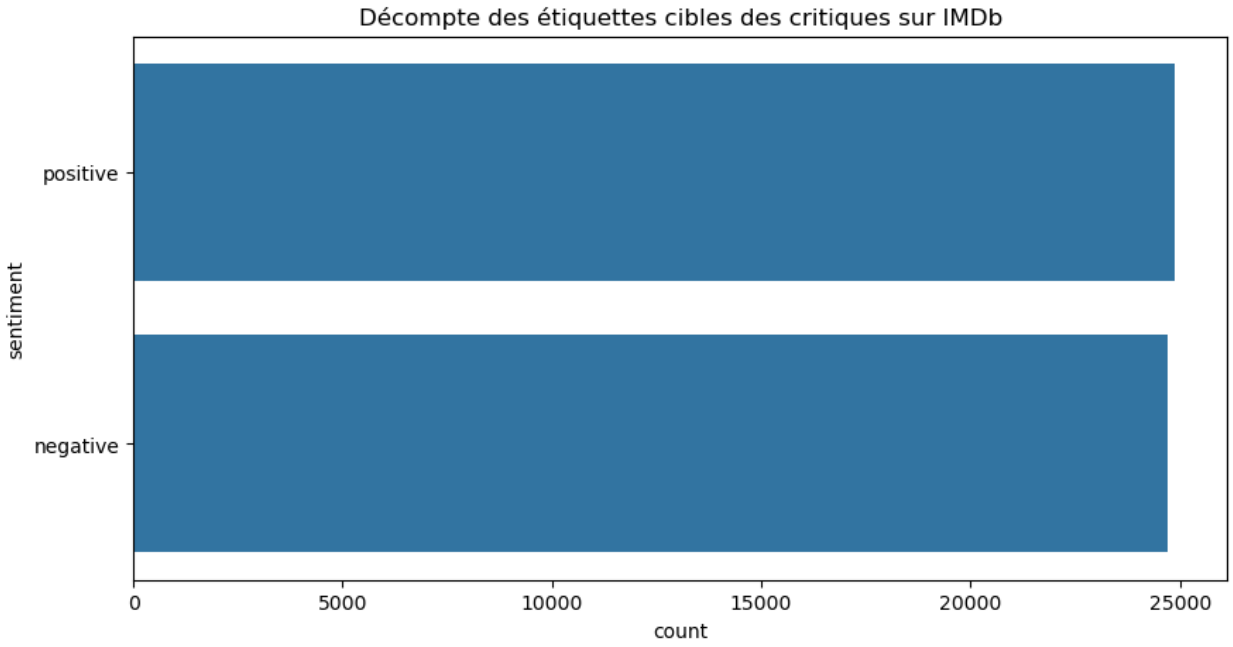


Figure 1: Distribution des valeurs cibles

Comme on le se doute bien, les critiques n'ont pas tous les même longueur:

