

Rapport d'évaluation

Table des matières

1. Matrice de confusion.....	2
a. Les prédictions positives - interprétation.....	2
b. Les prédictions négatives – interprétation.....	2
2. Calcul et interprétation des mesures	3
a. Cas positif	3
b. Cas négatif	3
3. Analyses des performances	4
a. Forces	4
b. Faiblesses	4
c. Biais	5
4. Recommandations claires pour améliorer le modèle	5

1. Matrice de confusion

a. Les prédictions positives - interprétation

```
(.env) PS C:\Users\UltraBook 3.1\OneDrive - Efrei\Documents\M2\Algo\SocialMetric_AI> python positive_training.py
Rapport de classification :
      precision    recall  f1-score   support

     0       0.90      0.69      0.78        13
     1       0.67      0.89      0.76         9

   accuracy          0.77        22
  macro avg          0.78        22
 weighted avg          0.80        22

Matrice de confusion :
[[9 4]
 [1 8]]
```

- 9 vrais négatifs : le modèle a correctement classé 9 commentaires négatifs
- 8 vrais positifs : le modèle a correctement classé 8 commentaires positifs
- 4 faux positifs : le modèle a classé 4 commentaires négatifs comme positifs
- 1 faux négatifs : le modèle a classé 1 commentaire positif comme négatifs

Calcul du raciaux

Vrai négatif = $9/13 \times 100 = 67\%$

Vrai positif = $8/9 \times 100 = 88$

Conclusion

Le modèle semble bien identifier les commentaires positifs (TP = 8, FN=1). Il a un taux d'erreur plus élevé sur les négatifs (FP =4), donc il confond certains commentaires négatifs avec des positifs.

b. Les prédictions négatives – interprétation

```
(.env) PS C:\Users\UltraBook 3.1\OneDrive - Efrei\Documents\M2\Algo\SocialMetric_AI> python negative_training.py
Rapport de classification :
      precision    recall  f1-score   support

     0       0.67      0.89      0.76         9
     1       0.90      0.69      0.78        13

   accuracy          0.77        22
  macro avg          0.78        22
 weighted avg          0.80        22

Matrice de confusion :
[[8 1]
 [4 9]]
```

- 8 vrais négatifs : le modèle a classé 8 commentaires négatifs
- 9 vrais positifs : le modèle a classé 9 commentaires positifs
- 1 faux positif : le modèle a classé 1 commentaire négatif comme positif

- 4 faux négatifs : le modèle a classé 4 commentaires positifs comme négatif

Calcul du raciaux

Vrai négatif = $8/9 \times 100 = 88\%$

Vrai positif = $4/13 \times 100 = 30\%$

Conclusion

Le modèle semble bien identifier les commentaires positifs (TP = 4, FN = 4). Il a un taux d'erreur moins élevé sur les négatifs (FP = 1).

2. Calcul et interprétation des mesures

a. Cas positif

Rapport de classification :

	precision	recall	f1-score	support
0	0.90	0.69	0.78	13
1	0.67	0.89	0.76	9

- **Précision**

Pour les négatifs (0) : 0.90 → lorsque le modèle prédit un commentaire négatif, il est correct 90%.
 Pour les positifs (1) : 0.67 → lorsque le modèle prédit un commentaire positif, il est correct à 67%.

- **Recall**

Pour les négatives (0) : 0.69 → 69% des commentaires négatifs sont bien détectés.
 Pour les positifs (1) : 0.89% → 89% des commentaires positifs sont bien détectés.

- **F1-score**

Sachant que le f1-score du cas négatifs est de 0.78 et celui du cas positif est 0.76, on peut dire que le model est performant. Le model détecte mieux les cas négatifs que les cas négatifs.

b. Cas négatif

	precision	recall	f1-score	support
0	0.67	0.89	0.76	9
1	0.90	0.69	0.78	13

- **Précision**

0(négatif) : 0.67 → 67% des prédictions négatives sont correctes

1 (positif) : 0.90 → 90% des prédictions positives sont correctes

- **Recall**

Pour les négatives (0) : 0.89 → 89% des vrais commentaires négatifs sont bien détectés

Pour les positifs (1) : 0.69% → 69% des vrais commentaires positifs sont bien détectés

- **F1-score**

Sachant que le f1-score du cas négatifs est de 0.76 et celui du cas positif est 0.78, on peut dire que le model est performant. Le model détecte mieux les cas positifs que les cas négatifs

3. Analyses des performances

a. Forces

Bonne précision pour les commentaires négatifs (0.90)

- ➔ Lorsqu'il prédit un commentaire comme négatif, il a 90% de chances d'avoir raison.
- ➔ Cela signifie qu'il fait peu d'erreurs en attribuant des commentaires négatifs.
- ➔ Il serait donc plus facile dans le système de détecter les commentaires négatifs et de les flouter par exemples.

