Universidad de Valladolid

FACULTAD DE CIENCIAS

TRABAJO FIN DE GRADO

Grado en Estadística

Estudio y visualización de datos abiertos de estudiantes de nuevo ingreso en la Universidad de Valladolid

Autor: David Aparicio Sanz

Tutora: M. Pilar Rodríguez del Tío

2024

Resumen

El propósito de este estudio es analizar los datos abiertos de estudiantes de nuevo ingreso en la Universidad de Valladolid, abarcando desde el curso académico 2017-18 hasta el 2022-23.

Con el objetivo de identificar las características que afectan al proceso de ingreso a una titulación, se ha diseñado un enfoque dividido en dos etapas. En la primera etapa, nos centramos en determinar las características más relevantes que influyen en la elección de la rama de enseñanza académica. En la segunda etapa, nos enfocamos en identificar las titulaciones que muestran una alta coincidencia con las características de un estudiante.

Se ha desarrollado un *Dashboard* para la visualización de los datos utilizados. Esta herramienta permite representar el número de estudiantes según diversas categorías, realizar comparativas de género, conocer las notas de admisión de las diferentes titulaciones y llevar a cabo diversos análisis relacionados con las ramas de enseñanza académicas. Estas visualizaciones nos han llevado a formular una serie de hipótesis que han sido evaluadas mediante contrastes.

A lo largo del trabajo se han empleado diversas metodologías, como *Gradient Boosting*, redes neuronales frecuentistas y bayesianas, implementadas mediante el lenguaje de programación *Python*.

Palabras clave
Estudiantes de nuevo ingreso, Universidad de Valladolid, *Gradient Boosting*, redes neuronales frecuentistas y bayesianas, *Dashboard*, contrastes de hipótesis.

Abstract

The purpose of this study is to analyse the open source data of incoming students at the University of Valladolid, covering the academic year 2017-18 to 2022-23.

In order to identify the features that affect the process of entering a degree, a two-stage approach has been designed. In the first stage, we focus on determining the most relevant features that influence the choice of the academic education branch. In the second stage, we focus on identifying the degrees that show a high match with a student's features.

A Dashboard has been developed to visualise the used data. This tool allows us to represent the number of students according to some categories, to make gender comparisons, to know the admission grades of the different degrees and to carry out some analyses related to the study modalities. These visualisations have led us to formulate some hypotheses that have been tested by contrasts.

Throughout the work, several methodologies have been used, such as Gradient Boosting, frequentist and bayesian neural networks, implemented using the programming language Python.

Key words
New incoming students, University of Valladolid, Gradient Boosting, frequentist and bayesian neural networks, Dashboard, hypothesis testing.

Agradecimientos

A los profesores del Grado en Estadística, quienes me han enseñado conocimientos de esta disciplina y me han transmitido interés por el análisis de datos y la visualización.

A Pilar, por su gran implicación durante el desarrollo de este trabajo. Su dedicación y paciencia han sido fundamentales.

A mi familia, por inculcarme valores como la perseverancia y la resiliencia, y por su constante apoyo, paciencia y comprensión a lo largo de este camino.

Índice general

1. Introducción **1**

1.1. Estructura de la memoria 1

1.2. Objetivos 2

1.2.1. Sistema de información para la elección de estudios universitarios de la UVa 2

1.2.2. Creación de un *Dashboard* 2

1.3. Herramientas utilizadas 3

2. Marco teórico **4**

2.1. Gradient Boosting 4

2.1.1. Árboles de decisión 4

2.1.2. Ensambles de clasificadores 6

2.1.3. Formulación 7

2.2. Enfoques estadísticos 8

2.2.1. Frecuentista 8

2.2.2. Bayesiano 8

2.2.3. Comparativa entre enfoques 9

2.3. Redes neuronales bayesianas 10

2.3.1. Inferencia bayesiana 10

2.3.2. Límite inferior de evidencia 12

2.3.3. Inferencia variacional estocástica 13

2.4. Métodos de validación de resultados 13

2.4.1. Validación cruzada estratificada 14

2.4.2. División en varios subconjuntos estratificados 14

3. Datos **15**

3.1. Descripción de los datos 15

3.1.1. Acceso 15

3.1.2. Titulaciones 17

3.2. Proceso de anonimización 17

3.3. Extractor de datos . . . 18
3.4. Procesamiento de los datos . . . 19
3.4.1. Validación . . . 19
3.4.2. Unificación . . . 20
3.4.3. Codificación . . . 20
3.4.4. Escalado . . . 21

4. Sistema de información para la elección de estudios universitarios de la UVa 22
4.1. Imputación de valores faltantes . . . 22
4.2. Arquitectura del sistema . . . 24
4.2.1. Ramas de enseñanza . . . 25
4.2.2. Titulaciones . . . 34

5. Creación de un Dashboard 43
5.1. Número de estudiantes de nuevo ingreso . . . 44
5.2. Comparativas de género . . . 44
5.3. Notas de admisión . . . 45
5.4. Otras visualizaciones . . . 47

6. Conclusiones y líneas futuras 49
6.1. Conclusiones . . . 49
6.2. Líneas futuras . . . 50
6.3. Limitaciones del estudio . . . 50

Bibliografía 51

Apéndices 53

A. Siglas 54
A.1. Generales . . . 54
A.2. Ramas de enseñanza . . . 54
A.3. Titulaciones de Artes y Humanidades . . . 55
A.4. Titulaciones de Ciencias . . . 55
A.5. Titulaciones de Ciencias Sociales y Jurídicas . . . 56
A.6. Titulaciones de Ciencias de la Salud . . . 57
A.7. Titulaciones de Ingeniería y Arquitectura . . . 57

B. Arquitecturas bayesianas **59**
B.1. Artes y Humanidades . . . 59
B.2. Ciencias, Ingeniería y Arquitectura . . . 60
B.3. Ciencias Sociales y Jurídicas . . . 61
B.4. Ciencias de la Salud . . . 62

C. Imágenes del *Dashboard* **63**
C.1. Inicio . . . 63
C.2. Número de estudiantes de nuevo ingreso . . . 64
C.3. Comparativas de género . . . 64
C.4. Notas de admisión . . . 65
C.5. Otras visualizaciones . . . 66

D. Contrastes de hipótesis **68**
D.1. Test de proporciones . . . 68
D.2. Test de rangos de *Wilcoxon Mann Whitney* . . . 71

E. Código **74**
E.1. Bibliotecas . . . 74
E.2. Extractor de datos . . . 74
E.3. Imputación de valores faltantes . . . 75
E.4. Creación de una red neuronal frecuentista . . . 76
E.5. Creación de una red neuronal bayesiana . . . 78
E.6. Contrastes de hipótesis . . . 80

Índice de figuras

2.1. Árbol de decisión para un problema de clasificación. 5
2.2. Ensamble *Gradient Boosting* para un problema de clasificación. 6
2.3. Estructura básica de una red neuronal bayesiana. 10
2.4. Divergencia de Kullback-Leibler 12
2.5. Ejemplo de validación cruzada. 14
2.6. Ejemplo de división en tres subconjuntos. 14

3.1. Arquitectura del extractor de datos. 19
3.2. Ejemplo de codificación *One-Hot* estándar. 20
3.3. Ejemplo de codificación *One-Hot* para las variables con valores faltantes. 21

4.1. Arquitectura del sistema. 24
4.2. Resumen paramétrico de la red neuronal frecuentista. 27
4.3. Función de pérdida versus época. 28
4.4. Tasa de aciertos unaria versus época. 28
4.5. Tasa de aciertos binaria versus época. 28
4.6. Matriz de confusión de la asignación binaria sobre el conjunto de prueba. 30
4.7. Top 9 variables globalmente más influyentes. 31
4.8. Top 6 variables más influyentes para Artes y Humanidades. 32
4.9. Top 6 variables más influyentes para Ciencias, Ingeniería y Arquitectura. 33
4.10. Top 6 variables más influyentes para Ciencias Sociales y Jurídicas. 33
4.11. Top 6 variables más influyentes para Ciencias de la Salud. 34
4.12. Puntuaciones asociadas a cada titulación para un estudiante del Grado en Lenguas Modernas y sus Literaturas. 37
4.13. Puntuaciones asociadas a cada titulación para un estudiante del Grado en Estadística. 38
4.14. Puntuaciones asociadas a cada titulación para un estudiante del Grado en Educación Primaria. 40
4.15. Puntuaciones asociadas a cada titulación para un estudiante del Grado en Medicina. 41

B.1. Resumen paramétrico de la red neuronal bayesiana de Artes y Humanidades. . . 59
B.2. Resumen paramétrico de la red neuronal bayesiana de Ciencias, Ingeniería y Arquitectura. 60
B.3. Resumen paramétrico de la red neuronal bayesiana de Ciencias Sociales y Jurídicas. 61
B.4. Resumen paramétrico de la red neuronal bayesiana de Ciencias de la Salud. . . . 62

C.1. Inicio. 63
C.2. Número de estudiantes de nuevo ingreso. 64
C.3. Comparativas de género. 64
C.4. Notas de admisión I. 65
C.5. Notas de admisión II. 65
C.6. Otras visualizaciones I. 66
C.7. Otras visualizaciones II. 66
C.8. Otras visualizaciones III. 67
C.9. Otras visualizaciones IV. 67

Índice de tablas

2.1. Comparación de características entre enfoques estadísticos. 9

3.1. Variables del conjunto de datos “Acceso” utilizadas en el trabajo. 16
3.2. Variables del conjunto de datos “Titulaciones” utilizadas en el trabajo. 17

4.1. Comparación del modelo de *Gradient Boosting* inicial con el óptimo. 23
4.2. División del conjunto de datos en la primera etapa. 25
4.3. Hiperparámetros óptimos de la red neuronal frecuentista. 26
4.4. Tasa de aciertos de las asignaciones unarias y binarias. 29
4.5. Tasa de aciertos unaria desglosada por rama de enseñanza académica. 29
4.6. Tasa de aciertos binaria desglosada por rama de enseñanza académica. 29
4.7. Número de titulaciones asociadas a cada modelo. 34
4.8. Hiperparámetros óptimos comunes de las redes neuronales bayesianas. 35
4.9. Hiperparámetros óptimos no comunes de las redes neuronales bayesianas. 36
4.10. Coincidencia asociada a cada titulación para un estudiante del Grado en Lenguas Modernas y sus Literaturas. 37
4.11. Coincidencia asociada a las titulaciones que superan el umbral de decisión para un estudiante del Grado en Lenguas Modernas y sus Literaturas. 38
4.12. Coincidencia asociada a cada titulación para un estudiante del Grado en Estadística. 39
4.13. Coincidencia asociada a las titulaciones que superan el umbral de decisión para un estudiante del Grado en Estadística. 39
4.14. Coincidencia asociada a cada titulación para un estudiante del Grado en Educación Primaria. 40
4.15. Coincidencia asociada a las titulaciones que superan el umbral de decisión para un estudiante del Grado en Educación Primaria. 41
4.16. Coincidencia asociada a cada titulación para un estudiante del Grado en Medicina. 41
4.17. Coincidencia asociada a las titulaciones que superan el umbral de decisión para un estudiante del Grado en Medicina. 42
4.18. Tasas de aciertos de los modelos especializados sobre el conjunto de prueba. . . . 42

D.1. Titulaciones en las que no se rechaza la hipótesis nula de igualdad de proporciones de género. 68
D.2. Titulaciones en las que se rechaza la hipótesis nula de igualdad de proporciones de género y hay más hombres que mujeres. 69
D.3. Titulaciones en las que se rechaza la hipótesis nula de igualdad de proporciones de género y hay más mujeres que hombres. 70
D.4. Titulaciones en las que no se confirma que haya un aumento de las notas de admisión en el curso 2020-21 mediante el contraste unilateral con el test de rangos de *Wilcoxon Mann Whitney* I. 71
D.5. Titulaciones en las que no se confirma que haya un aumento de las notas de admisión en el curso académico 2020-21 mediante el contraste unilateral con el test de rangos de *Wilcoxon Mann Whitney* II. 72
D.6. Titulaciones en las que se confirma que hay un aumento de las notas de admisión en el curso académico 2020-21 mediante el contraste unilateral con el test de rangos de *Wilcoxon Mann Whitney* I. 72
D.7. Titulaciones en las que se confirma que hay un aumento de las notas de admisión en el curso académico 2020-21 mediante el contraste unilateral con el test de rangos de *Wilcoxon Mann Whitney* II. 73

Capítulo 1

Introducción

El presente Trabajo Fin de Grado se basa en datos abiertos [1] del proyecto colaborativo UniversiDATA, una iniciativa conjunta de universidades públicas españolas promovida por la empresa Dimetrical. Este portal de datos tiene como objetivo principal fomentar el acceso a datos abiertos en el ámbito de la educación superior en España, facilitando su uso y aprovechamiento tanto por parte de las instituciones educativas como de la sociedad en general.

En este contexto, se acumulan conjuntos de datos de gran tamaño y complejidad, dado que constantemente se añaden nuevos datos, se actualizan los existentes para añadir información correspondiente a nuevos periodos de tiempo o se incorporan nuevas universidades al proyecto. Esta dinámica de crecimiento continuo no solo implica la acumulación de datos a gran escala, sino también la necesidad de gestionar, analizar y visualizar esta información de manera eficiente y automática.

Este trabajo es continuación de un proyecto [2] presentado en el año 2022 para el I *Datathon* UniversiDATA. El trabajo "*Dashboard* sobre el acceso de nuevos estudiantes a universidades españolas" obtuvo el segundo premio. Esta experiencia previa ha servido como punto de partida para seguir explorando el potencial del análisis de datos en el ámbito de la educación superior.

1.1. Estructura de la memoria

Este trabajo está dividido en varios capítulos para presentar la información de manera ordenada y estructurada. En primer lugar, se presentan los objetivos a abordar, destacando su relevancia e interés. A continuación, se detallan los métodos estadísticos empleados y las técnicas de validación de resultados. Luego, se describen los conjuntos de datos utilizados, se explica la forma en la que se accede a los datos en tiempo real y se detalla el procesamiento necesario.

Finalmente, se utiliza esta información para desarrollar un sistema de información para la elección de estudios universitarios de la Universidad de Valladolid, un *Dashboard* en el que se pueden visualizar los datos empleados y varios contrastes de hipótesis.

1.2. Objetivos

El objetivo principal de este trabajo es el estudio y la visualización de datos abiertos de estudiantes de nuevo ingreso en la Universidad de Valladolid. Teniendo presente este propósito global, se abordan los siguientes objetivos específicos:

1.2.1. Sistema de información para la elección de estudios universitarios de la UVa

El primer objetivo consiste en desarrollar un sistema de información para la elección de estudios universitarios ofrecidos por la Universidad de Valladolid. El análisis y la creación de modelos estadísticos con estos datos sirven para entender qué características de los usuarios influyen en esta elección. Es importante destacar que no se trata de un sistema de recomendación de estudios, ya que la decisión de elegir una titulación está influenciada por la subjetividad en las decisiones de los estudiantes y se ve limitada por la falta de datos sobre la finalización o el abandono de los programas académicos por parte de los estudiantes.

1.2.2. Creación de un *Dashboard*

El segundo objetivo consiste en crear una aplicación en línea que ponga a disposición de los usuarios una serie de herramientas de visualización de datos, permitiendo así explorar y comprender mejor los datos de estudiantes de nuevo ingreso en la Universidad de Valladolid. Se dispone de la aplicación en https://david-aparicio-sanz-tfg-estadistica-2024.streamlit.app/.

El *Dashboard* debe ser intuitivo y accesible, con gráficos interactivos y dinámicos que faciliten la interpretación de la información. Los usuarios podrán seleccionar y comparar diferentes categorías y variables, lo que les permitirá identificar patrones y tendencias. Además, se podrán activar o desactivar ciertas opciones para una visualización más clara.

A partir de estas visualizaciones, hemos formulado varias hipótesis que hemos evaluado utilizando contrastes de hipótesis. Por ejemplo, se analiza si existen diferencias significativas en la distribución de estudiantes según el género o si las notas de admisión han experimentado un aumento significativo debido a los cambios motivados por la pandemia en las pruebas de acceso del curso académico 2020-21.

1.3. Herramientas utilizadas

En esta sección, se presentan las tecnologías utilizadas en el desarrollo de este proyecto, las cuales incluyen un lenguaje de programación, una plataforma de código abierto y varias bibliotecas especializadas:

- **Python [3].** Lenguaje de programación de código abierto ampliamente utilizado en el ámbito de la ciencia de datos y el aprendizaje automático debido a su versatilidad y a la amplia variedad de bibliotecas y herramientas con las que cuenta.
- **PyCharm [4].** Entorno de desarrollo integrado para el desarrollo de aplicaciones con el lenguaje de programación Python.
- **Jupyter Notebook [5].** Plataforma de código abierto que ofrece un entorno interactivo para escribir y ejecutar código en múltiples lenguajes de programación.
- **Pandas [6].** Biblioteca dedicada a la manipulación de datos gracias a estructuras de datos flexibles y eficientes, como *DataFrames* y *Series*, que facilitan esta tarea.
- **PyTorch [7].** Biblioteca de código abierto diseñada para la creación y entrenamiento de modelos de aprendizaje profundo o *Deep Learning*, así como su despliegue.
- **Pyro [8].** Biblioteca de programación probabilística de código abierto que permite la especificación y manipulación de modelos probabilísticos complejos de una manera flexible y escalable.
- **Highcharts [9].** Biblioteca de *JavaScript* de pago que permite crear gráficos interactivos para páginas web. Se dispone de una licencia educativa asociada a este trabajo con el identificador 28203.
- **Streamlit [10].** *Framework* de *Python* de código abierto especializado en el desarrollo de aplicaciones web para proyectos de ciencia de datos.

Capítulo 2

Marco teórico

En este capítulo se exponen los fundamentos teóricos que respaldan el sistema de información para la elección de estudios universitarios propuesto en este trabajo. Inicialmente se explica el algoritmo de aprendizaje automático *Gradient Boosting*. Seguidamente, se analizan las diferencias entre los enfoques estadísticos frecuentista y bayesiano, sentando las bases necesarias para poder entender los tipos de redes neuronales.

2.1. Gradient Boosting

Gradient Boosting [11] es el algoritmo que se utiliza en este trabajo para llevar a cabo la imputación de los valores faltantes. Es una técnica de ensamblaje que combina múltiples modelos débiles para crear un modelo predictivo más robusto y preciso. Esto implica la construcción secuencial de árboles de decisión, donde cada árbol corrige los errores de predicción del árbol anterior. En cada etapa, se enfoca en las instancias, también denominadas observaciones o casos, que fueron clasificadas incorrectamente o cuyos residuos son altos, ajustando así el modelo para mejorar la predicción de esos casos específicos.

2.1.1. Árboles de decisión

El concepto de árboles de decisión tiene su origen en 1963, cuando James Nelson Morgan y John A. Sonquist propusieron su aplicación en el artículo "Problems in the Analysis of Survey Data, and a Proposal" [12]. Hoy en día, los árboles de decisión son ampliamente reconocidos como uno de los algoritmos más populares en el ámbito del aprendizaje supervisado. Se fundamentan en un enfoque descendente que busca crear un modelo predictivo dividiendo el espacio de los predictores en grupos de observaciones con valores similares para la variable respuesta. Para lograr esta división, se aplican una serie de reglas de decisión, teniendo en

cuenta el objetivo de que cada grupo contenga la mayor cantidad posible de individuos de una misma población. En caso de que un grupo contenga datos de diferentes clases, se subdividirá en regiones más pequeñas hasta que cada una contenga datos de una única clase.

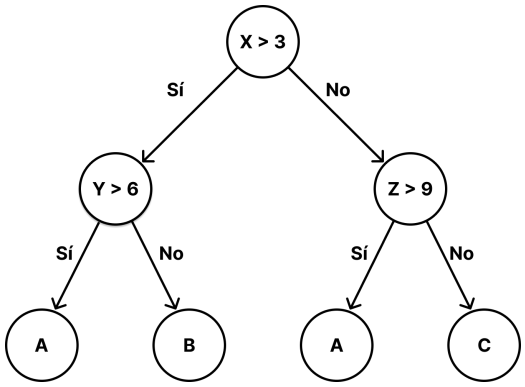

Figura 2.1: Árbol de decisión para un problema de clasificación.

En la figura 2.1 se puede observar que un árbol de decisión tiene una estructura jerárquica compuesta por nodos y conexiones, que representan una secuencia de decisiones lógicas para clasificar o predecir valores. Estos árboles constan de tres tipos de nodos:

- **Nodo raíz**: es el punto de partida del árbol y representa la primera bifurcación a partir de la variable más significativa del modelo.
- **Nodos internos**: representan las decisiones intermedias llevadas a cabo. Cada nodo interno tiene ramas que conducen a otros nodos internos o a nodos hoja.
- **Nodos hoja**: son los nodos terminales del árbol y representan predicciones.

Existen varios hiperparámetros [13] que se pueden optimizar para mejorar el rendimiento del modelo y evitar el sobreajuste. Los hiperparámetros más comunes son:

- **Profundidad máxima**: permite crecer al árbol con normalidad hasta alcanzar una profundidad máxima establecida.
- **Número de muestras**: establece criterios sobre la cantidad mínima de datos necesarios tanto para dividir un nodo como para formar un nodo hoja.
- **Criterio de división**: define la función utilizada para evaluar la calidad de una división. Los criterios más comunes son la ganancia de información, el índice de *Gini* y la entropía.
- **Regularización**: controla el crecimiento del árbol con técnicas de regularización como la poda posterior o la poda anticipada.

2.1.2. Ensambles de clasificadores

En la siguiente imagen, se ilustra un ensamblaje de *Gradient Boosting* de M clasificadores diseñado para abordar un problema de clasificación. Inicialmente, se entrena el primer árbol de decisión débil utilizando los datos disponibles. Luego, los residuos o errores de este modelo se utilizan como entrada para el segundo clasificador, enfocándose en las observaciones que no fueron correctamente clasificadas, lo que permite mejorar gradualmente la precisión del conjunto de modelos. Este proceso se repite sucesivamente en cada clasificador débil, como se puede ver en la figura 2.2, con el objetivo de mejorar la precisión general del ensamblaje.

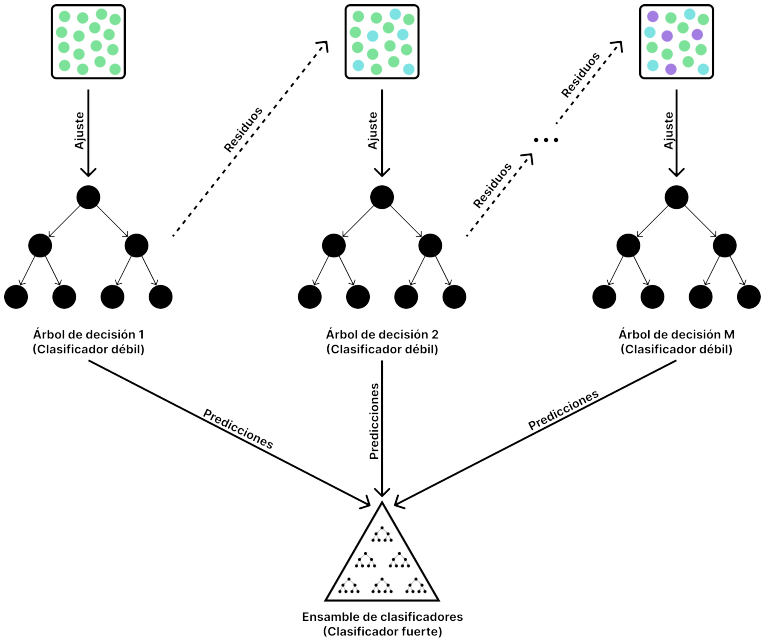

Figura 2.2: Ensamble *Gradient Boosting* para un problema de clasificación.