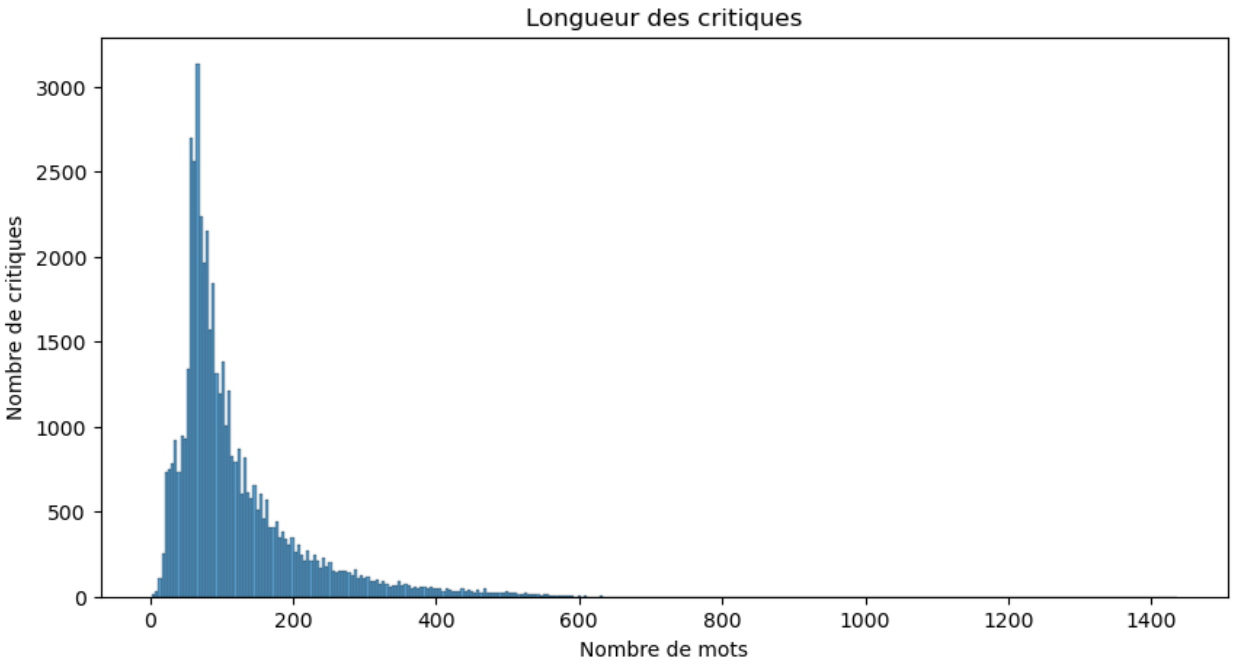


Figure 2: Longueur en mots des critiques.

Il serait d'ailleurs intéressant de voir si en moyenne les critiques négative sont plus longues que les positives et vice versa. La longueur de séquence moyenne est de 119.67 exactement.

Analysons maintenant la fréquence des mots. Les 20 mots les plus fréquents sont donnés dans la figure suivante:

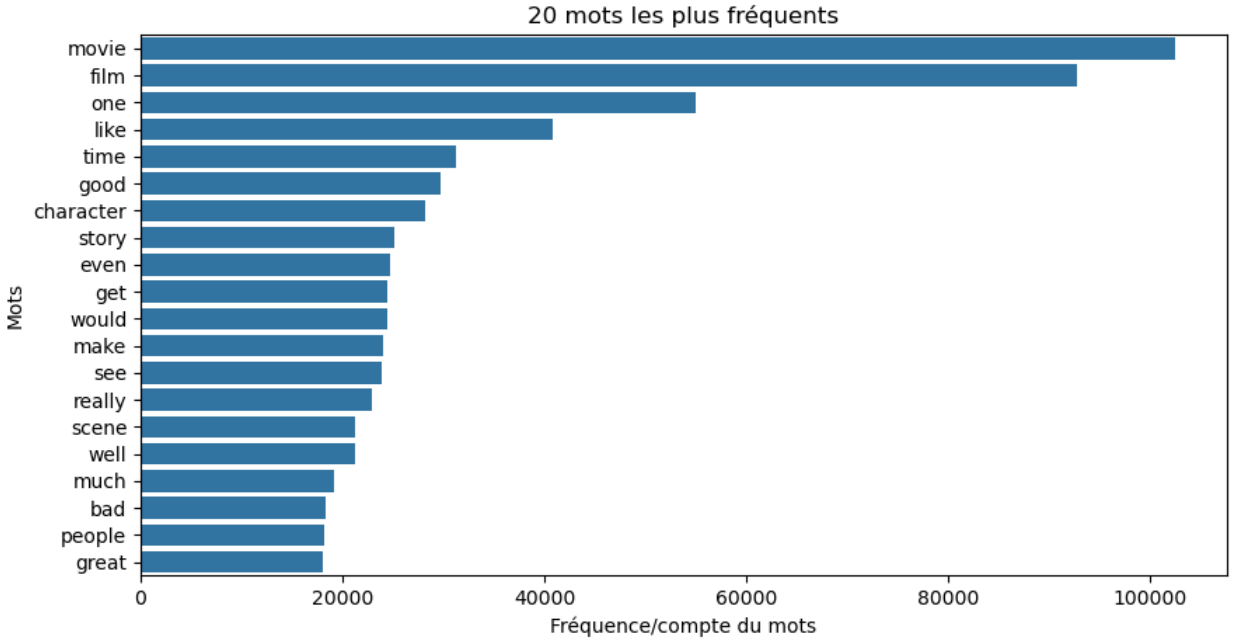


Figure 3: 20 mots les plus fréquents.

