

Reunion du 17/09/24

Rappel pour Aleksandra

Nous essayons de prédire les pauses entre les bursts. Nous cherchons donc à faire une régression pour obtenir au mieux une pause en fonction du burst lui étant rattaché (celui d'avant ou d'après). Dans notre tableau initial, nous avons plusieurs lignes par burst car certains features changent dépendamment de la partie du burst (ex: categorie, positions...). D'autres features concernent le burst dans son entièreté.

Rappel pour Aleksandra

Nous avons décidé d'utiliser un réseau de neurones récurrent car nous avons à faire à des données séquentielles avec une portée temporelle. Pour prédire la pause d'une séquence il est important de garder en mémoire la séquence précédente. Nous avons donc contracté les données pour n'avoir qu'un burst par ligne et ses features globaux et ses features 'arbores' donc des listes d'éléments dans le cas des catégories par exemple. Nous avons aussi ajouté des features comme les POS, les dépendances, les chunks ou encore des indices de rareté des mots du burst dans la langue française.

Rappel pour Aleksandra

Les résultats obtenus n'étaient pas du tout satisfaisants, nous avons toujours un R Carré négatif et des prédictions non satisfaisantes. Nous avons des améliorations en enlevant certains features, en calculant les hyperparamètres et en procédant à des transformations logarithmiques, attention mechanisms et autres...

Sommaire

Amélioration du réseau de neurones récurrent :

- Réduction de la variance des données

Test de modèles moins complexes :

- Réseau de neurones simple (sequential)

- Random Forest

- Gradient boosting

RNN - Réduction de la variance

Nous avons beaucoup de pauses a la longueur anormalement élevée dans notre input. Nous avons donc décidé de retirer les pauses les plus hautes et les plus basses pour réduire la variance de nos données.

On a calculé le interquartile range et on a éliminé les valeurs qui se trouvent dans le quart inférieur et supérieur. Les limites qui ont résulté du calcul sont:

Lower bound: 5.074999999999999

Upper bound: 14.004999999999999

Donc, les bursts dont la pause n'était pas comprise dans cet interval ont été retirés des données.

Aussi pour réduire la dimension de nos données, on a réduit la dimension des vecteurs, ou chaque mot est représenté par un vecteur de dimension 50 a la place de 100.

Mesures d'évaluation

Lors de l'exécution de notre script, nous obtenons pour chaque epoch les mesures suivantes qui permettent d'évaluer le modèle

1. **Loss (perte)** = la différence entre la sortie prédite par le modèle et la valeur réelle cible.

- La fonction de perte calcule cette différence et donne une valeur que le modèle essaie de minimiser pendant l'entraînement.

- Plus la perte est faible, mieux le modèle effectue ses prédictions. Les poids du modèle sont mis à jour durant l'entraînement pour minimiser cette valeur de perte.

2. **MAE (Erreur Absolue Moyenne)** = moyenne des différences absolues entre les valeurs prédites et les valeurs réelles.

3. **val_loss (perte de validation)** :

- C'est la perte calculée sur l'ensemble de validation, qui est séparé des données d'entraînement. La perte de validation permet de surveiller à quel point le modèle se généralise bien sur des données non vues auparavant.

- Un modèle avec une faible perte d'entraînement mais une perte de validation élevée pourrait être en train de surapprendre (overfitting), c'est-à-dire qu'il fonctionne bien sur les données d'entraînement, mais mal sur de nouvelles données.

4. **val_mae (Erreur Absolue Moyenne sur Validation)**:

- C'est l'erreur absolue moyenne calculée sur l'ensemble de **validation**. Comme pour la ``val_loss``, elle aide à suivre les performances du modèle sur des données qui n'ont pas été vues pendant l'entraînement.

- Si ``val_mae`` est significativement plus élevé que ``mae`` sur les données d'entraînement, cela peut indiquer un surapprentissage.

Pendant l'entraînement, on cherche à voir une diminution à la fois de la ``loss`` et de la ``val_loss`` (et de même pour ``mae`` et ``val_mae``). Si la ``val_loss`` commence à augmenter tandis que la ``loss`` continue de diminuer, cela peut signaler un surapprentissage.

Premier essai - 16 neurones

R² Score: -0.10818193437040446

True Value: 5.15, Predicted Value: 9.112425804138184

True Value: 5.39, Predicted Value: 7.585991859436035

True Value: 5.73, Predicted Value: 7.537899494171143

True Value: 6.95, Predicted Value: 9.144816398620605

True Value: 9.31, Predicted Value: 9.276642799377441

Epoch 1/10

26/26 - 68s - 3s/step - loss: 55.1117 - mae: 5.91

Epoch 2/10

26/26 - 60s - 2s/step - loss: 19.2027 - mae: 3.4176 - val_loss: 16.7163 - val_mae: 3.2798

Epoch 3/10

26/26 - 57s - 2s/step - loss: 13.2839 - mae: 2.8247 - val_loss: 10.0075 - val_mae: 2.4643

Epoch 4/10

26/26 - 57s - 2s/step - loss: 8.3795 - mae: 2.2927 - val_loss: 9.0830 - val_mae: 2.2774

Epoch 5/10

26/26 - 57s - 2s/step - loss: 7.1754 - mae: 2.1481 - val_loss: 7.3455 - val_mae: 2.1968

Epoch 6/10

26/26 - 58s - 2s/step - loss: 7.0318 - mae: 2.1620 - val_loss: 7.1722 - val_mae: 2.2306

Epoch 7/10

26/26 - 57s - 2s/step - loss: 6.9944 - mae: 2.1561 - val_loss: 7.3192 - val_mae: 2.1623

Epoch 8/10

26/26 - 57s - 2s/step - loss: 6.5244 - mae: 2.0932 - val_loss: 6.8612 - val_mae: 2.2126

Epoch 9/10

26/26 - 57s - 2s/step - loss: 6.5241 - mae: 2.1016 - val_loss: 6.9950 - val_mae: 2.2433

Epoch 10/10

26/26 - 57s - 2s/step - loss: 7.1010 - mae: 2.1729 - val_loss: 7.3109 - val_mae: 2.3042

9/9 - 4s - 403ms/step - loss: 6.9462 - mae: 2.2341

Premier essai - 16 neurones

```
R2 Score: -0.10818193437040446
True Value: 5.15, Predicted Value: 9.112425804138184
True Value: 5.39, Predicted Value: 7.585991859436035
True Value: 5.73, Predicted Value: 7.537899494171143
True Value: 6.95, Predicted Value: 9.144816398620605
True Value: 9.31, Predicted Value: 9.276642799377441
```

1. Perte d'entraînement et de validation :

- La **perte (MSE)** et la **MAE (erreur absolue moyenne)** diminuent progressivement au fil des 10 époques d'entraînement.
- Cela montre que le modèle apprend et s'adapte de mieux en mieux aux données d'entraînement au fur et à mesure que l'entraînement progresse.
- Cependant, les **valeurs de validation** ne sont pas beaucoup plus faibles que celles de l'entraînement, ce qui pourrait indiquer un **surapprentissage** (overfitting) ou une difficulté à améliorer davantage au-delà d'un certain point.

2. Évaluation sur le jeu de test :

- Lors de l'évaluation du modèle sur les données de test, on obtient :
 - **Perte** : 6.6933 (MSE)
 - **MAE** : 2.1319
 - Ces valeurs sont similaires à celles de la validation, ce qui indique une certaine cohérence dans les performances du modèle entre les données de validation et celles de test.

