Découverte de ML.NET

Afterwork ApolloNET – Guillaume Perouffe

11 juillet 2023

APOLLO
CODING LIFE, LIVING CODE

Petites informations préliminaires

- Cette présentation est une découverte
- Interrompez-moi à tout moment !
- Vous pouvez cloner les sources à cette adresse :
  - https://github.com/gperouffe/Apollo.ML
- Ces slides vous seront communiquées suite à l'afterwork

Sommaire de cet afterwork

1. Introduction : L'apprentissage automatique (AA)
    1. Pourquoi et depuis quand?
    2. Principes généraux
    3. Quels sont les outils utilisés aujourd'hui?
    4. Et ML.Net dans tout ça?
2. Première prise en main
    1. ModelBuilder : Création d'un modèle d'analyse de sentiments
    2. Revue du code produit

 PAUSE 

Sommaire de cet afterwork

3. ML.NET : Concepts et méthodes
    1. Chargement des jeux de données
    2. Préparation et apprentissage
    3. Évaluation du modèle
    4. Utilisation du modèle
4. Deux exemples
    1. Prédiction des prix d'une course de taxi
    2. Chargement d'un modèle pré-entrainé
5. À vous de jouer !

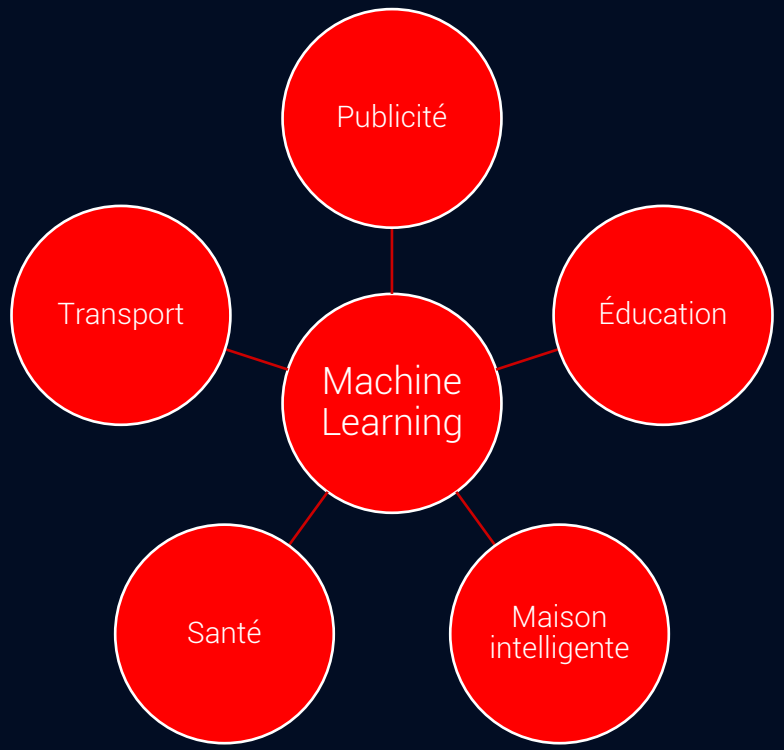
01

Introduction

Les sources de l'apprentissage automatique (AA)

- 1943 – Recherche sur les neurones de Warren McCulloch et Walter Pitts
  - Premier réseau neuronal électrique
- 1950 – C*omputing Machinery and Intelligence,* Alan Turing
  - Test de Turing – Le jeu de l'imitation
  - Introduit les machines apprenantes (« Learning Machines »)
- 1956 – Première utilisation du terme « Machine Learning », Arthur Samuel
  - Chercheur pour IBM, développe un programme jouant au Jeu de dames
  - 1961 : son programme bat le 4ème joueur américain

