Trabajo final de grado

GRADO DE INGENIERÍA INFORMÁTICA

Facultad de Matemáticas
Universidad de Barcelona

Predicción de ranking de asignaturas a partir de resultados académicos

---

Autor: Tian Lan

Directora: Laura Igual
Realizado a: Departamento de
Matemáticas e Informática

Barcelona, 01 de Febrero de 2018

Abstract

This final degree project is part of a teaching innovation project (PID), it aims to create an "intelligent support system for tutor studies" based on a data science analysis of academic results. The system will help the tutor to make better decisions in their work of supervision and guidance to their students. The project was launched 3 years ago, some students of the Faculty of Mathematics and Computing worked and contributed in several aspects. The work that's going to be done in this project will be a continuation. It will mainly focus on the implementation of a variation of the prediction of subject ranking. The ranking of subjects will consist of ordering the subjects, depending on their difficulty. That is, the qualifications of the subjects of the following course for a student are predicted, using the qualifications of the subjects studied up to the moment. Based on these qualifications, it is deduced which subjects will be more "difficult" for the student. It will also be involved in the data analysis. More specifically, in inspecting the different ways in which the data could be treated to manage the missing values, with the expectation that the prediction accuracy could improve. Finally, the aim is to organize the code through the MVC structure, so that it can become more standardized.

Resumen

Este trabajo final de grado forma parte de un proyecto de innovación docente (PID), que tiene como objetivo crear un "sistema inteligente de soporte al tutor de estudios" en base a un análisis de ciencia de datos sobre resultados académicos. Este sistema servirá de ayuda al tutor para tomar mejores decisiones en su labor de supervisión y orientación a sus estudiantes. El proyecto se puso en marcha hace 3 años, unos estudiantes de la Facultad de Matemáticas e Informática trabajaron y contribuyeron en varios aspectos. El trabajo que se hará en este proyecto será una continuación. Se enfocará principalmente en la implementación de una variación de la predicción de ranking de asignaturas. El ranking de asignaturas, consistirá en ordenar las asignaturas, pendientes de cursar por un alumno, en función de su dificultad. Es decir, se predicen las notas de las asignaturas del siguiente curso para un alumno, utilizando las notas de las asignaturas cursadas hasta el momento. En función de estas notas se deduce qué asignaturas serán más "difíciles para el alumno. También se involucrará en el análisis de datos. Más concretamente, en inspeccionar las diferentes maneras en que los datos pueden ser tratados para gestionar los valores ausentes (missing values), con la expectativa de que la precisión de predicción pudiera mejorar. Por último, se pretende organizar el código mediante la estructura de MVC, de modo que quede más estandarizado.

Resum

Aquest treball final de grau forma part d'un projecte d'innovació docent (PID), que té com a objectiu crear un "sistema intel·ligent de suport al tutor d'estudis" en base a una anàlisi de ciència de dades sobre resultats acadèmics. El sistema servirà d'ajuda al tutor per a prendre millors decisions en el seu treball de supervisió i orientació als seus alumnes. El projecte es va posar en marxa fa 3 anys, uns estudiants de la Facultat de Matemàtiques e informàtica van treballar i contribuir en diversos aspectes. El treball que es farà en aquest projecte serà una continuació. S'enfocarà principalment en la implementació d'una variació de la predicció de rànquing d'assignatures. El rànquing d'assignatures, consistirà en ordenar les assignatures, pendents de cursar per un alumne, en funció de la seva dificultat. És a dir, es prediuen les notes de les assignatures del següent curs per a un alumne, utilitzant les notes de les assignatures cursades fins el moment. En funció d'aquestes notes es dedueix quines assignatures seran més difícils per a l'alumne. També s'involucrarà en l'anàlisi de dades. Més concretament, en inspeccionar les diferents maneres en que les dades podrien ser tractades per gestionar els valors absents (missing values), amb l'expectativa que la precisió de predicció pogués millorar. Per últim, es pretén organitzar el codi mitjançant l'estructura de MVC, de manera que quedi més estandarditzat.

Índice

1. Introducción:

1.1. Contexto del proyecto

Este trabajo de fin de grado, forma parte de un Proyecto de Innovación Docente (PID) [1]. Fue iniciado en el Departamento de Matemáticas e Informática y el Departamento de Métodos de Investigación i Diagnóstico en Educación (MIDE).

Fue planteado por la necesidad de ofrecer un sistema inteligente de soporte al tutor de estudios, mediante el cual, el tutor tiene la facilidad de conocer la situación académica de sus alumnos, y de tomar decisiones más informadas, basadas en datos, y en predicciones.

Varios estudiantes de la Facultad de Matemáticas e Informática han participado y contribuido en el proyecto, este trabajo también contribuye al PID.

El proyecto PID se divide en 5 fases:

1. Fase 1: Adquisición, ordenación, centralización y anonimización de los datos curriculares disponibles de los alumnos. Es la fase en la que se obtienen y se preparan los datos necesarios, antes de pasar a la fase analítica.

2. Fase 2: Análisis de los datos mediante técnicas de ciencia de los datos. De los datos que disponemos, se les hace un análisis estadístico, y explorar la información que se esconde por detrás de los datos.

3. Fases 3: Realizar predicciones mediante técnicas de aprendizaje automático. Aplicando algoritmos de predicción, se sacarán los resultados que interesan a partir de los datos históricos de los alumnos.

4. Fase 4: Desarrollo del sistema inteligente. En esta fase, se pueden incluir tareas como: implementación de la interfaz gráfica, construcción de la base de datos, desarrollo de una herramienta de testeo, etcétera.

5. Fase 5: Evaluación del sistema inteligente. Probar el sistema y obtener las retroalimentaciones (rendimiento, seguridad, simplicidad, etcétera), para poder realizar las posibles mejoras.

Este trabajo se enfoca principalmente en la fase 2 y 3.

1.2. Ciencia de datos

La ciencia de datos es un conjunto de etapas con el fin de extraer información subyacente en los datos, y transformarla en conocimiento. Ésta puede contener muchas disciplinas como matemáticas, estadísticas, programación informática, minería de datos, visualización de datos, etcétera.

El proyecto está marcado dentro del área de ciencia de datos, por lo tanto, en la realización, se emplean diversas técnicas (Análisis de datos, agrupamiento, predicción, etc) y herramientas relativas (Python, librerías de ciencia de datos, etc).

1.3. Aportaciones

Una de las funcionalidades existente en el sistema desarrollado en el PID es "predicción de ranking de asignaturas (PRAS)". Se predice el ranking mediante los datos continuos (calificaciones históricas) de los alumnos.

Cabe enfatizar de nuevo el objetivo de la funcionalidad planteada, que es conocer de antemano qué asignaturas le irán bien y mal a un alumno en el futuro. El tutor, disponiendo de esta información, puede indicar a un alumno en qué asignaturas va a tener problemas, así como darle consejos oportunos para que tenga una actitud de aprendizaje más adecuada; desde otra perspectiva, la predicción también muestra en qué asignaturas el alumno es relativamente bueno.

Además de la parte funcional, el trabajo también se involucrará en el pre-procesamiento de los datos para tratar los missing values. Se considera necesario este proceso, ya que puede influir en la precisión de la predicción.

A continuación, se enumeran las aportaciones de este TFG:

- En este TFG, se trata de presentar una variación de PRAS, en que se predirá el ranking de asignaturas a partir de los datos discretos (rankings históricos) de los alumnos. Se hará una comparativa entre los dos modos de predicción, tanto en aspectos cualitativos: el tratamiento de datos, el algoritmo seleccionado, etcétera, como cuantitativos: precisión, error, etcétera. Por supuesto, también se esperaría que se produzca una mejora de eficiencia en la predicción.

- Se hará un estudio sobre los missing values de los datos. Los missing values serán procesados de varias maneras, y también se hará una comparativa entre ellas para evaluar cuál es mejor a la hora de predecir.

- Se organizará el código del programa mediante la estructura Model-View-Controller, con el fin de facilitar la comprensión y reutilización del mismo código en un futuro. También se hará un menú que permite navegar por las diferentes opciones y probar todos los testeos en una sola ejecución.

2. Planificación

2.1. Diagrama de Gantt

Se han dibujado dos diagramas de Gantt, uno para la planificación estimada, otro para la real. Véase Fig 2.2.1 y Fig 2.2.2:

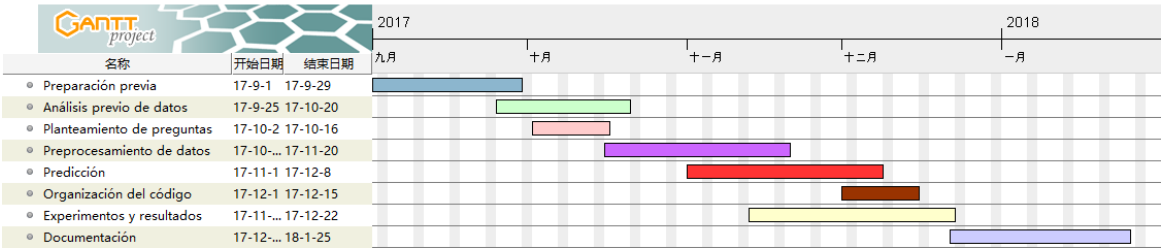


Fig 2.2.1: Planificación estimada

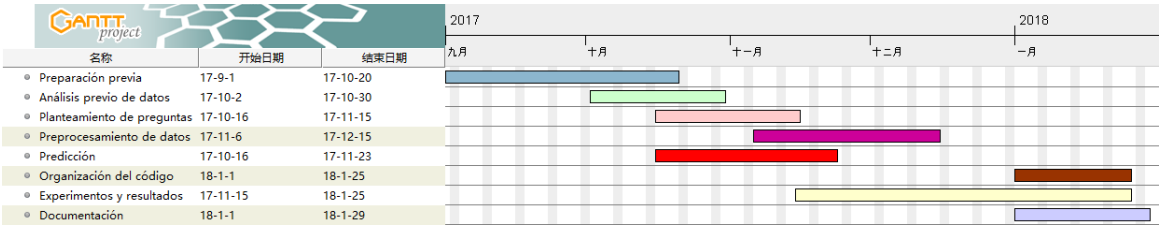


Fig 2.2.2: Planificación real

Según la comparación de los dos diagramas, se ve que muchas tareas se han iniciado con retraso, además, están prolongadas. Sobre todo, tres tareas están cargadas en las últimas semanas del tiempo previsto. La reflexión es, empezar las tareas puntualmente y tratar de acabarlas en el tiempo previsto.

Este TFG cuenta con un total de 18 créditos académicos, cada crédito requiere una dedicación de 25 horas, por lo tanto, el tiempo requerido para el TFG es de 450 horas. Estas horas están repartidas entre las tareas establecidas.

2.2. Evaluación económica

A continuación, vamos a calcular qué valor económico tiene este trabajo, véase el Cuadro 2.3.1:

Tarea	Horas (h)	Precio por hora (Euro)	Precio total (Euro)
Preparación previa	50	0	0
Análisis previo de datos	40	15	600
Planteamiento de preguntas	20	10	200
Preprocesamiento de datos	60	25	1500
Predicción	60	25	1500
Organización del código	20	10	200
Experimentos y resultados	100	25	2500
Documentación	100	10	1000
TOTAL	450	-	7500

Cuadro 2.3.1: Tabla de evaluación económica del trabajo

3. Desarrollo del proyecto

3.1. Etapas del proyecto

A continuación, se enumeran las etapas en las que el trabajo es dividido:

1. Disposición de los datos
2. Preparación previa (formación y estudio del trabajo previo)
3. Análisis previo de datos
4. Planteamiento de preguntas
5. Implementación
    a. Pre-procesamiento de datos
    b. Predicción
6. Organización del código
7. Experimentos y resultados

En los siguientes apartados, se dará una breve explicación de cada etapa, excepto la etapa 7, su explicación está en el capítulo 6.

