

Enseignant(s)

**VIDAL Nicolas**

Email(s)

[nvidal@myges.fr](mailto:nvidal@myges.fr)

## 4A RVJV Machine Learning T1

### 1 Matières, formations et groupes

Matière liée au projet :

Formations : -

Nombre d'étudiant  
par groupe :

**2**

Règles de constitution des groupes: **Libre**

Charge de travail  
estimée par étudiant : **30,00 h**

### 2 Sujet(s) du projet

Type de sujet : **Imposé**

### 3 Détails du projet

**Objectif du projet (à la fin du projet les étudiants sauront réaliser un...)**

Implémenter et utiliser des modèles et algorithmes simples relatifs au Machine Learning, combiner le tout dans un cas pratique réel.

## Descriptif détaillé

### Cadre général

Les étudiants devront remettre une application (sous la forme d'un client lourd, d'un site web ou d'une application mobile) interagissant avec une API hébergeant différents modèles pré-entraînés grâce aux datasets appropriés et préalablement constitués.

L'essentiel des démarches des étudiants, des résultats obtenus et de leurs analyses devront être présentés également lors d'une soutenance.

Avant d'appliquer les algorithmes et modèles vus en cours à la problématique choisie, il sera impératif de démontrer la justesse de l'implémentation de ces derniers sur les cas de tests proposés. Ainsi, il est proposé d'implémenter ceux-ci sur des jeux de données classiques (données linéairement séparables, non linéairement séparables, tâches de classification, tâches de régression, etc.) tels que vus en cours de Machine Learning.

Les modèles et algorithmes à appliquer au projet de test sont :

- Modèle linéaire
- Perceptron Multi Couches
- Radial Basis Function Network
- Optionnel : SVM ou proposition du groupe d'étudiants

L'ensemble de ces modèles et algorithmes devront être implémentés en C, C++, Rust, Zig (ou C# exceptionnellement) de manière à pouvoir être aisément utilisé comme une bibliothèque dynamique manipulée depuis Unity ou l'Unreal Engine ou autre.

Ces mêmes modèles et algorithmes devront être appliqués à la recherche d'un modèle tentant de solutionner une problématique applicative complexe, pour laquelle une implémentation humaine serait difficile (ex. jouer à la manière d'un humain à un jeu vidéo d'arcade classique).

Devront être mis en évidence par le biais de courbes les phénomènes vus en cours tels que (liste non exhaustive) :

- Le sur apprentissage
- La mise en évidence de biais dans la base d'exemples
- La difficulté de trouver le modèle « suffisamment complexe pour correctement traiter les données d'apprentissage » mais « suffisamment simple pour bien généraliser »
- Etc.

Il est bon de noter que l'implémentation sera nécessaire pour obtenir des résultats, mais qu'il ne s'agit que de la première étape. L'intérêt principal du projet se situant dans la capacité des étudiants à être capable de commenter les résultats obtenus et à porter un regard critique sur l'ensemble des outils vus au travers de différents cours et à l'utilité de chacun face à une problématique issue du monde réel.

Les étudiants seront également encouragés à proposer d'autres modèles et algorithmes si ceux-ci s'avèrent pertinent face au choix de leur problématique. De même ils pourront tout à fait faire usage de la bibliothèque fournie de tensorflow/keras/pytorch/... à titre de comparaison de leurs implémentations personnelles.

L'utilisation de GIT sera impérative pour assurer un suivi du projet aisé pour l'ensemble des membres du groupe ainsi que de l'encadrant.

L'utilisation d'implémentation externes des algorithmes et modèles vus en cours est proscrite (hormis à titre comparatif).

-----  
Concernant l'usage de code généré via des LLMs ou équivalents :

- Tout groupe d'étudiant n'étant pas en mesure de :
  - > Modifier son code durant la soutenance pour implémenter 'en live' une demande du jury
  - > Expliquer le fonctionnement d'une section du code produit
  - > ...

se verra attribuer un 0 pour l'évaluation de la section concernée.

Le travail de chaque membre du groupe devra être clairement identifié (header de fichier, document de suivi, etc.)

