

Inteligencia Artificial

Código: 102768
Créditos ECTS: 6

Titulación	Tipo	Curso	Semestre
2502441 Ingeniería Informática	OB	2	2

Contacto

Nombre: Maria Vanrell Martorell
Correo electrónico: maria.vanrell@uab.cat

Uso de idiomas

Lengua vehicular mayoritaria: catalán (cat)
Algún grupo íntegramente en inglés: Sí
Algún grupo íntegramente en catalán: Sí
Algún grupo íntegramente en español: No

Equipo docente

Pau Folch Codera

Equipo docente externo a la UAB

Marcel Vilalta Soler

Prerequisitos

Dada la naturaleza introductoria de esta asignatura, se asume que los estudiantes no tienen ningún conocimiento previo sobre el tema. Es por tanto responsabilidad de la asignatura dar a los estudiantes los medios para la adquisición de los conocimientos descritos en el apartado de contenidos de esta guía.

Sobre otros conocimientos que se esperan del estudiante:

- Haber cursado un bachillerato científico o tecnológico.
- Haber cursado la asignatura de programación de 1º y 2º de esta carrera
- Tener nociones básicas de programación en Python
- Tener competencias a nivel de usuario de alguna de las siguientes plataformas: Windows, Mac o Linux.
- Tener acceso a un ordenador, preferiblemente portátil.

Objetivos y contextualización

Esta asignatura tiene un carácter general e introductorio al campo de la inteligencia artificial (IA). Los objetivos principales que se persiguen son dos:

a) familiarizar los estudiantes en la resolución de problemas de IA con sus particularidades en la representación, la evaluación y la metodología específica de resolución, y

b) hacer que el estudiante conozca un amplio espectro de técnicas y algoritmos básicos que permitan solucionar los problemas planteados y mejorar sus habilidades de programación.

Más concretamente estos objetivos perseguirán:

- Proporcionar una introducción histórica y de los objetivos del campo de la IA.
- Introducir a los estudiantes al problema de la selección de una buena representación de conocimiento como eje principal para la resolución de los problemas de IA.
- Familiarizar a los estudiantes con 4 representaciones de conocimiento básicas y sus algoritmos correspondientes.
- Dotar a los estudiantes con la capacidad de diseñar soluciones a problemas contextualizados.
- Dotar a los estudiantes con la capacidad de representar y argumentar la adecuación de las soluciones diseñadas.

Competencias

- Adquirir hábitos de pensamiento.
- Adquirir hábitos de trabajo personal.
- Capacidad para concebir, desarrollar y mantener sistemas, servicios y aplicaciones informáticas empleando los métodos de la ingeniería del software como instrumento para el aseguramiento de su calidad.
- Capacidad para concebir, redactar, organizar, planificar, desarrollar y firmar proyectos en el ámbito de la ingeniería en informática que tengan por objeto la concepción, el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas.
- Conocimiento y aplicación de los principios fundamentales y técnicas básicas de los sistemas inteligentes y su aplicación práctica.
- Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.
- Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.

Resultados de aprendizaje

1. Aplicar diferentes métodos, arquitecturas, conceptos y patrones software para el diseño, desarrollo y análisis de programas informáticos.
2. Desarrollar el pensamiento sistémico.
3. Diseñar, desarrollar, seleccionar y evaluar aplicaciones asegurando su fiabilidad y seguridad.
4. Identificar la complejidad computacional de un algoritmo en términos de recursos de memoria y tiempo de ejecución.
5. Identificar y definir la solución computacional al reconocimiento de patrones, la toma de decisiones basada en la exploración de alternativas, el razonamiento con incertidumbre y la planificación de tareas.
6. Seleccionar y aplicar la combinación de estructuras de datos y estrategia de resolución más apropiada para resolver de modo eficiente un problema informático.
7. Trabajar de forma autónoma.

Contenido

1. Introducción a la IA. Objetivos y breve historia del campo. Definición de agentes racionales, metodología de resolución de problemas y representación de conocimiento.

2. Resolución de problemas por exploración de alternativas

2.1. Búsqueda informada. Definiciones básicas de búsqueda y revisión de algoritmos de búsqueda no informada. Análisis de algoritmos de búsqueda: Completitud, Optimalidad y Complejidad. Concepto de

heurística y ejemplos. Búsqueda Informada: básica y óptima. A* y sus propiedades. Concepto de factor de ramificación efectivo. Propiedades interesantes de las heurísticas.