Bon recall pour les commentaires positifs (0.89)

- ➔ 89% des vrais commentaires positifs sont bien détectés.
- ➔ Ceci signifie que le modèle capture presque tous les commentaires positifs, ce qui est utile si l'on veut éviter de manquer des retours positifs.

Bon équilibre entre précision et recall (F1-score sensiblement égale à 0.77)

- ➔ Avec 0.78 pour les positifs et 0.76 pour les négatifs, le modèle montre la performance globale.
- ➔ Il est capable de détecter à la fois les commentaires négatifs et positifs sans un biais trop fort en faveur de l'un ou de l'autre.

b. Faiblesses

Mauvaise précision pour les commentaires positifs

- ➔ Lorsqu'il prédit un commentaire comme positif, il se trompe dans 33% des cas.
- ➔ Il classe certains commentaires négatifs à tort comme positifs (4 faux positifs).
- ➔ Peut poser problème si l'on veut éviter de promouvoir des avis négatifs.

Faible recall pour les commentaires négatif (0.69)

- ➔ 31% des commentaires négatifs ne sont pas détectés et sont classés comme positifs.
- ➔ Si le but est d'éviter les retours négatifs, ce recall n'est pas optimal.

Tendances à sous-estimer les commentaires négatifs

- ➔ Le modèle classe trop de négatifs comme positifs (FP=4)
- ➔ Si le but est de filtrer correctement les commentaires négatifs, cette faiblesse peut l'affecter.

c. Biais

Influence du langage et du contexte

- ➔ Ironie et sarcasme
Exemple : "Super service ! J'ai attendu 2 heures pour une réponse..."
Ce type de commentaire peut être mal classé comme positif.
- ➔ Langage informel, emojis et variations linguistiques peuvent aussi poser problème.

Biais des données

- ➔ La distribution des commentaires dans l'ensemble d'apprentissage peut influencer les performances.
- ➔ Si les commentaires positifs sont sous-représentés, le modèle peut être moins performant pour les identifier correctement.

Biais dans le seuil de décision

- ➔ Un ajustement du seuil de classification pourrait améliorer la détection des commentaires négatifs ou positifs selon l'objectif.
- ➔ Baisser le seuil pour les positifs améliorerait le rappel mais pourrait augmenter les faux positifs.
- ➔ Augmenter le seuil pour les négatifs pourrait aider à mieux détecter les retours négatifs.

4. Recommandations claires pour améliorer le modèle

Amélioration des données d'entraînement

✓ Équilibrer le dataset

Vérifier si le dataset contient autant de tweets positifs que négatifs.
Si un déséquilibre existe, utiliser des techniques comme :

- Oversampling (ajouter des exemples de la classe sous-représentée).
- Undersampling (réduire le nombre d'exemples de la classe dominante).

✓ Améliorer la diversité des données

Ajouter des tweets contenant du sarcasme, des emojis et des expressions informelles pour mieux capter le contexte linguistique.

Vérifier que les différents styles d'écriture (langage formel/informel, fautes d'orthographe) sont bien représentés.

✓ **Nettoyage et prétraitement avancé**

Tester différentes méthodes de tokenization (ex. WordPiece, Byte-Pair Encoding) pour mieux représenter le texte.

Ajouter la prise en compte des emojis et du contexte (ex. 😡 souvent associé à un tweet négatif).

Optimisation du modèle

✓ **Tester un modèle NLP plus avancé**

- Remplacer la régression logistique par un modèle basé sur du deep learning comme :
 - BERT (Bidirectional Encoder Representations from Transformers)
 - RoBERTa
 - XLNet
- Ces modèles comprennent mieux le contexte des phrases et améliorent la détection du sarcasme et des ambiguïtés.

✓ **Ajuster le seuil de classification**

- Actuellement, le modèle produit trop de faux positifs (4).
- Tester différents seuils (ex. 0.6 au lieu de 0.5) pour réduire les erreurs de classification des tweets négatifs en positifs.

✓ **Expérimenter avec d'autres techniques de pondération**

- Tester des métriques alternatives comme le balanced accuracy ou focal loss pour mieux gérer le déséquilibre des classes.