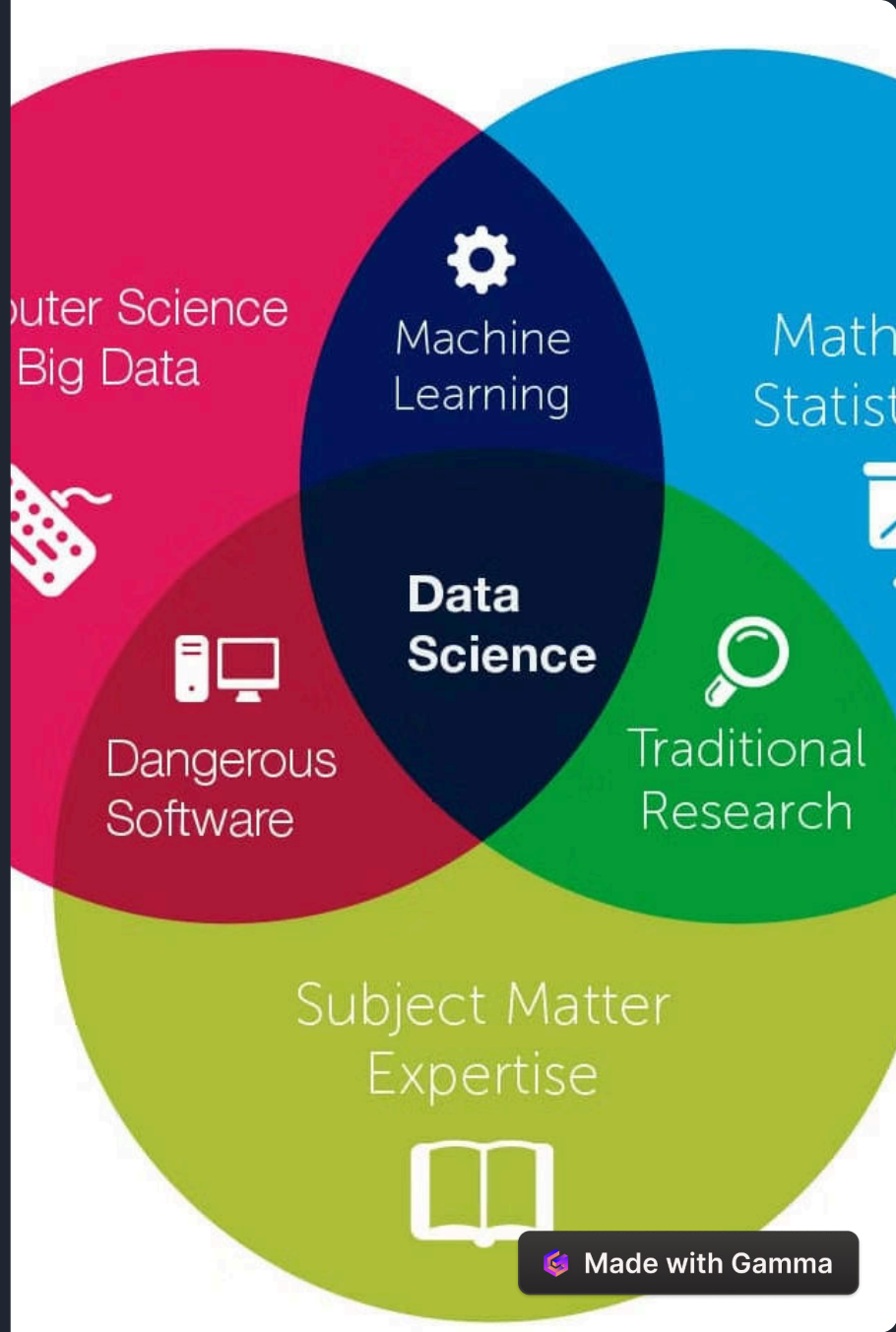
Maîtrisez l'Apprentissage Automatique avec Python

Découvrez comment devenir un expert en apprentissage automatique en utilisant Python. Grâce à cette série de formations, vous apprendrez les compétences essentielles pour décrocher un poste de scientifique en apprentissage automatique. Explorez les outils et techniques indispensables pour traiter des données, entraîner des modèles performants et résoudre des problèmes concrets du monde réel.

by etixi hacking



Made with Gamma

Python pour l'Apprentissage Automatique Supervisé

Données de Qualité

Apprenez à préparer et nettoyer vos données pour en extraire les meilleurs paramètres à utiliser dans vos modèles d'apprentissage supervisé. Une bonne ingénierie des fonctionnalités fait toute la différence.