Es necesario optimizar los hiperparámetros del ensamblaje [14] para mejorar su rendimiento. También es necesario optimizar los cuatro hiperparámetros de los árboles de decisión, ya que afectan a la construcción de cada uno de los clasificadores débiles. Los hiperparámetros del ensamblaje considerados son:

- **Tasa de aprendizaje** (ν): regula la velocidad con la que el modelo se adapta durante el proceso de entrenamiento al actualizar sus predicciones.
- **Número de árboles del ensamble**: indica la cantidad de árboles de decisión que se utilizan en el ensamblaje. Valores demasiado altos pueden causar subreajuste.
- **Porcentaje de muestra de entrenamiento**: establece el porcentaje de datos de entrenamiento utilizados para entrenar cada árbol de decisión del ensamble. Al no emplear todos los datos para ajustar cada árbol, se mejora la capacidad de generalización.

2.1.3. Formulación

Dado el conjunto de datos de entrenamiento representado por $\{(x_i, y_i)\}_{i=1}^{n}$, donde x_i son las características y y_i son las etiquetas correspondientes, M el número de clasificadores débiles y $L(y, F(x))$ una función de pérdida [15] que sea diferenciable y mida la discrepancia entre los valores reales y las predicciones, el objetivo es encontrar una función $F(x)$ que minimice la pérdida. El algoritmo [16] consta de los siguientes pasos:

1. Inicializar un modelo base con una predicción constante F_0.

$$F_0(x) = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_i, \gamma) \tag{2.1}$$

2. Repetir los siguientes pasos para cada uno de los clasificadores débiles $m = 1, 2, \ldots, M$.

   a) Obtener los residuos para cada muestra $i = 1, 2, \ldots, n$ tomando una derivada de la función de pérdida con respecto a la predicción anterior.

$$r_{i,m} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \tag{2.2}$$

   b) Entrenar un clasificador débil $h_m(x_i)$ utilizando los residuos en lugar de las etiquetas, este nuevo conjunto se representa como $\{(x_i, r_{i,m})\}_{i=1}^{n}$. El número de nodos hoja del clasificador débil m-ésimo viene denotado por J_m.

   c) Buscar el valor de $\gamma_{j,m}$ que minimice la función de pérdida en cada nodo terminal j.

$$\gamma_{j,m} = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + \gamma \cdot h_m(x_i)) \quad j = 1, 2, \ldots, J_m \tag{2.3}$$

   d) Actualizar las predicciones del modelo a partir de las predicciones del árbol recién ajustado, ponderadas con la tasa de aprendizaje ν. De esta forma se controla la contribución de la predicción del árbol adicional a la predicción combinada $F_m(x)$.

$$F_m(x) = F_{m-1}(x) + \nu \cdot h_m(x) \sum_{j=1}^{J_m} \gamma_{j,m} \quad 0 < \nu < 1 \tag{2.4}$$

3. Obtener predicciones con el ensamblaje de modelos $F_M(x)$.

2.2. Enfoques estadísticos

Para entender los métodos que utilizamos en este trabajo es esencial comprender a fondo las diferencias [17] entre los enfoques estadísticos frecuentista y bayesiano. Cada uno proporciona una perspectiva diferente sobre cómo interpretar los resultados. En esta sección, se exploran las características principales de ambos enfoques.

2.2.1. Frecuentista

La estadística frecuentista, también conocida como estadística clásica, fue desarrollada a principios del siglo XX por los matemáticos y estadísticos Ronald Fisher, Jerzy Neyman y Egon Pearson, entre otros. Sus trabajos sentaron las bases del enfoque estadístico frecuentista, basado en la idea de que las probabilidades se pueden interpretar como frecuencias relativas, que se pueden obtener a partir de la repetición de un experimento.

En base a esto, se asume que los parámetros que se intentan estimar son fijos, de modo que solo existe un único parámetro verdadero. Los estimadores puntuales capturan la información contenida en los datos y se diseñan de manera que sean representativos, permitiendo hacer inferencias sobre una población más amplia a partir de la muestra recopilada.

2.2.2. Bayesiano

La estadística bayesiana, también conocida como estadística probabilística, fue desarrollada a mediados del siglo XVIII por Thomas Bayes, un estadístico y filósofo inglés. Sin embargo, gran parte de su trabajo recibió poca atención hasta alrededor de 1950, cuando el auge de las computadoras facilitó la realización de estadísticas bayesianas utilizando máquinas con mayor potencia informática.

Este enfoque permite incorporar conocimiento previo reflejando las creencias iniciales sobre los parámetros mediante una distribución de probabilidad, conocida como distribución a priori. El teorema de Bayes tiene unna gran importancia dado que es el mecanismo que permite actualizar las creencias iniciales a través de la incorporación de nuevos datos, lo que lleva a la obtención de la distribución a posteriori.

En este contexto, los parámetros son considerados variables latentes, es decir, variables aleatorias no observables que siguen una distribución específica, lo que permitie abordar la incertidumbre de los datos.

2.2.3. Comparativa entre enfoques

Cada enfoque trata el concepto de probabilidad [18] de forma diferente. Por ejemplo, en el contexto de un problema de clasificación binaria, el enfoque frecuentista interpreta la probabilidad de asignación como una probabilidad puntual, es decir, una estimación específica de pertenencia a cada grupo. En contraste, bajo el enfoque bayesiano, la probabilidad de asignación se concibe como una distribución, representando un rango de posibles valores de pertenencia a cada grupo. A continuación, se muestra una tabla que compara las principales características de ambos enfoques:

Característica	Enfoque frecuentista	Enfoque bayesiano
Tamaño de muestra	▪ Requiere muestras grandes.	▪ Permite trabajar con muestras pequeñas.
Tipo de parámetros	▪ Estimaciones puntuales.	▪ Variables latentes.
Toma de decisiones	▪ Decisiones rígidas.	▪ Decisiones flexibles.
Información a priori	▪ No permite incorporar información previa.	▪ Requiere información previa.
Interpretación de la probabilidad	▪ Las probabilidades representan las frecuencias.	▪ Las probabilidades representan el grado de incertidumbre.
Complejidad computacional	▪ Baja.	▪ Alta.
Control de la incertidumbre	▪ Estimaciones puntuales e intervalos de confianza.	▪ Distribución a posteriori.

Tabla 2.1: Comparación de características entre enfoques estadísticos.

2.3. Redes neuronales bayesianas

Las redes neuronales bayesianas o probabilísticas [19] usan probabilidades para representar la incertidumbre en los parámetros de la red mediante una distribución de probabilidad con parámetros de localización μ y escala σ, lo que las hace más robustas frente a los datos ruidosos.

Sin embargo, las redes de este tipo plantean un desafío en términos de coste computacional. Esto se debe a que modelar la incertidumbre implica la necesidad de utilizar numerosas muestras aleatorias de los parámetros de la red. Estas muestras son fundamentales para construir distribuciones de probabilidad de los parámetros de la red, los cuales se actualizan durante el entrenamiento para permitir que la red aprenda de la incertidumbre de los datos. Este proceso de muestreo repetido y la aplicación de la inferencia bayesiana requieren una gran cantidad de recursos computacionales, lo que ha llevado a que estas redes sean menos estudiadas en comparación con las redes frecuentistas. La figura 2.3 representa la estructura básica de una red neuronal bayesiana.

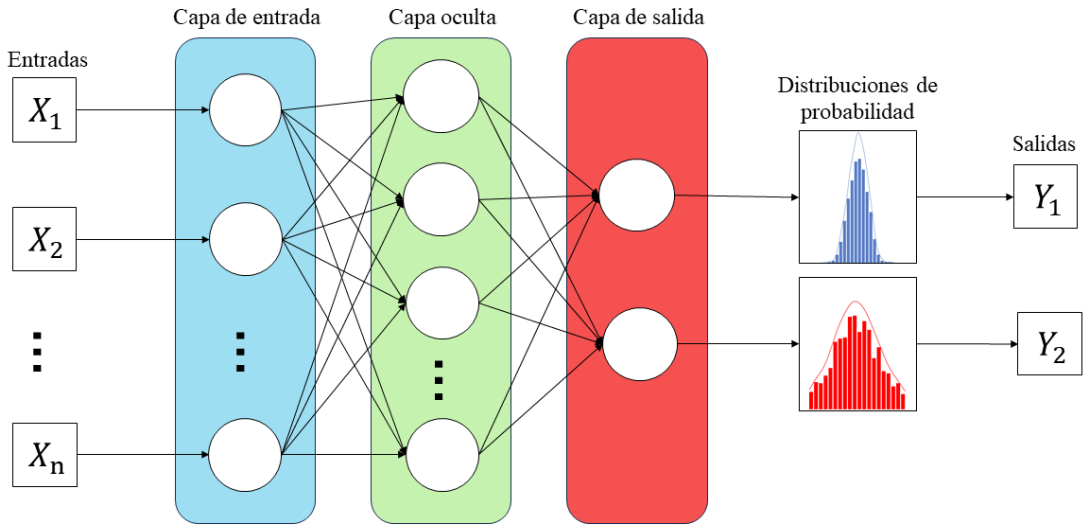

Figura 2.3: Estructura básica de una red neuronal bayesiana.

2.3.1. Inferencia bayesiana

Para construir un modelo bayesiano, expresamos todo lo que sabemos sobre las variables del problema y las relaciones entre ellas en forma de modelo probabilístico en el que el vector X representa un conjunto de variables aleatorias observables. Esta información se utiliza para entrenar el modelo y efectuar inferencias. Por otra parte, nos encontramos con el vector Z, el cual refleja un conjunto de variables aleatorias latentes que, aunque son desconocidas, se supone que ejercen una influencia notable en el proceso de generación de los datos observados X.

Suponemos que X y Z se distribuyen conjuntamente según una distribución de probabilidad parametrizada por θ y representada como $P_\theta(X, Z)$. Podemos obtener esta distribución de probabilidad conjunta [20] utilizando las variables observadas dadas las variables aleatorias latentes $p_\theta(x \mid z)$ y la distribución de probabilidad a priori de las variables latentes $p_\theta(z)$.

$$p_\theta(x, z) = p_\theta(x \mid z)\ p_\theta(z) \tag{2.5}$$

Donde se define p_θ como la función de masa de probabilidad parametrizada por θ, x como las instancias de las variables aleatorias observadas X y z como las instancias de las variables aleatorias latentes Z.

El objetivo es obtener la distribución de probabilidad a posteriori $p_\theta(z \mid x)$ que nos permita realizar inferencias sobre las variables latentes a partir de los datos observados. En una situación ideal se obtendría con el teorema de Bayes [21].

$$p_\theta(z \mid x) = \frac{p_\theta(x \mid z)\ p_\theta(z)}{p_\theta(x)} \tag{2.6}$$

En la práctica suele ser difícil calcular la distribución de probabilidad a posteriori $p_\theta(z \mid x)$ puesto que la evidencia $p_\theta(x)$ no tiene una expresión algebraica exacta, pero podemos obtener una aproximación.

$$p_\theta(x) = \int p_\theta(x, z)\ dz \tag{2.7}$$

El siguiente paso consiste en obtener la distribución de probabilidad posterior predictiva, la cual se utiliza para hacer predicciones. Su importancia está en que permite conocer la distribución de probabilidad de un evento futuro o una variable no observada a partir de la información que tenemos hasta ahora, de esta forma se cuantifica la incertidumbre de las predicciones.

$$p_\theta(x' \mid x) = \int p_\theta(x' \mid z)\ p_\theta(z|x)\ dz \tag{2.8}$$

Finalmente es importante que los parámetros del modelo aprendan de los datos observados [22] de forma que se maximize el logaritmo de la evidencia.

$$\theta_{\max} = \underset{\theta}{\operatorname{argmax}}\ \log\ p_\theta(x) = \underset{\theta}{\operatorname{argmax}}\ \log \int p_\theta(x, z)\ dz \tag{2.9}$$

2.3.2. Límite inferior de evidencia

El límite inferior de evidencia, también conocido como ELBO, es una función de pérdida fundamental en el campo de la inferencia bayesiana [23] ya que el algoritmo de inferencia variacional estocástica le necesita. Su objetivo es encontrar la aproximación óptima a la verdadera distribución a posteriori, utilizando su descomposición en la evidencia y la divergencia de Kullback-Leibler. La evidencia es la capacidad de reconstrucción de los datos observados mientras que la divergencia de KL es una medida estadística que se utiliza para cuantificar la diferencia entre dos distribuciones de probabilidad. Fue propuesta por Solomon Kullback y Richard Leibler [24] en el año 1951 como la divergencia direccionada entre dos distribuciones.

$$KL = Evidencia - ELBO \tag{2.10}$$

La utilidad de esta descomposición consiste en que si maximizamos el ELBO, la divergencia de KL se minimizará sujeta a la restricción de ser mayor o igual que cero, como se ilustra en la figura 2.4.

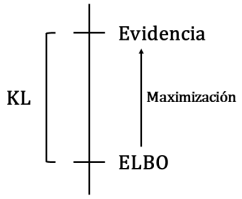

Figura 2.4: Divergencia de Kullback-Leibler

Inicialmente consideramos una distribución parametrizada $P_\theta(X, Z)$ y fijamos un valor inicial para los parámetros variacionales θ. Suponemos que Z sigue una distribución de probabiliad parametrizada por ϕ, la cual sirve de aproximación a la distribución a posteriori y la denotamos como $q_\phi(z)$.

$$KL(q_\phi(z) \,||\, p_\theta(z \mid x)) = \log p_\theta(x) - \mathbb{E}_{q_\phi(z)}\left[\log \frac{p_\theta(x, z)}{q_\phi(z)}\right] \tag{2.10}$$

La anterior descomposición resulta problemática debido a que tenemos dos términos intratables, que son $KL(q_\phi(z) \,||\, p_\theta(z \mid x))$ y $\log p_\theta(x)$. Sin embargo, si despejamos ambos términos al lado derecho de la ecuación, tenemos que maximizar la cantidad del otro lado, ya que al hacer esto, simultáneamente se maximiza la evidencia y se minimiza la divergencia de KL, la cual está sujeta a la restricción de ser mayor o igual que cero.

2.3.3. Inferencia variacional estocástica

La inferencia variacional estocástica o SVI, es el algoritmo principal de la inferencia bayesiana ya que nos permite formular nuestro problema de inferencia bayesiana como un problema de optimización con el objetivo de estimar la verdadera distribución a posteriori cuando el cálculo explícito de la evidencia es intratable. En los problemas de optimización es necesario definir una función objetivo. En este contexto se emplea el ELBO como función de pérdida y se combina con técnicas de descenso del gradiente estocástico para realizar actualizaciones iterativas de los parámetros.

Conceptualmente consiste en introducir una distribución parametrizada $q_\phi(z)$, en la cual ϕ representa los parámetros variacionales y define un espacio paramétrico o una familia de distribuciones de probabilidad. Nuestro objetivo es encontrar una distribución de este espacio que se aproxime de manera óptima a la distribución a posteriori. Es fundamental destacar que la distribución óptima puede no ser única, lo cual añade un aspecto de flexibilidad en la elección de la mejor aproximación.

Partiendo de los fundamentos teóricos de los modelos bayesianos, sabemos que típicamente es difícil calcular la distribución a posteriori ya que no no tiene una expresión algebraica exacta. La SVI busca una distribución $q_\phi(z)$ que sea similar a $p_\theta(z \mid x)$ en base a algún tipo de medida de distancia, y si ambas son similares, podemos usar $q_\phi(z)$ como aproximación de $p_\theta(z \mid x)$.

Restringimos la búsqueda de $q_\phi(z)$ a la familia de distribuciones sobre Z, conocidas como la familia de distribuciones variacionales y denotada por el conjunto de distribuciones Q. Nuestro objetivo es encontrar una distribución $q \in Q$ tal que $q_\phi(z)$ esté tan próxima a $p_\theta(z \mid x)$ como sea posible. Cada elemento de Q se caracteriza por un conjunto de parámetros variacionales ϕ de los que tenemos que encontrar los valores óptimos $\hat{\phi}$, de esta forma obtenemos una aproximación a la verdadera distribución a posteriori $q_{\hat{\phi}}(z)$.

Este planteamiento involucra un problema de optimización, en el que cada iteración de entrenamiento implica un avance en el espacio paramétrico $\{\theta, \phi\}$, acercando de manera gradual los parámetros variacionales hacia la mejor aproximación de la distribución a posteriori, la cual nos permitirá obtener la distribución posterior predictiva para efectuar inferencias.

2.4. Métodos de validación de resultados

Validar los resultados de un modelo consiste en evaluar su capacidad de generalización, es decir, obtener buenos resultados a partir de datos de entrenamiento. Según el tipo de modelo utilizado se emplean diferentes técnicas de validación:

2.4.1. Validación cruzada estratificada

La validación cruzada estratificada, también conocida como *Stratified Cross-Validation*, es una técnica ampliamente utilizada para mitigar el sobreajuste en problemas de *Machine Learning*. En esta técnica, el conjunto de datos se divide en K particiones estratificadas de igual tamaño, por lo que se mantiene la misma proporción de clases en cada una. Luego, se emplean $K-1$ particiones para entrenar el modelo y la partición restante para la validación. Este proceso se repite K veces, utilizando una partición diferente como conjunto de validación en cada iteración. Finalmente, se promedian los resultados obtenidos en las K iteraciones para evaluar el rendimiento del modelo de manera más robusta. Tras esto, se puede entrenar el modelo empleando todos los datos. En la figura 2.5 se presenta un ejemplo.

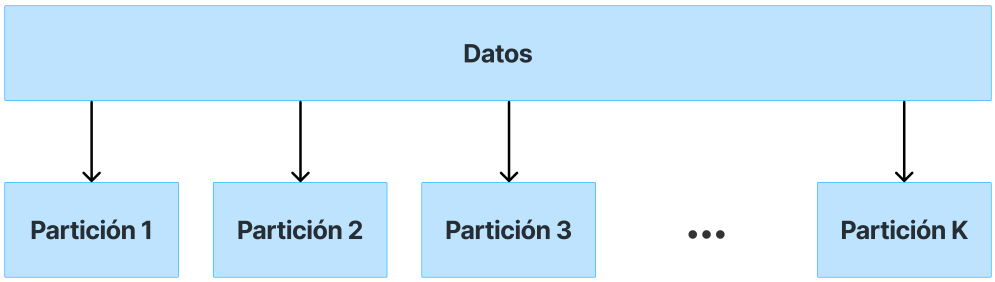

Figura 2.5: Ejemplo de validación cruzada.

2.4.2. División en varios subconjuntos estratificados

La división en varios subconjuntos estratificados es una técnica valiosa para dividir conjuntos de datos en varias particiones, manteniendo las mismas proporciones de clases en cada subconjunto. Esto es crucial para garantizar resultados confiables al entrenar y evaluar modelos de aprendizaje automático. Un ejemplo específico de esta técnica es el método *Hold-Out*, que implica dividir el conjunto de datos en dos subconjuntos. También es común dividir los datos en tres subconjuntos llamados entrenamiento, validación y prueba. En la figura 2.6 se presenta un ejemplo.

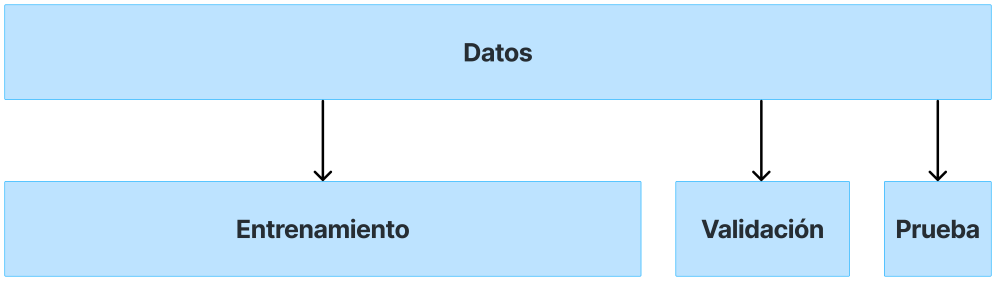

Figura 2.6: Ejemplo de división en tres subconjuntos.

Capítulo 3

Datos

El portal de datos UniversiDATA ofrece información sobre alumnado, profesorado, titulaciones, investigación, finanzas e infraestructuras, en formatos abiertos y estandarizados, facilitando su reutilización. Estos datos se actualizan frecuentemente tanto de forma aditiva para añadir datos nuevos como de forma correctiva para solucionar errores.

Cada conjunto de datos está disponible para las universidades participantes en el proyecto, que hasta el momento son: Universidad Autónoma de Madrid, Universidad Carlos III de Madrid, Universidad Complutense de Madrid, Universidad de Huelva, Universidad de Valladolid y Universidad Rey Juan Carlos.

3.1. Descripción de los datos

En esta sección se detallan los conjuntos de datos utilizados a lo largo del trabajo. Debido al gran número de variables disponibles en cada conjunto, nos enfocaremos únicamente en aquellas empleadas.

3.1.1. Acceso

El conjunto de datos "Acceso" [25] proporciona información detallada sobre los estudiantes que se matriculan por primera vez en cada titulación durante el curso académico correspondiente. Con un total de 79 variables, este conjunto de datos ha sido anonimizado para salvaguardar la privacidad y la seguridad de la información personal de los estudiantes. En la tabla 3.1 se presentan las 20 variables utilizadas a lo largo del trabajo.

ID	Variable	Tipo	Ejemplo
01	Curso académico	Cuantitativa discreta	2019-20
02	Titulación	Cualitativa nominal	Grado en Estadística
03	Género	Cualitativa nominal	Hombre
04	Nota de admisión	Cuantitativa continua	13,69
05	Año de nacimiento	Cuantitativa discreta	2001
06	Comunidad del centro secundario	Cualitativa nominal	Castilla y León
07	Naturaleza del centro secundario	Cualitativa nominal	Concertado
08	Estudios de acceso	Cualitativa nominal	Bachillerato
09	Especialidad de acceso	Cualitativa nominal	Ciencias y Tecnología
10	País de fin de estudio de acceso	Cualitativa nominal	España
11	Año de fin de estudios previos	Cuantitativa discreta	2019
12	Primer acceso al SUE	Cuantitativa discreta	2019
13	País de residencia	Cualitativa nominal	España
14	Nacionalidad	Cualitativa nominal	Española
15	Familia numerosa	Cualitativa nominal	No
16	Ocupación del estudiante	Cualitativa nominal	Otra situación
17	Ocupación del padre	Cualitativa nominal	Ocupaciones militares
18	Ocupación de la madre	Cualitativa nominal	Directores/as y gerentes
19	Nivel de estudios del padre	Cualitativa nominal	Estudios Primarios
20	Nivel de estudios de la madre	Cualitativa nominal	Estudios Secundarios

Tabla 3.1: Variables del conjunto de datos “Acceso” utilizadas en el trabajo.

La mayoría de los nombres de variables son autoexplicativos, pero se proporcionan explicaciones para aquellos que puedan resultar ambiguos:

- **Estudios de acceso:** Nombre de los estudios que dan acceso al estudiante a esta titulación universitaria.
- **Especialidad de acceso:** Nombre de la especialidad de los estudios que dan acceso al estudiante a esta titulación universitaria.
- **País de fin de estudio de acceso:** Nombre del país en el que el estudiante finalizó los estudios que le dan acceso a la universidad.
- **Año de fin de estudios previos:** Último año en que el estudiante cursó los estudios que le dan acceso a la universidad.

- **Primer acceso al SUE:** Curso académico en el que el estudiante ingresó por primera vez al Sistema Universitario Español.

3.1.2. Titulaciones

El conjunto de datos "Titulaciones" [26] contiene información de todas las titulaciones ofertadas por cada Universidad, incluyendo las oficiales de grado, máster y doctorado. Dispone de 35 variables y no está anonimizado dado que no se trata de información sensible. En la tabla 3.2 se presentan las 2 variables utilizadas a lo largo del trabajo.

ID	Variable	Tipo	Ejemplo
01	Titulación	Cualitativa nominal	Grado en Estadística
02	Rama de enseñanza	Cualitativa nominal	Ciencias

Tabla 3.2: Variables del conjunto de datos "Titulaciones" utilizadas en el trabajo.

