

Aprendizaje Computacional

Código: 102787
Créditos ECTS: 6

Titulación	Tipo	Curso	Semestre
2502441 Ingeniería Informática	OB	3	1
2502441 Ingeniería Informática	OT	4	1

La metodología docente y la evaluación propuestas en la guía pueden experimentar alguna modificación en función de las restricciones a la presencialidad que impongan las autoridades sanitarias.

Contacto

Nombre: Jordi Gonzalez Sabaté
Correo electrónico: Jordi.Gonzalez@uab.cat

Uso de idiomas

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

Otras observaciones sobre los idiomas

En el caso de estudiantes Erasmus o de fuera de Cataluña, se harán clases adicionales en la lengua común entre estudiante y profesor para que el estudiante pueda seguir la materia sin problemas lingüísticos

Prerequisitos

Se recomienda que para cursar esta asignatura se hayan alcanzado las competencias mínimas en las asignaturas de Álgebra, Cálculo, Matemática Discreta, Fundamentos de Informática y Metodología de la Programación (primer curso), así como Inteligencia Artificial, Estadística y Laboratorio de Programación (segundo curso).

Objetivos y contextualización

La asignatura de Aprendizaje Computacional, se enmarca dentro de la mención de "Computación", junto con asignaturas como "Conocimiento, Razonamiento e Incertidumbre", "Visión por Computador" y "Robótica, Lenguaje y Planificación". Por su temática, esta materia no sólo es para los alumnos que cursan la mención de "Computación", sino para cualquier Mención de la carrera de Ingeniería Informática ya que está estrechamente relacionada con la asignatura de "Inteligencia Artificial" de segundo curso. También es muy recomendable haber cursado y sentirse cómodo con los conceptos vistos en las asignaturas de "Cálculo", "Álgebra" y "Matemática discreta" de primer curso, y de "Estadística" de segundo curso, debido al fuerte contenido matemático de la asignatura.

La asignatura pretende tanto ampliar algunos de los temas desarrollados durante "Inteligencia Artificial", como introducir nuevos problemas asociados a la inteligencia artificial, principalmente el aprendizaje de conceptos y tendencias a partir de datos. Se trata de formar al alumno para ser un "ingeniero o científico de datos", y es una de las profesiones con más futuro y más demandadas en la actualidad por empresas como Facebook, Google, Microsoft y Amazon. De hecho, se prevé que el crecimiento de la demanda de estos profesionales en ingeniería de datos sea exponencial a nivel europeo, sobre todo debido al crecimiento en la generación de datos masivos. Así, el principal objetivo de la asignatura es que el alumno sepa encontrar una buena solución (a veces la mejor es imposible) a problemas en contextos distintos de los tratados en clase, a partir de

identificar las necesidades de representación del conocimiento en cada caso y, según sea éste, aplicar la/s técnica/ s más adecuada/s para generar automáticamente buenos modelos matemáticos que expliquen los datos con un error aceptable.

Los contenidos elegidos para esta asignatura también se dan en las Universidades con más prestigio a nivel mundial, como Stanford, Toronto, Imperial College London, MIT, Carnegie Mellon y Berkeley, por poner los centros más representativos. Por lo tanto, por un lado se ofrece al alumno una oportunidad de alcanzar unos conocimientos y habilidades comparables a los impartidos en las mejores universidades. Por otra parte, el alumno debe ser consciente de que este conocimiento que es punta de lanza del estado del arte tiene una dificultad inherente, implicando un estudio y una dedicación considerables, cuantificada en horas en la sección de Actividades formativas de esta guía. Esto es porque en esta asignatura no sólo se enseñan los contenidos más importantes para convertirse en ingeniero de datos, sino además se trabaja una línea de currículo que permita ampliar el abanico de puestos de trabajo a los que podrá acceder tras la carrera, así como poner las bases metodológicas necesarias para realizar un Master en ingeniería de datos o en inteligencia artificial.

Si se busca una asignatura (i) para aprender seriamente y no sólo para aprobar a la ligera, (ii) para abrir un mercado laboral también a nivel internacional, y (iii) para descubrir los propios límites de programación aprendiendo los algoritmos de aprendizaje computacional más utilizados no sólo para las grandes empresas tecnológicas mencionadas anteriormente, sino también en muchas spin-offs de ingeniería de datos en nuestro país, esta asignatura no defrauda si se pone actitud y aptitud.

