Reunion 28 novembre

To do

- **Essayer un modèle séquentiel**
- **Tester RoBERTa au lieu de CamemBERT pour la classification binaire**
- **Explorer l'utilisation des grands modèles de langage (LLMs)**
- **Rédiger l’abstrait pour la conférence de Miami**
- **Organiser le dépôt GitHub**

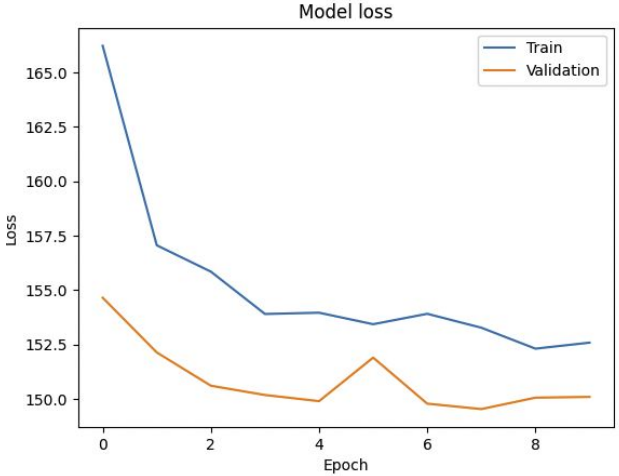
Sequential model

Pour rappel, nous avons obtenu de bons résultats en classification binaire (pause/pas de pause) avec BERT. Lors de notre dernière réunion, nous avons évoqué l'idée de tester un réseau de neurones simple – un modèle séquentiel.

J'ai essayé de mettre cela en œuvre avec les entrées suivantes : **totalActions**, **totalChars**, **finalChars**, **totalDeletions**, **innerDeletions**, **docLen**, **avg_shift**, et **num_actions**.

J'ai mené plusieurs expériences à partir de ces paramètres.

Premier essai



Paramètres d'entraînement :

- **Epochs :** 10
- **batch_size :** 32

Résultats :

- **Erreur Absolue Moyenne (Mean Absolute Error) sur le jeu de test :** 5,2734
- **Score R^2 :** 0,0110
  - Cela signifie que seulement **1,1 %** de la variance de la variable dépendante (durée de la pause) est expliquée par les variables indépendantes (features) du modèle.
  - Un score R^2 proche de 0 indique que le modèle n'explique pas bien la variance des données, tandis qu'un score proche de 1 montre une bonne capacité explicative.

En moyenne, les prédictions du modèle présentent une erreur d'environ **5,27 secondes**.

En utilisant toutes features

Nous avions précédemment enrichi les données avec de nombreuses informations supplémentaires (POS, dépendances, fréquences relatives, etc.). En utilisant **Word2Vec**, j'ai vectorisé les données textuelles et réessayé le modèle précédent, ce qui a généré **59 218 colonnes**.

Paramètres d'entraînement :

- **Epochs :** 100

Résultats :

- **Erreur Absolue Moyenne (Mean Absolute Error) sur le jeu de test :** 1,2525
- **Score R² :** -6,7084

Cela révèle une **grande contradiction** entre la MAE (relativement basse) et le score R² (fortement négatif). Cela pourrait indiquer que, bien que les prédictions soient globalement proches des valeurs réelles en moyenne, le modèle ne capture pas correctement la variance dans les données.

Discordance entre MAE et Score R^2

La contradiction entre une **Erreur Absolue Moyenne (MAE)** faible et un **mauvais score R^2** (R^2 = -6,7084) indique que, bien que le modèle effectue des prédictions numériquement proches des valeurs cibles en termes d'erreur absolue, il échoue à expliquer la variance des données.

