Detecting the difficulty level of French texts

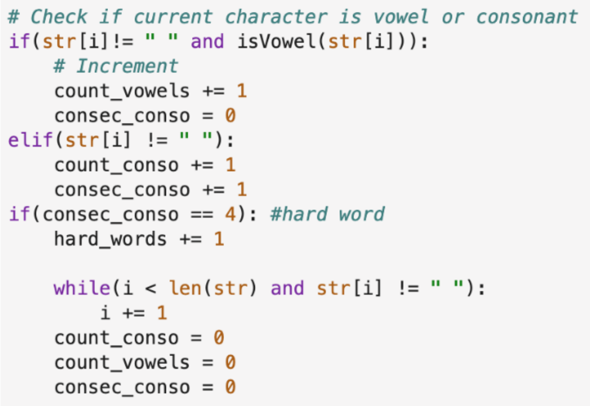
The idea

The concept is based on the creation of a model for English speakers learning French. This model is capable of predicting the difficulty of a text written in French, by classifying them according to European language proficiency levels, from A1 to C2. The aim is to make this technology usable, for example, in a recommendation system that would suggest texts, such as recent press articles, that correspond precisely to the user's level of linguistic competence.

The idea is to suggest texts containing mainly words known to the learner, while incorporating a few unknown words to encourage learning and continuous improvement. For example, for a user with an A1 level in French, it would be inappropriate to present them with a B2 text, as they would have difficulty understanding it. The aim is therefore to adapt the reading content to each level to optimize learning.

First simple model

We relied on word phonetics to predict the text's difficulty. The method was to count the number of vowels and consonants, then assign a score per word based on recurrence. Then a score is assigned for each word based on their composition and the consequential consonants.



```
# Check if current character is vowel or consonant
if(str[i]!= " " and isVowel(str[i])):
    # Increment
    count_vowels += 1
    consec_conso = 0
elif(str[i] != " "):
    count_conso += 1
    consec_conso += 1
if(consec_conso == 4): #hard word
    hard_words += 1

    while(i < len(str) and str[i] != " "):
        i += 1
    count_conso = 0
    count_vowels = 0
    consec_conso = 0
```

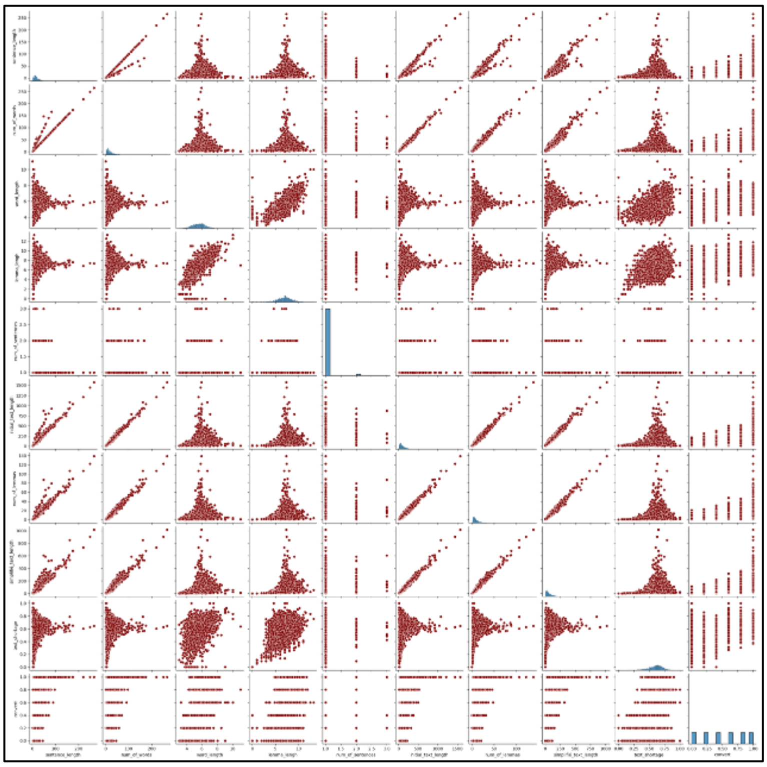
The advantage of this model is its ease of modification. It is possible to modify the score assigned to a particular composition of letters and thus adapt it to its use. We can see this on this part of the code.