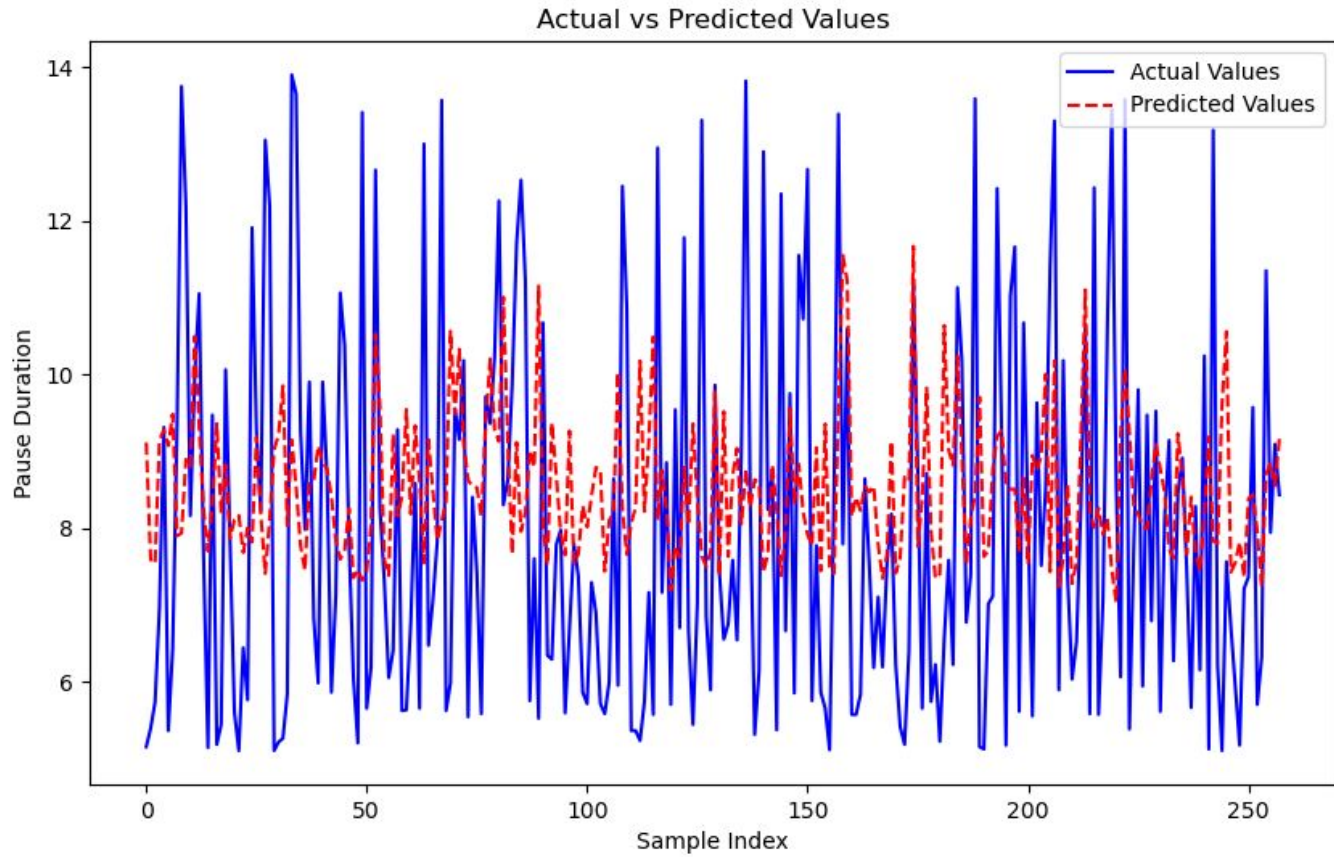
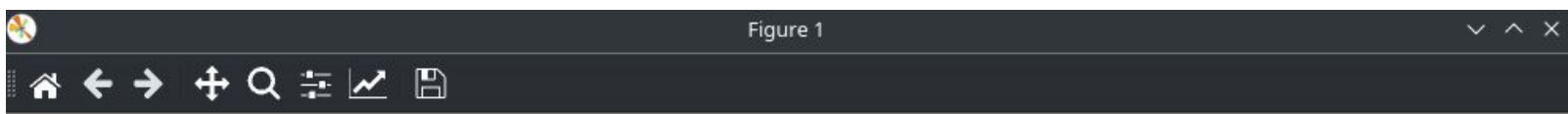
3. Score R² :

- Le score R² est de **-0.108**, ce qui est assez faible. Le score R² mesure dans quelle mesure le modèle explique la variance des données, avec 1 représentant un ajustement parfait et 0 indiquant que le modèle ne fait pas mieux que la moyenne de la variable cible.
- Un R² négatif signifie que le modèle performe **moins bien** que simplement prédire la moyenne des valeurs cibles. En gros, le modèle ne capture pas bien les motifs dans les données de test.

4. Prédictions vs Valeurs réelles :

- Dans certains cas, les valeurs prédites sont proches (par exemple, 9.31 vs 9.28), mais dans d'autres, les prédictions sont très éloignées (par exemple, 5.15 vs 9.11 ou 5.73 vs 7.54).

Cela suggère que, bien que le modèle ait appris certains motifs, il a du mal à prédire correctement les valeurs les plus basses et a tendance à surestimer.



Deuxième essai - 64 neurones

```
R2 Score: -0.07786798718868404
True Value: 5.15, Predicted Value: 8.166297912597656
True Value: 5.39, Predicted Value: 8.108725547790527
True Value: 5.73, Predicted Value: 8.622583389282227
True Value: 6.95, Predicted Value: 8.481106758117676
True Value: 9.31, Predicted Value: 8.74366569519043
```

```
Total params: 172,673 (674.50 KB)
Trainable params: 172,673 (674.50 KB)
Non-trainable params: 0 (0.00 B)
Epoch 1/10
26/26 - 96s - 4s/step - loss: 155.6333 - mae: 8.8215 - val_loss: 21.5545 - val_mae: 3.9288
Epoch 2/10
26/26 - 86s - 3s/step - loss: 16.2428 - mae: 3.1266 - val_loss: 10.7552 - val_mae: 2.4449
Epoch 3/10
26/26 - 86s - 3s/step - loss: 9.7269 - mae: 2.4556 - val_loss: 6.5982 - val_mae: 2.1862
Epoch 4/10
26/26 - 87s - 3s/step - loss: 9.9071 - mae: 2.5144 - val_loss: 8.9429 - val_mae: 2.5750
Epoch 5/10
26/26 - 86s - 3s/step - loss: 9.6003 - mae: 2.4329 - val_loss: 8.3737 - val_mae: 2.2305
Epoch 6/10
26/26 - 87s - 3s/step - loss: 8.5610 - mae: 2.3332 - val_loss: 7.5101 - val_mae: 2.1559
Epoch 7/10
26/26 - 87s - 3s/step - loss: 6.3497 - mae: 2.0195 - val_loss: 7.4634 - val_mae: 2.1693
Epoch 8/10
26/26 - 94s - 4s/step - loss: 6.0664 - mae: 2.0043 - val_loss: 6.8997 - val_mae: 2.2269
Epoch 9/10
26/26 - 90s - 3s/step - loss: 6.1387 - mae: 2.0112 - val_loss: 7.0968 - val_mae: 2.2775
Epoch 10/10
26/26 - 85s - 3s/step - loss: 6.1336 - mae: 2.0269 - val_loss: 6.5128 - val_mae: 2.1013
9/9 - 8s - 837ms/step - loss: 6.1708 - mae: 2.0728
```

Deuxième essai - 64 neurones

```
R2 Score: -0.07786798718868404
True Value: 5.15, Predicted Value: 8.166297912597656
True Value: 5.39, Predicted Value: 8.108725547790527
True Value: 5.73, Predicted Value: 8.622583389282227
True Value: 6.95, Predicted Value: 8.481106758117676
True Value: 9.31, Predicted Value: 8.74366569519043
```

1. Perte (Loss) et MAE (Mean Absolute Error) pendant l'entraînement :

- **Epoch 1** : Le modèle démarre avec une **perte de 155.6333** (MSE) et une **MAE de 8.8215**, ce qui est typique pour les premières itérations d'entraînement. La perte et la MAE sont encore élevées car le modèle n'a pas encore appris suffisamment de motifs dans les données.
- **Epochs 2 à 10** : Au fur et à mesure des époques, la perte et la MAE diminuent :
 - À la **10e époque**, la perte d'entraînement est descendue à **6.1336** avec une MAE de **2.0269**.
 - La **perte de validation** a diminué également, atteignant **6.5128** avec une MAE de **2.1013**.

Cela montre que le modèle apprend progressivement à minimiser les erreurs pendant l'entraînement, ce qui est un bon signe.