Puis, les 20 mots les moins fréquents sont donnés dans la figure suivante. Il faut noter cependant que une fréquence de 1 est le minimum, il y a donc beaucoup de mots à une fréquence de 1. Pour être exact, il y en a 36451!

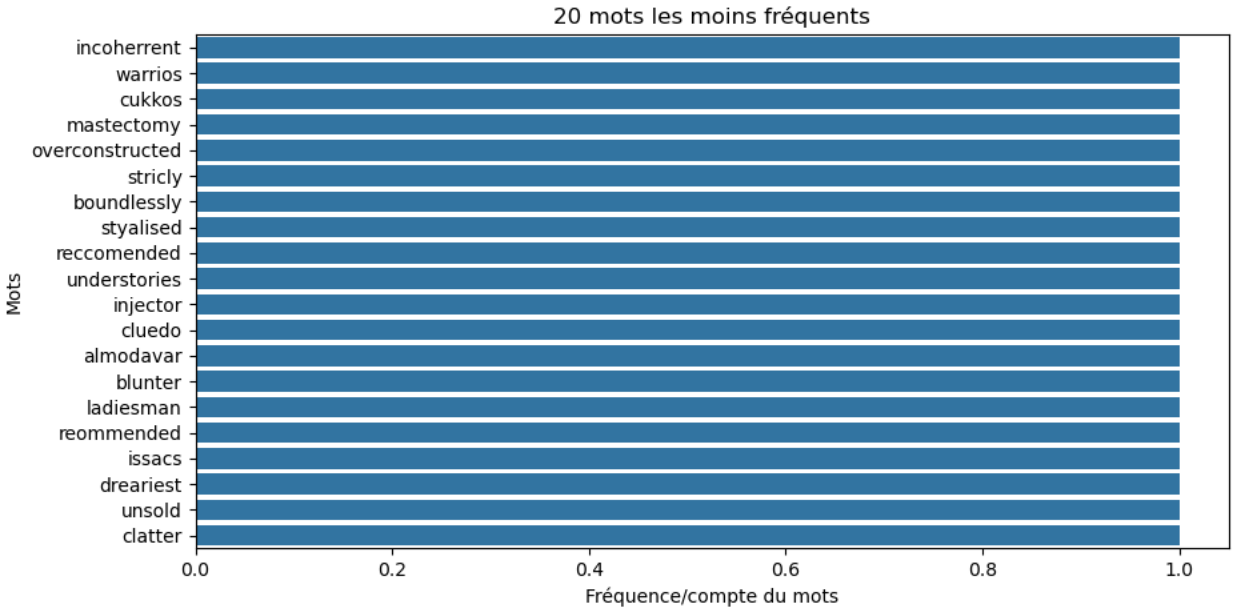


Figure 4: 20 mots les moins fréquents.

La variation considérable dans la longueur des critiques, allant de quelques mots à plus

de 600, pose un défi significatif pour la classification. Les algorithmes de traitement du langage naturel sont souvent conçus pour traiter des entrées de longueur fixe. Les critiques extrêmement courtes peuvent ne pas fournir suffisamment de contexte pour déterminer le sentiment, tandis que les longues peuvent contenir trop de bruit ou d'informations non pertinentes. Ainsi, une standardisation de la longueur des entrées est nécessaire, mais elle peut entraîner la perte d'informations importantes ou l'introduction de données superflues, ce qui complique la tâche de classification.

