

Plan de cours N° : 1043**Durée :** 3 jours (21h)**Participants**

Développeur, chef de projets proche du développement, ingénieur scientifique sachant coder.

Pré-Requis

Maîtriser l'algorithmique, avoir une appétence pour les mathématiques. La connaissance de Python et des statistiques est un plus.

Objectifs

Savoir mettre en place une stratégie de Machine Learning en Python afin de créer le modèle le plus satisfaisant possible en le mesurant et en affichant les résultats, le tout en utilisant des algorithmes performants.

Méthode pédagogique

Alternance entre apports théoriques et exercices pratiques

Support de cours fourni lors de la formation

Moyens d'encadrement mis en oeuvre

1 à 8 personnes maximum par session
1 poste informatique par personne
Une assistance post-formation, d'une durée d'un an, sur le contenu de la formation

Moyens permettant de suivre son exécution et d'en apprécier les résultats

Emargement par demi-journée
Evaluation des acquis par mise en situation de travail
Evaluation qualitative de fin de stage
Remise d'une attestation individuelle de formation en fin de stage

Assistance

formateurs@atp-formation.com

Introduction aux Data Sciences

- Qu'est que la data science ?
- Qu'est-ce que Python ?
- Qu'est que le Machine Learning ?
- Apprentissage supervisé vs non supervisé
- Les statistiques
- La randomisation
- La loi normale

Introduction à Python pour les Data Sciences

- Les bases de Python
- Les listes
- Les tuples
- Les dictionnaires
- Les modules et packages
- L'orienté objet
- Le module math
- Les expressions lambda
- Map, reduce et filter
- Les générateurs
- Le module CSV
- Anaconda

Introduction aux DataLake, DataMart et DataWarehouse

- Qu'est-ce qu'un DataLake ?
- Les différents types de DataLake
- Le Big Data
- Qu'est-ce qu'un DataWarehouse ?
- Qu'est qu'un DataMart ?
- Mise en place d'un DataMart
- Les fichiers
- Les bases de données SQL
- Les bases de données No-SQL
- Qu'est que MongoDB, Cassandra, Redis, CouchDb

Python Package Installer

- Utilisation de PIP
- Installation de package PIP
- PyPi

MathPlotLib

- Utilisation de la bibliothèque scientifique de graphes MathPlotLib
- Affichage de données dans un graphique 2D
- Affichages de sous-graphes
- Affichage de polynômes et de sinusoidales

Machine Learning

- Filtrage
- Qu'est qu'un modèle et un dataset
- Qu'est qu'une régression
- Les différents types de régression
- La régression linéaire
- Gestion du risque et des erreurs
- Quarter d'Ascombe
- Trouver le bon modèle
- La classification
- Loi normale, variance et écart type
- Apprentissage
- Mesure de la performance
- No Fee Lunch

La régression linéaire en Python

- Programmer une régression linéaire en Python
- Utilisation des expressions lambda et des listes en intention
- Afficher la régression avec MathPlotLib
- L'erreur quadratique
- La variance
- Le risque

Numpy et SciPy

- Les tableaux et les matrices
- L'algèbre linéaire avec Numpy
- Numpy et MathPlotLib

Plan de cours N° : 1043

ScikitLearn

- Régressions polynomiales
- La régression linéaire
- La création du modèle
- L'échantillonnage
- La randomisation
- L'apprentissage avec fit
- La prédiction du modèle
- Les metrics
- Choix du modèle
- PreProcessing et Pipeline
- Régressions non polynomiales

Nearest Neighbors

- Algorithme des k plus proches voisins (k-NN)
- Modèle de classification
- K-NN avec SciKitLearn
- Choix du meilleur k
- Sérialisation du modèle
- Variance vs Erreurs
- Autres modèles : SVN, Random Forest
- La clusterisation avec k-Means

Pandas

- L'analyse des données avec Pandas
- Les DataFrames
- La théorie ensembliste avec Pandas
- L'importation de données CSV
- L'importation de données SQL
- Pandas et SKLearn

Jupyter

- Présentation de Jupyter et Ipython
- Installation
- Utilisation de Jupyter avec MathPlotLib et SKLearn

Les réseaux neuronaux

- Le perceptron
- Les réseaux neuronaux
- Les heuristiques
- Le deep learning