2. Évaluation sur les données de test :

- Lors de l'évaluation finale sur les données de test :
 - **Perte (MSE)** : 6.1708
 - **MAE** : 2.0728

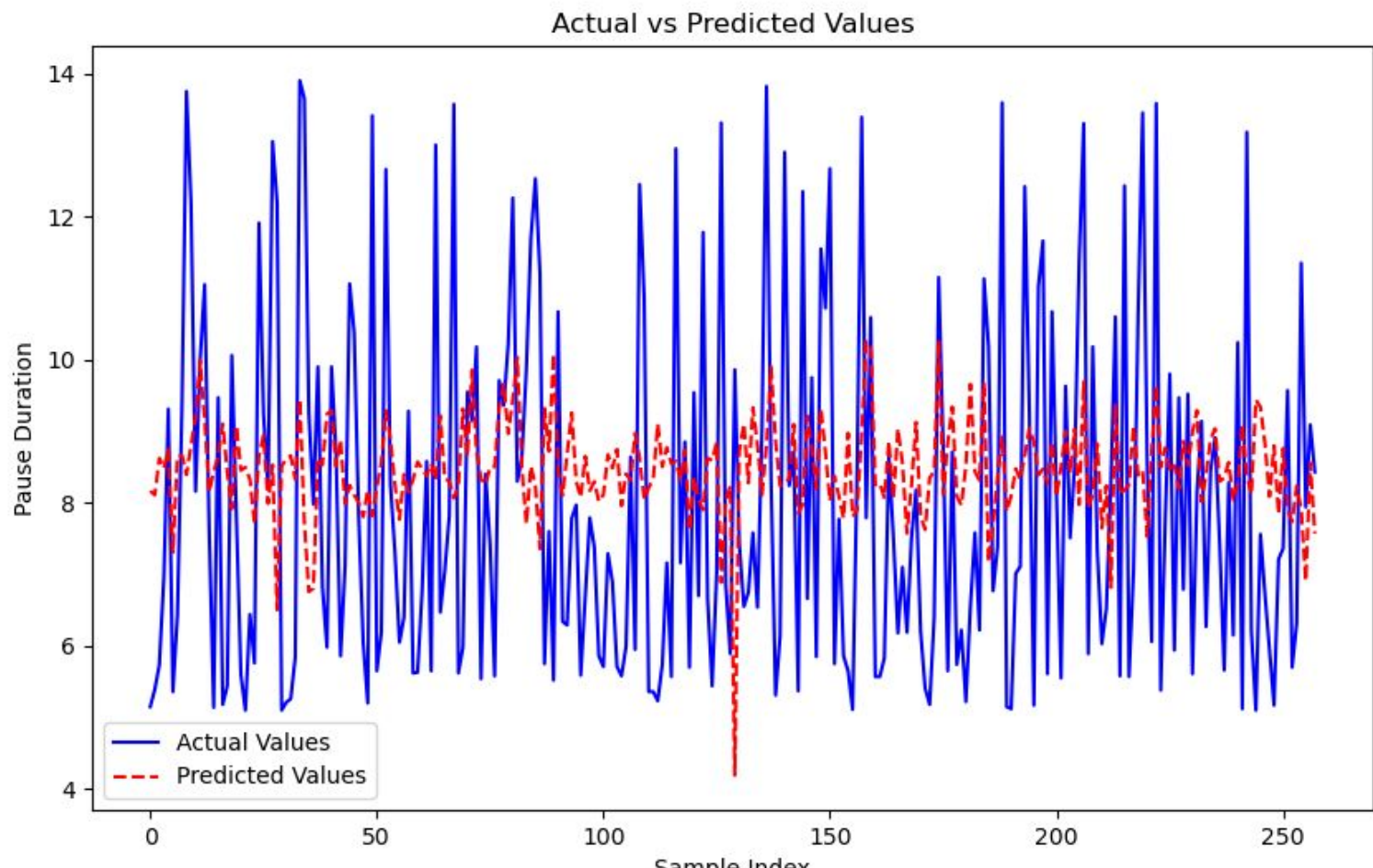
Ces résultats sont cohérents avec les performances de validation, ce qui signifie que le modèle ne souffre pas trop de surapprentissage (overfitting), mais qu'il a encore du mal à prédire avec précision.

3. Score R^2 :

- Le **score R^2** est de **-0.0216**, ce qui est très faible.
 - Un **R^2 négatif** indique que le modèle fait pire que si on avait simplement prédit la moyenne des valeurs cibles.
 - Cela montre que, malgré la diminution de la perte et de la MAE, le modèle ne s'ajuste pas bien aux motifs des données de test et ne parvient pas à généraliser efficacement.

4. Comparaison des valeurs réelles vs prédictions :

On observe que les **prédictions sont souvent éloignées des valeurs réelles**. Par exemple, pour une vraie valeur de **9.31**, la prédiction est seulement **6.90**, ce qui montre que le modèle a du mal à capturer la variabilité des valeurs élevées.



Réseau simple - 1er essai

```
Epoch 1/10
2024-09-13 14:45:48.243823: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:117] Plugin optimizer for device_type GPU is enabled.
26/26 - 4s - 173ms/step - loss: 512989344.0000 - mae: 13989.0479 - val_loss: 1995899.3750 - val_mae: 1315.9073
Epoch 2/10
26/26 - 0s - 9ms/step - loss: 2154571.0000 - mae: 805.1268 - val_loss: 74.1251 - val_mae: 8.2508
Epoch 3/10
26/26 - 0s - 9ms/step - loss: 71.3128 - mae: 8.0848 - val_loss: 74.1188 - val_mae: 8.2504
Epoch 4/10
26/26 - 0s - 9ms/step - loss: 71.3060 - mae: 8.0844 - val_loss: 74.1083 - val_mae: 8.2497
Epoch 5/10
26/26 - 0s - 9ms/step - loss: 71.2953 - mae: 8.0838 - val_loss: 74.0991 - val_mae: 8.2492
Epoch 6/10
26/26 - 0s - 9ms/step - loss: 71.2878 - mae: 8.0833 - val_loss: 74.0894 - val_mae: 8.2486
Epoch 7/10
26/26 - 0s - 9ms/step - loss: 71.2822 - mae: 8.0830 - val_loss: 74.0795 - val_mae: 8.2480
Epoch 8/10
26/26 - 0s - 9ms/step - loss: 71.2716 - mae: 8.0823 - val_loss: 74.0692 - val_mae: 8.2474
Epoch 9/10
26/26 - 0s - 9ms/step - loss: 71.2627 - mae: 8.0817 - val_loss: 74.0582 - val_mae: 8.2467
Epoch 10/10
26/26 - 0s - 9ms/step - loss: 71.2513 - mae: 8.0810 - val_loss: 74.0468 - val_mae: 8.2460
9/9 1s 86ms/step - loss: 2928.0210 - mae: 11.6129
Mean Absolute Error on Test Set: 11.0370
Loss on Test Set: 2487.4871
```

Les valeurs initiales de la perte et de la MAE sont extrêmement élevées, ce qui indique que les performances du modèle sont médiocres au départ. Cependant, la perte de validation et la MAE sont nettement inférieures, ce qui pourrait suggérer que le modèle fait du surapprentissage (overfitting) ou que les poids initiaux ne sont pas bien adaptés aux données d'entraînement.

La perte et la MAE chutent drastiquement après la première époque et se stabilisent autour de 71,3 et 8,08 respectivement. Les valeurs de la perte et de la MAE de validation se stabilisent également autour de 74,0 et 8,25 respectivement. Cette diminution rapide des valeurs de perte et de MAE montre que le modèle apprend rapidement à ajuster les données d'entraînement. Cependant, la faible différence entre les métriques d'entraînement et de validation suggère que le modèle ne fait pas de surapprentissage.

Phase d'Évaluation

Évaluation sur le jeu de test :

- Perte : 2 928.0210
- MAE : 11.6129

L'évaluation sur le jeu de test montre une perte et une MAE plus élevées par rapport aux ensembles d'entraînement et de validation, ce qui indique que les performances du modèle sur des données non vues ne sont pas aussi bonnes que sur les données d'entraînement et de validation.

Observations Clés

- **Perte et MAE initiales élevées** : Les valeurs extrêmement élevées au début suggèrent que les poids initiaux du modèle n'étaient pas bien adaptés aux données.
- **Convergence rapide** : Le modèle converge rapidement vers une perte et une MAE stables, ce qui montre qu'il apprend bien les données d'entraînement.
- **Métriques de validation** : Les pertes et MAE de validation sont proches des métriques d'entraînement, suggérant que le modèle ne surapprend pas.
- **Performances sur le jeu de test** : Les valeurs plus élevées de perte et de MAE sur le jeu de test montrent que la généralisation du modèle sur des données non vues est moins bonne que sur les ensembles d'entraînement et de validation.