Los objetivos de la asignatura se pueden resumir en:

Conocimientos:

- Describir las técnicas básicas de aprendizaje computacional.
- Enumerar los pasos esenciales de los diferentes algoritmos de aprendizaje
- Identificar las ventajas e inconvenientes de los algoritmos de aprendizaje que se explican.
- Resolver problemas computacionales aplicando diferentes técnicas de aprendizaje para encontrar la solución óptima.
- Entender el resultado y las limitaciones de las técnicas de aprendizaje en diferentes casos de estudio.
- Saber escoger el algoritmo de aprendizaje más adecuado para solucionar problemas contextualizados.

Habilidades:

- Reconocer las situaciones en las que la aplicación de algoritmos de aprendizaje computacional puede ser adecuado para solucionar un problema
- Analizar el problema a resolver y diseñar la solución óptima aplicando las técnicas aprendidas
- Redactar documentos técnicos relacionados con el análisis y la solución de un problema
- Programar los algoritmos básicos para solucionar los problemas propuestos
- Evaluar los resultados de la solución implementada y valorar las posibles mejoras
- Defender y argumentar las decisiones tomadas en la solución de los problemas propuestos

Competencias

Ingeniería Informática

- Actitud personal.
- Adquirir hábitos de pensamiento.
- Capacidad para adquirir, obtener, formalizar y representar el conocimiento humano en una forma computable para la resolución de problemas mediante un sistema informático en cualquier ámbito de

aplicación, particularmente los relacionados con aspectos de computación, percepción y actuación en ambientes o entornos inteligentes.

- Capacidad para conocer los fundamentos teóricos de los lenguajes de programación y las técnicas de procesamiento léxico, sintáctico y semántico asociadas, y saber aplicarlas para la creación, diseño y procesamiento de lenguajes.
- Capacidad para conocer y desarrollar técnicas de aprendizaje computacional y diseñar e implementar aplicaciones y sistemas que las utilicen, incluyendo las dedicadas a extracción automática de información y conocimiento a partir de grandes volúmenes de datos.
- Trabajar en equipo.

Resultados de aprendizaje

1. Asumir y respetar el rol de los diversos miembros del equipo, así como los distintos niveles de dependencia del mismo.
2. Conocer y aplicar las técnicas de aprendizaje más adecuadas en diferentes casos de estudio.
3. Conocer y comprender las técnicas de representación del conocimiento humano.
4. Desarrollar la capacidad de análisis, síntesis y prospectiva.
5. Entender y evaluar el resultado y limitaciones de las técnicas de aprendizaje más comunes.
6. Generar propuestas innovadoras y competitivas en la actividad profesional.
7. Identificar, gestionar y resolver conflictos.
8. Resolver problemas computacionales aplicando diferentes mecanismos de aprendizaje necesarios para encontrar la solución óptima.

Contenido

TEMA 1: INTRODUCCIÓN

1.1 Conceptos básicos

1.2 Historia del aprendizaje computacional

TEMA 2: REGRESIÓN DE DATOS

2.1 Regresión lineal y descenso del gradiente

2.2 Regularización y regresión polinomial

TEMA 3: CLASIFICACIÓN DE DATOS

3.1 Regresión logística

3.2 Máquinas de vectores de soporte

TEMA 4: REGRESIÓN Y CLASIFICACIÓN BIOINSPIRADA

4.1 Multilayer Perceptron

4.2 backpropagation

TEMA 5: AGRUPACIÓN DE DATOS

5.1 Memorización de datos: aprendizaje perezoso

5.2 Clustering de datos: k-means y Expectation-Maximization

Metodología

Toda la información de la asignatura y los documentos relacionados que los estudiantes necesiten se encontrarán en la página de Caronte (<https://caronte.uab.cat/course/index.php?categoryid=2>), el menú de la asignatura Aprendizaje Computacional (102787). Servirá para poder ver los materiales, gestionar los grupos de prácticas, hacer las entregas correspondientes, ver las notas, comunicarse con los profesores, etc. Para poder utilizarlo es necesario hacer los siguientes pasos:

- Darse de alta como usuario dando el nombre, NIU, y una foto carnet en formato JPG. Si ya se ha dado de alta por alguna otra asignatura, no es necesario volver a hacer, puede ir al siguiente paso.
- Inscribirse en el tipo de docencia "Aprendizaje Computacional (102787)", dando como código de asignatura "apc2020" (sin las comillas).

