Universidad Autónoma Metropolitana
Unidad Iztapalapa
División de Ciencias Básicas e Ingeniería

Posgrado en Ciencias
(Matemáticas Aplicadas e Industriales)

# La Ciencia de Datos y Análisis Multivariante aplicados al análisis y predicción de diabetes

T E S I S

QUE PARA OPTAR POR EL GRADO DE:

**Maestro en Ciencias**
**(Matemáticas Aplicadas e Industriales)**

PRESENTA:

**Eduardo Antonio Santiago Toledo**
**Matrícula: 2202800086**
**Correo: antonio_santi91@hotmail.com**

ASESORES:
Dra. Blanca Rosa Pérez Salvador
Dr. Asael Fabian Martínez Martínez

JURADO:
PRESIDENTE: DR. JOAQUIN DELGADO FERNANDEZ
SECRETARIO: DR. ASAEL FABIAN MARTINEZ MARTINEZ
VOCAL: DRA. BLANCA ROSA PEREZ SALVADOR
VOCAL: DRA. LIZBETH NARANJO ALBARRAN

Iztapalapa, Ciudad de México, 14 de diciembre de 2022

*DEDICATORIA*

*A mis padres por brindarme siempre su apoyo incondicional en cada una de mis metas.*

# Índice general

Capítulo 1

# Introducción

La diabetes mellitus es un grupo de enfermedades caracterizadas por hiperglucemia como consecuencia de defectos en la secreción y/o acción de la insulina. La hiperglucemia crónica está asociada con lesiones a largo plazo en diversos órganos, particularmente ojos, riñón, nervios, vasos sanguíneos y corazón. Diversos procesos patológicos están involucrados en el desarrollo de diabetes mellitus, pero la mayor parte de los casos pertenecen a dos categorías. La primera categoría es llamada diabetes mellitus tipo 1, esta es causada por una deficiencia absoluta en la secreción de insulina. En la segunda categoría, llamada diabetes mellitus tipo 2, la causa es una combinación de resistencia a la acción de la insulina (generalmente ocasionada por la obesidad) y una inadecuada respuesta secretora compensatoria. [1]

En el presente proyecto, los marcos de datos con los cuales estaremos trabajando serán exclusivos de la diabetes tipo 2.

La diabetes tipo 2 (DT2) es un problema a nivel mundial. Es una enfermedad compleja caracterizada por defecto en la secreción de insulina por parte del páncreas y por resistencia a la insulina, lo que contribuye a tener concentraciones altas de glucosa en sangre. El aumento en la glucosa llega a desencadenar complicaciones tales como; retinopatía, neuropatía, pie diabético y amputaciones. De igual forma está sumamente relacionada con el riesgo de aterosclerosis, lo cual puede desencadenar eventos cardiovasculares y cerebrovasculares. [2]

La Federación Internacional de Diabetes estimó para el 2019 una prevalencia de diabetes a nivel mundial de 9.3 %. Esta prevalencia se corresponde a 463 millones de adultos con diabetes y se calcula que esta cifra aumente a 700 millones para el año 2045, con una prevalencia de 10.9 % (11.1 % en hombres y 10.8 % en mujeres). La prevalencia es mayor en la población urbana con un 10.9 % que en la rural con un 7.2 %. Por otro lado se sabe que cada una de cada dos personas con diabetes no sabe que tiene la enfermedad. La mortalidad reportada a nivel mundial para el año 2019 fue de 4.2 millones de personas con un gasto estimado de 760.3 mil millones de dólares, el cual se

prevé aumentará a 845 mil millones de dólares para el año 2045. [2]

En el caso de México, la prevelencia de diabetes ha seguido la tendencia mundial de aumento. En 1993 se reportó una prevalencia de 6.7 %, que para el año 2006 aumentó a más del doble, cuando se estimó que 7.3 millones de personas vivían con la enfermedad, representando el 14.4 % de la población (7.3 % con diagnóstico previo y 7.1 % recién diagnosticada), siendo la proporción mayor en hombres (15.8 %) que en mujeres (13.2 %). Por otro lado, la Federación Internacional de Diabetes reportó que en 2019 existían en México 12.8 millones de personas con diabetes, y proyectó que para el año 2045 la cantidad de personas con diabetes aumentará a 22.9 millones. En la actualidad la diabetes ocupa el segundo lugar como causa de muerte en nuesto país, tanto en hombres como en mujeres, totalizando un total de 104,352 muertes en 2019. [2]