RS - 2eme essai - Standard scaler

```
non-trainable params: 0 (0.00 B)
Epoch 1/10
2024-09-13 14:49:24.451546: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:117] Plugin optimizer for device_type GPU is enabled.
26/26 - 1s - 49ms/step - loss: 55.7882 - mae: 4.1885 - val_loss: 41.8336 - val_mae: 2.9887
Epoch 2/10
26/26 - 0s - 9ms/step - loss: 24.5393 - mae: 3.0787 - val_loss: 58.9883 - val_mae: 3.1170
Epoch 3/10
26/26 - 0s - 9ms/step - loss: 114.7943 - mae: 3.3433 - val_loss: 30.0825 - val_mae: 2.7375
Epoch 4/10
26/26 - 0s - 9ms/step - loss: 104.1312 - mae: 3.3748 - val_loss: 20.1416 - val_mae: 2.5497
Epoch 5/10
26/26 - 0s - 9ms/step - loss: 32.3986 - mae: 3.0027 - val_loss: 7.2495 - val_mae: 2.0948
Epoch 6/10
26/26 - 0s - 9ms/step - loss: 37.8947 - mae: 2.8792 - val_loss: 10.2958 - val_mae: 2.2152
Epoch 7/10
26/26 - 0s - 9ms/step - loss: 35.6498 - mae: 2.8073 - val_loss: 38.7780 - val_mae: 2.5838
Epoch 8/10
26/26 - 0s - 9ms/step - loss: 83.8065 - mae: 2.9310 - val_loss: 8.2023 - val_mae: 2.1567
Epoch 9/10
26/26 - 0s - 9ms/step - loss: 15.3095 - mae: 2.5321 - val_loss: 7.2288 - val_mae: 2.1157
Epoch 10/10
26/26 - 0s - 9ms/step - loss: 24.7100 - mae: 2.6885 - val_loss: 7.7684 - val_mae: 2.1440
9/9 0s 7ms/step - loss: 80548.2734 - mae: 21.3961
Mean Absolute Error on Test Set: 18.3637
Loss on Test Set: 68152.4219
```


Les valeurs initiales de perte et de MAE sont relativement élevées, mais pas aussi extrêmes que lors du précédent essai. La perte et la MAE de validation sont plus faibles que les métriques d'entraînement, ce qui pourrait indiquer un surapprentissage initial ou que le modèle est mieux adapté aux données de validation.

Les valeurs de perte et de MAE fluctuent de manière significative d'une époque à l'autre, indiquant une instabilité dans le processus d'apprentissage. La perte et la MAE de validation fluctuent également, montrant que les performances du modèle sur l'ensemble de validation sont incohérentes. Ces fluctuations suggèrent que le modèle pourrait avoir du mal à apprendre une représentation stable des données. Les métriques d'entraînement et de validation ne montrent pas de tendance claire à l'amélioration, ce qui est préoccupant.

Phase d'Évaluation

Évaluation sur le jeu de test :

- Perte : 80 548.2734
- MAE : 21.3961

L'évaluation sur le jeu de test montre une perte et une MAE très élevées par rapport aux ensembles d'entraînement et de validation. Cela indique que les performances du modèle sur des données non vues sont médiocres et qu'il ne s'est pas bien généralisé.

Observations Clés

- **Fluctuations des métriques** : Les valeurs de perte et de MAE fluctuent considérablement pendant l'entraînement, ce qui indique une instabilité dans le processus d'apprentissage.
- **Métriques de validation** : La perte et la MAE de validation ne montrent pas de tendance claire à l'amélioration, ce qui suggère que le modèle n'apprend pas efficacement.
- **Performances sur le jeu de test** : Les pertes et MAE très élevées sur le jeu de test indiquent une mauvaise généralisation sur des données non vues.

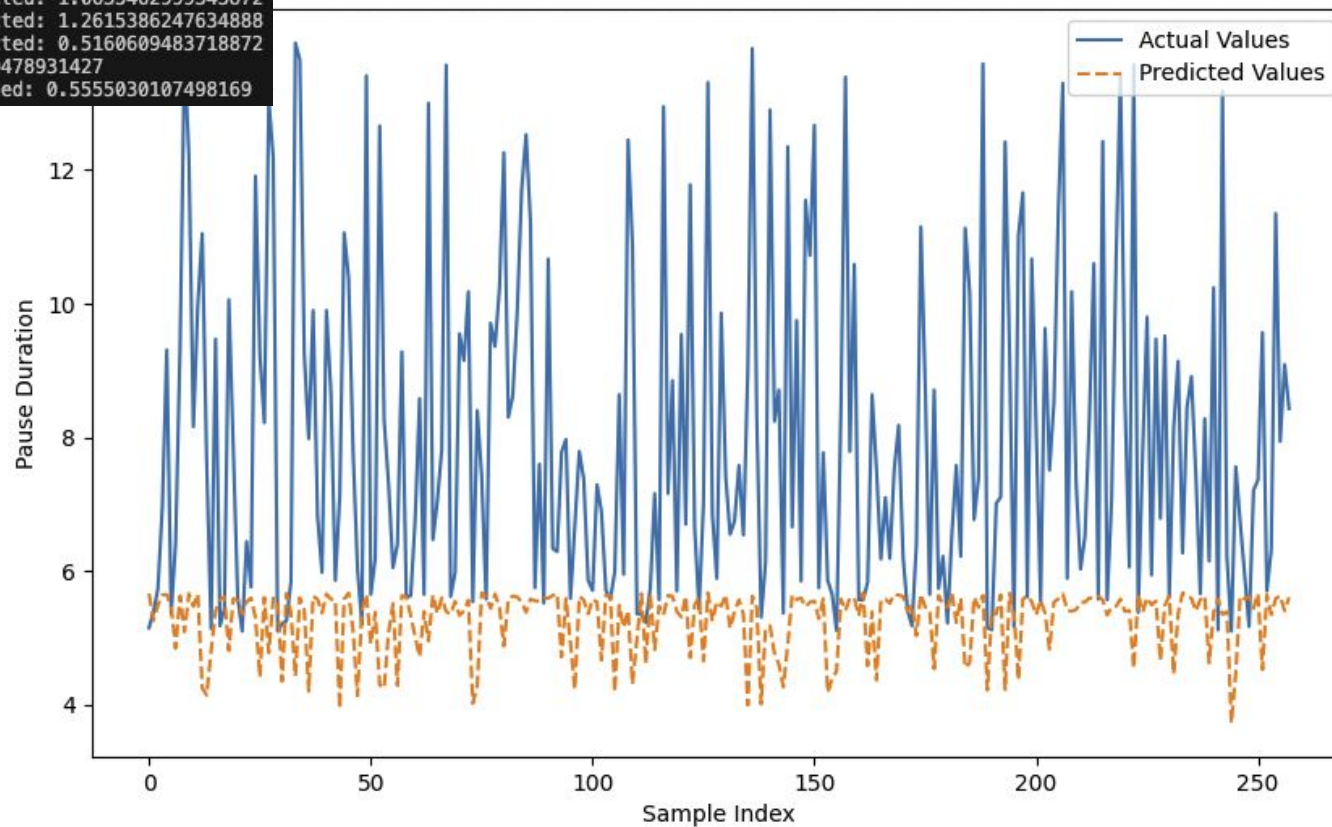
RS - 3eme essai - Standard scaler

- Réduction de la learning rate
- Réduire le batch size a 16
- Augmenter le nombre d'epochs à 20 pour laisser le temps au modèle d'apprendre