3.2. Proceso de anonimización

A diferencia de lo que comúnmente se cree, el proceso de anonimización de un conjunto de datos no se limita únicamente a eliminar los identificadores directos, como el DNI o los nombres completos. Es crucial abordar también la prevención de la reidentificación, proceso que implica la capacidad de identificar los datos de un individuo dentro del conjunto de datos, a partir de un subconjunto de información conocida, denominada identificadores indirectos. Algunos ejemplos de identificadores indirectos incluyen la fecha de nacimiento, el género o el código postal, entre otros. El riesgo radica en que, al conocer un subconjunto de identificadores indirectos, el individuo pueda ser reidentificado debido a la falta de otros individuos en el conjunto de datos con los mismos valores. El proceso de anonimización de los datos es el siguiente [27]:

1. Eliminación de los identificadores directos:

   a) Se eliminan todos los identificadores directos, como el DNI, los nombres completos, los números de teléfono o las direcciones particulares.

2. Segmentación del conjunto de datos en grupos:

   a) Seleccionamos un conjunto de variables pivote.

   b) Dividimos el conjunto de datos en grupos basados en la cantidad de combinaciones únicas de las variables pivote.

3. Permutación aleatoria de bloques de variables:

   a) Organizamos las variables restantes, excluyendo las variables pivote, en bloques coherentes, agrupando aquellas cuyos valores estén estrechamente relacionados o dependan entre sí. Esto garantiza que, al permutar los bloques, dichos valores conserven su coherencia interna.

   b) Permutamos aleatoriamente los bloques coherentes entre los registros del grupo.

4. Detección y desidentificación de grupos pequeños:

   a) Identificamos los grupos generados en el segundo paso que contienen un número de registros inferior a un umbral predeterminado.

   b) En estos grupos, eliminamos valores de las variables pivote según sea necesario para disolverlos en un grupo más grande y menos identificable.

Entender este proceso da una idea clara de cómo analizar y trabajar con estos conjuntos de datos. En base a esto, se puede deducir lo siguiente:

1. Se mantiene la relación entre las variables pivote y todas las demás variables.
2. Se mantiene la relación entre las variables que comparten un mismo bloque de coherencia.
3. Se pierde la relación entre variables no pivote que no comparten un mismo bloque de coherencia.
4. Se pierde información de las variables pivote de los grupos pequeños.

3.3. Extractor de datos

Para acceder a los conjuntos de datos en tiempo real hemos desarrollado una función que actúa como un extractor de datos. Esta función interactúa con la API de UniversiDATA, que es una interfaz que define la manera en que se comparten los datos de forma estructurada y segura como se puede ver en la figura 3.1. Esta función recibe como argumentos el nombre del conjunto de datos, la universidad y el curso académico deseado. A partir de esta información, la función realiza una consulta a la base de datos del proveedor de datos, utilizando los parámetros proporcionados para filtrar y extraer los datos relevantes. Esta es una manera eficiente de obtener información en tiempo real sin necesidad de descargar archivos localmente de forma repetida.

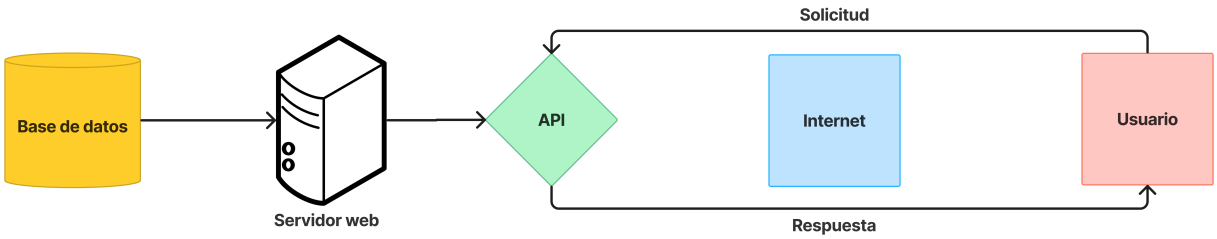

Figura 3.1: Arquitectura del extractor de datos.

3.4. Procesamiento de los datos

El procesamiento de los datos es una etapa esencial que garantiza la integridad y la calidad de la información utilizada en nuestro análisis. En este proceso, los datos brutos son sometidos a una serie de operaciones y transformaciones diseñadas para corregir errores, eliminar inconsistencias y adaptar el formato de los datos.

3.4.1. Validación

El primer paso consiste en verificar la validez de los datos mediante una validación automática de las variables de los conjuntos de datos.

En el caso de las variables cualitativas, también denominadas categóricas, nos aseguramos de que solamente presenten los valores definidos en la descripción del conjunto de datos. Por ejemplo, al examinar la variable “Naturaleza del centro secundario”, verificamos que únicamente existan observaciones que tengan los valores predefinidos, que son: {“Público”, “Privado”, “Concertado”}.

Por otro lado, en las variables cuantitativas, también denominadas continuas, garantizamos que los valores se encuentren dentro del rango establecido. En el caso de la variable “Nota de admisión”, nos aseguramos de que los valores se encuentren en el rango $[5, 14]$ dado que es el rango de valores que puede tomar la nota de admisión a la universidad.

Las instancias que no cumplen el proceso de validación han sido eliminadas del conjunto de datos. Esto se debe a que dichas instancias contienen errores o discrepancias con las especificaciones establecidas. Además, se ha informado al portal de datos sobre estos casos, con el objetivo de contribuir a la mejora continua de la calidad de la información proporcionada.

3.4.2. Unificación

En la variable "Titulación", se han identificado nombres distintos que hacen referencia a la misma titulación, ya sea debido a errores en la creación de los registros o a cambios en la denominación de la titulación a lo largo del tiempo. El objetivo es establecer una identificación única para cada una de las titulaciones. Por ejemplo, los programas de doble titulación oficial, desde el año 2017 hasta el 2022, se denominaban como "Programa de estudios conjunto de", mientras que la nueva nomenclatura es "Programa de Doble Titulación Oficial". Para realizar este proceso de unificación ha sido de gran ayuda la lista de grados ofertados por la UVa [28].

3.4.3. Codificación

Otra parte importante de este proceso es la codificación de las variables categóricas. Una técnica comúnmente utilizada para este propósito es la codificación *One-Hot*, la cual convierte una variable categórica con K grupos en K-1 variables binarias, cada una de las cuales indica si la observación pertenece o no a una clase específica de la variable categórica. Existe una variable codificada menos que categorías existentes para evitar la llamada *dummy variable trap*, que ocurre cuando una variable puede ser deducida completamente a partir de las otras, lo que introduce multicolinealidad en los datos.

Instancia	Naturaleza del centro secundario
1	Público
2	Privado
3	Concertado

One-Hot →

Instancia	Público	Privado
1	1	0
2	0	1
3	0	0

Figura 3.2: Ejemplo de codificación *One-Hot* estándar.

Para crear un sistema de información de estudios preciso, es fundamental manejar adecuadamente las variables que contienen valores faltantes. Estos valores pueden ser una fuente importante de información y no deben ser simplemente descartados. En lugar de ello, podemos aprovecharlos al máximo en el dominio de nuestro problema, considerando que la ausencia de un valor puede ser significativa y debe ser tratada como una categoría adicional. Esto significa que necesitamos incluir una variable binaria adicional para representar la ausencia de valor en esa variable. Por lo tanto, en estos casos, necesitamos K variables binarias en lugar de K-1. Este enfoque nos aporta la flexibilidad necesaria para utilizar toda la información disponible. Por otro lado, para las variables que no tienen valores faltantes, podemos seguir utilizando la codificación *One-Hot* estándar.

Instancia	Naturaleza del centro secundario
1	Público
2	Privado
3	Concertado
4	---------

One-Hot →

Instancia	Público	Privado	Concertado
1	1	0	0
2	0	1	0
3	0	0	1
4	0	0	0

Figura 3.3: Ejemplo de codificación *One-Hot* para las variables con valores faltantes.

Las variables relacionadas con las comunidades autónomas han sido categorizadas previamente a la codificacion en dos grupos: "Propia" para las comunidades referentes a Castilla y León y "Externa" para el resto. Esta categorización se ha realizado con el objetivo de reducir el número de categorías, ya que no era esencial conocer la comunidad específica. Del mismo modo, este procedimiento se aplicó a las variables relacionadas con los países, siendo categorizadas como "Propio" si se trata de España y "Externo" en caso contrario.

3.4.4. Escalado

El escalado es una técnica de transformación de los datos fundamental en el preprocesamiento de las variables continuas para asegurar que todas estén dentro de un rango específico y que sean medidas comparables entre sí. Además, permite que los algoritmos trabajen con los datos de forma más rápida. En este caso, he aplicado un escalado para llevar todas las variables continuas al rango $[0, 1]$.

$$X_{\text{escalado}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{3.1}$$

Para escalar la variable "Nota de admisión", se utilizan los valores $X_{min} = 5$ y $X_{max} = 14$, ya que es el rango de valores que puede tomar esta variable.

Para la variable "Año de fin de estudios previos", es necesario tener en cuenta que los datos disponibles se limitan al curso académico 2022-23. Por lo tanto, se establece el año máximo como $X_{max} = 2022$, mientras que el mínimo se fija en $X_{min} = 1981$, que es el valor mínimo presente en los datos

Para la variable "Año de nacimiento", se toma en cuenta que la edad típica de ingreso a la universidad es de 18 años. Esto implica que, en el curso 2022-23, los estudiantes deben haber nacido en el año 2004 como mínimo. Sin embargo, se han identificado 8 casos de estudiantes nacidos en 2005 en los datos, lo que significa que han sido adelantados un curso académico por lo que establecemos $X_{max} = 2005$. El valor mínimo presente en los datos es $X_{min} = 1940$.

Capítulo 4

Sistema de información para la elección de estudios universitarios de la UVa

En este capítulo se explica todo el proceso de creación de un sistema de información para la elección de estudios universitarios de la Universidad de Valladolid. Se han escogido 19 variables del conjunto de datos "Titulaciones" y 2 del conjunto "Acceso". Ambos conjuntos contienen más variables pero se realizó una selección manual considerando si eran relevantes en este contexto. Ambos conjuntos se han combinado entre sí, dando lugar a uno nuevo formado por 20 variables, aprovechando una variable común denominada "Titulación".

4.1. Imputación de valores faltantes

La variable "Nota de admisión" presenta 1.363 valores faltantes y como es potencialmente importante debido a su influencia en el proceso de admisión, se ha tomado la decisión de no imputar sus valores ausentes para evitar sesgar el sistema y, además, porque la cantidad de observaciones con valores faltantes supone menos de un 4,40 % del tamaño total del conjunto de datos. En su lugar, se ha optado por descartar estos 1.363 casos.

Sin embargo, la variable "Año de fin de estudios previos" presenta originalmente 7.450 valores faltantes, pero quedan 7.000 tras eliminar los casos con nota de admisión faltante. En lugar de eliminar estos casos y perder información, se pueden imputar estos valores utilizando todas las variables del conjunto de datos "Acceso" excepto el curso académico y la titulación. Si separamos estos casos del conjunto original, nos quedan 22.299 para el desarrollo de un buen modelo de imputación de valores. Para abordar este problema hemos utilizado un enfoque de regresión mediante el algoritmo de aprendizaje automático *Gradient Boosting*. La regresión funciona mejor que la clasificación debido al gran número de clases que suponen esos años.

Es importante tener en cuenta que la salida del modelo de regresión no tiene por qué ser un año exacto, por lo que si la cifra inmediatamente posterior al punto decimal es 5 o mayor, se redondea hacia arriba añadiendo una unidad al último dígito significativo. En cambio, si dicha cifra es menor que 5, se redondea hacia abajo, dejando el último dígito significativo sin cambios.

La división de los datos y el entrenamiento del modelo se han llevado a cabo siguiendo estos pasos:

1. Se divide el conjunto de datos en dos utilizando el método de *Hold-Out* estratificado, resultando en un conjunto de entrenamiento que tiene el 66,66 % de los datos del conjunto original, lo que se traduce en 14.865 casos, mientras que el de prueba tiene el 33,33 % y cuenta con 7.434 casos. Entrenamos el modelo inicial con el conjunto de entrenamiento y unos hiperparámetros iniciales.

2. Seguidamente se realiza validación cruzada estratificada de 10 particiones con 10 repeticiones sobre el conjunto de entrenamiento. En cada iteración se modifica el valor de un hiperparámetro con el fin de ver si el modelo mejora con el cambio.

3. Finalmente se seleccionan los mejores hiperparámetros y se utiliza todo el conjunto de datos de entrenamiento para entrenar el modelo. Utilizamos este modelo para predecir los casos del conjunto de prueba.

	Modelo inicial	Modelo óptimo
Tasa de aciertos (%)	67,74	76,48
Número mínimo de muestras para dividir un nodo interno	2	4
Número mínimo de muestras para formar un nodo hoja	1	1
Profundidad máxima de los árboles	3	4
Número máximo de hojas permitidas en un árbol	Ilimitado	6
Tasa de aprendizaje	0.1	0.05
Número de árboles	100	100
Proporción de muestras a utilizar para ajustar cada árbol	1	0.9

Tabla 4.1: Comparación del modelo de *Gradient Boosting* inicial con el óptimo.

Con este modelo se pueden obtener predicciones de la variable respuesta sobre los casos con valores faltantes. Es necesario escalar esta variable según el proceso explicado en 3.4.4 para obtener el conjunto de datos definitivo que se utilizará para entrenar los siguientes modelos.

4.2. Arquitectura del sistema

Para lograr nuestro objetivo es esencial diseñar una arquitectura apropiada para el problema, ya que abordarle simplemente como un problema de clasificación directa o múltiple no sería factible debido a la gran cantidad de titulaciones disponibles (71), la naturaleza de los datos y el proceso de anonimización. Por ello, se ha diseñado una arquitectura que consta de dos etapas para abordar este problema de manera efectiva:

En la primera etapa, el objetivo principal es determinar la rama o ramas de enseñanza que mejor se ajustan al perfil del estudiante. Esta etapa inicial actúa como un filtro, reduciendo el espacio de búsqueda y proporcionando una base sólida para las predicciones posteriores.

La segunda etapa consta de varios modelos independientes, como se ve en la figura 4.1. Cada uno de estos modelos está asociado a una rama de enseñanza académica, lo que permite que estén diseñados específicamente para informar de titulaciones dentro de su área correspondiente. Dado el reducido número de casos en la rama de enseñanza de "Ciencias" (1.909) y su estrecha similitud con la rama de "Ingeniería y Arquitectura", se ha optado por fusionar los casos en una nueva categoría.

Las técnicas bayesianas empleadas en esta segunda fase ayudan a reducir la incertidumbre generada por el proceso de anonimización y la variabilidad de las decisiones de los estudiantes, dado que es posible que dos personas con características idénticas elijan titulaciones distintas.

En ambas etapas, se han utilizado como variables explicativas todas las variables del conjunto de datos "Acceso", a excepción del curso académico y la titulación. La variable respuesta de la primera etapa es la rama de enseñanza académica del conjunto "Titulaciones", mientras que la variable respuesta de la segunda etapa es la titulación del conjunto "Acceso".

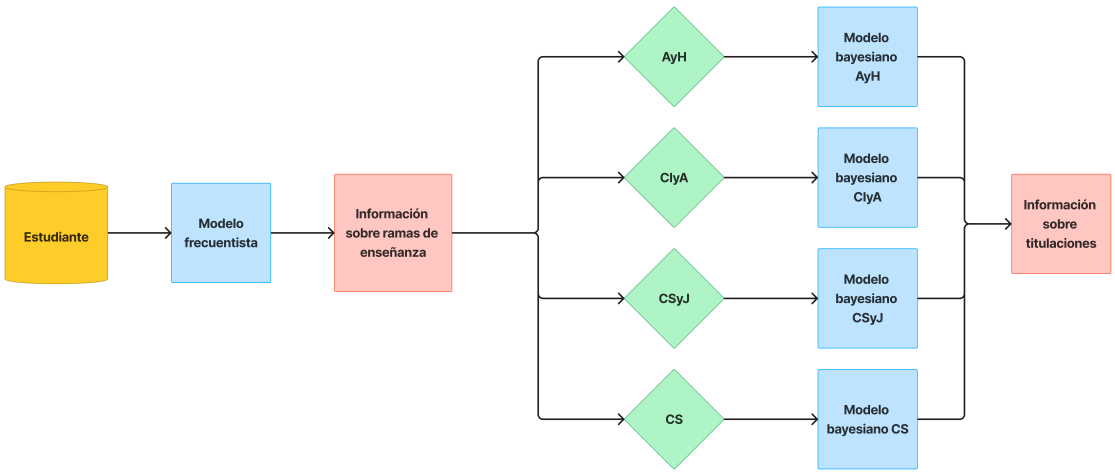

Figura 4.1: Arquitectura del sistema.

4.2.1. Ramas de enseñanza

En esta fase inicial, se desarrolla una red neuronal frecuentista para estimar la probabilidad de asignar a un individuo a cada una de las ramas de enseñanza académica, que son: "Artes y Humanidades", "Ciencias, Ingeniería y Arquitectura", "Ciencias Sociales y Jurídicas" y "Ciencias de la Salud".

El conjunto de datos está formado por 29.299 instancias, 82 variables explicativas teniendo en cuenta la codificación *One-Hot* de las variables originales y la rama de enseñanza académica como variable respuesta. Se divide en tres subconjuntos disjuntos de forma estratificanda según la variable respuesta: el primero se destina al entrenamiento del modelo, el segundo para la selección del mejor modelo y el tercero para probar su capacidad de generalización. La división se presenta en la tabla 4.2, con proporciones del 66,66 %, 22,22 %, y 11,11 %, respectivamente.

Conjunto de datos	**Casos**	**Rama de enseñanza**			
		AyH	**CIyA**	**CSyJ**	**CS**
Entrenamiento	19.532	1.403	5.115	10.584	2.430
Validación	6.511	468	1.709	3.511	823
Prueba	3.256	218	870	1.743	425
Total	29.299	2.089	7.694	15.838	3.678

Tabla 4.2: División del conjunto de datos en la primera etapa.

Se observa que la distribución de clases está desequilibrada, lo que sugiere la necesidad de utilizar la entropía cruzada aplicada a un problema de clasificación de múltiples clases como función de pérdida. Esta función permite evaluar la diferencia entre la distribución de probabilidad prevista por el modelo y la distribución de probabilidad real de las clases.

$$L(\hat{y}, y) = -\sum_{c=1}^{C} y_c \log(\hat{y}_c) \tag{4.1}$$

Donde:

- C es el número de clases.
- y_c es la probabilidad real de la clase c.
- $\hat{y}_c$ es la probabilidad prevista por el modelo de la clase c.

La red neuronal produce valores numéricos para cada clase, conocidos como puntuaciones. Estos valores, que pueden ser difíciles de interpretar directamente, se pueden transformar en probabilidades mediante la función *SoftMax*, lo que hace que sean más comprensibles. Esta función recibe un vector de puntuaciones como entrada y produce un nuevo vector cuyos elementos representan las probabilidades normalizadas de cada clase. La ecuación es la siguiente:

$$P(y = i) = \frac{e^{z_i}}{\sum_{c=1}^{C} e^{z_c}} \tag{4.2}$$

Donde:

- C es el número de clases.
- z_i es la puntuación correspondiente a la clase i.

Durante el proceso de clasificación se determina la clase con la probabilidad más alta de pertenencia. Si esta probabilidad supera un umbral de confianza predeterminado, establecido en 0,75 en este caso, el sistema asigna exclusivamente a esa clase. Por otro lado, si la probabilidad es menor que el umbral, el sistema asigna a la clase principal junto con la siguiente que mayor probabilidad tenga. Esta estrategia asegura una respuesta robusta, ya que si la red neuronal no tiene suficiente certeza, es preferible considerar dos clases potenciales en lugar de una única.

Tras haber probado varios modelos con diferentes hiperparámetros y secuencias de capas, el modelo que mejores resultados ha presentado sobre el conjunto de validación consta de los hiperparámetros de la tabla 4.3.

Épocas	200
Batch	64
Learning rate	0.0001
Optimizador	AdamW
Función de pérdida	CrossEntropyLoss

Tabla 4.3: Hiperparámetros óptimos de la red neuronal frecuentista.

Donde:

- **Épocas** es el número de veces que un modelo se entrena con todo el conjunto de datos de entrenamiento.

- ***Batch*** es el tamaño del lote de datos de entrenamiento que se utiliza para actualizar los pesos de un modelo durante el entrenamiento.
- ***Learning rate*** es una tasa de ajuste que determina la magnitud de los cambios realizados en los parámetros de un modelo durante el entrenamiento.
- **Optimizador** es el algoritmo que utiliza para ajustar los parámetros de un modelo de manera que minimice la función de pérdida
- **Función de pérdida** es una medida que evalúa si las predicciones de un modelo son buenas en comparación con los valores reales.

El modelo óptimo consta de 41.428 parámetros como se puede ver en el resumen parámetrico de la figura 4.2.