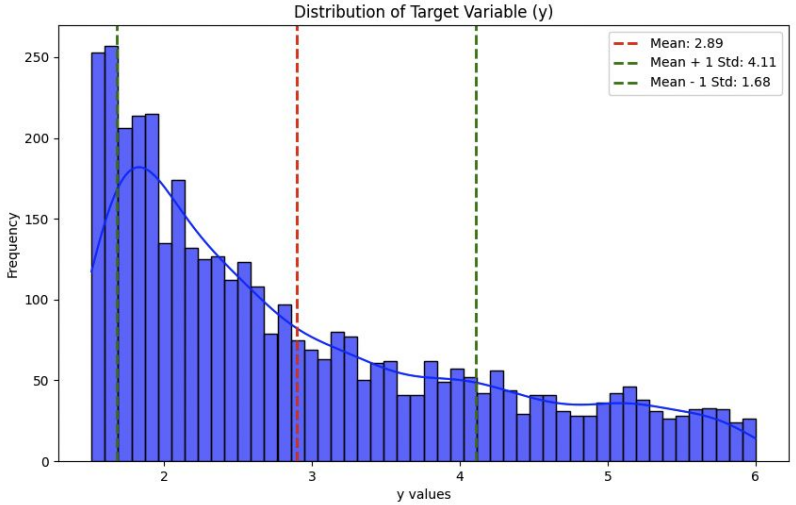
Plusieurs raisons peuvent expliquer ce phénomène :

1. **Référence mal adaptée pour le calcul de R^2**
   Le score R^2 est calculé par rapport à un modèle de base, généralement la moyenne des valeurs cibles. Si les valeurs cibles (**y**) présentent une très faible variance, le score R^2 pénalisera fortement le modèle à moins qu'il ne prévoie précisément la variance des données.
   Vérification de la variance des cibles :
   Il fallait donc vérifier la distribution de **y** pour voir si les valeurs sont fortement regroupées autour de la moyenne. Si c'est le cas, le modèle peut atteindre une faible MAE tout en échouant à capturer suffisamment de variance pour obtenir un bon score R^2.

Analyse de la distribution de pauses (valeurs de y)

Pour approfondir, j'ai représenté graphiquement les pauses (**y**). Voici les observations principales :

1. **Concentration autour de la moyenne :**
   Une part significative des données est concentrée près de la moyenne (**2,89**) et en dessous. Les valeurs semblent étroitement distribuées autour de cette plage, mais on observe également un certain nombre de valeurs plus élevées, bien que moins fréquentes.
2. **Asymétrie (skewness) :**
   La distribution est **asymétrique vers la droite**, ce qui signifie qu'il existe quelques grandes valeurs qui tirent la moyenne vers la droite. Ces valeurs élevées pourraient rendre difficile la généralisation du modèle.
3. **Faible variance :**
   Bien que l'écart-type (**1,21**) ne soit pas extrêmement faible, la concentration près de la moyenne combinée à l'asymétrie suggère que les prédictions du modèle sont probablement "suffisantes" (**faible MAE**) pour la majorité de la distribution, mais qu'elles échouent à expliquer efficacement la variance ($R^2 < 0$).

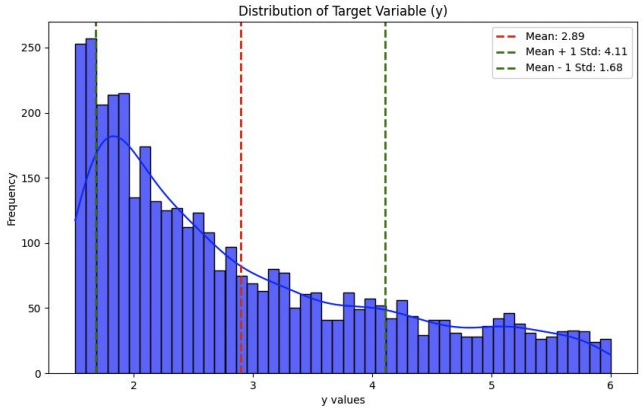


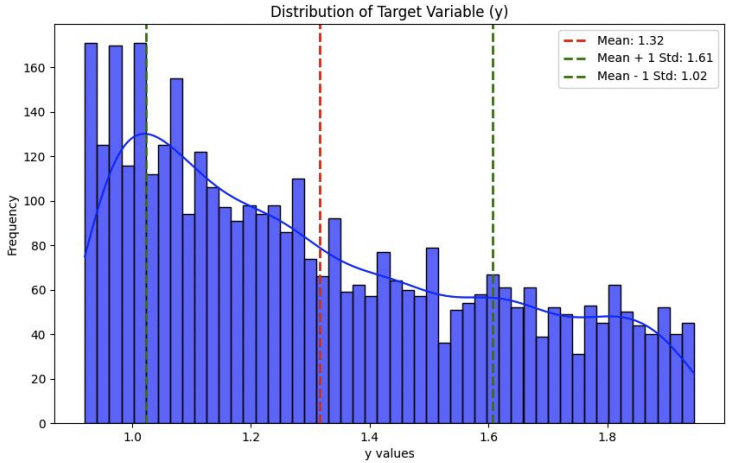
Étapes pour résoudre les problèmes d'asymétrie et de variance