2.2. Búsqueda local. Definiciones básicas, pros y contras. Interpretación de la búsqueda local como la exploración de la función heurística. Búsqueda local con objetivo conocido: Algoritmo *Hill-climbing*. Problemas de la búsqueda local: máximos locales, mesetas y crestas. Búsqueda local con objetivo desconocido: *Steepest Ascent*, *Steepest Ascent* con control de máximos locales, *Simulated Annealing*

2.3. Búsqueda con adversarios basada en Minimax. Definiciones básicas. Algoritmo del *Minimax* y Poda alfa-beta. Análisis de complejidad. Modificaciones del *Minimax*: profundización progresiva, heurística de extensión singular (efecto Horizonte). Ejemplos de funciones heurísticas.

2.4. Búsqueda con adversarios basada en Simulaciones aleatorias. Definiciones básicas. Algoritmo de Monte-Carlo Tree Search. Ejemplos.

3. Resolución de problemas de reconocimiento de patrones.

3.1. Técnicas estadísticas. Espacios de características como representaciones para el razonamiento basado en casos. Definiciones básicas sobre espacios de características. Selección de características y reducción de la dimensión. Clasificación basada en aprendizaje supervisado: funciones de decisión, asunciones sobre la distribución de los datos, búsqueda local para encontrar las funciones de decisión, decisión basada en el vecino más próximo. Clasificación basada en aprendizaje no supervisado: algoritmo *k-means*, búsqueda de la mejor k con el discriminante de Fisher. Análisis de los algoritmos.

3.2. Técnicas estructurales. Grafos y redes semánticas. Definiciones básicas y representación con matrices de adyacencia. El problema de la correspondencia de grafos, algoritmos básicos, mejoras y complejidad. Correspondencia de grafos inexacta: medidas de similitud, distancia de edición. Caso de estudio: *String-matching*.

4. Resolución de problemas de razonamiento

4.1. Lógica y mecanismos de inferencia. Representación del conocimiento: lógica proposicional y lógica de predicados. Revisión general de algoritmos básicos: deducción natural, mecanismo de la resolución, unificación y algoritmo de paso a forma clausal. Definiciones básicas sobre sistemas basados en reglas: base de reglas, memoria de trabajo, encadenamiento de reglas y estrategias de resolución de conflictos.

Metodología

La inteligencia artificial se define por el tipo de problemas que se intentan solucionar. Por tanto será la tipología de problemas quien dirigirá la organización de los contenidos. Se harán tres tipos de sesiones:

Clases de teoría: Son sesiones que combinarán clases de tipo magistral en las que el profesor explica los contenidos teóricos de la asignatura, con sesiones de grupos reducidos de estudiantes donde se resolverán cuestionarios que permitan evaluar que el estudiante va adquiriendo los contenidos teóricos dados en las clases teóricas o bien a través de vídeos.

Clases de problemas: Son clases en grupos reducidos de estudiantes para facilitar la interacción. En estas clases se persigue reforzar la comprensión de los temas planteados en las clases de teoría planteando casos prácticos que requieren el diseño de una solución en la que se usen los métodos vistos en las clases de teoría. Es imposible seguir estas clases de problemas si no se van siguiendo las clases de teoría.

Sesiones Prácticas: Son sesiones en las que se llevan a término diferentes tipos de actividades relacionadas con la ejecución en grupo de un proyecto contextualizado, se desarrollan 3 tipos de actividades: (a) sesiones de trabajo en equipo tutorizadas por el profesor, (b) sesiones para la resolución de cuestionarios para evaluar el conocimiento individual de los estudiantes sobre el código desarrollado, (c) sesiones de presentación de resultados, en las que todos los miembros del equipo de trabajo explican los resultados de su proyecto.

Las competencias transversales que se trabajan son:

- La adquisición de hábitos de pensamiento, y en concreto el desarrollo del pensamiento sistémico.

Esta competencia se adquiere tanto a partir de los conocimientos teóricos, ya que todos los contenidos están basados en la construcción de sistemas inteligentes con módulos que interactúan, como desde el punto de vista de las prácticas en que los estudiantes trabajan en equipo y experimentan la dependencia que se genera en su trabajo. Esta competencia es evaluada en los exámenes de teoría, en la evaluación de las entregas de problemas y en la evaluación del proyecto a todos los niveles.

- La adquisición de hábitos de trabajo personal, y en concreto el trabajar de manera autónoma.