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv1d-1               [-1, 8, 80]              32
              ReLU-2               [-1, 8, 80]               0
         MaxPool1d-3               [-1, 8, 40]               0
            Conv1d-4              [-1, 16, 38]             400
              ReLU-5              [-1, 16, 38]               0
         MaxPool1d-6              [-1, 16, 19]               0
            Conv1d-7              [-1, 32, 17]           1,568
              ReLU-8              [-1, 32, 17]               0
         MaxPool1d-9               [-1, 32, 8]               0
           Conv1d-10               [-1, 64, 6]           6,208
             ReLU-11               [-1, 64, 6]               0
        MaxPool1d-12               [-1, 64, 3]               0
           Linear-13                 [-1, 128]          24,704
             ReLU-14                 [-1, 128]               0
          Dropout-15                 [-1, 128]               0
           Linear-16                  [-1, 64]           8,256
             ReLU-17                  [-1, 64]               0
          Dropout-18                  [-1, 64]               0
           Linear-19                   [-1, 4]             260
================================================================
Total params: 41,428
Trainable params: 41,428
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.00
Forward/backward pass size (MB): 0.05
Params size (MB): 0.16
Estimated Total Size (MB): 0.20
----------------------------------------------------------------
```

Figura 4.2: Resumen paramétrico de la red neuronal frecuentista.

En las figuras 4.3, 4.4 y 4.5 se pueden visualizar los gráficos utilizados para seleccionar el modelo óptimo. Hemos comprobado que el modelo de la época 127 es el mejor en la asignación unaria, mientras que el modelo de la época 101 obtiene mejores resultados en la asignación binaria. El primero logra una tasa de aciertos del 88,74 % sobre el conjunto de validación, mientras que el segundo alcanza un 96,68 %. Estos resultados indican una generalización efectiva en ambas modalidades de clasificación. Además, estos modelos no muestran signos de sobreajuste ya que se puede ver que la función de pérdida se ha estabilizado aproximadamente a partir de la época 100 y la función de pérdida de entrenamiento disminuye mientras que la de validación oscila en torno a unos mismos valores.

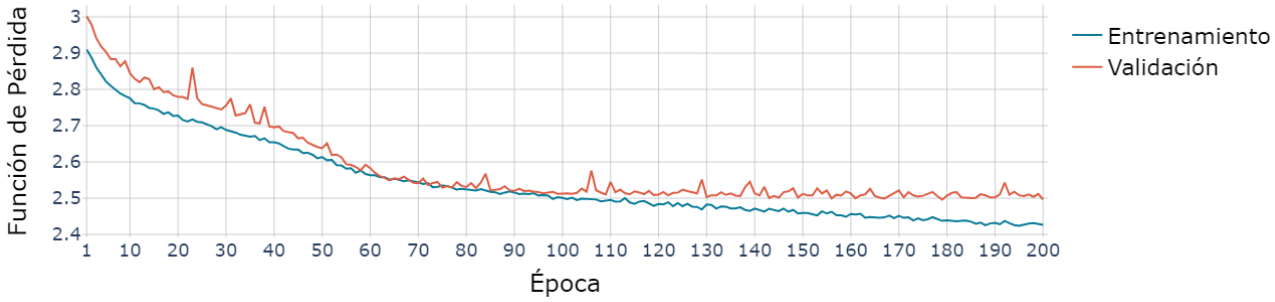

Figura 4.3: Función de pérdida versus época.

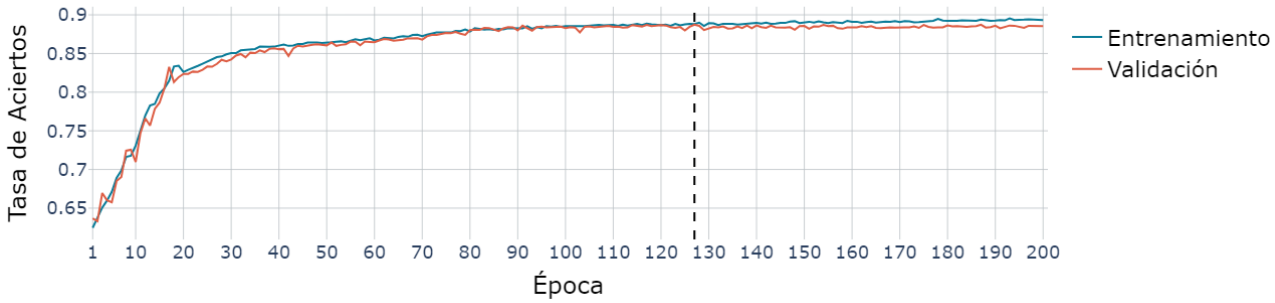

Figura 4.4: Tasa de aciertos unaria versus época.

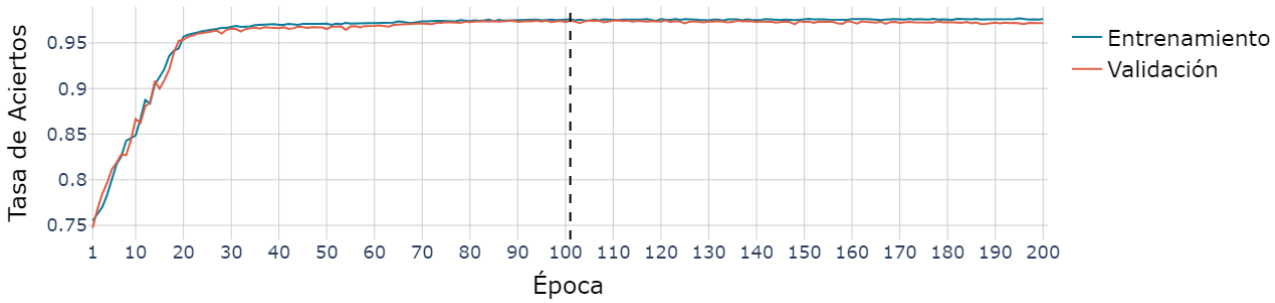

Figura 4.5: Tasa de aciertos binaria versus época.

Con el objetivo de ver qué tal se comporta nuestro modelo sobre datos que no han sido previamente contemplados ni para el entrenamiento ni para la selección del modelo, se prueban los modelos óptimos sobre el conjunto de prueba. En la tabla 4.4 se muestran las tasas de aciertos de asignación unaria y binaria para cada conjunto de datos. Resulta evidente que los modelos son adecuados dado que presentan buenas tasas de acierto y hay una mejora notable en la tasa de aciertos al considerar dos ramas de enseñanza académica en lugar de una, lo que se traduce en información más precisa y de mejor calidad.

Conjunto de datos	**Tasa de aciertos (%)**	
	Asignación unaria	**Asignación binaria**
Entrenamiento	88,80	97,37
Validación	88,74	96,68
Prueba	87,32	96,34

Tabla 4.4: Tasa de aciertos de las asignaciones unarias y binarias.

Es fundamental verificar la posible existencia de desequilibrios en las predicciones, con el fin de detectar cualquier tendencia o sesgo en alguna clase en particular. En las tablas 4.5 y 4.6 se puede observar que las tasas de aciertos son razonablemente buenas para todas las clases, siendo "Artes y Humanidades" la que menor tasa ha obtenido, lo que tiene sentido ya que se trata del grupo con menos observaciones por lo que el modelo no ha podido aprender tan bien de los datos como en el resto de clases.

Conjunto de datos	**Tasa de aciertos (%)**				
	Asignación unaria	**Rama de enseñanza**			
		AyH	**CIA**	**CSyJ**	**CS**
Entrenamiento	88,80	75,05	88,26	92,09	83,53
Validación	87,74	74,35	87,12	90,40	85,29
Prueba	87,32	74,77	86,55	89,67	85,64

Tabla 4.5: Tasa de aciertos unaria desglosada por rama de enseñanza académica.

Conjunto de datos	**Tasa de aciertos (%)**				
	Asignación unaria	**Rama de enseñanza**			
		AyH	**CIA**	**CSyJ**	**CS**
Entrenamiento	97,37	81,75	99.34	99,79	91,77
Validación	96,68	80,77	98.24	99,34	91,13
Prueba	96,34	82,57	97,59	99,02	89,88

Tabla 4.6: Tasa de aciertos binaria desglosada por rama de enseñanza académica.

Podemos mejorar la evaluación del modelo óptimo visualizando la figura 4.6 que representa la matriz de confusión de la asignación binaria sobre el conjunto de prueba, lo que nos permitirá comprender mejor cómo generaliza nuestro modelo. Para crear una matriz de confusión a partir de una asignación binaria, el proceso es sencillo: si se asigna a una única clase, se realiza como de costumbre. Sin embargo, si se asigna a dos clases, en caso de acierto se procede como siempre, pero en caso de error, se indica el fallo en la clase con mayor probabilidad de pertenencia. Esto es una forma de obtener una matriz de confusión a partir de una asignación binaria, ya que este tipo de matrices están diseñados para asignaciones unarias, por lo que afecta a la posición de los errores en la matriz de confusión pero no a las tasas de acierto obtenidas anteriormente.

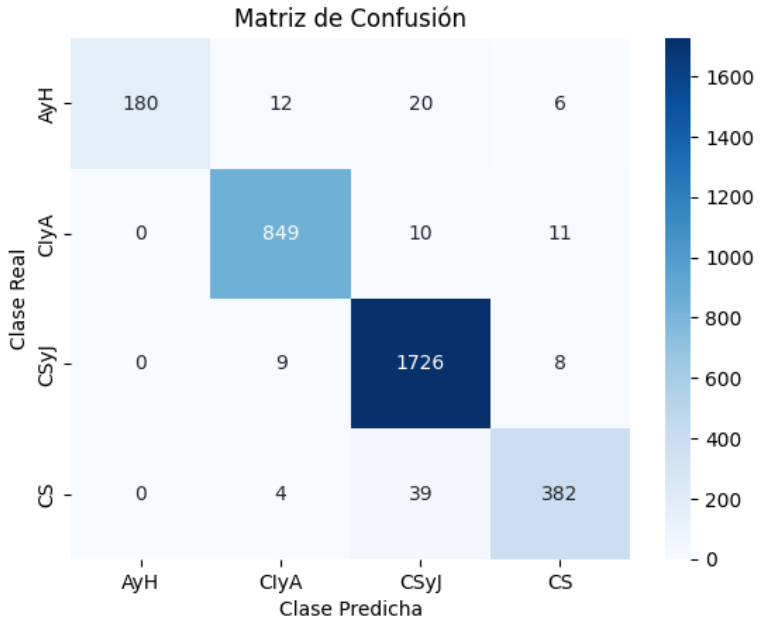

Figura 4.6: Matriz de confusión de la asignación binaria sobre el conjunto de prueba.

Los resultados obtenidos sugieren que este modelo es idóneo para la etapa inicial del sistema. No obstante, es de interés comprender las razones detrás de estos resultados. Para este propósito, se emplean los valores SHAP (*SHapley Additive exPlanations*), que es una técnica que permite analizar la explicabilidad de los modelos de aprendizaje automático ya que proporciona una forma de entender cómo se llega a una predicción específica [29]. El proceso es el siguiente:

Para una instancia de datos específica x, el valor SHAP para la variable explicativa j se define como:

$$SHAP_j(x) = \phi_0 + \sum_{S \subseteq J \setminus \{j\}} \frac{|S|!(|J| - |S| - 1)!}{|J|!} \left[f(x_S) - f(x_{S \cup \{j\}}) \right] \tag{4.3}$$

Donde:

- $SHAP_j(x)$ es el valor SHAP para la variable j en la instancia x.
- ϕ_0 es un valor de referencia que se establece por defecto.
- J es el conjunto de todas las variables.
- S es un subconjunto de J que excluye la variable j.
- x_S es la instancia x con todas las variables de S.
- $f(x_S)$ es la predicción del modelo para la instancia x_S.
- $f(x_{S\cup\{j\}})$ es la predicción del modelo para la instancia x_S e incluyendo el valor de la variable j.

El sumatorio abarca todos los subconjuntos posibles de características de S que excluyen la variable j.

El término de normalización $\frac{|S|!(|J|-|S|-1)!}{|J|!}$ normaliza la contribución de cada subconjunto por el número total de permutaciones posibles. El operador $|S|$ denota la cardinalidad o el tamaño de un conjunto.

La diferencia $\left[f(x_S) - f(x_{S\cup\{j\}})\right]$ cuantifica la discrepancia en las predicciones del modelo al agregar la característica j al subconjunto S, en comparación con mantenerle sin ella.

De esta forma se asigna una contribución a cada variable explicativa para cada predicción del modelo, mostrando cómo cada característica a afecta la predicción del modelo en comparación con un valor de referencia. En la figura 4.7 se muestra la importancia global de cada una de las 9 variables explicativas más influyentes en la variable respuesta, en términos absolutos.

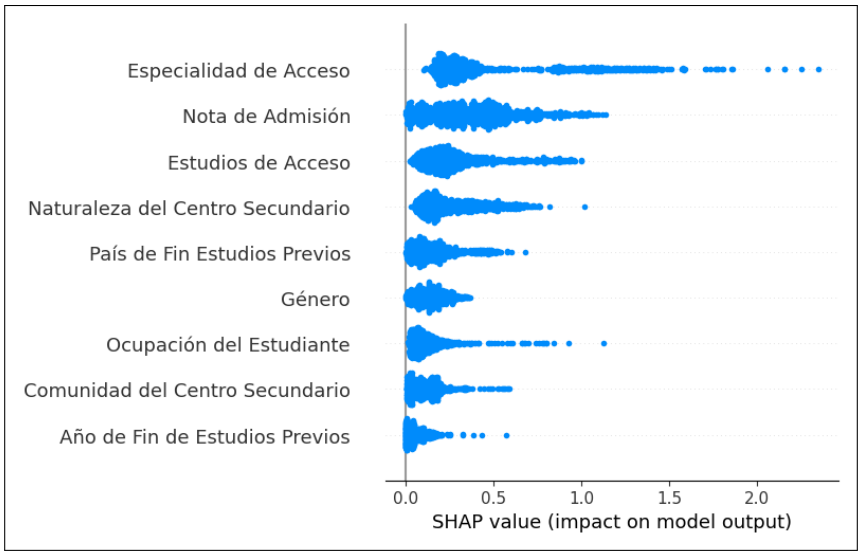

Figura 4.7: Top 9 variables globalmente más influyentes.

Se observa que las variables más influyentes son la especialidad de acceso y la nota de admisión. Esto implica que al conocer la especialidad de estudios cursada previamente por el estudiante, podemos inferir bastante acerca de la modalidad de estudios que es probable que elija. Además, es destacable el impacto de la comunidad de origen del centro secundario, diferenciando si es de Castilla y León o de fuera, dado que los estudiantes de fuera de la comunidad tienden a ser quienes más acceden a ciertas titulaciones en la UVa.

Es posible desglosar la contribución para cada una de las ramas de enseñanza así como para cada una de las variables cualitativas que han sido codificadas. De esta forma podemos identificar qué categorías concretas son las que más influyen en cada rama de enseñanza. A continuación pasamos a describir los resultados obtenidos en cada rama de enseñanza:

Artes y Humanidades

Para interpretar los resultados de la figura 4.8, utilizamos la variable nota de admisión como referencia y el resto de las variables se mantienen iguales, por lo que valores más bajos en la nota de admisión aumentan la probabilidad de que el individuo pertenezca a este grupo, mientras que valores más altos disminuyen esa probabilidad. Este dato tiene sentido, considerando que las titulaciones dentro de esta rama suelen tener notas de admisión menos exigentes. Además, observamos que la segunda y sexta variables más influyentes se relacionan con dos especialidades de acceso específicas, ya que aumentan la probabilidad de pertenecer a este grupo, lo cual es un factor muy influyente, como se explicó anteriormente. Por otro lado, se evidencia que las personas con formación profesional, como estudios de acceso, muestran una menor probabilidad de buscar titulaciones dentro de esta rama de enseñanza académica.

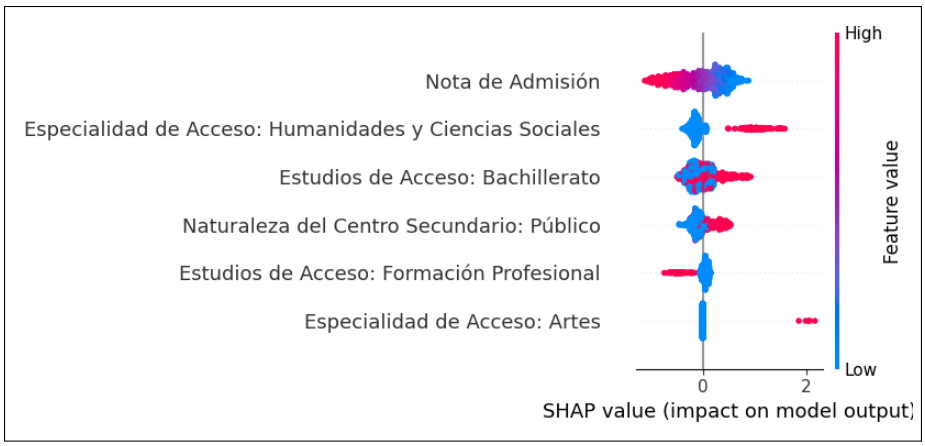

Figura 4.8: Top 6 variables más influyentes para Artes y Humanidades.

Ciencias, Ingeniería y Arquitectura

En la figura 4.9 observamos que una nota de admisión alta es un factor determinante que aumenta significativamente la probabilidad de que un individuo pertenezca a esta rama. Por otro lado, observamos que la especialidad de acceso en humanidades y ciencias sociales tiende a disminuir la probabilidad, mientras que la especialidad en ciencias y tecnología la incrementa, lo cual es coherente.

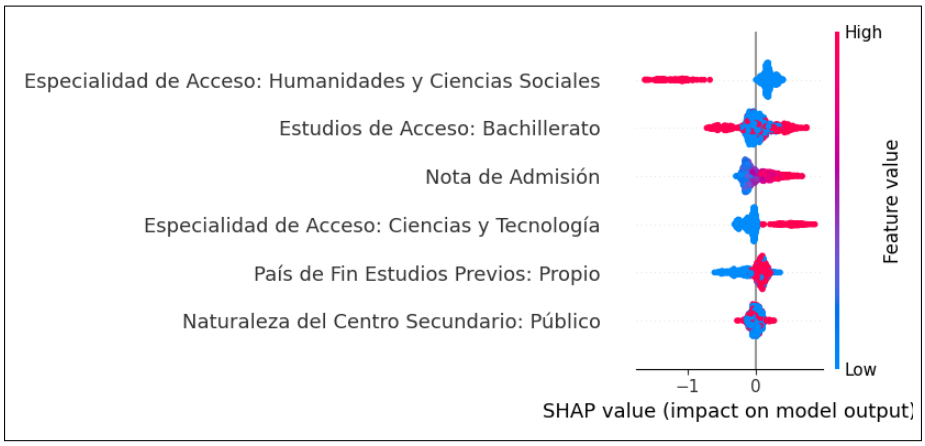

Figura 4.9: Top 6 variables más influyentes para Ciencias, Ingeniería y Arquitectura.

Ciencias Sociales y Jurídicas

En la figura 4.10 se aprecia un fenómeno similar al de artes y humanidades en cuanto a la nota de admisión, ya que generalmente requiere notas de admisión más bajas. Es más común que las personas con la especialidad de acceso de humanidades y ciencias sociales opten por esta opción, mientras que es menos probable que lo hagan aquellos con formación en ciencias y tecnología. Otro aspecto relevante es que la elección de esta rama es más probable entre las mujeres.

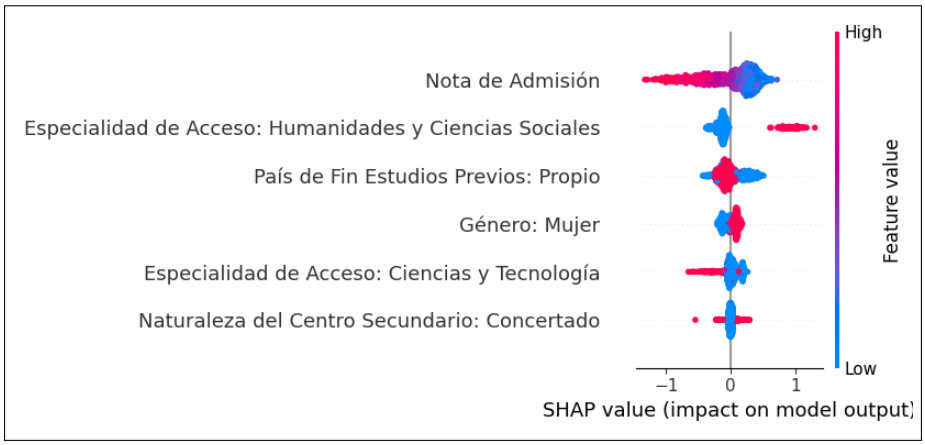

Figura 4.10: Top 6 variables más influyentes para Ciencias Sociales y Jurídicas.

Ciencias de la Salud

En la figura 4.11 la nota de admisión tiene un comportamiento similar a ciencias, ingeniería y arquitectura ya que valores más elevados en esta variable incrementan la probabilidad de que un estudiante elija esta rama, ya que generalmente se exigen notas muy elevadas. También es más probable que las mujeres elijan esta opción, mientras que haber estudiado la especialidad en humanidades y ciencias sociales disminuye notablemente la probabilidad

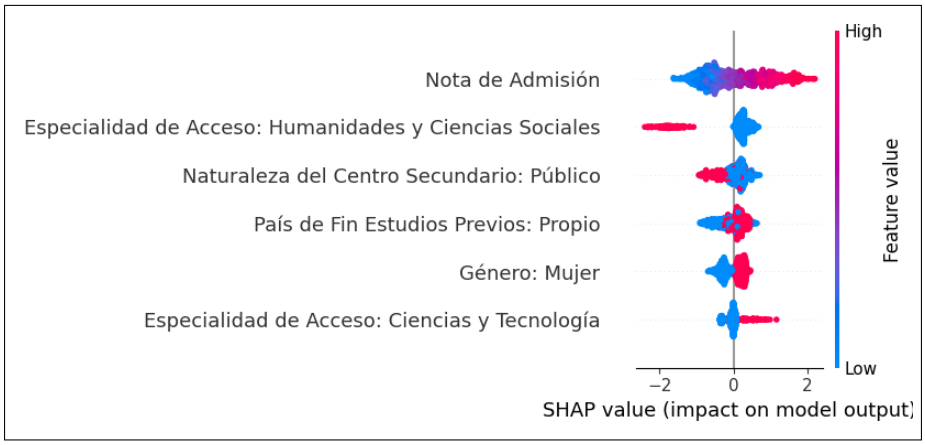

Figura 4.11: Top 6 variables más influyentes para Ciencias de la Salud.

4.2.2. Titulaciones

En esta segunda etapa, el objetivo principal es analizar la información relacionada con las titulaciones que mejor se ajustan al perfil de los estudiantes. Para ello, se crean cuatro redes neuronales bayesianas independientes, con diferentes arquitecturas e hiperparámetros y el objetivo de especializar a cada una en la información de los estudios asociados a una rama de enseñanza académica. La tabla 4.7 muestra el número de titulaciones asociadas a cada modelo.

Red neuronal	AyH	CIyA	CSyJ	CS
Titulaciones	10	32	23	6

Tabla 4.7: Número de titulaciones asociadas a cada modelo.

El conjunto de datos está formado or 29.299 instancias y 82 variables explicativas teniendo en cuenta la codificación de las variables originales. En esta etapa, la variable respuesta es la titulación. La rama de enseñanza académica se utiliza para dividir el conjunto de datos en cuatro subconjuntos disjuntos. Además, cada uno de estos subconjuntos se divide en tres partes, siguiendo la metodología y las proporciones habituales. Esta división es la misma que la de la tabla 4.2, pero es importante tener en cuenta que las observaciones empleadas no son las mismas, sino que representan las mismas cantidades.

La diferencia principal de estas redes con la de la anterior etapa, es que los parámetros de estas son distribuciones en lugar de ser estimaciones puntuales. A continuación, se presentan los hiperparámetros óptimos y la secuencia de capas empleada para cada una de estas redes. En la tabla 4.8 se pueden consultar los hiperparámetros que se han mantenido fijos para cada una de las redes neuronales.

Épocas	200
Optimizador	AdamW
Función de pérdida	TraceELBO
Guía	AutoDiagonalNormal
Desviación de la guía	0.1
Distribución inicial	N(0,1)
Muestras	10

Tabla 4.8: Hiperparámetros óptimos comunes de las redes neuronales bayesianas.

Donde:

- **Guía**: es el mecanismo encargado de explorar el espacio potencial de distribuciones para cada uno de los parámetros variacionales.
- **Desviación de la guía**: es el parámetro de escala de la distribución que siguen inicialmente los parámetros variacionales. El parámetro de localización de la guía no se puede modificar.
- **Distribución inicial**: se refiere a la distribución que genera los valores iniciales de las variables latentes, también conocida como distribución a priori.
- **Muestras**: es el número de muestras que se utilizan en la distribución posterior predictiva.

El resto de hiperparámetros se han modificado en las diferentes redes como se puede ver en la tabla 4.9. El número de partículas se refiere a la cantidad de muestras utilizadas en el cálculo de la función de pérdida mediante el ELBO. Un valor más alto generalmente produce mejores resultados, aunque supone mayor coste computacional. Por otro lado, las capas bayesianas indican cuáles de ellas tienen variables latentes, mientras que el resto de capas son frecuentistas. Es importante encontrar un equilibrio ya que los modelos con un gran número de variables latentes tienden a converger más lentamente e incluso podrían no converger si la variabilidad es excesiva.

AyH	
Batch	64
Learning rate	0.00001
Partículas	10
Capas bayesianas	3

CIyA	
Batch	128
Learning rate	0.0001
Partículas	20
Capas bayesianas	2

CSyJ	
Batch	256
Learning rate	0.0001
Partículas	20
Capas bayesianas	2

CS	
Batch	64
Learning rate	0.00001
Partículas	10
Capas bayesianas	2

Tabla 4.9: Hiperparámetros óptimos no comunes de las redes neuronales bayesianas.

El resumen paramétrico de cada modelo se puede consultar en el apéndice B, donde se observa que los modelos con mayor cantidad de titulaciones asociadas requieren arquitecturas más complejas. Además, aquellos modelos con un conjunto de datos de entrenamiento más amplio han mostrado un mejor rendimiento al utilizar un tamaño de lote más grande. Se ha optado por mantener el número de variables latentes reducido para equilibrar el coste computacional, la variabilidad, la convergencia y la obtención de buenos resultados.

Las variables latentes incluídas en estos modelos permiten generar varias muestras de los resultados, obteniendo a su vez distribuciones de las puntuaciones para cada una de las titulaciones. En base a las réplicas de las puntuaciones, podemos elaborar una métrica que permita obtener el porcentaje de coincidencia del alumno con cada titulación:

1. Calcular la mediana de las puntuaciones para cada titulación.
2. Determinar el rango de escalado con el percentil 5 y 95 de todas las puntuaciones.
3. Escalar las medianas de las puntuaciones de cada titulación con el rango de escalado.

A continuación, se muestran figuras de las puntuaciones asociadas a cada titulación para un estudiante de cada rama de enseñanza académica. De esta forma se puede ver un ejemplo del funcionamiento de cada uno de los modelos especializados.

Artes y Humanidades

Comenzamos con un ejemplo de un estudiante que accedió al Grado en Lenguas Modernas y sus Literaturas. Utilizando el modelo especializado correspondiente, se han obtenido las puntuaciones asociadas a cada una de las titulaciones, las cuales se representan en la figura 4.12. Las siglas de las titulaciones se pueden consultar en el apéndice A.3.

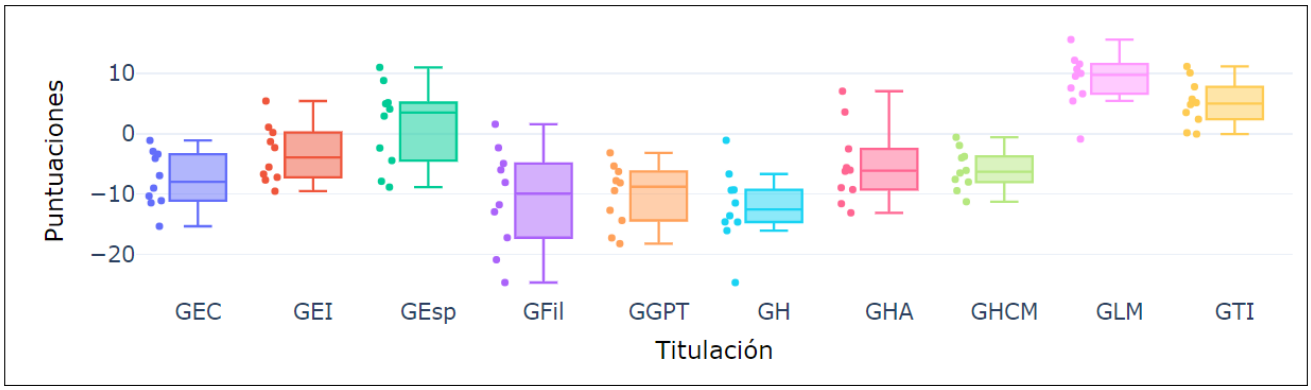

Figura 4.12: Puntuaciones asociadas a cada titulación para un estudiante del Grado en Lenguas Modernas y sus Literaturas.

En la tabla 4.10 se pueden ver los porcentajes de coincidencia asociados a cada titulación a partir de las de las puntuaciones obtenidas para el anterior estudiante. Esto nos permite informar sobre el porcentaje de coincidencia asociado a cada una de las posibles titulaciones. Sin embargo, para la evaluación de la precisión de los modelos, es necesario elaborar una métrica seleccionando únicamente las titulaciones cuyo porcentaje de coincidencia sea mayor a un umbral de decisión determinado, establecido en 0,70. En el caso de que una de esas titulaciones, sea la correcta, se considera un acierto.

Titulación	**Coincidencia (%)**
GEC	45,26
GEI	56,27
GEsp	76,42
GFil	39,97
GGPT	43,06

Titulación	**Coincidencia (%)**
GH	32,89
GHA	50,29
GHCM	49,82
GLM	93,38
GTI	80,42

Tabla 4.10: Coincidencia asociada a cada titulación para un estudiante del Grado en Lenguas Modernas y sus Literaturas.

En la tabla 4.11 vemos que la titulación que mayor coincidencia tiene es el Grado en Lenguas Modernas y sus Literaturas, lo que supone un acierto para el modelo. Otras titulaciones con alta coincidencia son el Grado en Traducción e Interpretación y el Grado en Español: Lengua y Literatura, las cuales están relacionadas con las lenguas y la literatura.

Titulación	GLM	GTI	GEsp
Coincidencia (%)	93,38	80,42	76,42

Tabla 4.11: Coincidencia asociada a las titulaciones que superan el umbral de decisión para un estudiante del Grado en Lenguas Modernas y sus Literaturas.

Ciencias, Ingeniería y Arquitectura

Para el ejemplo de esta rama de enseñanza académica se ha seleccionado un estudiante que accedió al Grado en Estadística. Utilizando el modelo especializado correspondiente, se han obtenido las puntuaciones asociadas a cada una de las titulaciones, las cuales se representan en la figura 4.13. Las siglas de las titulaciones se pueden consultar en los apéndices A.4 y A.7.

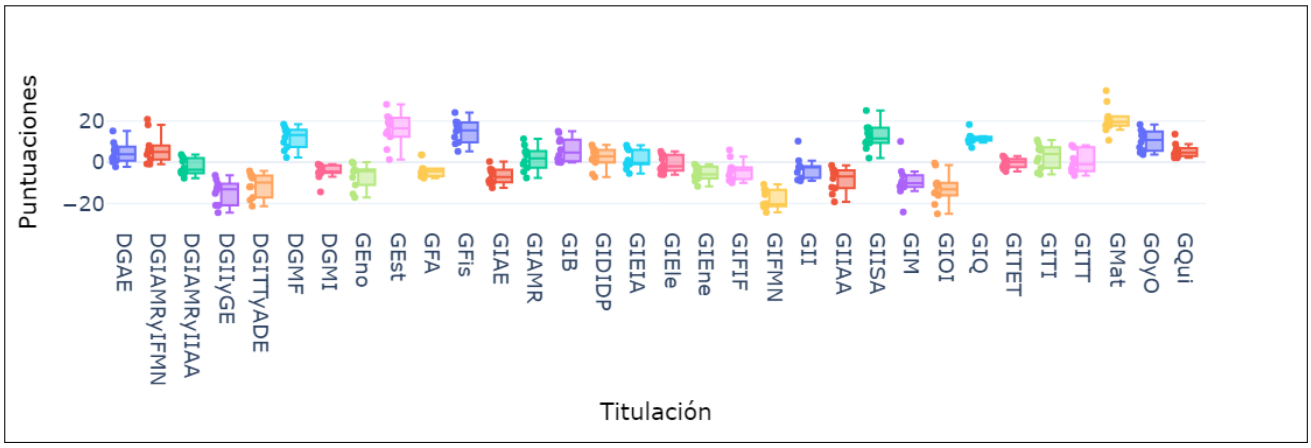

Figura 4.13: Puntuaciones asociadas a cada titulación para un estudiante del Grado en Estadística.

En la tabla 4.12 se pueden ver los porcentajes de coincidencia asociados a cada titulación a partir de las de las puntuaciones obtenidas para el anterior estudiante.

Titulación	**Coincidencia (%)**
DGAE	56,82
DGIAMRyIFMN	59,01
DGIAMRyIIAA	41,40
DGIIyGE	21,84
DGITTyADE	28,59
DGMF	75,88
DGMI	39,55
GEno	39,90
GEst	82,78
GFA	39,86
GFis	80,69
GIAE	34,02
GIAMR	52,78
GIB	58,60
GIDIDP	54,63
GIEIA	48,04

Titulación	**Coincidencia (%)**
GIEle	44,88
GIEne	36,51
GIFIF	41,02
GIFMN	6,96
GII	43,80
GIIAA	34,56
GIISA	72,42
GIM	27,99
GIOI	21,69
GIQ	71,79
GITET	49,03
GITI	56,90
GITT	46,75
GMat	88,97
GOyO	70,93
GQui	58,51

Tabla 4.12: Coincidencia asociada a cada titulación para un estudiante del Grado en Estadística.

En la tabla 4.13 vemos que la titulación que mayor coincidencia tiene es el Grado en Matemáticas, seguida del Grado en Estadística, lo que supone un acierto. Otras titulaciones con alta coincidencia son el Grado en Física, el Programa de Doble Titulación Oficial de Grado en Matemáticas y Grado en Física, el Grado en Ingeniería Informática de Servicios y Aplicaciones, el Grado en Ingeniería Química y el Grado en Óptica y Optometría. Estas titulaciones tienen una sólida base en fundamentos matemáticos y están relacionados con las ciencias.

Titulación	GMat	GEst	GFis	DGMF	GIISA	GIQ	GOyO
Coincidencia (%)	88,97	82,78	80,69	75,88	72,42	71,79	70,93

Tabla 4.13: Coincidencia asociada a las titulaciones que superan el umbral de decisión para un estudiante del Grado en Estadística.

Ciencias Sociales y Jurídicas

De esta rama de enseñanza académica se ha seleccionado un estudiante que accedió al Grado en Educación Primaria. Utilizando el modelo especializado correspondiente, se han obtenido las puntuaciones asociadas a cada una de las titulaciones, las cuales se representan en la figura 4.14. Las siglas de las titulaciones se pueden consultar en el apéndice A.5.

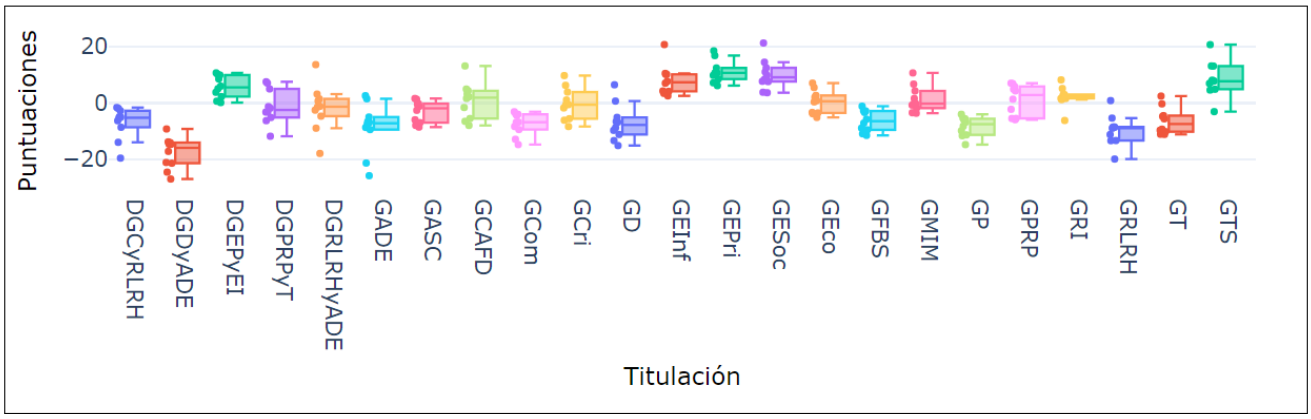

Figura 4.14: Puntuaciones asociadas a cada titulación para un estudiante del Grado en Educación Primaria.

En la tabla 4.14 se pueden ver los porcentajes de coincidencia asociados a cada titulación a partir de las de las puntuaciones obtenidas para el anterior estudiante.

Titulación	**Coincidencia (%)**
DGCyRLRH	50,23
DGDyADE	21,02
DGEPyEI	79,61
DGPRPyT	57,71
DGRLRHyADE	60,91
GADE	44,82
GASC	59,32
GCAFD	69,65
GCom	45,87
GCri	62,83
GD	43,25
GEInf	84,58

Titulación	**Coincidencia (%)**
GEPri	93,65
GESoc	89,41
GEco	66,19
GFBS	46,88
GMIM	63,79
GP	43,97
GPRP	72,20
GRI	71,01
GRLRH	40,51
GT	44,13
GTS	85,56

Tabla 4.14: Coincidencia asociada a cada titulación para un estudiante del Grado en Educación Primaria.

En la tabla 4.15 vemos que la titulación que mayor coincidencia tiene es el Grado en Educación Primaria, lo que supone un acierto en nuestro sistema. Otras titulaciones con alta coincidencia son el Grado en Educación Social, el Grado en Trabajo Social, el Grado en Educación Infantil y el Programa de Doble Titulación Oficial de Grado en Educación Primaria y Grado en Educación Infantil, entre otros. La relación que tienen estas titulaciones radica en su enfoque educativo y de trabajo social.

Titulación	GEPri	GESoc	GTS	GEInf	DGEPyEI	GPRP	GRI
Coincidencia (%)	93,65	89,41	85,56	84,58	79,61	72,2	71,01

Tabla 4.15: Coincidencia asociada a las titulaciones que superan el umbral de decisión para un estudiante del Grado en Educación Primaria.

Ciencias de la Salud

Finalmente, de esta rama de enseñanza académica se ha seleccionado un estudiante que accedió al Grado en Medicina. Utilizando el modelo especializado correspondiente, se han obtenido las puntuaciones asociadas a cada una de las titulaciones, las cuales se representan en la figura 4.15. Las siglas de las titulaciones se pueden consultar en el apéndice A.6.

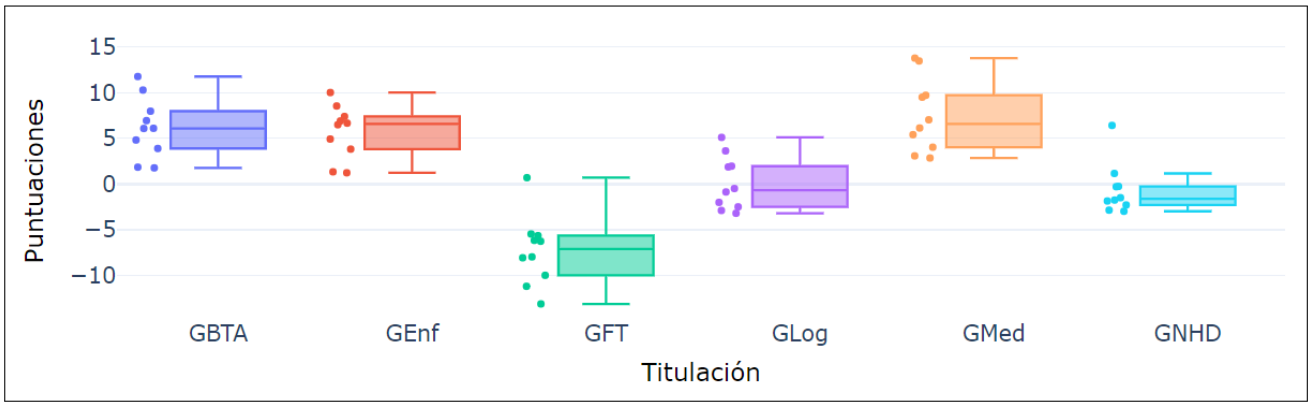

Figura 4.15: Puntuaciones asociadas a cada titulación para un estudiante del Grado en Medicina.

En la tabla 4.16 se pueden ver los porcentajes de coincidencia asociados a cada titulación a partir de las de las puntuaciones obtenidas para el anterior estudiante.

Titulación	**Coincidencia (%)**
GBTA	70,66
GEnf	72,56
GFT	19,02

Titulación	**Coincidencia (%)**
GLog	44,19
GMed	72,57
GNHD	40,49

Tabla 4.16: Coincidencia asociada a cada titulación para un estudiante del Grado en Medicina.

En la tabla 4.17 vemos que la titulación que mayor coincidencia tiene es el Grado en Medicina, lo que supone un acierto en nuestro sistema. Otras titulaciones con alta coincidencia son el Grado en Enfermería y el Grado en Biomedicina y Terapias Avanzadas. Su relación se basa en que están relacionadas con la sanidad.

Titulación	GMed	GEnf	GBTA
Coincidencia (%)	72,57	72,56	70,66

Tabla 4.17: Coincidencia asociada a las titulaciones que superan el umbral de decisión para un estudiante del Grado en Medicina.

Los resultados obtenidos sobre el conjunto de prueba para cada uno de los modelos especializados se pueden consultar en la tabla 4.18.

Modelo especializado	AyH	CIyA	CSyJ	CS
Tasa de aciertos (%)	73,39	46,32	62,65	77,41

Tabla 4.18: Tasas de aciertos de los modelos especializados sobre el conjunto de prueba.

Capítulo 5

Creación de un Dashboard

En este capítulo se detalla el proceso de creación de un *Dashboard* interactivo para la visualización de los datos de estudiantes de nuevo ingreso en la Universidad de Valladolid utilizados a lo largo del trabajo. Los gráficos son interactivos, lo que significa que al pasar el cursor sobre ellos muestran información adicional. Estos gráficos se han desarrollado utilizando la biblioteca *Highcharts* y se han integrado en una aplicación creada con la biblioteca *Streamlit*. Esto permite agregar diferentes componentes que hacen los gráficos modificables en función de las selecciones del usuario. Finalmente, la aplicación ha sido desplegada y está disponible a través del siguiente enlace https://david-aparicio-sanz-tfg-estadistica-2024.streamlit.app/.

Para una comprensión adecuada del funcionamiento del *Dashboard*, es esencial tener en cuenta las siguientes consideraciones para garantizar una interpretación precisa de la información representada:

1. Algunos datos podrían no reflejar completamente la realidad debido a dos posibles factores: el proceso de anonimización y problemas en la obtención o manipulación de datos por parte de los proveedores. Esto puede verse reflejado en la ausencia de ciertos campos en algunas observaciones, como la nota de admisión o la titulación, entre otros, lo cual nos ha supuesto dificultades a lo largo del trabajo.

2. Los datos de las titulaciones individuales incluyen la información de los programas de doble titulación. Esto implica que, por ejemplo, si una titulación individual indica un número de estudiantes, este número es la suma de los estudiantes de los programas individuales y los de doble titulación.

A continuación, explicamos las cuatro secciones de la aplicación.

5.1. Número de estudiantes de nuevo ingreso

En esta página nos enfocamos en visualizar el número de estudiantes de nuevo ingreso a lo largo de los diferentes cursos académicos, con el objetivo de ver la evolución de la población estudiantil. Para lograr este objetivo, hemos optado por utilizar un gráfico de series temporales, el cual representa el número de estudiantes de nuevo ingreso según el curso académico. Para proporcionar una experiencia personalizada y facilitar la exploración de los datos, hemos incorporado un selector que permite elegir una categoría específica de referencia entre {Centros, Titulaciones, Modalidades}. Dependiendo de la categoría seleccionada, podemos elegir las series que deseamos representar, lo que aporta flexibilidad. Esto es crucial, ya que si representáramos todas las series simultáneamente, el gráfico resultante sería difícil de interpretar debido al solapamiento entre las líneas. También es posible visualizar el total de estudiantes sin distinción de categoría. Junto al gráfico se incluye una tabla que contiene los datos representados y ofrece la opción de descargarlos en formato CSV. En el apéndice C.2 se incluye una figura que muestra la página.

5.2. Comparativas de género