L'équilibre entre les sentiments positifs et négatifs est un point positif, car il évite à un modèle d'apprendre à favoriser une classe au détriment de l'autre. La présence de 36 451 mots uniques dans les critiques IMDb enrichit le corpus de données mais complique aussi la tâche de classification des sentiments. Car un mot peut changer de signification selon son environnement et cette diversité lexicale peut mener à une sparsité des données, rendant difficile pour les modèles d'apprentissage automatique de déceler des motifs pertinents.

Étant donné que les critiques sont rédigées par des milliers de personnes différentes, l'expressionnisme individuel, les styles d'écriture variés et les différences de vocabulaire ajoutent à la complexité. Un classifiur pourrait avoir du mal à reconnaître les nuances dans l'expression des sentiments de chacun, ce qui pourrait entraîner des erreurs. Par exemple, une utilisation sarcastique de mots généralement positifs d'une certaine personne dans un contexte négatif pourrait tromper un modèle moins sophistiqué.

Il y a un grand risque de biais. Si un film ou une série est largement apprécié, les mots fréquemment associés à celui-ci dans les critiques pourraient être mal interprétés comme toujours positifs, même lorsqu'ils sont utilisés dans des contextes différents. Un classifieur pourrait donc à tort attribuer un sentiment positif à des critiques négatives de films ou de séries différents mais mentionnant des termes similaires. On aurait une sur-représentation de termes liés à des films ou séries populaires qui pourraient fausser les résultats de classification.

Tâche 4

Tf-idf (Term-frequency times inverse document-frequency):

Désavantages de la méthode TF-IDF: TF-IDF traite le texte comme un sac de mots, par conséquent ignore l'ordre et la structure des phrases. Cela peut être problématique pour comprendre le contexte et certaines nuance des critiques. De plus, il pourrait mal interpréter le contexte d'un mot, comme les mots à double sens ou avec des nuances. Certes, la méthode TF-IDF, accorde une plus grande importance aux mots qui sont moins fréquents dans l'ensemble du corpus (ont une haute fréquence inverse de document). Mais lorsqu'on rencontre des mots qui n'étaient pas présents dans le corpus sur lequel le vectorisateur a été ajusté, ces mots inconnus ne seront pas reconnus par le modèle et donc ne contribueront pas à la représentation vectorielle du document. TF-IDF ne gére pas bien les mots qui sont complètement absents de sont corpus et a donc cette limite. Malgrès tout, la performance

de notre modèle de Naive Bayes avec une précision de 87% est commendable, surtout avec une approche générale de TF-IDF. Reste à voir comment le modèle performerait sur des nouvelles revues pas dans notre ensemble de données.

Alternative: Une méthode alternative qu'on a vu en classe, pourrait être le embeddings de mots. Contrairement à TF-IDF qui crée des vecteurs sporadique, les embeddings de mots génèrent des vecteurs denses, capturant plus d'informations dans un espace réduit. Des modèles comme Word2Vec ou GloVe prennent en compte le contexte dans lequel les mots apparaissent, offrant une meilleure compréhension du sens. Ainsi, certains modèles d'embeddings sont capables de gérer les mots inconnus ou rares de manière plus efficace que TF-IDF. Cependant, cette méthode requiert plus de mémoire et de puissance de computation.

LIME (Local Interpretable Model-agnostic Explanations):

