Classification des Langues Européen

Oubaha Oumaima
Ecole national des sciences appliqués d'Al hoceima
oumaima.oubaha@etu.uae.ac.ma

Résumé :

Dans un monde de plus en plus interconnecté, la détection automatique des langues joue un rôle essentiel dans des applications telles que les traducteurs automatiques et la gestion de contenu multilingue. Ce projet se concentre sur la classification des langues européennes en utilisant des modèles d'apprentissage automatique.

Après un nettoyage soigneux des données pour éliminer les bruits et les valeurs manquantes, un prétraitement a été réalisé par la conversion des textes en représentations vectorielles grâce à la méthode TF-IDF et label encode,tandis que la colonne cible des langues a été transformée en valeurs numériques grâce au Label Encoder.Ensuite, plusieurs algorithmes tels que la régression logistique,KNN,K Means ,SVM,arbre de décision,random Forest gradient boosting(lightboost,Xg Boost),réseau de neurone ont été évalués en utilisant des

métriques telles que l'accuracy, le rappel et la précision. Enfin, le modèle offrant les meilleures performances a été approfondi pour examiner sa capacité à détecter avec précision les langues européennes.

I. Introduction

La classification des langues est une tâche essentielle dans le domaine du traitement automatique des langues (TAL). Avec l'augmentation exponentielle des données multilingues sur Internet, la capacité à identifier automatiquement la langue d'un texte est cruciale pour de nombreuses applications, notamment les moteurs de recherche, les assistants vocaux et les systèmes de traduction. Ce projet vise à développer un modèle performant capable de prédire la langue d'un texte donné

parmi un ensemble de langues européennes. L'objectif principal est d'explorer divers modèles d'apprentissage automatique pour déterminer la meilleure approche en termes de précision, de robustesse et de temps de calcul.

II. Travaux liés

Plusieurs travaux ont contribué à l'avancement de la classification automatique des langues. En 2017, Smith et al. [1] ont montré que l'utilisation de TF-IDF avec des algorithmes comme la Régression Logistique et le Naïve Bayes permet une classification efficace, en particulier pour les textes courts.

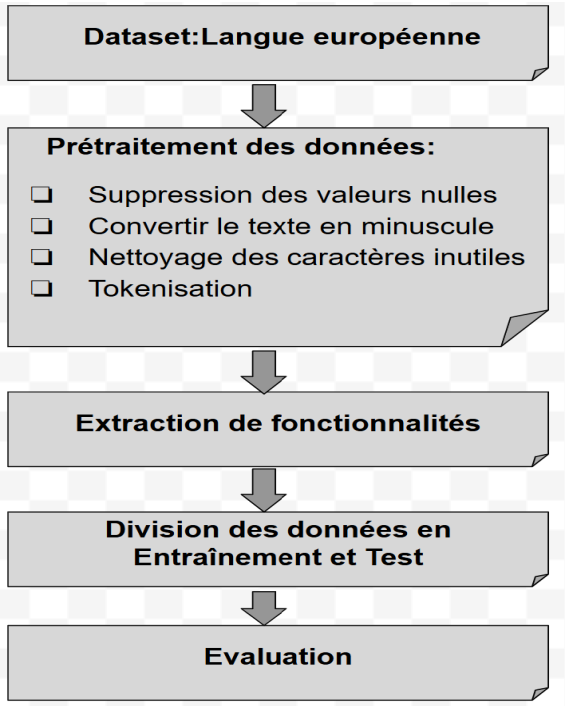
En 2019, Wang et Zhou [2] ont démontré l'efficacité des modèles d'ensemble, tels que Random Forest et XGBoost, pour gérer les différences linguistiques dans des textes multilingues. Enfin, en 2021, Lee et Park [3] ont mis en évidence que les réseaux neuronaux, bien que plus coûteux en calcul, offrent une excellente généralisation, même sur des ensembles de données déséquilibrés.

Ces travaux soulignent l'importance d'un prétraitement solide et de modèles avancés pour obtenir des résultats fiables en classification de langues.

Mon projet s'appuie sur ces travaux en intégrant des approches de traitement automatique des langues pour la détection des langues européennes, tout en proposant des améliorations spécifiques adaptées à ce contexte.

III. méthodologie:

La méthodologie utilisée pour le développement d'un modèle de détection des langues suit une approche efficace. Les données sont nettoyées et prétraitées, puis transformées en une représentation numérique à l'aide de techniques de tokenisation et de vectorisation. Le jeu de données est divisé en ensembles d'entraînement et de test pour évaluer les performances du modèle. Plusieurs algorithmes d'apprentissage automatique sont ensuite évalués pour choisir celui offrant les meilleures performances. Le schéma suivant illustre les étapes suivies durant ce processus.

