# Roadmap complète Maths pour IA – Méthode Concepts → Visualisation → Code

## Module 1 : Algèbre Linéaire (Vecteurs et Matrices)

**Objectifs :** Manipuler vecteurs, matrices, et comprendre leurs rôles dans les datasets.

**Cours :**

1. Vecteurs : addition, multiplication par scalaire, norme, distance euclidienne.
2. Matrices : addition, multiplication, transposition, inverse.
3. Produit matriciel et application sur datasets (features × exemples).
4. Valeurs propres et vecteurs propres : introduction à PCA.

**Exercices pratiques :**

- Visualiser vecteurs dans 2D ou 3D (matplotlib).
- Calculer produit matriciel et distances avec **Numpy**.
- Réduire la dimension d'un dataset avec PCA (**Scikit-learn**).

## Module 2 : Calcul différentiel et optimisation

**Objectifs :** Comprendre comment optimiser les modèles ML via gradients.

**Cours :**

1. Dérivées simples et règles de dérivation.
2. Dérivées partielles pour fonctions multivariables.
3. Gradients et optimisation.
4. Application à fonctions de coût simples (ex : MSE).

**Exercices pratiques :**

- Calculer dérivées de fonctions simples avec Python.

- Implémenter **descente de gradient** pour une fonction simple.
- Observer l'effet du learning rate sur la convergence.

## Module 3 : Probabilités et Statistiques

**Objectifs :** Comprendre l'incertitude et les modèles probabilistes.

**Cours :**

1. Probabilité de base : événements, indépendance.
2. Variables aléatoires et distributions : normale, binomiale, uniforme.
3. Espérance, variance, covariance.
4. Probabilité conditionnelle et théorème de Bayes.

**Exercices pratiques :**

- Tirages aléatoires et simulation avec **Numpy**.
- Calcul de moyenne, variance, covariance sur un dataset réel.
- Application du **Naive Bayes** pour classification simple.

## Module 4 : Statistiques avancées et analyse des données

**Objectifs :** Préparer les données pour ML.

**Cours :**

1. Statistiques descriptives : min, max, quartiles, histogrammes.
2. Normalisation et standardisation des données.
3. Corrélation et covariance.
4. Analyse en composantes principales (PCA).

**Exercices pratiques :**

- Normaliser un dataset avec **Scikit-learn**.
- Visualiser la corrélation entre features (matplotlib / seaborn).
- PCA sur dataset Iris ou MNIST.

## Module 5 : Optimisation et introduction au Machine Learning

**Objectifs :** Relier mathématiques aux modèles ML.

**Cours :**

1. Fonction de coût : MSE, Cross-Entropy.
2. Gradient descent et variantes (SGD, Adam).
3. Sur-apprentissage et sous-apprentissage (overfitting/underfitting).
4. Introduction aux réseaux neuronaux simples.

**Exercices pratiques :**

- Régression linéaire et classification simple avec **Scikit-learn**.
- Visualiser la fonction de coût et descente de gradient.
- Construire un réseau neuronal simple avec **PyTorch** ou **TensorFlow**.

## Module 6 : Mathématiques avancées pour Deep Learning (optionnel mais conseillé)

**Objectifs :** Comprendre les modèles complexes.

**Cours :**

1. Fonctions non linéaires : sigmoïde, ReLU, softmax.
2. Matrices pour réseaux neuronaux : poids, biais, propagation avant et arrière.
3. Gradients vectoriels et rétropropagation.
4. Optimisation avancée : momentum, Adam, RMSprop.

**Exercices pratiques :**

- Implémenter propagation avant et rétropropagation pour un petit réseau.
- Comparer différents optimizers sur un dataset simple.
- Visualiser les effets de différentes fonctions d'activation.

## Méthode recommandée pour chaque module

1. **Concepts :** Lire la théorie et comprendre les définitions.
2. **Visualisation :** Tracer graphiques, vecteurs, matrices, distributions avec matplotlib / seaborn.
3. **Code :** Implémenter en Python avec **Numpy, Pandas, Scikit-learn**, et pour Deep Learning, **PyTorch / TensorFlow**.