En el desarrollo de la asignatura se podrán diferenciar siete tipos de actividades docentes, todas ellas online:

MD1 Exposición de contenidos de teoría: Presentación de los contenidos teóricos a trabajar en la asignatura. Estos contenidos se deberán haber preparado antes de la clase a partir de la lectura de textos, búsqueda de información, etc. Los contenidos presentados estarán directamente relacionados con los problemas, proyectos y seminarios propuestos en otras actividades docentes, de forma que serán la base sobre la que se desarrollarán otras actividades del curso. Los contenidos se encontrarán en la página de Caronte y constarán de dos partes: una presentación donde se exponen los principales conceptos teóricos y matemáticos relacionados con tareas concretas de aprendizaje computacional (este temario será la base del examen teórico de la asignatura, ver apartado evaluación de esta guía docente), y una segunda parte de código en python sobre Jupyter notebooks que ejemplifiquen los detalles de codificación y librerías para implementar en un caso práctico los principales conceptos vistos en la hora anterior. El alumno pues podrá descargar las presentaciones y los notebooks de Python y probar todos los códigos en su ordenador, para hacer las pruebas que hagan falta y poder jugar con los diversos parámetros para acabar de entender las razones de los diferentes rendimientos y precisiones que se alcanzan en una base de datos específica con configuraciones concretas de los algoritmos explicados en la asignatura.

MD2 Resolución de problemas numéricos: Resolución de un conjunto de 4 problemas propuestos a los estudiantes. Todos los temas de teoría irán acompañados de una relación de problemas que el estudiante deberá resolver y entregar. Estas actividades deben permitir al estudiante profundizar en la comprensión y personalizar el conocimiento teórico en un caso numérico concreto. Se plantearán pues unos ejemplos de datos que requieran el diseño de una solución en la que se utilicen los métodos vistos en las clases de teoría. Es imposible seguir las clases de problemas si no se siguen los contenidos de las clases de teoría. El resultado de estas sesiones es alcanzar las competencias necesarias para la resolución de problemas que deberán entregarse según el mecanismo específico para la entrega que se indicará en la página web de la asignatura (espacio Caronte).

MD3 Implementación de proyectos cortos: Realización de prácticas más amplias para profundizar en aspectos aplicados de la teoría. La parte práctica de la asignatura quedará completada con sesiones prácticas, donde los estudiantes deberán resolver problemas concretos de cierta complejidad implementados en python. Estos proyectos se resolverán en pequeños grupos de 2-3 personas, y cada miembro del grupo deberá hacer una parte y ponerla en común con el resto para tener la solución final. Estos grupos de trabajo se deberán mantener hasta el final del curso y deberán autogestionar: reparto de roles, planificación del trabajo, asignación de tareas, gestión de los recursos disponibles, conflictos, etc. Aunque el profesor guiará el proceso de aprendizaje, su intervención en la gestión de los grupos será mínima. Para desarrollar el proyecto, los grupos trabajarán de forma autónoma y las sesiones de prácticas se dedicarán principalmente por el profesor a hacer el seguimiento del estado del proyecto, indicar errores a corregir, proponer mejoras, etc. Las dudas que puedan surgir para la realización de las prácticas se transmitirán a través del foro de Caronte, donde los demás alumnos podrán contestarles.

MD4 Explicación didáctica de un caso práctico: cada estudiante realizará un jupyter notebook donde se explicarán los diversos pasos realizados para la resolución de un problema de Aprendizaje Computacional. Los proyectos serán aplicados a bases de datos escogidas de la plataforma Kaggle (<https://www.kaggle.com/search?q=machine+learning>), y constarán de tres partes: una explicación de los atributos más importantes de la base de datos y del atributo predecir / clasificar; breve descripción del método

de aprendizaje computacional aplicado, junto con los parámetros escogidos; y una presentación de los resultados que se han obtenido. Ejemplos de jupyter notebooks se pueden encontrar en el siguiente repositorio:

<https://datauab.github.io/>

los 5 mejores notebooks se publicarán en este repositorio oficial de laUAB.

MD5 Análisis de un repositorio de código libre en GitHub: en sesiones de seminarios, cada grupo (de 2-3 personas, que puede ser el mismo grupo de las prácticas o no) expondrá durante 10 minutos un proyecto científico de aprendizaje computacional que investigadores de todo el mundo han desarrollado en código libre y compartido en la plataforma GitHub. Los grupos escogerán un proyecto de la siguiente web, pero cualquier propuesta de repositorio justificada será bienvenida:

<https://medium.mybridge.co/amazing-machine-learning-open-source-tools-projects-of-the-year-v-2019-95d772e4>

MD6 Consultas y dudas: Horas de libre disposición por el estudiante para consultas y tutorías sobre aspectos en los que necesite ayuda adicional por parte del profesorado. Todas las consultas se realizarán vía online, mediante el foro de la asignatura, o correos a los profesores, por ejemplo. Se valorará que los estudiantes contesten las dudas de sus compañeros así como que en estas respuestas aporten información que ayude en la comprensión del contenido de las actividades docentes.

