

# INFO 905 - Machine Learning

## Prix des maisons

Votre rendu se fera à l'aide d'un notebook jupyter.  
L'objectif est du TP est de prédire le prix des maisons, sur un jeu de données californien.

### Données et environnement

1. Installez python, vérifiez que votre version est supérieur à la version 3.

```
pip3 --version
```

2. Installez les modules nécessaires au différents TPs :

```
pip3 install --upgrade jupyter matplotlib numpy pandas scipy scikit-learn six
```

3. Récupérez et décompressez l'archive *housing.csv*.
4. Lancer un notebook jupyter :

```
jupyter notebook monfichier.ipynb
```

### Exploration des données

Via la librairie *Pandas*, utilisez les fonctions *read\_csv*, *head*, *info*, *value\_counts*, *describe*, *count*, *mean*, *min*, *max*,... pour regarder les données.

Combien y a-t-il de variables ? Quel est leur type ? Y a-t-il des données manquantes ?

### Jeu de test/validation

Séparez vos données en 2 jeux de données : l'un servira de jeu de test, l'autre de jeu de validation, en utilisant *train\_test\_split* de *sklearn*.

### Visualisation des données et recherche de corrélations

1. Proposez une visualisation de vos données, en explorant la fonction *plot* de la librairie *pandas*.
2. Quelle semble être la variable la plus corrélée au prix des maisons (via la fonction *corr*) ?

### Préparation des données

1. Séparez les variables explicatives de la variable à expliquer (*drop* et *copy*).
2. Remédiez au manque de données, au choix : en supprimant les lignes correspondants (*dropna*), en supprimant la variable (*drop*), ou en mettant une valeur par défaut (*fillna*).
3. Gérez la variable qualitative : en regardant les classes *LabelEncoder* et *OneHotEncoder*, et la méthode *fit\_transform*.
4. Pour que les algorithmes de Machine Learning fonctionnent, il faut en général recalibrer les variables pour qu'elles soient dans les mêmes échelles. Recalibrez-les via le *min-max scaling* ou la *normalization*. expliquez ces deux transformations.

## Selection et entraînement de modèle

### Régression linéaire

Grâce à la classe *LinearRegression* de *sklearn*, faites une prédiction du prix des maisons. Vous aurez besoin des fonctions *fit*, *predict* et *mean\_squared\_error*.

Que pensez-vous de cette prédiction ?

### Arbre de décision

Grâce à la classe *DecisionTreeRegressor* de *sklearn*, faites une prédiction du prix des maisons. Vous aurez besoin des fonctions *fit*, *predict* et *mean\_squared\_error*.

Que pensez-vous de cette prédiction ?

### Forêts Aléatoires

Grâce à la classe *RandomForestRegressor* de *sklearn*, faites une prédiction du prix des maisons. Vous aurez besoin des fonctions *fit*, *predict* et *mean\_squared\_error*.

Que pensez-vous de cette prédiction ?

### Recherche d'un modèle

Utilisez les classes *GridSearchCV* et *RandomizedSearchCV* pour sélectionner le meilleur modèle.