UNIVERSITÉ CÔTE D'AZUR

30/01/2025

Rapport du projet de Machine Learning

Prédiction des salaires en USD à l'aide de modèles de régression linéaire et Random Forest

Aminata SALL
MASTER 2 EADE

Introduction

L'objectif principal de ce projet est de développer un modèle de prédiction permettant d'estimer le salaire en USD (variable cible salary_in_usd) à partir de plusieurs variables explicatives. Le critère d'évaluation utilisé pour mesurer la performance des modèles est le coefficient de détermination R^2, qui indique la proportion de la variance de la variable cible expliquée par le modèle. Deux algorithmes ont été comparés : la régression linéaire (LinearRegression) et la forêt aléatoire (RandomForestRegressor). Ces modèles ont été testés avec des contraintes spécifiques concernant le nombre de variables explicatives et les hyperparamètres.

I. Données

Le jeu de données comprend plusieurs variables, dont la variable cible salary_in_usd (salaire en USD), ainsi que des variables explicatives telles que work_year, experience_level, employment_type, job_title, salary, salary_currency, employee_residence, remote_ratio, company_location et company_size. Le prétraitement des données a consisté en plusieurs étapes :

- Sélection des variables pertinentes,
- Gestion des potentielles valeurs manquantes,
- Encodage des variables catégorielles,
- Normalisation des données.

Le jeu de données a été divisé en deux ensembles : un ensemble d'entraînement (75%) et un ensemble de test (25%).

2. Méthodologie

Pour le modèle de régression linéaire, toutes les combinaisons possibles de quatre variables explicatives ont été testées, en conformité avec les contraintes du projet, afin de déterminer les meilleures variables pour expliquer le salaire en USD. Les performances du modèle ont été évaluées à l'aide de deux critères :

- Le coefficient de détermination R^2,
- L'erreur quadratique moyenne (RMSE).

Concernant Random Forest, une recherche par grille (GridSearchCV) a été effectuée pour optimiser les hyperparamètres suivants : n_estimators, max_depth, et min_samples_split. Comme pour la régression linéaire, les modèles ont été évalués en utilisant le R^2 et le RMSE et avec le nombre de variables explicatives pour une comparaison plus optimale.

Le critère privilégié reste le coefficient de détermination R^2.

3. Résultats

Les résultats obtenus pour la régression linéaire ont montré des valeurs de R^2 de la validation croisée très faibles, souvent négatives, accompagnées de RMSE élevés, ce qui indique que le modèle n'a pas réussi à saisir correctement la relation entre les variables explicatives et la

variable cible. Ce phénomène peut être dû à des relations non linéaires entre les variables ou à de la multicolinéarité.

En revanche, le modèle de Random Forest a montré des performances nettement meilleures, avec des valeurs de R^2 positives, allant jusqu'à 0,395 en moyenne pour la validation croisée, et des RMSE raisonnables. La meilleure combinaison de variables explicatives identifiée pour la forêt aléatoire était : experience_level, job_title, employee_residence, remote_ratio, où le modèle explique environ 39,5% de la variance des salaires.

Conclusion

Le modèle de régression linéaire n'a pas donné de bons résultats, probablement en raison de la nature non linéaire des données et des contraintes imposées sur le nombre de variables explicatives. En revanche, le modèle Random Forest a fourni des résultats plus satisfaisants, bien qu'il reste des opportunités d'amélioration.

Le modèle de forêt aléatoire est ainsi le modèle que nous retiendrions pour de futures prédictions. Une amélioration des performances pourrait cependant être effectuées en explorant d'autres algorithmes de régression non linéaire, l'utilisation de techniques avancées de sélection de caractéristiques, ainsi que l'enrichissement du jeu de données avec de nouvelles variables ou des transformations non linéaires.

Perspectives

Pour améliorer les performances des modèles, plusieurs pistes peuvent être explorées :

- **Ingénierie des caractéristiques** : Créer de nouvelles variables à partir des données existantes, telles que des interactions entre certaines variables.
- **Optimisation des hyperparamètres** : Utiliser des techniques plus sophistiquées comme la recherche bayésienne pour l'optimisation des hyperparamètres.
- **Validation croisée robuste** : Appliquer une validation croisée plus approfondie pour obtenir une estimation plus fiable des performances des modèles.
- **Modèles plus complexes** : Tester des modèles plus puissants, comme les réseaux de neurones ou des modèles ensemblistes, qui pourraient capturer des relations plus complexes dans les données.

Annexes

En effectuant la prédiction sans hyperpatamètres et avec une partition 80%-20%, le modèle de Random Forest a montré une performence légèrement supérieure et la meilleure combinaison de variables explicatives identifiée est : experience_level, job_title, employee_residence, et company_location, où le modèle explique environ 40% de la variance des salaires.

```python
# Validation croisée pour Random Forest
cross_val_r2s_rf = []
for train_index, val_index in kf.split(X_train_set_selected, y_train_set):
    X_train_fold, X_val_fold = X_train_set_selected.iloc[train_index], X_train_set_selected.iloc[val_index]
    y_train_fold, y_val_fold = y_train_set.iloc[train_index], y_train_set.iloc[val_index]

    #Avec hyperparamètres
    clone_rf_reg = RandomForestRegressor(n_estimators=200, max_depth=10, min_samples_split=5, random_state=42)
    #Sans hyperparamètres
    clone_rf_reg = RandomForestRegressor(random_state=42)
    clone_rf_reg.fit(X_train_fold, y_train_fold)
    y_val_fold_pred_rf = clone_rf_reg.predict(X_val_fold)
    val_r2_rf = r2_score(y_val_fold, y_val_fold_pred_rf)
    cross_val_r2s_rf.append(val_r2_rf)

cross_val_r2_rf_mean = np.mean(cross_val_r2s_rf)
```

Le code utilisé pour ce projet est disponible dans deux fichiers principaux :

- **pipeline.py** : Script de prétraitement des données.
- **main.py** : Script pour l'entraînement et l'évaluation des modèles.

Les résultats détaillés pour chaque combinaison de variables sont stockés dans le fichier Results.csv.

Les résultats du test du modèle sans hyperparamètres sont disponibles dans le fichier Annexe_Results.csv