Se ha observado que la diabetes es una de las comorbilidades más frecuentes en personas con COVID-19, los diabéticos infectados con SARS-CoV-2 tienen una mayor tasa de admisión hospitalaria, neumonía severa y mayor mortalidad en comparación con pacientes no diabéticos. Reciente evidencia ha mostrado que el SARS-CoV-2 es capaz de producir daños directos al páncreas, lo cual puede empeorar la hiperglucemia e incluso inducir la aparición de diabetes en personas no diabéticas. [3]

Dado el gran número de personas con diabetes en las sociedades actuales, y el hecho de que algunas personas no saben que tienen diabetes, diagnosticar la diabetes es de particular importancia. La detección oportuna de la enfermedad puede reducir el número de pacientes diabéticos que padecen COVID-19. En este contexto, cobra especial importancia el proponer un modelo de predicción que permita diagnosticar y pronosticar a las personas que padecen la enfermedad o que están en riesgo de padecerla en base a un grupo de variables explicativas con suficiente antelación para que reciban el tratamiento médico adecuado.

La Ciencia de Datos, es una herramienta valiosa para el reconocimiento temprano de enfermedades y el apoyo en la monitorización del estado del paciente. Puede aumentar la confiabilidad de la cura y la toma de decisiones mediante el desarrollo de sistemas y algoritmos útiles. Los trabajadores de la salud, especialmente las enfermeras y los médicos, están sobrecargados de trabajo debido a un aumento masivo e inesperado en el número de pacientes durante la pandemia de coronavirus. En tales situaciones, las técnicas de Ciencia de Datos podrían usarse para diagnosticar a un paciente con enfermedades potencialmente mortales. En particular, las enfermedades que aumentan el riesgo de hospitalización y muerte en pacientes con coronavirus, como presión arterial alta, enfermedades cardíacas y diabetes, deben diagnosticarse en una etapa temprana. Si se es capaz de diagnosticar la diabetes en las primeras etapas de la enfermedad, se pueden recomendar medidas preventivas a pacientes de alto riesgo, y de este modo reducir su riesgo de infectarse con el coronavirus. [4]

## 1.1. Análisis y predicción de diabetes

En medicina, el diagnóstico de diabetes se basa en la glucemia en ayunas, la tolerancia a la glucosa y los niveles aleatorios de glucemia. Cuanto antes se obtenga el diagnóstico, más fácil puede controlarse. El aprendizaje automático (el cual se enmarca dentro del contexto más general de la Ciencia de Datos) puede ayudar a las personas a realizar un juicio preliminar sobre la diabetes mellitus de acuerdo con los datos de su examen físico diario, y puede servir como referencia para los médicos. [5]

Para reducir el número de muertes atribuibles a la diabetes, es fundamental que se diseñen métodos y técnicas que ayuden al diagnóstico precoz de la diabetes, ya que un gran número de muertes en pacientes diabéticos se deben a un diagnóstico tardío.

El aprendizaje automático se ocupa del desarrollo de tecnologías que permiten que las máquinas aprendan. El desafío es crear algoritmos que puedan tomar un grupo de patrones (en un rango más amplio, el conocimiento existente) y hacer automáticamente nuevas inferencias a partir de la información inicial, con o sin intervención humana. Desde la perspectiva del aprendizaje automático, la clasificación es el problema de identificar un conjunto de observaciones en varias categorías, en función del resultado del entrenamiento de un subconjunto de observaciones cuya categoría de pertenencia se conoce. [7]

En el presente proyecto, como modelo de aprendizaje automático principal, tendremos el modelo de regresión logística. La regresión logística es un modelo de clasificación en aprendizaje automático ampliamente utilizado en análisis clínicos. Utiliza estimaciones probabilísticas que ayudan a comprender la relación entre la variable dependiente y una o más variables independientes. La regresión logística, es ampliamente empleada en el marco del aprendizaje automático debido a su eficacia y simplicidad. Con este modelo no es necesario el disponer de grandes recursos computacionales, contando además con la ventaja de que sus resultados suelen ser altamente interpretables respecto a otras técnicas de clasificación.

