**SYLLABUS / PLAN DE COURS**

*Règle de nommage à appliquer pour le nom du fichier : « Syllabus\_ » puis Année puis Nom de la matière puis Nom de l’Enseignant.*

*Exemple : « Syllabus\_2\_LangageC\_Dupont »*

|  |  |  |  |
| --- | --- | --- | --- |
| **Matière :** | **Machine Learning** | | |
| *Année et spécialisation(s) :* | ***5A-DJV*** | | |
| Volume horaire du cours : | 30h | Charge de travail personnel de l’étudiant : | 15h | |

|  |  |  |  |
| --- | --- | --- | --- |
| Enseignant : | Nicolas Vidal | Mail de l’enseignant : | pro@nicolasvidal.fr |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Contrôle continu : | Questions de Cours | Evaluation de TP | Travail personnel hors cours | Suivi intermédiaire de projet | Dossier Individuel | Dossier Collectif | Exposé |
| *Cocher la case* |  | X |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Evaluation finale : | Partiel de 2 heures | | Projet |
| *Cocher la case* |  | | X |
|  | Documents autorisés | Documents non autorisés | *Déclarer le projet sur MyGES* |
| *Cocher la case* |  |  |

**Objectifs pédagogiques**

|  |
| --- |
| Sensibiliser les étudiants aux multiples problématiques liées à l’apprentissage artificiel, et plus particulièrement à l’apprentissage supervisé. |

**Méthodologie utilisée**

|  |
| --- |
| Apprentissage par la pratique et dirigé par l’exemple |

**Ouvrages de Référence (livres, articles, revues, sites Web)**

|  |
| --- |
| <https://work.caltech.edu/telecourse.html>  <https://www.coursera.org/learn/machine-learning/>  <http://www.ai-junkie.com/> |

**Outils informatiques à installer**

|  |
| --- |
| Un environnement de développement 3D temps réel (Unity, UnrealEngine, etc.), un outil d’analyse numérique (Octave, Mathematica, etc.) |

**Programme détaillé**

|  |
| --- |
| 1. Installation de l’écosystème    1. Unity/Unreal Engine/Blender    2. Octave/Mathematica    3. Présentation du projet 2. Qu’est-ce qu’apprendre ?    1. Les différentes formes d’apprentissage       1. Supervisé       2. Semi supervisé       3. Non supervisé    2. Apprentissage supervisé       1. Apprendre par cœur VS généraliser       2. Qu’est-ce que généraliser ?       3. Quelles validations théoriques sur la généralisation ?    3. Classification VS Régression       1. Exemples 3. Classification et Séparations linéaires    1. Retours sur le perceptron    2. Apprentissage sur un modèle linéaire       1. Règle de Hebb       2. Règle de Rosenblatt       3. ‘Pas’ d’apprentissage, et élan    3. Exemples 4. Données non linéairement séparables    1. Travail dans un espace de plus grande dimension    2. Exemples    3. Complexification des modèles et impact sur la généralisation 5. Perceptron Multi Couches    1. Principes    2. Rétro propagation du gradient    3. Exemples    4. Améliorations à la rétro propagation du gradient 6. Surapprentissage    1. Retour sur la notion d’erreur de classement    2. Combattre le surapprentissage de manière empirique       1. Prétraitement des données       2. Boosting       3. Estimation de la complexité du modèle à utiliser 7. Support Vector Machines    1. Notion de ‘meilleure’ séparation linéaire    2. Brèves notions de programmation quadratique (et rappels de programmation linéaire)    3. Considérations théoriques    4. Projection dans un espace de dimension infinie 8. Machines à noyau    1. Problématiques liées aux SVM classiques    2. Choix du noyau 9. Réseau de neurones à fonction de Base Radiales    1. Relations avec les k plus proches voisins    2. Apprentissage non supervisé et Algorithme de Lloyd    3. Comparaison avec les précédents modèles 10. Conclusion     1. Recommandations et pièges     2. Quel modèle choisir ?     3. Quels autres modèles ? 11. Ouverture     1. Apprentissage par renforcement     2. Métaheuristiques et Apprentissage Artificiel     3. Deep learning |