Universitat Autònoma de Barcelona

2023/2024

Aprendizaje Computacional

Código: 104403
Créditos ECTS: 6

Titulación	Tipo	Curso	Semestre
2503740 Matemática Computacional y Analítica de Datos	OB	3	1

Contacto

Nombre: Jordi Gonzalez Sabate

Correo electrónico: jordi.gonzalez@uab.cat

Idiomas de los grupos

Puede consutarlo a través de este enlace. Para consultar el idioma necesitará introducir el CÓDIGO de la asignatura. Tenga en cuenta que la información es provisional hasta el 30 de noviembre del 2023.

Equipo docente

Marc Borras Camarasa

Prerrequisitos

Se considera muy importante un buen conocimiento de los contenidos de las asignaturas cursadas durante el primer curso como Probabilidad y Cálculo y de segundo curso como Inteligencia Artificial y Modelización e Inferencia. Aunque alguna parte teórica de los contenidos ya haya sido tratada en aquellas asignaturas desde un punto de vista matemático, en esta asignatura nos centraremos en su implementación en python y su aplicación en varios casos reales.

Objetivos y contextualización

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 masivas. 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

- Evaluar de manera crítica y con criterios de calidad el trabajo realizado.
- Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio.
- Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.
- Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.

- Resolver problemas relacionados con el análisis de grandes volúmenes de datos a través del diseño de sistemas inteligentes y de aprendizaje computacional.
- Trabajar cooperativamente en un contexto multidisciplinar asumiendo y respetando el rol de los diferentes miembros del equipo.
- Utilizar eficazmente bibliografía y recursos electrónicos para obtener información.

Resultados de aprendizaje

1. Conocer y aplicar las técnicas de aprendizaje más adecuadas para resolver problemas computacionales en diferentes casos de estudio.
2. Entender y evaluar los resultados y limitaciones de las técnicas de aprendizaje más comunes.
3. Evaluar de manera crítica y con criterios de calidad el trabajo realizado.
4. Identificar y definir soluciones computacionales en múltiples dominios para la toma de decisiones basada en la exploración de alternativas, el razonamiento incierto y la planificación de tareas.
5. Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio.
6. Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.
7. Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.
8. Trabajar cooperativamente en un contexto multidisciplinar asumiendo y respetando el rol de los diferentes miembros del equipo.
9. Utilizar eficazmente bibliografía y recursos electrónicos para obtener información.

Contenido

TEMA 1: INTRODUCCIÓN

1.1 Conceptos básicos

1.2 Historia y sesgos del aprendizaje computacional

TEMA 2: REGRESIÓN DE DATOS

2.1 Descenso del gradiente

2.2 Regularización

TEMA 3: CLASIFICACIÓN DE DATOS

3.1 Regresión logística regularizada

3.2 Máquinas de vectores de apoyo

TEMA 4: MEMORIZACIÓN DE DATOS

4.1 Aprendizaje perezoso: los hiperparámetros

4.2 El algoritmo K-Nearest neighbors (ponderado)

TEMA 5: AGRUPACIÓN DE DATOS

5.1 Modelización por Mixtura de Gaussianas

5.2 El algoritmo 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/view.php?id=94), en el menú de la asignatura Aprendizaje Computacional (104403). Servirá para poder ver los materiales, gestionar los grupos de prácticas, realizar las entregas correspondientes, ver las notas, comunicarse con los profesores, etc. Para poder utilizarlo hay que dar los siguientes pasos:

- Darse de alta como usuario dando el nombre, NIU, y una foto carné en formato JPG. Si ya se ha dado de alta por alguna otra asignatura, no es necesario volver a hacerlo, puede ir al siguiente paso.
- Inscribirse en el tipo de docencia "Aprendizaje Computacional (104403)", dando como código de asignatura "apc2023" (sin las comillas).

En el desarrollo de la asignatura se podrán diferenciar siete tipos de actividades docentes:

MD0 Exposición de contenidos de teoría: Presentación de los contenidos teóricos a trabajar en la asignatura. Estos contenidos tendrán que haberse 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 modo que serán la base sobre la que se desarrollarán otras actividades del curso. Los contenidos se encontrarán en la página de Caronte (presentaciones y videos) y constarán de dos partes: una presentación en la que 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 ejemplifican los detalles de codificación y de librerías para implementar en un caso práctico los principales conceptos vistos en la hora anterior . El alumnado podrá ver los vídeos de las clases, 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 distintos rendimientos y precisiones que se alcanzan en una base de datos específica con configuraciones concretas de los algoritmos explicados en la asignatura.