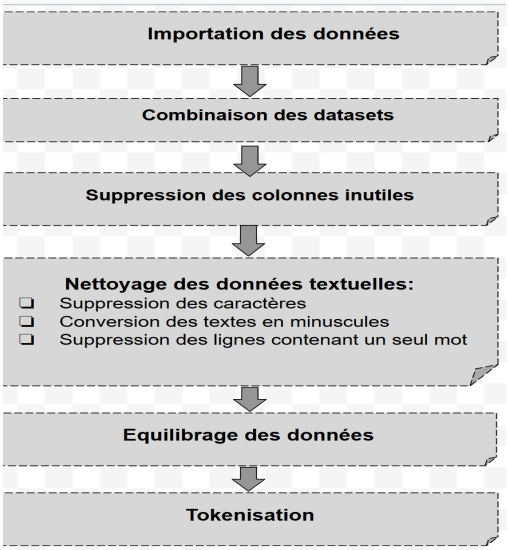


1)Collection des données:

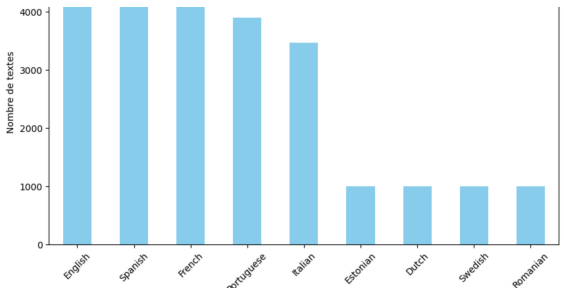
Les données utilisées pour ce projet proviennent de plusieurs jeux de données multilingues importés depuis Kaggle, chacun contenant des textes annotés dans différentes langues européennes. Ces datasets ont été combinés pour former un corpus de 28 000. Chaque ligne inclut un texte ainsi qu'une étiquette indiquant la langue correspondante. Cette approche a permis de disposer d'un ensemble de données varié et équilibré, essentiel pour garantir une performance robuste des modèles de classification appliqués.

2)Nettoyage et prétraitement des données:

Plusieurs étapes de nettoyage ont été appliquées pour garantir la qualité et la pertinence des données avant leur utilisation dans les modèles de machine learning. Tout d'abord, les colonnes inutiles ou non informatives ont été supprimées afin de simplifier le dataset et de se concentrer sur les informations essentielles. Ensuite, les textes ont été convertis en minuscules pour assurer une uniformité dans le traitement textuel. Les lignes contenant un seul mot ont été éliminées, car elles ne fournissent pas suffisamment de contexte pour une classification fiable. Ensuite, un équilibrage des données a été réalisé afin de corriger toute disparité dans la répartition des classes.Une fois ces étapes terminées, les textes ont été tokenisés, transformant chaque phrase en une liste de mots (tokens) exploitables pour les étapes suivantes. Cela a permis de préparer les données textuelles pour les modèles de machine learning.



Avant équilibrage:



Langue	
English	5005
Spanish	4921
French	4865
Portuguese	3901
Italian	3459
Estonian	1000
Dutch	1000
Swedish	1000
Romanian	1000

Après équilibrage:

```
Langue
dutch         2000
italian       2000
swedish       2000
spanish       2000
romanian      2000
estonian      1999
french        1999
portuguese    1998
english       1997
```

3)Transformations des données:

Pour permettre aux modèles de machine learning de traiter les données textuelles, il a été nécessaire de transformer ces textes en représentations numériques. Cette étape a été réalisée grâce à la méthode TF-IDF (Term Frequency-Inverse Document Frequency).

1-Définition de TF-IDF :

TF-IDF est une technique de pondération utilisée pour représenter un texte sous forme de vecteurs numériques. Elle attribue à chaque mot une importance basée à la fois sur sa fréquence dans un document (TF) et sur sa rareté dans l'ensemble des documents (IDF). Cette méthode permet de mettre en avant les mots pertinents tout en atténuant l'impact des mots très fréquents mais peu informatifs (comme "le", "et", "de").

L'équation de TF-IDF est donnée par: $TF\text{-}IDF(t,d)=TF(t,d)\times IDF(t)$

TF (Term Frequency) : fréquence d'un terme t dans un document d, calculée comme :

$$TF(t,d) = \frac{\text{Nombre d'occurrences de } t \text{ dans } d}{\text{Nombre total de mots dans } d}$$

DF (Inverse Document Frequency) : mesure de la rareté d'un terme t dans l'ensemble des documents, calculée comme :

$$IDF(t) = \log\left(\frac{N}{1+df(t)}\right)$$

où N est le nombre total de documents et df(t) est le nombre de documents contenant le terme t.

2-Label Encoding :

Pour traiter la colonne cible (Langue), un encodage numérique a été appliqué à l'aide de la méthode Label Encoding. Chaque langue a été transformée en une valeur numérique unique, rendant les données exploitables par les modèles d'apprentissage supervisé.

Ces transformations ont permis de préparer efficacement les données pour l'entraînement et la validation des modèles de classification des langues.