## 1.2. Metodología y Modelo

En el presente trabajo, se diseña un modelo de predicción que predice si un paciente tiene diabetes en función de ciertas medidas de diagnóstico incluidas en el conjunto de datos y se exploran varias técnicas de preprocesamiento de datos para mejorar el rendimiento y la precisión.

Uno de los problemas más comunes encontrados al explorar modelos de aprendizaje automático, fue que muchos adolecen del problema de la fuga de datos. En este sentido,

en todos los modelos implementados se fue riguroso en aplicar las técnicas de preprocesamiento de datos y transformaciones de datos, de modo tal que se evitara totalmente la fuga de datos. De igual forma, debido al problema específico que se nos presenta, el cual es un marco de datos con clases desbalanceadas, se hizo uso del método Stratified cross-validation para evaluar el rendimiento de los modelos construidos.

La **primera fase** del presente proyecto fue analizar de forma rigurosa (evitando todo tipo de fuga de datos y a través de la técnica Stratified cross-validation), el enfoque de utilizar el algoritmo de K-means para extraer patrones, previo a la clasificación por regresión logística, para esto, se implementó el flujo de trabajo propuesto en [6], el cual aplicado con rigor, se ejemplifica con el Modelo 1 (ver **sección 3.4.2**).

Por otro lado, en la **segunda fase** del proyecto, nos enfocamos en diseñar un modelo que a través de técnicas de preprocesamiento de datos permitieran mejorar el rendimiento del clasificador de regresión logística. Para esto, fueron analizadas técnicas de imputación y limpieza de datos, y fueron diseñadas diversas técnicas de creación automatizada de características. Finalmente, el modelo final construido es el observado en la **sección 3.4.3.5**.

Los capítulos del proyecto, quedan estructurados de la siguiente manera, en:

**2. Preliminares**: Se da el marco teórico de las diversas técnicas utilizadas en el proyecto. En esta sección, se analiza el Flujo de trabajo en Machine Learning (**sección 2.2**), la técnica de reducción de dimensionalidad Análisis de Componentes Principales (**sección 2.3**), el Análisis de Clústeres (**sección 2.4**), el Modelo de Regresión Logística (**sección 2.5**) y finalmente en Evaluación del Modelo (**sección 2.6**), se estudian los métodos y métricas de evaluación a ser utilizados en el proyecto.

**3. Diseño del Modelo de predicción de Diabetes en base al clasificador de Regresión Logística**: En esta sección se explican los conjuntos de datos sobre los que trabajaremos (**sección 3.2**), la metodología para evaluar los modelos implementados (**sección 3.3**), y finalmente en la **sección 3.4**, se construyen los modelos predictivos.

**4. Discusión y conclusiones**: En este capítulo, se discuten aspectos importantes del modelo final construido y se dan las conclusiones del proyecto.

Capítulo 2

# Preliminares

## 2.1. Introducción

La Ciencia de Datos, es un campo interdisciplinario que involucra métodos científicos, procesos y sistemas para extraer conocimiento o un mejor entendimiento de datos en sus diferentes formas, ya sea estructurados o no estructurados, lo cual es una continuación de algunos campos de análisis de datos como la estadística, la minería de datos, el aprendizaje automático y la analítica predictiva. Todavía no hay consenso sobre la definición de Ciencia de Datos. [8]

El aprendizaje automático, por otro lado, se refiere a un grupo de técnicas utilizadas por los científicos de datos que permiten que las computadoras aprendan de los datos. Estas técnicas producen resultados que funcionan bien sin reglas explícitas de programación.