L'AA est en train de changer le monde

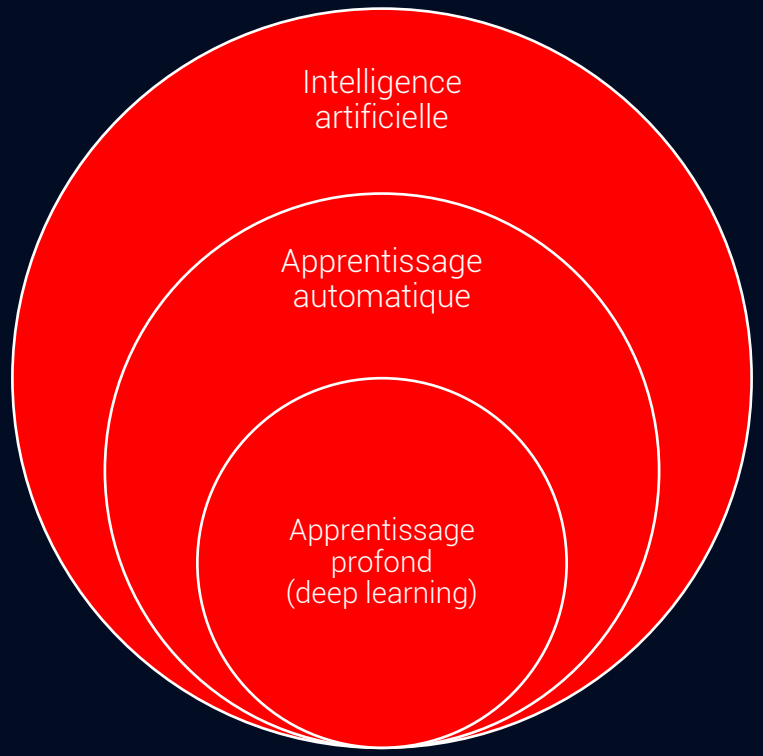


... merci le big data!

“Ten years ago, we struggled to find 10 machine learning-based business applications.
Now we struggle to find 10 that don’t use it“

Alexander Linden, research vice president at Gartner, 2016

Lien de l'AA avec d'autres domaines



- Créer une intelligence artificielle ne nécessite pas forcément d'AA
- L'AA n'est pas forcément profond

L'AA est un domaine de recherche à part entière, en lien avec ceux de l'IA, des Statistiques, ou de l'Exploration de données

Principe général

L'apprentissage d'un modèle a pour but principal :

Généraliser à partir de l'expérience acquise

Modes d'apprentissage

Il existe plusieurs modes d'apprentissage pour arriver à ce but :

- Apprentissage supervisé
- Apprentissage non-supervisé
- Apprentissage par renforcement
- ...

➔ À choisir en fonction du problème

Les problèmes pouvant être résolus par l'AA

Classification de données

Prédictions

Systèmes de recommandation

Clustering

Association

… et beaucoup, beaucoup d'autres

Types de modèles

Chaque apprentissage implique la création d'un modèle à entraîner.

Ce modèle accepte en entrée un point de donnée et peut avoir une ou plusieurs sorties.

Il en existe de nombreux types :

- Réseaux de neurones
- Arbres décisionnels
- Régressions
- SVMs (séparateurs vaste marge)
- Algorithmes génétiques

Alimentation en données

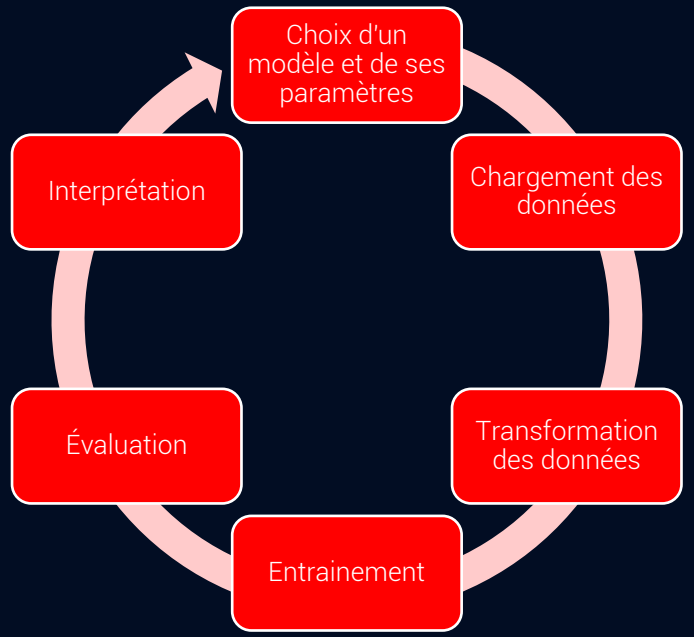
En temps normal, les données brutes :

- Ne sont pas structurées
- Sont lacunaires
- Doivent être retravaillées pour être passées en entrée du modèle

➔ On doit systématiquement passer par une étape de pre-processing

L'AA demande un processus itératif

Entrainement



Consommation

Limites usuelles

L'apprentissage automatique n'est pas une solution universelle, et peut être mis en défaut par :

- Des biais liés aux données
- Un manque d'explicabilité
- Le surapprentissage

Outils utilisés

Et ML.NET?