MD7 Actividades de evaluación: para cada una de las actividades descritas anteriormente. Ver apartado de evaluación de esta guía docente.

En el caso de los repetidores, si se pide al profesor responsable, se les convalidarán las notas de las actividades docentes que hayan hecho el curso anterior, en caso de que hayan aprobado. Los repetidores si que tendrán que volver a hacer obligatoriamente las pruebas teóricas individuales (MD1).

Competencias Transversales

-T01 Hábitos de pensamiento (T01.02 Desarrollar la capacidad de análisis, síntesis y prospectiva): en las actividades autónomas y supervisadas (estudio de la teoría MD1, realización de las prácticas MD3, realización de los problemas MD2, descripción de un caso práctico MD4 y seminario de un repositorio de código libre MD5)

- T03 Trabajo en equipo (T03.02 Asumir y respetar el rol de los diversos miembros del equipo, así como los diferentes niveles de dependencia del mismo; T03.03 Identificar, gestionar y resolver conflictos): en las prácticas MD3, como actividad autónoma en su preparación y entrega, y como actividad supervisada en su preparación y presentación en un seminario.

- T06 Actitud personal (T06.03 Generar propuestas innovadoras y competitivas en el ámbito profesional): en las actividades autónomas (estudio de la teoría MD1, participación en el foro de la asignatura en Caronte MD6), dirigidas (resolución de proyectos prácticos MD3) y supervisadas (análisis de un caso práctico MD4).

Actividades

Título	Horas	ECTS	Resultados de aprendizaje
Tipo: Dirigidas			
MD1: Contenido teórico	16	0,64	2, 3, 5, 6, 8
MD2: Resolución de problemas	8	0,32	2, 4, 5, 8
MD3: Resolución de proyectos prácticos	12	0,48	1, 2, 3, 4, 5, 6, 7, 8

Tipo: Supervisadas

MD3: Programación de proyectos	22	0,88	1, 2, 3, 4, 5, 7, 8
MD4: Análisis en python de un caso práctico de Aprendizaje Computacional	8	0,32	2, 3, 4, 5, 8
Tipo: Autónomas			
MD1: Estudio individual	10	0,4	2, 3, 4, 5, 6, 8
MD2: Resolución de problemas (individual)	8	0,32	2, 3, 4, 5, 6, 8
MD3: Preparación, programación, documentación y presentación de los proyectos de prácticas	22	0,88	1, 2, 3, 4, 5, 7, 8
MD4: Explicación en python de un caso práctico de aprendizaje computacional en jupyter python	12	0,48	2, 4, 5, 8
MD5: Seminario de un repositorio código libre	12	0,48	1, 2, 3, 4, 5, 6, 8

Evaluación

Actividades e instrumentos de evaluación:

a) Proceso y actividades de evaluación programadas

La asignatura consta de las actividades de evaluación:

- MD1: Exámenes teóricos, donde por cada examen el alumno presencialmente deberá contestar individualmente y por escrito 5 preguntas (para desarrollar en una llanura máximo) sobre conceptos de aprendizaje computacional vistos en las clases de teoría. Representa el 20% sobre la calificación final, es obligatoria, individual y recuperable (habrán dos parciales y sus respectivas recuperaciones). En el caso de que no se pueda hacer exámenes presenciales, por indicación del Equipo de Gobierno de la UAB, se sustituirá el examen por un trabajo de un máximo de 15 páginas explicando el repositorio de código libre que se haya escogido para hacer el seminario.
- MD2: Entrega de informe con problemas resueltos, donde cada alumno individualmente hará entrega de un informe escrito de hasta 4 problemas resueltos vistos en clases de problemas (regresión, backpropagation, memorización, y clustering). Representa el 10% sobre la calificación final, es optativa, individual y recuperable (se puede entregar el día del segundo parcial de la asignatura).
- MD3: Resolución de prácticas con entrega de informe explicando la resolución y los resultados de cada práctica, donde cada grupo compuesto por dos personas entregarán el código python (10% sobre la calificación final) para cada uno de los dos proyectos (regresión y clasificación) aplicadas a 2 bases de datos diferentes, así como un informe de hasta 20 páginas (30% sobre la calificación final) donde describirán cada base de datos, la estrategia que han utilizado para analizar sus datos, así como losproves con diferentes valores de los parámetros quehan probado y los resultados que han obtenido con la mejor configuración possible. Se hará una presentación de cada proyecto vía online. Representa el 40% (código + informe) sobre la calificación final, es obligatoria, grupal y no es recuperable.
- MD4: Elaboración de un jupyter notebook con la descripción en código python de un caso concreto de aprendizaje computacional (ya sea regresión, clasificación, clustering o memorización, por ejemplo escogido aquí: <https://www.kaggle.com/search?q=machine+learning>), y el notebook describirá con código los datos, los modelos utilizados (con los parámetros que funcionan mejor para los datos), y los resultados del problema elegido. Ejemplos de jupyter notebooks aplicados a casos concretos se pueden encontrar aquí <https://datauab.github.io/>. Representa un 20% de la calificación final, es optativa, individual y es recuperable (se puede entregar el día del segundo parcial de teoría).