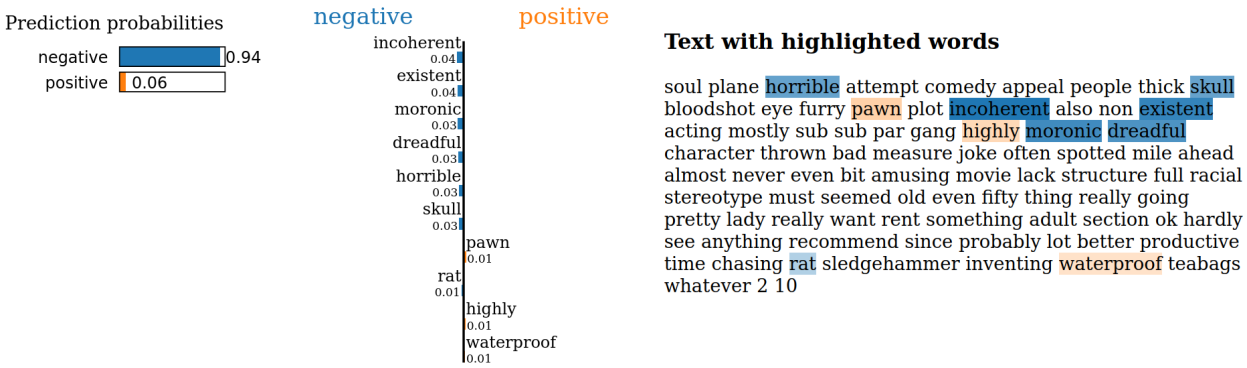


Figure 5: LIME du premier élément dans l'ensemble de test

LIME aide à comprendre sur quelle base le modèle de Naïve Bayes a pris sa décision, en mettant en évidence les mots spécifiques qui ont le plus influencé la prédiction. Cela rend le modèle plus transparent et aide à valider si les caractéristiques utilisées pour la prédiction ont du sens. Le modèle de Naïve Bayes prédit que le texte est de sentiment négatif avec une probabilité de 0.94 (94%) et de sentiment positif avec une probabilité de 0.06 (6%), très confiant que la critique est négative.

A droite on voit les mots du texte qui contribuent le plus à la classification. Les mots sont colorés en bleu ou en orange selon leur influence sur la prédiction négative ou positive, respectivement. Des mots comme "horrible", "incoherent","moronic", et "dreadful" sont fortement associés à une connotation négative, comme indiqué par leur coloration en bleu et leur poids affiché au milieu. Il semble y avoir peu (ou plutôt aucun mais le modèle croit, on en discutera dans un instant) de mots contribuant significativement à une connotation positive, ce qui est cohérent avec la faible probabilité de prédiction positive.

L'analyse LIME révèle plusieurs points intéressants sur la façon dont le modèle interprète le corpus à l'aide de TF-IDF. Il est notable que certains mots typiquement négatifs, tels que "bad", manquent à l'appel, ce qui pourrait indiquer une limite dans la capture de toutes les

nuances négatives par le modèle. Les mots identifiés comme ayant une connotation positive, comme "pawn" et "waterproof", sont surprenants et semblent être une identification incorrecte, suggérant que le modèle pourrait mal interpréter des mots neutres ou non pertinents comme étant positifs. Bien que les mots négatifs soient un peu mieux représentés, la présence de mots comme "existent", qui n'ont pas une connotation clairement négative sans le "non", indique également une certaine confusion du modèle. Malgré ces anomalies, le modèle a correctement capté que le commentaire était majoritairement négatif, ce qui est rassurant quant à sa performance globale. Cependant, il est évident que le modèle pourrait mal classifier des mots spécifiques au film, les considérant à tort comme négatifs ou positifs, en l'absence de contexte suffisant pour ces mots. Cela souligne l'importance de prendre en compte le contexte complet lors de l'analyse des sentiments, une complexité que le vectorization actuel peut ne pas saisir pleinement.

Tâche 5

2 -

Pour améliorer les performances de classification du modèle, j'ai suivi une approche méthodique, en commençant par définir le nombre d'époques à 2 et la taille du lot à 16.

Tout d'abord, j'ai utilisé l'optimiseur par défaut de Trainer des Transformers: AdamW, sans spécifier de weight_decay, qui par défaut est de 0. J'ai fixé le taux d'apprentissage à 0.0001, une valeur couramment utilisée. Observant que la perte d'entraînement du modèle était dans tous les sens durant les 2 epochs et n'avait pas de tendence de convergence. J'ai donc décidé d'ajouter un weight_decay à 0.001 pour mon deuxième essai. Les résultats étaient bien meilleurs, avec une perte d'entraînement descendant jusqu'à 0.1054, mais avec beaucoup de fluctuations, indiquant une oscillation au-dessus du minimum. Pour vérifier s'il y avait du surapprentissage, j'ai évalué le modèle sur le jeu de données de test, obtenant une précision de 0.90 et confirmant que non.