Tensorflow

Star 176k

Star 55k

Scikit-learn

Star 68.6k

Pytorch

Star 8.5k

Image by Freepik

Pourquoi l'utiliser alors?

1. Pour intégrer du ML à vos applications .NET
2. Pour bénéficier de meilleures performances
3. Pour son côté multiplateforme

Exemple de comparaison

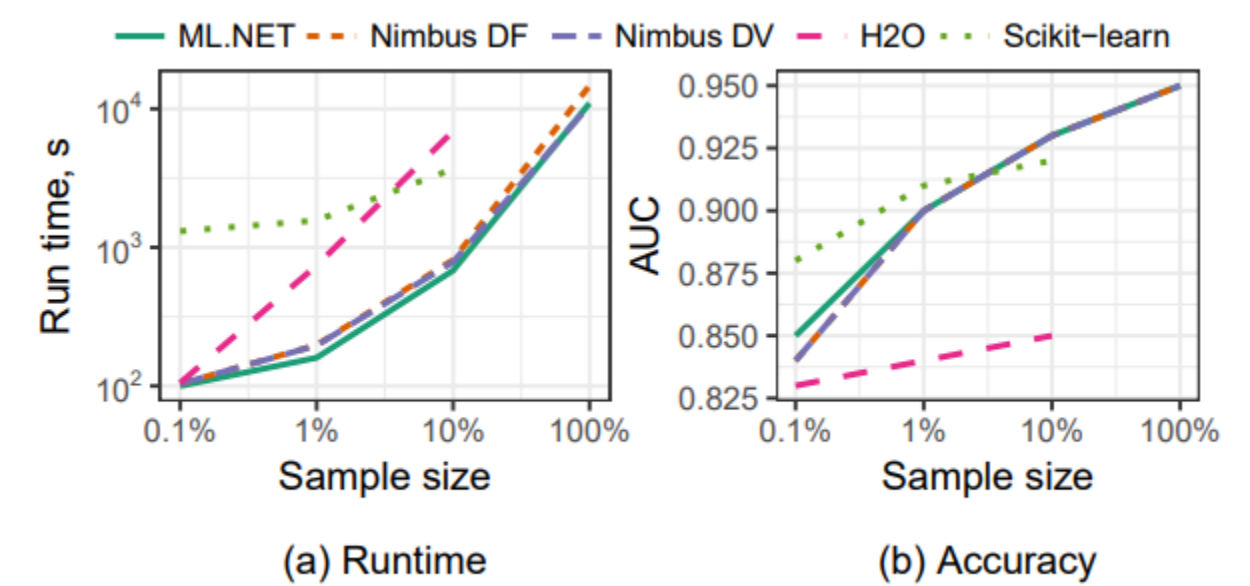


Figure 7: Experimental results for the Amazon dataset.

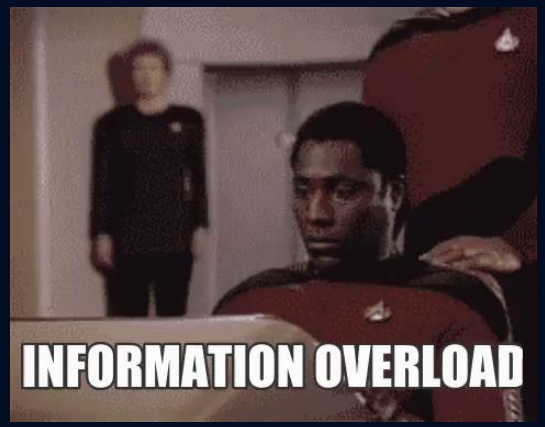
02

Première prise en main

Si on résume

Pour résoudre un problème grâce au machine learning, on doit avoir :

- Des données propres
- Un type de modèle
- Un mode d'apprentissage



➔ Si on ne sait pas quoi faire, il existe le ModelBuilder avec AutoML!

