PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2433 — Minería de Datos

Programa de curso

Actualización: 27 de agosto de 2020

Profesor:	Vicente Domínguez (vidominguez@ing.puc.cl)
Ayudantes:	Hernán Valdivieso (hfvaldivieso@uc.cl), Daniela Flores (diflores@uc.cl), Astrid San Martín (aesanmar@uc.cl), Javier Ruiz (jiruiz2@uc.cl), Vicente Castro (vvcastro@uc.cl), Federico Taladriz (fntaladriz@uc.cl), Sebastian Ricke (sricke@uc.cl), Ricardo Schilling (reschilling@uc.cl)
Clases:	Martes, módulos 4 y 5
Ayudantías:	Jueves, módulo 4
Requisitos:	IIC1103 – Introducción a la Programación, MAT1203 – Álgebra Lineal y EYP1113 – Probabilidad y Estadística
Sitio web:	https://github.com/IIC2433/Syllabus-2020-2

Descripción

El desarrollo de la tecnología ha hecho que la mayoría de los datos almacenados de forma física ahora lo estén de forma digital. Esto ha permitido que mediante algoritmos computacionales podamos extraer información de estos datos, ya sea patrones, modelos de predicción o identificar anomalías. Minamos estos datos para obtener conocimiento. En este curso se espera enseñar todo el proceso para poder minar conjuntos de datos, también conocido como Descubrimiento de Conocimiento en Base de Datos (Knowledge Discovery in Databases o KDD).

Objetivo General

El objetivo de este curso es proporcionar al alumno elementos que le permitan entender las principales teorías y prácticas de la emergente área de Minería de Datos. Al final del curso, el alumno debera tener un conocimiento teorico y práctico de las principales técnicas utilizadas actualmente en la creación de programas capaces de extraer conocimiento relevante y patrones desde distintas fuentes y bases de datos. Ademas, el alumno conocera algunas de las principales aplicaciones donde en la actualidad este tipo de técnicas están teniendo una amplia aceptacion, comprendiendo sus potencialidades y limitaciones.

Contenidos

- **Introducción:** el concepto y proceso de minería de datos, tipos de problemas relevantes.
- ***Data Warehouse* y OLAP:** arquitecturas, implementaciones, aplicaciones en minería de datos
- ***Web scrapping*:** obtención de *datasets* de páginas web.
- **Preparación de la información:** datos ruidosos, datos faltantes, reducción de la dimensionalidad y transformaciones, integración e inconsistencias.
- **Reglas de asociación:** algoritmo Apriori, FP-growth.
- **Reducción de dimensionalidad:** análisis de las componentes principales (PCA).
- **Regresión:** regresión lineal con funciones de base polinomial y exponencial.

- **Clasificación:** regresión logística, árboles de decisión y *random forest*, razonamiento en base a casos: KNN, KD-Trees. Naïve Bayes.
- **Métodos de evaluación de clasificadores:** *hold out*, *cross validation*, *bootstrapping*, *confusion matrix*, *recall*, *precision*, *F1-score*.
- ***Clustering*:** *clustering* particional: K-Means, Mean Shift, EM-GMM. Medidas de similaridad, *clustering* aglomerativo, *clustering* en subespacios. Evaluación de *clustering*.
- **Aplicaciones:** visualizacion, detección de anomalías.
- **Aplicaciones avanzadas:** *recommender systems*, *deep learning*, *reinforcement learning*.

Metodología

En general las clases seran los días martes en los modulos 4 y 5, para poder realizar actividades prácticas luego de la clase, utilizando los dos módulos. Debido a la contingencia actual, todas las clases se harán mediante la plataforma Zoom a modo de video conferencia.

Los dias de actividad, los alumnos resolverán un ejercicio de programación donde se apliquen los nuevos conocimientos. Tanto el profesor como ayudantes del curso estarán presentes durante toda la duración de esta actividad para resolver dudas y guiar la actividad. Esto se realizará mediante la plataforma Discord.

Sobre las ayudantías, no habrán todas las semanas. Se realizarán ayudantías los días jueves modulo 4, estas complementarán con práctica la teoría estudiada en clases, donde se podran realizar evaluaciones. También, algunos días jueves se utlizarán para rendir los controles del curso.