En conséquence, pour réduire l'oscillation, j'ai créé un nouveau modèle avec un taux d'apprentissage de 0.00005 et le même weight_decay. Cela a significativement réduit l'oscillation de la perte d'entraînement, bien qu'elle soit toujours présente, et nous avons observé une tendance à la baisse. Pour pousser encore plus loin et réduire complètement l'oscillation, j'ai ajouté un warmup de 10% des pas totaux de l'entraînement, baissé légèrement le taux d'apprentissage à 0.000025, intégré un gradient clipping avec 'max_grad_norm=1.0' et augmenté la taille du lot à 32 pour lisser le processus. Nous avons obtenu une courbe de perte d'entraînement stable pendant le warmup, indiquant que le taux d'apprentissage devrait être encore plus bas car nous avons observé quelques petites oscillations plus tard. Cependant, les résultats étaient satisfaisants avec une perte d'évaluation de 0.24 et une précision d'évaluation de 0.915. Poursuivre dans la quête d'une amélioration aurait été une utilisation excessive des ressources.

3 -

Batch	Learning Rate	Decay	Clipping	Loss	Accuracy	Precision	Recall	F1
16	0.0001	None	No	0.2793	0.8986	0.8914	0.9086	0.8999
16	0.0001	0.001	No	0.2808	0.9004	0.8878	0.9174	0.9024
16	0.00005	0.001	No	0.2900	0.9145	0.9114	0.9190	0.9152
32	0.000025	0.001	Yes	0.2389	0.9147	0.9083	0.9233	0.9157

Table 1: Comparaison de la performance sur les données de test (évaluation) des modèles à travers les différentes expériences

Accuracy	Precision	Recall	F1
0.866	0.87	0.87	0.87

Table 2: Performance du modèle de Naive Bayes

Comme observé, les expériences montrent une progression des indicateurs de performance à mesure que divers paramètres sont ajustés :

Expérience 1 : C'était la base avec des paramètres standards. Bien qu'elle ait montré des performances décentes et meilleure que le modèle de Naive Bayes, il y avait une petite marge d'amélioration.
Expérience 2 : L'introduction d'une décroissance de poids (weight decay) a légèrement amélioré la précision et le score F1, suggérant que cette régularisation a été quelque peu bénéfique.
Expérience 3 : La diminution du taux d'apprentissage a conduit à une amélioration notable de tous les indicateurs, soulignant l'importance de ce paramètre dans l'entraînement du modèle.
Expérience 4 : La dernière expérience, avec une réduction supplémentaire du taux d'apprentissage, une augmentation de la taille du lot et l'introduction du clipping de gradient, a obtenu la meilleure perte d'évaluation et maintenu des indicateurs de performance élevés. Cela suggère un processus d'entraînement plus stable et potentiellement une meilleure généralisation.

Il est évident que les ajustements apportés au taux d'apprentissage, à la taille du lot et l'ajout du clipping de gradient ont contribué positivement à la performance du modèle. L'amélioration constante de la précision et du score F1 à travers les expériences, en particulier dans la dernière, indique que la capacité du modèle à généraliser s'est améliorée avec ces ajustements de paramètres. Cependant on constate que l'amélioration ralenti et que le retour devient moins signifiant surtout pour le temps d'entraînement d'un transformeur.

Les résultats obtenus à partir des différentes configurations du modèle DistilBERT montrent une amélioration significative par rapport aux performances du classifieur Naive Bayes . En particulier, les modèles avec DistilBERT ont démontré une augmentation notable de la précision, du rappel et du score F1. Dans l'expérience 4, le modèle a atteint une précision de

0.9083, un rappel de 0.9233 et un score F1 de 0.9157. En comparaison, le classifieur Naïve Bayes a obtenu des scores globalement inférieurs, avec une précision, un rappel et un score F1 de 0.87 chacun. Même le modèle le plus simple du transformeur DistilBERT, est meilleur de 0.03 dans toutes les metriques par rapport au Naive Bayes.

