NLP : Prédiction du sexe à partir des données personnelles

Anton Conrad

Avril 2024

1 Introduction

Nous souhaitons utiliser les données personnelles issues des documents de recensement français pour prédire le sexe des individus dont cette information est manquante. Cela est crucial car connaître le sexe des individus dans les documents de recensement permet d'analyser les structures démographiques, et étudier les disparités de genre. Cette donnée aide à comprendre les rôles sociaux et comportements liés au genre, et à suivre les évolutions sociales vers plus d'égalité. En somme, elle est essentielle pour aborder de manière informée les enjeux de justice sociale et d'équité.

1.1 Analyse des données

Nous disposons de deux jeux de données. Le premier contient des prénoms et la fréquence du sexe des personnes le portant, le second contient une colonne

"groundtruth" faite de données personnelles transcrites : informations de référence comprenant le nom, prénom, l'occupation, la relation au chef de famille ; une colonne "prediction" contenant ces même informations mais prédites ; et enfin une colonne "sex" contenant le sexe de l'individu. On note par ailleurs que le fichier de prénoms contient 6946 données, et que le fichier de données personnelles contient 241 lignes ce qui est relativement faible. On peut s'attendre à ce que l'entraînement d'un algorithme de Deep Learning ne soit pas pertinent, à moins de trouver le moyen de générer plus de données. Les données de sexe s'organisent de la sorte :

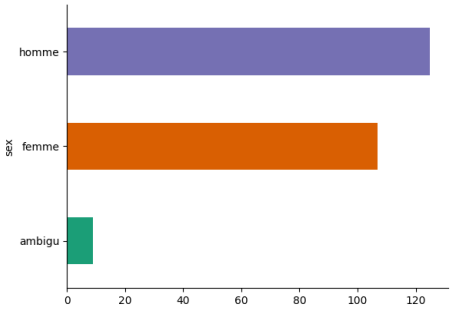

Figure 1: organisation des données par sexe

1.2 Stratégie

Nous constatons dans l'analyse du fichier de prénom que certains prénoms sont très liés à un sexe, ainsi le prénom peut servir de caractéristique discriminante forte pour un modèle de prédiction. Nous allons ainsi implémenter une methode naïve qui sur la base du prénom prédit le sexe en fonction du nombre de personnes possédant ledit prénom. Nous essayerons ensuite d'utiliser des algorithmes de Machine Learning puis des algorithmes de Deep Learning préentraînés. Cette tâche serait facilement réalisable par un être humain, le principal enjeu est l'automatisation. Nous souhaitons ainsi obtenir la meilleure précision possible, sans peur de surentraînement. Du fait du faible nombre de données, de la présence de prénoms mixtes, et des erreurs dans la colonne "prediction" du fichier de données personnelles, nous pouvons nous attendre à ce qu' aucune de ces méthodes ne soit optimale. Nous verrons donc si nous pouvons les combiner et comment nous pouvons extraire le plus informations pertinentes possibles de nos données.

2 Comparaison de différents modèles

2.1 Méthode naïve

Cette stratégie est simple, nous extrayons le prénom de chaque individu du fichier de données personnelles, et nous cherchons la proportion d'hommes et de femmes qui portent ce prénom. Cela nous donne la "probabilité" que ce prénom soit porté par un homme ou une femme et nous en déduisons une prédiction (en comparant cette "probabilité" avec la valeur seuil 0.5). Nous rencontrons deux obstacles liées à la qualité des données qui sont l'incapacité à extraire le prénom ainsi que l'absence de ce prénom du fichier de prénoms (du fait d'une retranscription erronnée dudit prénom). Dans ces cas là, nous prédisons le sexe au hasard. Cette méthode parait peu rigoureuse, pourtant elle donne la précision de : **0,95** (caculée en faisant la moyenne de 40 expériences du fait de la partie stochastique liée à la prédiction). Nous avons la confirmation que le prénom est une variable très intéressante à utiliser.

2.2 Modèles de Machine Learning

Les modèles évalués incluent la Régression Logistique, le Support Vector Machine (SVM), le K-Nearest Neighbors (KNN), l'Arbre de Décision, la Forêt Aléatoire et le Gradient Boosting.

Modèle	Précision
Régression Logistique	0.8008
SVM	0.6639
KNN	0.5187
Arbre de Décision	0.7842
Forêt Aléatoire	0.7759
Gradient Boosting	0.8174

Table 1: Précisions des différents modèles de classification

Les résultats montrent que le modèle de Gradient Boosting surpasse les autres modèles avec une précision de 0.8174, tandis que le modèle KNN affiche la plus faible précision de 0.5187. Nous avons évalué ces modèles sans traitement supplémentaire des données et en utilisant la vectorisation par fréquence de mots (plutôt que TF-IDF qui a donné des résutats similaires, moins bons, ou très légerement meilleurs) des données textuelles.

2.3 Modèles de Deep Learning pré-entraînées

La classification zero-shot est une technique de machine learning qui permet à un modèle de reconnaître des objets, des concepts ou des entités qu'il n'a jamais vus auparavant durant sa phase d'entraînement. Cette méthode est particulièrement utile dans des situations où il est impraticable de collecter des

données d'entraînement pour chaque catégorie possible.

Nous avons évalué plusieurs modèles de classification de séquences basés sur des architectures de type BERT et CAMEMBERT. Les résultats de précision, ainsi que les mesures de précision, rappel et F1-score pour chaque catégorie sont présentés ci-dessous.