En esta página analizamos la distribución de estudiantes según su género. Para este fin, hemos diseñado una combinación de gráficos que ofrece una visualización completa tanto con un enfoque temporal como global. Al igual que en la página anterior, hemos incluido opciones configurables que permiten seleccionar la categoría específica de referencia. Sin embargo, a diferencia de la página anterior, en esta ocasión solo podemos seleccionar un subgrupo perteneciente a la categoría. En base a la selección, se crea un diagrama de barras apiladas que muestra el número de estudiantes de nuevo ingreso hombres y mujeres por cada curso académico. Este gráfico nos permite observar las variaciones a lo largo del tiempo y detectar posibles patrones o tendencias. Al colocar el cursor sobre las barras, se muestra el número exacto de estudiantes de cada género, lo que permite una interpretación más precisa de los datos. Además, hemos incorporado un gráfico circular que presenta los porcentajes totales de hombres y mujeres, lo que proporciona una visión global de la distribución de género. Junto al gráfico se incluye una tabla que contiene los datos representados. En el apéndice C.3 se incluye una figura que muestra la página.

Test de proporciones

Las visualizaciones de esta página nos hacen cuestionar si hay diferencias significativas entre la proporción de hombres y la de mujeres en algunas titulaciones. Para verificar esto, realizamos un test de proporciones bilateral para cada titulación. Nuestra hipótesis nula es que la proporción de mujeres es igual a la de hombres.

- H_0: $p_M = \frac{1}{2}$

Estadístico de prueba

Para contrastar esta hipótesis, utilizamos el estadístico de prueba para una proporción, que se basa en la distribución normal estándar. Se calcula de la siguiente manera.

$$z = \frac{\hat{p}_M - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}} \tag{5.1}$$

Donde:

- $\hat{p}_M$ es la proporción muestral de mujeres.
- $p_0 = \frac{1}{2}$ es la proporción bajo la hipótesis nula.
- n es el tamaño de la muestra.
- p-valor = $2 \cdot \mathrm{P}(\mathrm{Z} > |\mathrm{z}|)$

Resultados

Hemos utilizado los datos desde el curso académico 2017-18 hasta el 2022-23 para comprobar si existen diferencias significativas en la proporción de estudiantes de nuevo ingreso según el género para cada una de las titulaciones disponibles. De las 71 titulaciones disponibles, el 14,08 % (10) no muestran diferencias significativas de género, mientras que en el 85,92 % restante (61) sí que se observan estas diferencias. Es interesante apreciar que dentro del grupo donde se observan diferencias, hay una distribución bastante equilibrada entre las titulaciones donde predominan las mujeres y aquellas donde predominan los hombres ya que en el 50,82 % (31) de las titulaciones hay más mujeres que hombres, mientras que en el 49,18 % (30) restante ocurre lo contrario. Esto sugiere una diversidad en las preferencias de elección de carrera según el género. Los resultados completos se pueden consultar en el apéndice D.1.

5.3. Notas de admisión

Esta página permite explorar la distribución y la evolución de la nota de admisión de las titulaciones a lo largo de los cursos académicos. Es importante tener en cuenta que la nota de corte de una titulación es la nota más baja de los estudiantes admitidos en un grado universitario para cada cupo de admisión. En nuestros datos, no podemos distinguir en todos los casos si los estudiantes accedieron a través de los cupos minoritarios (deportistas de alto nivel, estudiantes con discapacidad, titulados universitarios, mayores de 25 años o mayores de 40 y 45 años con experiencia laboral o profesional).

Al seleccionar una titulación académica específica, se muestran diversos gráficos que representan esta variable. La primera visualización es un gráfico de dispersión que representa las notas de admisión en función de cada curso académico. Para evitar que los puntos se superpongan se ha añadido un ligero ruido horizontal. Además, se pueden identificar los valores exactos de las notas al pasar el cursor sobre los puntos. En este caso, es necesario considerar que las titulaciones individuales incluyen información sobre las dobles titulaciones como se ha explicado anteriormente.

En contraste, el segundo gráfico, es un diagrama de cajas y bigotes que ilustra la distribución de las notas de admisión por curso académico. Este tipo de representación nos permite destacar características sobre la distribución de la nota de admisión, en lugar de visualizar valores individuales, como ocurre en el primer gráfico. Además, este diagrama nos permite identificar valores atípicos o *outliers*, es decir, aquellos datos que se encuentran significativamente alejados del resto. Para identificar los valores atípicos presentes en un curso académico específico de una titulación, se utiliza el método del rango intercuartílico. En muchos casos, los valores atípicos inferiores se corresponden a estudiantes admitidos a través de los cupos minoritarios. En el apéndice C.4 se incluyen varias figuras que muestran la página.

Test de rangos de *Wilcoxon Mann Whitney*

Las visualizaciones de esta página nos hacen cuestionar si las notas de admisión en el curso académico 2019-20 fueron significativamente menores que las del 2020-21, posiblemente debido a los cambios en las pruebas de acceso a la universidad motivadas por las circunstancias especiales derivadas de la pandemia de COVID-19. Con este fin hemos optado por emplear una prueba no paramétrica unilateral que nos permite contrastar dos muestras independientes. Para centrarnos en los datos del cupo general, hemos decidido eliminar los *outliers* inferiores a la nota de corte de dicho cupo.

Utilizamos el test de rangos de *Wilcoxon Mann Whitney* en lugar del test t de *Student* debido a que no podemos asumir normalidad en los datos. Esta falta de normalidad se puede comprobar en un histograma del *Dashboard*, el cual representa las notas de admisión para una titulación y curso académico específico. Para las titulaciones en las que las muestras cuentan con menos de 30 casos, se ha utilizado el cálculo del p-valor con la distribución exacta del estadístico, mientras que para aquellas con tamaños muestrales mayores, se ha utilizado la distribución asintótica. Planteamos las hipótesis unilaterales que nos permiten confirmar la sospecha en algunas titulaciones.

- H_0: La mediana de las notas de admisión del curso académico 2020-21 es menor o igual que la mediana de las notas de admisión del curso académico 2019-20.

Resultados

Globalmente podemos confirmar un aumento significativo en las notas de admisión durante el curso académico 2020-21, el contraste unilateral nos da un p-valor menor de 0.001, por lo que podemos confirmar el aumento significativo de la mediana de las notas de admisión del curso 20-21 respecto de las del curso 19-20. Además, vamos a realizar contrastes más específicos para cada una de las titulaciones. De las 71 titulaciones disponibles, existen 8 titulaciones de las que no disponemos de datos de las notas de admisión de uno o ambos cursos académicos, lo que impide utilizarlas en los contrastes. De las 63 titulaciones restantes, se observa que en un 39,68 % (25) rechazamos la hipótesis nula, lo que quiere decir que estas titulaciones han experimentado un aumento significativo en la nota de admisión en el curso académico 2020-21. Por otra parte, no podemos rechazar la hipótesis nula del 60,32 % (38) restante de las titulaciones, por lo que no podemos determinar que en estas titulaciones las notas de admisión hayan aumentado significativamente. Los resultados completos se pueden consultar en el apéndice D.2.

5.4. Otras visualizaciones

En esta página se comparan gráficamente una serie de variables entre las diferentes ramas de enseñanza académicas. Inicialmente se presentan una serie de indicadores que muestran la variación del número de estudiantes de nuevo ingreso desde el curso académico 2017-18 hasta el curso académico 2022-23 para cada una de las ramas de enseñanza académica, así como el total. Estos indicadores se complementan con un gráfico de áreas apiladas que muestra el porcentaje de estudiantes de nuevo ingreso según la rama de enseñanza académica. Todo esto proporciona una visión rápida de cómo ha fluctuado el número de estudiantes, pero estos cambios han sido muy ligeros. Utilizando los datos de estudiantes de nuevo ingreso desde el curso académico 2017-18 hasta el 2022-23, hemos seleccionado aquellos estudiantes de nuevo ingreso que no era la primera vez que accedían al Sistema Universitario Español, lo que significa que finalizaron unos estudios o los abandonaron y luego comenzaron otros. Teniendo esto en cuenta, hemos calculado la diferencia de años entre el curso académico del que disponemos los datos de acceso y el curso académico en el que accedieron por primera vez.

Hemos representado en un diagrama de barras la distribución del tiempo entre accesos al SUE por ramas de enseñanza académica. Se puede observar que los estudiantes que han accedido previamente al SUE y optan por volver a acceder a alguna titulación de “Ciencias” es frecuente que lo hagan al año siguiente de su primer acceso. En contraste, en otras ramas académicas, la frecuencia relativa del año siguiente a su primer acceso es menor, distribuyéndose en los años posteriores. También se incluye una tabla que contiene información relacionada con los estudiantes de nuevo ingreso que no era la primera vez que accedían al SUE según la rama de enseñanza académica.

Otro aspecto a visualizar es el porcentaje de estudiantes de nuevo ingreso que trabajan según la rama de enseñanza y el curso académico. Esta información se puede obtener a partir de la variable que identifica la ocupación del estudiante, permitiendo determinar si trabaja o no. Esto se representa en un gráfico de barras múltiples, que ilustra el porcentaje de estudiantes de nuevo ingreso que trabajan, categorizados por la rama de enseñanza académica y desglosados por año académico. La leyenda del gráfico permite habilitar o deshabilitar los años que se desea comparar, facilitando así un análisis más detallado. Además del gráfico, se incluye una tabla que contiene información de los estudiantes de nuevo ingreso que trabajan. La tabla muestra que la rama de enseñanza de "Ciencias" tiene el menor porcentaje de estudiantes de nuevo ingreso que trabajan sobre el total, mientras que "Ciencias Sociales y Jurídicas" registra el mayor porcentaje. El resto de ramas de enseñanza presentan porcentajes bastante similares entre sí.

La mayoría de los estudiantes de nuevo ingreso pertenecen a la categoría "Otra situación", ya que muchos de ellos se dedican exclusivamente al estudio y no trabajan, por eso es interesante ver la distribución de los que sí que trabajan. Utilizando el mismo tipo de gráfico que el anterior, para una rama de enseñanza determinada que puede ser seleccionada por el usuario, se representan las categorías que abarcan a los estudiantes que trabajan. Con el fin de mejorar la visualización, se han agrupado las ocupaciones en nuevas categorías que comparten similitudes entre sí. Esto simplifica la representación de los datos y facilita su interpretación. Al igual que en el gráfico anterior, la leyenda permite habilitar o deshabilitar los años que queramos comparar. En el apéndice C.5 se incluyen varias figuras que muestran la página.

Capítulo 6

Conclusiones y líneas futuras

En este capítulo se resumen las conclusiones obtenidas en el trabajo y se identifican las líneas de investigación futuras que permitirán aumentar los conocimientos en este campo de estudio.

6.1. Conclusiones

Se ha realizado un estudio utilizando el lenguaje de programación *Python* sobre los datos abiertos de estudiantes de nuevo ingreso en la Universidad de Valladolid, abarcando desde el curso académico 2017-18 hasta el 2022-23. El objetivo del estudio es identificar las características que influyen en el proceso de ingreso, realizar contrastes de hipótesis y visualizar los datos utilizados. Para ello, se han empleado diversas metodologías, como *Gradient Boosting* y redes neuronales tanto frecuentistas como bayesianas.

En relación al sistema de información para la elección de estudios universitarios, hemos analizado la importancia global de diversas características, determinando que las más relevantes son la especialidad de acceso, la nota de admisión y los estudios previos. Además, hemos desglosado esta importancia para cada modalidad académica, identificando tanto las variables influyentes como su impacto en los resultados según sus valores. Por ejemplo, observamos que notas de admisión altas suelen estar asociadas a titulaciones en el ámbito de las Ciencias, Ingeniería y Arquitectura o Ciencias de la Salud. Hemos visto que las mujeres tienen una mayor probabilidad de ingresar en titulaciones relacionadas con Ciencias Sociales y Jurídicas o Ciencias de la Salud. También hemos logrado cuantificar la incertidumbre asociada al proceso de admisión para los estudiantes. Es posible obtener un porcentaje de coincidencia asociado a cada titulación para un estudiante específico. Este porcentaje refleja otras titulaciones posibles a las que el estudiante podría haber ingresado, considerando los datos disponibles de otros estudiantes.

También hemos desarrollado un *Dashboard* que facilita la visualización de los datos empleados. Esta herramienta nos permite explorar los datos de manera detallada e interactuar con ellos. Estas visualizaciones nos han llevado a formular varias hipótesis que hemos contrastado mediante el test de proporciones y el test de *Wilcoxon-Mann-Whitney*. Hemos comprobado que existen diferencias significativas en las preferencias de elección de la titulación según el género y que las notas de admisión sufrieron un aumento significativo en el curso académico 2020-21.

6.2. Líneas futuras

A la vista de los resultados obtenidos, surgen varias vías de trabajo para el futuro:

- Mejorar el sistema de información para la elección de estudios universitarios incluyendo datos sobre la finalización o el abandono de los programas académicos por parte de los estudiantes. Con acceso a esta información, podríamos desarrollar un sistema de recomendación de estudios que ofreciese orientación basada en el conocimiento real de las trayectorias académicas de antiguos estudiantes, ayudando a los nuevos estudiantes a tomar decisiones más adecuadas a sus intereses y capacidades.
- Analizar si existen diferencias significativas en las notas de admisión obtenidas por los estudiantes que cursaron sus estudios previos a la universidad en distintas comunidades, dado que las pruebas de acceso a la universidad y los contenidos asociados pueden variar entre comunidades.
- Explorar otros conjuntos de datos disponibles en UniversiDATA. Por ejemplo, realizar comparaciones entre distintas universidades.

6.3. Limitaciones del estudio

Dado que estos datos han sido anonimizados para evitar la identificación directa o indirecta de los estudiantes, se han perdido valores de algunas variables para ciertos individuos, lo que ha complicado el trabajo. Sería útil que el portal de datos ofreciese los datos con estructuras alternativas, además del formato tabular. Por ejemplo, se podrían ofrecer listas para cada variable en función de las variables pivote. Esto significa que si se quieren consultar las notas de admisión para un curso académico y una titulación específica, se podrían obtener todos los valores sin perder datos, aunque se perdería la estructura tabular. Este enfoque sería útil para poder utilizar los datos completos en las visualizaciones y los contrastes de hipótesis.

Bibliografía

[1] UniversiDATA el portal de datos abiertos sobre educación superior. Último acceso el 15 de Mayo de 2024. https://www.universidata.es/

[2] Dashboard sobre el acceso de nuevos estudiantes a universidades españolas. Último acceso el 15 de Mayo de 2024. https://david-aparicio-sanz.shinyapps.io/UniversiDATA-2022_Proyecto/

[3] Python. Último acceso el 15 de Mayo de 2024. https://www.python.org/

[4] PyCharm. Último acceso el 15 de Mayo de 2024. https://www.jetbrains.com/es-es/pycharm/

[5] Project Jupyter. Último acceso el 15 de Mayo de 2024. https://jupyter.org/

[6] Pandas Python Data Analysis Library. Último acceso el 15 de Mayo de 2024. https://pandas.pydata.org/

[7] PyTorch. Último acceso el 15 de Mayo de 2024. https://pytorch.org/

[8] Pyro Deep Universal Probabilistic Programming. Último acceso el 15 de Mayo de 2024. https://pyro.ai/

[9] Highcharts. Último acceso el 15 de Mayo de 2024. https://www.highcharts.com/

[10] Streamlit. Último acceso el 15 de Mayo de 2024. https://streamlit.io/

[11] Gradient Boosting. Wikipedia, la Enciclopedia Libre. Último acceso el 15 de Mayo de 2024. https://en.wikipedia.org/wiki/Gradient_boosting

[12] Morgan, J. N. & Sonquist, J. A. (1963). Problems in the analysis of survey data, and a proposal. Journal of the American Statistical Association, Vol. 58, No. 302. Último acceso el 15 de Mayo de 2024. https://www.jstor.org/stable/2283276?origin=JSTOR-pdf

[13] Mantovani, R. G., Horváth, T., Rossi, A. L. D., Cerri, R., Barbon, S., Vanschoren, J. & De Carvalho, A. C. (2018). Better trees: an empirical study on hyperparameter tuning of classification decision tree induction algorithms. Data Mining And Knowledge Discovery. Último acceso el 15 de Mayo de 2024. https://doi.org/10.1007/s10618-024-01002-5

[14] Jeroen, V. H. & Vanschoren, J. (2021). Hyperboost: Hyperparameter Optimization by Gradient Boosting surrogate models. arXiv. Último acceso el 15 de Mayo de 2024. https://arxiv.org/abs/2101.02289

[15] He, Z., Lin, D., Lau, T. & Wu, M. (2019). Gradient Boosting Machine: A survey. arXiv. Último acceso el 15 de Mayo de 2024. https://arxiv.org/abs/1908.06951

[16] Masui, T. (2024). All You Need to Know about Gradient Boosting Algorithm - Part 1. Regression. Medium. Último acceso el 15 de Mayo de 2024. https://towardsdatascience.com/all-you-need-to-know-about-gradient-boosting-algorithm-part-1

[17] Historia de la estadística. Wikipedia, la Enciclopedia Libre. Último acceso el 15 de Mayo de 2024. https://es.wikipedia.org/wiki/Historia_de_la_estad%C3%ADstica#Estad%C3%ADsticas_bayesianas

[18] Van de Schoot, R., Kaplan, D. H., Denissen, J. J., Asendorpf, J. B., Neyer, F. J., & Van Aken, M. A. (2013). A gentle introduction to Bayesian Analysis: Applications to Developmental Research. Society for Research in Child Development. Último acceso el 15 de Mayo de 2024. https://doi.org/10.1111/cdev.12169

[19] Qinhgui Yu, J., Creager, E., Duvenaud, D. & Bettencourt, J. (s.f.). Bayesian Neural Networks. University of Toronto. Último acceso el 15 de Mayo de 2024. https://www.cs.toronto.edu/~duvenaud/distill_bayes_net/public/

[20] Introduction to Pyro. Pyro Tutorials 1.9.0 documentation. Último acceso el 15 de Mayo de 2024. https://pyro.ai/examples/intro_long.html

[21] Cornfield, J. (1967). Bayes Theorem. Review of the International Statistical Institute, Vol. 35, No. 1. Último acceso el 15 de Mayo de 2024. https://www.jstor.org/stable/1401634

[22] SVI Part I: An Introduction to Stochastic Variational Inference in Pyro. Pyro Tutorials 1.9.0 documentation. Último acceso el 15 de Mayo de 2024. https://pyro.ai/examples/svi_part_i.html

[23] Bernstein, M., N. (2020). The evidence lower bound (ELBO). Último acceso el 15 de Mayo de 2024. https://mbernste.github.io/posts/elbo/

[24] Kullback, S. & Leibler, R. (1951). On Information and Sufficiency. Último acceso el 15 de Mayo de 2024. https://www.jstor.org/stable/2236703?seq=1

[25] Dataset Acceso. UniversiDATA. Último acceso el 15 de Mayo de 2024. https://dimetrical.atlassian.net/wiki/spaces/UNC/pages/1608286213/Dataset+Acceso

[26] Dataset Titulaciones. UniversiDATA. Último acceso el 15 de Mayo de 2024. https://dimetrical.atlassian.net/wiki/spaces/UNC/pages/417529859/Dataset+Titulaciones

[27] Procesos de anonimización. UniversiDATA. Último acceso el 15 de Mayo de 2024. https://dimetrical.atlassian.net/wiki/spaces/UNC/pages/515571713/Procesos+de+Anonimizaci+n

[28] Oferta de grados Universidad de Valladolid. Último acceso el 15 de Mayo de 2024. https://www.uva.es/export/sites/uva/2.estudios/2.03.grados/.gabinete/

[29] Lundberg, S., & Lee, S. (2017). A Unified Approach to Interpreting Model Predictions. arXiv. Último acceso el 15 de Mayo de 2024. https://arxiv.org/abs/1705.07874

Apéndice A

Siglas

A.1. Generales

- **UVa**: Universidad de Valladolid.
- **ELBO**: Evidence Lower Bound.
- **KL**: Kullback-Leibler.
- **SVI**: Stochastic Variational Inference.
- **DNI**: Documento Nacional de Identidad.
- **API**: Application Programming Interface.
- **SHAP**: SHapley Additive exPlanations.
- **AdamW**: Adaptive Moment Estimation with Weight Decay.
- **SUE**: Sistema Universitario Español.

A.2. Ramas de enseñanza

- **AyH**: Artes y Humanidades.
- **CIA**: Ciencias, Ingeniería y Arquitectura.
- **CSyJ**: Ciencias Sociales y Jurídicas.
- **CS**: Ciencias de la Salud.

A.3. Titulaciones de Artes y Humanidades

- **GEsp**: Grado en Español: Lengua y Literatura.
- **GEC**: Grado en Estudios Clásicos.
- **GEI**: Grado en Estudios Ingleses.
- **GFil**: Grado en Filosofía.
- **GGPT**: Grado en Geografía y Planificación Territorial.
- **GH**: Grado en Historia.
- **GHA**: Grado en Historia del Arte.
- **GHCM**: Grado en Historia y Ciencias de la Música.
- **GLM**: Grado en Lenguas Modernas y sus Literaturas.
- **GTI**: Grado en Traducción e Interpretación.

A.4. Titulaciones de Ciencias

- **GEno**: Grado en Enología.
- **GEst**: Grado en Estadística.
- **GFis**: Grado en Física.
- **GMat**: Grado en Matemáticas.
- **GOyO**: Grado en Óptica y Optometría.
- **GQui**: Grado en Química.
- **DGAE**: Programa de Doble Titulación Oficial de Grado en Ingeniería de las Industrias Agrarias y Alimentarias y Grado en Enología.
- **DGMF**: Programa de Doble Titulación Oficial de Grado en Matemáticas y Grado en Física.
- **DGMI**: Programa de Doble Titulación Oficial de Grado en Matemáticas y Grado en Ingeniería Informática de Servicios y Aplicaciones.

A.5. Titulaciones de Ciencias Sociales y Jurídicas

- **GADE**: Grado en Administración y Dirección de Empresas.
- **GASC**: Grado en Antropología Social y Cultural.
- **GCAFD**: Grado en Ciencias de la Actividad Física y del Deporte.
- **GCom**: Grado en Comercio.
- **GCri**: Grado en Criminología.
- **GD**: Grado en Derecho.
- **GEco**: Grado en Economía.
- **GEInf**: Grado en Educación Infantil.
- **GEPri**: Grado en Educación Primaria.
- **GESoc**: Grado en Educación Social.
- **GFBS**: Grado en Finanzas, Banca y Seguros.
- **GMIM**: Grado en Marketing e Investigación de Mercados.
- **GP**: Grado en Periodismo.
- **GPRP**: Grado en Publicidad y Relaciones Públicas.
- **GRI**: Grado en Relaciones Internacionales.
- **GRLRH**: Grado en Relaciones Laborales y Recursos Humanos.
- **GTS**: Grado en Trabajo Social.
- **GT**: Grado en Turismo
- **DGCyRLRH**: Programa de Doble Titulación Oficial de Grado en Comercio y Grado en Relaciones Laborales y Recursos Humanos.
- **DGDyADE**: Programa de Doble Titulación Oficial de Grado en Derecho y Grado en Administración y Dirección de Empresas.
- **DGEPyEI**: Programa de Doble Titulación Oficial de Grado en Educación Primaria y Grado en Educación Infantil.

- **DGPRPyT**: Programa de Doble Titulación Oficial de Grado en Publicidad y Relaciones Públicas y Grado en Turismo.
- **DGRLRHyADE**: Programa de Doble Titulación Oficial de Grado en Relaciones Laborales y Recursos Humanos y Grado en Administración y Dirección de Empresas.

A.6. Titulaciones de Ciencias de la Salud

- **GBTA**: Grado en Biomedicina y Terapias Avanzadas.
- **GEnf**: Grado en Enfermería.
- **GFT**: Grado en Fisioterapia.
- **GLog**: Grado en Logopedia.
- **GMed**: Grado en Medicina.
- **GNHD**: Grado en Nutrición Humana y Dietética.

A.7. Titulaciones de Ingeniería y Arquitectura

- **GFA**: Grado en Fundamentos de la Arquitectura.
- **GIAE**: Grado en Ingeniería Agraria y Energética.
- **GIAMR**: Grado en Ingeniería Agrícola y del Medio Rural.
- **GIB**: Grado en Ingeniería Biomédica.
- **GIEle**: Grado en Ingeniería Eléctrica.
- **GIEne**: Grado en Ingeniería Energética.
- **GIFMN**: Grado en Ingeniería Forestal y del Medio Natural.
- **GIFIF**: Grado en Ingeniería Forestal: Industrias Forestales.
- **GII**: Grado en Ingeniería Informática.
- **GIISA**: Grado en Ingeniería Informática de Servicios y Aplicaciones.
- **GIM**: Grado en Ingeniería Mecánica.
- **GIQ**: Grado en Ingeniería Química.

- **GITET**: Grado en Ingeniería de Tecnologías Específicas de Telecomunicación.
- **GITT**: Grado en Ingeniería de Tecnologías de Telecomunicación.
- **GIIAA**: Grado en Ingeniería de las Industrias Agrarias y Alimentarias.
- **GIDIDP**: Grado en Ingeniería en Diseño Industrial y Desarrollo de Producto.
- **GIEIA**: Grado en Ingeniería en Electrónica Industrial y Automática.
- **GIOI**: Grado en Ingeniería en Organización Industrial.
- **GITI**: Grado en Ingeniería en Tecnologías Industriales.
- **DGIAMRyIFMN**: Programa de Doble Titulación Oficial de Grado en Ingeniería Agrícola y del Medio Rural y Grado en Ingeniería Forestal y del Medio Natural.
- **DGIAMRyIIAA**: Programa de Doble Titulación Oficial de Grado en Ingeniería Agrícola y del Medio Rural y Grado en Ingeniería de las Industrias Agrarias y Alimentarias.
- **DGIIyGE**: Programa de Doble Titulación Oficial de Grado en Ingeniería Informática y Grado en Estadística.
- **DGITTyADE**: Programa de Doble Titulación Oficial de Grado en Ingeniería de Tecnologías de Telecomunicación y Grado en Administración y Dirección de Empresas.

Apéndice B

Arquitecturas bayesianas

B.1. Artes y Humanidades

El modelo óptimo de "Artes y Humanidades" consta de 37.450 parámetros, de los cuales 1.216 son variables latentes como se puede ver en el resumen parámetrico de la figura B.1. Está formado por tres capas convolucionales bayesianas, que son result-1, result-4 y result-7.

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            result-1                [-1, 8, 80]              32
              ReLU-2                [-1, 8, 80]               0
         MaxPool1d-3                [-1, 8, 40]               0
            result-4               [-1, 16, 38]             400
              ReLU-5               [-1, 16, 38]               0
         MaxPool1d-6               [-1, 16, 19]               0
            result-7               [-1, 16, 17]             784
              ReLU-8               [-1, 16, 17]               0
            Linear-9                  [-1, 128]          34,944
             ReLU-10                  [-1, 128]               0
           Linear-11                   [-1, 10]           1,290
================================================================
Total params: 37,450
Trainable params: 36,234
Non-trainable params: 1,216
----------------------------------------------------------------
Input size (MB): 0.00
Forward/backward pass size (MB): 0.03
Params size (MB): 0.14
Estimated Total Size (MB): 0.17
----------------------------------------------------------------
```