4)Division des données:
Après le prétraitement et la transformation des données en un format exploitable, celles-ci ont été divisées en deux ensembles distincts : un ensemble d'entraînement et un ensemble de tests. L'ensemble d'entraînement, représentant 80 % des données, a été utilisé pour ajuster les modèles d'apprentissage automatique. Les 20 % restants ont été alloués à l'ensemble de tests pour évaluer les performances des modèles sur des données inédites, garantissant ainsi une évaluation objective et fiable. Cette division permet de s'assurer que le modèle généralise bien au-delà des données d'entraînement.
5)les modèles:

-La régression logistique:
est un algorithme de classification qui permet de prédire la probabilité qu'une instance appartienne à une classe spécifique. Il utilise une fonction logistique pour transformer les résultats obtenus en valeurs comprises entre 0 et 1, rendant ainsi l'algorithme adapté aux tâches de classification binaire. e. Dans le contexte de la prédiction d'une langue , la régression logistique est employée pour prédire la partenance d'une donnée a une classe .
il est caractérisé par sa fonction sigmoïde suivante :

$$h_\theta(X) = \frac{1}{1 + e^{-\theta X}}$$

-SVM (Support Vector Machine)
L'algorithme SVM (Support Vector Machine) est largement utilisé pour la classification multi-classe et la régression. Son objectif principal est de trouver une frontière optimale qui sépare les différentes classes dans un espace multidimensionnel. Pour cela, SVM sélectionne les vecteurs support, qui sont les points les plus influents dans la définition de la frontière de séparation. Grâce à cette approche, SVM minimise les erreurs et maximise la distance entre les classes, ce qui permet une meilleure généralisation sur de nouvelles données. De plus, SVM est particulièrement adapté pour traiter des données non linéaires, grâce à des méthodes comme le noyau qui permettent d'étendre la capacité de séparation des classes dans des espaces plus complexes.

$$\max_{w,b} \quad \frac{1}{2}||w||^2 \quad \text{subject to} \quad y_i(w \cdot x_i + b) \geq 1 \quad \forall i$$

- **w** : Vecteur normal à l'hyperplan (poids attribués aux caractéristiques).
- **b** : Terme de décalage (offset).
- **xi** : Point de données d'entrée.
- **yi** : Étiquette de classe (valeur +1 ou -1).
- **||w||^2** : Norme du vecteur w, servant à maximiser la distance entre les classes.

Cette relation est utilisée pour trouver l'hyperplan optimal qui maximise la distance entre les différentes classes tout en minimisant les erreurs de classification.

-XGBoost (Extreme Gradient Boosting):

XGBoost est un algorithme d'apprentissage supervisé basé sur la technique de boosting. Il combine plusieurs modèles faibles (arbre de décision) pour améliorer la performance globale. XGBoost utilise des approches avancées telles que :

- ☐ Régularisation : Prévient le surapprentissage.
- ☐ Gradient Boosting : Améliore les performances en optimisant la perte grâce à une descente du gradient.
- ☐ Optimisation parallèle : Permet une exécution rapide sur de grandes bases de données.
- ☐ Pruning : Réduction de la complexité des arbres pour éviter le surapprentissage.

-LightGBM (Light Gradient Boosting Machine)

LightGBM est un autre algorithme de boosting qui vise à améliorer l'efficacité et la rapidité du modèle en traitant des données en feuilles plutôt qu'en nœuds complets. Ses principaux avantages sont :

- Optimisation parallèle : Utilisation efficace des ressources pour des modèles distribués.
- Réduction de la consommation en mémoire : Exploitation des données de manière plus concise.
- Support des données discrètes : Traitement efficace des données catégoriques.
- Moins de surapprentissage : Capacité à gérer des modèles plus complexes tout en évitant les biais excessifs.

-K-Nearest Neighbors (KNN)

KNN est un algorithme de machine learning largement utilisé pour la classification et la régression. Il fonctionne en trouvant les **k** plus proches voisins d'une instance donnée dans un espace multidimensionnel de caractéristiques. L'idée principale est de mesurer la similarité entre les points en utilisant une fonction de distance, comme la distance euclidienne ou Manhattan, pour déterminer les plus proches voisins. Ensuite, en se basant sur ces voisins, KNN fait une prédiction en utilisant la majorité des classes pour la classification ou la moyenne des valeurs pour la régression. Ce modèle est simple à implémenter, mais sa précision dépend directement de la valeur de **k** et de la distance choisie pour mesurer la similarité.

-Arbre de decision:

Les arbres de décision sont des modèles de machine learning utilisés pour la classification et la régression. Ils construisent un ensemble de règles binaires sous forme d'un arbre, où chaque nœud interne correspond à une décision basée sur une feature spécifique. À chaque nœud, une condition est appliquée pour séparer les données en sous-groupes, ce qui continue jusqu'à atteindre des nœuds terminaux représentant les classes ou les valeurs prédictives. L'objectif est d'optimiser la prédiction en utilisant des critères tels que l'entropie ou la réduction de Gini pour déterminer les meilleures split points. Ces critères permettent de mesurer l'hétérogénéité ou l'incertitude des données à chaque nœud.

