

TP N°3 Énoncé (Classification)

Version 1

L'objectif de ce TP est d'appliquer :

- ✓ La régression logistique binaire
- ✓ La régression logistique multiclass
- ✓ K plus proches voisins (K-NN)
- ✓ L'arbre de décision (Decision Tree)
- ✓ La forêt aléatoire (Random Forest)

Tâche 1 : Classification de la réponse à un offre en fonction de l'âge et du genre

- ✓ Algorithme : Régression logistique binaire
- ✓ Jeu de données (Dataset) : offre (Age, Gender, Offer)
- ✓ Le fichier associé : TP3_1_Régression logistique_Binaire_Offre
- ✓ Les étapes essentielles :
 - Normaliser avec la méthode de RobustScaler
 - Réduire les variables indépendantes en 3 composantes en utilisant PCA
 - Pour créer le model de la régression logistique

```
from sklearn.linear_model import LogisticRegression
```

```
model=LogisticRegression()
```

- Calculer la matrice de confusion
- Calculer les quatre métriques d'évaluations
- ✓ Classifier la réponse à un offre d'une femme de l'âge 30 ans

Tâche 2 : La classification des plantes iris en fonction de leurs caractéristiques botaniques

- ✓ Algorithmes : Régression logistique multiclass et K plus proches voisins
- ✓ Jeu de données (Dataset) : Iris (sepal length, sepal width, petal length, petal width, espèce)

Dans le jeu de données Iris, la variable cible (target= **espèce**) correspond à l'espèce d'iris à laquelle chaque observation appartient. Les espèces d'iris sont étiquetées numériquement comme suit :

0 : setosa

1 : versicolor

2 : virginica

✓ **Les fichiers associés :**

- TP3_2_Régression logistique_Multiclass_Iris
- TP3_3_Classification_KNN_Iris

✓ **Les étapes essentielles :**

- **Charger le jeu de données *iris* à partir de `sklearn.datasets`**
- **Normaliser avec la méthode de `RobustScaler`**
- **Réduire les variables indépendantes en 3 composantes en utilisant PCA**
- **Pour créer le model de la régression logistique**

```
from sklearn.linear_model import LogisticRegression
```

```
model = LogisticRegression(multi_class='multinomial')
```

- **Pour créer le model de KNN**

```
from sklearn.neighbors import KNeighborsClassifier
```

```
model = KNeighborsClassifier(n_neighbors=3)
```

- **Calculer la matrice de confusion**
- **Calculer les quatre métriques d'évaluations**

✓ **Classifier l'espèce de plante iris de caractéristiques (4.7, 2.5, 3.2, 0.2)**

Tâche 3 (à la maison) : Même chose que la tâche 2 en utilisant l'arbre de décision (Decision tree)

Tâche 4 (à la maison) : Même chose que la tâche 2 en utilisant la forêt aléatoire (Random Forest)