Rapport TP Deep Learning

Le problème des SPAMs

- Chargez l'ensemble de données.
- Inspectez et nettoyez les données si nécessaire.
  - Vérifier les nulls => y'a pas de valeurs nulls
  - Vérifier les doublons => y'a 404 linges en double, on a supprimé en utilisant le code suivant

    ```
    data = data.drop_duplicates(keep='first')
    ```

- Encodez les caractéristiques catégorielles en numérique.
  - J'ai encoder les valeurs de la colonne Class en transformant les chaines de caractère en valeurs binaires (0 et 1) en utilsant

    ```
    encoder = LabelEncoder()
    ```

- Divisez vos données en ensembles d'entraînement et de test.
- Utilisez FEATURE ENGINEERING pour créer des nouvelles colonnes
  - J'ai rajouté trois colonnes : nombre de caractères, nombre de mot et nombre de phrases.
  - Aussi j'ai utilisé le tfidf pour tokenizer la colonne SMS, ce qui a été utilisé dans l'entrainement du modèle MultinomialNB `mnb = MultinomialNB()`
  - Enfin j'ai utilisé le word2Vec, plus précisement `word_vectors = api.load("glove-twitter-25")` pour tokenizer le SMS, et pour ce cas on a utilisé le modèle LogisticRegression `logreg = LogisticRegression()`
- L'évaluation des modèles est faites en utilisant la matrice de confusion et l'accuracy :

- Pour le modèle MultinomialNB on a eu ceci comme résultat :

```
Classification Report:
              precision    recall  f1-score   support

           0       0.97      1.00      0.99       891
           1       1.00      0.81      0.90       143

    accuracy                           0.97      1034
   macro avg       0.99      0.91      0.94      1034
weighted avg       0.97      0.97      0.97      1034
```

  Le rapport de classification fournit des mesures de performance pour chaque classe et des moyennes globales. Pour la classe 0, le modèle a une excellente précision, rappel et score F1 de 0.97, 1.00 et 0.99 respectivement, ce qui indique qu'il prédit correctement la classe 0 dans la plupart des cas. Pour la classe 1, la précision est parfaite à 1.00, mais le rappel est légèrement plus bas à 0.81, ce qui signifie que le modèle peut manquer quelques exemples positifs. L'accuracy globale du modèle est de 0.97, ce qui indique qu'il prédit correctement la classe pour 97% des exemples dans l'ensemble de données de test. En moyenne, le modèle a une précision, un rappel et un score F1 de 0.99, 0.91 et 0.94 respectivement, ce qui montre une performance globalement élevée, mais avec une certaine différence entre les classes. En conclusion, le modèle semble être performant dans la prédiction des deux classes, mais il peut être légèrement biaisé vers la classe majoritaire.

- Pour le modèle LogisticRegression on a eu ceci comme résultat :

```
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.97      0.95       891
           1       0.77      0.57      0.66       143

    accuracy                           0.92      1034
   macro avg       0.85      0.77      0.80      1034
weighted avg       0.91      0.92      0.91      1034
```

- Précision (Precision) : Pour la classe 0, la précision est de 0.93, ce qui signifie que sur toutes les prédictions faites pour la classe 0, 93% sont correctes. Pour la classe 1, la précision est de 0.77, indiquant que sur toutes les prédictions faites pour la classe 1, 77% sont correctes.

- Rappel (Recall) : Pour la classe 0, le rappel est de 0.97, ce qui signifie que le modèle a réussi à capturer 97% des exemples réels de la classe 0. Pour la classe 1, le rappel est de

0.57, indiquant que le modèle n'a réussi à capturer que 57% des exemples réels de la classe 1.

- F1-score : Le score F1 est la moyenne harmonique de la précision et du rappel. Il fournit une mesure de l'équilibre entre la précision et le rappel. Pour la classe 0, le score F1 est de 0.95, et pour la classe 1, il est de 0.66.

En résumé, les données ont été chargées, nettoyées, encodées et divisées, puis des nouvelles caractéristiques ont été ajoutées via le Feature Engineering. Les modèles MultinomialNB et Logistic Regression ont été entraînés et évalués avec succès, fournissant des résultats d'accuracy satisfaisants.

Test et example :

```python
def predict_message(message):
    # Transform the message using the TF-IDF vectorizer
    message_tfidf = tfidf.transform([message])

    prediction = mnb.predict(message_tfidf)
    return "Spam" if prediction == 1 else "Not Spam"

example_message = "hello Lydia, How are you ?"
print(predict_message(example_message))

example_message = "Congratulations! You won iphone 15 pro max, call now!!!"
print(predict_message(example_message))
```

```
Not Spam
Spam
```