Pour améliorer les performances du modèle et équilibrer l'asymétrie de la variable cible, les actions suivantes ont été entreprises :

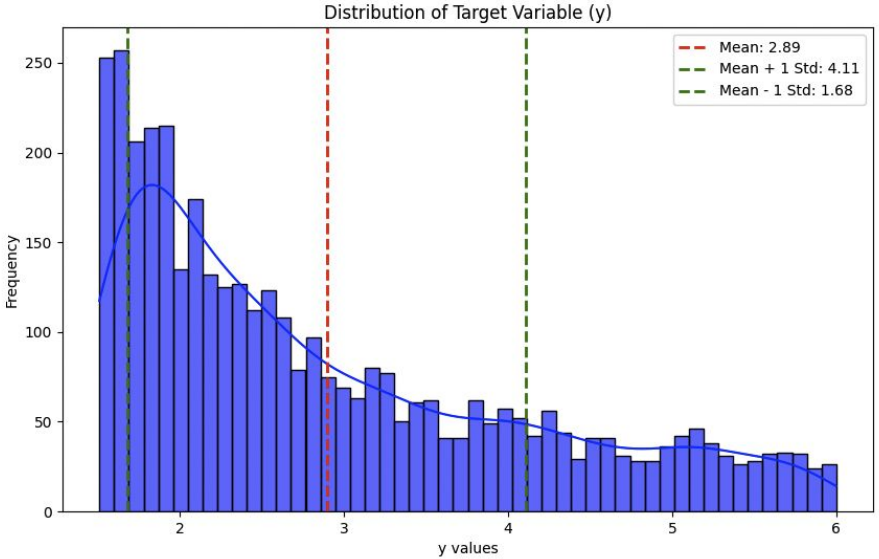
1. **Transformation de la variable cible :**
   Une transformation visant à rendre la variable cible plus symétrique peut aider le modèle à mieux capturer la variance.
   Approche : Application d'une **transformation logarithmique**.
   Résultat :
   La transformation logarithmique a :
   - Réduit l'asymétrie de la distribution.
   - Permis une meilleure répartition des valeurs, réduisant ainsi l'impact des grandes valeurs sur l'entraînement.

Cependant, les performances du modèle post-transformation doivent encore être évaluées en fonction des métriques telles que le **MAE** et le **R^2** pour déterminer si cette approche a réellement amélioré la capacité explicative.

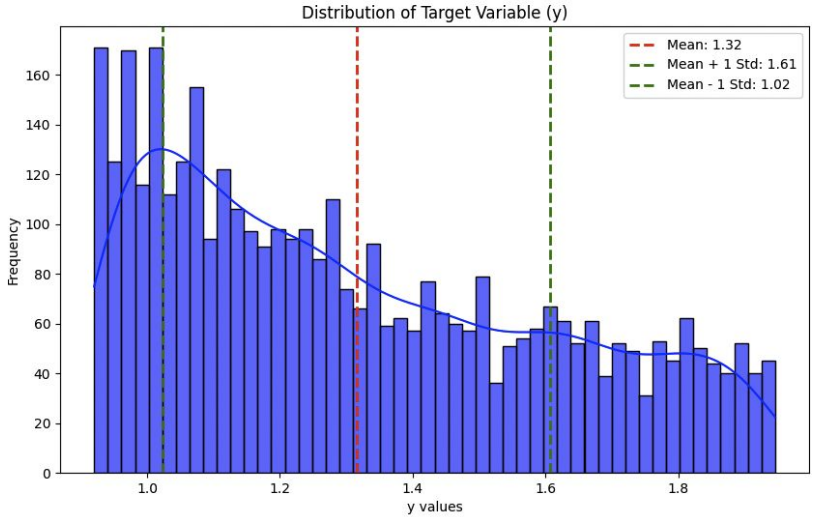




Distribution originale



Resultat de la transformation logarithmique



Réduction des valeurs aberrantes extrêmes (*cette fois après la transformation log)*