El aprendizaje automático, implica crear un modelo que se entrena con ciertos datos de entrenamiento y que luego pueda procesar datos externos al ajuste del modelo para hacer predicciones. El presente proyecto, busca crear un modelo de predicción centrado en el modelo estadístico de la regresión logística. Pero, antes de entrar en más detalle con el flujo de trabajo del aprendizaje automático, veamos primero como es que se presentan generalmente los datos multivariantes y definamos algunos conceptos que nos serán de utilidad más adelante. [13]

Los datos multivariantes, surgen cuando los investigadores registran los valores de varias variables aleatorias en un cierto número de objetos o sujetos (comúnmente se usa el término general “unidades”) en los que están interesados, lo que conduce a un vector valuado o una observación multidimensional para cada unidad. Dichos datos se recopilan en una amplia gama de disciplinas y se puede afirmar que la mayoría de los conjuntos de datos en la práctica son multivariantes. [9]

En algunos estudios, las variables se eligen por diseño, debido a que se sabe que son descripciones esenciales del sistema bajo investigación. En otros estudios, particularmente aquellos que han sido difíciles o costosos de organizar, muchas variables pueden medirse simplemente para recopilar tanta información como sea posible por conveniencia o economía.

La mayoría de los conjuntos de datos multivariantes, se pueden representar de la misma manera, es decir, en un formato rectangular conocido de hojas de cálculo, en el que elementos de cada renglón corresponde a los valores de las variables de una unidad en particular y los elementos de las columnas corresponden a los valores tomados por una variable particular.

De este modo el formato rectangular típico de los datos multivariantes es el siguiente:

| **Unidad** | **Variable 1** | . | . | . | **Variable** $q$ |
|---|---|---|---|---|---|
| 1 | $x_{11}$ | | | | $x_{1q}$ |
| . | . | | | | . |
| . | . | | | | . |
| . | . | | | | . |
| $n$ | $x_{n1}$ | . | . | . | $x_{nq}$ |

**Tabla 1.** Típico formato de datos multivariantes.

La parte de las observaciones en la **Tabla 1**, es generalmente representada por una matriz de datos $n \times q$ denotada por $\mathbf{X}$.

En contraste a los datos observados, las entidades teóricas que describen las distribuciones univariadas de cada una de las variables $q$ y su distribución conjunta, son llamadas variables aleatorias, las cuales son denotadas como $X_1, ..., X_q$.

Aunque en algunos casos, donde se han recolectado datos multivariantes, puede tener sentido aislar cada variable y estudiarla por separado, en general no tiene sentido, ya que, como el conjunto de variables se mide en cada unidad, las variables estarán relacionadas en mayor o menor grado. Debido a esto, si cada variable se analiza de forma aislada, es posible que no se revele la estructura completa de los datos.

El análisis estadístico multivariante o multivariado, es el análisis estadístico simultáneo de cada variable mediante el uso de información sobre las relaciones entre las variables. Las unidades en un conjunto de datos multivariados a veces se toman como muestra de una población de interés, una población sobre la que se desea hacer alguna inferencia, sin embargo, sucede con mucha frecuencia que no se puede decir que las unidades hayan sido muestreadas de alguna población en algún sentido significativo,

y las preguntas sobre los datos, son en gran parte de carácter exploratorio. Consecuentemente, existen métodos de análisis multivariante que son esencialmente exploratorios y otros que pueden utilizarse para la inferencia estadística.

Para la exploración de datos multivariantes, no se requieren modelos formales diseñados para obtener respuestas específicas a preguntas rígidamente definidas. En vez de esto, se usan métodos que permiten la detección de patrones posiblemente imprevistos en los datos, abriendo una amplia gama de diversas explicaciones competitivas. Tales métodos, generalmente se caracterizan tanto por el énfasis en la importancia de las visualizaciones gráficas como por la visualización de los datos y la falta de cualquier modelo probabilístico asociado que permita inferencias formales, ejemplos de este tipo de técnicas son: El análisis de componentes principales (PCA), escalado multidimensional, análisis factorial y el análisis de clústeres.

Sin embargo, en muchos casos cuando se trata de datos multivariantes, esta distinción implícita entre lo exploratorio y lo inferencial puede ser una pista falsa, ya que el objetivo general de la mayoría de los análisis multivariantes, ya sean implícitamente exploratorios o inferenciales, es descubrir, mostrar o extraer cualquier "señal" en los datos en presencia de ruido y descubrir lo que los datos tienen que decirnos. [9]