- MD5: Defensa oral (online) en grupo de 2-3 personas (puede ser el mismo grupo de prácticas o diferente) durante 10 minutos de un repositorio de código libre hecho por un grupo de investigadores o ingenieros donde aplican algún algoritmo de aprendizaje computacional (por ejemplo escogido aquí: <https://medium.mybridge.co/amazing-machine-learning-open-source-tools-projects-of-the-year-v-2019-95d772e4> Representa el 10% sobre la calificación final, es optativa, grupal y no es recuperable.

A continuación se describe como poder aprobar la asignatura con evaluación continua:

- MD1: Exámenes teóricos individuales

La nota final de teoría se calculará a partir de dos exámenes parciales:

$$\text{Nota Teoría} = (0.6 * \text{Parcial1}) + (0.4 * \text{Parcial2})$$

Parcial1 se hace en la mitad del semestre y sirve para eliminar parte de la materia si es aprobada. Parcial2 se hace al final del semestre lectivo y sirve para eliminar la parte del temario que viene después de Parcial1.

Estos exámenes pretenden una evaluación individualizada del estudiante con sus capacidades de contestar a 5 preguntas largas (desarrollar hasta ocupar una página de folio máximo) sobre las técnicas explicadas en clase, así como evaluar el nivel de conceptualización que el estudiante ha hecho de las técnicas vistas.

En el caso de que no se pueda hacer exámenes presenciales, por indicación del Equipo de Gobierno de la UAB, se sustituirá el examen por un trabajo de un máximo de 15 páginas explicando el repositorio de código libre que se haya escogido para hacer el seminario MD5.

Para aprobar la parte de teoría de la asignatura haciendo exámenes presenciales, habrá que cumplir dos requisitos:

- será necesario que las notas de los parciales 1 y 2 sean igual o superior a 4.0 (en ambos parciales). En caso de que se quite menos de un 4.0 en alguno de los dos Parciales, se deberá volver a hacer el parcial que corresponda durante el examen de recuperación.
- la nota final de teoría debe ser mayor o igual que 4.0. En caso de que la nota de teoría final no sea igual o superior a 4.0, los estudiantes se pueden presentar al examen de recuperación para ser evaluados de todos los contenidos vistos en la asignatura.

Examen de recuperación (finales de enero o principios de febrero). En este examen se puede recuperar el (los) parcial (es) que no haya (n) superado el 4.0, o recuperar todo el temario en el caso de que la nota final de teoría no supere el 4.0.

- MD2: Entrega individual de un informe con problemas resueltos

Los problemas tienen como objetivo provocar que el estudiante entre con los contenidos de la asignatura de manera continuada y, a partir de pequeños problemas, que se familiarice directamente en la aplicación de la teoría. Como evidencia de este trabajo se pide la presentación obligatoria de un portafolio en el que habrá ido guardando los problemas que habrá ido realizando (competencia T06).

Nota Problemas = Evaluación del portafolio con 4 problemas resueltos (según el calendario indicado a Caronte).

Hay que entregar un mínimo de 2 problemas para aprobar esta parte. Habrá recuperación de los problemas (entregando los problemas no entregados durante el curso el día del segundo parcial de teoría).

- MD3: Resolución de prácticas en grupo

La evaluación de cada uno de los 2 proyectos de prácticas incluirá:

- Evaluación conjunta de cada proyecto (competencia T03): nota única para todos los miembros del grupo de trabajo que valorará el resultado global del proyecto, la calidad del código, la estructura general de la presentación final y los documentos entregados a lo largo del proyecto.

- Evaluación individual (competencia T01): se valorará el trabajo individual a partir de las respuestas a las preguntas en las sesiones de control online, de la presentación final del proyecto online y principalmente de la participación activa en los foros de Caronte. En los casos requeridos por cualquier grupo (en casos de incidencias entre compañeros), se evaluará un breve formulario confidencial calificando la contribución de cada compañero de grupo al resultado final.