Modèle	**Précision**	**Homme**		**Femme**	
		Précision	F1-score	Précision	F1-score
BERT Multilingual	0.2931	0.32	0.37	0.25	0.19
CAMEMBERT Base	0.5302	0.44	0.13	0.54	0.68
Flaubert Large	0.5560	0.51	0.58	0.61	0.53
BART Large	0.6078	0.61	0.50	0.61	0.68
CAMEMBERT Large	0.4828	0.39	0.28	0.51	0.60

Table 2: Résultats détaillés des évaluations des modèles de classification de séquences

Ces données montrent que, certains modèles comme BART Large et Flaubert Large ont démontré une certaine capacité à prédire correctement les catégories avec une précision élevée, surtout dans la catégorie "femme". Il serait intéressant de pouvoir entraîner ces modèles à notre tâche mais pour l'instant la méthode naïve et les modèles de ML semblent plus intéressants.

3 Combinaisons

3.1 Utilisation de la prédiction naïve liée au prénom

Nous créons une nouvelle colonne dans notre fichier de données personnelles qui contient la probabilité que le prénom appartienne à un homme si on réussit à l'obtenir (on utilise 0,5 sinon comme valeur par défaut). Les algorithmes de Machine Learning donnent alors les résultats suivants :

Algorithme	**Précision**
Régression Logistique	0.8091
SVM	0.7635
KNN	0.6307
Arbre de Décision	0.7552
Forêt Aléatoire	0.7718
Gradient Boosting	0.8050

Table 3: Précisions obtenues pour différents modèles de classification

Les résultats montrent que la Régression Logistique et le Gradient Boosting offrent les meilleures performances avec des précisions respectives de 0,8091 et

0,8050, tandis que le K-Nearest Neighbors présente la précision la plus faible avec 0.6307. Ces résultats suggèrent que les modèles basés sur l'ensemble et les techniques régularisées tendent à mieux performer sur ce jeu de données.

3.2 Utilisation des autres attributs

Par la suite, nous utilisons le rapport au chef de famille afin de créer une nouvelle colonne indiquant si l'on s'attend à ce que l'individu soit un homme (1), une femme (0) ou si l'on a pas l'information avec la variable "prediction" (0,5). Ainsi, si le rapport au chef de famille appartient à ['chef', 'fils', 'Chef', 'père', 'Fils', 'petit-fils', 'assisté', 'ouvrier', 'frère', 'Son', 'son'], on s'attend à ce qu'il soit un homme et réciproquement pour une femme si cet attribut appartient à ['fille', 'femme', 'épouse', 'mère', 'belle-mère', 'petite-fille', 'bru', 'mére']. On obtient alors les résultats suivants pour nos méthodes de Machine Learning :

Algorithme	**Précision**
Régression Logistique	0.8631
SVM	0.8423
KNN	0.7178
Arbre de Décision	0.8091
Forêt Aléatoire	0.8382
Gradient Boosting	0.8340

Table 4: Précisions obtenues pour différents modèles de classification

Les résultats montrent que la Régression Logistique a obtenu la meilleure performance avec une précision de 0,8631, indiquant une efficacité supérieure pour ce modèle dans le contexte actuel.

De la même manière, nous désirons utiliser l'occupation de chaque individu. Ainsi, si le suffixe de l'occupation appartient à ["ière", "euse", "ice"] ou à ["eur", "ier", "er"] nous obtenons une information supplémentaire sur le sexe de l'individu. L'utilisation simultannée de toutes ces informations donnent les résultats suivants pour nos algorithmes de Machine Learning :

Algorithme	**Précision**
Régression Logistique	0.8548
SVM (Support Vector Machine)	0.8423
KNN (K-Nearest Neighbors)	0.7261
Arbre de Décision	0.8216
Forêt Aléatoire	0.8340
Gradient Boosting	0.8382

Table 5: Précision des différents modèles de classification

L'algorithme de Régression Logistique affiche toujours la meilleure précision avec 0,8548. On note quand même une baisse de performance par rapport à auparavant, ce qui laisse penser à une mauvaise retranscription automatique de la variable occupation.

3.3 Entraînement d'un modèle de BERT

Les résultats après trois époques d'entraînement montrent une amélioration significative, avec une précision initiale sur les données d'entraînement qui passe de 48.56% à une précision parfaite de 100% sur les données de validation à la fin de l'entraînement. Les pertes d'entraînement ont diminué de manière conséquente, passant de 0.7037 à 0.2074, indiquant une adaptation réussie du modèle aux caractéristiques des données.

Époque 1:	Perte d'entraînement = 0.7037, Précision = 48.56%
	Perte de validation = 0.4330, Précision = 91.67%
Époque 2:	Perte d'entraînement = 0.3995, Précision = 88.94%
	Perte de validation = 0.0713, Précision = 100%
Époque 3:	Perte d'entraînement = 0.2074, Précision = 93.75%
	Perte de validation = 0.0791, Précision = 100%

Il est important de noter que cet algorithme n'avait pas abouti avec les données de base (probablement du fait du manque de RAM de Colab). Ces résultats démontrent l'efficacité de l'entraînement ciblé. Ils ont été obtenus en utilisant le jeu de données contenant toutes les informations extraites auparavant et montrent qu'avec un peu de processing, nous pouvons aboutir à des résultats très précis, même avec un nombre restreint de données. Enfin, un tel entraînement avec le jeu de données ne contenant pas la déduction du sexe liée à l'occupation donne des résultats similaires à la méthode naïve.

4 Conclusion

Finalement, nous répondons à la problématique initiale en ayant une méthode dont la précision est parfaite. Nous constatons le pouvoir des transformers et de l'entraînement en terme de performances, comparé aux algorithmes de Machine Learning et à une méthode simple et automatique. En revanche, nous constatons bien comme l'extraction d'informations simples peut fluidifier et améliorer cette tâche de classification. Les problèmes rencontrés auront globalement été liés au faible nombre de données et à la qualité des transcriptions automatiques.