Figura B.1: Resumen paramétrico de la red neuronal bayesiana de Artes y Humanidades.

B.2. Ciencias, Ingeniería y Arquitectura

El modelo óptimo de "Ciencias, Ingeniería y Arquitectura" consta de 254.016 parámetros, de los cuales 1.632 son variables latentes como se puede ver en el resumen parámetrico de la figura B.2. Está formado por dos capas convolucionales bayesianas, que son result-1 y result-4.

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            result-1               [-1, 16, 80]              64
              ReLU-2               [-1, 16, 80]               0
         MaxPool1d-3               [-1, 16, 40]               0
            result-4               [-1, 32, 38]           1,568
              ReLU-5               [-1, 32, 38]               0
         MaxPool1d-6               [-1, 32, 19]               0
            Conv1d-7               [-1, 64, 17]           6,208
              ReLU-8               [-1, 64, 17]               0
            Conv1d-9               [-1, 64, 15]          12,352
             ReLU-10               [-1, 64, 15]               0
           Conv1d-11               [-1, 64, 13]          12,352
             ReLU-12               [-1, 64, 13]               0
           Linear-13                  [-1, 256]         213,248
             ReLU-14                  [-1, 256]               0
           Linear-15                   [-1, 32]           8,224
================================================================
Total params: 254,016
Trainable params: 252,384
Non-trainable params: 1,632
----------------------------------------------------------------
Input size (MB): 0.00
Forward/backward pass size (MB): 0.10
Params size (MB): 0.97
Estimated Total Size (MB): 1.07
----------------------------------------------------------------
```

Figura B.2: Resumen paramétrico de la red neuronal bayesiana de Ciencias, Ingeniería y Arquitectura.

B.3. Ciencias Sociales y Jurídicas

El modelo óptimo de "Ciencias Sociales y Jurídicas" consta de 124.663 parámetros, de los cuales 1.632 son variables latentes como se puede ver en el resumen parámetrico de la figura B.3. Está formado por dos capas convolucionales bayesianas, que son result-1 y result-4.

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            result-1               [-1, 16, 80]              64
              ReLU-2               [-1, 16, 80]               0
         MaxPool1d-3               [-1, 16, 40]               0
            result-4               [-1, 32, 38]           1,568
              ReLU-5               [-1, 32, 38]               0
         MaxPool1d-6               [-1, 32, 19]               0
            Conv1d-7               [-1, 64, 17]           6,208
              ReLU-8               [-1, 64, 17]               0
         MaxPool1d-9                [-1, 64, 8]               0
           Conv1d-10                [-1, 64, 6]          12,352
             ReLU-11                [-1, 64, 6]               0
           Linear-12                  [-1, 256]          98,560
             ReLU-13                  [-1, 256]               0
           Linear-14                   [-1, 23]           5,911
================================================================
Total params: 124,663
Trainable params: 123,031
Non-trainable params: 1,632
----------------------------------------------------------------
Input size (MB): 0.00
Forward/backward pass size (MB): 0.08
Params size (MB): 0.48
Estimated Total Size (MB): 0.55
----------------------------------------------------------------
```

Figura B.3: Resumen paramétrico de la red neuronal bayesiana de Ciencias Sociales y Jurídicas.