Les problèmes adaptés à AutoML

Sur ordinateur local :

- Classification
- Régression
- Prévision de série chronologique

Via Azure ML :

- Vision : classification, détection, segmentation
- NLP (traitement du langage naturel)

03

ML.NET : Concepts et méthodes

Les motivations derrière ML.NET

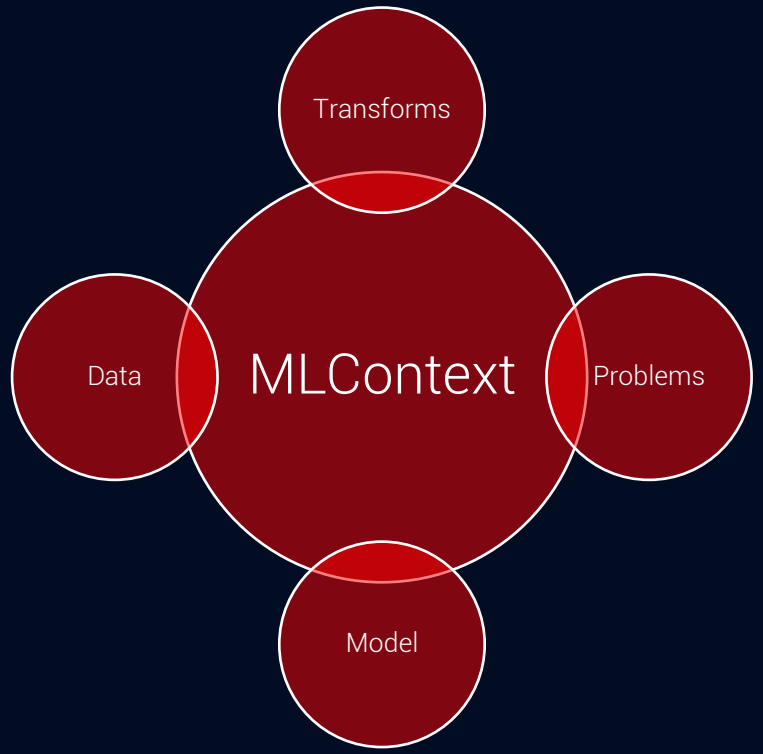
Partant de 3 constats :

- L'exploration interactive n'est pas la plus utilisée en vie réelle
- Pour utiliser et essayer plusieurs algos d'AA, il fallait souvent recréer des pipelines personnalisés
- Au moment de l'intégration du modèle entrainé, une réécriture était souvent nécessaire

Les solutions apportées par ML.NET :

- Unification : permettre l'utilisation de modèles variés, et le déploiement sans surcoût
- Extensibilité : permettre d'expérimenter facilement de nouveaux algos
- Performance : permettre une mise à l'échelle facile, et utiliser le matériel à son maximum

Vue d'ensemble



L'abstraction DataView

Les données sont toujours chargées dans une DataView au préalable.

Les DataView sont :

- Composables
- Virtuelles
- Immutables et déterministes

L'abstraction DataView

Les colonnes sont identifiées uniquement par leur position.

Elles peuvent être cachées, en ajoutant une colonne portant le même nom.

Elles peuvent aussi être multidimensionnelles : VectorType

Charger une DataView

Plusieurs sources possibles :

- Directement depuis un Enumerable en mémoire
- Depuis un fichier texte (csv, tsv...)
- Depuis un fichier binaire
- Depuis une base de données

Les méthodes correspondantes sont dans MLContext.Data

Les pipelines

Les opérations sur les DataView ont lieu dans un pipeline, ou chaîne d'estimateurs.
Ces estimateurs peuvent être de trois types :

- Transform : DataView ➔ DataView
  Transformation simple (ajout d'une ou plusieurs colonnes)
- Trainable Transform : DataView ➔ DataView
  Transformation qui nécessite de voir tout le dataset
- Trainer : DataView ➔ Modèle
  Similaire, mais produit un modèle qui peut générer des prédictions

Pour créer un pipeline, il suffit de chaîner des estimateurs avec leur méthode Append

Quelques estimateurs courants

Les estimateurs de type Transform sont situés sous MLContext.Transforms.

On a notamment :

- Des opérations de conversion (ex: MapValue)
- Des méthodes de normalisation (ex: NormalizeMinMax)
- Des manipulations de texte (ex: FeaturizeText)

Quelques estimateurs courants

On peut accéder à chaque type de problème directement sur MLContext :

```
mlContext.BinaryClassification
```

Les estimateurs de type Trainer se trouvent dans la propriété Trainers de chaque problème :

```
mlContext.BinaryClassification.Trainers.LbfgsLogisticRegression()
```

Opérations sur les modèles

Pour créer un modèle : appeler la méthode Fit(dv) du pipeline avec la DataView de l'ensemble d'entraînement.