L'entropie est définie comme :

$$\text{Entropie}(S) = -\sum_{i=1}^{k} p_i \log_2(p_i)$$

où pi est la probabilité d'un exemple appartenant à une classe spécifique parmi k classes.

Le Gini, quant à lui, mesure la distribution des classes à un nœud, et est défini comme :

$$Gini(S) = 1 - \sum_{i=1}^{k} p_i^2$$

-Random forest:

Les forêts aléatoires (**Random Forest**) sont une méthode d'apprentissage automatique basée sur l'agrégation d'un grand nombre d'arbres de décision pour effectuer des tâches de classification, de régression ou d'autres types de prédictions. Chaque arbre dans une forêt aléatoire est construit de manière indépendante en utilisant un sous-échantillon aléatoire des données et un sous-ensemble aléatoire de features. L'ensemble des arbres est ensuite combiné pour obtenir une prédiction finale en utilisant une approche d'agrégation, comme la majorité des votes pour la classification ou la moyenne pour la régression.

-K_means:

Les K-Means est un algorithme de clustering utilisé pour regrouper les données en différents clusters. Il vise à partitionner l'espace de données en plusieurs sous-groupes de manière à minimiser la distance intra-cluster, c'est-à-dire réduire la somme des distances quadratiques entre chaque point de données et le centre de son cluster. La formule mathématique utilisée dans K-Means est:

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2$$

J :représente la fonction de coût (ou objectif), qui est minimisée.

k :est le nombre de clusters.

C_i : est le cluster **i**.

x est un point de données.

μ_i est le centre du cluster **i**.

||x - μ_i||² est la distance euclidienne entre un point **x** et le centre **μ_i** du cluster **i**.

IV. Résultats et discussion

1)Les métriques de performances:

Les métriques de performance jouent un rôle clé dans l'évaluation des modèles d'apprentissage automatique. Dans ce projet, nous avons utilisé l'accuracy, la précision, le

rappel et le score F1 pour analyser et comparer les performances des modèles appliqués à la classification des langues européennes. Ces métriques, issues de la matrice de confusion, permettent de mesurer la qualité des prédictions et de déterminer les axes d'amélioration nécessaires.

- Matrice de confusion:

La matrice de confusion est un outil crucial pour évaluer les performances d'un modèle de classification. Elle divise les observations en quatre catégories principales : vrais positifs (TP), vrais négatifs (TN), faux positifs (FP) et faux négatifs (FN). Ces valeurs permettent de comparer les étiquettes de classes, en distinguant les prédictions correctes des erreurs. **Les vrais positifs** sont les cas correctement identifiés comme positifs, tandis que **les faux positifs** sont des cas prédits positifs mais qui sont en réalité négatifs. **Les vrais négatifs** représentent les cas correctement identifiés comme négatifs, et **les faux négatifs** sont des cas prédits négatifs mais qui sont en réalité positifs.

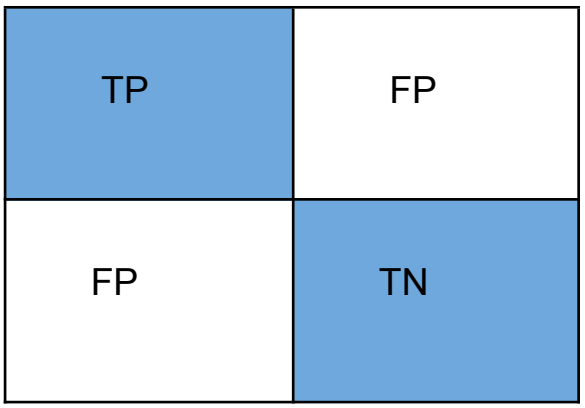


figure 4: Matrice de confusion

- **Accuracy**:

L'accuracy représente la proportion de prédictions correctes sur l'ensemble des données. Elle est calculée à l'aide de la relation suivante :

$$\text{Accuracy} = \frac{\text{Nombre de prédictions correctes}}{\text{Nombre total d'échantillons}}$$

- **Précision** :La précision mesure la capacité du modèle à éviter les faux positifs, en évaluant la proportion des prédictions positives correctes. Elle est donnée par la formule:

$$\text{Precision} = \frac{\text{Vrais positifs (TP)}}{\text{Vrais positifs (TP)} + \text{Faux positifs (FP)}}$$

- **Rappel (Recall)** :Le rappel, aussi appelé sensibilité, évalue la capacité du modèle à identifier correctement tous les échantillons d'une classe donnée. Il est calculé comme suit :

$$\text{Recall} = \frac{\text{Vrais positifs (TP)}}{\text{Vrais positifs (TP)} + \text{Faux négatifs (FN)}}$$

- **F1 Score** :Le score F1 est la moyenne harmonique de la précision et du rappel. Il est particulièrement utile pour évaluer les modèles dans les cas où les données sont déséquilibrées. La formule est la suivante :

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

2) Comparaison algorithmique et évaluation des performances:

Les résultats obtenus montrent que les différents algorithmes testés, notamment la Régression Logistique, le Naïve Bayes, le Support Vector Machine (SVM), le Random Forest, le Gradient Boosting, le K-Nearest Neighbors (KNN) et les réseaux de neurones (MLP), ont démontré des performances globalement élevées lorsqu'ils ont été appliqués à la tâche de classification des langues européennes.

	precision	recall	f1-score	support
0	1.00	0.99	0.99	400
1	0.96	0.99	0.98	399
2	1.00	0.98	0.99	400
3	0.97	0.97	0.97	400
4	0.93	0.94	0.94	400
5	0.95	0.97	0.96	400
6	0.99	0.99	0.99	400
7	0.96	0.92	0.94	400
8	1.00	1.00	1.00	400
accuracy			0.97	3599
macro avg	0.97	0.97	0.97	3599
weighted avg	0.97	0.97	0.97	3599

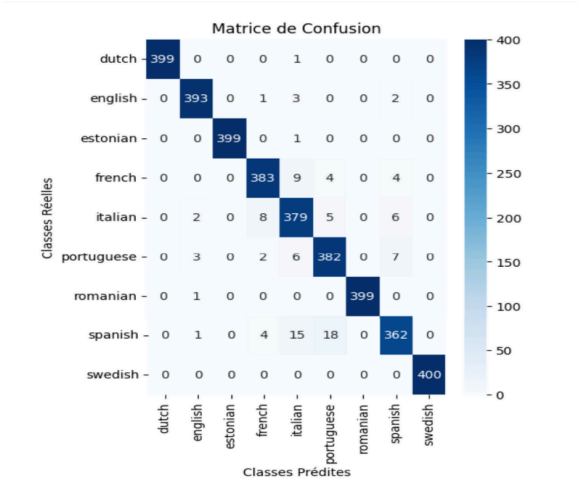


figure 5: Rapport de classification et Matrice de confusion de la régression logistique

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

           0       1.00      1.00      1.00       400
           1       0.97      0.97      0.97       399
           2       1.00      1.00      1.00       400
           3       0.97      0.94      0.95       400
           4       0.89      0.95      0.92       400
           5       0.93      0.95      0.94       400
           6       1.00      1.00      1.00       400
           7       0.95      0.90      0.92       400
           8       1.00      1.00      1.00       400

    accuracy                           0.97      3599
   macro avg       0.97      0.97      0.97      3599
weighted avg       0.97      0.97      0.97      3599
```

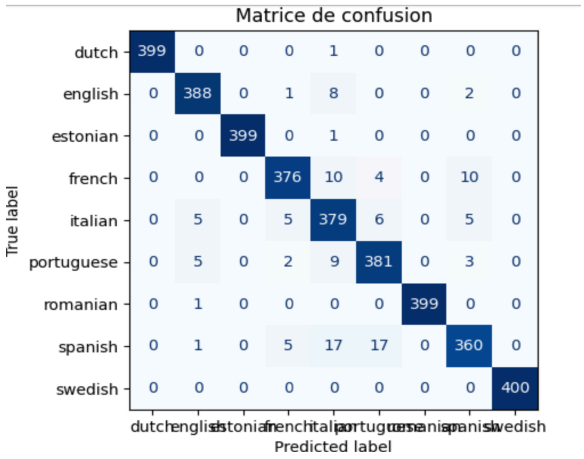


figure 6 : Rapport de classification Matrice de confusion de random_forest

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

           0       1.00      0.99      1.00       400
           1       0.98      0.98      0.98       399
           2       1.00      0.99      1.00       400
           3       0.97      0.97      0.97       400
           4       0.93      0.94      0.94       400
           5       0.92      0.96      0.94       400
           6       1.00      0.99      1.00       400
           7       0.96      0.91      0.94       400
           8       1.00      1.00      1.00       400

    accuracy                           0.97      3599
   macro avg       0.97      0.97      0.97      3599
weighted avg       0.97      0.97      0.97      3599
```

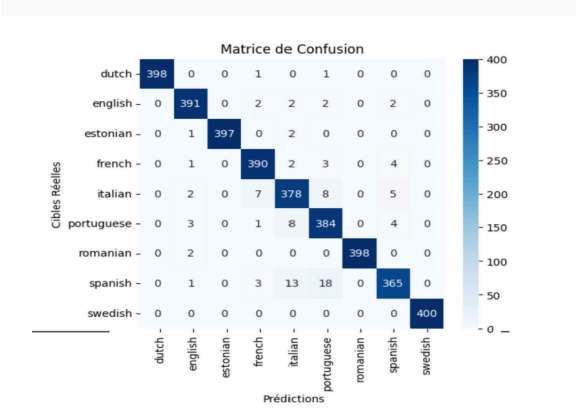


Figure 7: rapport de classification et matrice de confusion de svm

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

           0       0.97      0.99      0.98       400
           1       0.95      0.99      0.97       399
           2       1.00      0.96      0.98       400
           3       0.96      0.97      0.97       400
           4       0.97      0.94      0.95       400
           5       0.97      0.94      0.96       400
           6       1.00      0.99      0.99       400
           7       0.92      0.96      0.94       400
           8       1.00      1.00      1.00       400