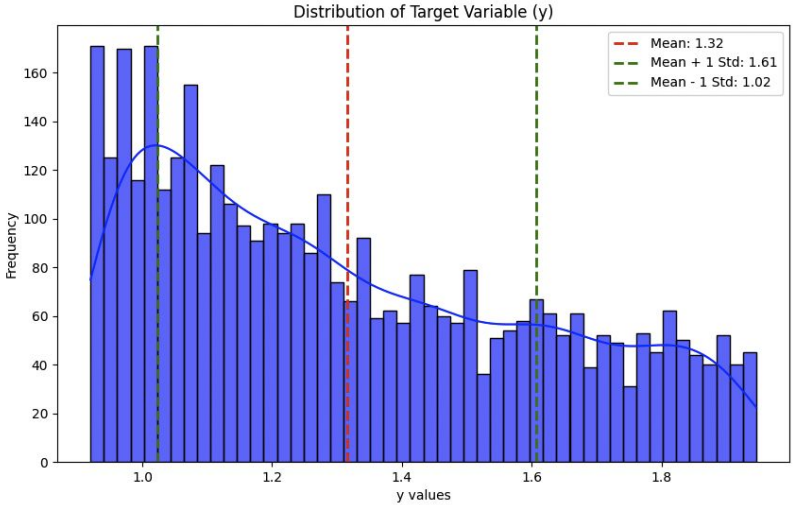
Pour limiter l'influence des valeurs extrêmes, une méthode de **clipping** a été appliquée à la variable cible. Cela consiste à plafonner les valeurs à un seuil spécifique, réduisant ainsi la dominance des valeurs les plus élevées.

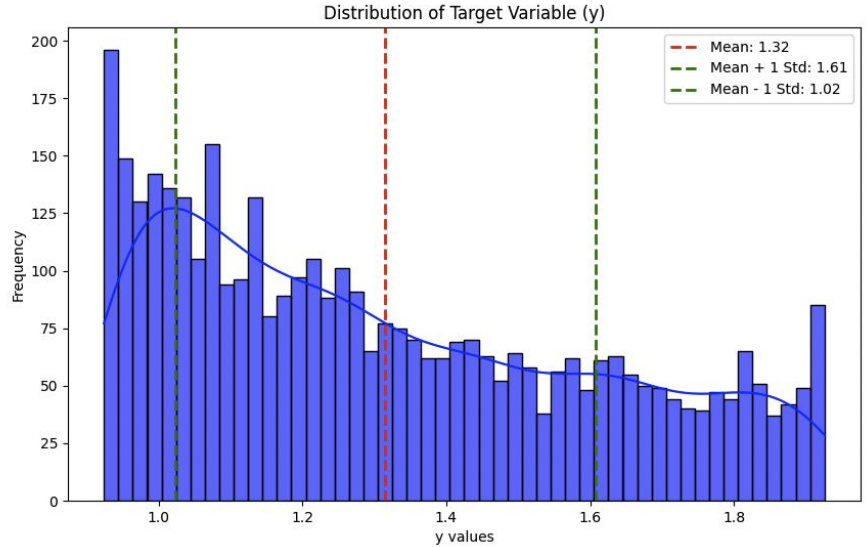
Approche (un peu différente de ce qu'on avait essayé avant):

- Définir un seuil supérieur pour la variable cible. Les valeurs dépassant ce seuil sont ramenées à la valeur maximale autorisée.

Avantages attendus :

- Réduction de l'effet disproportionné des grandes valeurs sur l'entraînement du modèle.
- Amélioration de la généralisation du modèle sur la majorité des données.





3. Normalisation de la variable cible

Si la variable cible présente une large plage de valeurs, la normaliser pour qu'elle ait une moyenne de zéro et une variance unitaire peut aider le modèle à apprendre plus efficacement.