However, the fact that the model doesn't take into account the length of the text to compute the score makes it obsolete. A long sentence with an easy difficulty will have a bigger score than a short sentence and a higher difficulty.

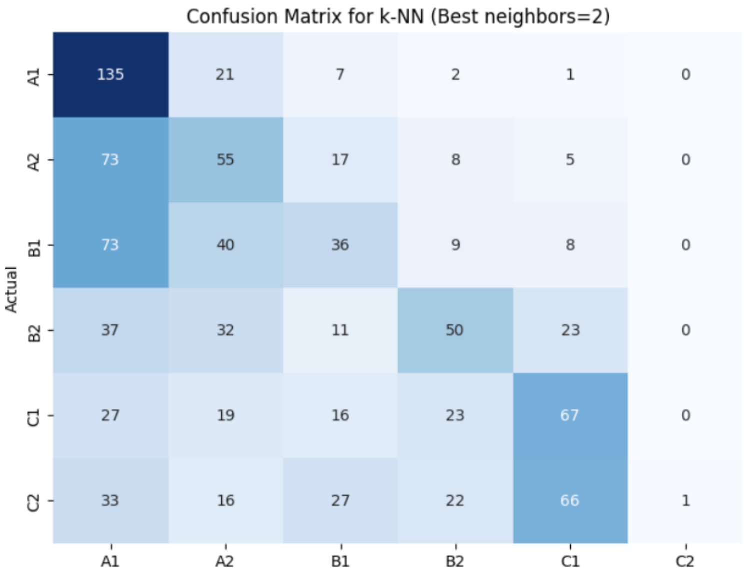
Second model

The second model is relatively more complex. It relies on the characteristics of the text before passing it through a Gradient Boosting Regressor.

We tried to find a correlation between different text characteristics and difficulty but obtained nothing significant. There was mostly a correlation between features, but this did not allow us to predict difficulty accurately.



k-NN model



The k-NN model is feature proximity and similarity. It is sensitive to noise in the data and to the curse of dimensionality, which can adversely affect its ability to classify complicated sentences well.

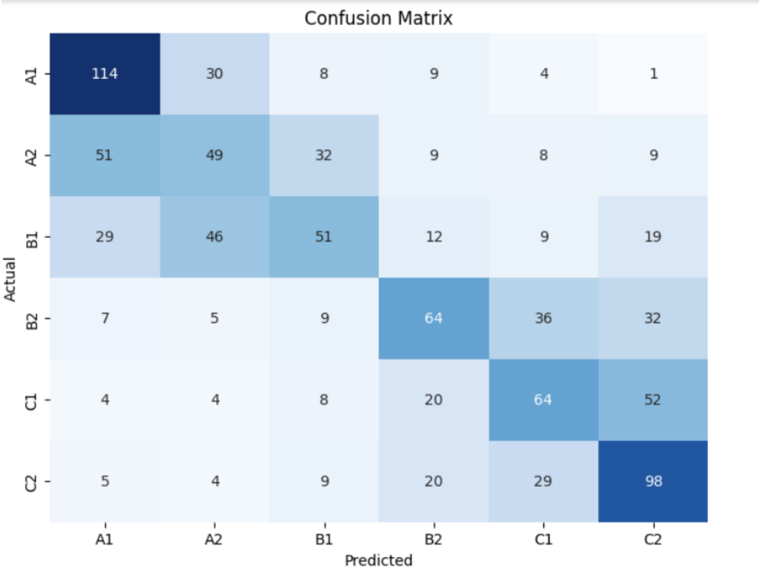
If complicated sentences (high levels C1, C2) have features that are more dispersed in the feature space, k-NN may struggle to find good "neighbors" for classification.