    accuracy                           0.97      3599
   macro avg       0.97      0.97      0.97      3599
weighted avg       0.97      0.97      0.97      3599
```

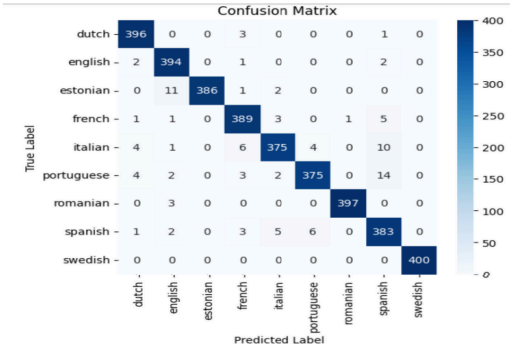


Figure 8: Rapport de classification et matrice de confusion de k_Means

```
Accuracy : 0.9560989163656571
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99       400
           1       0.98      0.95      0.97       399
           2       1.00      0.99      0.99       400
           3       0.97      0.92      0.94       400
           4       0.84      0.94      0.89       400
           5       0.93      0.94      0.93       400
           6       1.00      0.99      1.00       400
           7       0.91      0.89      0.90       400
           8       1.00      1.00      1.00       400

    accuracy                           0.96      3599
   macro avg       0.96      0.96      0.96      3599
weighted avg       0.96      0.96      0.96      3599
```

figure 9: Rapport de classification et matrice deconfusion de lightGBM

```
Accuracy : 0.9580439010836344
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99       400
           1       0.99      0.94      0.96       399
           2       1.00      0.99      0.99       400
           3       0.97      0.93      0.95       400
           4       0.91      0.92      0.92       400
           5       0.85      0.98      0.91       400
           6       0.99      0.99      0.99       400
           7       0.94      0.88      0.91       400
           8       1.00      1.00      1.00       400

    accuracy                           0.96      3599
   macro avg       0.96      0.96      0.96      3599
weighted avg       0.96      0.96      0.96      3599
```

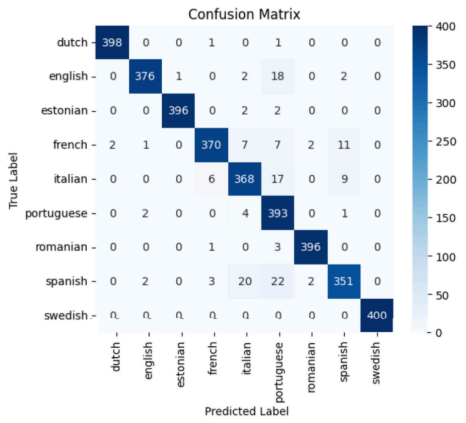


figure 10: Rapport de classification et matrice de confusion de xgboost

Classification Report:

	precision	recall	f1-score	support
0	1.00	0.59	0.74	400
1	0.95	0.29	0.45	399
2	1.00	0.55	0.71	400
3	0.85	0.29	0.44	400
4	0.25	0.73	0.37	400
5	0.26	0.79	0.39	400
6	1.00	0.39	0.56	400
7	0.86	0.17	0.28	400
8	1.00	0.64	0.78	400
accuracy			0.49	3599
macro avg	0.80	0.49	0.52	3599
weighted avg	0.80	0.49	0.52	3599

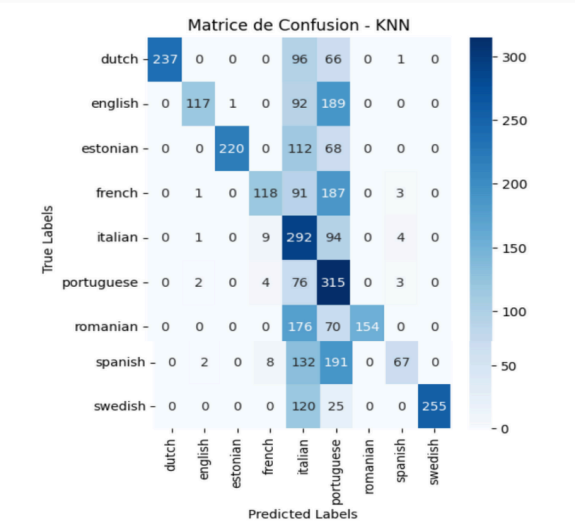


Figure 11:Rapport de classification et matrice de confusion de KNN

	precision	recall	f1-score	support
0	0.99	0.99	0.99	400
1	0.95	0.92	0.94	399
2	0.98	0.98	0.98	400
3	0.94	0.85	0.90	400
4	0.81	0.93	0.86	400
5	0.91	0.93	0.92	400
6	0.97	0.98	0.98	400
7	0.89	0.83	0.86	400
8	1.00	1.00	1.00	400
accuracy			0.94	3599
macro avg	0.94	0.94	0.94	3599
weighted avg	0.94	0.94	0.94	3599

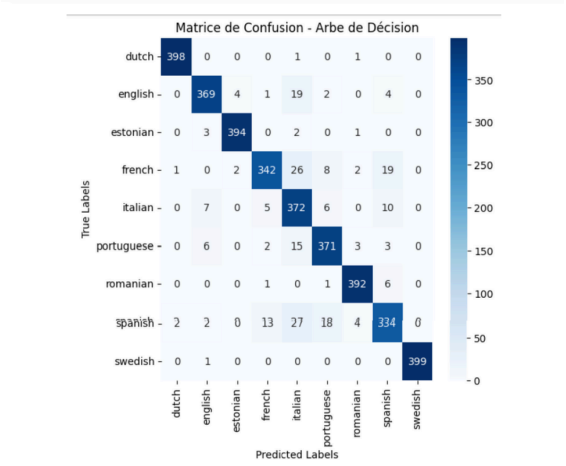


figure 12: Rapport de classification et matrice de confusion d'arbre de décision