MD1 Resolución de problemas de computación: Entrega de hasta un máximo de 3 problemas implementados en un Jupyter Notebook, de los 5 trabajados en clase. Todos los temas de teoría irán acompañados de una relación de notebooks, a partir de los cuales el estudiante tendrá que trabajar en sesiones de problemas y entregar opcionalmente. 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 se entregarán según el mecanismo específico para la entrega que se indicará en la página web de la asignatura (espacio Caronte).

MD2 Implementación de proyecto corto guiado grupal: Realización de 1 práctica guiada para profundizar en aspectos aplicados de la teoría. La parte práctica de la asignatura quedará completada con sesiones prácticas, donde las y los estudiantes tendrán que resolver problemas concretos de una cierta complejidad implementados en python. Estos proyectos se resolverán en pequeños grupos de 2-3 personas (en casos justificados, se permitirán grupos de 1 persona), donde cada miembro del grupo tendrá que hacer una parte y ponerla en común con el resto para tener la solución final. Estos grupos de trabajo se tendrán que mantener hasta mitad del curso y tendrán que autogestionarse: 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 deben dedicarse principalmente por el profesor a realizar 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.

MD3 Resolución individual de un caso práctico Kaggle: cada estudiante realizará individualmente un jupyter notebook donde se explicarán los distintos 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 a 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/

MD4 Consultas y dudas: Horas de libre disposición para 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 a las dudas de sus compañeros así como que en estas respuestas aporten información que ayude en la comprensión del contenido de las actividades docentes.

MD5 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 realizado el curso anterior, en caso de que hayan aprobado. Los repetidores sí que tendrán que volver a hacer obligatoriamente las pruebas teóricas individuales (MD0).

Competencias Transversales

Las competencias transversales T01, T02 y T04 se trabajan y evaluaron en el hogar del curso en las siguientes actividades

- T01 Evaluar de forma crítica y con criterios calidad el trabajo realizado: en las actividades MD0 (estudio de la teoría), MD1 (realización de los problemas), MD2 (realización de las prácticas de laboratorio) y MD3 (explicar un caso práctico de aprendizaje computacional)

- T02 Trabajar cooperativamente en un contexto multidisciplinar asumiendo y respetando el rol de los distintos miembros del equipo: en la práctica de MD2 y el análisis de proyectos en MD3.

- T04 Utilizar eficazmente la bibliografía y los recursos electrónicos para obtener información: en la preparación del material de teoría de MD0, y en la preparación de la descripción del caso práctico en MD3

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			
MD0: Contenido teórico	12	0,48	3, 1, 2, 5, 9
MD1: Resolución de problemas	10	0,4	3, 1, 4, 5, 7

MD2: Resolución de proyectos prácticos	6	0,24	3, 1, 2, 4, 5, 7, 8
MD3: Caso Kaggle	12	0,48	3, 2, 4, 5, 6, 9
Tipo: Supervisadas			
MD2: Programación de proyectos	6	0,24	3, 1, 2, 4, 5, 6, 8
MD3: Caso Kaggle	12	0,48	4, 5, 6, 7, 8, 9
Tipo: Autónomas			
MD0: Estudio individual	12	0,48	3, 2, 4, 5, 6, 7, 9
MD1: Resolución de problemas (individual)	24	0,96	3, 1, 2, 4, 5, 6
MD2: Resolución de casos prácticos (en grupo)	16	0,64	3, 2, 4, 5, 6, 8, 9
MD3: Explicación en python de un caso práctico de aprendizaje computacional	32	1,28	3, 1, 2, 5, 6, 7, 9

Evaluación

Actividades e instrumentos de evaluación:

a) Proceso y actividades de evaluación programadas

La asignatura consta de las siguientes actividades de evaluación:

- MD0: Exámenes teóricos, donde por cada examen el alumno presencialmente deberá contestar individualmente y por escrito 5 preguntas (para desarrollar en una página máxima) sobre conceptos de aprendizaje computacional vistos en las clases de teoría. Representa el 20% sobre la calificación final, es opcional, individual y recuperable (habrá dos parciales y sus respectivas recuperaciones, sin penalización).