La nota del proyecto se calculará según la fórmula:

$$\text{Nota Prácticas} = (0.5 * \text{Nota Proyecto 1 Regresión}) + (0.5 * \text{Nota Proyecto 2 Clasificación})$$

$$\text{Nota Proyectos 1 y 2} = (0.9 * \text{Nota Grupo}) + (0.1 * \text{Nota Individual})$$

$$\text{Nota Grupo} = (0.3 * \text{Programa}) + (0.1 * \text{Presentación}) + (0.6 * \text{Documentación})$$

No hay recuperación de las prácticas: en caso de no presentar una entrega o considerarla copiada, si la Nota Proyecto final no supera el 5.0, se considera la asignatura suspendida.

En casos muy justificados (p.ej. por cuestiones laborales, familiares o de salud, ...), en vez de realizar estos 2 proyectos, el alumno podrá realizar el llamado itinerario Cousera: previo visto bueno del profesor, el alumno que lo solicite y lo justifique podrá entregar las prácticas que se piden en el curso online de aprendizaje computacional de la plataforma educativa Coursera (<https://es.coursera.org/learn/machine-learning>). En este caso, la máxima nota de prácticas que el alumno podrá alcanzar es de 7 en vez de 10 (debido a que no hay informe ni presentación, sólo se entrega código en este itinerario).

- MD4: Realización de un jupyter notebook en python describiendo un caso concreto de aprendizaje computacional

La evaluación se basará en el código python y la explicación del código que se encontrará en el jupyter notebook que se entregará a más tardar el día del segundo parcial de la asignatura. La nota del notebook se calculará según la fórmula:

$$\begin{aligned} \text{Nota Notebook} = & (0.1 * \text{Introducción a la base de datos}) + (0.3 * \text{Análisis de los atributos, correlaciones, ...}) + \\ & (0.3 * \text{Descripción del método utilizado, cómo encontrar los mejores parámetros, comparativa de métodos ...}) + \\ & (0.3 * \text{Descripción de los resultados, matrices de confusión, gráficas de los modelos y los datos, ejemplos de falsos positivos / negativos, curvas ROC, ...}) \end{aligned}$$

Ejemplos de jupyter notebooks aplicados a casos concretos se pueden encontrar aquí <https://datauab.github.io/>.

- MD5: Defensa oral en grupo de un trabajo científico

La defensa oral o seminario online tiene como objetivo provocar que el grupo vea y entienda como otro grupo de ingenieros o científicos han abordado un problema de aprendizaje computacional, dado que existe Código fuente compartido en la plataforma GitHub. Durante 10 minutos, el grupo explicará el algoritmo que se ha utilizado, la estrategia sobre cómo se ha abordado el análisis de datos, y los resultados. Como evidencia de este trabajo se pide la presentación no obligatoria del proyecto escogido (competencia T06).

Nota Seminario = Presentación online de un proyecto elegido (de la web <https://medium.mybridge.co/amazing-machine-learning-open-source-tools-projects-of-the-year-v-2019-95d772e4>).

No hay recuperación de la defensa oral, ni hay una nota mínima en esta actividad para aprobar la asignatura.

- Evaluación de competencias transversales

Los exámenes parciales permitirán evaluar su adquisición de hábitos de pensamiento y de trabajo personal (T01 Hábitos de pensamiento, Nota Teoría). Con Nota Proyecto de prácticas, también se evaluará el trabajo en equipo (T03 Trabajo en equipo, Nota Grupo) y la presentación individual (T01 Hábitos de pensamiento, Nota Individual). Con la realización de los problemas y la realización de un seminario sobre un proyecto científico, se evaluará la adquisición de hábitos para solucionar una tarea predeterminada con unos valores de datos totalmente diferentes a los vistos en clase (T06 Actitud personal, Nota Problemas y Seminario).

La nota final de la asignatura se obtiene combinando la evaluación de estas 4 actividades de la siguiente manera:

$$\text{Nota Final} = (0.2 * \text{Teoría}) + (0.4 * \text{Proyecto}) + (0.1 * \text{Problemas}) + (0.2 * \text{Notebook}) + (0.1 * \text{Seminario})$$

Condiciones para aprobar:

Para aprobar es necesario que la evaluación de cada una de las dos actividades obligatorias (MD1 y MD3) supere el mínimo exigido y que la evaluación total supere los 5 puntos. En caso de no superarla, la nota numérica del expediente será el valor menor entre 4.5 y la media ponderada de las notas:

- La nota final de teoría MD1 debe ser mayor o igual que 4.0 para poder aprobar la parte de teoría.
- La nota del proyecto MD3 debe ser mayor o igual que 5.0 para poder aprobar la parte de prácticas.
- La nota final de la asignatura debe ser mayor o igual que 5.0 para poder aprobar la asignatura.