3.2. Disposición de los datos

Los datos académicos que están a nuestra disposición contienen una amplia información de los alumnos que han cursado los grados de Matemáticas, de Ingeniería Informática y de Derecho en la Universidad de Barcelona.

1. Fuente

Estos datos han sido proporcionados por el departamento de planificación y gestión académica de la Universidad de Barcelona. Después de varios procesamientos sobre los datos en los trabajos anteriores, éstos han sido limpiados, enriquecidos, estructurados y recopilados en dos ficheros .csv.

2. Ficheros de datos

qualifications_mates_info.csv: contiene datos de dos grados, Matemáticas e Informática.
qualifications_dret.csv: contiene datos del grado Derecho.

3. Dimensión de los datos

Matemáticas: al inicio, tiene 9227 registros, después de resumirlos en una tabla de calificaciones, son 516 alumnos.
Informática: al inicio, tiene 9227 registros, después de resumirlos en una tabla de calificaciones, son 455 alumnos.
Derecho: al inicio, tiene 9227 registros, después de resumirlos en una tabla de calificaciones, son 3463 alumnos.

3.3. Preparación previa

1. Formación previa

Para familiarizarse con la ciencia de datos y el aprendizaje automático, se han mirado algunos vídeos, que son [2], [3], [4].

2. Estudio del trabajo anterior

Como ya se ha explicado, este trabajo es una continuación de lo implementado anteriormente. Para empezar, se han estudiado los puntos relacionados con la predicción de ranking que se había hecho, para poder por un lado, comprender mejor el mismo problema mediante sus explicaciones, y por otro lado, obtener las ideas relativas. También se ha estudiado el código implementado y que está publicado en un Github [5], haciendo el testeo, para conocer el flujo de ejecución, y cómo funciona cada bloque de códigos. Después de haber tomado ideas, se ha adaptado una porción del código (código responsable de cargar datos) desde el trabajo anterior al propio trabajo.

3. Traslado de IPython Notebook al Eclipse Pydev

Para un estudio más fácil y cómodo del flujo de ejecución, así como la inspección del contenido de las variables, hace falta la depuración del código. Sin duda, el Ipython Notebook es una herramienta muy potente, permite escribir el código en bloques diferentes, como una prosa, y todo en una hoja. Pero, el Ipython Notebook no ofrece una herramienta de depuración, es más, cuando un trabajo se hace largo, se necesita una organización, para ello, se ha buscado otro entorno de programación en Python, llamado Pydev, éste permite crear proyectos en Python y cuenta con unas características beneficiosas de ayuda al trabajo de programación. Véase el apartado 5.3.2 para más información.

3.4. Análisis previo de datos

En este apartado, se explica el estudio de datos que se ha hecho, junto con algunos cuadros de ilustración. En este trabajo, se usará la tabla del Pandas.DataFrame para el procesamiento de datos.

3.4.1. Selección de datos a procesar

Los datos adquiridos de cada grado contienen información de tres cursos académicos, véase el Cuadro 3.4.1 en que se muestra cómo son los datos en la tabla.

	Curso 1				Curso 2				Curso 3			
	Asig 1	Asig 2	...	Asig N	Asig 1	Asig 2	...	Asig N	Asig 1	Asig 2	...	Asig N
Alumno 1												
Alumno 2												
...					Calificaciones							
Alumno M												

Cuadro 3.4.1: Tabla de calificaciones de los tres cursos académicos

Usaremos datos de los primeros dos cursos, de los cuales, los del primero servirá para entrenar el predictor, y los del segundo para hacer la predicción.

3.4.2. Conversión de datos continuos a discretos

Al principio, los datos son continuos (calificaciones). Si queremos entrenar el predictor con los datos en forma del ranking, es necesario hacer una conversión. Véase Cuadro 3.4.3 en la que se ordenan, en forma de ranking del 1 (la nota más alta) al 10 (la nota más baja), las calificaciones de un alumno.

	Asig 1	Asig 2	Asig 3	Asig 4	Asig 5	Asig 6	Asig 7	Asig 8	Asig 9	Asig 10
Calificaciones	8.0	5.5	3.2	7.8	6.0	9.3	8.5	6.5	7.0	5.6
Ranking	3	9	10	4	7	1	2	6	5	8

Cuadro 3.4.3: Ejemplo de conversión de calificaciones a ranking

3.4.3. Distribución de los missing values

Muchas veces, podemos disponer de un conjunto de datos en el que se encuentran los missing values (MV) [6]. Un MV no tiene un valor, ya sea porque no se ha adquirido, o nunca ha existido.

En los datos que disponemos, un MV representa que la nota de una asignatura determinada está indefinida, esto puede ser debido a diversas causas:

1. El alumno no ha querido matricularse en la asignatura.
2. El alumno la matriculó y la ha anulado.
3. El alumno la tiene convalidada.

Una vez hemos cargado los datos, se guardan en una tabla DataFrame, vamos a mirar la distribución de MV en nuestros datos, véase el cuadro 3.4.5:

	Curso 1	Curso 2
Número de registros	516	516
Número de MV	445	3171
Número de registros MV	94	373

Cuadro 3.4.5: Recuentos de MV en la tabla inicial (grado de mates)

Número de registros: indica el número de filas en la tabla.

Número de MV: indica el número total de aparición de MV en la tabla, según recuentos.

Número de registros MV: indica el número de filas que contienen al menos un MV.

Se observa que hay un total de 270 abandonos curriculares, por lo tanto, de los 373 registros MV del curso 2, hay 270 que están plenos de MV. Después de eliminarlos de los datos, las cifras quedan así, véase el Cuadro 3.4.6.

	Curso 1	Curso 2
Número de registros	246	246
Número de MV	230	471
Número de registros MV	48	103

Cuadro 3.4.6: Recuentos de MV después de eliminar de la tabla los abandonos (grado de mates)

3.4.4. Estudio de alumnos outliers

Los alumnos outliers son aquéllos que llevan muchos missing values (MV) en sus notas, o tienen notas extremadamente bajas. Si incluimos los datos de ellos en el proceso de predicción, podrían influenciar de forma negativa al resultado, ya que estos datos son muy poco significativos o comunes.

Aquí se quiere estudiar si en los datos existen alumnos muy especiales. En concreto, se observan sus notas, y se tienen en cuenta dos puntos:

1. El número de los MV que lleva un alumno. Si es mayor que 5 en curso 1 o curso 2, se eliminará toda la fila. La razón es:
   En curso 1: cuando un alumno dispone de un número escaso de calificaciones válidas, no se considera bueno a la hora de calcular la correlación con los demás, porque esta correlación no será fiable.
   En curso 2: cuando un alumno dispone de un número elevado de MV, no se considera bueno a la hora de la predicción, porque podrá aumentar el riesgo de un resultado nulo.

2. Si el alumno tiene una nota media extremadamente baja. Los alumnos que cumplen esta condición pueden tener problemas con el estudio, por lo que es frecuente encontrar algunas calificaciones 0.0 en sus notas. Una calificación 0.0 para un alumno indica que éste no se ha presentado al examen final, pero la calificación real no está reflejada.

Al excluirlos de la tabla, quedan 173 filas (alumnos).

3.4.5. Pre-procesamiento de los missing values

Una vez hemos determinado que existe una cantidad considerable de missing values (MV) en los datos, tenemos que buscar algunas maneras de tratarlos, con el fin de evitar problemas producidos por MV y mejorar la predicción. Las soluciones propuestas son:

1. Eliminación: cuando se encuentran MV (incluso si sólo uno) en las notas de un alumno, ya sea en primer curso o segundo, este alumno será descartado de la tabla. Después de la eliminación, quedan 130 filas en la tabla.

Esta acción es sencilla y rápida, pero puede bajar la dimensión de datos.

2. Reemplazo: pretende sustituir cada MV de la tabla por un valor razonable en su lugar, este proceso se realiza mediante un recomendador de reemplazo de MV.
   Esta acción requiere cierto tiempo, y la fiabilidad del valores sustituyentes depende de la precisión del recomendador. La ventaja es que puede prevenir la pérdida de datos respecto al caso anterior.

3. Mantenimiento: se mantienen los MV en la tabla, o sea, no hay que hacer nada respecto a los datos. Pero, como que no se pueden aplicar operaciones numéricas sobre los MV, hay que establecer condiciones o filtros a la hora de tratarlos en el proceso de predicción.
   Su implementación es más complicada respecto a los dos anteriores. La ventaja es que mantiene los datos en su forma original.

3.5. Planteamiento de preguntas

A continuación, se da una enumeración de las preguntas planteadas, que son los puntos que queremos alcanzar en este trabajo:

1. **Predicción de ranking mejora si se utilizan datos discretos para entrenar el predictor?**

   Se pretende hacer una comparativa entre las dos maneras de realizar la predicción de ranking, como lo explicado en la parte introductoria, antes se predecía usando los datos continuos (calificaciones) para el entrenamiento, ahora se intenta hacerlo con los datos discretos (ranking en sí). Véase en Fig 3.5.1 y Fig 3.5.2 una ilustración del flujo de las dos formas de predicción.

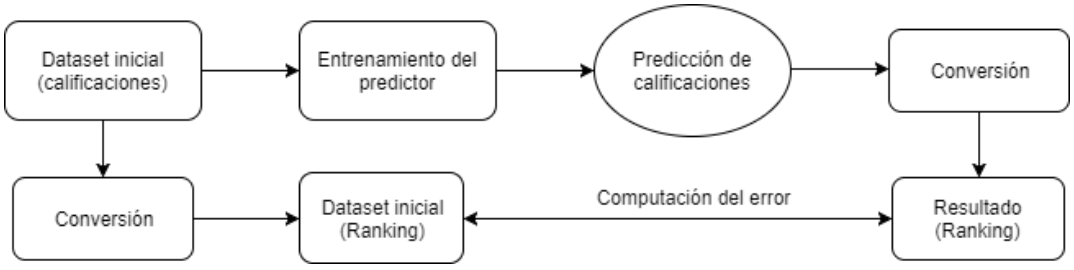


Fig 3.5.1: Flujo de predicción de ranking mediante datos continuos

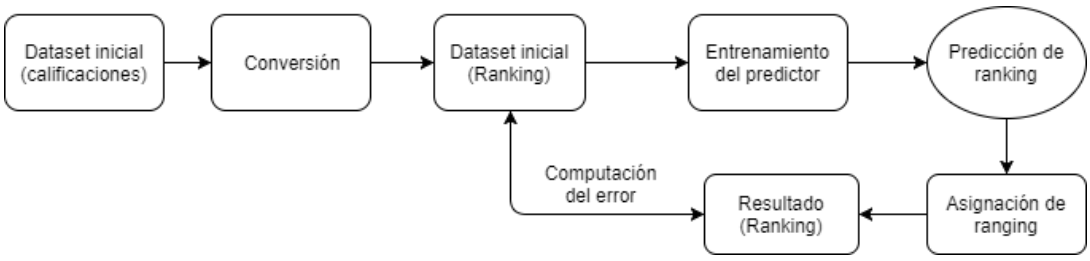


Fig 3.5.2: Flujo de predicción de ranking mediante datos discretos

2. **Comparativa del rendimiento entre las diferentes maneras de pre-procesamiento de datos.**
   Debido a la distribución de los missing values en los datos, se vuelve difícil la predicción de ranking, ya que pueden impedir la ordenación de los valores. Para ello, han surgido tres maneras de tratar el problema de MV, que son la eliminación, reemplazo y mantenimiento de MV en los datos. Se hará una comparativa del resultado de los tres casos. Véase el apartado 3.4.5 para más información.