Pour évaluer le modèle : appliquer Transform(dv) sur la DataView de l'ensemble de test. En fonction du problème choisi, une fonction Evaluate permet d'obtenir des métriques.

Pour créer un moteur de prédiction : `mlContext.Model.CreatePredictionEngine`

Chargement de modèles pré-entrainés

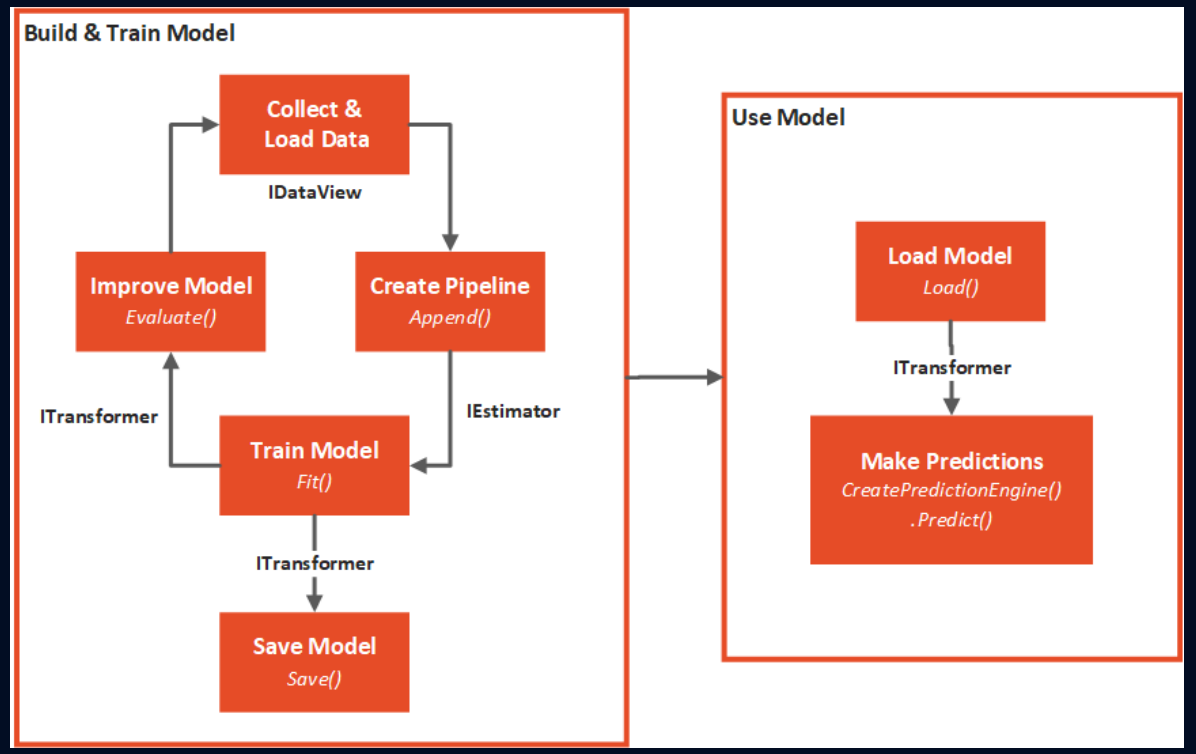
Souvent, on aura sauvegardé un modèle qui devra être rechargé.

`mlContext.Model.Save, mlContext.Model.Load`

On pourra également charger des modèles pré-entrainés à l'extérieur de ML.NET :

- Tensorflow (NuGet : Microsoft.ML.Tensorflow)
- ONNX (NuGet : Microsoft.ML.OnnxTransformer / OnnxRuntime)

Flux de travail avec ML.NET



Source : https://rubikscode.net/2022/08/29/machine-learning-with-ml-net-introduction/

À propos de l'explicabilité des modèles ML.NET

Grâce aux DataViews, on peut facilement retrouver les transformations intermédiaires des données initiales, ayant abouti au résultat final.

Sur certains problèmes, on retrouve également une méthode qui offre des indications sur les features qui ont le plus d'impact sur le résultat final.

