

Rapport de TP IDM

-

Duchateau Florian

Lebosq Yannaël

# Introduction

L'idée du projet qui nous a été proposé est de développer un langage, le Multi Machine Learning Language qui doit pouvoir exprimer un algorithme de machine learning qui sera compilé dans différents langages (python, R, weeka ...), celui ci constitue donc un DSL.

C'est particulièrement utile puisque le machine learning fait usage de nombreux algorithmes variés et que chaque langage de programmation en fait sa propre implémentation, on offrira donc un outil qui permet de faire abstraction de ces différents implémentations pour obtenir le résultat souhaité.

A terme, grâce à ce langage spécialisé il sera facile de lancer des comparaisons de performances entre ces différents langages pour cette tâche donnée.

## Création de notre propre grammaire et prise en main d'Xtext.

Dans un premier temps nous avons proposé une première grammaire en Xtext dont voici la syntaxe :

```
grammar org.xtext.example.mydsl.MyDsl with org.eclipse.xtext.common.Terminals

generate myDsl "http://www.xtext.org/example/mydsl/MyDsl"

Programm :
    ('begin'
    init+=InitDataSet
    data+=DataStrategy?
    algo+=AlgoChoice
    print+=Print
    'end')+;

InitDataSet:
    'initCsv' url=STRING;

DataStrategy:
    Split | Cross;

Split:
    'split' (targetCol=STRING |
    splitSize=DOUBLE |
    targetAndSize=STRING DOUBLE);
```

```

Cross:
    'cross' test=ID;

AlgoChoice:
    'use' algo=Algo params+=(ParamAlgo)*;

ParamAlgo:
    name=ParamNames '=' ParamTypes;

ParamTypes:
    INT|DOUBLE|STRING|'None'|'False'|'True';

Print:
    'print' params+=(ParamPrint)*;

enum Algo:
    tree='tree' |
    linear='linearRegression';

enum ParamNames:
    criterion='criterion' |
    maxDepth='max_depth' |
    presort='presort';

enum ParamPrint:
    squared='squared' |
    absolute='absolute';

terminal DOUBLE : INT.INT;

```

Avec le recul, on constate que certains de nos choix d'implémentation étaient plutôt naïfs, par exemple, nous n'avons pas anticipé que l'on puisse cibler finement quelles colonnes prendre pour nos predictors et quelle colonne prendre pour notre prédictive.

# Développement d'un compilateur

Partant désormais d'une grammaire commune à tout le groupe de TP nous nous sommes attaqué au développement du compilateur permettant de faire la traduction MML -> Python. Ecore ayant généré les classes et les attributs nécessaires à cette tâche nous nous sommes consacrés à l'analyse de la grammaire et à sa traduction dans le langage cible.

L'une des principales difficultés que nous avons rencontré lors de cette étape est notre manque de connaissance sur le langage python et sur le machine learning en général. On a beaucoup appris sur ces sujets au travers du tp, mais cela a pas mal freiné notre vitesse de progression.

Lorsque nous avons été satisfait de ce premier compilateur vers python nous avons rédigés un certains nombre de tests junit pour essayer de couvrir un maximum de cas et vérifier que notre compilateur donnait le bon résultat.

## Le reste à faire

Pour parfaire ce tp, il faudrait qu'on rajoute d'autres langages cibles (Weka, R etc.), cela nous permettrait de lancer des analyses comparatives poussées de performances.

Certaines des fonctionnalités proposées par la grammaire fournie pourraient également être étoffées pour intégrer plus de nuances dans les algorithmes de machine learning afin de pouvoir en réaliser plus.

## Conclusion

Ce projet nous a permis de nous familiariser avec des outils essentiels à la programmation dirigée par les modèles. Xtext pour poser notre grammaire, ainsi qu'Ecore qui a généré nos objets que l'on manipule ensuite pour gérer notre compilation.

Ceci nous a permis de bien comprendre chaque étapes de la créations d'un dsl, ce qui sera très utile pour poursuivre ce projet et intégrer les autres langages qui nous reste à faire.