Feature engineering

1. Introduction

Ce projet porte sur l'application du **Feature Engineering** pour la classification de textes sportifs. L'objectif est d'explorer différentes approches de transformation et d'extraction des caractéristiques des textes afin d'améliorer la performance des modèles de classification.

Le jeu de données provient du répertoire `bbcsport`, contenant des articles classés en différentes catégories sportives :

- **Football**
- **Rugby**
- **Cricket**
- **Tennis**
- **Athletics**

Nous allons explorer les données, les nettoyer, extraire des caractéristiques, tester différents modèles de classification et comparer leurs performances.

---

2. Lecture des données

Les données sont constituées de fichiers texte organisés en sous-répertoires selon leurs catégories respectives. Le processus d'importation consiste à :

- Lire les fichiers texte et les stocker dans un **DataFrame pandas** avec deux colonnes : `texte` (contenu de l'article) et `categorie` (sport associé).
- Éliminer les **doublons** (10 doublons supprimés).
- Vérifier la répartition des catégories :
  - Football : 262 articles
  - Rugby : 146 articles
  - Cricket : 121 articles
  - Tennis : 99 articles
  - Athletics : 99 articles
- **Rééquilibrage** des classes à **149 observations** par catégorie grâce au rééchantillonnage.
- Dimensions finales après nettoyage : **614 observations**.

---

3. Prédicteurs utilisés

Les caractéristiques extraites sont classées en plusieurs catégories :

1. Prédicteurs structurels

- Nombre de **caractères**, **mots**, **phrases**.
- Longueur moyenne des **mots** et **phrases**.
- Nombre de **mots courts** et **mots longs**.

2. Prédicteurs linguistiques

- Nombre de **verbes**, **noms**, **stopwords**.
- **Richesse lexicale** (rapport entre mots uniques et mots totaux).
- Nombre de **chiffres**, **majuscules**, **ponctuation**.

3. Analyse sémantique avec mots-clés

- Extraction de mots-clés spécifiques à chaque catégorie :
  - **Football** : *goal, penalty, striker, league*.
  - **Tennis** : *serve, racket, wimbledon*.
  - **Cricket** : *bowler, batsman, wicket*.
  - **Rugby** : *scrum, tackle, conversion*.
  - **Athletics** : *100m, sprint, marathon*.

4. Techniques d'extraction de texte

- **TF-IDF** (Term Frequency - Inverse Document Frequency).
- **Bag-of-Words** (simple comptage des mots).
- **Word Embeddings (Word2Vec)**.

---

4. Technique de nettoyage (Transformation)

- Conversion en **minuscules**.
- Suppression de la **ponctuation**.
- Suppression des **chiffres**.
- Tokenization et suppression des **stopwords**.
- Lemmatisation des mots (réduction à leur forme de base).

---

5. Algorithmes utilisés

Les modèles de classification testés :

1. **Régression Logistique**

2. **Random Forest**
3. **SVM (Support Vector Machine)**
4. **KNN (K-Nearest Neighbors)**

Les modèles sont testés avec différentes techniques d'extraction de texte :

- **TF-IDF**
- **Bag-of-Words**
- **Prédicteurs sélectionnés**
- **Word Embeddings**

---

6. Résultats obtenus

Les performances sont évaluées selon **F1-score** et **Accuracy** :

Technique	Logistic Regression	Random Forest	SVM	KNN
TF-IDF	**98.38%**	**98.38%**	97.56%	93.50%
Bag-of-Words	97.55%	95.94%	86.99%	65.04%
Word Embeddings	37.40%	**50.41%**	24.39%	35.77%
Prédicteurs de base	97.57%	**98.37%**	86.99%	68.29%
Prédicteurs sélectionnés	75.61%	**76.42%**	73.17%	67.48%

Analyse des résultats :

- **TF-IDF et Random Forest** sont les plus efficaces, atteignant **98.38%** de précision.
- **Bag-of-Words** fonctionne bien avec **Régression Logistique (97.55%)**, mais moins avec KNN.
- **Word Embeddings** sous-performe, probablement en raison d'un entraînement insuffisant.
- **Les Prédicteurs sélectionnés** sont utiles mais offrent des résultats inférieurs à TF-IDF.

---

7. Difficultés rencontrées

- **Déséquilibre des classes** : nécessité d’un rééchantillonnage.
- **Complexité des modèles** : Word2Vec nécessite plus de données.
- **Problèmes de surapprentissage** : nécessité d’optimiser les hyperparamètres.
- **Temps de calcul** : le nettoyage et l’extraction des caractéristiques sont gourmands en ressources.

---

8. Compétences acquises

- **Prétraitement avancé de texte** (nettoyage, tokenization, lemmatisation).
- **Application de techniques de Feature Engineering**.
- **Comparaison de plusieurs algorithmes de Machine Learning**.
- **Utilisation d'outils de visualisation et d’analyse de données**.
- **Interprétation des performances des modèles et sélection des meilleurs prédicteurs**.

Conclusion

- **Random Forest + TF-IDF** donne les meilleurs résultats.
- **Les Prédicteurs sélectionnés** permettent d'améliorer la classification sans surcharger le modèle.
- **L’utilisation de Word Embeddings** nécessite davantage de données pour être efficace.

Ce projet illustre commen

t le **Feature Engineering** améliore les performances des modèles de classification de texte. 🚀