Conseils pour le TP 2

A n’utiliser que si vous êtes bloqué.

Vous apprendrez mieux si vous n’utilisez pas ces conseils, mais cherchez par vous même.

1.1

Utiliser la méthode pandas read_csv.
Utiliser la méthode pandas head pour afficher les premiers exemples.

1.2

Utiliser la méthode pandas read_csv.
Utiliser la méthode pandas head pour afficher les premiers exemples.

1.3

Utiliser la propriété values pour passer du DataFrame au tableau numpy
Utiliser ensuite les indices pour sélectionner les bonnes colonnes

1.4

Utiliser la propriété values pour passer du DataFrame au tableau numpy
Utiliser ensuite les indices pour sélectionner la colonne indiquant la survie

1.5

Utiliser la fonction sklearn train_test_split.
La propriété shape permet de connaître la forme des arrasy numpy

1.6

Appliquer la méthode tolist() à la propriété columns du DataFrame Panda,

2.1

Utiliser un DecisionTreeClassifier de sklearn et la fonction fit()

2.2

Utiliser les fonctions sklearn predict et accuracy_score

2.3

Modifier le paramètre max_depth dans la question 2.1, et d’autres paramètre que vous pouvez trouver sur le site de scikit learn.

2.4

Voir l’exemple Cours1_ex2_Visualisation.ipynb utilisant graphwiz

2.5

La fonction generer_resultats, vous est donnée.
Il faut utiliser le lien kaggle sur la Learning Box, et se connecter avec son compte Google (ou autre si vous préférez).

3.1

Utiliser un RandomForestClassifier de sklearn et la fonction fit()

3.2

propriété feature_importances_ du modèle entraîné

3.3

Boucler sur différentes valeurs de n_estimator, max_features et max_depth de RandomForestClassifier. Dans la boucle : Construire le Classifier Entraîner le modèle et calculer la précision sur les données de test.
La fonction python range peut être utilisée.

3.4

La fonction generer_resultats, vous est donnée.
Il faut utiliser le lien kaggle sur la Learning Box, et se connecter avec son compte Google (ou autre si vous préférez).

3.5

Pour être affichable en 2D, le modèle ne doit contenir que 2 features.
Ensuite, la fonction plot_decision_boundary vous est donnée.

4.1

Utiliser un AdaBoostClassifier de sklearn (avec des DecisionTree comme base_estimator) et la fonction fit()

4.2

Fonctions predict et accuracy_score

4.3

Consulter la doc de GradientBoostingClassifier, notamment les paramètres n_iter_no_change et tol.
L'exemple Cours1_ex6_Boosting.ipynb est une bonne source.

4.4

Fonction generer_resultats

4.5

Repartir du DataFrame train. Refaire une décomposition avec train_test_split
Ensuite s’inspirer de Cours1_ex6_Boosting.ipynb.
Réaliser l’apprentissage avec les données de train issues de train_test_split

4.6

Réaliser l’affichage des données dev issues de train_test_split