En el caso de no llegar al mínimo exigido en alguna de las actividades obligatorias de evaluación (MD1 y MD3), si el cálculo de la nota final de la asignatura fuera igual o superior a 5, se pondrá un 4,5 de nota final de la asignatura al expediente.

En caso de no superar la asignatura debido a que alguna de las actividades obligatorias de evaluación no alcanza la nota final mínima requerida (5.0), la nota numérica del expediente será el valor menor entre 4.5 y la media ponderada de las notas.

Si el alumno se presenta a algún examen o si entrega alguna práctica, ya no podrá ser evaluado como "No Evaluable" en caso de que no se presente a ninguna de las otras evaluaciones, sino que se le calculará la nota final a partir de aquellas evaluaciones continuadas a las que se haya presentado.

b) Programación de actividades de evaluación

Las fechas de evaluación continua y entrega de trabajos se publicarán en Caronte (<http://caronte.uab.cat/>), en el espacio de esta asignatura y pueden estar sujetos a cambios de programación por motivos de adaptación a posibles incidencias; siempre se informará a caronte.uab.cat sobre estos cambios ya que esta plataforma se convertirá en el mecanismo habitual de intercambio de información entre profesor y estudiantes.

Se prevé la siguiente calendarización (semana 1 corresponde a la semana del 14 de Septiembre 2020):

- Exámenes teóricos individuales: semanas 8 y 17-19 de la asignatura. En el caso de que no se pueda hacer exámenes presenciales, por indicación del Equipo de Gobierno de la UAB, se sustituirá el examen por un trabajo de un máximo de 15 páginas explicando el repositorio de código libre que se haya escogido para hacer el seminario, y que se entregará durante la semana 15 de la asignatura.
- Entrega individual de un informe con los problemas resueltos: entrega semanas 6, 9, 12 y 14 de la asignatura (todos recuperables el día del segundo parcial).
- Resolución en grupo de prácticas: control semanas 5 y 9, evaluación semanas 7 y 11 de la asignatura.
- Desarrollo notebook python: control semana 12 y evaluación semana 13 de la asignatura.
- Seminario en grupo de un repositorio científico: evaluación semanas 14 y 15 de la asignatura.

c) Proceso de recuperación

El estudiante puede presentarse a la recuperación siempre que se haya presentado a un conjunto de actividades que representen un mínimo de dos terceras partes (6 actividades de evaluación de 10 total: 2 exámenes parciales; 2 evaluaciones de prácticas; 4 entregas de problemas; 1 entrega de un notebook en python; y 1 seminario) de la calificación total de la asignatura.

De estos, se podrán presentar en la recuperación aquellos estudiantes que tengan como media de todas las actividades de la asignatura una calificación superior a 3.0.

Hay que tener presente que la Resolución de Prácticas (MD3) y la Defensa Oral en grupo de un repositorio (MD5) no son recuperables.

d) Procedimiento de revisión de las calificaciones

Para cada examen teórico individual, se indicará un lugar, fecha y hora de revisión en la que el estudiante podrá revisar la actividad con el profesor. En este contexto, se podrán hacer reclamaciones sobre la nota de la actividad, que serán evaluadas por el profesorado responsable de la asignatura. Si el estudiante no se presenta en esta revisión, no se revisará posteriormente esta actividad.

e) Calificaciones

Matrículas de Honor: Se concederán Matrículas de Honor a decisión del profesorado responsable de la asignatura, hasta llegar al cinco por ciento o fracción de los alumnos matriculados en todos los grupos de docencia de la asignatura. La normativa de la UAB indica que las MH sólo se podrán otorgar a estudiantes que hayan obtenido una calificación final igual o superior a 9.00.