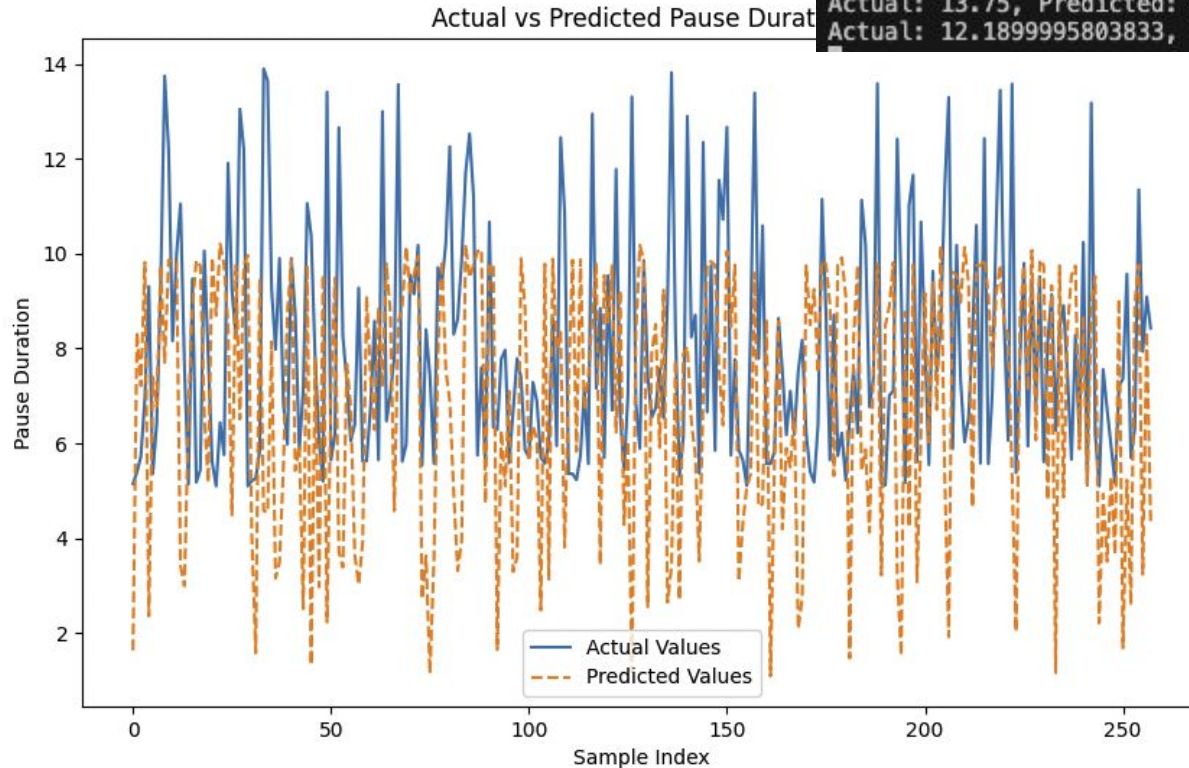
```
Epoch 1/20
2024-09-13 14:54:46.092826: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:117] Plugin optimizer for device_type GPU
52/52 - 4s - 74ms/step - loss: 748553.0625 - mae: 631.0253 - val_loss: 57086.1055 - val_mae: 238.7955
Epoch 2/20
52/52 - 0s - 8ms/step - loss: 102165.7031 - mae: 249.8122 - val_loss: 225.5349 - val_mae: 14.3918
Epoch 3/20
52/52 - 0s - 8ms/step - loss: 40007.9141 - mae: 158.3183 - val_loss: 887.6035 - val_mae: 29.0829
Epoch 4/20
52/52 - 0s - 8ms/step - loss: 21265.6855 - mae: 113.5323 - val_loss: 183.8191 - val_mae: 12.9282
Epoch 5/20
52/52 - 0s - 8ms/step - loss: 10162.1006 - mae: 78.6449 - val_loss: 27.3057 - val_mae: 4.2150
Epoch 6/20
52/52 - 0s - 8ms/step - loss: 6538.4478 - mae: 64.4728 - val_loss: 424.3934 - val_mae: 20.2975
Epoch 7/20
52/52 - 0s - 8ms/step - loss: 3422.2437 - mae: 45.0087 - val_loss: 62.0413 - val_mae: 7.0575
Epoch 8/20
52/52 - 0s - 8ms/step - loss: 2297.6509 - mae: 37.1988 - val_loss: 181.1839 - val_mae: 10.9747
Epoch 9/20
52/52 - 0s - 8ms/step - loss: 1272.0406 - mae: 27.9071 - val_loss: 122.5725 - val_mae: 8.9756
Epoch 10/20
52/52 - 0s - 8ms/step - loss: 752.6978 - mae: 21.7870 - val_loss: 67.1920 - val_mae: 7.3395
Epoch 11/20
52/52 - 0s - 8ms/step - loss: 479.7690 - mae: 17.0092 - val_loss: 54.8974 - val_mae: 6.3630
Epoch 12/20
52/52 - 0s - 8ms/step - loss: 351.6375 - mae: 14.8377 - val_loss: 47.7895 - val_mae: 5.8715
Epoch 13/20
52/52 - 0s - 8ms/step - loss: 229.0891 - mae: 12.0892 - val_loss: 56.1406 - val_mae: 6.8865
Epoch 14/20
52/52 - 0s - 9ms/step - loss: 191.5984 - mae: 10.9733 - val_loss: 62.6762 - val_mae: 7.5146
Epoch 15/20
52/52 - 0s - 9ms/step - loss: 133.6948 - mae: 9.2569 - val_loss: 86.1685 - val_mae: 8.9165
Epoch 16/20
52/52 - 0s - 8ms/step - loss: 91.6602 - mae: 7.8254 - val_loss: 59.9048 - val_mae: 7.2638
Epoch 17/20
52/52 - 0s - 8ms/step - loss: 84.9211 - mae: 7.4754 - val_loss: 59.3475 - val_mae: 7.2403
Epoch 18/20
52/52 - 0s - 8ms/step - loss: 62.5874 - mae: 6.4508 - val_loss: 55.7215 - val_mae: 7.0212
Epoch 19/20
52/52 - 0s - 8ms/step - loss: 49.4276 - mae: 5.7342 - val_loss: 42.5047 - val_mae: 5.9131
Epoch 20/20
```

07.9
Actual: 5.150000095367432, Predicted: 1.7980719804763794
Actual: 5.389999866485596, Predicted: 0.9483557939529419
Actual: 5.730000019073486, Predicted: 1.0637255907058716
Actual: 6.949999809265137, Predicted: 0.5955535173416138
Actual: 9.3100004196167, Predicted: 2.161482810974121
Actual: 5.360000133514404, Predicted: 1.0635462999343872
Actual: 6.429999828338623, Predicted: 1.2615386247634888
Actual: 9.359999656677246, Predicted: 0.5160609483718872
Actual: 13.75, Predicted: 1.12879478931427
Actual: 12.1899995803833, Predicted: 0.5555030107498169

Actual vs Predicted Pause Duration



Avec un batch size de 256



```
Actual: 5.150000095367432, Predicted: 1.6348860263824463  
Actual: 5.389999866485596, Predicted: 8.348241806030273  
Actual: 5.730000019073486, Predicted: 7.2147698402404785  
Actual: 6.949999809265137, Predicted: 9.817865371704102  
Actual: 9.3100004196167, Predicted: 2.357130289077759  
Actual: 5.360000133514404, Predicted: 7.401678562164307  
Actual: 6.429999828338623, Predicted: 6.768320560455322  
Actual: 9.359999656677246, Predicted: 9.747552871704102  
Actual: 13.75, Predicted: 7.705328464508057  
Actual: 12.1899995803833, Predicted: 9.868146896362305
```

De régression à classification

Réseau de neurones simple modèle séquentiel

```
Bin edges for 'pauseDur':
```

```
Bin 0: 5.099999904632568 to 5.430999994277954
```

```
Bin 1: 5.430999994277954 to 5.7039999008178714
```

```
Bin 2: 5.7039999008178714 to 6.067999935150146
```

```
Bin 3: 6.067999935150146 to 6.699999809265137
```

```
Bin 4: 6.699999809265137 to 7.3999998569488525
```

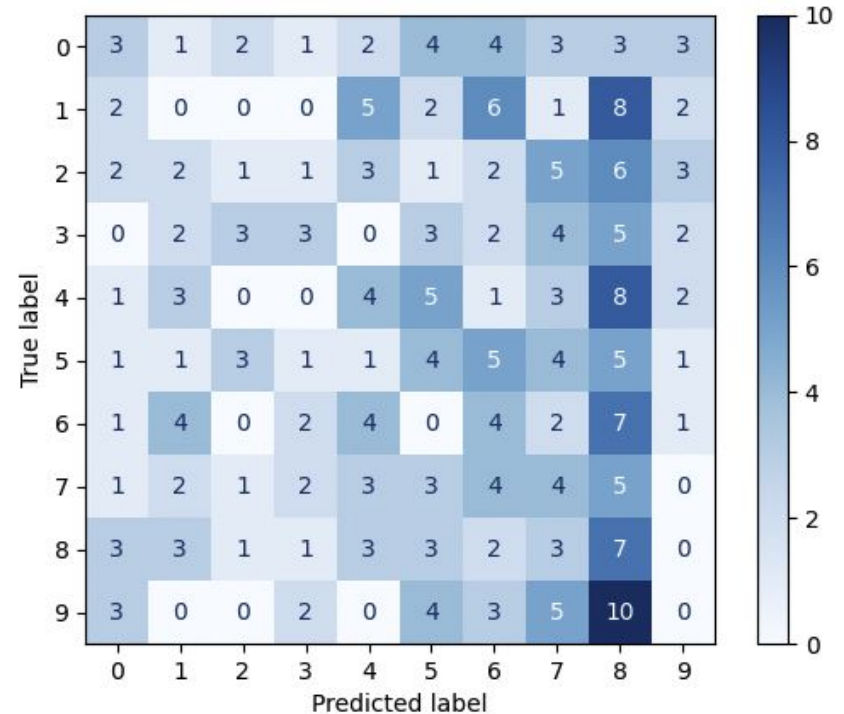
```
Bin 5: 7.3999998569488525 to 8.188000297546386
```

```
Bin 6: 8.188000297546386 to 9.135000324249267
```

```
Bin 7: 9.135000324249267 to 10.180000305175781
```

```
Bin 8: 10.180000305175781 to 12.1899995803833
```

```
Bin 9: 12.1899995803833 to 13.899999618530273
```