Ces différences de performance peuvent être attribuées à plusieurs facteurs. D'abord, le modèle DistilBERT, en tant que variante allégée de BERT, est spécifiquement conçu pour comprendre le contexte et la nuance du langage, ce qui le rend plus adapté pour des tâches complexes de traitement de langage naturel telles que notre analyse de sentiment revues. En revanche, bien que le classifieur Naïve Bayes soit moins performant pour les tâches nécessitant une compréhension contextuelle profonde du texte, sont temps d'entraînement pour sa performance est impressionnant.

Passons aux matrice de confusion:

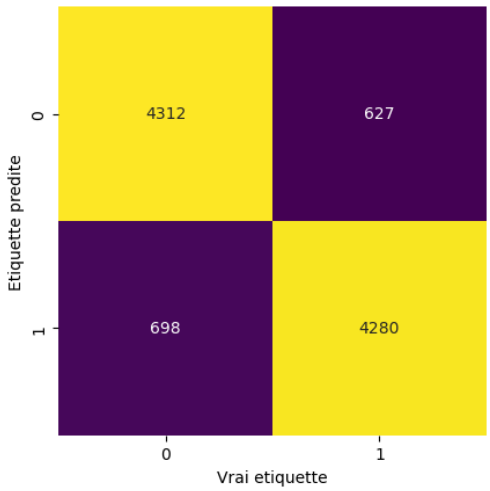
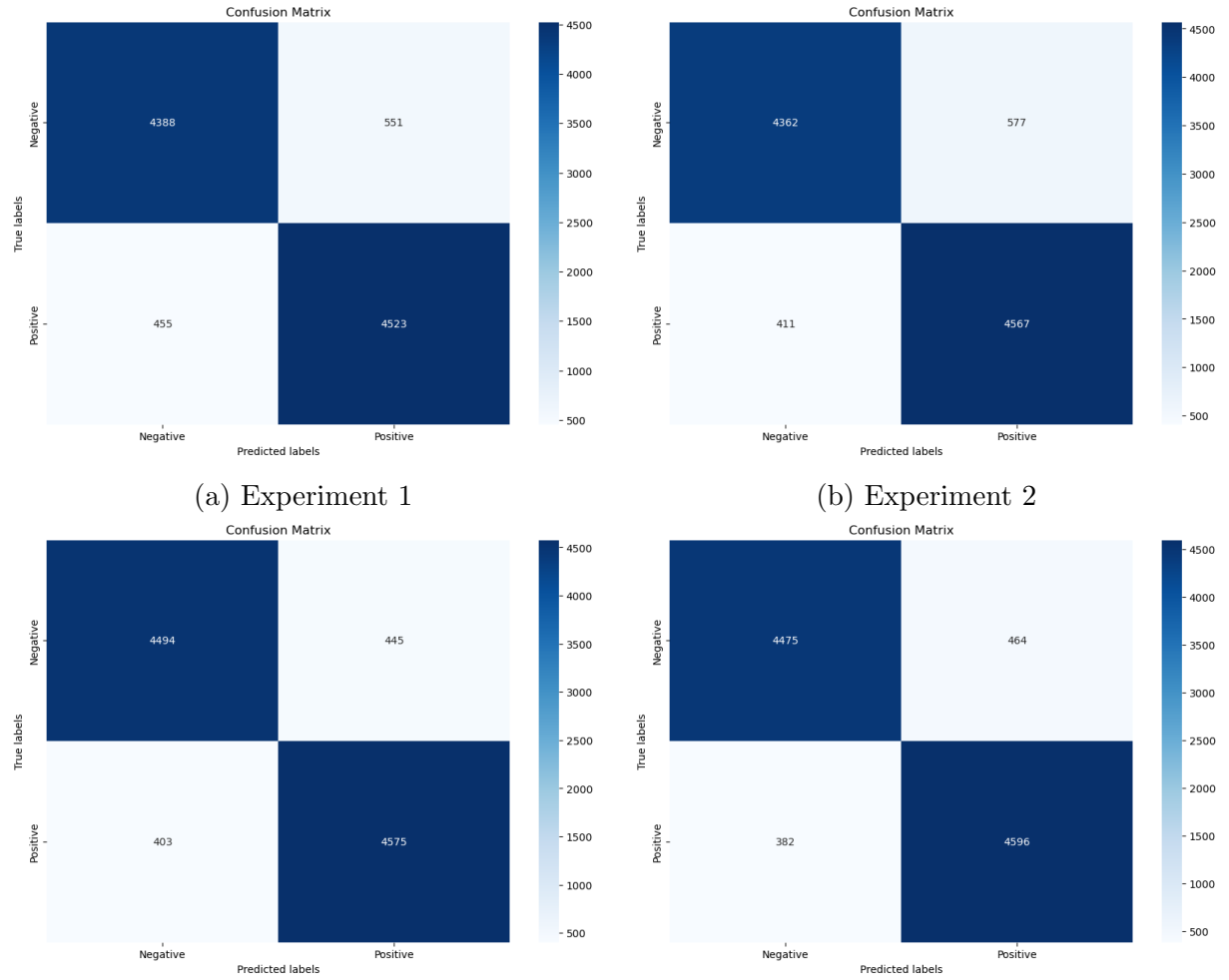