No evaluable: Un estudiante se considerará no evaluable (NA) si no se ha presentado a ninguno de los exámenes parciales y en ninguna de las 2 evaluaciones de las prácticas.

f) Irregularidades perpart del estudiante, copia y plagio

Sin perjuicio de otras medidas disciplinarias que se estimen oportunas, se calificarán con un cero lesirregularitats cometidos por el estudiante que puedan conducir a una variación de la calificación de un acto de evaluación. Por tanto, la copia, el plagio, el engaño, dejar copiar, etc. en cualquiera de las actividades de evaluación implicará suspender con un cero. Las actividades de evaluación 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 recuperarlo en el mismo curso. Estas irregularidades incluyen, entre otros:

- la copia total o parcial de una práctica, informe, o cualquier otra actividad de evaluación;
- dejar copiar;
- presentar un trabajo de grupo no hecho íntegramente por los miembros del grupo (aplicado a todos los miembros, no sólo a los que no han trabajado);
- presentar como propios materiales elaborados por un tercero, encara que sean traducciones o adaptaciones, y en general trabajos con elementos no originales y exclusivos del estudiante;
- tener dispositivos de comunicación (como teléfonos móviles, smart watches, bolígrafos con cámara, etc.) accesibles durante los provees d'avaluació teórico-prácticas individuales (exámenes);
- hablar con compañeros durante las pruebas de evaluación teórico-prácticas individuales (exámenes);
- copiar o intentar copiar de otros alumnos durante las pruebas de evaluación teórico-prácticas (exámenes);
- usar o intentar usar escritos relacionados con la materia durante la realización de las pruebas de evaluación teórico-prácticas (exámenes), cuando éstos no hayan sido explícitamente permitidos.

La nota numérica del expediente será el valor menor entre 3.0 y la media ponderada de las notas en caso de que el estudiante haya cometido irregularidades en un acto de evaluación (y por tanto será posible el aprobado por compensación). En ediciones futuras de esta asignatura, el estudiante que haya cometido irregularidades en un acto de evaluación no se le convalidará ninguna de las actividades de evaluación realizadas.

En resumen: copiar, dejar copiar o plagiar (o el intento de) en cualquiera de las actividades de evaluación equivale a un SUSPENSO, no compensable ni recuperable y sin convalidaciones de partes de la asignatura en cursos posteriores.

g) Evaluación de los estudiantes repetidores

A partir de la segunda matrícula, la evaluación de la asignatura consistirá en el examen teórico individual, más la nota correspondiente a las prácticas obtenida la primera vez que el estudiante ha matriculado de la asignatura, siempre que las notas de prácticas sean superiores o iguales a 5.0.

Para poder optar a esta evaluación diferenciada, el estudiante repetidor debe pedir al profesor como muy tarde hasta la semana 4.

Actividades de evaluación

Título	Peso	Horas	ECTS	Resultados de aprendizaje
Defensa oral de un repositorio de código libre	10%	2	0,08	1, 2, 3, 4, 5, 8
Documentación escrita, presentación, seguimiento proyectos de prácticas	30%	6	0,24	1, 2, 3, 4, 5, 7, 8
Entrega de problemas	10%	2	0,08	2, 3, 4, 5, 6, 8
Implementación proyectos de prácticas	10%	2	0,08	1, 2, 3, 4, 5, 7, 8
Jupyter notebook en python de un caso práctico de aprendizaje computacional	20%	4	0,16	2, 4, 5, 8
Pruebas teóricas individuales	20%	4	0,16	2, 3, 4, 5, 6, 8

Bibliografía

Enlaces web

- Caronte: <http://caronte.uab.cat>
- Artificial Intelligence: A Modern Approach. <http://aima.cs.berkeley.edu/>

Bibliografía básica

- S. Russell, P. Norvig. Artificial Intelligence: A Modern Approach. Ed. Prentice Hall, Second Edition, 2003. (Existeix traducció al castellà: Inteligencia artificial: Un Enfoque Moderno)

Bibliografía complementaria

- L. Igual, S. Seguí. Introduction to Data Science. Ed. Springer, 2017
- Bishop, Pattern Recognition and Machine Learning, 2007.
- Duda, Hart, and Stork, Pattern Classification, 2nd Ed., 2002.
- Marlsand, Machine Learning: an Algorithmic Perspective, 2009
- Mitchell, Machine Learning, 1997
- Ripley, Pattern Recognition and Neural Networks, 1996.

Bibliografía relacionada

- Eberhart, Shi, Computational Intelligence: Concepts to Implementations, 2007
- Friedman, Tibshirani, The Elements of Statistical Learning, 2009.
- Gilder, Kurzweil, Richards, Are we spiritual machines? Ray Kurzweil vs. the Critics of Strong AI, 2011
- Kurzweil, The Singularity is Near: When Humans transcend Biology, 2006
- Rosen, Life Itself: A Comprehensive Inquiry into the Nature, Origin, and Fabrication of Life (Complexity in Ecological Systems), 2005

- Witten, Frank, Hall, Data Mining: Practical Machine Learning Tools and Techniques, 2011