B.4. Ciencias de la Salud

El modelo óptimo de "Ciencias de la Salud" consta de 72.534 parámetros, de los cuales 432 son variables latentes como se puede ver en el resumen parámetrico de la figura B.4. Está formado por dos capas convolucionales bayesianas, que son result-1 y result-4.

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            result-1                [-1, 8, 80]              32
              ReLU-2                [-1, 8, 80]               0
         MaxPool1d-3                [-1, 8, 40]               0
            result-4               [-1, 16, 38]             400
              ReLU-5               [-1, 16, 38]               0
         MaxPool1d-6               [-1, 16, 19]               0
            Conv1d-7               [-1, 32, 17]           1,568
              ReLU-8               [-1, 32, 17]               0
            Linear-9                  [-1, 128]          69,760
             ReLU-10                  [-1, 128]               0
           Linear-11                    [-1, 6]             774
================================================================
Total params: 72,534
Trainable params: 72,102
Non-trainable params: 432
----------------------------------------------------------------
Input size (MB): 0.00
Forward/backward pass size (MB): 0.03
Params size (MB): 0.28
Estimated Total Size (MB): 0.31
----------------------------------------------------------------
```

Figura B.4: Resumen paramétrico de la red neuronal bayesiana de Ciencias de la Salud.

Apéndice C

Imágenes del *Dashboard*

C.1. Inicio

Figura C.1: Inicio.

C.2. Número de estudiantes de nuevo ingreso

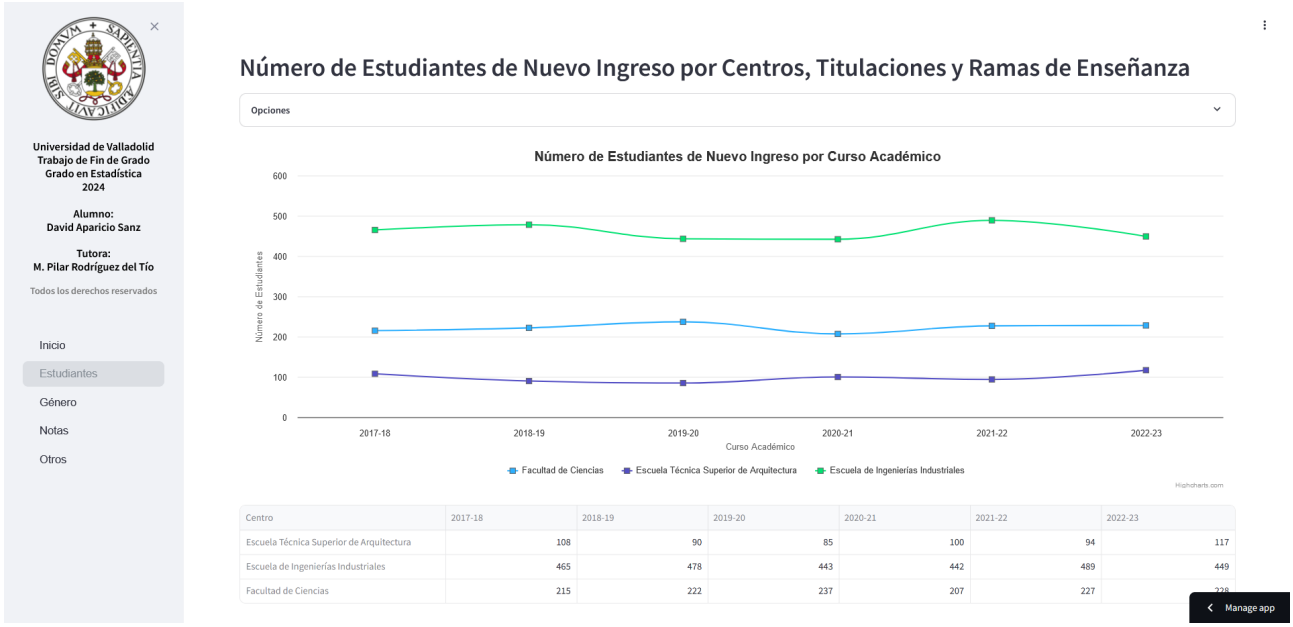

Figura C.2: Número de estudiantes de nuevo ingreso.

C.3. Comparativas de género

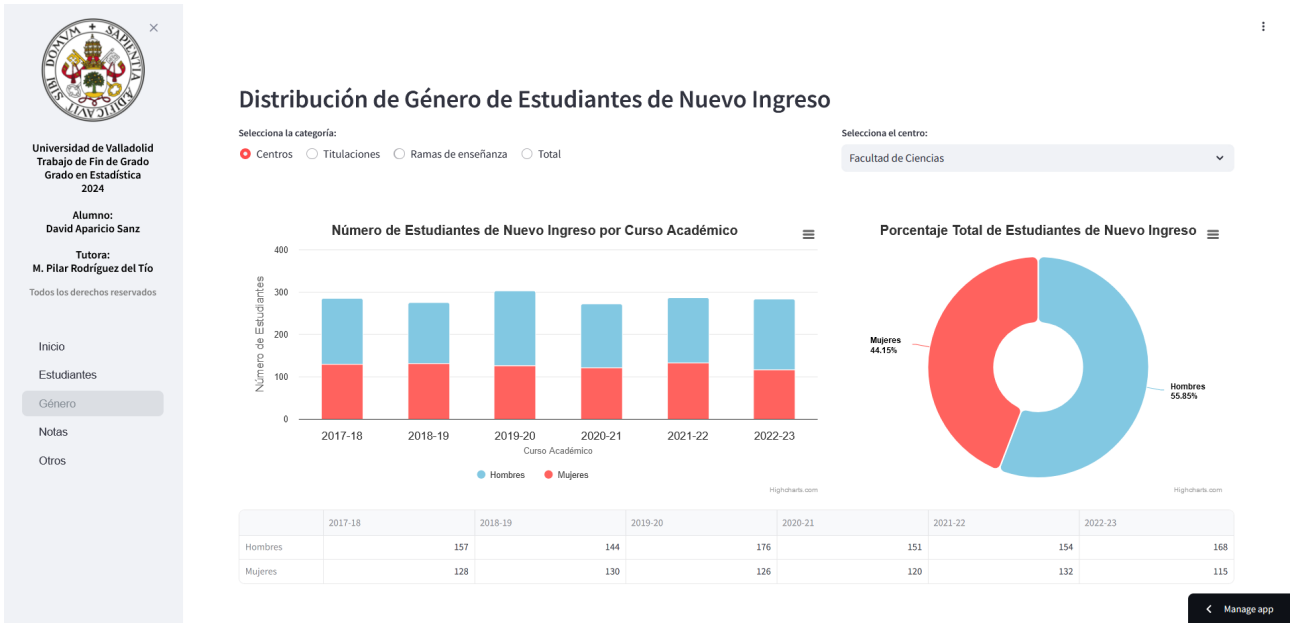

Figura C.3: Comparativas de género.

C.4. Notas de admisión

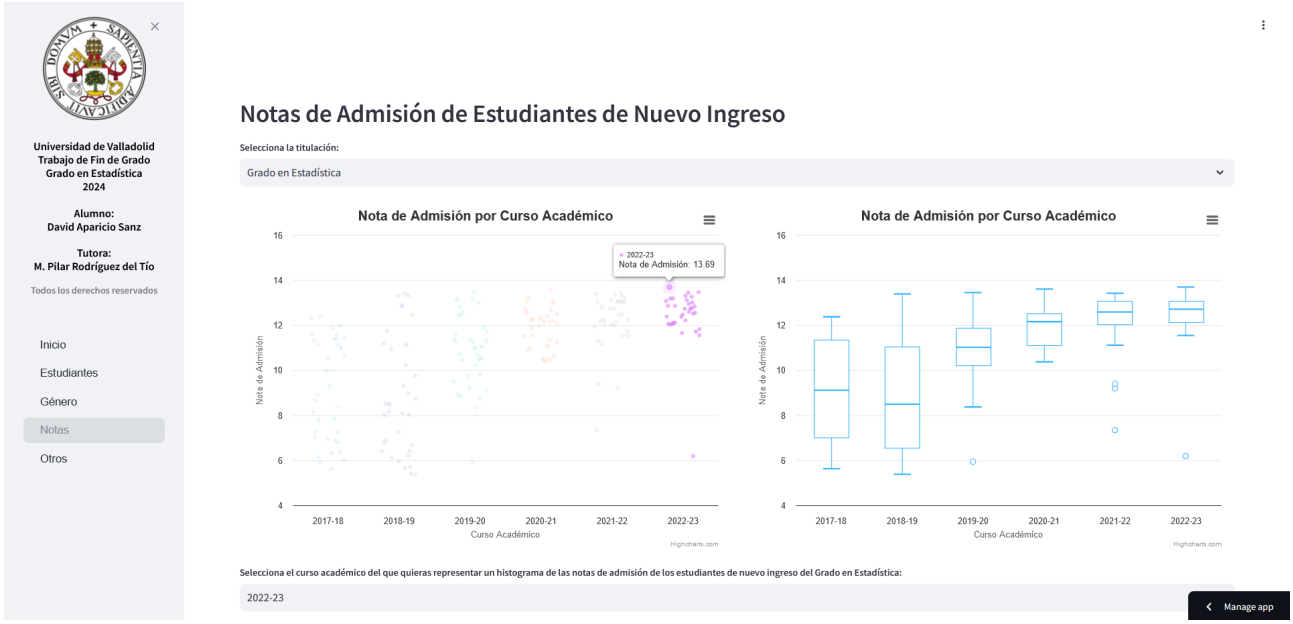

Figura C.4: Notas de admisión I.

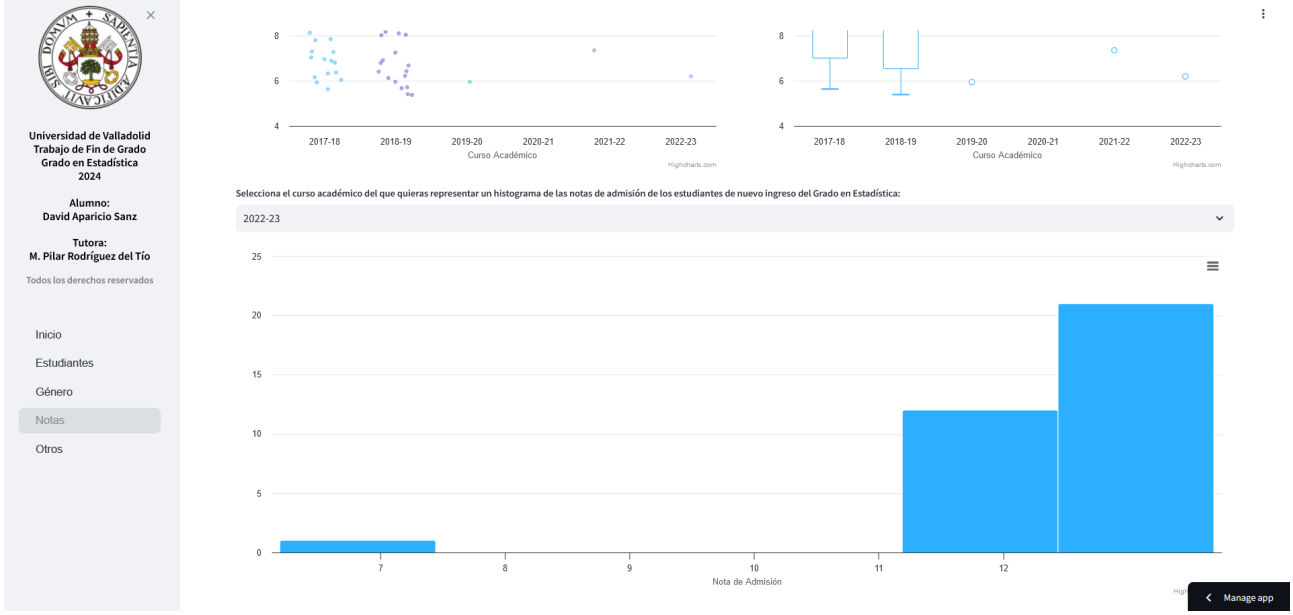

Figura C.5: Notas de admisión II.

C.5. Otras visualizaciones

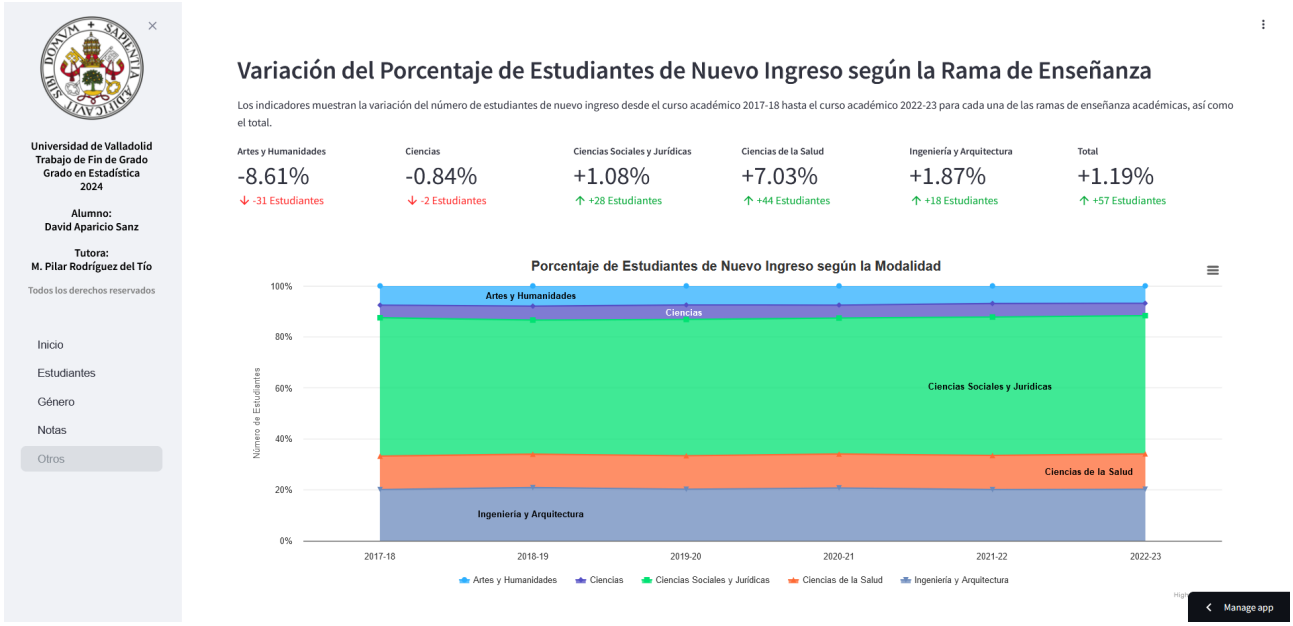

Figura C.6: Otras visualizaciones I.

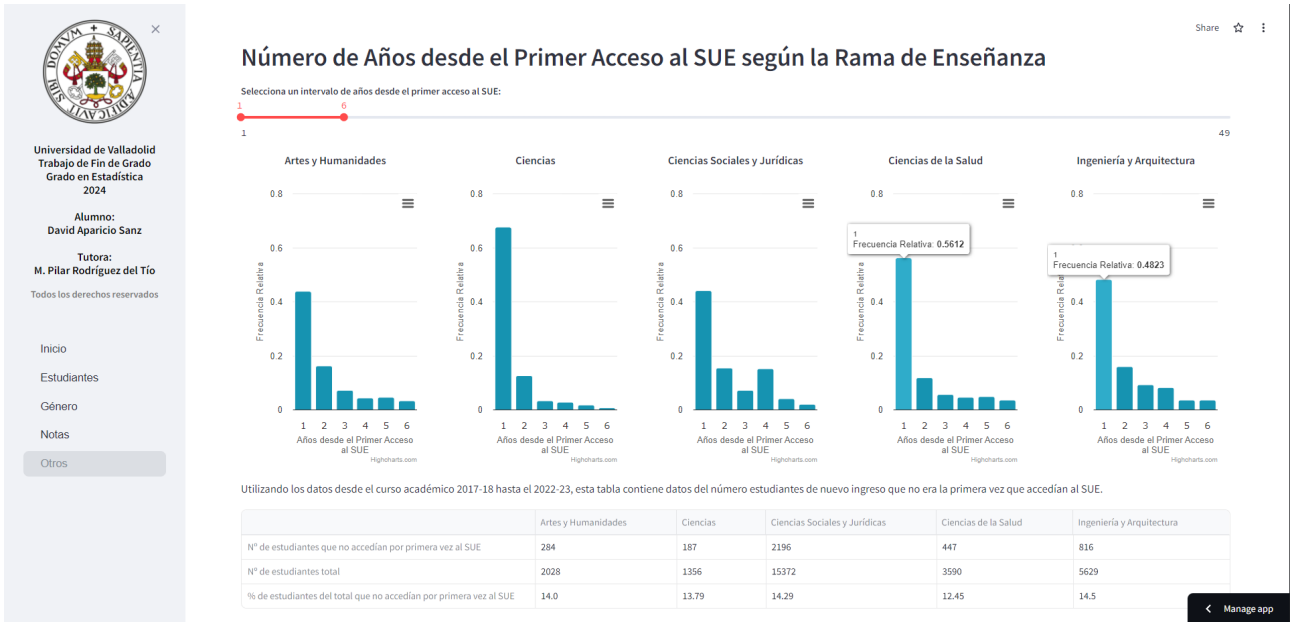

Figura C.7: Otras visualizaciones II.

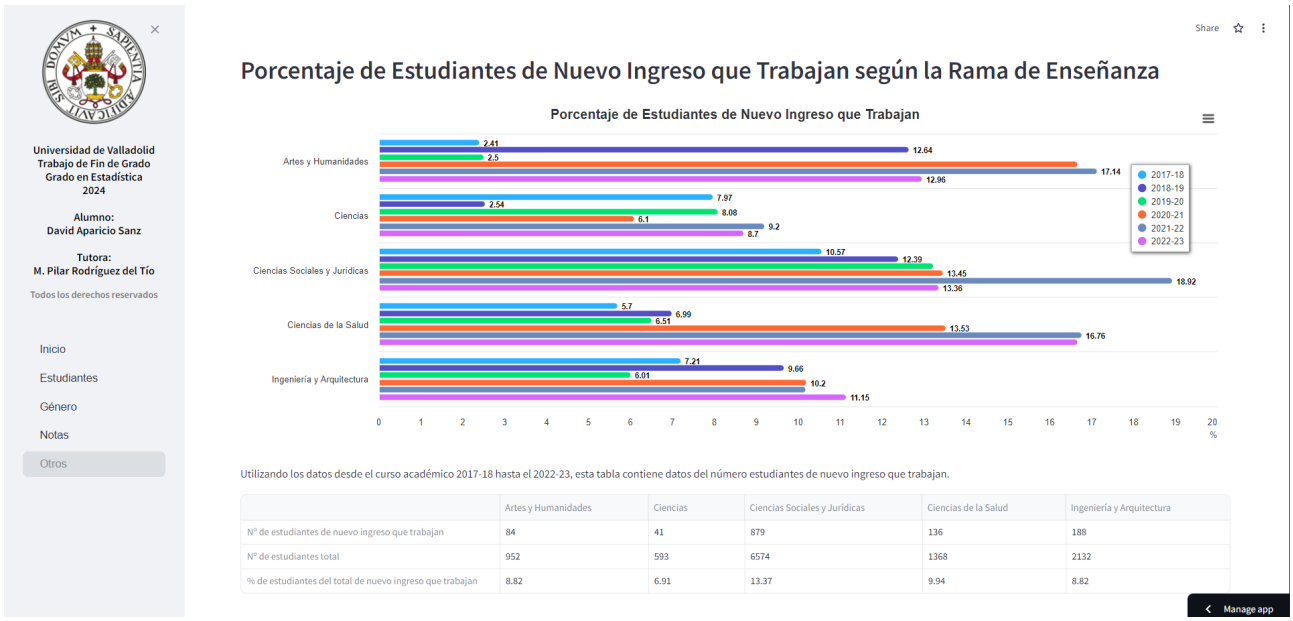

Figura C.8: Otras visualizaciones III.

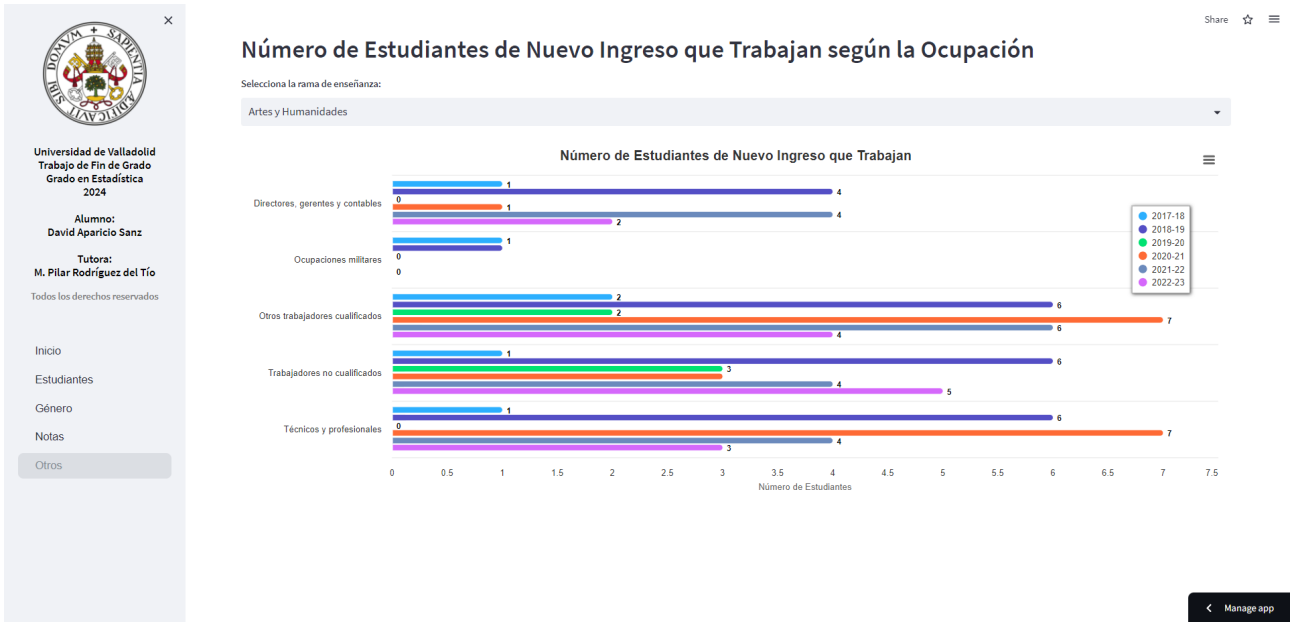

Figura C.9: Otras visualizaciones IV.

Apéndice D

Contrastes de hipótesis

La significancia estadística de los contrastes se determina según el p-valor tal que:

$$\text{Significancia} = \begin{cases} *** & \text{si p-valor} \leq 0{,}001 \\ ** & \text{si } 0{,}001 < \text{p-valor} \leq 0{,}01 \\ * & \text{si } 0{,}01 < \text{p-valor} \leq 0{,}05 \\ \bullet & \text{si } 0{,}05 < \text{p-valor} \leq 0{,}1 \\ & \text{si p-valor} > 0{,}1 \end{cases}$$

D.1. Test de proporciones

Titulación	**Hombres**	**Mujeres**	**Estadístico Z**	**Significancia**
GCri	73	96	1.7692	•
GEno	75	65	-0.8451	
GFil	81	73	-0.6446	
GHCM	62	55	-0.6471	
GIQ	141	133	-0.4832	
GIIAA	69	52	-1.5454	
GP	349	391	1.5439	
GQui	218	251	1.5237	
DGAE	12	14	0.3922	
DGRLRHyADE:	59	70	0.9684	

Tabla D.1: Titulaciones en las que no se rechaza la hipótesis nula de igualdad de proporciones de género.

Titulación	Hombres	Mujeres	Estadístico Z	Significancia
GADE	1140	915	-4.9633	***
GCAFD	103	26	-6.7794	***
GCom	685	493	-5.5940	***
GEco	265	140	-6.2112	***
GEst	153	66	-5.8789	***
GFBS	135	92	-2.8540	**
GFis	291	131	-7.7886	***
GGPT	58	19	-4.4444	***
GH	260	150	-5.4325	***
GIAE	62	0	-7.8740	***
GIAMR	166	49	-7.9793	***
GIEle	106	13	-8.5252	***
GIEne	81	24	-5.5626	***
GIFMN	141	57	-5.9696	***
GIFIF	50	0	-7.0710	***
GII	686	118	-20.0318	***
GIISA	271	54	-12.0369	***
GIM	635	93	-20.0878	***
GITET	176	50	-8.3813	***
GITT	269	70	-10.8081	***
GIEIA	415	119	-12.8091	***
GIOI	201	131	-3.8417	***
GITI	248	96	-8.1952	***
GMat	248	154	-4.6882	***
DGIAMRyIFMN	28	12	-2.5298	*
DGIAMRyIIAA	5	0	-2.2360	*
DGIIyGE	88	10	-7.8791	***
DGITTyADE	47	5	-5.8243	***
DGMF	62	7	-6.6212	***
DGMI	49	0	-7.0000	***

Tabla D.2: Titulaciones en las que se rechaza la hipótesis nula de igualdad de proporciones de género y hay más hombres que mujeres.

Titulación	Hombres	Mujeres	Estadístico Z	Significancia
GASC	0	5	2.2360	*
GBTA	17	69	5.6073	***
GD	663	1151	11.4577	***
GEInf	227	1793	34.8430	***
GEPri	1058	2032	17.5218	***
GESoc	42	431	17.8862	***
GEnf	310	1450	27.1736	***
GEsp	49	148	7.0534	***
GEC	40	62	2.1783	*
GEI	96	302	10.3258	***
GFT	128	181	3.01506	**
GFA	241	353	4.5954	***
GHA	38	147	8.01383	***
GIB	69	97	2.1732	*
GIDIDP	127	203	4.1836	***
GLM	0	89	9.4339	***
GLog	18	219	13.0563	***
GMIM	156	213	2.9673	**
GMed	344	850	14.6436	***
GNHD	68	173	6.7636	***
GPRP	327	890	16.1384	***
GRI	17	44	3.4569	***
GRLRH	320	546	7.6797	***
GTS	50	314	13.8373	***
GTI	70	315	12.4863	***
GT	59	184	8.0187	***
GOyO	40	149	7.9285	***
DGCyRLRH	54	79	2.1677	*
DGDyADE	123	209	4.7198	***
DGEPyEI	65	436	16.5750	***
DGPRPyT	21	121	8.3918	***

Tabla D.3: Titulaciones en las que se rechaza la hipótesis nula de igualdad de proporciones de género y hay más mujeres que hombres.

D.2. Test de rangos de *Wilcoxon Mann Whitney*

Titulación	Nº 2019-20	Nº 2020-21	Estadístico U	Significancia
GADE	321	320	49163.0	
GEno	25	28	259.5	•
GEsp	39	21	411.0	
GEC	4	20	31.0	
GEI	61	74	2241.0	
GFBS	34	33	663.5	
GGPT	18	15	99.0	
GH	74	65	2119.5	
GHA	25	31	374.0	
GHCM	19	17	158.0	
GIAE	8	18	58.0	
GIAMR	31	26	370.0	
GIB	38	40	607.5	•
GIEle	17	12	93.0	
GIFMN	32	32	509.0	
GIFIF	15	7	33.0	•
GIISA	46	53	1103.5	
GIM	104	104	5190.0	
GIQ	37	36	709.0	
GITET	31	37	516.0	
GIIAA	13	26	153.0	
GIEIA	98	83	3580.5	•
GITI	58	48	1491.0	
GLM	12	15	86.0	
GPRP	204	198	19870.5	
GQui	81	62	2479.5	
GRLRH	93	131	5327.0	•
GTS	58	59	1664.0	
GTI	62	69	2274.0	
GT	53	29	666.0	
DGDyADE	54	48	1163.5	
DGIAMRyIFMN	5	7	10.0	
DGIIyGE	13	17	102.0	

Tabla D.4: Titulaciones en las que no se confirma que haya un aumento de las notas de admisión en el curso 2020-21 mediante el contraste unilateral con el test de rangos de *Wilcoxon Mann Whitney* I.

Titulación	Nº 2019-20	Nº 2020-21	Estadístico U	Significancia
DGITTyADE	9	8	30.0	
DGAE	7	5	9.0	
DGMI	10	15	51.0	•
DGPRPyT	21	20	271.0	
DGRLRHyADE	28	14	237.0	

Tabla D.5: Titulaciones en las que no se confirma que haya un aumento de las notas de admisión en el curso académico 2020-21 mediante el contraste unilateral con el test de rangos de *Wilcoxon Mann Whitney* II.

Titulación	Nº 2019-20	Nº 2020-21	Estadístico U	Significancia
GCom	172	178	13187.0	*
GD	293	284	36005.0	**
GEco	61	58	1350.0	*
GEInf	326	322	48464.5	*
GEPri	501	470	106333.5	**
GESoc	79	74	2424.5	*
GEnf	267	263	13988.5	***
GEst	37	35	428.0	**
GFil	31	25	283.0	*
GFT	44	46	495.5	***
GFA	83	94	3082.0	**
GFis	67	62	1189.0	***
GII	128	132	5917.5	***
GITT	67	42	1107.5	*
GIDIDP	48	48	634.0	***
GIOI	56	49	616.5	***
GLog	42	39	387.0	***
GMIM	60	58	1067.5	***
GMat	69	60	1485.5	**
GMed	162	168	4905.0	***
GNHD	36	41	556.0	*
GP	120	119	5535.0	**
GOyO	29	27	154.0	***

Tabla D.6: Titulaciones en las que se confirma que hay un aumento de las notas de admisión en el curso académico 2020-21 mediante el contraste unilateral con el test de rangos de *Wilcoxon Mann Whitney* I.

Titulación	**Nº 2019-20**	**Nº 2020-21**	**Estadístico U**	**Significancia**
DGEPyEI	83	85	2636.0	**
DGMF	11	7	13.0	*

Tabla D.7: Titulaciones en las que se confirma que hay un aumento de las notas de admisión en el curso académico 2020-21 mediante el contraste unilateral con el test de rangos de *Wilcoxon Mann Whitney* II.

Apéndice E

Código

E.1. Bibliotecas