3. **Al excluir los alumnos outliers de los datos, se mejora la precisión de predicción?**
   En este punto, vamos a comparar los resultados de la predicción de los dos casos (con y sin alumnos outliers). Se observa si realmente va a mejorar la precisión de la predicción como se había supuesto. Véase el apartado 3.4.4 para más información.

4. **Comparativa del rendimiento entre los modelos de predicción: Random Forest y Recomendador Colaborativo basado en Estudiantes.**
   En esta pregunta, se pretende medir el rendimiento de cada predictor, y compararlos.
   El dicho rendimiento incluye:
   1. Precisión: se computa con varias medidas como MAE, accuracy, etc.
   2. Velocidad: se calcula el tiempo usado.

   Los dos indicadores son computados bajo el proceso de Cross Validation (CV, validación cruzada). Véase el apartado 4.3 para una explicación más detallada.

5. **Menú textual de navegación**
   Después de haber implementado todas las partes, se quiere hacer un menú multinivel, para poder ir navegando entre las diferentes opciones, de manera que, se pueden testear todos los posibles casos planteados en una sola ejecución. Véase el apartado 3.6.5 para más información.

3.6. Implementación (componentes del programa)

1. **Datos**

Se implementa una clase para gestionar los datos de los que procesamos (datos de los tres grados), la clase tiene una estructura de datos y métodos de gestión.

2. **Predictores**

Los dos predictores que aplicaremos son Random Forest Regressor y Recomendador Colaborativo basado en Estudiantes. No vamos a implementar el primero, sino usarlo de la librería Scikit Learn; el segundo sí vamos a

implementarlo, con la ventaja de poder diseñar un algoritmo de predicción propio y manejable. Véase los apartados 4.1.1 y 4.1.2 para más información.

3. Pre-procesamiento de los missing values

Según lo explicado en el apartado 3.4.5, hemos pensado tres maneras de pre-procesar missing values (MV). La eliminación es la manera más obvia y fácil de implementar. Aquí cabe destacar la de reemplazo y la de mantenimiento de MV.

a. Reemplazo: véase en el apartado 4.1.2.4.
b. Mantenimiento: es el único caso que permite la aparición de MV en los datos, para ello, se han establecido los filtros pertinentes a la hora de predecir y validar los resultados, porque hay que evitar operaciones con los MV. Este caso sólo se aplicará a predicción de calificaciones (Fig 3.5.2), ya que los MV impiden el proceso de ranking.

4. Validación cruzada de 10 iteraciones

Usaremos la técnica del 10 Fold Cross Validation para evaluar los predictores, la implementación consiste en tres partes:

1. Separación de datos en 10 porciones de training_set y testing_set emparejados.
2. Realizar la predicción con cada una de las 10 parejas de datos, guardar los resultados.
3. Aplicar medidas de validación para calcular y obtener el promedio de los errores y/o precisiones, así como el tiempo de la predicción.

5. Menú de navegación

Se implementará un menú textual que permite la navegación entre las diferentes opciones del programa. Así, podemos hacer testeos de predicción uno tras otro en una sola ejecución, sin tener que volver a cargar los datos o modificar el código para probar con una opción diferente. Véase la Fig 3.6.1 en que se muestra una ilustración del menú.

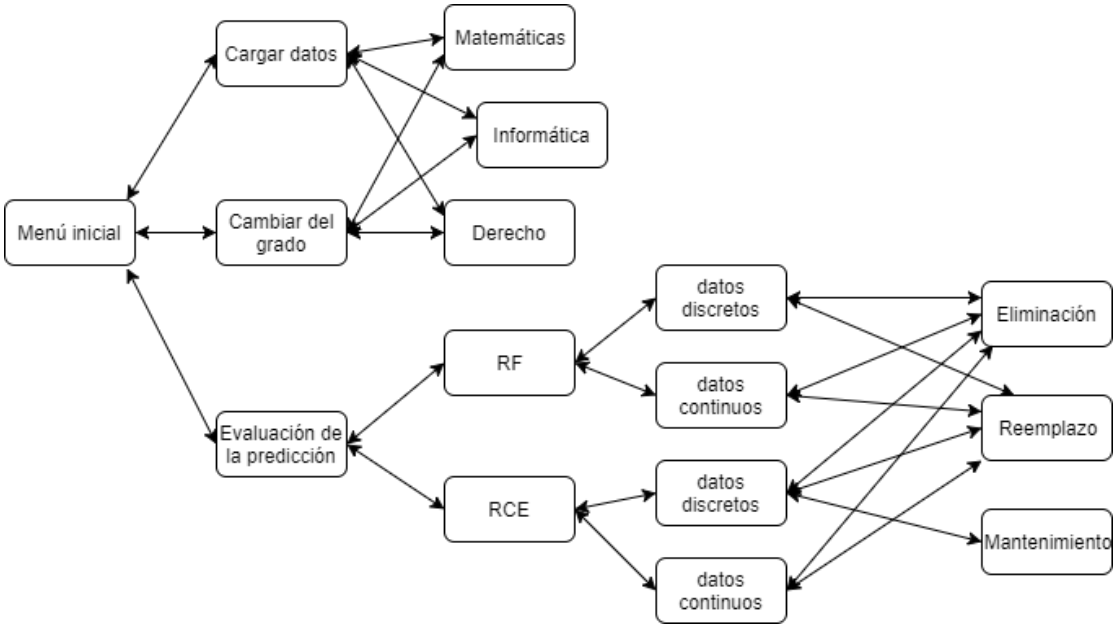


Fig 3.6.1: Diagrama que muestra el flujo de navegación del programa

6. Funciones de soporte

Se implementarán funciones para hacer tareas específicas, por ejemplo: computar errores, cargar datos, conversión de datos, validación de datos, etc. Serán altamente reutilizados en el programa.

7. Visualización

Son aquellas funciones que sirven para preparar los datos y visualizarlos.

3.7. Organización del código de trabajo.

1. Patrón MVC

El código del trabajo está estructurado según el patrón Model-View-Controller (MVC, modelo-vista-controlador) para tener una estructura simple y la fácil comprensión del flujo de ejecución del código.

2. Modularización del código

Se trata de ordenar y poner el código del programa en bloques adecuados, de manera que cada bloque (módulo o función) tenga una función específica, con el fin de disponer de un código compacto y limpio, así como fácil de reutilizar para futuros trabajos.

4. Técnicas empleadas

4.1. Predictores

4.1.1. Random Forest (RF)

1. Qué es un RF?

RF es uno de los algoritmos más populares y potentes del aprendizaje automático supervisado, y está basado en la técnica de bagging. Puede resolver problemas de regresión y clasificación. Como el nombre indica, RF está formado por un conjunto de árboles de decisión independientes.

Las dos fases importantes son:

a. Construcción, separa un conjunto de datos en M subconjuntos de forma aleatoria, cada uno de los subconjuntos representa un clasificador de árbol de decisión.
b. Predicción: predecir mediante los clasificadores, el promedio de los cuales es el resultado.

2. Random Forest Regressor (RFR) vs Random Forest Classifier (RFC)

Véase en Cuadro 4.1.1.1 una comparativa entre RFR y RFC:

Tipo de RF	Tipo de datos que gestiona	Obtención del resultado
Regressor	Valores numéricos	Media aritmética de los árboles
Classifier	Clases	Voto mayoritario de los árboles

Cuadro 4.1.1.1: Comparación entre RFR y RFC

Elección del tipo para las dos maneras de predicción:

a. Predicción mediante datos continuos (calificaciones): es obvio que se elige el RFR [7] como el predictor.
b. Predicción mediante datos discretos (ranking): aparentemente, se puede elegir cualquiera de los dos, porque los valores del ranking (1 a 10) pueden ser tratados como números o clases. Al inicio, se ha usado RFC, y se observa que no permite una clasificación estricta, pues en el resultado aparecen clases repetidas. Pero el RFR no presenta este problema, por lo que ha sido elegido como el predictor.

Se utiliza el RFR de la librería Sklearn.

4.1.2. Recomendador Colaborativo basado en Estudiantes (RCE)

1. Qué es un recomendador colaborativo?

RCE [8] es un tipo de recomendador que aplica la técnia del Filtrado Colaborativo (FC). En general, el FC es el proceso de filtrado de información o modelos, que usa técnicas que implican la colaboración entre múltiples agentes, fuentes de datos, etc. En el enfoque más reciente, el FC es un método para hacer predicciones automáticas (filtrado) sobre los intereses de un usuario mediante la recopilación de las preferencias o gustos de información de muchos usuarios (colaborador).

En el contexto de nuestro proyecto, los colaboradores son estudiantes.

2. Cómo funciona el RCE en el programa?

El RCE se basa en la idea de que si un estudiante E1 tiene notas similares a otro estudiante E2 en el curso 1, es más probable que E1 tenga notas similares que E2 en el curso 2, respecto a un estudiante elegido al azar.

Para computar la similitud entre dos estudiantes, se usa el algoritmo de K Nearest Neighbours (KNN) basado en la correlación de Pearson. El proceso de KNN consiste en computar la correlación de Pearson [9] entre el "student" (el estudiante a quien aplicar la recomendación) y los demás. Luego elegir aquéllos K estudiantes que han obtenido mayor puntuación.

Una vez hemos encontrado los K estudiantes respecto al estudiante E1, procedemos a predecir (recomendar) notas que sacaría el E1, haciendo la media ponderada de los K estudiantes. Tomamos un ejemplo de la predicción de ranking mediante datos continuos (calificaciones), véase el Cuadro 4.1.2.2 para una mejor comprensión del cómputo.

	Similitud(Sim)	A1	A2	A3	Sim x A1	Sim x A2	Sim x A3
Estudiante 1	0.75	-	6.5	7.0	-	4.875	5.25
Estudiante 2	0.40	7.0	8.3	-	2.8	3.32	-
Estudiante 3	0.66	5.0	9.2	7.2	3.0	6.072	4.752
Total					5.8	14.267	10.002
Suma Sims					1.06	1.81	1.41
Total/Suma Sims					**5.47**	**7.88**	**7.09**

Cuadro 4.1.2.2: Un ejemplo de la computación de la media de ponderación.

Como se observa en el Cuadro, La recomendación para el "student" da este resultado: (A1=5.47, A2=7.88, A3=7.09); después de convertirlas en ranking,: (A1=3, A2=1, A3=2), que es el resultado final de la predicción.

3. **Aplicación de RCE en el programa**
    a. Predicción del ranking mediante datos continuos
    b. Predicción del ranking mediante datos discretos
    c. Reemplazar los missing values

Ya hemos visto el funcionamiento de las dos primeras, en el siguiente punto, se explica detalladamente el de la tercera.

4. Reemplazador de Missing Values (RMV) usando recomendador

RMV tiene la misma esencia que un RCE, pero su propósito es diferente. Dado un conjunto de alumnos (de un solo curso), RMV pretende reemplazar los missing values (MV) que están en las notas. Del conjunto, las filas que contienen al menos un missing value forman el subconjunto del testeo, los demás forman el subconjunto de ambos entrenamiento y predicción. Véase la Fig 4.1.2.3 para ilustrarse:

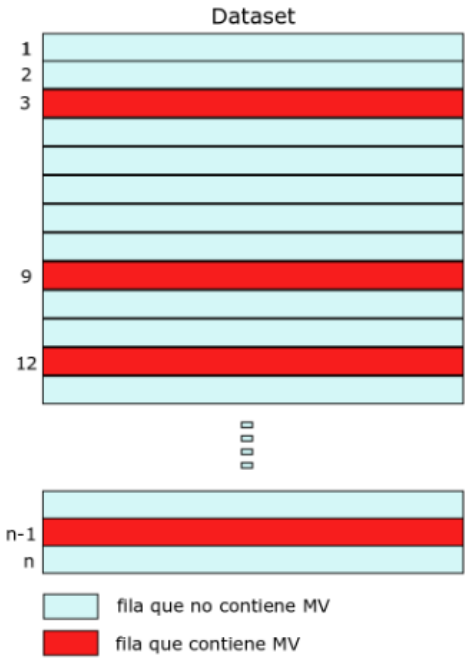


Fig 4.1.2.3: Conjunto en que se marcan filas MV y filas normales.

