

Projet sur la reconnaissance d'activité avec un Smartphone

Dieula Sylvie Dareus

Plan

- Introduction
- Bibliothèques utilisées
- Jeu de données
- Algorithme utilisé
- Méthode suivie
- Résultats obtenus

Introduction

Dans cette présentation, nous allons présenter un modèle de reconnaissance d'activité humaine utilisant un smartphone. L'utilisation de la bibliothèque de Machine Learning ML.Net et l'algorithme SVM (Machine à vecteurs de support) qui permettra de classifier les données de capteurs et de reconnaître les activités humaines à partir de ces données.

Le modèle SVM a été implémenté en utilisant la bibliothèque ML.Net de Microsoft. Tout d'abord, les données ont été préparées en les convertissant en un format utilisable par ML.Net. Les données ont ensuite été divisées en données d'entraînement et données de test.

Le modèle SVM a été entraîné sur les données d'entraînement et testé sur les données de test pour évaluer sa performance.

Bibliothèques utilisées

CSVHelper: CsvHelper est une bibliothèque open-source pour .NET qui permet de lire et écrire facilement des fichiers CSV (Comma-Separated Values) en utilisant des objets de type fort. La bibliothèque prend en charge les opérations de lecture et d'écriture de données à partir de et vers des fichiers CSV, ainsi que la manipulation de ces données une fois qu'elles ont été lues ou écrites.

ML.Net : ML.NET qui est un framework puissant et flexible open source créé par Microsoft pour le développement d'applications d'apprentissage automatique, il permet aux développeurs .NET d'ajouter des fonctionnalités d'apprentissage automatique à leurs applications de manière simple et efficace

Jeu de données

Le jeu de données utilisé pour ce projet est le "**Human Activity Recognition Using Smartphones Data Set**", disponible sur Kaggle. Le jeu de données contient des données de capteurs pour six activités différentes : WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, LAYING (marcher, monter les escaliers, descendre les escaliers, s'asseoir, se lever et se tenir debout).

Ce jeu de données contient des données collectées auprès de 30 sujets portant des smartphones Samsung Galaxy S II sur leur taille, tout en effectuant six activités précédentes. Les données sont collectées en utilisant l'accéléromètre et le gyroscope du smartphone, et chaque sujet a effectué chaque activité plusieurs fois.

Algorithme Machine à Vecteur de Support (SVM)

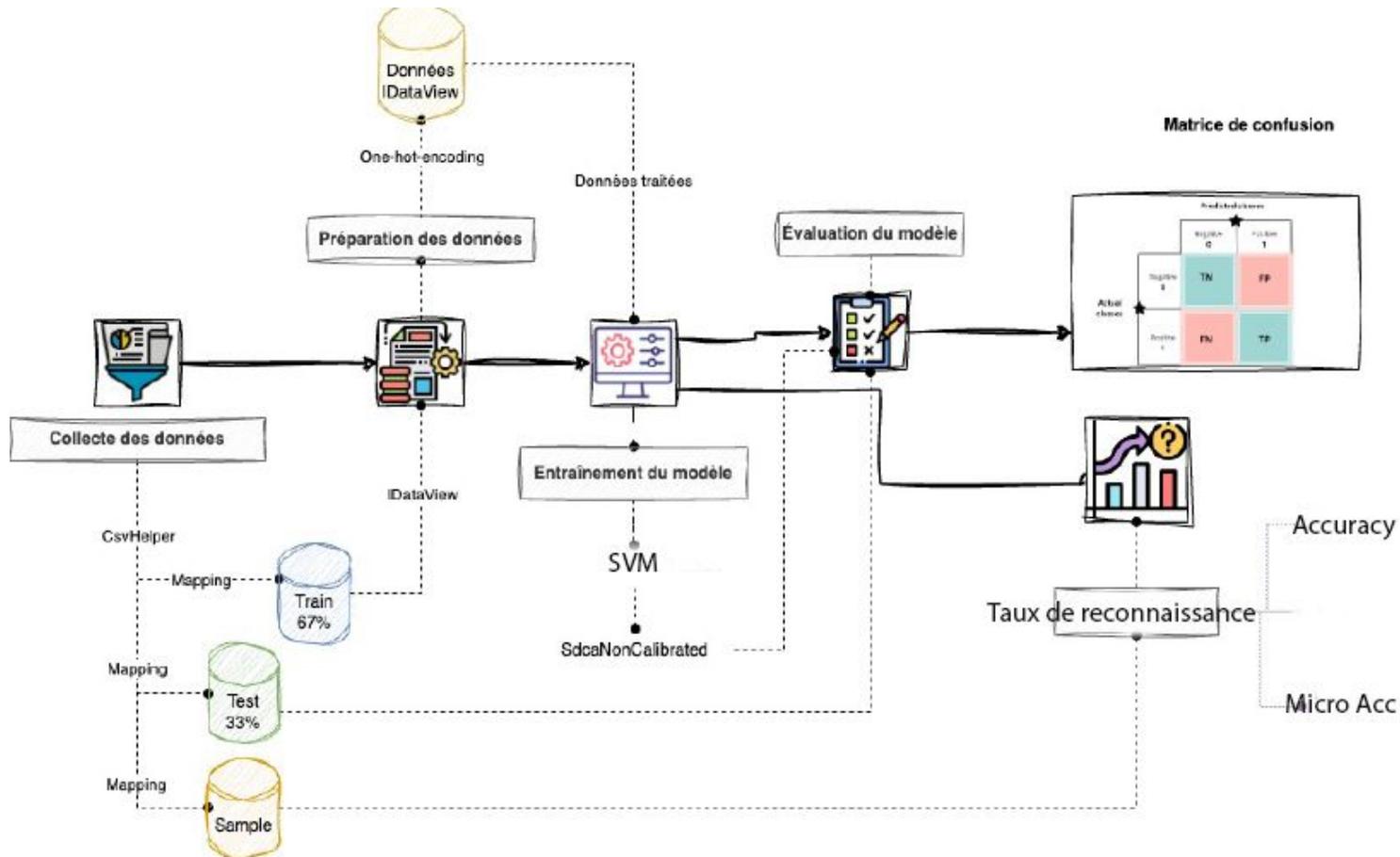
SVM (Support Vector Machine) est un algorithme d'apprentissage automatique supervisé utilisé pour la classification et la régression selon le type de problème à résoudre. Les SVM sont particulièrement utiles pour les problèmes de classification binaire, mais ils peuvent également être utilisés pour la classification multiclass