Esta competencia se adquiere también con la capacidad de estudiar autónomamente todos los contenidos teóricos, con la entrega individual de los problemas y con la participación en la práctica. En los tres ámbitos el trabajo individual es evaluado, en teoría a través de los exámenes, en problemas a través de la corrección de los problemas entregados y en prácticas a partir del cuestionario individual, la participación del estudiante en la presentación conjunta de los resultados y la intra-evaluación que hacen los miembros del equipo entre ellos.

Nota: se reservarán 15 minutos de una clase dentro del calendario establecido por el centro o por la titulación para que el alumnado rellene las encuestas de evaluación de la actuación del profesorado y de evaluación de la asignatura o módulo.

Actividades

Título	Horas	ECTS	Resultados de aprendizaje
Tipo: Dirigidas			
Clases teóricas	30	1,2	2, 5, 4
Tipo: Supervisadas			
Clases de problemas	10	0,4	2, 5, 4, 7
Sesiones de prácticas	12	0,48	1, 5, 6
Tipo: Autónomas			
Estudio individual	38	1,52	2, 5, 7
Trabajo práctico en el proyecto	51	2,04	1, 2, 3, 4, 6, 7

Evaluación

Para evaluar el nivel de aprendizaje del estudiante se establece una fórmula que combina el aprendizaje de conocimientos, la capacidad de resolución de problemas y las capacidades de trabajar en equipo, así como de la presentación de los resultados obtenidos.

La nota final se calcula ponderadamente de la siguiente manera y de acuerdo con las diferentes actividades que se llevan a término durante el curso:

$$\text{Nota final} = 0.5 * \text{Nota Teoría} + 0.1 * \text{Nota Problemas} + 0.4 * \text{Nota Prácticas}$$

Se aplicará esta fórmula siempre que la nota de teoría sea superior o igual a 5, y la nota de prácticas sea superior o igual a 6. No hay ninguna restricción sobre la nota de problemas. Si el cálculo de la fórmula es superior o igual a 5, pero no llega al mínimo exigido en alguna de las actividades de evaluación, entonces la nota que se pondrá en el expediente es de 4,5.

La Nota de Teoría se calcula a partir del promedio de las notas de los dos exámenes parciales:

Nota Teoría = 0.5 * Nota Parcial 1 + 0.5 * Nota Parcial 2

para poder calcular esta nota, las notas de los dos Parciales deben ser superiores a 3.5. Cuando se publiquen las notas de los exámenes se anunciará fecha y hora de una sesión de revisión de examen.

Examen de recuperación. En el caso que la nota de teoría no llegue al nivel adecuado para tener la nota final aprobada, el estudiante puede presentarse a un examen de recuperación. Para calcular la nota final de un examen se considerará el máximo entre la primera nota y la correspondiente del examen de recuperación.

La nota de problemas tiene como objetivo provocar que el estudiante trabaje con los contenidos de la asignatura de manera continuada y a partir de pequeños problemas que hacen que se vaya familiarizando progresivamente del uso de las técnicas teóricas en situaciones concretas. Como evidencia de este trabajo se pide una entrega semanal de problemas resueltos que irá realizando progresivamente.

Nota Problemas = considera la evaluación de los problemas entregados y el % de entregas realizadas en total

para poder calcular esta nota el % de entregas ha de ser superior al 70%.

La nota de Prácticas tiene un peso esencial en la nota, y pretende que el estudiante implemente y explore las técnicas estudiadas dentro de un proyecto con un objetivo global y contextualizado. Además, el estudiante deberá demostrar sus habilidades para realizar todo este trabajo tanto individualmente como en equipo y presentando convincentemente los resultados. La nota se calcula de la siguiente manera:

Nota Prácticas = 0.5 * Nota Proyecto 1 + 0.5 * Nota Proyecto 2

Para poder calcular esta nota, cada proyecto deberá tener una nota superior o igual a 6. La nota de cada proyecto se calcula a partir de un compendio de notas:

$Nota Proyecto 1 = 0.6 * Nota Código + 0.4 * Nota Cuestionario$

- **Nota Código:** se evalúa mediante una serie de pruebas de test que se irán pasando con cada entrega. Estas evalúan la corrección y la eficiencia del código.
- **Nota Cuestionario:** se evalúa en forma de un examen con acceso al propio código.

La nota de Código y de Cuestionario deben ser superiores o iguales a 5.

Nota Proyecto 2 = 0.75 Nota Grupo + 0.25 * Nota Individual

- **Nota Grupo** = $0.6 * Nota Código + 0.3 * Informe + 0.1 * Presentación Grupo$
- **Nota Individual** = $0.5 * Presentación Individual + 0.5 * Participación Grupal$

La Nota de código y la Nota Individual deben ser superiores o iguales a 5.