Figure 6: Matrices de confusions pour le classifieur de Naive Bayes



(a) Experiment 1

(b) Experiment 2

(c) Experiment 3

(d) Experiment 4

Figure 7: Matrices de confusions pour les 4 expériences de distilBERT

Nous allons seulement considérer notre meilleur modèle, donc l'experience 4 soit la matrice (d).

La matrice de confusion pour le modèle DistilBERT montre un nombre élevé de vrais positifs (4596) et de vrais négatifs (4475), avec un nombre relativement faible de faux négatifs (382) et de faux positifs (464). Le modèle DistilBERT a réussi à classer correctement la majorité des évaluations, aussi bien pour la classe positive que pour la classe négative. Il a une bonne capacité de généralisation et une forte performance globale sur l'ensemble de données. Le petit taux d'erreur peut-être lié à des revues mal écrite comme nous avions vu dans la partie 4 de la tâche 4.

D'autre part, la matrice de confusion pour le modèle Naïve Bayes montre également un grand nombre de vrais positifs (4280) et de vrais négatifs (4312), mais avec une proportion plus élevée de faux positifs (627) et de faux négatifs (698) par rapport au modèle DistilBERT. Bien que la performance soit relativement bonne, nous avons tout de même une fréquence

plus grande de classifications incorrectes. Il est moins précis que le modèle DistilBERT, surtout lorsqu'il s'agit de différencier finement entre les classes positives et négatives.

Le modèle DistilBERT non seulement produit moins d'erreurs de classification en général, mais il montre également une meilleure capacité à distinguer les deux classes. Ça confirme donc l'hypothèse que les modèles de Transformeurs, sont particulièrement bien adaptés aux tâches complexes de corpus avec des nuances et contextualisation, tendent à surpasser les approches classiques comme le Naïve Bayes, qui reposent sur des hypothèses simplifiées de distribution des caractéristiques et d'indépendance.

En conclusion, l'analyse comparative entre le classifieur Naïve Bayes et les différentes configurations du modèle DistilBERT démontre clairement la supériorité des approches basées sur les transformateurs pour la tâche d'analyse de sentiment. Les ajustements méthodiques des hyperparamètres du modèle DistilBERT ont entraîné des améliorations substantielles des performances, se traduisant par une précision, un rappel et un score F1 plus élevés par rapport au classifieur Naïve Bayes. Mais il ne faut pas oublier que les transformeurs sont lourd et prennes beaucoup de temps/ressources, il est donc crucial de trouver un bon compromis entre la performance souhaitée et les ressources disponibles.