L'objectif de SVM est de trouver la meilleure séparation possible entre deux classes de données en trouvant un hyperplan qui maximise la marge entre les deux classes.

L'hyperplan est une frontière de décision qui sépare les données en deux régions, une région pour chaque classe.

- Les SVM ont une capacité de généralisation élevée
- Les SVM sont efficaces pour les jeux de données de grande dimension
- Les SVM peuvent gérer des données non linéaires

Méthode suivie



Taux de reconnaissance

La méthode **MeasureAccuracy** est utilisée pour évaluer la précision du modèle entraîné. Elle utilise les données de test stockées dans une variable (TestDataLoaded) pour tester le modèle et mesurer le taux de reconnaissance.

Dans le code je commence par transformer les données de test en utilisant le modèle entraîné. Ensuite, la méthode utilise la méthode MulticlassClassification.Evalueate de MLContext pour évaluer les performances du modèle en utilisant les données transformées.

```
Taux de reconnaissance
```

```
-----  
Accuracy : 0.9859998017556092
```

```
Micro Acc : 0.9849020674646355
```

```
LogLoss : 21.55075835467622
```

```
LogLossReduction : -11.080853325374624
```

```
----- Matrice de confusion -----
```

Matrice de confusion

La matrice de confusion ici est un tableau qui montre les performances du modèle pour chaque classe en termes de vrai positif, faux positif, vrai négatif et faux négatif. Les valeurs réelles et prédites sont comparées pour chaque instance du jeu de données pour évaluer les performances du modèle.

Le modèle a des performances élevées avec des scores de précision allant de 95,46% à 99,65% et des scores de rappel allant de 95,71% à 100%.

```
----- Matrice de confusion -----  
-----  
Confusion table  
=====
```

PREDICTED		0	1	2	3	4	5	Recall
TRUTH	0. STANDING	1,315	59	0	0	0	0	0.9571
	1. SITTING	41	1,240	5	0	0	0	0.9642
	2. LAYING	0	0	1,407	0	0	0	1.0000
	3. WALKING	0	0	0	1,222	2	2	0.9967
	4. WALKING_DOWNSTAIRS	0	0	0	0	984	2	0.9980
	5. WALKING_UPSTAIRS	0	0	0	0	0	1,073	1.0000

```
-----  
Precision | 0.9698 | 0.9546 | 0.9965 | 1.0000 | 0.9980 | 0.9963 |
```

Merci !