```
import numpy as np # 1.24.2
import pickle # 4.0
import plotly # 5.18.0
import pyro # 1.8.5
import sklearn # 1.4.0
import sklearn.metrics
import torch # 2.2.0+cu118
import torchsummary # 1.5.1
from tqdm import tqdm # 4.66.1
```

E.2. Extractor de datos

```
def load_universidata(universities, resource, years, requests_limit=1000000):
    universities = [universities] if not isinstance(universities, list) else universities
    years = [years] if not isinstance(years, list) else years
    with open('./API.json') as file:
        API = json.load(file)
    # URLs para acceder a la API de UniversiDATA
    URL_PATH = "https://www.universidata.es/api/action/datastore/search.json?resource_id="
    URL_LIMIT = "&limit=" + str(requests_limit)
    df = pd.DataFrame()
    for university, year in product(universities, years):
        # ID del recurso indicado
        URL_RESOURCE = API[university][resource][year]
        if URL_RESOURCE is None:
```

```
            errors += f"\n[ERROR 1] API {university} {resource} {year} not available."
            continue
        URL =  URL_PATH + URL_RESOURCE + URL_LIMIT
        for _ in range(MAX_ATTEMPTS):
            response = requests.get(URL, timeout=100)
            if response.status_code == 200:
                num_resources_loaded += 1
                break
        # Añadir los datos
        df = pd.concat([df, pd.DataFrame(response.json()["result"]["records"])])
    return df.reset_index(drop = True)
```

E.3. Imputación de valores faltantes

```
# División de los datos
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=1/3, stratify=y)
# Modelo inicial
best_parameterss = {
    "min_samples_split": 2,
    "min_samples_leaf": 1,
    "max_depth": 3,
    "max_leaf_nodes": None,
    "learning_rate": 0.1,
    "n_estimators": 100,
    "subsample": 1
}
model = GradientBoostingRegressor(**best_parameterss)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = np.mean(y_test == np.round(y_pred, 0))
# Modelo óptimo
new_parameters = {
    "min_samples_split": list(range(2, 5, 1)),
    "min_samples_leaf": range(1, 5, 1),
    "max_depth": range(1, 10, 1),
    "max_leaf_nodes": range(2, 20, 2),
    "learning_rate": np.arange(0.1, 1, 0.1),
    "n_estimators": range(50, 300, 50),
    "subsample": np.arange(0.1, 1.1, 0.1)
}
for parameter in new_parameters:
    final_results = pd.DataFrame(columns=["accuracy"])
    for idx, candidate in enumerate(new_parameters[parameter]):
```

```
        best_params[parameter] = candidate
        NROW, FOLDS = 10, 10
        results = pd.DataFrame(columns=range(FOLDS), index=range(NROW))
        # Repetición i
        for i in range(NROW):
            skf = StratifiedKFold(n_splits=FOLDS, shuffle=True)
            # Partición j
            for j, (train_index, test_index) in enumerate(skf.split(X, y)):
                X_train_aux, y_train_aux = X[train_index], y[train_index]
                X_test_aux, y_test_aux = X[test_index], y[test_index]
                model = GradientBoostingRegressor(**best_parameters)
                model.fit(X_train_aux, y_train_aux)
                y_pred = model.predict(X_test_aux)
                accuracy = np.mean(y_test_aux == np.round(y_pred, 0))
                results.iloc[i,j] = accuracy
                accuracy = np.mean(np.mean(results.dropna(), axis=1), axis=0)
        final_results.loc[idx,"accuracy"] = accuracy
    # Almacenar el mejor parámetro
    best_parameters[parameter] = new_parameters[parameter][final_results.idxmax()
        ["accuracy"]]
# Modelo final
model = GradientBoostingRegressor(**best_parameters)
model.fit(X, y)
y_pred = model.predict(X)
accuracy = np.mean(y == np.round(y, 0))
```

E.4. Creación de una red neuronal frecuentista

```
# División de los datos
X = df.drop(columns=["rama", "titulacion"]).to_numpy()
y = df["rama"].to_numpy().astype(int)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=1/3,
    stratify=y)
X_test, X_valid, y_test, y_valid = train_test_split(X_test, y_test, test_size=1/3,
    stratify=y_valid)
X_train = torch.tensor(X_train, dtype=torch.float32).unsqueeze(1)
y_train = torch.tensor(y_train, dtype=torch.long)
X_valid = torch.tensor(X_valid, dtype=torch.float32).unsqueeze(1)
y_valid = torch.tensor(y_valid, dtype=torch.long)
X_test = torch.tensor(X_test, dtype=torch.float32).unsqueeze(1)
y_test = torch.tensor(y_test, dtype=torch.long)
# Creación de la red
hyperparameters = {
```

```
    "epochs": 200,
    "batch": 64,
    "lr": 0.0001
}
class Dataset(torch.utils.data.Dataset):
    def __init__(self, features, labels):
        self.features = torch.Tensor(features).float()
        self.labels = torch.tensor(labels).long()
    def __len__(self):
        return self.features.shape[0]
    def __getitem__(self, idx):
        return self.features[idx], self.labels[idx]
train_dataset = Dataset(X_train, y_train)
valid_dataset = Dataset(X_test, y_test)
test_dataset = Dataset(X_valid, y_valid)
train_loader = torch.utils.data.DataLoader(train_dataset,
    batch_size=hyperparameters["batch"], shuffle=True)
valid_loader = torch.utils.data.DataLoader(valid_dataset,
    batch_size=hyperparameters["batch"], shuffle=True)
test_loader = torch.utils.data.DataLoader(test_dataset,
    batch_size=hyperparameters["batch"], shuffle=False)
# Definición de la arquitectura
class Model(torch.nn.Module):
    def __init__(self):
        super(Model, self).__init__()
        self.relu = torch.nn.ReLU()
        self.conv1 = torch.torch.nn.Conv1d(in_channels=1, out_channels=8, kernel_size=3)
        self.conv2 = torch.nn.Conv1d(in_channels=8, out_channels=16, kernel_size=3)
        self.pool = torch.nn.MaxPool1d(kernel_size=2)
        self.fc1 = torch.nn.Linear(320, 128)
        self.fc2 = torch.nn.Linear(128, 5)
        self.flat = torch.nn.Flatten()
    def forward(self, x):
        x = self.conv1(x)
        x = self.relu(x)
        x = self.pool(x)
        x = self.conv2(x)
        x = self.relu(x)
        x = self.pool(x)
        x = self.flat(x)
        x = self.fc1(x)
        x = self.relu(x)
        x = self.fc2(x)
        return x
```

```
model = Model()
criterion = torch.nn.CrossEntropyLoss()
optimizer = torch.optim.AdamW(model.parameters(), lr=hyperparameters["lr"])
# Entrenamiento de la red neuronal frecuentista
for epoch in range(hyperparameters["epochs"]):
    accuracy_value, loss_value = 0, 0
    model.train()
    for features, labels in train_loader:
        y_pred = model(features)
        loss = criterion(y_pred, labels)
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
        accuracy_value += torch.sum(torch.argmax(torch.softmax(y_pred, axis=1), axis=1)
            == labels).item()
        loss_value += loss.item()
    accuracy_value /= train_loader.dataset.__len__()
    loss_value /= train_loader.__len__()
```

E.5. Creación de una red neuronal bayesiana

```
# Selección de los datos para el modelo correspondiente
df = df.loc[df["rama"] == 0]
# División de los datos
X = df.drop(columns=["rama", "titulacion"]).to_numpy()
y = df["titulacion"].to_numpy().astype(int)
X_train, X_valid, y_train, y_valid = train_test_split(X, y, test_size=1/3,
    stratify=y)
X_valid, X_test, y_valid, y_test = train_test_split(X_valid, y_valid, test_size=1/3,
    stratify=y_valid)
# Los dataset y loader se crean igual que en la red neuronal frecuentista
hyperparameters = {
    "batch": 64,
    "learning_rate": 0.00001,
    "gamma": 0.1,
    "epochs": 200,
    "model_location": 0.0,
    "model_scale": 1.0,
    "guide_scale": 0.1,
    "particles": 10,
    "samples": 10,
    "train_valid_repetitions": 10,
    "freq_repetitions": 1,
```

```
    "test_repetitions": 100
}
# Definición de la arquitectura
class Model(pyro.nn.PyroModule):
    def __init__(self, device):
        super(Model, self).__init__()
        self.relu = torch.nn.ReLU()
        self.conv1 = pyro.nn.PyroModule[torch.nn.Conv1d](in_channels=1, out_channels=8,
            kernel_size=3)
        self.conv1.weight = weight_distribution(self.conv1, device,
            hyperparameters["model_location"], hyperparameters["model_scale"])
        self.conv1.bias = bias_distribution(self.conv1, device,
            hyperparameters["model_location"], hyperparameters["model_scale"])
        self.pool1 = torch.nn.MaxPool1d(kernel_size=2)
        self.conv2 = pyro.nn.PyroModule[torch.nn.Conv1d](in_channels=8, out_channels=16,
            kernel_size=3)
        self.conv2.weight = weight_distribution(self.conv2, device,
            hyperparameters["model_location"], hyperparameters["model_scale"])
        self.conv2.bias = bias_distribution(self.conv2, device,
            hyperparameters["model_location"], hyperparameters["model_scale"])
        self.pool2 = torch.nn.MaxPool1d(kernel_size=2)
        self.conv3 = pyro.nn.PyroModule[torch.nn.Conv1d](in_channels=16, out_channels=16,
            kernel_size=3)
        self.conv3.weight = weight_distribution(self.conv3, device,
            hyperparameters["model_location"], hyperparameters["model_scale"])
        self.conv3.bias = bias_distribution(self.conv3, device,
            hyperparameters["model_location"], hyperparameters["model_scale"])
        self.flat = torch.nn.Flatten()
        self.fc1 = torch.nn.Linear(in_features=16*17, out_features=128)
        self.fc2 = torch.nn.Linear(in_features=128, out_features=10)
    def forward(self, x, y=None):
        x = self.conv1(x)
        x = self.relu(x)
        x = self.pool1(x)
        x = self.conv2(x)
        x = self.relu(x)
        x = self.pool2(x)
        x = self.conv3(x)
        x = self.relu(x)
        x = x.view(x.shape[0], -1)
        x = self.fc1(x)
        x = self.relu(x)
        x = self.fc2(x)
        return x
```

```
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model = Model(device).to(device)
torchsummary.summary(model, (1, 82))
# Creación del SVI (Algoritmo de inferencia variacional)
guide = pyro.infer.autoguide.AutoDiagonalNormal(model,
    init_scale=hyperparameters["guide_scale"])
optimizer = pyro.optim.AdamW({"lr": hyperparameters["learning_rate"]})
criterion = pyro.infer.Trace_ELBO(num_particles=hyperparameters["particles"])
svi = pyro.infer.SVI(model, guide, optimizer, criterion)
# Entrenamiento de la red neuronal bayesiana
for epoch in range(hyperparameters["epochs"]):
    model.train()
    for features, labels in tqdm(train_loader):
        features, labels = features.to(device), labels.to(device)
        svi.step(features, labels)
    # Crear distribución posterior predictiva
    model.eval()
    predictive = pyro.infer.Predictive(model, guide=guide, return_sites=["_RETURN"],
        num_samples=hyperparameters["samples"], parallel=False)
    predictive.eval()
    # Evaluación del conjunto de entrenamiento
    with torch.no_grad():
        num_repetitions = hyperparameters["train_valid_repetitions"]
        accuracy_rep_value, loss_rep_value = [], []
        for rep in tqdm(range(num_repetitions)):
            model.eval()
            accuracy_value, loss_value = 0, 0
            for features, labels in train_loader:
                features, labels = features.to(device), labels.to(device)
                y_pred = predictive(features)
                y_pred_score = y_pred["_RETURN"]
                y_pred_softmax = torch.softmax(y_pred_score, axis=2)
                y_pred_labels = torch.argmax(torch.mean(y_pred_softmax, axis=0), axis=1)
                accuracy_value += torch.sum(y_pred_labels == labels).item()
                loss_value += svi.evaluate_loss(features, labels)
            # Calcular promedios
            accuracy_value /= train_loader.dataset.__len__()
            loss_value /= train_loader.__len__()
```

E.6. Contrastes de hipótesis

```
# Función que codifica los p valores en niveles de significancia
def significance(p):
```

```
    if p < 0.001:
        return "***"
    elif p < 0.01:
        return "**"
    elif p < 0.05:
        return "*"
    elif p < 0.1:
        return "."
    else:
        return ""
# Función que permite filtrar los outliers que están por debajo del límite inferior
# y eliminar los cupos minoritarios
def filtrar_outliers(datos):
    q1 = np.percentile(datos, 25)
    q3 = np.percentile(datos, 75)
    iqr = q3 - q1
    limite_inferior = q1 - 1.5 * iqr
    limite_superior = q3 + 1.5 * iqr
    datos_filtrados = [valor for valor in datos if limite_inferior <= valor]
    return datos_filtrados
#------ Test de proporciones ------
# Proporción hipotética
p_0 = 0.5
# Valor crítico para un nivel de significancia del 5%, prueba de dos colas
z_critico = scipy.stats.norm.ppf(1 - 0.05 / 2)
df = pd.DataFrame(columns=["Titulación", "Hombres", "Mujeres", "Estadístico Z", "P-Valor",
    "H0"])
# Contraste para cada titulación
degrees = sorted(np.unique(df_h.index.tolist() + df_m.index.tolist()))
for degree in degrees:
    # Número de hombres, mujeres y total
    h = df_h.loc[degree].sum() if degree in df_h.index else 0
    m = df_m.loc[degree].sum() if degree in df_m.index else 0
    n = h + m
    # Proporción muestral
    p_hat = m / n
    # Estadístico Z
    z = (p_hat - p_0) / np.sqrt(p_0 * (1 - p_0) / n)
    # P-Valor
    p_valor = 2 * (1 - scipy.stats.norm.cdf(abs(z)))
    df.loc[len(df),:] = {"Titulación": degree, "Hombres": h, "Mujeres": m, "Estadístico Z":
        z, "P-Valor": significance(p_valor), "H0": "No rechazo" if abs(z) < z_critico else
            "Rechazo"}
df = df.sort_values(by="Titulación")
```

```
#------ Test de rangos de Wilcoxon Mann Whitney ------
df = pd.DataFrame(columns=['Titulación', "Casos 2019-20", "Casos 2020-21", "Estadístico U",
    'P-Valor', 'H0'])
datos_filtrados_1920 = []
datos_filtrados_2021 = []
# Contraste para cada titulación
lista1 = df_notas_19_20["titulacion"].unique().tolist()
lista2 = df_notas_20_21["titulacion"].unique().tolist()
diferentes_lista1 = [elemento for elemento in lista1 if elemento not in lista2]
diferentes_lista2 = [elemento for elemento in lista2 if elemento not in lista1]
degrees = [elemento for elemento in lista1 if elemento in lista2]
for degree in degrees:
    poblacion1 = filtrar_outliers(df_notas_19_20.loc[df_notas_19_20["titulacion"] ==
        degree,]["nota_admision"].tolist())
    poblacion2 = filtrar_outliers(df_notas_20_21.loc[df_notas_20_21["titulacion"] ==
        degree,]["nota_admision"].tolist())
    datos_filtrados_1920.extend(poblacion1)
    datos_filtrados_2021.extend(poblacion2)
    if len(poblacion1) < 30 and len(poblacion2) < 30:
        # Realizar la prueba de una cola exacta
        stat, p_valor = scipy.stats.mannwhitneyu(poblacion1, poblacion2,
            alternative='less', method='exact')
    else:
        # Realizar la prueba de una cola asintótica
        stat, p_valor = scipy.stats.mannwhitneyu(poblacion1, poblacion2,
            alternative='less', method='asymptotic')
    df.loc[len(df),:] = {'Titulación': degree, "Casos 2019-20": len(poblacion1),
    "Casos 2020-21": len(poblacion2), "Estadístico U": stat, 'P-Valor':
        significance(p_valor), 'H0': "Rechazo" if p_valor < 0.05 else "No rechazo"}
df = df.sort_values(by='Titulación')
# Contraste global
stat, p_valor = scipy.stats.mannwhitneyu(datos_filtrados_1920, datos_filtrados_2021,
    alternative='less', method='asymptotic')
stat, p_valor, significance(p_valor)
```