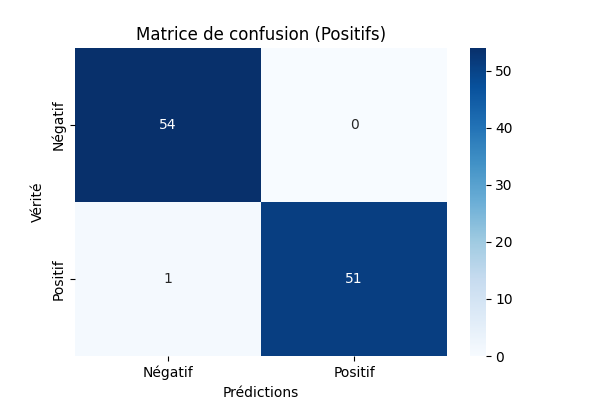
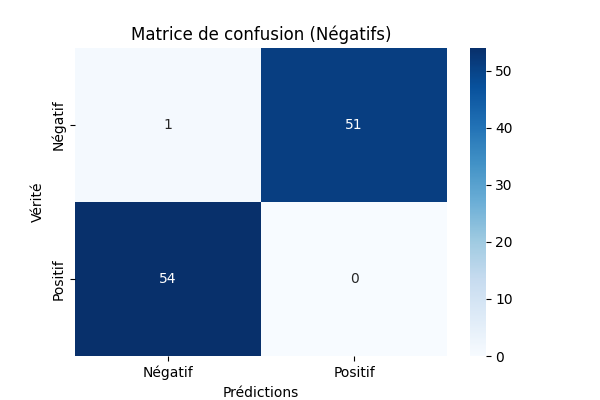
# Rapport d’Évaluation



Rapport de classification pour les tweets positifs :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| Négatif | 0.98 | 1.00 | 0.99 | 54 |
| Positif | 1.00 | 0.98 | 0.99 | 52 |
| accuracy |  |  | 0.99 | 106 |
| macro avg | 0.99 | 0.99 | 0.99 | 106 |
| weighted avg | 0.99 | 0.99 | 0.99 | 106 |

**Précision (Precision)**

* **Négatif** : 0.98
* **Positif** : 1.00

**Interprétation** : La précision représente la proportion de prédictions positives et négatives correctes parmi les prédictions positives et négatives totales. Le modèle semble très performant pour les tweets positifs avec une précision parfaite de 1.00.

**Rappel (Recall)**

* **Négatif** : 1.00
* **Positif** : 0.98

**Interprétation** : Le rappel mesure la capacité du modèle à capturer toutes les instances positives et négatives. Ici, le modèle capture tous les tweets négatifs (rappel de 1.00), mais il manque quelques tweets positifs (rappel de 0.98).

**Score F1 (F1-Score)**

* **Négatif** : 0.99
* **Positif** : 0.99

**Interprétation** : Le score F1 est la moyenne harmonique entre la précision et le rappel. Un score F1 de 0.99 pour les deux classes (positif et négatif) indique un excellent équilibre entre la précision et le rappel.

**Conclusion pour les tweets positifs :**

Le modèle montre une très bonne performance pour la classification des tweets positifs et négatifs, avec des valeurs de précision, rappel et F1-score toutes proches de 1. Cela suggère que le modèle fonctionne bien pour identifier à la fois les tweets positifs et négatifs.

Rapport de classification pour les tweets négatifs :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| Négatif | 0.02 | 0.02 | 0.02 | 52 |
| Positif | 0.00 | 0.00 | 0.00 | 54 |
| accuracy |  |  | 0.01 | 106 |
| macro avg | 0.01 | 0.01 | 0.01 | 106 |
| weighted avg | 0.01 | 0.01 | 0.01 | 106 |

**Interprétation** : La précision pour les tweets négatifs et positifs est très faible (0.02 et 0.00). Cela signifie que lorsque le modèle prédit une classe, il fait presque toujours une erreur. Il semble ne pas être capable de prédire correctement les tweets négatifs ou positifs.