Pour entraîner leur modèle dans le but de l'utiliser au sein d'une application cliente, les étudiants devront se munir d'un dataset étiqueté approprié.

Dans le but de permettre aux étudiants de réaliser une synergie entre ce projet et le projet Godot, les étudiants devront s'attaquer à une tâche d'imitation learning pour créer un joueur artificiel jouant 'à la manière d'un humain' au jeu qu'ils auront créé.

Il est souhaitable que la visualisation du joueur artificiel 'jouant' en live soit réalisée dans Unity pour mettre en évidence :

- la réutilisabilité de leur game engine logique indépendant du moteur
- la possibilité d'utiliser leur bibliothèque dans un environnement de jeu vidéo

Le jeu doit comme précisé dans le projet godot un gameplay simple type jeu Atari 2600 (Pong / Breakout / ...)

Ainsi, à partir du jeu implémenté, les étudiants devront se constituer un dataset étiqueté composé de couples "états du jeu" / "action choisie par l'humain".

Une fois ce dataset réalisé, ils devront faire des expérimentations pour entraîner des modèles de différents types (Linéaire / PMC / RBF / ...).

Faire varier les hyperparamètres pour réussir à trouver pour chaque type de modèle une version essayant de trouver un compromis entre underfitting et overfitting.

Permettre l'utilisation dans Unity et/ou Godot d'un modèle entraîné se comportement 'comme un humain'.

#### Ouvrages de référence (livres, articles, revues, sites web...)

- MOOC de référence pour la partie théorique des modèles et algorithmes : <https://work.caltech.edu/telecourse.html>

#### Outils informatiques à installer

- IntelliJ CLion ou Rust Rover (partie lib c++ ou Rust) ou Visual Studio
- Partie cliente au choix :
  - \* Unity / Rider
  - \* Front Web quelconque
  - \* JavaFX / XXX

## 4 Livrables et étapes de suivi

1	Etape intermédiaire	Problématiques applicatives choisies, repository git créé, jeu choisi	jeudi 02/10/2025 11h30
2	Etape intermédiaire	Modèle linéaire appliqué aux cas de tests <ul style="list-style-type: none"><li>• Livrables :<ul style="list-style-type: none"><li>o Projet de test opérationnel et sources</li></ul></li></ul>	lundi 27/10/2025 23h59
3	Etape intermédiaire	Modèle linéaire et PMC appliqué aux cas de tests ainsi qu'à une portion du dataset en cours de constitution <ul style="list-style-type: none"><li>• Livrables :<ul style="list-style-type: none"><li>o Projet de test opérationnel et sources</li><li>o Léger rapport commentant les résultats observés</li><li>o Début d'application</li></ul></li></ul>	dimanche 23/11/2025 23h59

4	Rendu final	<p>Radial Basis Function Network et SVM (ou autre) appliqué aux cas de tests</p> <ul style="list-style-type: none"> <li>• Démonstration de l'implémentation de l'ensemble des algorithmes et modèles de réseaux de neurones étudiés sur les cas de tests ainsi que sur le dataset constitué</li> <li>• Possibilité de sauvegarder/charger des modèles entraînés et de les utiliser grâce à l'application sur de nouvelles données</li> <li>• Livrables : <ul style="list-style-type: none"> <li>o Projet de démonstration et sources</li> </ul> </li> </ul> <p>Soutenance publique</p> <ul style="list-style-type: none"> <li>• Présentation de la démarche scientifique des étudiants pour aborder leurs problématique à l'aide des méthodes et algorithmes vu en cours. Rapide démonstration, analyse et critique des résultats obtenus.</li> <li>• Livrables : <ul style="list-style-type: none"> <li>o Slides</li> <li>o Projet de démonstration et sources</li> </ul> </li> </ul>	<p><b>dimanche</b>  <b>14/12/2025</b>  <b>23h59</b></p>
---	-------------	--	---

5	Soutenance		
Durée de présentation par groupe :	<b>20 min</b>	Audience :	<b>A huis clos</b>
Type de présentation :	<b>Présentation / PowerPoint - Démonstration</b>		
Précisions :			