Durante todo el curso los alumnos deberán implementar soluciones en Python 3.8, se recomienda estar preparados con este lenguaje de programacion antes de comenzar las clases.

Evaluación

La evaluación será efectuada mediante controles, tareas individuales de programación, y un proyecto grupal. También habrán actividades, pero estan tendrán un caracter formativo.

Controles. El objetivo de los controles es evaluar el aprendizaje teórico de los contenidos del curso. Estos serán publicados los jueves. El módulo de clases se dará para resolver dudas, y se dará plazo hasta las 23:59 del mismo día para la entrega del control, a través de un buzón por la plataforma Canvas. La nota de los controles se especifica como **C**, siendo el promedio ponderado de los seis controles del curso. **Ningún control se eliminará del cálculo de la nota.**

Actividades formativas. Su objetivo es conseguir que el estudiante practique el contenido de la semana, resuelva sus dudas de forma guiada por el profesor y ayudantes, y reflexione sobre su aprendizaje. Cada entrega será revisada superficialmente, se le asignará un nivel de cumplimiento entre: no logrado (0), medianamente logrado ($1/2$) y logrado (1). El nivel de cumplimiento acumulado de actividades formativas se considerará parcialmente como parte de una nota: **AF**. Esta nota se calcula proporcionalmente a la suma de nivel de cumplimiento acumulado Σ de las siete actividades, pero solo se necesita un total de 6 para alcanzar la nota máxima. Entonces, su cálculo es: $\mathbf{AF} = 1 + 6 \times \text{mín}\{\Sigma, 6\}/6$. Ausentarse a una actividad formativa no es justificable e implica que no se revisará y será catalogado como no logrado.

Tareas. Se publicarán tres tareas de programación las que deberán ser resueltas **individualmente** por cada alumno. Cada tarea tendrá un plazo de realización de al menos dos semanas. La nota de cada tarea se especifica como $\mathbf{T_1}$, $\mathbf{T_2}$ y $\mathbf{T_3}$. La nota **T** de tareas corresponderá al promedio aritmético de las tres tareas, a excepción que

la nota **AF** sea mayor a este promedio. En cuyo caso, el cálculo de la nota **T** se calcula como el promedio ponderado de las notas $\mathbf{T_1}, \mathbf{T_2}, \mathbf{T_3}$ y **AF** del estudiante.

Proyecto. El proyecto del curso, representado por (**P**) en el cálculo del promedio final, buscará la aplicación de los contenidos aprendidos en el curso a un problema real. Se desarrollará en grupos de 4 personas, los cuales podrán formarse inicialmente por solicitud. Si no tienen compañer@s, serán asignados de manera aleatoria entre las personas que no tengan grupo. Consistirá en 3 entregas de avances y una presentación final. Cada grupo contará con un ayudante asignado para el desarrollo del proyecto. Las fechas y más detalles sobre el proyecto, se darán a conocer a mediados del semestre. Para asegurar que todos los miembros del equipo colaboren en el desarrollo del proyecto, se deberá completar una evaluación de pares tras cada entrega.

Proceso de recorrección. Luego de publicadas las notas de una evaluación, se dará un periodo de una semana para recibir solicitudes de recorrección. Cada solicitud debe estar debidamente justificada, y debe ser enviada por los canales que el curso disponga para este propósito.

En caso de que la respuesta a la solicitud de recorrección no sea satisfactoria, se deberá llenar un formulario —dentro de una semana de publicada la recorrección— para solicitar que el profesor revise el caso. La decisión que se tome en esta instancia es inapelable.

GitHub Classroom Tanto actividades, proyecto y tareas de programación se entregarán **únicamente** a través de repositorios privados alojados en GitHub. **Este es el medio de entrega oficial y único del curso.**

Nota final y aprobación. La nota del curso se calcula como $\mathbf{NC} = (0{,}4 \times \mathbf{T} + 0{,}3 \times \mathbf{C} + 0{,}3 \times \mathbf{P})$. La nota final del curso **NF** se calculará como:

$$\mathbf{NF} = \begin{cases} \mathbf{NC} & \text{si } \mathbf{T} \geq 3{,}95 \text{ y } \mathbf{C} \geq 3{,}95 \text{ y } \mathbf{P} \geq 3{,}95 \\ \text{mín}(\mathbf{NC};\, 3{,}9) & \text{en otro caso.} \end{cases}$$

El alumno aprobará el curso si su nota final del curso **NF** es mayor a 3,95.