Modèles Performants

Explorez une variété de techniques d'apprentissage supervisé comme la régression linéaire, la classification, les arbres de décision et bien d'autres. Apprenez à les mettre en œuvre avec des bibliothèques comme scikit-learn.

Évaluation et Optimisation

Mesurez les performances de vos modèles à l'aide de métriques pertinentes. Ajustez les hyperparamètres pour améliorer la prédiction et obtenir des résultats précis.

Made with Gamma

Apprentissage Non Supervisé avec Python

1 Clustering

Explorez des algorithmes de clustering comme K-Means pour découvrir des groupes naturels au sein de vos données sans étiquettes.

2 Réduction de Dimensionnalité

Utilisez des techniques comme l'Analyse en Composantes Principales (ACP) pour simplifier vos données tout en préservant l'information essentielle.

3 Apprentissage Probabiliste

Apprenez à modéliser la distribution de vos données avec des méthodes comme les Machines à Vecteurs de Support (SVM) et les Modèles de Mélange Gaussien.

Made with Gamma

CLASSIFICATEURS LINÉAIRES EN PYTHON

Connaissance présumée

- Dans ce cours, nous supposerons que vous avez déjà été exposé à :
- Python, au niveau Python Intermédiaire scikit-learn,
- Au niveau de l'Apprentissage Supervisé avec scikit-learn
- Apprentissage supervisé, au niveau de l'apprentissage supervisé avec scikit-learn

Ajustement et prévision

```
import sklearn.datasets

newsgroups = sklearn.datasets.fetch_20newsgroups_vectorized()

X, y = newsgroups.data, newsgroups.target
```

```
X.shape
```

```
(11314, 130107)
```

```
y.shape
```

```
(11314,)
```

Ajustement et prédiction (suite)

```
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors=1)
```

```
knn.fit(X,y)
```

```
y_pred = knn.predict(X)
```

Évaluation du modèle

```
knn.score(X,y)
```

```
0.99991
```

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y)
```

```
knn.fit(X_train, y_train)

knn.score(X_test, y_test)
```

```
0.66242
```

Application de la régression logistique et du SVM

Utiliser la régression logistique

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

```
lr = LogisticRegression()
lr.fit(X_train, y_train)
lr.predict(X_test)
lr.score(X_test, y_test)
```

Exemple de régression logistique

```
import sklearn.datasets
wine = sklearn.datasets.load_wine()
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(wine.data, wine.target)
lr.score(wine.data, wine.target)
```

```
0.966
```

```
lr.predict_proba(wine.data[:1])
```

```
array([[9.966e-01, 2.740e-03, 6.787e-04]])
```

Utilisation de LinearSVC

- **LinearSVC** fonctionne de la même manière

```
import sklearn.datasets

wine = sklearn.datasets.load_wine()
from sklearn.svm import LinearSVC

svm = LinearSVC()