```
Accuracy : 0.9705
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       400
           1       0.97      0.98      0.98       399
           2       1.00      1.00      1.00       400
           3       0.97      0.96      0.96       400
           4       0.94      0.94      0.94       400
           5       0.93      0.94      0.94       400
           6       1.00      1.00      1.00       400
           7       0.93      0.92      0.93       400
           8       1.00      1.00      1.00       400

    accuracy                           0.97      3599
   macro avg       0.97      0.97      0.97      3599
weighted avg       0.97      0.97      0.97      3599
```

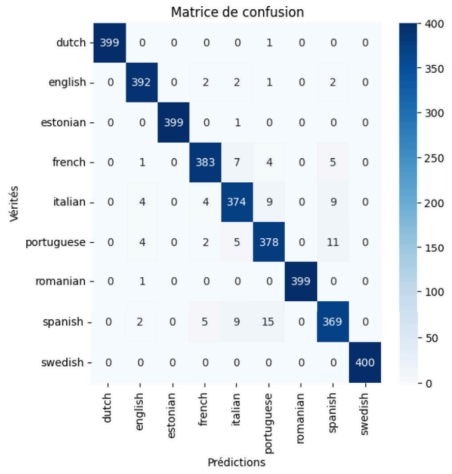


figure 13: Rapport de classification et matrice confusion de réseau de neuron

modèle	Accuracy	Précision	Recall	F1-score
Régression logistique	0.9747	0.97	0.97	0.97
Random Forest	0.9708	0.97	0.97	0.97
SVM	0.9708	0.97	0.97	0.97
K-Means	0.9708	0.97	0.97	0.97
LightGBM	0.9561	0.96	0.96	0.96
XGBoost	0.9580	0.96	0.96	0.96
KNN	0.9461	0.95	0.95	0.95
Arbre de Décision	0.9400	0.94	0.94	0.94
MLP	0.9705	0.97	0.97	0.97

table 1:résultat des métriques avant l'ajustement des paramètres

En analysant les matrices de confusion et les métriques de performance, MLP, SVM, et la Régression Logistique se distinguent avec une précision d'environ 97%, offrant un bon équilibre entre précision, rappel et score F1. Random Forest affiche également une précision de 97%, prouvant son efficacité. XGBoost et LightGBM, avec des précisions légèrement inférieures mais dépassant 95%, restent des choix fiables.

En revanche, KNN et les Arbres de Décision affichent des performances moindres, mais conviennent mieux pour des tâches simples ou nécessitant une interprétabilité accrue.

En conclusion, MLP, SVM, et la Régression Logistique sont les meilleurs modèles pour ce projet. Random Forest, XGBoost, et LightGBM constituent des alternatives solides, tandis que KNN et les Arbres de Décision peuvent être utilisés dans des contextes moins complexes.

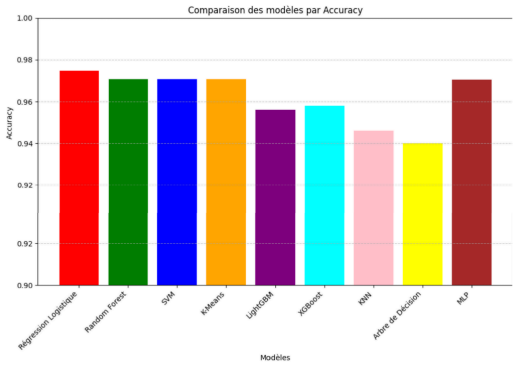


figure 14: Comparaison des modèles

3) Amélioration des modèles par ajustement d'hyperparamètres

L'ajustement d'hyperparamètres est une étape essentielle pour améliorer les performances des modèles d'apprentissage automatique. Il s'agit d'affiner les paramètres internes des algorithmes afin de les adapter aux particularités de chaque jeu de données. Dans notre projet, nous avons appliqué cette approche pour optimiser les performances de nos modèles.

Nous avons utilisé GridSearchCV et RandomizedSearchCV pour ajuster les hyperparamètres de nos modèles. GridSearchCV explore toutes les combinaisons possibles d'hyperparamètres, ce qui permet de trouver des configurations précises, mais nécessite plus de temps de computation. Quant à RandomizedSearchCV, il sélectionne aléatoirement un sous-ensemble d'hyperparamètres à tester, ce qui réduit le temps nécessaire tout en maintenant une bonne exploration des configurations. Les deux méthodes ont permis d'optimiser les modèles, chaque méthode ayant ses avantages selon les besoins en termes de précision et d'efficacité.Les résultats finaux montrent que cet ajustement précis des hyperparamètres a contribué à une amélioration significative des performances globales de nos modèles.

Modèle	accuracy	Précision	Recall	F1-score
Regression logistique	0.97	0.9718	0.9705	0.9709