Todas las notas serán calculadas con **dos decimales**, salvo la nota final del curso que se calculará con **un decimal**.

Faltas de Ética

Cualquier situación de **copia** o **falta a la ética** detectada en alguna evaluación tendrá como **sanción un 1**,**1 final en el curso**. Esto sin perjuicio de sanciones posteriores que estén de acuerdo a la Política de Integridad Académica de la Escuela de Ingeniería y de la Universidad, que sean aplicables al caso. Rige para este curso tanto la política de integridad académica del Departamento de Ciencia de la Computación (ver anexo) como el **Código de honor de la Escuela de Ingeniería**.

Debido a la naturaleza de la disciplina en la que se enmarca el curso, está permitido el uso de código escrito por un tercero, pero solo bajo ciertas condiciones. Primero que todo, el uso de código ajeno **siempre debe** estar correctamente referenciado, indicando la fuente de donde se obtuvo. Y por otro lado, se permite el uso de código encontrado en internet u otra fuente de información similar, siempre y cuando su autor sea **externo al curso**, o en su defecto, sea parte del **equipo docente actual** del curso. Es decir, se puede hacer referencia a código ajeno al curso y código perteneciente al curso pero solo aquel escrito por el equipo docente, como material o ayudantías. Luego, compartir o usar código de una evaluación **actual o pasada** se considera **copia.** Contactarse con algún ajente externo para la ayuda o resolución derechamente de una evaluación individual, es considerado también como una **falta a la ética.**

Bibliografía

- K. Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems) by Ian H. Witten and Eibe Frank (Paperback - Jun 10, 2005)
- Introduction to Data Mining by Pang-Ning Tan, Michael Steinbach, and Vipin Kumar (Hardcover, May 12, 2005)
- Machine Learning, Tom M. Mitchell
- Kimball, R. "The data warehouse toolkit : the complete guide to dimensional modeling", John Wiley and Sons, 2002
- Data Mining: Concepts and Techniques, Second Edition (The Morgan Kaufmann Series in Data Management Systems) by Micheline Kamber Jiawei Han (Hardcover - Nov 3,2005)

Anexo: Política de integridad académica del Departamento de Ciencia de la Computación

Se espera los alumnos de la Escuela de Ingeniería de la Pontificia Universidad Católica de Chile mantengan altos estándares de honestidad académica, acorde al Código de Honor de la Universidad. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un Procedimiento Sumario. Es responsabilidad de cada alumno conocer y respetar el documento sobre Integridad Académica publicado por la Dirección de Pregrado de la Escuela de Ingeniería (Disponible en SIDING, en la sección Pregrado/Asuntos Estudiantiles/Reglamentos/Reglamentos en Ingeniería/Integridad Académica).

Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente *política de integridad académica*. Todo trabajo presentado por un alumno para los efectos de la evaluación de un curso debe ser hecho **individualmente** por el alumno, **sin apoyo en material de terceros**. Por "trabajo" se entiende en general las interrogaciones escritas, las tareas de programación u otras, los trabajos de laboratorio, los proyectos, el examen, entre otros.

En particular, si un alumno copia un trabajo, o si a un alumno se le prueba que compró o intentó comprar un trabajo, **obtendrá nota final 1.1 en el curso** y se solicitará a la Dirección de Pregrado de la Escuela de Ingeniería que no le permita retirar el curso de la carga académica semestral.

Por "copia" se entiende incluir en el trabajo presentado como propio, partes hechas por otra persona. En caso que corresponda a "copia" a otros alumnos, la sanción anterior se aplicará a todos los involucrados. En todos los casos, se informará a la Dirección de Pregrado de la Escuela de Ingeniería para que tome sanciones adicionales si lo estima conveniente.

Obviamente, está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, **siempre y cuando se incluya la referencia correspondiente**.

Lo anterior se entiende como complemento al Reglamento del Alumno de la Pontificia Universidad Católica de Chile (**http://admisionyregistros.uc.cl/alumnos/informacion-academica/reglamentos-estudiantiles**). Por ello, es posible pedir a la Universidad la aplicación de sanciones adicionales especificadas en dicho reglamento.