Après ces étapes, le modèle séquentiel a été relancé avec :

- **Batch size :** 32
- **Epochs :** 100

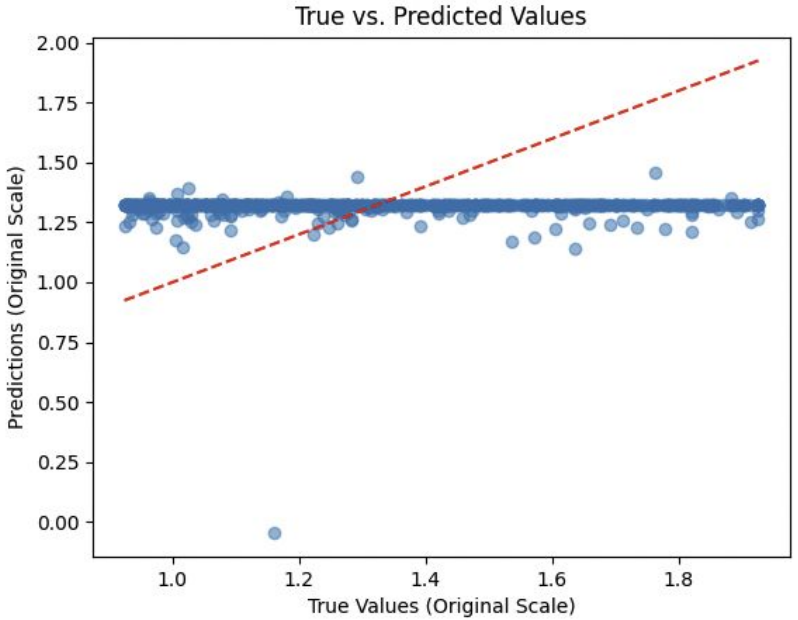
Résultats :

- **Erreur Absolue Moyenne (MAE) sur le jeu de test :** 0,2559
- **Score R^2 :** -0,0259

- **Erreur Absolue Moyenne (MAE) sur le jeu de test :** 0,2559
- **Score R² :** -0,0259

Interprétation du graphique :

1. **Répartition des points (scatter plot) :**
   - Les valeurs prédites (**y_pred**) sont étroitement regroupées dans une plage étroite (**environ 1,2–1,3**).
   - À l'inverse, les valeurs réelles (**y_test**) varient davantage le long de l'axe des x.
2. **Ligne rouge en pointillés :**
   - La ligne rouge représente l'idéal théorique : des prédictions parfaites où **y_pred = y_test**.
   - Les prédictions du modèle s'en écartent significativement, car la majorité des points sont regroupés horizontalement près de 1,2, sans correspondre aux valeurs réelles.
3. **Prédictions plates :**
   - Le regroupement horizontal des valeurs prédites indique que le modèle n'apprend pas de relations significatives entre les caractéristiques d'entrée et la variable cible.
   - Il semble se contenter de prédire une plage restreinte proche de la moyenne de la variable cible.



Conclusion

Bon MAE mais mauvais score R^2 :

- Le faible MAE indique que les prédictions sont numériquement proches des valeurs cibles en termes absolus. Cela est probablement dû au fait que les valeurs réelles sont également concentrées dans une plage spécifique.
- Cependant, le R^2 négatif et le graphique montrent que le modèle échoue à expliquer la variance de la variable cible.

Problème avec le modèle :

- Le modèle semble **sous-ajusté** (**underfitting**).
- Il ne capture pas la complexité ou la variance de la variable cible, et ses prédictions restent excessivement simplistes (proches de la moyenne ou d'une petite plage de valeurs).

2. Ajustements supplémentaires des données et gestion du surapprentissage (overfitting)

Après les premiers essais infructueux, de nombreux ajustements des données et outliers ont été réalisés pour tenter d'améliorer les performances.

Cependant, un autre problème pourrait être lié au **surapprentissage (overfitting)** et à une **mauvaise utilisation du dropout**.

Observations :

- Le modèle inclut une couche de **dropout** (0,5) pour prévenir le surapprentissage, mais ce taux peut être trop élevé pour notre taille de données.
- Un dropout excessif peut empêcher le modèle d'apprendre des patterns significatifs, surtout si le jeu de données est relativement petit.

Approche testée :

1. **Réduction du taux de dropout :**
    - Tentative avec des valeurs plus faibles (par ex. Dropout(0,2)) ou suppression temporaire du dropout.
    - **Résultat :** Sans dropout, les performances se sont fortement dégradées à cause du surapprentissage.
2. **Utilisation de la régularisation L2 au lieu du dropout :**
    - Une couche supplémentaire a également été ajoutée au réseau.