- MD1: Entrega de informe con hasta 3 problemas resueltos, donde cada alumno individualmente hará entrega de un informe escrito de hasta 3 de los 5 problemas vistos en clases de problemas (regresión, clasificación, memorización y clustering). Representa el 25% sobre la calificación final, es opcional, individual y recuperable (sin penalización. el dia del examen de teoria de recuperación).

- MD2: Resolución de una práctica con entrega de informe explicando la resolución y los resultados, donde cada grupo compuesto por dos o tres personas (en casos justificados los grupos podrán ser de 1 persona), entregarán el código python, así como un informe de hasta 30 páginas donde describirán las bases de datos, la estrategia que han utilizado para analizar los datos, así como las pruebas con distintos valores de los parámetros que han probado y los resultados que han obtenido con la mejor configuración posible. Se realizará una presentación opcional de cada proyecto. Representa el 25% (código 5% + informe 15% + presentación 5%) sobre la calificación final, es opcional, grupal y es recuperable (sin penalización. el dia del examen de teoria de recuperación).

- MD3: Elaboración individual de proyecto python con la descripción en un informe de hasta 30 páginas del código y los resultados de un caso concreto de aprendizaje computacional (ya sea regresión, clasificación, clustering o memorización, asignado por el profesor de aquí: https://www.kaggle.com /search?q=machine+learning), y realización de una presentación opcional donde se describirá con código los datos, los modelos utilizados (con los parámetros que mejor funcionan para los datos), y los resultados del problema escogido. Ejemplos de jupyter notebooks aplicados a casos concretos se pueden encontrar aquí https://datauab.github.io/. Representa un 30% de la calificación final (código 5% + informe 15% + github 5% + presentación 5%), es opcional, individual y NO es recuperable.

A continuación se describe cómo poder aprobar la asignatura con evaluación continuada:

– MD0: Exámenes teóricos individuales

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

Nota Teoría = (0.5 * Parcial1) + (0.5 * Parcial2)

Parcial1 se realiza a mitad de semestre y sirve para eliminar parte de la materia si es aprobada. Parcial2 se realiza 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 técnicas vistas.

Para aprobar la parte de teoría de la asignatura haciendo exámenes presenciales, será necesario 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 saque menos de un 4.0 en alguno de los dos Parciales, deberá volver a realizarse 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 pueden presentarse 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 caso de que la nota final de teoría no supere el 4.0.

– MD1: Entrega individual de un informe con hasta 3 problemas resueltos

Los problemas tienen como objetivo provocar que el estudiante entre con los contenidos de la asignatura de forma 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 portfolio en el que habrá ido guardando los problemas que habrá ido realizando (competencia T04).

Nota Problemas = Evaluación del portfolio con 3 problemas resueltos de los 5 vistos en clase. Pueden recuperarse entregando por Caronte el día del segundo parcial de teoría, sin penalización.

– MD2: Resolución de 1 práctica en grupo

La evaluación del 2 proyectode prácticas incluirá:

– Evaluación conjunta de cada proyecto (competencia T02): 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:

Nota Prácticas = (0.2 * Programa) + (0.2 * Presentación) + (0.6 * Documentación)

En casos muy justificados (p.ej. PIUNE, por cuestiones laborales, familiares o de salud,...), los grupos pueden ser de 1 persona.

– MD3:Realización de un caso concreto de aprendizaje computacional de la plataforma Kaggle

La evaluación se basará en el código python y la explicación en un informe de hasta 30 páginas del código y los resultados que se encontrará en un repositorio github. La nota del caso kaggle se calculará según la fórmula:

Nota Caso Kaggle = (0.1 * Introducción a la base de datos) + (0.25 * Análisis de los atributos, correlaciones,...) + (0.25 * Descipción del método utilizado, como encontrar los mejores parámetros, comparativa de métodos...) + (0.3 * Descipción de los resultados, matrices de confusión, gráficas de los modelos y datos, ejemplos de falsos positivos/negativos, curvas ROC, ...) +(0.1 * Presentación del repositorio Github)

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

– Evaluación de competencias transversales

Los exámenes parciales permitirán evaluar su adquisición de hábitos de pensamiento y de trabajo personal (T01 Evaluar de forma crítica y con criterios calidad el trabajo realizado, Nota Teoría). Con Nota Proyecto de prácticas, también se evaluará el trabajo en equipo (T02 Trabajar cooperativamente en un contexto multidisciplinar asume respetando el rol de las diferentes miembros del equipo, Nota Grup). Con la realización de los problemas y la realización de un caso Kaggle, se evaluará la adquisición de hábitos para solucionar una tarea predeterminada con unos valores de datos totalmente diferentes a los vistos en clase (T04 Utilizar eficazmente la bibliografía y los recursos electrónicos para obtener información, Nota Problemas y Caso Kaggle).