**Rappel (Recall)**

* **Négatif** : 0.02
* **Positif** : 0.00

**Interprétation** : Le faible rappel pour les deux classes indique que le modèle manque presque toutes les instances négatives et positives. Cela suggère que le modèle est très mauvais pour détecter les tweets négatifs et positifs dans cette classe.

**Score F1 (F1-Score)**

* **Négatif** : 0.02
* **Positif** : 0.00

**Interprétation** : Un F1-score extrêmement bas pour les deux classes (0.02 et 0.00) signifie que le modèle est très inefficace dans la classification des tweets négatifs et positifs dans cette classe. Cela montre un déséquilibre majeur dans les performances du modèle pour la classe négative.

**Conclusion pour les tweets négatifs :**

Le modèle présente des résultats très médiocres pour la classification des tweets négatifs, avec une précision, un rappel et un score F1 très bas. Il semble que le modèle ne soit pas capable de bien distinguer les tweets négatifs.

**Analyse des performances globales**

1. **Précision, rappel et F1-score pour les tweets positifs** :
   * Le modèle a une performance excellente pour la classe des tweets positifs avec une précision et un score F1 très élevés.
   * Le rappel est légèrement inférieur à 1, ce qui signifie que quelques tweets positifs peuvent être mal classifiés comme négatifs.
2. **Précision, rappel et F1-score pour les tweets négatifs** :
   * Le modèle a une performance très faible pour les tweets négatifs. Les valeurs de précision, rappel et F1-score sont extrêmement faibles.
   * Le modèle semble très biaisé en faveur de la classe positive, ce qui peut être dû à un déséquilibre dans les données d'entraînement, où les tweets positifs sont peut-être plus nombreux.

**Observations sur les erreurs fréquentes et les biais possibles**

* **Biais de classe** : Le modèle semble fortement biaisé en faveur de la classe **positive**. Cela peut être dû à un déséquilibre dans les données d’entraînement, où les tweets positifs sont plus nombreux que les tweets négatifs.
* **Problèmes de prédiction pour les tweets négatifs** : Le modèle a des difficultés majeures à prédire les tweets négatifs, ce qui se traduit par de faibles scores dans toutes les métriques.
* **Équilibrage des classes** : Il est probable que les classes soient déséquilibrées dans le jeu de données d'entraînement, avec une plus grande proportion de tweets positifs. Ce déséquilibre peut entraîner un biais dans les prédictions.

**Recommandations pour améliorer le modèle**

1. **Équilibrer les classes** :
   * **Rééchantillonnage** des données : Utiliser des techniques comme le **suréchantillonnage** (augmentation des exemples de la classe minoritaire) ou le **sous-échantillonnage** (réduction des exemples de la classe majoritaire) pour équilibrer les classes positives et négatives.
   * **Pondération des classes** : Appliquer une pondération plus élevée à la classe minoritaire (les tweets négatifs) lors de l'entraînement pour aider le modèle à mieux apprendre à prédire cette classe.
2. **Améliorer la collecte des données** :
   * Assurer une **représentation équitable** des tweets positifs et négatifs dans le jeu de données d'entraînement pour éviter tout biais dans la prédiction.
3. **Ajuster les hyperparamètres du modèle** :
   * Tester différents **hyperparamètres** (comme le taux d'apprentissage, le nombre d'itérations, etc.) pour améliorer la capacité du modèle à distinguer les classes négatives.
4. **Utiliser des techniques d'augmentation de données** :
   * Pour la classe négative, générer des exemples supplémentaires en utilisant des techniques telles que la **paraphrase** ou la **traduction inverse** pour créer des données synthétiques qui aident à mieux entraîner le modèle.
5. **Explorer d'autres modèles ou architectures** :
   * Essayer d'autres modèles plus adaptés aux déséquilibres de classe, comme **les réseaux de neurones** avec **apprentissage pondéré des classes** ou **les modèles d'ensembles** (comme les forêts aléatoires ou XGBoost) qui peuvent mieux gérer les déséquilibres.