svm.fit(wine.data, wine.target)
svm.score(wine.data, wine.target)
```

```
0.955
```

Utiliser SVC

```
import sklearn.datasets
wine = sklearn.datasets.load_wine()
from sklearn.svm import SVC
svm = SVC()
svm.fit(wine.data, wine.target);
svm.score(wine.data, wine.target)
```

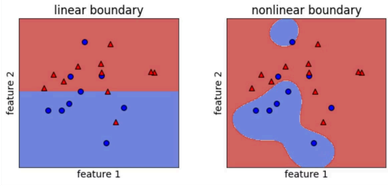
```
0.708
```

Examen de la complexité du modèle :

- **Underfitting** : le modèle est trop simple, faible précision de l'entraînement
- **Overfitting** : le modèle est trop complexe, la précision des tests est faible

Limites de décision linéaires

Limites de décision linéaires

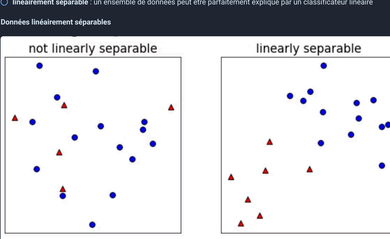


Définitions

Vocabulaire:

- **classication** : apprendre à prédire les catégories
- **limite de décision** : la surface séparant les différentes classes prédites
- **classeur linéaire** : un classeur qui apprend les limites de décision linéaires
- par exemple, régression logistique, SVM linéaire
- **linéairement séparable** : un ensemble de données peut être parfaitement expliqué par un classificateur linéaire

Données linéairement séparables



Made with Gamma

Apprentissage Profond avec Python

Réseaux de Neurones

Construisez vos premiers réseaux de neurones avec PyTorch et Keras. Apprenez les concepts fondamentaux comme les couches, les fonctions d'activation et la rétropropagation.

Applications Pratiques

Appliquez l'apprentissage profond à des cas d'usage concrets tels que la classification d'images, la reconnaissance vocale et le traitement du langage naturel.

Apprentissage Avancé

Explorez des architectures profondes comme les Réseaux de Neurones Convolutifs (CNN) et les Réseaux de Neurones Récurrents (RNN) pour relever des défis complexes en traitement d'images et de séquences.

Made with Gamma

Traitement du Langage Naturel avec Python

Analyse de Texte

Explorez des techniques de traitement du langage naturel comme l'extraction d'entités, la classification de sentiments et la génération de texte.

Prétraitement de Texte

Apprenez à nettoyer, tokeniser et lemmatiser vos données textuelles pour les préparer à l'entraînement de vos modèles.

Représentations de Mots

Plongez-vous dans les techniques de plongement de mots (word embeddings) pour capturer la sémantique et les relations entre les termes.

Modèles Neuronaux

Utilisez des architectures de réseaux de neurones comme les LSTM et les Transformers pour résoudre des tâches avancées de NLP.

Made with Gamma

Traitement d'Images avec Python

Chargement et Affichage

Apprenez à charger et à afficher des images en utilisant des bibliothèques comme Matplotlib et Pillow.

Prétraitement d'Images

Explorez des techniques de prétraitement comme la normalisation, la réduction de bruit et l'augmentation de données pour améliorer la qualité de vos images.

Extraction de Caractéristiques

Utilisez des méthodes comme la détection de contours et la segmentation pour extraire des informations pertinentes de vos images.

Apprentissage Profond

Appliquez les réseaux de neurones convolutifs (CNN) pour réaliser des tâches avancées de classification et de détection d'objets.

Made with Gamma

Apprentissage Automatique Distribué avec PySpark

1

Spark et PySpark

Découvrez les fondamentaux de Spark, le moteur de traitement de données distribué, et son API Python PySpark.

2

Préparation des Données

Apprenez à manipuler et à transformer vos données à l'échelle avec les DataFrames Spark.

3

Modèles Distribués

Explorez les algorithmes d'apprentissage automatique de PySpark pour entraîner des modèles sur de grands volumes de données.

4

Déploiement en Production

Déployez vos modèles en production et automatisez les tâches d'apprentissage automatique à l'échelle.

Made with Gamma

Projets Pratiques d'Apprentissage Automatique

Projet 1	Modélisation Prédictive pour l'Agriculture	Utilisez l'apprentissage supervisé et la sélection de fonctionnalités pour résoudre des problèmes concrets du domaine agricole.
Projet 2	Regroupement des Espèces de Manchots	Explorez des techniques de clustering non supervisé pour analyser les données sur les populations de pingouins antarctiques.
Projet 3	Prédiction des Durées de Location de Films	Créez un modèle de régression pour prédire la durée de location de films pour une entreprise.

Ces projets pratiques vous permettront d'appliquer les compétences que vous avez acquises et de résoudre des problèmes réels du monde de l'entreprise. Ils vous donneront l'expérience nécessaire pour décrocher un poste de scientifique en apprentissage automatique.

Made with Gamma