

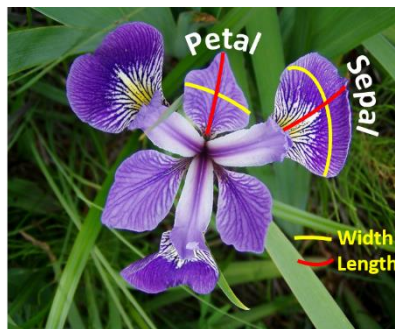
TP 3

Accédez au dossier TP3_IA :

- Ouvrez le 1^{er} Lab (**Lab1_TrainTest**) pour apprendre à éviter le sur-apprentissage :
 - Exécutez le 1^{er} exemple pour générer des données.
 - Pour éviter le sur-apprentissage des données, exécutez le 2^{ème} exemple pour diviser les données en une partie pour l'entraînement du modèle soit 80 % et 20% pour tester sa performance.
 - Exécutez le 3^{ème} exemple pour afficher les données d'entraînement.
 - Exécutez le 4^{ème} exemple pour afficher les données test.
 - Dans le 5^{ème} exemple définissez un polynôme d'ordre 8 comme modèle d'apprentissage. On apprend les paramètres du modèle en utilisant les données d'entraînement.
 - Dans le 6^{ème} exemple affichez le modèle appris ainsi que les données d'entraînement. Décrire les résultats affichés.
 - Dans le 7^{ème} exemple affichez le modèle appris ainsi que les données du test. Décrire les résultats affichés.
 - Exécutez le 8^{ème} exemple pour comparer les résultats prédits par le modèle `p4(testx)` pour les données test `testx` et les résultats réels `testy` en utilisant la métrique `r2_score()`.
 - Exécutez le 9^{ème} exemple pour comparer les résultats prédits par le modèle `p4(trainX)` pour les données d'entraînement `trainX` et les résultats réels `trainY` en utilisant la métrique `r2_score()`.
 - Faire l'activité.



Iris Virginica



Iris Versicolor



Iris Setosa

- Ouvrez le 2^{ème} Lab (**Lab2_KfoldCrossValidation**) qui explique l'utilisation de la validation croisée (Cross-Validation)
 - Exécutez le 1^{er} exemple pour importer la dataset `Iris`. Iris comprend 50 échantillons de trois espèces d'iris (`Iris setosa`, `Iris virginica` et `Iris versicolor`). Quatre caractéristiques ont été mesurées pour chaque échantillon : la longueur et la largeur des sépales et des pétales, en centimètres.

- Dans le 2^{ème} exemple utilisez la fonction `train_test_split()` pour séparer les données en une partie `train` que vous allez utiliser pour apprendre les paramètres du classificateur SVC et une partie `test` pour tester sa performance. Interprétez le résultat.
 - Dans le 3^{ème} exemple appliquez une validation croisée sur la dataset Iris en utilisant la fonction prédéfini `cross_val_score()` avec un nombre de folds égale à 5. Interprétez les résultats.
 - Dans le 4^{ème} exemple utilisez un noyau (Kernel) plus complexe, par exemple utilisez un noyau polynômial au lieu d'un simple noyau linéaire et testez le modèle en utilisant une validation croisée avec 5 folds.
 - Exécutez le 5^{ème} exemple pour tester la performance du classificateur utilisant un noyau polynômial. Comparez les résultats donnés par le classificateur linéaire et polynômial, puis essayez de trouver une explication à ces résultats.
 - Faire l'activité
- Ouvrez le 3^{ème} Lab (**Lab3_Metrics**) où vous allez apprendre comment sélectionner et utiliser différentes métriques d'évaluation de performances des modèles d'apprentissage automatique (Machine Learning)

🚦 Métriques d'évaluation de performances des modèles de classification (pour tous les exemples qui suivent, on va utiliser *Pima Indians dataset*) :

- Comment marche la régression logistique comme algorithme d'apprentissage ?
- Dans le 1^{er} exemple utilisez le modèle régression logistique comme modèle d'apprentissage et appliquez la métrique `accuracy` pour évaluer sa performance. Comment jugez-vous la performance du modèle.
- Dans le 2^{ème} exemple utilisez le modèle régression logistique comme modèle d'apprentissage et appliquez la métrique `log_loss` (Erreur Logistique) pour évaluer sa performance. Comment jugez-vous la performance du modèle.
- Dans le 3^{ème} exemple utilisez le modèle régression logistique comme modèle d'apprentissage et dessinez la **courbe ROC** pour évaluer sa performance. Comment jugez-vous la performance du modèle.
- Dans le 4^{ème} exemple utilisez le modèle régression logistique comme modèle d'apprentissage et appliquez la métrique `matrice de confusion` pour évaluer sa performance. Comment jugez-vous la performance du modèle.
- Dans le 5^{ème} exemple utilisez le modèle régression logistique comme modèle d'apprentissage et appliquez la fonction `classification_report()` pour calculer la **precision, recall, f1-score** et le **support** pour chaque classe. Comment jugez-vous la performance du modèle.

🚦 Métriques d'évaluation de performances des modèles de régression (Pour tous les exemples qui suivent on va utiliser *Boston House Price dataset*) :

- Dans le 1^{er} exemple utilisez le modèle régression linéaire comme modèle d'apprentissage et mesurez l'erreur moyenne absolue (MAE) entre les prédictions du modèle et les données de la dataset. Comment jugez-vous la performance du modèle.
- Dans le 2^{eme} exemple utilisez le modèle régression linéaire comme modèle d'apprentissage et mesurez l'erreur quadratique moyenne (MSE) entre les prédictions du modèle et les données de la dataset. Comment jugez-vous la performance du modèle.
- Dans le 2^{eme} exemple utilisez le modèle régression linéaire comme modèle d'apprentissage et appliquez la métrique R-squared pour évaluer sa performance. Comment jugez-vous la performance du modèle.