Détection des Offres d'Emploi Frauduleuses

Projet de Traitement du Langage Naturel

Zahra Boucheta -- Roubache Islam

Mai 2025

Résumé

Le développement d'Internet a facilité la publication d'offres d'emploi en ligne, mais cela a également conduit à une augmentation des arnaques. Ce projet propose une approche basée sur le Traitement Automatique du Langage Naturel (TALN ou NLP) pour détecter automatiquement les fausses annonces d'emploi. En utilisant un ensemble de données publiques et diverses techniques de prétraitement et de classification, nous construisons un modèle capable de distinguer les annonces frauduleuses des véritables. L'objectif final est de fournir une base pour une intégration dans les systèmes de sécurité des plateformes de recrutement.

1. Introduction

Avec l'essor des plateformes de recherche d'emploi, les utilisateurs sont de plus en plus exposés à des offres frauduleuses. Ces annonces visent souvent à soutirer des informations personnelles ou financières des candidats. Automatiser la détection de telles annonces représente un défi pertinent dans le domaine du NLP, notamment face à la diversité des formulations utilisées dans les arnaques.

2. Objectifs

Ce projet vise à :

— Explorer un ensemble de données d'annonces d'emploi réelles et frauduleuses.

— Appliquer des techniques de prétraitement NLP adaptées aux données textuelles.

— Classifier les annonces en utilisant différents modèles (Régression Logistique, SVM, Random Forest).

— Évaluer les performances à l'aide de métriques standards.

— Proposer des pistes d'amélioration pour un système réel.

3. Cadre Théorique

3.1. Traitement Automatique du Langage Naturel (TALN)

Le NLP (Natural Language Processing) est un domaine de l'intelligence artificielle qui permet aux machines d'analyser, comprendre et générer du langage humain. Il repose sur des étapes telles que le nettoyage de texte, la tokenisation, la lemmatisation et la vectorisation. Le TALN est utilisé ici pour extraire des caractéristiques pertinentes permettant la détection de la fraude.

3.2. Modèles de classification

Nous avons utilisé plusieurs algorithmes de Machine Learning supervisé :

— **Régression Logistique** : efficace pour les tâches de classification binaire.

— **SVM** (Support Vector Machine) : efficace pour trouver un hyperplan de séparation optimal.

— **Random Forest** : algorithme d'ensemble basé sur de multiples arbres de décision.

4. Description du Dataset

Le jeu de données utilisé est issu de Kaggle :
`https://www.kaggle.com/datasets/shivamb/real-or-fake-fake-jobposting-prediction`

Il comprend environ 17 000 offres d'emploi réparties entre réelles et frauduleuses. Chaque ligne du dataset contient :

— Des attributs textuels (titre, description, exigences, bénéfices).

— Des informations catégorielles (type d'emploi, secteur, télétravail, etc.).

— Une étiquette cible `fraudulent` indiquant 1 (fraude) ou 0 (réelle).

5. Méthodologie

5.1. Prétraitement des Données

Le texte a été normalisé à l'aide des étapes suivantes :

— Conversion en minuscules

— Suppression des caractères spéciaux, balises HTML et chiffres

— Suppression des stopwords

— Lemmatisation avec spaCy (langue anglaise)

5.2. Modélisation

Trois modèles ont été testés :

1. **Régression Logistique** : pour sa simplicité et sa rapidité.
2. **SVM Linéaire** : bon compromis entre performance et généralisation.
3. **Random Forest** : modèle robuste capable de gérer la complexité.

La validation croisée à 5 plis (k-fold) a été utilisée pour assurer la robustesse des résultats.

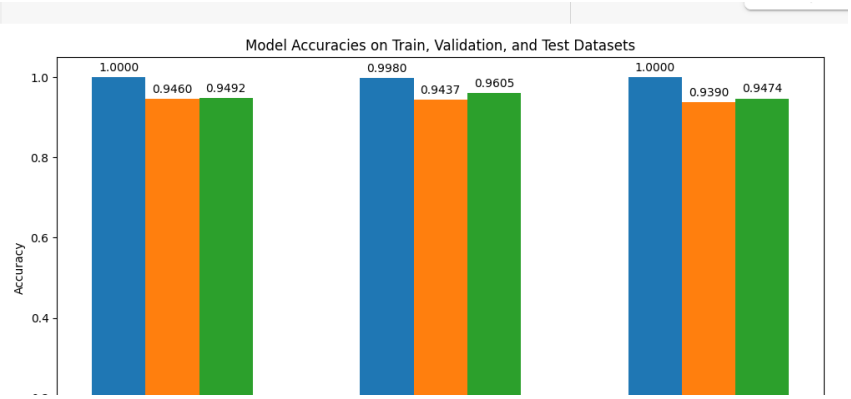

FIGURE 1 – Précision des modèles : Régression Logistique, SVM et Random Forest

5.3. Évaluation des Modèles

Les métriques choisies sont :