Logistic Regression

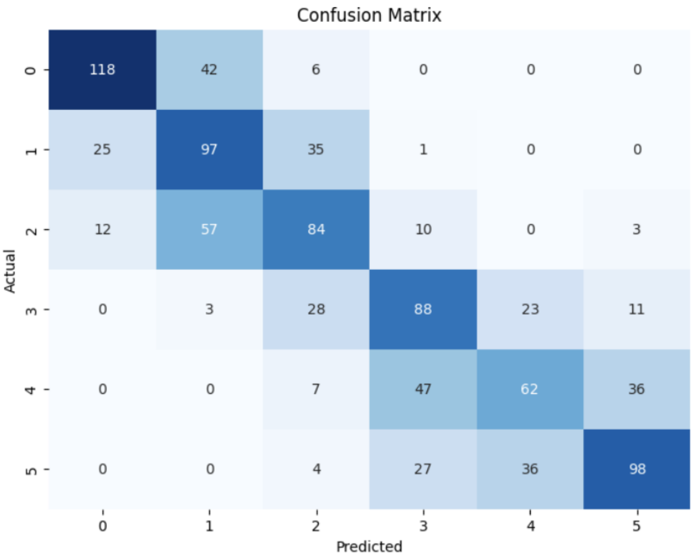
Logistic regression is a linear model that may be best suited to linear or near-linear decision frontiers. It may therefore work better for medium-difficulty sentences (B1, B2) if they lie more clearly on either side of a linear boundary in feature space.

Low-difficulty (A1, A2) and high-difficulty (C1, C2) sentences may not be so well separated linearly, which would explain why the model performs less well for these classes.

Logistic regression may perform better with well-defined, quantifiable features, and could therefore handle intermediate complexity better than extremes.



CamemBERT model



Class 0 (A1): The model correctly predicts 118 instances of A1, with a notable confusion of 42 instances predicted as A2, indicating a difficulty in distinguishing between basic levels.

Class 1 (A2): There is an improvement on the previous matrix with 97 correct predictions, but 35 instances are confused with B1, suggesting that the boundaries between A2 and B1 levels are not clearly captured by the model.
Class 2 (B1): With 84 correct predictions and 57 confusions with B2, the model still shows difficulties in differentiating the intermediate levels.
Class 3 (B2): The model correctly classified 88 instances of B2, but there is significant confusion with C1 (28 instances), highlighting the challenge of accurately classifying more advanced levels.
Class 4 (C1): There is a mixed performance with 47 correct predictions and 62 instances confused with C2, suggesting that the distinguishing features between C1 and C2 are not sufficiently captured by the model.
Class 5 (C2): The model is most accurate with this class, with 98 correct predictions and less confusion compared to other classes, although 36 instances were incorrectly classified as C1.

Comparison with k-NN

The CamemBERT model appears to have better overall accuracy than the k-NN model, especially for extreme classes (A1, C2), which may be due to CamemBERT's ability to capture complex linguistic contexts.
The k-NN model showed high confusion for intermediate and advanced classes, probably due to its sensitivity to noise and the spatial distribution of features, whereas CamemBERT, as a language-based model, can better distinguish contextual nuances.

Comparison with logistic regression

Logistic regression tended to work better for medium difficulty levels but struggled with extremes. In contrast, CamemBERT shows improvement for extreme levels but shares similar challenges with medium to high levels.
CamemBERT may have a better generalization on extremes thanks to its deep contextual understanding of language, whereas logistic regression, being a linear model, may be limited by the complexity of linguistic features that are not linearly separable.

In conclusion, CamemBERT demonstrates a superior ability to capture the contextual complexity of language, resulting in a better classification of difficulty levels, especially at the extremes, compared to k-NN and logistic regression models. However, it shares difficulties with logistic regression in clearly separating intermediate levels. Confusion between intermediate and advanced classes remains a challenge and could benefit from additional training strategies, such as fine-tuning hyperparameters or boosting training data for these specific levels.

Model	Precision	Recall	F1 Score	Accuracy
First model	0.35	0.34	0.35	0.34
Second model	-	-	-	0.39
k-NN	0.47	0.35	0.31	0.35
Logistic Regression	0.45	0.46	0.45	0.45
CamemBERT	0.58	0.56	0.56	0.60

For the first we didn't pay much attention to it because it was more of a test than really a try to have a good score, even though the second model worked better than the k-NN model for accuracy. This can be explained by the fact that we didn't try a lot to improve the K-NN because we switched on the CamemBERT, and it immediately worked much better than the K-NN.

k-NN:

The k-Nearest Neighbors model displays a precision of 0.47, a recall of 0.35 and an F1 score of 0.31, with a better precision of 0.35. Precision is relatively better than recall, suggesting that the model is more selective but lacks the sensitivity to capture all true positive cases. The F1 score, which is a harmonic mean of precision and recall, is the lowest among the three models analyzed, which may indicate an imbalance between precision and recall.