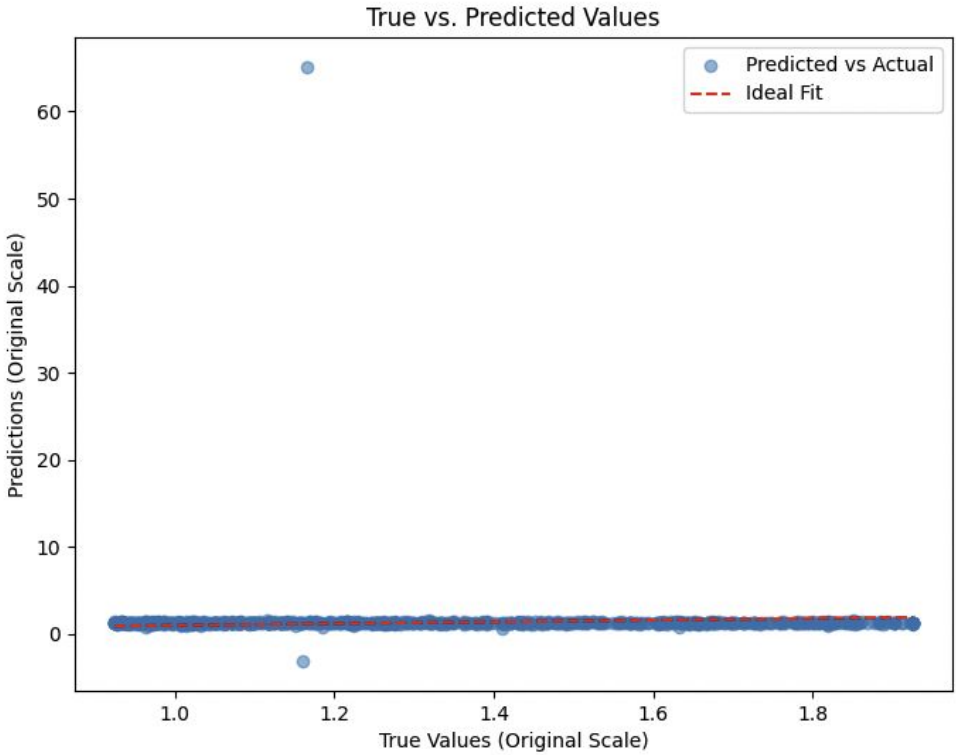
Résultats :

- **Erreur Absolue Moyenne (MAE) :** 0,4975
    - L'erreur a légèrement augmenté par rapport aux essais précédents, indiquant que les prédictions s'écartent en moyenne de **0,5 secondes** des valeurs réelles.
    - Cela reste dans la même plage que les résultats précédents, mais sans amélioration significative en précision.
- **Score R^2 :** -70,0718
    - Le score R^2 reste très négatif, ce qui indique que le modèle performe bien moins qu'un simple modèle de base (par ex., prédire la moyenne).
    - Cela renforce l'idée que le modèle n'explique aucune variance dans les données et qu'il est gravement **sous-ajusté (underfitting)**.

Apres plein d’ajustement on obtient…

Mean Absolute Error: 0.3434
R^2 Score: -60.4108

Pas horrible, mais l’erreur reste significative…?



Pourquoi on obtient pas un mieux modele

Valeurs aberrantes dans la variable cible :

- Malgré les efforts de **clipping** ou de **transformation**, des valeurs aberrantes extrêmes sont toujours présentes et affectent considérablement les performances du modèle.
- Le prétraitement actuel ne semble pas avoir traité efficacement ces anomalies.

Sous-ajustement du modèle (Underfitting) :

- Le réseau de neurones reste insuffisamment performant, échouant à apprendre des patterns significatifs à partir des caractéristiques.
- Plusieurs facteurs peuvent contribuer à cela :
    - **Complexité insuffisante du modèle** : Le modèle pourrait nécessiter davantage de couches ou de neurones pour mieux capturer la structure des données.
    - **Qualité médiocre des caractéristiques** : Les entrées actuelles (X) pourraient ne pas être suffisamment informatives ou bien prétraitées.
    - **Problème de taux d'apprentissage ou d'optimiseur** : Un taux d'apprentissage mal ajusté ou un choix inapproprié d'optimiseur peut nuire à l'entraînement.
    - **Nombre insuffisant d'époques ou de données** : Le modèle pourrait nécessiter davantage de cycles d'entraînement ou de données pour apprendre correctement.