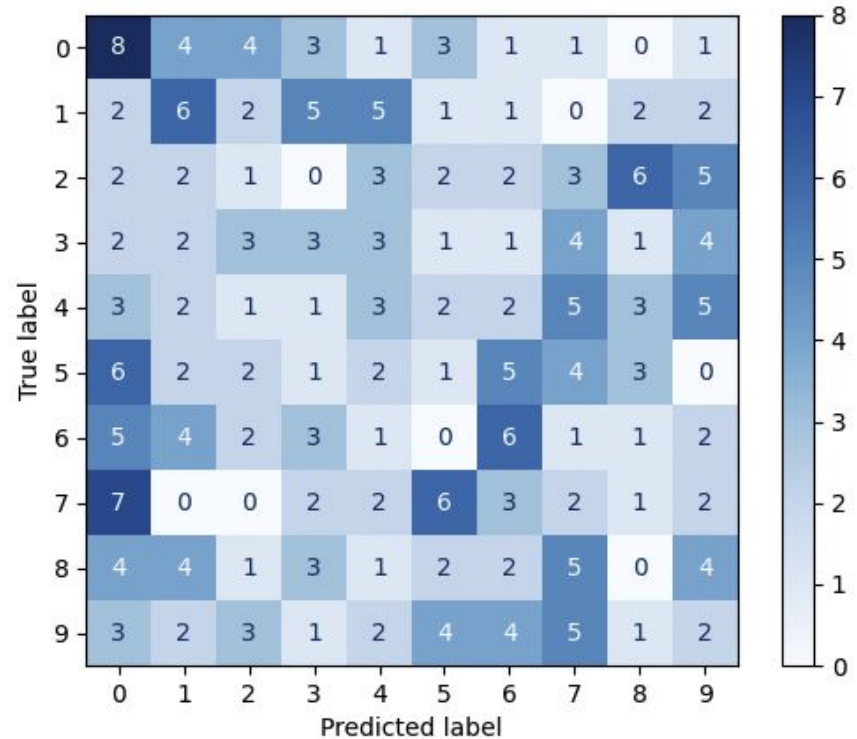


Sequential model results

```
Epoch 5/20
26/26 ————— 0s 7ms/step - accuracy: 0.2490 - loss: 2.1612 - val_accuracy: 0.0874 - val_loss: 2.3720
Epoch 6/20
26/26 ————— 0s 7ms/step - accuracy: 0.2614 - loss: 2.1072 - val_accuracy: 0.0971 - val_loss: 2.3961
Epoch 7/20
26/26 ————— 0s 7ms/step - accuracy: 0.2970 - loss: 2.0505 - val_accuracy: 0.0777 - val_loss: 2.4358
Epoch 8/20
26/26 ————— 0s 7ms/step - accuracy: 0.3159 - loss: 2.0002 - val_accuracy: 0.1019 - val_loss: 2.4586
Epoch 9/20
26/26 ————— 0s 7ms/step - accuracy: 0.3278 - loss: 1.9641 - val_accuracy: 0.1262 - val_loss: 2.5036
Epoch 10/20
26/26 ————— 0s 7ms/step - accuracy: 0.3089 - loss: 1.9539 - val_accuracy: 0.1165 - val_loss: 2.5591
Epoch 11/20
26/26 ————— 0s 7ms/step - accuracy: 0.3394 - loss: 1.8873 - val_accuracy: 0.1117 - val_loss: 2.5896
Epoch 12/20
26/26 ————— 0s 7ms/step - accuracy: 0.3578 - loss: 1.8547 - val_accuracy: 0.1165 - val_loss: 2.6144
Epoch 13/20
26/26 ————— 0s 8ms/step - accuracy: 0.3841 - loss: 1.7992 - val_accuracy: 0.1019 - val_loss: 2.6562
Epoch 14/20
26/26 ————— 0s 7ms/step - accuracy: 0.3985 - loss: 1.7536 - val_accuracy: 0.0874 - val_loss: 2.6916
Epoch 15/20
26/26 ————— 0s 7ms/step - accuracy: 0.4021 - loss: 1.7346 - val_accuracy: 0.0971 - val_loss: 2.7394
Epoch 16/20
26/26 ————— 0s 7ms/step - accuracy: 0.4311 - loss: 1.7028 - val_accuracy: 0.1068 - val_loss: 2.7779
Epoch 17/20
26/26 ————— 0s 7ms/step - accuracy: 0.4675 - loss: 1.6217 - val_accuracy: 0.0971 - val_loss: 2.8024
Epoch 18/20
26/26 ————— 0s 7ms/step - accuracy: 0.4670 - loss: 1.6238 - val_accuracy: 0.1165 - val_loss: 2.8756
Epoch 19/20
26/26 ————— 0s 7ms/step - accuracy: 0.4851 - loss: 1.5638 - val_accuracy: 0.0971 - val_loss: 2.8882
Epoch 20/20
26/26 ————— 0s 7ms/step - accuracy: 0.4798 - loss: 1.6086 - val_accuracy: 0.0971 - val_loss: 2.9452
9/9 ————— 0s 46ms/step - accuracy: 0.1188 - loss: 2.8491
Test Loss: 2.885045051574707
Test Accuracy: 0.11627907305955887
```


Random Forest Classifier

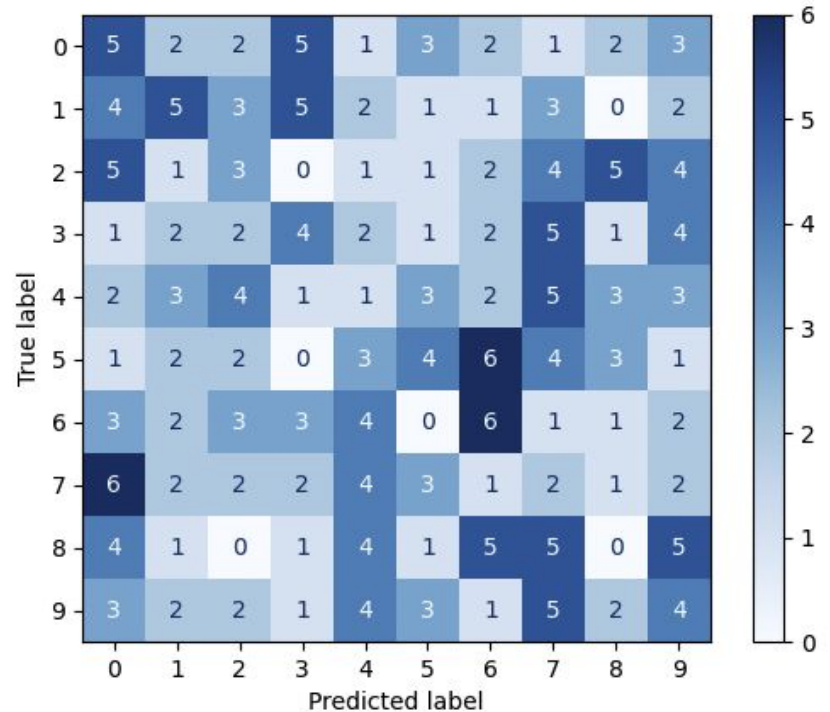
Accuracy: 0.12



Avec les pauses AVANT les bursts

Random forest:

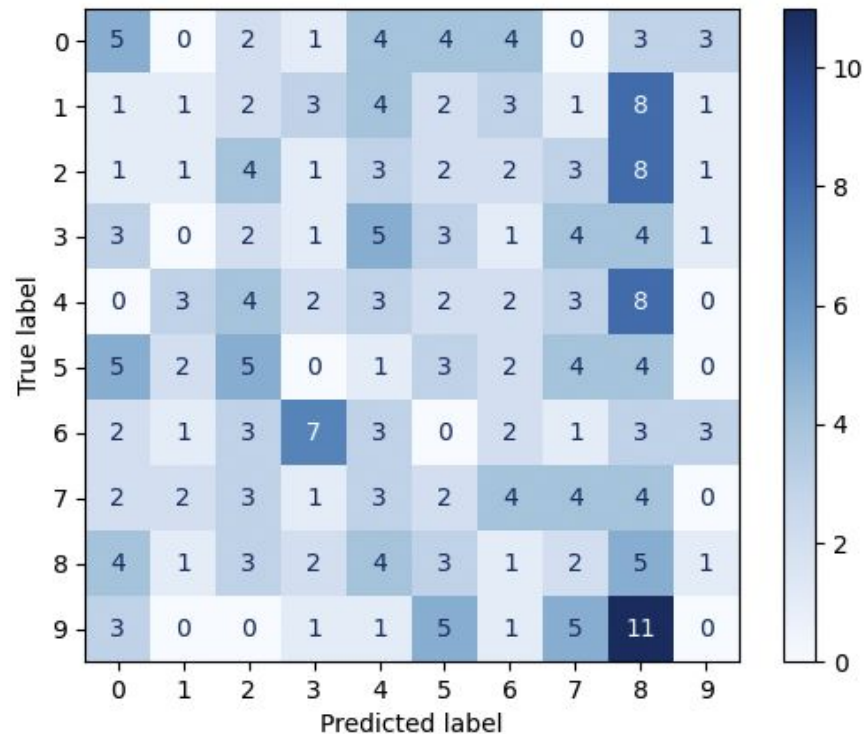
Test Accuracy: 0.13178294573643412



Sequential model avec pauses AVANT les bursts

Test Loss: 2.8643276691436768

Test Accuracy: 0.10852713137865067

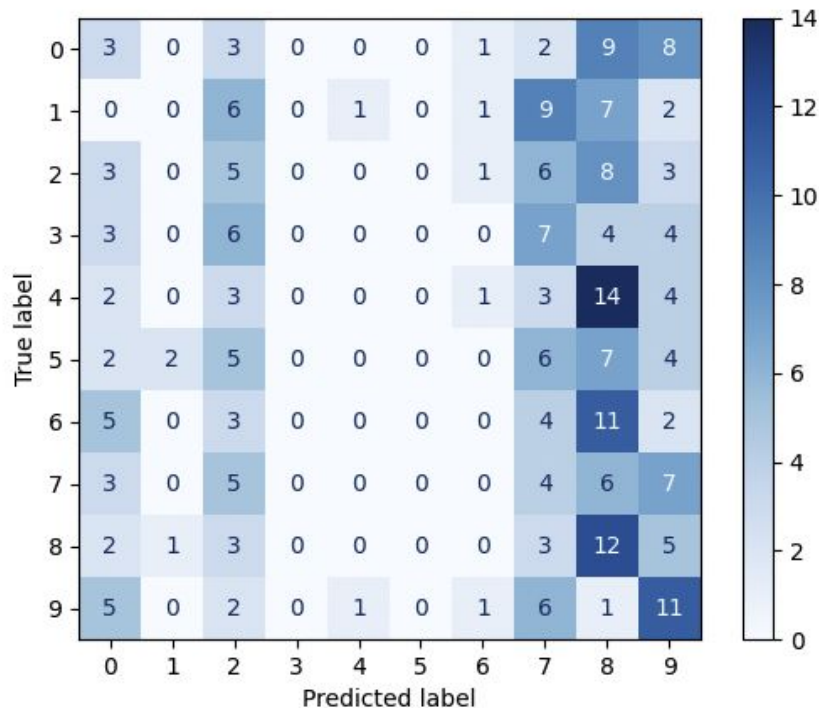


Sequential model avec pauses AVANT les bursts

On a ajouté un dropout rate et early stopping:

Test Loss: 2.2997472286224365

Test Accuracy: 0.13565891981124878



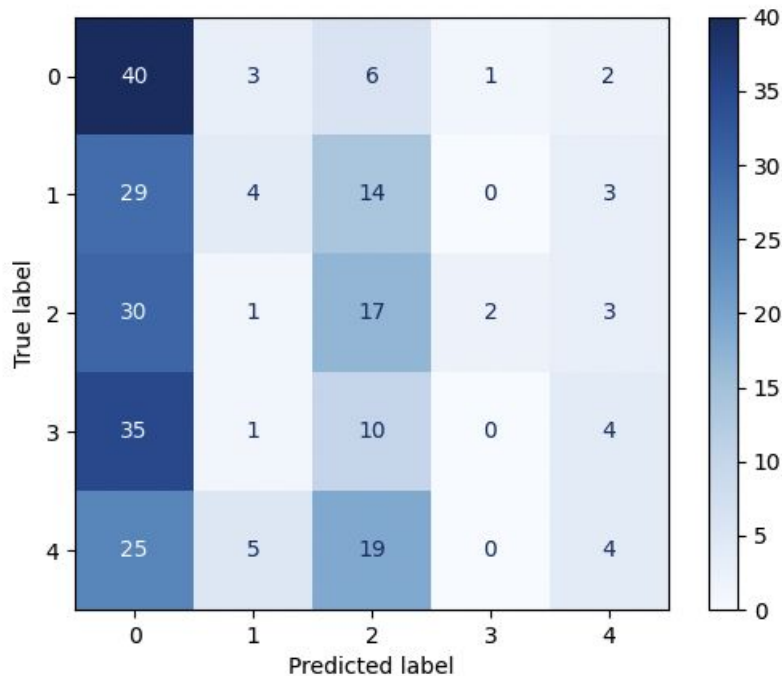
Si on essaye avec 5 classes à la place, dropout rate and early stopping

Test Loss: 1.6147950887680054

Test Accuracy: 0.25193798542022705

Bin edges for 'pauseDur':

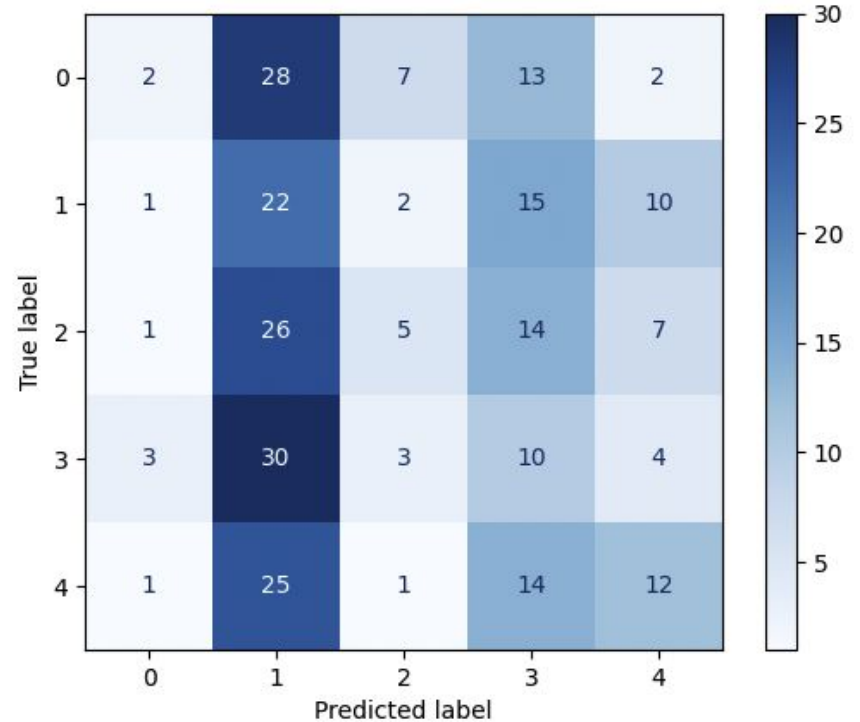
```
Bin 0: 5.099999904632568 to 5.7039999008178714  
Bin 1: 5.7039999008178714 to 6.699999809265137  
Bin 2: 6.699999809265137 to 8.188000297546386  
Bin 3: 8.188000297546386 to 10.180000305175781  
Bin 4: 10.180000305175781 to 13.899999618530273
```