Las filas rojas en posiciones de 3, 9, 12, n-1 forman el subconjunto del testeo; Las azules claros, por un lado, sirven para entrenar y por otro lado, para recomendar valores de reemplazo a los MV que hay en el subconjunto del testeo. Mediante la predicción, los MV que hay en el subconjunto del testeo serán reemplazados (imputados).

4.2. Métricas:

A continuación, se enumeran las medidas utilizadas para evaluar el resultado de una predicción, y se da una explicación de cada una bajo el contexto de las predicciones realizadas.

4.2.1. Mean Absolute Error (MAE, error promedio absoluto)

Es una medida que está basada en calcular el promedio de errores absolutos entre y_{true} (y real) e y_{pred} (y predicho). La fórmula es:

$$\boldsymbol{MAE = \frac{1}{n} \cdot \sum_{i=1}^{n} |y_{true_i} - y_{pred_i}|}$$

En la práctica, esta medida se usa para calcular la diferencia entre el ranking real (y_{true}) y el predicho (y_{pred}). A continuación, se ilustra con un ejemplo el cómputo de MAE:

$$y_{true} = \begin{bmatrix} 3 & 2 & 1 & 4 \\ 1 & 3 & 2 & 4 \\ 2 & 4 & 1 & 3 \\ 3 & 2 & 1 & 4 \end{bmatrix}, \quad y_{pred} = \begin{bmatrix} 2 & 1 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 4 & 2 & 1 & 3 \\ 2 & 1 & 4 & 3 \end{bmatrix}$$

$$\begin{aligned} \text{nominador} = \; & |3-2| + |2-1| + |1-3| + |4-4| + |1-1| + |3-2| + |2-3| \\ & + |4-4| + |2-4| + |4-2| + |1-1| + |3-3| + |3-2| \\ & + |2-1| + |1-4| + |4-3| = 16 \end{aligned}$$

$$\text{denominador} = 16$$

$$\text{MAE} = \frac{16}{16} = 1$$

Este 1 indica que el error medio de la predicción (de cada elemento de la matriz) es 1.

4.2.2. Accuracy (Acc, precisión) / Non Strict Accuracy (Acc', precisión no estricta)

Se basa en recontar el número de aciertos entre y_{true} (y real) e y_{pred} (y predicho). La fórmula es:

$$\boldsymbol{\text{Acc} = \frac{1}{n} \cdot \sum_{i=1}^{n} y_{c_i}, where} \begin{cases} \boldsymbol{y_{c_i} = 0 \; if \; y_{true} \neq y_{pred}} \\ \boldsymbol{y_{c_i} = 1 \; if \; y_{true} = y_{pred}} \end{cases}$$

Tomamos el mismo ejemplo del anterior para mostrar un ejemplo de computar Acc:

$$y_{true} = \begin{bmatrix} 3 & 2 & 1 & 4 \\ 1 & 3 & 2 & 4 \\ 2 & 4 & 1 & 3 \\ 3 & 2 & 1 & 4 \end{bmatrix}, \quad y_{pred} = \begin{bmatrix} 2 & 1 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 4 & 2 & 1 & 3 \\ 2 & 1 & 4 & 3 \end{bmatrix}$$

$$\text{Acc} = \frac{0+0+0+1+1+0+0+1+0+0+1+1+0+0+0+0}{16} = 0.31$$

La precisión estricta es 31%.

Acc' es una variación de la medida Acc, calcula de la misma manera, pero no es estricto porque permite un cierto error, la fórmula es:

$$\mathbf{Acc'} = \frac{\mathbf{1}}{\boldsymbol{n}} \cdot \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{y_{c_i}}, \boldsymbol{where} \begin{cases} \boldsymbol{y_{c_i}} = \mathbf{0}\ \boldsymbol{if}\ \boldsymbol{y_{true}} - \boldsymbol{y_{pred}} > \boldsymbol{E} \\ \boldsymbol{y_{c_i}} = \mathbf{1}\ \boldsymbol{if}\ \boldsymbol{y_{true}} - \boldsymbol{y_{pred}} \leq \boldsymbol{E} \end{cases}$$

Donde la E es un error discreto previamente definido.

Si definimos E=1, el resultado del ejemplo sería:

$$\text{Acc} = \frac{1+1+0+1+1+1+1+1+1+0+0+1+1+1+1+0+1}{16} = 0.75$$

La precisión no estricta (que permite un error ≤ 1) es 75%.

4.2.3. Pearson Correlation (PC, correlación de Pearson)

A diferencia de las medidas de error numérico, PC es una medida que mide la relación lineal entre dos vectores, computando la similitud de la distribución de los valores. Está definido por la siguiente fórmula:

$$\mathbf{PC}\ (\mathbf{X}, \mathbf{Y}) = \frac{\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} (\boldsymbol{X_i} - \overline{\boldsymbol{X}}_{\boldsymbol{i}}) \cdot (\boldsymbol{Y_i} - \overline{\boldsymbol{Y}}_{\boldsymbol{i}})}{\sqrt{\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} (\boldsymbol{X_i} - \overline{\boldsymbol{X}}_{\boldsymbol{i}})^{\mathbf{2}}} \cdot \sqrt{\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} (\boldsymbol{Y_i} - \overline{\boldsymbol{Y}}_{\boldsymbol{i}})^{\mathbf{2}}}}$$

En la práctica, PC se usa para medir la similitud (relación lineal) entre dos estudiantes según sus notas. Véase el siguiente ejemplo:

$\mathrm{v}_1 = [1,2,3,4,5,6], \qquad v_2 = [1,3,5,2,6,4], \qquad \mathrm{v}_3 = [1,2,4,3,6,5]$

$\mathrm{PC}(\mathrm{v}_1, v_2) = 0.60$

$\mathrm{PC}(v_2, v_3) = 0.89$

v_3 ha obtenido una puntuación de correlación mayor que v_{s2}, en computación de similitud al v_{s1}. Véase la Fig 4.2.3.1 para una visualización gráfica de los tres vectores:

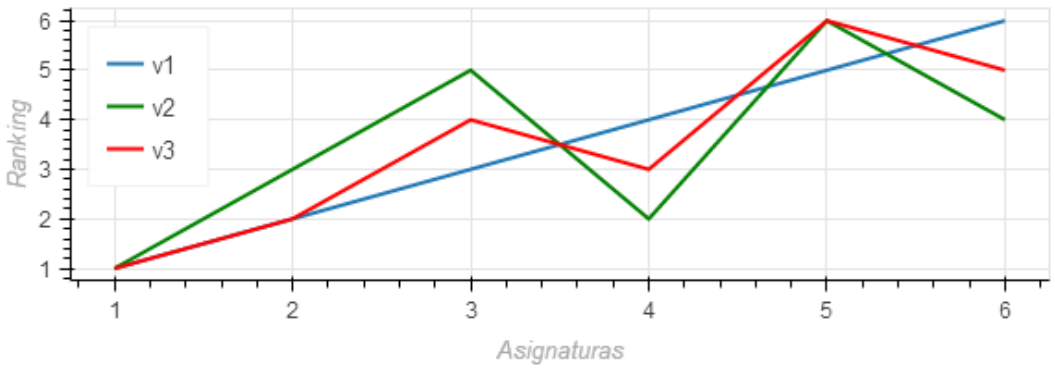


Fig 4.2.3.1: Gráfica que muestra el ranking de notas de tres alumnos en forma lineal.

4.2.4. Standard Deviation (STD, desviación típica)

STD es una medida de dispersión para variables cuantitativas. Al contrario de las medidas de tendencia central, STD indica la desviación que presentan los datos en su distribución respecto de la media aritmética de dicha distribución. La definición de su fórmula es:

$$\mathrm{STD} = \sqrt{\frac{1}{n}\sum_{n}^{i=1}(x_i - \bar{x})^2}$$

En la práctica, STD se usa para computar la dispersión de los errores producidos respecto a la media aritmética de dichos errores, en una predicción. Tener en cuenta que $x_i = |\boldsymbol{y_{true_i}} - \boldsymbol{y_{pred_i}}|$, y tomamos el siguiente ejemplo:

$$y_{true} = \begin{bmatrix} 3 & 2 & 1 & 4 \\ 1 & 3 & 2 & 4 \\ 2 & 4 & 1 & 3 \\ 3 & 2 & 1 & 4 \end{bmatrix}, \quad y_{pred} = \begin{bmatrix} 2 & 1 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 4 & 2 & 1 & 3 \\ 2 & 1 & 4 & 3 \end{bmatrix}$$

$$\mathrm{x} = [1,1,2,0,0,1,1,0,2,2,0,0,1,1,3,1], \qquad \mathrm{n} = 16, \qquad \bar{x} = 1$$

$$\mathrm{STD} = \sqrt{\frac{(1-1)^2 + (1-1)^2 + (2-1)^2 + \cdots + (3-1)^2(1-1)^2}{16}} = \frac{12}{16} = 0.75$$

Los errores tienen una dispersión de 0.75 respecto a su media (que es 1).

4.3. Cross Validation (CV, validación cruzada)

La validación cruzada [10] es una técnica utilizada para evaluar los resultados de un análisis estadístico. Consiste en separar los datos en dos partes diferentes llamadas entrenamiento y prueba, bajo K iteraciones, y calcular la media aritmética obtenida de las medidas de evaluación sobre diferentes particiones. Véase la Fig 4.3.1 para ilustrarse.

Fig 4.3.1: Una ilustración de la validación cruzada en K=10 iteraciones.

Se utiliza en entornos donde el objetivo principal es la predicción y se quiere estimar la precisión de un modelo que se llevará a cabo a la práctica.

La puntuación del cv se puede obtener mediante tres vías:

a. **cross_val_score (clf, X, y, cv)**: el cálculo es muy directo, ya que al pasarle los parámetros necesarios, te da la puntuación. Si el "cv"=10, indica que se harán 10 validaciones, por tanto, la porción de test-set es de un décimo en cada validación. Cuando "cv"=integer, por defecto se usa el KFold como el iterador del cross validation. En cuanto a la medida de errores, por defecto, esta función emplea el método "score" del estimador "clf".

b. **kf = KFold (n_splits=10, random_state=0)**: reparte los índices de un conjunto en dos subconjuntos (índices del entrenamiento, índices del testeo) 10 veces, como indica la Fig XXX. Después, se itera 10 veces para ir posicionando los elementos del conjunto según estos índices. Podemos realizar predicciones y computaciones de errores de forma libre.

c. **cross_validate (clf, X, y, scoring, cv)**: hace lo mismo que la "cross_val_score", y permite especificar múltiples médidas de errores para la evaluación. Además, ofrece varios tipos de puntuaciones.

Se ha elegido la vía b para evaluar bajo CV.

5. Herramientas empleadas

5.1. Herramientas de soporte

Github

El Github [11] es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones de Git. Te permite almacenar el código de los proyectos públicamente, aunque si utilizas una cuenta de pago, también puedes hospedar repositorios privados.