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

Nota Final = (0.20 * Teoría) + (0.25 * Práctica) + (0.25 * Problemas) + (0.30 * Caso Kaggle)

Adicionalmente, se organizará una Datathon para toda la clase, donde se podrá alcanzar hasta 1.0 puntos extra para quedar entre los primeros en la competición que se realizará el último día del curso, participación opcional.

Condiciones para aprobar:

Todas las actividades evaluativas son opcionales, para aprobar la asignatura es necesario que la suma de la evaluación de cada una de las actividades supere los 5 puntos. En caso de no superar la asignatura, la nota numérica del expediente será el valor menor entre 4.5 y lamedia ponderada de las notas alcanzadas:

La nota final de teoría MD0 debe ser mayor o igual que 4.0 para poder sumar la parte de teoría a la nota final de la asignatura.
La nota del proyecto MD2 debe ser mayor o igual que 5.0 para poder sumar la parte de la práctica a la nota final de la asignatura.
La nota final de la asignatura debe ser mayor o igual que 5.0 para poder aprobar la asignatura.

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 en las que se haya presentado.

b) Programación de actividades de evaluación

Las fechas de evaluación continua y entrega de trabajos se publicarán en el 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.

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 (4 actividades de evaluación sobre 7 total: 2 exámenes de teoría parciales; 1 evaluación de la práctica; 3 entregas de problemas, y 1 entrega del caso kaggle) de la calificación total de la asignatura.

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

Hay que tener en cuenta que la Resolución del Caso Kaggle (MD3) no es recuperable.

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 realizar reclamaciones sobre la nota de la actividad, que serán evaluadas por el profesorado responsable de la asignatura. Si el estudiante no se presenta a 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 podrán otorgarse 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 en ninguno de los exámenes parciales ni en ninguna de las 2 evaluaciones de las prácticas.

f) Irregularidades por parte del estudiante, copia y plagio

Sin perjuicio de otras medidas disciplinarias que se estimen oportunas, se calificarán con un cero las irregularidades cometidas 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 deevaluación supondrá suspenderla 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 recuperarla en el 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 de grupo no realizado 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, aunque 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 las pruebas de evaluación 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 casode que el estudiante haya cometido irregularidades en un acto de evaluación (y por tanto no será posible el aprobado por compensación). En ediciones futuras de esta asignatura, al estudiante que haya cometido irregularidades en un acto de evaluación no se le convalidará ninguna de las actividades de evaluación realizadas.

Enresum: 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, añadiendo las notas correspondientes a las MD obtenidas la primera vez que el estudiante se ha matriculado de la asignatura, siempre que las notas de prácticas MD2 sean superiores o iguales a 5.0.

Para poder optar a esta evaluación diferenciada, el estudiante repetidor debe pedirlo al profesor a más tardar hasta la semana 5.

Actividades de evaluación continuada

Título	Peso	Horas	ECTS	Resultados de aprendizaje
Documentación escrita, implementación y presentación de la práctica	25%	2	0,08	3, 2, 4, 5, 7, 8
Documentación escrita, implementación y presentación del caso Kaggle	30%	2	0,08	3, 1, 5, 6, 8, 9
Entrega de problemas	25%	0	0	3, 2, 4, 5, 7, 9
Pruebas teóricas inviduales	20%	4	0,16	2, 4, 5, 6, 7, 9

Bibliografía

Enlaces web

- Caronte: http://caronte.uab.cat

- Artificial Intelligence: A Modern Approach. http://aima.cs.berkeley.edu/

- Web del Catálogo de las Bibliotecas de la UAB: https://bit.ly/35jalzm

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 trascend 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

Software

El software necesario será el lenguaje de programación Python, un entorno de programación (como Spyder, Pycharm o Visual Studio Code), la aplicación web Jupyter Notebook, y las librerías necesarias para análisis de datos: scipy (contiene NumPy, matplotlib, pandas), sklearn y Seaborn.