Même test, avec la pause d'après

Test Loss: 1.6137007474899292

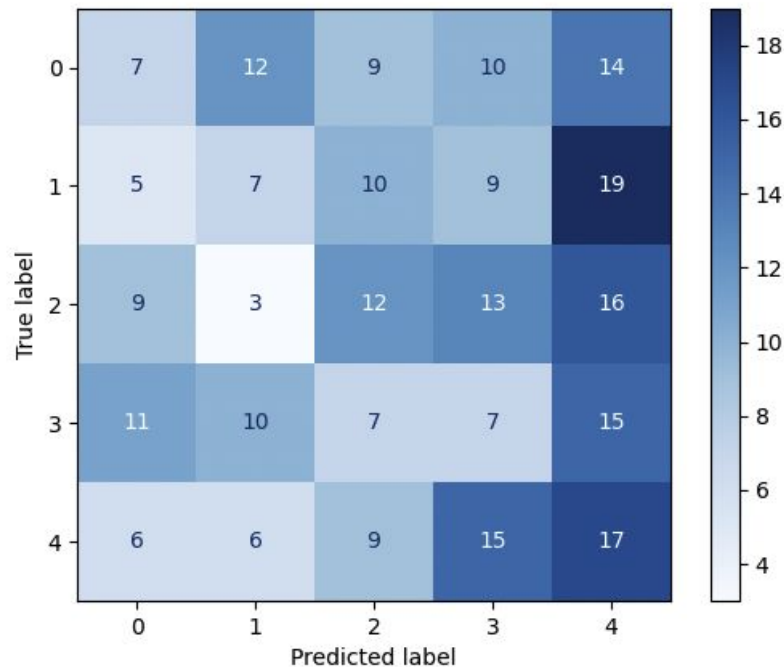
Test Accuracy: 0.19767442345619202



Pause d'avant, sans dropout rate et early stopping

Test Loss: 2.178428888320923

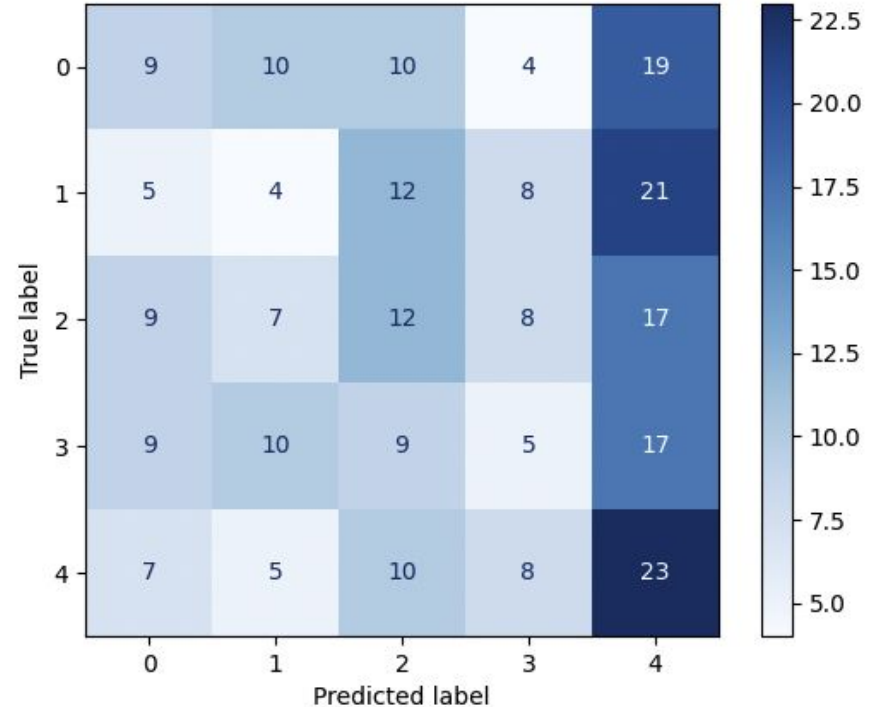
Test Accuracy: 0.19379845261573792



Ensuite pause après, sans early stopping ou dropout rate

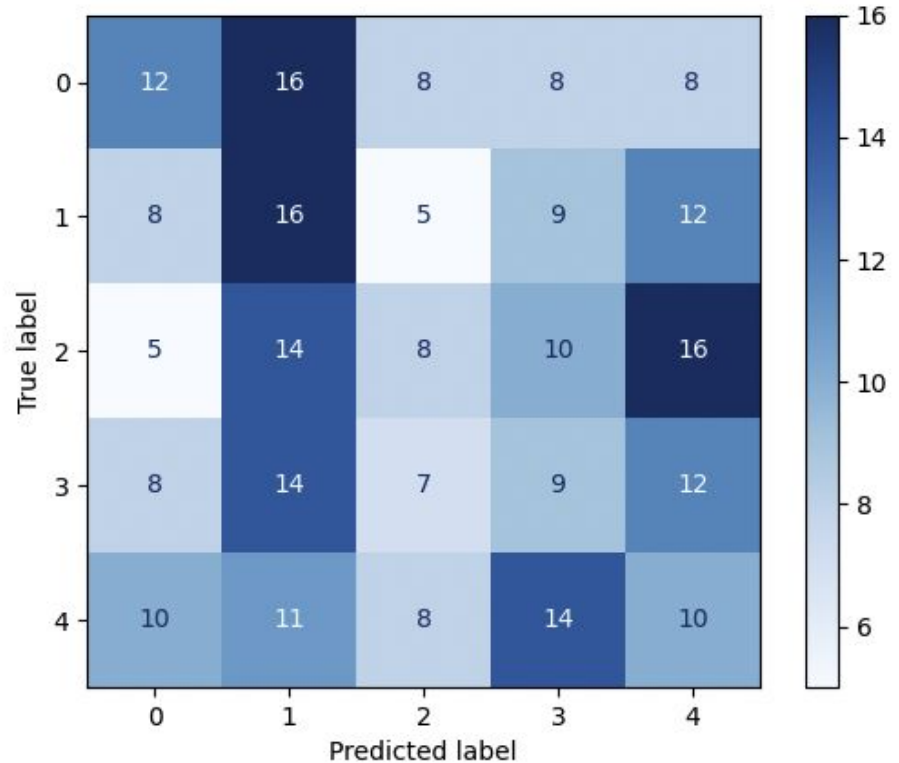
Test Loss: 2.207338809967041

Test Accuracy: 0.20542635023593903



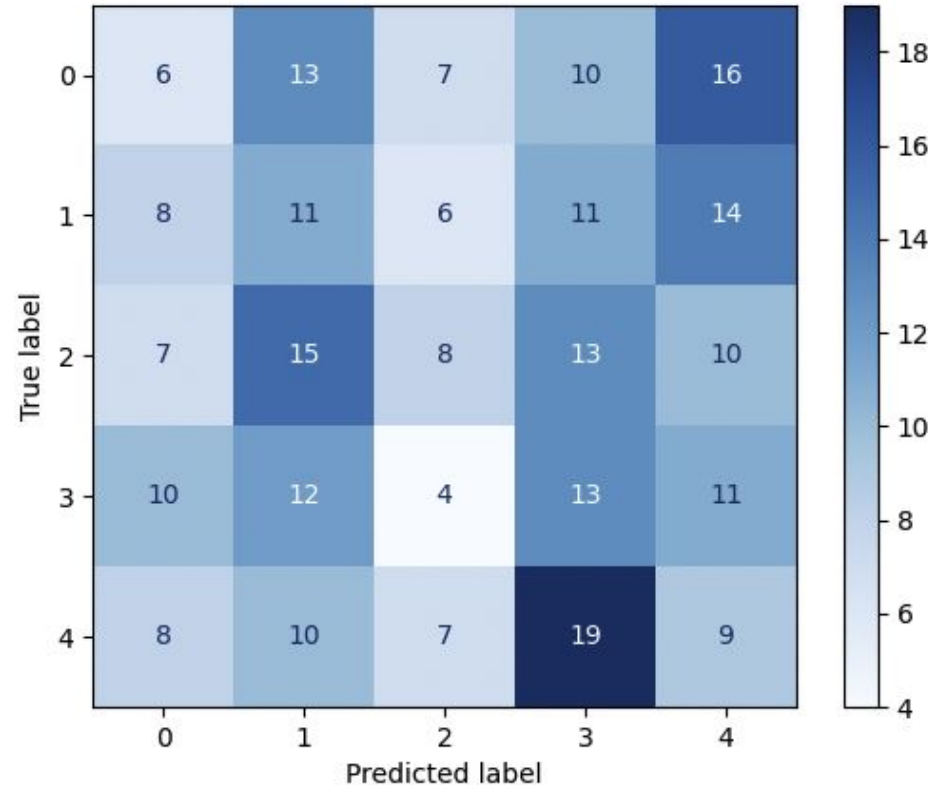
Pause avant, gradient boosting, 5 catégories

Test Accuracy: 0.2131782945736434



Pause après, gradient boosting

Test Accuracy: 0.1821705426356589



Conclusion

Meilleurs résultats : réseau de neurones simple, modèle séquentiel, dropout rates, early stopping et 5 classes (accuracy 0.25). Résultats toujours peu élevés.

Après tous ces essais, il semblerait que les pauses sont plus influencées par le burst suivant et non le précédent. Nous avons de meilleurs résultats quand la pause rattachée à un burst est celle qui précède ce dernier.

Reste à essayer : utiliser CamemBert pour les données textuelles et autres transformers...