El Github también cuenta con unas características que ayudan al trabajo colaborativo entre los programadores, se destacan las siguientes:

- Wiki para cada proyecto
- Página web para cada proyecto
- Gráfico para ver cómo los desarrolladores trabajan en sus repositorios y ramas del proyecto.
- Gestor de proyectos de estilo Kanban

5.2. Herramientas de programación

1. Python

Python [12] es un lenguaje de programación interpretado de alto nivel, funcional, orientado a objetos e interactivo. Python tiene la poderosa capacidad de contar con una sintaxis muy limpia y clara, es decir, altamente legible y comprensible para el ser humano. Cuenta con módulos, clases, excepciones, tipos de datos de muy alto nivel así como tipado dinámico.

El motivo principal de optar por Python para la implementación de este proyecto es que dispone de unas amplias librerías muy potentes para trabajar en la ciencia de datos.

2. Pandas

Pandas [13] es una librería de Python para la manipulación y análisis de datos. Dispone de unas estructuras de datos que son especialmente útiles a la hora de manipular tablas, p.e: consultas a la tabla, inserción o eliminación de información, operaciones matemáticas aplicadas a los datos, etc.

3. Numpy

Numpy [14] es una librería de Python y es fundamental para la computación científica. Nos permite trabajar con vectores y matrices de una forma cómoda y eficiente. Además de sus usos científicos obvios, Numpy también se puede usar como un contenedor multidimensional de datos genéricos. Es más, ofrece una serie de funciones matemáticas de alto nivel para poder operar con dichos objetos.

4. Sklearn

Sklearn [15] es una librería de Python que ofrece herramientas simples y eficientes para la minería y análisis de datos. Proporciona varios modelos (o algoritmos) para resolver problemas de clasificación, predicción, agrupación, entre otros.

5. Matplotlib

Matplotlib [16] es una biblioteca de trazado 2D de Python, que produce figuras de calidad en una variedad de formatos impresos y entornos interactivos en todas las plataformas. Matplotlib se puede utilizar en scripts Python, el shell Python e IPython, el bloc de notas jupyter, servidores de aplicaciones web y cuatro toolkits de interfaz gráfica de usuario.

6. Anaconda

Anaconda [17] es una distribución de código abierto del lenguaje de programación Python para procesamiento de datos de gran escala, analítica predictiva y computación científica, que trata de simplificar la gestión y despliegue de paquetes. Provee la conveniencia de tener el Python y más de 150 paquetes científicos instalados automáticamente de una vez.

5.3. Herramientas de edición

1. Microsoft Word

Microsoft Word [18] es un procesador de texto híper-desarrollado, cuenta con una gran posibilidad y libertad a la hora de trabajar con el texto. Dispone de una gran variedad de características para una elaboración textual más diversificada, estética y cómoda, entre las cuales se destacan:

- Inserciones de elementos no textuales como imágenes, tablas, figuras, ecuaciones, etc.
- Un sistema de navegación bien desarrollado: numeración de páginas, enlaces internos.
- Enumeraciones personalizadas
- Tipografía personalizada
- El sistema de visualización
- Diseño de página

La versión de Microsoft Word utilizada es de 2013.

2. Eclipse-Pydev

Pydev [19] es un complemento de terceros para Eclipse. Es un entorno de desarrollo integrado (IDE) utilizado para programar en Python. Tiene soporte a

muchísimas características útiles que ayudan con una programación más eficiente. A continuación, se enumeran aquéllas características de las que se ha ayudado:

- Depuración gráfica
- Consola de depuración (permite la exploración interactiva en modo suspendido)
- refactorización del código
- el análisis del código
- Autocompleto del código (incluyendo auto importación)
- Código plegable
- Analizador de errores
- Bloques de comentario
- Etc.

La combinación del Pydev bajo Eclipse, ha facilitado dramáticamente la programación, el análisis y la organización del código en este trabajo.

6. Experimentos y resultados

En este capítulo, vamos a testear y evaluar el programa implementado. El objetivo es dar respuestas a las cinco preguntas planteadas (apartado 3.5). Para las cuatro primeras, nos focalizamos en la comparativa del rendimiento de la predicción para los casos planteados; en cuanto a la quinta, trataremos de testear el menú (de varios niveles) y asegurar que no haya ningún error inesperado.
Después de haber evaluado y recopilado los resultados, es la hora de buscar maneras para poder visualizarlos, ya sea por tablas o gráficamente.

6.1. Preparación de los experimentos

Como ya hemos visto en los capítulos anteriores, se han planteado diferentes casos que se deben validar. Los casos se indican a continuación:

- Grados: Matemáticas, Informática y Derecho.
- Modos de predicción: predicción de ranking mediante datos continuos y datos discretos.
- Predictores: RF y RCE.
- Maneras del pre-procesamiento de missing values (MV): eliminación, reemplazo y mantenimiento.

Todos estos casos se pueden combinar entre sí, excepto el mantenimiento de MV, que sólo es aplicable a la predicción de ranking mediante datos continuos con RCE. En fin, tenemos un total de 27 casos a experimentar.

Todos los casos se validan de la misma manera. Para simplificar, nos focalizamos en la explicación del experimento de un solo caso. Escogemos la combinación de Matemáticas, predicción de ranking mediante datos discretos, RCE y eliminación. Lo llamamos experimento 1.

El segundo experimento que vamos a realizar consiste en comparativas entre dos o varios casos, y visualizarlos conjuntamente para conocer de forma visual sus diferencias. Lo llamamos experimento 2.

Por último, se categorizan todos los casos, y se muestran sus resultados en diferentes tablas. Lo llamamos experimento 3.

6.2. Experimento 1

6.2.1. Descripción

La predicción de ranking que estudiamos pertenece a un problema de multilabel (multi-etiquetas), ya que la target (etiqueta/s) tiene más de un valor. En la predicción, la target es : 1,2,3,4,5,6,7,8,9,10, que corresponde a los valores del ranking.

Queremos conocer la relación entre los valores del ranking real y del ranking predicho. Mediante la matriz de confusión que se presenta a continuación (Fig 6.2.1), podemos ver los porcentajes de todas las combinaciones ([1,1],[1,2],…,[10,9],[10,10]).

6.2.2. Gráficas y análisis

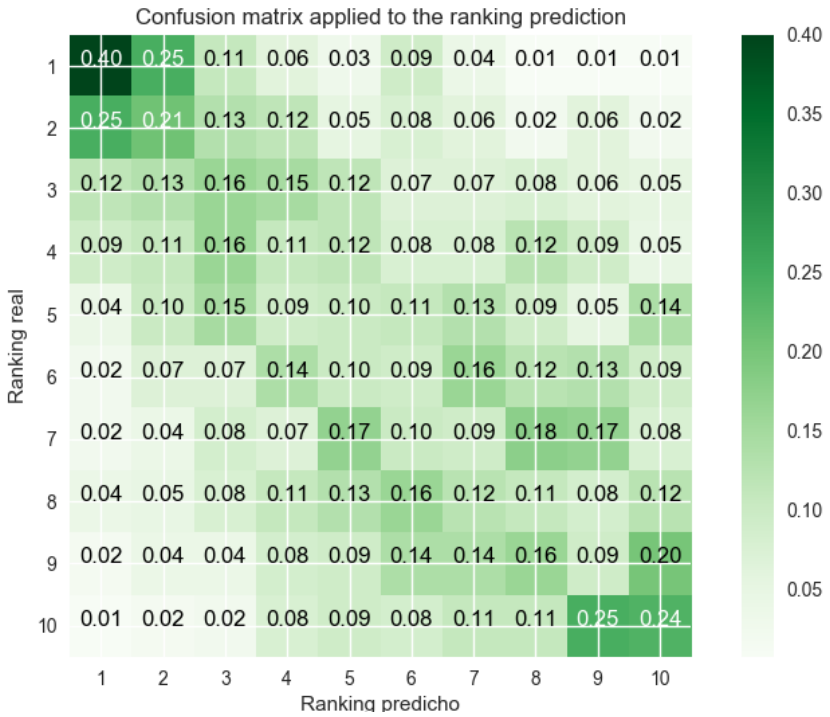


Fig 6.2.1: Matriz de confusión aplicada a la predicción del ranking

En esta gráfica, podemos ver dos elementos:

1. Barra de color: es una barra vertical que presenta un gradiente de color verde. Se pinta un color en cada casilla del cuadro del ranking, cuando más oscuro es el color, mayor es el porcentaje de aciertos.
2. Cuadro del ranking: es una matriz 10x10. El eje X representa el ranking predicho, el eje y representa el ranking real. En cada una de las casillas hay un color y el porcentaje correspondiente. La diagonal descendiente representa el conjunto de los verdaderos positivos, en el cual el ranking real y predicho coinciden.

Primero, vamos a mirar el porcentaje de aciertos para cada valor del ranking. Véase el Cuadro 6.2.2:

Valor del ranking	1	2	3	4	5	6	7	8	9	10
% de aciertos	40%	21%	16%	11%	10%	9%	9%	11%	9%	24%

Cuadro 6.2.2: Porcentajes de aciertos de 1 a 10.

El 1 ha obtenido el porcentaje más alto (40%), es decir, cuando realmente el valor es 1, la predicción da 1 con una probabilidad de 40%. Después, el 2 y 10 tienen un porcentaje relativamente alto que los demás (3 a 9), pero no deja de ser bajo. La métrica aplicada para calcular los porcentajes es la precisión estricta, definida en el apartado 4.2.2.

Vamos a relajar un poco el estándar, y usamos la precisión no estricta, y que permita un error E=2. La siguiente Fig 6.2.3 ofrece una ilustración:

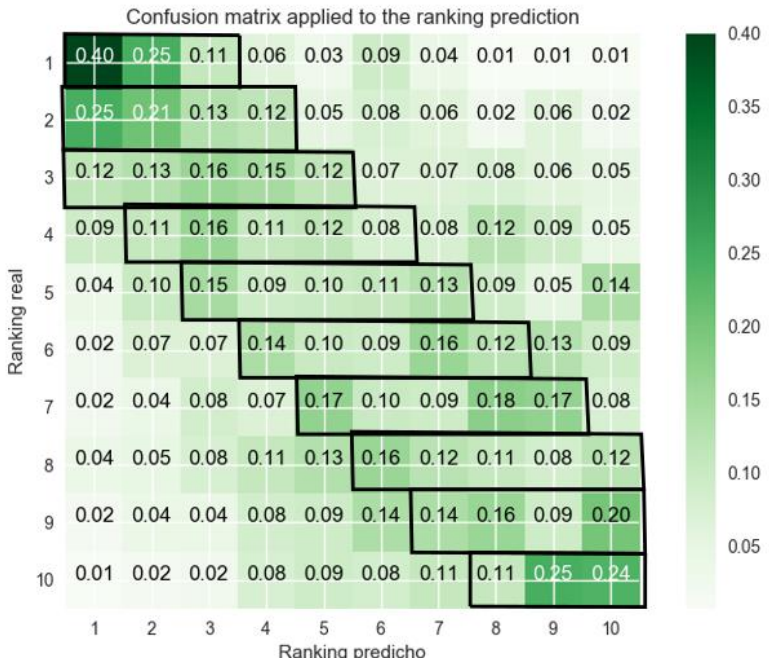


Fig 6.2.3: Matriz de confusión aplicada a la predicción del ranking (zonas marcadas)

En la gráfica se resaltan 10 zonas marcadas, cada una indica el rango de aciertos correspondiente a su valor del ranking real. Por ejemplo, veamos el 5 del ranking real (fila 5). La precisión estricta es 10%; la no estricta consiste en el sumatorio de los porcentajes dentro del rango, que da 58%.