`MLContext.Regression.PermutationFeatureImportance`

Enfin, on dispose d'un estimateur qui peut être ajouté à un pipeline, et permet de récupérer combien chaque feature a contribué à une prédiction.

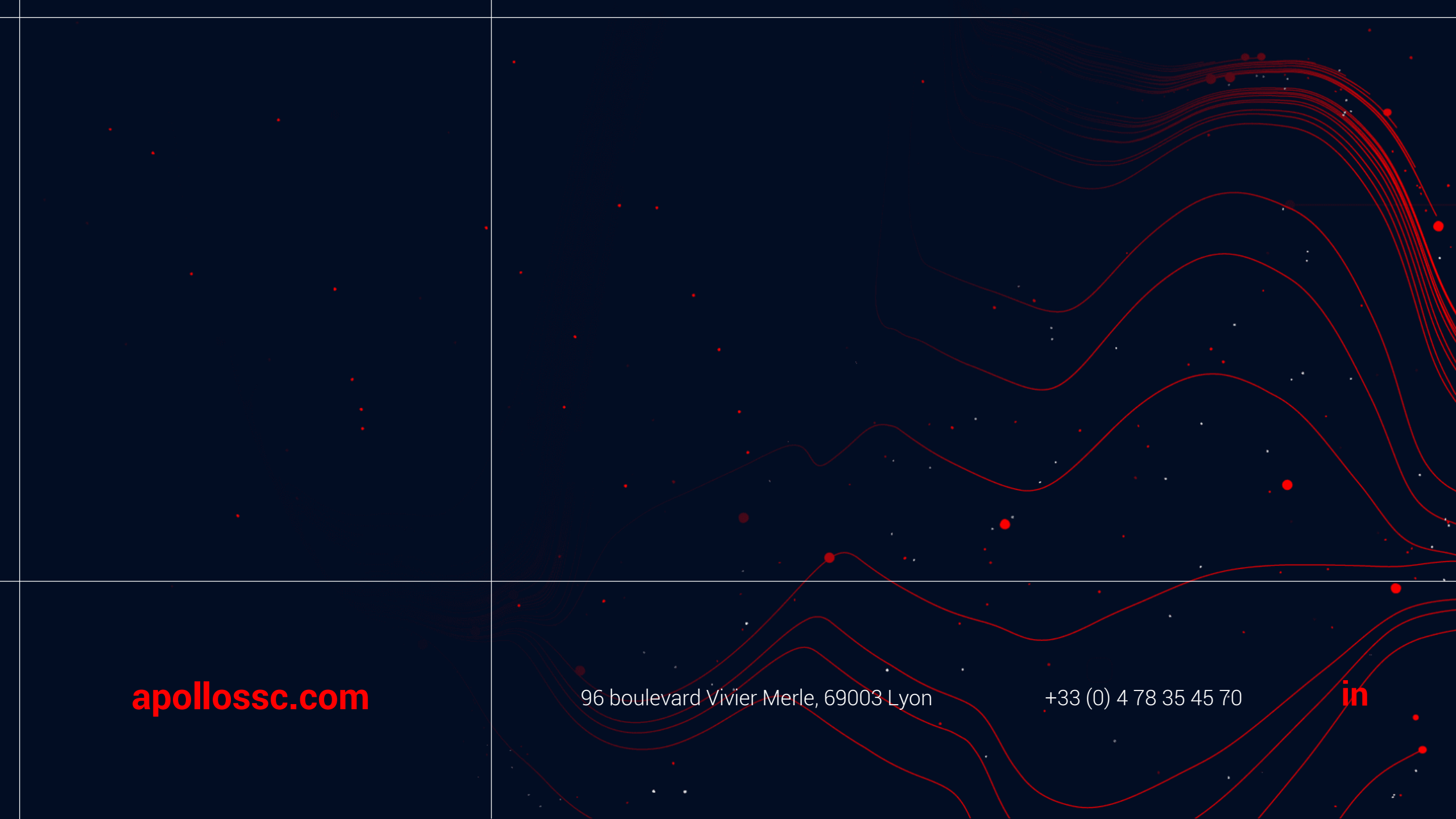
`MLContext.Transforms.CalculateFeatureContribution`

04

Exemples

05

À vous de jouer !



apollossc.com

96 boulevard Vivier Merle, 69003 Lyon

+33 (0) 4 78 35 45 70

in