

# Rapport Projet Machine Learning

ENCADRANT:

MR.BOUCHENTOUF TOUMI

MR.ZAKARIA HAJA

RÉALISÉ PAR:

BOUTAHAR NAOUAL

BOUFADDI IKRAM

HAJJ NADA

# Topic Prédiction de salaire

Expérience, intitulé de poste, taille de l'entreprise...etc sont tous des facteurs et variables qui affectent le salaire. Cela peut être constaté d'une simple vue superficielle sur le marché du travail. Néanmoins, il existe d'autres variables qui influencent le salaire. Pour cela, ce projet a pour but de faire une étude sur des enregistrements collectés auprès des Data Scientists de plusieurs pays pour prédire le salaire d'une personne du même domaine, ainsi de déterminer les variables qui affectent le plus celui-ci en général .

L'usage et l'importance de ce projet réside dans le fait que plusieurs employés démissionnent de leur emploi à cause du mauvais salaire, et ce projet vient pour prédire leur salaire dans une autre entreprise de n'importe quelle taille (grande, start-up...) en prenant en compte la résidence de l'employé ainsi que le type du travail(à temps plein ou partiel).

Pour développer ce système, on va utiliser des algorithmes d'apprentissage supervisé et plus précisément des **algorithmes de régression** puisqu'il s'agit d'un problème de régression. Donc, le but serait de construire un modèle efficient et efficace pour prédire le salaire.

# Dataset



On a choisi un dataset qui contient 245 samples et 11 features.

<https://www.kaggle.com/datasets/saurabhshahane/data-science-jobs-salaries>

Ces données ont été récupérées depuis le formulaire suivant :

<https://salaries.ai-jobs.net/>

## Features

- **work\_year**: l'année quand le salaire a été payé
- **experience\_level**: contient 4 valeurs possibles:
  - EN : Entry-level / Junior
  - MI : Mid-level / Intermediate
  - SE : Senior-level / Expert
  - EX : Executive-level / Director
- **employment\_type**: contient aussi 4 valeurs possibles:
  - PT : Part-time
  - FT : Full-time
  - CT : Contract
  - FL : Freelance

- **job\_title**: le rôle du salarié dans l'entreprise
- **salary**: le total du salaire.
- **salary\_currency**: devise
- **salary\_in\_usd**: salaire en Dollar Américain \$
- **employee\_residence**: la résidence de l'employé durant son travail dans l'entreprise
- **remote\_ratio**: le pourcentage du travail fait à distance; les valeurs possibles sont :
  - 0 : travail présentiel (moins de 20% à distance)
  - 50 : Partiellement à distance
  - 100 : totalement à distance (plus de 80%)
- **company\_location**
- **company\_size**:
  - S : moins de 50 employés (small)
  - M : 50 à 250 employés (medium)
  - L : plus de 250 employés (large)

# Démarche suivie:

Definir l'objectif  
prédire le salair

01



02

EDA



Preprocessing

03



04



Modeling

# 02 prétraitement des données

## 03 + EDA

Un modèle d'apprentissage réussi passe avant tout par des données de qualité : il est donc nécessaire de prétraiter les données recueillies .

Pour cela on a commencer par :

- On a vérifié s'il y a des duplcats, des valeurs null(NaN) et on les a enlevé ; dans notre cas , on a trouvé un seul duplcats et pas de valeur NaN.
- On a effectué le **Data cleaning** sur:  
job\_title en minuscules.  
company\_location : vérifier s'il ya des ',' et les enlever
- D'après la **visualisation** des features, on a trouvé que les features 'job\_title', 'company\_location'et 'employee\_residence' contiennent plus que 10 catégories , c'est pour cela on a effectué le **data trimming** ; en prenant les 9 catégories qui étaient majoritaires et remplaçant les autres par Other.

Pour voir le taux d'influence des valeurs numériques(salary, salary\_in\_usd, remote\_ratio) l'une sur l'autre, on a utilisé la **matrice de corrélation** qui nous a permit de déduire qu'il n'y a pas de corrélation entre les valeurs numériques, car on a obtenue des taux **faibles et négatifs**.

Cela nous a permis de passer à l'étape de **réduction des features** qui ne sont pas importante jusqu'à ce niveau d'analyse qui sont 'salary', 'salary currency', 'work\_year', parce que 'salary' était relative à la devise du pays de résidence et l'entreprise d'une part et d'autre part, on s'intéresse au salaire converti à une devise globale qui est 'salary\_in\_usd'.

- En ce qui concerne **l'encodage des données** catégorique en valeurs numériques, on a utilisé LabelEncoder():  
‘employee\_residence’ → ‘residence’  
‘company\_size’ → ‘size’  
‘experience\_level’ → ‘experience’  
‘employment\_type’ → ‘type’  
‘Job\_title’ → ‘job’  
‘company\_location’ → ‘location’.

## Choix des Features:

Après avoir réduit le nombre de features et séparé notre target qui est 'salary\_in\_usd', des features il nous reste 7 features qui sont : 'experience\_level', 'employment\_type', 'employment\_residence', 'remote\_ratio', 'company\_location', 'company\_size', 'Job\_title', on voulait encore réduire le nombre de features jusqu'au 4 qui seront le plus susceptible d'affecter le 'salary\_in\_usd' pour obtenir une meilleure accuracy.

Pour cela, on a utilisé le modèle **SelectKBest** de sklearn avec score\_func = mutual\_info\_regression, qui peut être utilisé sur des données de régression et de classification et qui sélectionne les features qui vont nous donner le plus grand score. Dans notre cas : 'residence', 'experience', 'job', 'location' sont les 4 best features, ce qui nous a amenés à supprimer les autres features : 'size', 'type', 'remote\_ratio'.

# 04 Modeling

## Choix du modèle:

Pour vérifier les erreurs de prédiction de chaque model, on a utilisé cross\_val\_score en lui passant les features,target et scoring='neg\_root\_mean\_squared\_error', en comparant la moyen du NMRSE par chaque modèle :

Linear Regression : -76399.324

Lasso Regression : -76399.265

KNeighbors Regressor : -69044.814

Random Forest Regressor : -59606.962

Gradient Boosting Regressor : -60102.215

Voting Regressor : -58764.503

>C/C : Voting Boosting a la plus faible valeur de RMSE (NRMSE en val absolue :Normalized Root Mean Square Error) qui représente la valeur standard de la déviation des erreurs quand une prédiction est faite. On conclut que Voting Regressor va bien fitter notre dataset

# Vérification et validation:

Pour éviter de sur évaluer notre modèle (over-fitting) on a découpé le dataset en deux (20% pour les données de test et 80% pour l'entraînement), puis on a essayé plusieurs algorithmes de régressions pour pouvoir vérifier qui est le meilleur modèle :

- Lasso Regression : -----> score :6.21%
- Linear Regression :-----> score :6.19%
- KNN Regressor :-----> score :47.44%
- Random Forest Regressor :-----> score :62.59%
- Gradient Boosting Regressor :-----> score :61.41%

Voting regressor est un modèle d'estimation qui utilise des méthodes ensemblistes de combinaison de multiples algorithmes de machine learning pour accroître les performances du modèle d'apprentissage, et parvenir à un niveau de précision supérieur à celui qui serait réalisé si on utilisait un de ces algorithmes pris séparément.

**>C/C : on va utiliser voting regressor pour prédire les salaires**