Revisemos de nuevo los resultados mediante el Cuadro 6.2.4:

Valor del ranking	1	2	3	4	5	6	7	8	9	10
% de aciertos (E=2)	76%	71%	68%	58%	58%	61%	61%	59%	59%	60%

Cuadro 6.2.4: Porcentajes de aciertos de 1 a 10, permitiendo un error E=2

Con la E=2, los porcentajes de aciertos de la predicción han mejorado conjuntamente, por una media de 50%. Esto quiere decir que el recomendador es bueno a la hora de predecir valores cercanos respecto al valor real de ranking, aunque presenta una carencia en poder predecir con la exactitud. Según la Fig 6.2.3, también se observa que los valores lejanos respecto al real, es decir, aquéllos que dan un error elevado (de 5 arriba, están ubicados en la esquina inferior izquierda y superior derecha de la

matriz), suelen tener un porcentaje muy bajo, eso implica que el predictor comete errores graves con una probabilidad muy baja.

6.3. Experimento 2

6.3.1. Descripción

En este experimento, vamos a responder las cuatro primeras preguntas planteadas al inicio, que consisten en comparar diferentes formas de predecir. Se ilustra la diferencia con los cuadros y diagramas de barras, y se usan las métricas MAE, STD, Acc , Acc' (E=2) y CV time (tiempo usado en validación cruzada de 10 iteraciones) para evaluar.

Hacemos un recordatorio de las preguntas:

1. **Predicción de ranking mejora si se utilizan datos discretos para entrenar el predictor?**
2. **Comparativa del rendimiento entre las diferentes maneras de pre-procesamiento de datos.**
3. **Al excluir los alumnos outliers de los datos, se mejora la precisión de predicción?**
4. **Comparativa del rendimiento entre los modelos de predicción: RF y RCE.**

6.3.2. Gráficas y análisis

6.3.2.1. Pregunta 1

Selección de los casos:

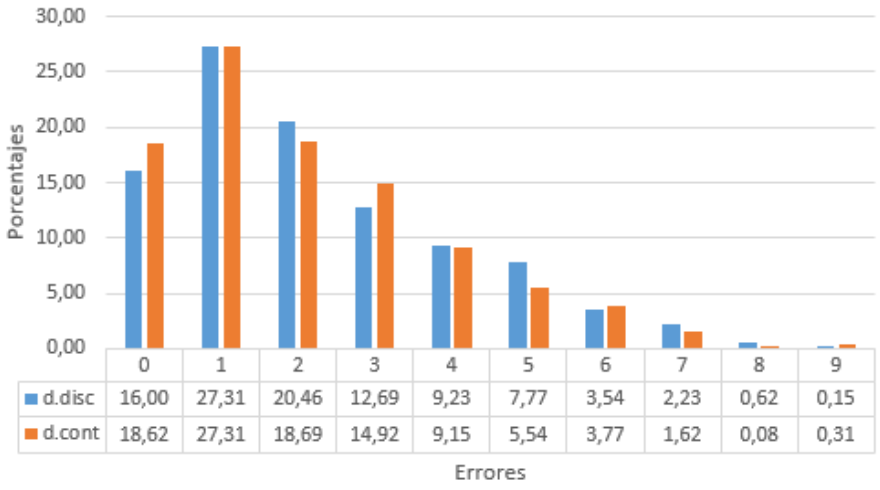
a. Matemáticas, eliminación, RF, **datos discretos**
b. Matemáticas, eliminación, RF, **datos continuos**

Las predicciones del caso "a" y del caso "b" dan el siguiente resultado. Véase el cuadro 6.3.2.1 que muestra el porcentaje de los errores:

Error	0	1	2	3	4	5	6	7	8	9
d.discretos	16%	27.31%	20.46%	12.69%	9.23%	7.77%	3.54%	2.23%	0.62%	0.15%
d.continuos	18.62%	27.31%	18.69%	14.92%	9.15%	5.54%	3.77%	1.62%	0.08%	0.31%

Cuadro 6.3.2.1: Porcentaje para cada valor de los errores (datos discretos y datos continuos)

Los resultados de los dos casos son muy parecidos. Vamos a verlos también en la gráfica Fig 6.3.2.2:



	0	1	2	3	4	5	6	7	8	9
d.disc	16,00	27,31	20,46	12,69	9,23	7,77	3,54	2,23	0,62	0,15
d.cont	18,62	27,31	18,69	14,92	9,15	5,54	3,77	1,62	0,08	0,31

Fig 6.3.2.2: Diagrama de barras que presenta el porcentaje para cada valor de los errores (datos discretos y continuos)

Los valores de las medidas obtenidos son:

Medidas	MAE	STD	Acc	Acc'	CV time
d.disc	2.25	1.85	16%	64%	0.33s
d.cont	2.11	1.78	19%	65%	0.37s

La respuesta a la pregunta 1 es: la predicción del ranking con datos discretos no mejora respecto con datos continuos, porque los resultados son parecidos, aunque el primero es un poquito peor en la precisión.

6.3.2.2. Pregunta 2

Siguiendo con la pregunta 2, seleccionamos los siguientes casos:

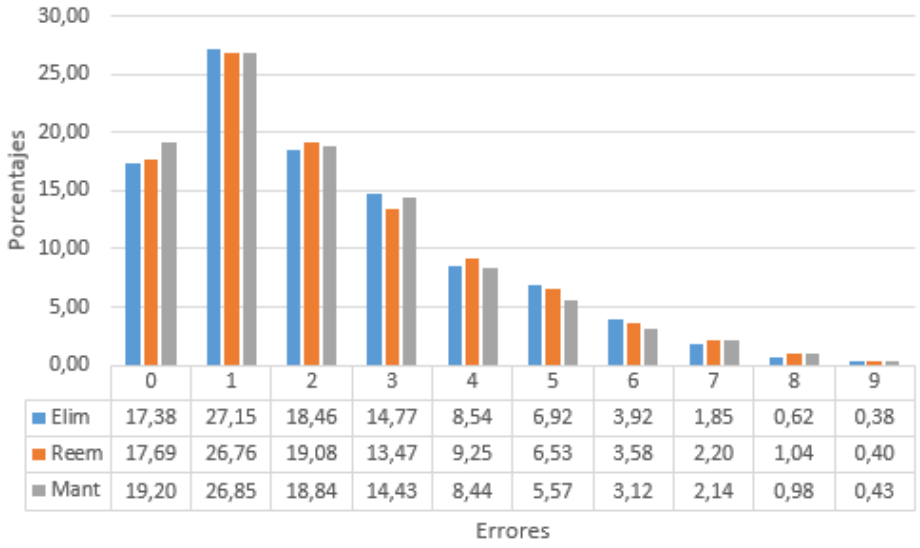
a. Matemáticas, **eliminación**, RCE, datos continuos
b. Matemáticas, **reemplazo**, RCE, datos continuos
c. Matemáticas, **mantemiento**, RCE, datos continuos

Las predicciones de los tres casos dan el siguiente resultado. Véase el Cuadro 6.3.2.3 que muestra el porcentaje de los errores:

Error	0	1	2	3	4	5	6	7	8	9
eliminación	17.38%	27.15%	18.46%	14.77%	8.54%	6.92%	3.92%	1.85%	0.62%	0.38%
reemplazo	17.69%	26.76%	19.08%	13.47%	9.25%	6.53%	3.58%	2.20%	1.04%	0.40%
mantenimiento	19.20%	26.85%	18.84%	14.43%	8.44%	5.57%	3.12%	2.14%	0.98%	0.43%

Cuadro 6.3.2.3: Porcentaje para cada valor de los errores (eliminación, reemplazo y mantenimiento de MV)

Vamos a visualizar la estadística del cuadro anterior con el diagrama de barras. Véase la Fig 6.3.2.4:



	0	1	2	3	4	5	6	7	8	9
Elim	17,38	27,15	18,46	14,77	8,54	6,92	3,92	1,85	0,62	0,38
Reem	17,69	26,76	19,08	13,47	9,25	6,53	3,58	2,20	1,04	0,40
Mant	19,20	26,85	18,84	14,43	8,44	5,57	3,12	2,14	0,98	0,43

Fig 6.3.2.4: Diagrama de barras que presenta el porcentaje de los errores (tres maneras de pre-procesamiento de MV)

Los valores de las medidas obtenidos son:

Medidas	MAE	STD	Acc	Acc'	CV time	RMV time
Elimimación	2.22	1.86	17%	63%	32.41s	-
Reemplazo	2.24	1.91	18%	64%	56.29s	16s
mantenimiento	2.14	1.87	19%	65%	62.93s	-

La respuesta de la pregunta 2 es: la eliminación y el reemplazo de MV cuentan con una precisión parecida. La manera del mantenimiento de MV presenta una precisión ligeramente mejor que las dos primeras. En cuanto al coste temporal, la eliminación es la más ágil, ya que ha reducido la dimensión de los datos desde el principio; el reemplazo lleva un coste adicional (rmv time), porque requiere un proceso de recomendación previo para completar los missing values, por lo tanto, es la más lenta; el mantenimiento requiere un poco más de tiempo en CV que los demás, por haber aplicado varios filtros para missing values.

6.3.2.3. Pregunta 3

Siguiendo con la pregunta 3, seleccionamos los siguientes casos:

a. Matemáticas, reemplazo, RCE, datos continuos (Exclusión de alumnos outliers)
b. Matemáticas, reemplazo, RCE, datos continuos (No exclusión)

Las predicciones de los dos casos dan el siguiente resultado. Véase el Cuadro 6.3.2.5 que muestra el porcentaje de los errores:

Error	0	1	2	3	4	5	6	7	8	9
Exclusión	18.67%	25.61%	20.06%	13.18%	8.73%	7.75%	3.18%	1.91%	0.81%	0.12%
No exclusión	16.22%	23.58%	18.17%	14.15%	9.84%	7.03%	5.04%	3.82%	1.54%	0.61%

Cuadro 6.3.2.5: Porcentaje de los errores (Exclusión y no exclusión de los alumnos outliers)

Vamos a visualizar la estadística del cuadro anterior con el diagrama de barras. Véase la Fig 6.3.2.6:

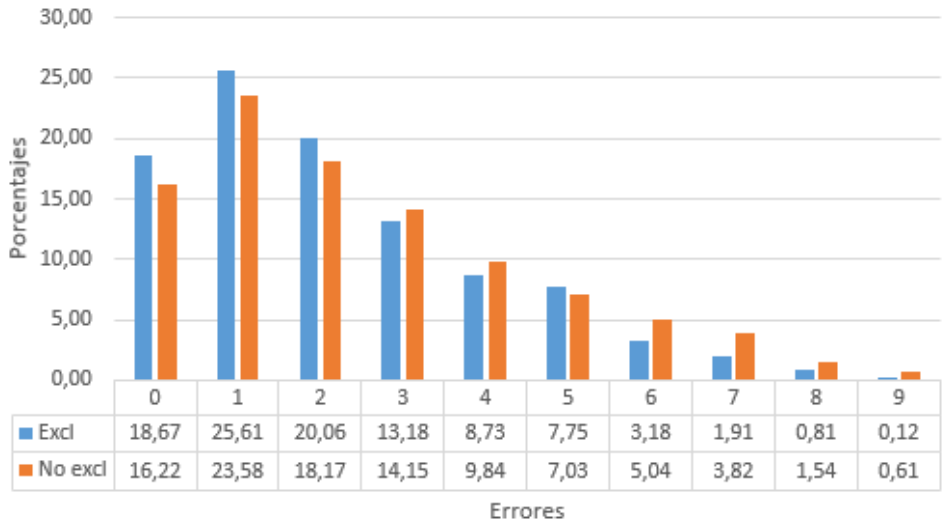


Fig 6.3.2.6: Diagrama de barras que presenta el porcentaje de los errores (Exclusión y no exclusión de los alumnos outliers)

Los valores de las medidas obtenidos son:

Medidas	MAE	STD	Acc	Acc'	CV time	RMV time
Exclusión	2.24	1.87	18%	64%	56.29s	16s
No exclusión	2.51	2.09	16%	58%	114s	45s

La respuesta a la pregunta 3 es: como se ha previsto, al excluir los alumnos outliers de los datos, la precisión de la predicción mejora bastante.

