

TP N°2

Énoncé

Version 1

L'objectif de ce TP est d'appliquer :

- ✓ La régression linéaire simple
- ✓ La régression linéaire multiple

Tâche 1 : Prévoir le salaire d'un employé en utilisant la régression linéaire simple

- ✓ **Algorithme** : régression linéaire simple
- ✓ **Jeu de données (Dataset)** : salaire (*AnneesExpérience, Salaire*)
- ✓ **Le fichier associé** : TP2_1_Régression linéaire simple_Salaire
- ✓ **Appliquer les étapes suivantes** :
 - Import des bibliothèques **pandas** pour la manipulation de données, **numpy** pour le calcul numérique et **matplotlib.pyplot** pour la visualisation
 - Import des modules nécessaires de scikit-learn pour le prétraitement des données, la division des données en ensembles d'entraînement et de test, la création de modèles linéaires et l'évaluation des performances des modèles

```
from sklearn import preprocessing
```

```
from sklearn.model_selection import train_test_split
```

```
from sklearn import linear_model
```

```
from sklearn import metrics
```

- Chargement des données à partir du fichier CSV "salaire.csv" dans un DataFrame df
- Affichage des cinq premières lignes du DataFrame df pour examiner les données initiales
- Création d'un nuage de points pour visualiser la relation entre les années d'expérience et les salaires
- Extraction des caractéristiques (X) et de la variable cible (Y) à partir du DataFrame df
- Normalisation des caractéristiques (X) en utilisant RobustScaler
- Division des données en ensembles d'entraînement (X_train, Y_train) et de test (X_test, Y_test) avec une proportion de test de 20%

```
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state=42)
```

- Affichage des dimensions des ensembles d'entraînement et de test pour les données d'entrée (X) et les variables cibles (Y)
- Initialisation d'un modèle de régression linéaire

```
model=linear_model.LinearRegression()
```

- Entraînement du modèle de régression linéaire sur les données d'entraînement (X_train, Y_train)

model.fit(X_train,Y_train)

- Prédiction des valeurs cibles (Y_pred) à partir des données de test (X_test)

Y_pred=model.predict(X_test)

- Calcul et affichage des différentes métriques d'évaluation du modèle (Erreurs) qu'on a vu dans le cours
- Visualiser des données réelles (X,Y) et des prédictions (Y_pred2), telle que :

Y_pred2=model.predict(X)

- Prévoir le salaire d'un employé ayant 9.2 années d'expérience

Tâche 2 : Prévoir le profit d'une entreprise en utilisant la régression linéaire multiple

- ✓ **Algorithme** : régression linéaire multiple
 - ✓ **Jeu de données (Dataset)** : Profit (R&D, Spend, Administration, Marketing Spend, State,Profit)
 - ✓ **Le fichier associé** : TP2_2_Régression linéaire multiple_Profit
 - ✓ **Appliquer les étapes suivantes** :
 - Import des bibliothèques pandas pour la manipulation de données, numpy pour le calcul numérique et matplotlib.pyplot pour la visualisation
 - Import des modules nécessaires de scikit-learn pour la prétraitement des données, la division des données, la création de modèles linéaires et l'évaluation des performances des modèles
- ```
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn import linear_model
from sklearn import metrics
```
- Chargement des données à partir du fichier CSV "profit.csv" dans un DataFrame df
  - Afficher les premiers ligne du jeu de données
  - Création d'un nuage de points pour visualiser la relation entre les dépenses administratives et les profits
  - Extraction des données d'entrée (X) et de la variable cible (Y)
  - Normalisation des trois premières caractéristiques de X en utilisant RobustScaler
  - Encodage des données catégoriques dans X en utilisant LabelEncoder
  - Séparation des données en ensembles d'entraînement (X\_train, Y\_train) et de test (X\_test, Y\_test)
- ```
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state=42)
```
- Initialisation d'un modèle de régression linéaire
- ```
model=linear_model.LinearRegression()
```
- Entraînement du modèle sur les données d'entraînement (X\_train, Y\_train)c
- ```
model.fit(X_train,Y_train)
```
- Prédiction des valeurs cibles (Y_pred) à partir des données de test (X_test)
- ```
Y_pred=model.predict(X_test)
```
- Création d'un DataFrame pour comparer les valeurs réelles (Y\_test) et prédites (Y\_pred)
  - Calcul et affichage des différentes métriques d'évaluation du modèle (Erreurs)
  - Prévoir le profit de l'entreprise pour l'entrée E (142007, 91321, 366268, California)