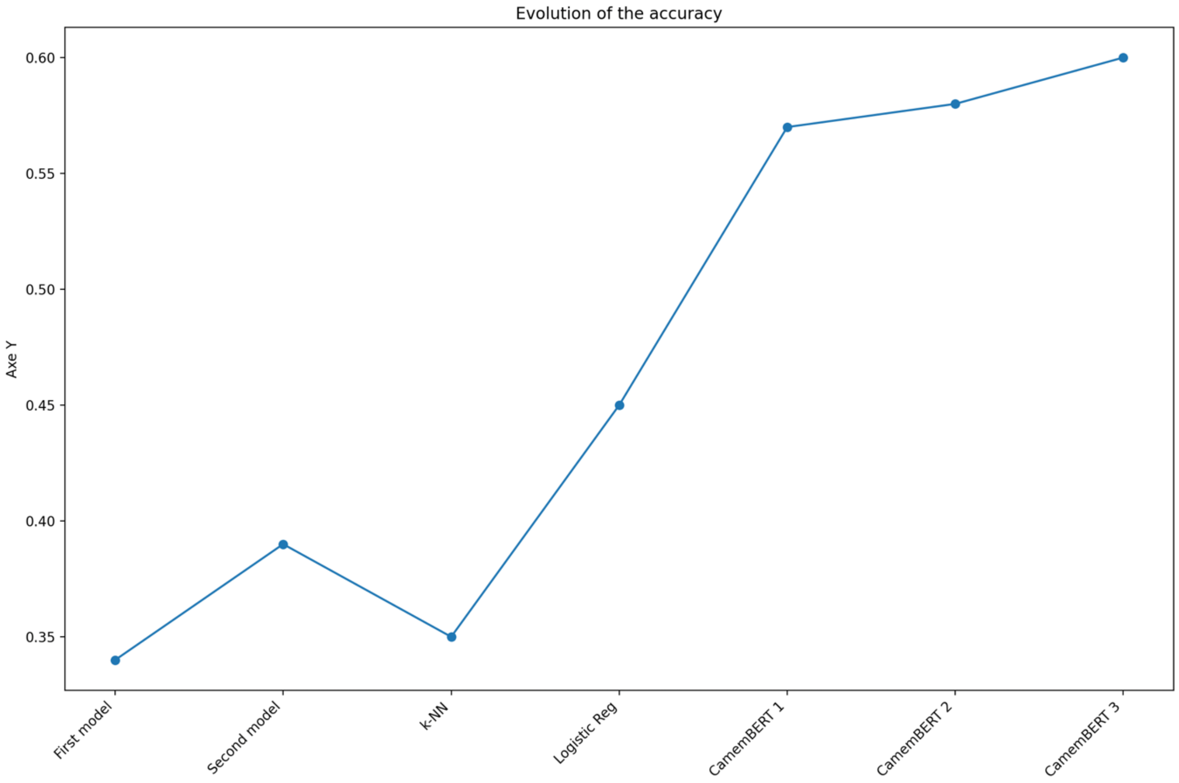
Logistic regression:

Logistic regression shows an overall better performance with precision, recall and F1 score relatively balanced around 0.45 and also the best precision at 0.45. This indicates that the model maintains a balance between precision and the ability to identify all relevant cases. The F1 score equal to the other metrics reflects this balance.

CamemBERT:

The CamemBERT model outperforms other models with the highest scores in all metrics: precision of 0.58, recall of 0.56, F1 score of 0.56, and best accuracy at 0.60. These results indicate that the CamemBERT model is both accurate and sensitive, effectively capturing true positive cases while maintaining a high rate of correct predictions. The high F1 score demonstrates that the model is well balanced and performs consistently well.

In summary, the CamemBERT model stands out as the best performing of the models evaluated, which is probably due to its ability to better understand the context and nuances of natural language. Logistic regression also showed strong results despite its simplicity and could be preferred for applications where model interpretability is important. The k-NN model, while useful for certain use cases, seems to be less suited for this specific task compared to models based on deep learning and logistic regression.


Evolution of the accuracy
Axe Y
0.60
0.55
0.50
0.45
0.40
0.35
First model
Second model
k-NN
Logistic Reg
CamemBERT 1
CamemBERT 2
CamemBERT 3