6.3.2.4. Pregunta 4

Siguiendo con la pregunta 4, seleccionamos los siguientes casos:

a. Matemáticas, eliminación, **RF**, datos discretos
b. Matemáticas, eliminación, **RCE**, datos discretos

Las predicciones de los dos casos dan el siguiente resultado. Véase el Cuadro 6.3.2.7 que muestra el porcentaje de los errores:

Error	0	1	2	3	4	5	6	7	8	9
RF	16%	27.31%	20.46%	12.69%	9.23%	7.77%	3.54%	2.23%	0.62%	0.15%
RCE	16.85%	24.92%	20.38%	14.38%	9.08%	7.46%	4.23%	2.00%	0.69%	0%

Cuadro 6.3.2.7: Porcentaje de los errores (RF y RCE)

Vamos a visualizar la estadística del cuadro anterior con el diagrama de barras. Véase la Fig 6.3.2.8:

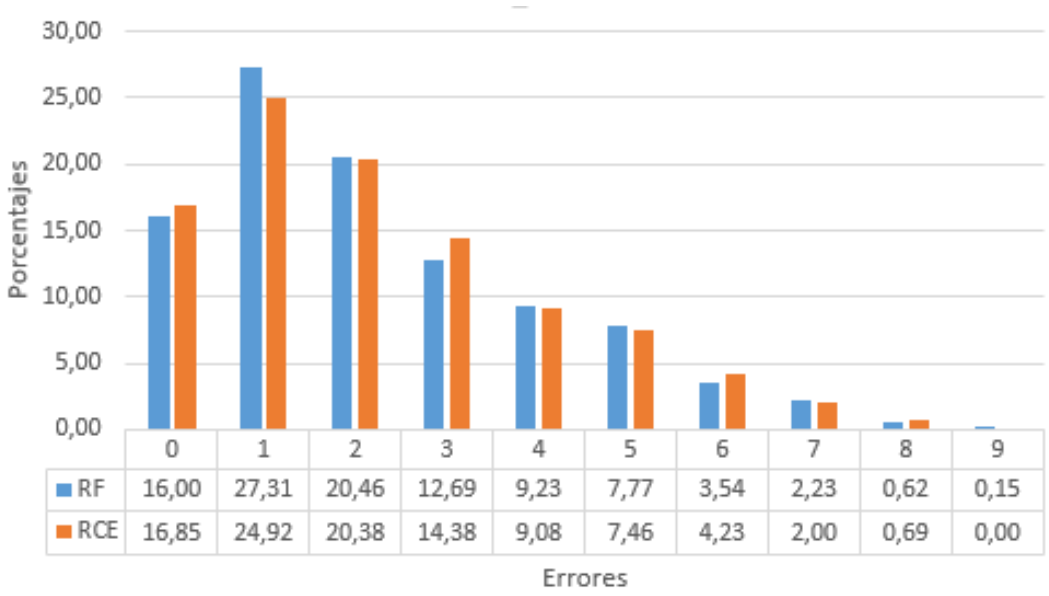


Fig 6.3.2.8: Diagrama de barras que presenta el porcentaje de los errores (RF y RCE)

Los valores de las medidas obtenidos son:

Medidas	MAE	STD	Acc	Acc'	CV time
RFR	2.25	1.85	16%	64%	0.33s
RCE	2.27	1.83	17%	62%	34.61s

La respuesta a la pregunta 4 es: aunque el RF y el RCE usan diferentes algoritmos para computar el resultado de predicciones, presentan una precisión parecida. En cuanto al coste temporal, el RF supone una velocidad computacional mucho mayor que el RCE.

6.4. Experimento 3

En este experimento, vamos a realizar las pruebas sólo para el grado de Matemáticas y asumimos los tres resultados mejores para las pruebas de Informática y Derecho, posteriormente.

6.4.1. Matemáticas

A continuación, se muestran conjuntamente los resultados relacionados con el grado de Matemáticas, en el Cuadro 6.4.1 y el 6.4.2:

Predictores\Métricas		MAE	STD	Acc	Acc'	CV time
RFR	*1. eliminar*	2.25	1.85	0.16	0.64	0.33s
	2. reemplazar	2.19	1.85	0.19	0.64	0.37s
RCE	*1. eliminar*	2.27	1.83	0.17	0.62	34.61s
	2. reemplazar	2.20	1.88	0.18	0.65	59.77s

Cuadro 6.4.1: Predicción de ranking mediante datos discretos, Matemáticas

Predictores\Métricas		MAE	STD	Acc	Acc'	CV time
RFR	*1. eliminar*	2.11	1.78	0.19	0.65	0.57s
	2. reemplazar	2.12	1.85	0.19	0.65	0.73s
RCE	*1. eliminar*	2.22	1.86	0.17	0.63	32.41s
	2. reemplazar	2.24	1.91	0.18	0.64	56.29s
	3. mantener	2.14	1.87	0.19	0.65	62.93s

Cuadro 6.4.2: Predicción de ranking mediante datos continuos, Matemáticas

Según los cuadros anteriores, podemos ver que los resultados con RFR y RCE son similares. No obstante, se resaltan tres casos que son ligeramente mejores que los demás, y los tres pertenecen a la predicción de ranking mediante **datos continuos**.

6.4.2. Informática y Derecho

En este apartado, aplicamos los tres casos que han logrado mayor precisión al grado de Informática y Derecho. Véase el Cuadro 6.4.3 y el 6.4.4:

Predictores\Métricas		MAE	STD	Acc	Acc'	CV times
RFR	*1. eliminar*	2.45	2.04	0.16	0.61	0.42s
	2. reemplazar	2.45	2.00	0.16	0.59	0.64s
RCE	*3. mantener*	2.45	1.98	0.16	0.58	89.68s

Cuadro 6.4.3: Predicción de ranking mediante datos continuos, Informática

Predictores\Métricas		MAE	STD	Acc	Acc'	CV times
RF	*1. eliminar*	2.31	1.79	0.15	0.60	2.67s
	2. reemplazar	2.31	1.78	0.15	0.60	1.03s
RCE	*3. mantener*	2.33	1.81	0.16	0.59	517.52s

Cuadro 6.4.4: Predicción de ranking mediante datos continuos, Derecho

Según los cuadros anteriores, el grado de Informática cuenta con una precisión ligeramente peor que la del Derecho, en general.

Y si comparamos ambos grados con el de Matemáticas, podemos ver que la precisión de Matemáticas es mucho mejor.

Para visualizar las gráficas de los resultados (los tres mejores) de cada grado, véase en el capítulo de Anexo.

7. Conclusiones y trabajos futuros

7.1. Conclusiones

El objetivo principal, que era implementar la predicción de ranking mediante datos discretos (ranking), se ha cumplido. Según la comparativa, hemos observado que no ha mejorado la precisión de la predicción, respecto a datos continuos (calificaciones). La hipótesis de partida era que para predecir el ranking utilizar el ranking en sí como datos de entrada podría ser mejor. De todas formas, lo hemos verificado.

Acerca de pre-procesamiento de missing values, véase el siguiente Cuadro 7.1.1:

Pre-procesamientos	Pros	Cons
Eliminación	Fácil de implementar	Decrementa la dimensión de datos, puede producir la falta de datos.
	Rápido	
Reemplazo	Prevenir la pérdida de datos	Requiere un proceso previo de recomendación antes de predecir
	Otorgar valores válidos a los missing values	Si no se hace bien, puede influir a la eficiecia de la predicción.
Mantenimiento	La implementación es complicada (predecir y validar), por las restricciones de los missing values	Mantiene los datos en su forma original
	Ha servido sólo para la predicción con datos continuos. Aunque se trabaja duro, se podría llegar a predecir con datos discretos.	

Cuadro 7.1.1: Pros y contras de las tres maneras de pre-procesamiento de missing values

Hemos observado que al excluir los alumnos outliers de los datos, la precisión de la predicción mejora de forma considerable.

Los predictores RF y RCE cuentan con una precisión similar, pero el primero es mucho más ágil.

Este trabajo final de grado ha sido una gran ayuda para aprender cosas en el ámbito de la ciencia de datos, especialmente en el aprendizaje automático y el análisis de datos.

7.2. Trabajo futuro

Como trabajo futuro, hay que mejorar la precisión de predicción de ranking, ya que es un factor muy importante para que los tutores puedan dar consejos correctos. A pesar de que el RF es potente, éste ha dejado un margen de error superior a 2, y una desviación superior a 1.70, aproximadamente. Una de las posibilidades es,

implementar un algoritmo apropiado para el problema del ranking, así como hacer un análisis de datos más extenso sobre el tema.

Como se ha explicado al inicio, este trabajo forma parte del Proyecto de Innovación Docente, entonces hay que empezar con las siguientes fases que se enumeran a continuación:

- **Fase 4**: Desarrollo del sistema inteligente. En esta fase, se pueden incluir tareas como: implementación de la interfaz gráfica, construcción de la base de datos, desarrollo de una herramienta de testeo, etcétera.
- **Fase 5**: Evaluación del sistema inteligente. Probar el sistema y obtener las retroalimentaciones (rendimiento, seguridad, simplicidad, etcétera), para poder realizar las posibles mejoras.

Además se pueden implementar más funcionalidades que puedan servir para el trabajo de tutorización.

8. Bibliografía

[1] Presentación del Proyecto de Innovación Docente:
Página web:
a. http://pid-ub.github.io/
b. http://mid.ub.edu/webpmid/content/sistema-intel%E2%80%A2ligent-de-suport-al-tutor-d%E2%80%99estudis

[2] Python aplicado en ciencia de datos (tutorial de 15 lecciones cortas):
Vídeo Youtube: https://youtu.be/D4zuOGyytm0

[3] Primer paso al aprendizaje automático – Teoría (tutorial de 17 lecciones cortas):
Vídeo Youtube: https://youtu.be/8yE0D_62bVY

[4] Introducción al aprendizaje profundo—Aprendizaje automático (tutorial-primeras 20 lecciones):
Vídeo Youtube: https://youtu.be/kjhiXQfaFeo

[5] Código del trabajo previo (público):
Página web: https://github.com/pid-ub/pid-UB

[6] Trabajar con los Missing Values:
Página web:
https://pandas.pydata.org/pandas-docs/stable/missing_data.html

[7] La clase RandomForestRegressor en la librería Scikit-Learn:
Página web: http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

[8] Jordi Vitrià, Sistema de recomendación-Taller de Nous Usos de la Informática, curso 2011-2012.