Sesiones de recuperación. En caso que la Nota de un Proyecto no llegue a un nivel adecuado para obtener la nota final de aprobado, los estudiantes tendrán una opción de recuperación. En el caso de tener que recuperar la Nota Individual del Proyecto 2, esta recuperación se hará con un Cuestionario similar al del Proyecto 1. En el caso que alguna nota parcial de un proyecto se haya tenido que recuperar, entonces la nota final de ese proyecto será como máximo de 7.

Notas Importantes:

En caso de que no se supere la asignatura debido a que alguna de las actividades de evaluación no llegue a la nota mínima requerida, la nota numérica del expediente será la mínima entre 4.5 y la media ponderada de las notas, con la excepción que la nota numérica del expediente será la mínima entre 3.0 y la media ponderada de las notas en caso que el estudiante haya cometido alguna irregularidad en un acto de evaluación, tal como las que se listan más abajo (y por tanto no será posible aprobar por compensación).

El estudiante tendrá la cualificación de "No Evaluable" en el caso que no haya participado en ninguna de las actividades de evaluación.

El estudiante tendrá una nota de "Matrícula de Honor" si su número en el ranking global está por debajo del número de máximos de MH permitidas en el curso, y su nota final sea superior a un umbral que establecerá el profesor en el momento que tenga todas las notas finales.

Las fechas de la evaluación continuada y la entrega de trabajos se publicarán en el Campus Virtual (cv.uab.cat), y pueden estar sometidas a cambios de programación por motivos de adaptación a posibles incidencias. Siempre se informará a través de cv.uab.cat sobre estos cambios, ya que se entiende que ésta es la plataforma habitual de intercambio de comunicaciones entre profesores y estudiantes.

En el caso de estudiantes repetidores de la asignatura no se hará ninguna convalidación con una nota obtenida en un curso anterior.

Sin perjuicio de otras medidas disciplinarias que se estimen oportunas, y de acuerdo con la normativa académica vigente, las irregularidades cometidas por un estudiante que puedan conducir a una variación de las calificaciones, se calificarán con un cero (0). Las actividades calificadas de esta forma y por este procedimiento no serán recuperables. Si es necesario superar cualquiera de estas actividades de evaluación para aprobar la asignatura, esta asignatura quedará suspendida directamente, sin oportunidad de recuperarla dentro del mismo curso. Estas irregularidades incluyen, entre otras:

- la copia total o parcial de una práctica, informe, o cualquier otra actividad de evaluación;
- dejar copiar;
- presentar un trabajo en grupo no hecho íntegramente por los miembros del grupo;
- presentar como propios materiales elaborados por un tercero, aunque sean traducciones o adaptaciones, y e general trabajos con elementos no originales y exclusivos del estudiante;
- tener dispositivos de comunicación (como teléfonos móviles, Smart watches, etc.) accesibles durante las pruebas de evaluación teórico-prácticas individuales (exámenes).

En resumen: copiar, dejar copiar o plagiar en cualquiera de las actividades de evaluación equivale a un SUSPENSO con nota inferior o igual a 3.0.

Actividades de evaluación

Título	Peso	Horas	ECTS	Resultados de aprendizaje
Defensa del proyecto práctico (Informe+Código+Intra_grupo+Presentación+Cuestionario)	0.4	3	0,12	1, 2, 3, 5, 4, 6, 7
Entrega de problemas resueltos	0.1	0	0	5, 4, 6, 7
Examen individual	0.24	6	0,24	5, 4, 6

Bibliografía

- S. Russell i P. Norvig, Artificial Intelligences - A modern approach. Prentice Hall, 2003, <http://aima.cs.berkeley.edu/>
- Tvester, Donald R., (1998), The Pattern Recognition basis of Artificial Intelligence. IEEE Computer Society.
- Stuart Russell. Human Compatible: AI and the Problem of Control Penguin Publishing Group, Octubre 2019
- Melanie Mitchell. Artificial Intelligence: A Guide for Thinking Humans. Farrar, Straus and Giroux, Octubre 2019.

Vídeos interesantes:

- Documental CODEBREAKER <http://www.turingfilm.com/about/overview>

- Documental Netflix AlphaGo (2017) [https://es.wikipedia.org/wiki/AlphaGo_\(pel%C3%ADcula\)](https://es.wikipedia.org/wiki/AlphaGo_(pel%C3%ADcula))

Software

Entorno de programación en Python con especial atención a la librería Numpy