Random_forest	0.9500	0.9512	0.9500	0.9505
SVM	0.9700	0.9708	0.9705	0.9706
k_Means	0.9700	0.9705	0.9703	0.9704
LightGBM	0.9659	0.9668	0.9656	0.9658
XGBoost	0.9800	0.9805	0.9795	0.9800
KNN	0.9500	0.9512	0.9501	0.9506
Arbre de decision	0.9100	09112	0.9103	0.9108

MLP	0.9700	0.9708	0.9708	0.9707

Table 2:Résultats des métriques après l'ajustement des paramètres

XGBoost se distingue comme le modèle le plus performant avec une précision remarquable de 98.00 %. Il offre une amélioration significative de la précision à 98.05 % tout en maintenant un rappel élevé à 97.95 %, ce qui se traduit par un F1-score exceptionnel de 98.00 %. Cette performance remarquable illustre sa capacité à gérer des données complexes et à optimiser les performances, particulièrement dans des situations où la précision est cruciale. Le Support Vector Machine (SVM) suit de près avec une précision de 97.00 %, démontrant des résultats impressionnants en termes de précision (97.08 %), rappel (97.05 %) et F1-score (97.06 %). Ces modèles montrent une supériorité évidente dans la prédiction des données, s'avérant particulièrement adaptés aux situations nécessitant une haute fidélité des résultats.

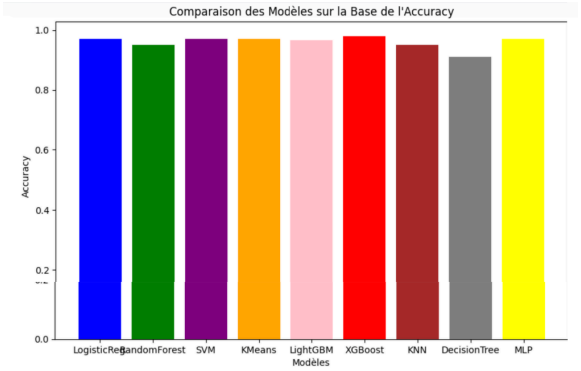


figure 14: Comparaison des modèles après l'ajustement des paramètres

V. Limitations et Défis

Les limitations et défis rencontrés dans ce projet de classification des langues européennes incluent plusieurs aspects. Tout d'abord, bien que les modèles testés, tels que SVM, XGBoost et les réseaux de neurones, aient démontré des performances élevées, ils restent sensibles à la qualité et à la quantité des données. Les jeux de données déséquilibrés ou contenant des erreurs peuvent fausser les résultats, réduisant la capacité du modèle à généraliser sur des données réelles.

De plus, le temps de calcul et les ressources nécessaires pour entraîner certains modèles complexes, comme les réseaux de neurones et les algorithmes d'ensemble, constituent un défi, surtout avec de grands jeux de données. Enfin, certains modèles, bien qu'efficaces, manquent d'interprétabilité, rendant difficile l'analyse approfondie des décisions prises par le modèle.

VI. Conclusion

En résumé, la détection des langues européennes basée sur les nuances de langage offre un potentiel significatif pour comprendre et classifier les langues avec précision. Ce projet peut être utilisé dans divers domaines tels que la recherche linguistique, la gestion des données multilingues et l'amélioration des systèmes de traduction automatique. Bien que des défis

subsistent, tels que la similarité entre certaines langues et la qualité des données, les avancées dans l'apprentissage automatique permettront de surmonter ces obstacles avec le temps. À mesure que les modèles évoluent, cette technologie pourrait devenir un outil essentiel pour des applications plus complexes et un meilleur traitement des langues européennes.

VII. Travaux liés

[1]Le traitement automatique des langues : tendances et enjeux:

URL=https://journals.openedition.org/lalies/289?utm_source=chatgpt.com

[2]Les politiques linguistiques européennes et la gestion de la diversité linguistique

URL:https://shs.cairn.info/revue-langue-francaise-2010-3-page-95?lang=fr

[3]dataset langue kaggle:https://www.kaggle.com/datasets/zarajamshaid/language-identification-datasst

[4]État de l'art des technologies linguistiques pour la langue française

URL:https://hal.science/hal-03637784v1