[9] Correlación de Pearson en Python:
Página web: https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.pearsonr.html

[10] Cross Validation:
Página web: http://scikit-learn.org/stable/modules/cross_validation.html

[11] Plataforma de Github:
Página web: http://pid-ub.github.io/

[12] Lenguaje de programación Python:
Página web: https://www.python.org/

[13] Biblioteca informática—Pandas:
Página web: https://pandas.pydata.org/

[14] Biblioteca informática—Numpy:
Página web: http://www.numpy.org/

[15] Biblioteca informática--Scikit Learn Documentation:
Página web: http://scikit-learn.org/stable/index.html

[16] Biblioteca informática—Matplotlib:
Página web: https://matplotlib.org/

[17] Anaconda Python—Inicio:
Página web: https://www.anaconda.com/what-is-anaconda/

[18] Microsoft Office—Inicio:
Página web: https://www.microsoft.com/es-es/

[19] Entorno de programación Python: Plataforma de Eclipse-Pydev:
Página web: http://www.pydev.org/index.html

[20] Plataforma de Stack Overflow:
Página web: https://stackoverflow.com/

[21] Matriz de la confusión:
a. https://en.wikipedia.org/wiki/Confusion_matrix
b. http://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html#sphx-glr-auto-examples-model-selection-plot-confusion-matrix-py

[22] Operaciones de agrupación en Pandas:
https://chrisalbon.com/python/data_wrangling/pandas_apply_operations_to_groups/

[23] Dibujar las gráficas circulares en Python:
https://pythonspot.com/matplotlib-pie-chart/

9. Anexo:

A continuación, se muestran las gráficas que corresponden a los tres mejores casos de cada grado. Los tres mejores casos son:

1. ***Datos continuos, RF, eliminación***
2. ***Datos continuos, Rf, reemplazo***
3. ***Datos continuos, RCE, mantenimiento***

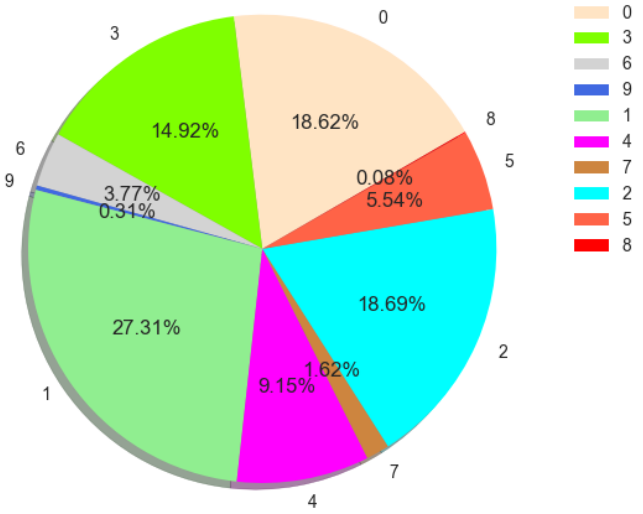


Fig 9.1: Gráfica circular que muestra los porcentajes de los errores, el caso 1 de Matemáticas

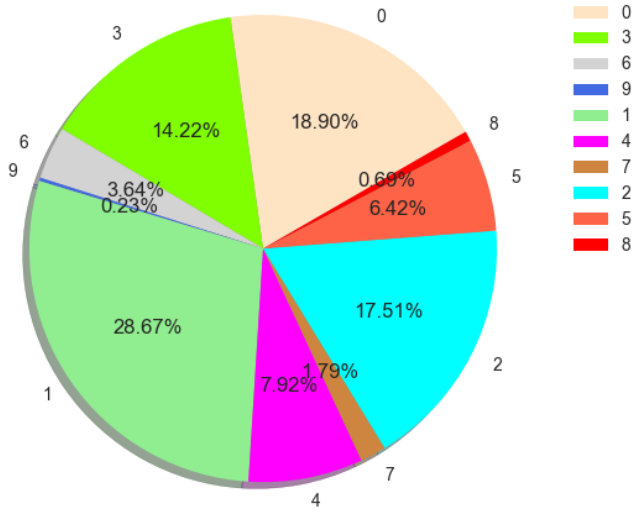


Fig 9.2: Gráfica circular que muestra los porcentajes de los errores, el caso 2 de Matemáticas

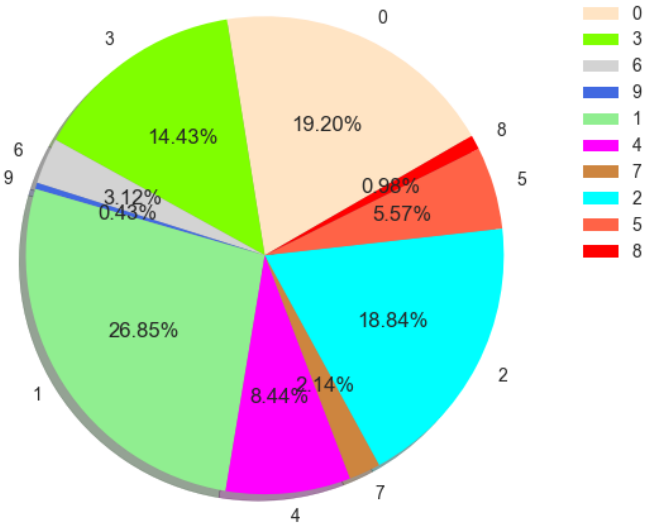


Fig 9.3: Gráfica circular que muestra los porcentajes de los errores, el caso 3 de Matemáticas

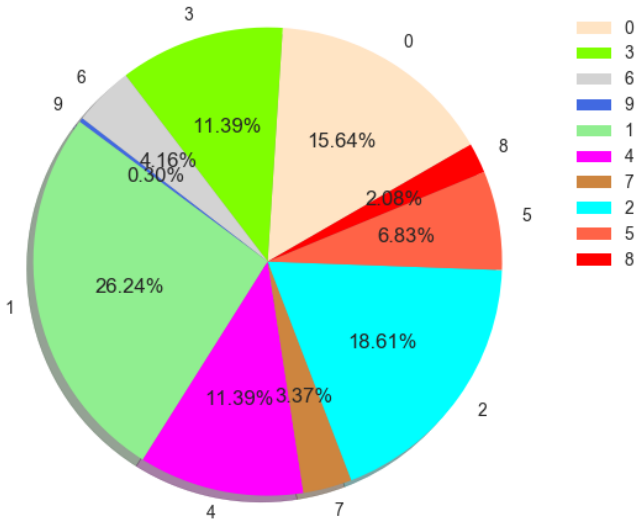


Fig 9.4: Gráfica circular que muestra los porcentajes de los errores, el caso 1 de Informática

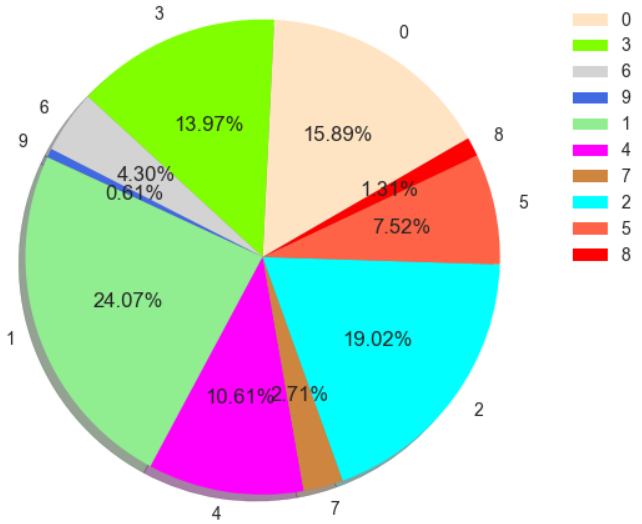


Fig 9.5: Gráfica circular que muestra los porcentajes de los errores, el caso 2 de Informática

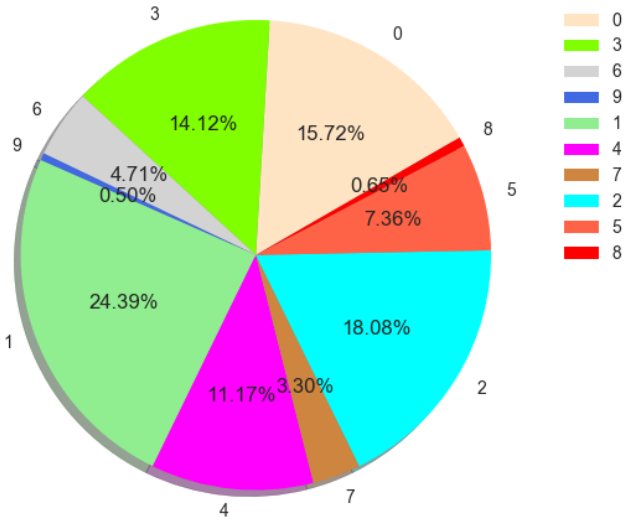


Fig 9.6: Gráfica circular que muestra los porcentajes de los errores, el caso 3 de Informática

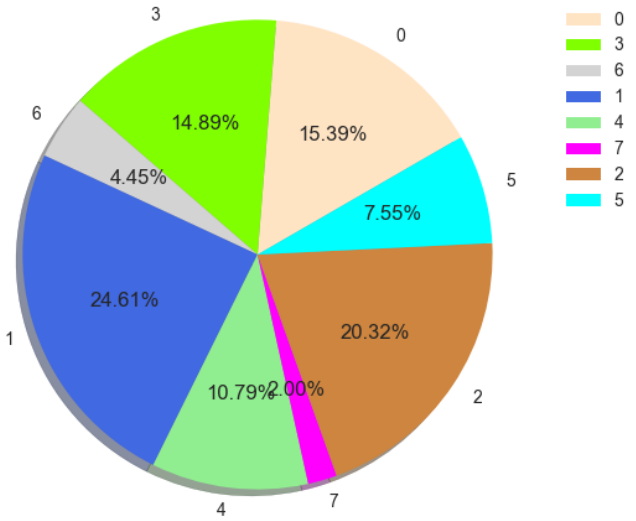


Fig 9.7: Gráfica circular que muestra los porcentajes de los errores, el caso 1 de Derecho

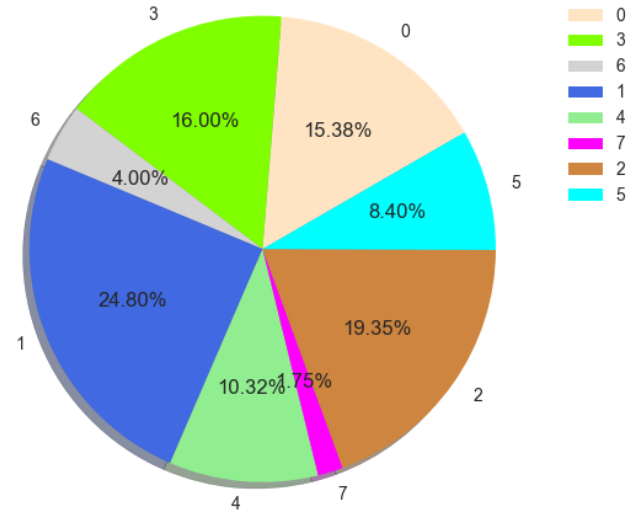


Fig 9.8: Gráfica circular que muestra los porcentajes de los errores, el caso 2 de Derecho

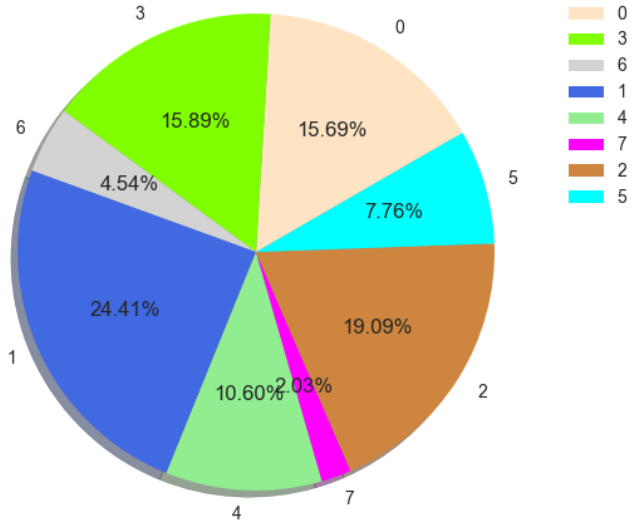


Fig 9.9: Gráfica circular que muestra los porcentajes de los errores, el caso 3 de Derecho

NOTA: Para evitar el solapamiento de la información (porcentajes) en la gráfica, la distribución de los errores no está ordenada.