— **Accuracy** (Exactitude globale)
— **Precision** (Taux de vrais positifs parmi les positifs prédits)
— **Recall** (Taux de vrais positifs parmi les réels positifs)
— **F1-score** (Moyenne harmonique entre précision et rappel)
— **Matrice de confusion** (Analyse détaillée des erreurs)

6. Résultats et Analyse

— Le **SVM** a obtenu la meilleure précision (97%) avec un bon équilibre précision/rappel.
— **Random Forest** a montré des performances solides mais tendait au sur-apprentissage.
— **Régression Logistique** était plus rapide mais légèrement moins performante.

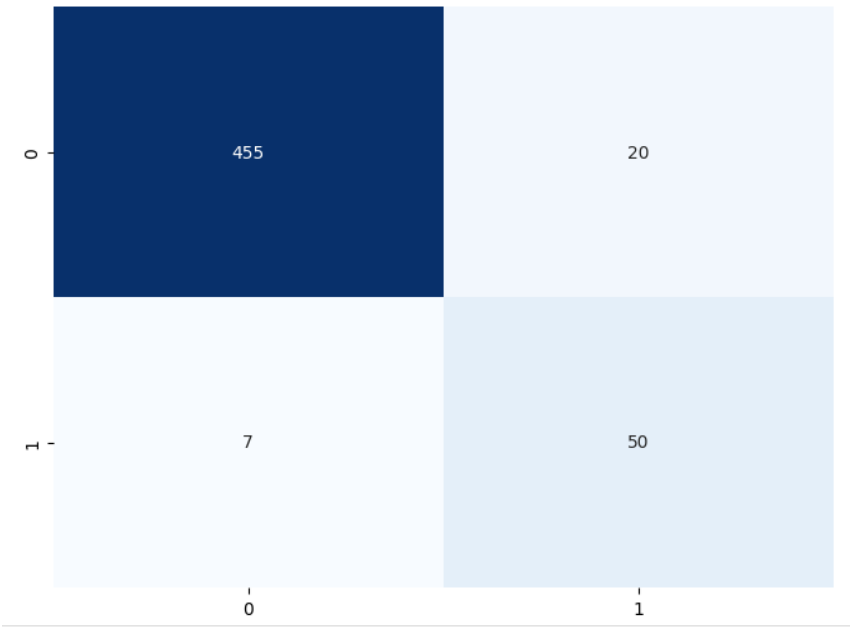

FIGURE 2 – Matrice de confusion du modèle SVM

7. Application Flask pour la Détection

Pour faciliter l'utilisation du modèle de détection, une application Flask a été développée. Elle permet à l'utilisateur de soumettre une offre d'emploi sous forme de texte qui est ensuite analysée.

- **Nettoyage du texte** : suppression des caractères spéciaux, emojis, liens, mots inutiles.
- **Prédiction** : utilisation du modèle entraîné pour classer l'annonce.
- **Interface utilisateur** : formulaire simple affichant la prédiction.

Extrait du code Flask :

```
from flask import Flask, render_template_string, request
import re, nltk, joblib
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer

nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')

app = Flask(__name__)
best_model = joblib.load('best_model.pkl')
stop_words = set(stopwords.words('english'))
```

```
lemmatizer = WordNetLemmatizer()

def clean_text(text):
    # Nettoyage du texte
    return text  # à remplacer par la vraie version nettoyée

@app.route('/', methods=['GET', 'POST'])
def index():
    prediction = None
    if request.method == 'POST':
        job_posting = request.form['job_posting']
        cleaned = clean_text(job_posting)
        pred = best_model.predict([cleaned])[0]
        prediction = " Annonce Réelle" if pred == 0 else " Annonce Frauduleuse"
    return render_template_string(
        "<html><body><h2>Résultat :</h2><p>{{ prediction }}</p></body></html>",
        prediction=prediction
    )

if __name__ == "__main__":
    app.run(debug=True, port=5001)
```

8. Conclusion

Ce projet démontre l'efficacité du NLP dans la détection d'offres d'emploi frauduleuses. Le pipeline complet de traitement, classification et évaluation permet de construire un système automatique fiable. Le modèle SVM s'est révélé particulièrement efficace, et l'application Flask fournit une interface simple et intuitive pour l'utiliser dans un environnement réel.

Références

— Jurafsky, D., & Martin, J. H. (2021). *Speech and Language Processing*. Pearson.

— Kaggle Dataset : `https://www.kaggle.com/datasets/shivamb/real-or-fake-fake-jobposti`

— Documentation Scikit-learn : `https://scikit-learn.org/stable/`

— spaCy NLP Toolkit : `https://spacy.io/`