Relation entre caractéristiques et cible :

- Il est pertinent de se demander si les caractéristiques (**X**) sont suffisamment prédictives pour expliquer la variance de la variable cible (**y**).
- Une faible corrélation entre les caractéristiques et la cible pourrait être à l'origine de la tendance plate observée.

Autres pistes…

Sélection des caractéristiques (Feature Selection)

La sélection des caractéristiques peut aider à identifier les variables les plus importantes qui contribuent aux performances du modèle. Plusieurs méthodes permettent de réaliser cette sélection, notamment :

- Les **tests statistiques**,
- La **sélection basée sur un modèle**,
- L'**élimination récursive des caractéristiques (Recursive Feature Elimination, RFE)**.

Étapes :

1. **Entraîner un Random Forest Regressor :**
   - Le modèle Random Forest entraîné sur l'ensemble des données pour calculer l'importance des caractéristiques.
2. **Sélection des caractéristiques importantes :**
   - Les importances des caractéristiques fournies par le Random Forest utilisées pour sélectionner les plus pertinentes.
   - Cela permet de réduire la complexité du modèle et de supprimer les variables non significatives ou bruitées.
3. **Réentraîner le réseau neuronal avec les caractéristiques sélectionnées :**

2eme to do - implémenter et essayer d'autres transformers que CamemBERT

RoBERTa:

- meilleurs résultats
- faut encore rechercher la correcte methodologie pour avoir des resultats definitifs

Classification Report:

	precision	recall	f1-score	support
No Pause	1.00	0.97	0.98	776
Pause	0.80	1.00	0.89	106
accuracy			0.97	882
macro avg	0.90	0.98	0.94	882
weighted avg	0.98	0.97	0.97	882

Classification Report:

	precision	recall	f1-score	support
No Pause	1.00	0.92	0.96	733
Pause	0.72	0.99	0.84	149
accuracy			0.93	882
macro avg	0.86	0.96	0.90	882
weighted avg	0.95	0.93	0.94	882

Classification Report:

	precision	recall	f1-score	support
No Pause	0.95	0.92	0.94	743
Pause	0.64	0.74	0.69	139
accuracy			0.89	882
macro avg	0.80	0.83	0.81	882
weighted avg	0.90	0.89	0.90	882

3eme to do - essayer des LLMs:

Nous avons utilisé un **LLM (T5-small)** pour classifier le texte au niveau des tokens, en transformant une annotation spécialisée (par exemple, **cat_1**, **cat_5**) en étiquettes binaires.

T5-small a été choisi pour sa légèreté et son efficacité dans des tâches de classification au niveau granulaire (token).

Comment on peut utiliser des grands modèles de langage (LLMs) dans notre cas:

Reformulation du problème :

La tâche est abordée comme un problème de **séquence-à-séquence**, une approche dans laquelle les LLMs sont excellent.

- **Entrée :** Une séquence de texte (par exemple, *"The cat sat cat_1 on the mat cat_5"*).
- **Sortie :** Une séquence binaire (par exemple, *"0 1 0 0 0 1"*).

Utilisation de T5-small :

1. **Choix du modèle :**
   - **T5** est un modèle de type "texte-vers-texte", ce qui signifie qu'il peut prendre une séquence de texte en entrée et produire une séquence de texte en sortie.
2. **Pourquoi T5 ?**
   - Efficace pour les tâches nécessitant une compréhension fine au niveau des tokens.
   - Moins volumineux que des modèles de type GPT, ce qui le rend plus pratique et économique pour ce cas d'usage.

Prétraitement des données :

1. **Tokenisation :**
   - Le texte d'entrée est tokenisé à l'aide du tokenizer de **T5**, divisant les mots et annotations en unités distinctes.
2. **Alignement des étiquettes :**
   - Les étiquettes binaires (**0 ou 1**) sont alignées avec les tokens correspondants.

Resultat:

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

    No Pause       0.88      1.00      0.94       342
       Pause       0.00      0.00      0.00        46

    accuracy                           0.88       388
   macro avg       0.44      0.50      0.47       388
weighted avg       0.78      0.88      0.83       388
```