Usaremos los términos **respuesta**, **resultado** o **variable dependiente** para mediciones que pueden variar libremente en respuesta a otras variables llamadas **variables explicativas** o **variables predictoras** o **variables independientes**. Las respuestas se consideran variables aleatorias, mientras que las variables explicativas generalmente se tratan como si fueran mediciones u observaciones no aleatorias; por ejemplo, pueden ser fijados por el diseño experimental. [11]

Las respuestas y las variables explicativas se miden en una de las siguientes escalas:

- **Clasificaciones nominales**: por ejemplo, rojo, verde, azul; sí, no, no aplica. En particular, para las variables binarias, dicotómicas o binomiales, solo hay dos categorías: masculino, femenino; muerto vivo; hojas lisas, hojas dentadas. Si hay más de dos categorías, la variable se denomina policotómica, politómica o multinomial.
- **Clasificaciones ordinales**: en las que existe algún orden o clasificación natural entre las categorías; por ejemplo, joven, mediana edad, anciano; presiones sanguíneas diastólicas agrupadas como $\leq 70$, $71-90$, $91-110$, $111-130$, $\geqslant 131\, mmHg$.
- **Mediciones continuas**: donde las observaciones pueden, al menos en teoría, caer en cualquier lugar de un continuo; por ejemplo, peso, longitud o tiempo.

Los datos nominales y ordinales a veces se denominan **variables categóricas** o **discretas** y, por lo general, se registra el número de observaciones, recuentos o frecuencias

en cada categoría. Para **datos continuos**, se registran las mediciones individuales. El término cuantitativo se usa a menudo para una variable medida en una escala continua y el término cualitativo para mediciones nominales y algunas veces para mediciones ordinales. Los métodos de análisis estadístico, dependen de las escalas de medición de la respuesta y las variables explicativas. [11]

En muchos libros de texto de estadística, la discusión de los diferentes tipos de medidas suele ir seguida de recomendaciones sobre qué técnicas estadísticas son adecuadas para cada tipo; por ejemplo, los análisis de datos nominales deben limitarse a estadísticas resumidas como el número de casos, la moda, etc. Y, para los datos ordinales, las medias y las desviaciones estándar no son adecuadas. Pero Velleman y Wilkinson [12], señalan el punto importante de que restringir la elección de métodos estadísticos de esta manera, puede ser una práctica peligrosa para el análisis de datos, debido a que estas restricciones a menudo son demasiado estrictas para aplicarlas a datos del mundo real. [9]

Por otro, lado dentro de un modelo de aprendizaje automático, los ejemplos que utiliza el sistema para aprender se denominan **conjunto de entrenamiento**. Cada ejemplo de entrenamiento se denomina **instancia** de entrenamiento (o muestra). [13]

La única forma de saber qué tan bien se generalizará un modelo a casos nuevos es probarlo en casos nuevos. Una forma de hacerlo, es poner el modelo en producción y monitorear su desempeño, sin embargo, una mejor opción es dividir los datos en dos conjuntos: **el conjunto de entrenamiento** y el **conjunto de prueba**. Como indican estos nombres, el modelo se entrena con el conjunto de entrenamiento y se prueba con el conjunto de prueba. La tasa de error en casos nuevos se denomina error de generalización (o error fuera de la muestra) y, al evaluar el modelo en el conjunto de prueba, se obtiene una estimación de este error. Este valor le indica qué tan bien funcionará su modelo en instancias que nunca antes había visto.

Si el error de entrenamiento es bajo (es decir, el modelo comete pocos errores en el conjunto de entrenamiento) pero el error de generalización es alto, significa que el modelo está **sobreajustando** (overfitting) los datos de entrenamiento. Mientras que el **subajuste** (underfitting) es lo opuesto al sobreajuste: ocurre cuando el modelo es demasiado simple para aprender la estructura subyacente de los datos. [13]

El aprendizaje automático es ideal para [9]:

1. Problemas para los que las soluciones existentes requieren muchos ajustes manuales o largas listas de reglas: un algoritmo de aprendizaje automático a menudo puede simplificar el código y funcionar mejor.

2. Problemas complejos para los que no existe una buena solución utilizando un enfoque tradicional: las mejores técnicas de Machine Learning pueden encontrar

una solución.

3. Entornos fluctuantes: un sistema de aprendizaje automático puede adaptarse a nuevos datos.

4. Obtener conocimientos sobre problemas complejos y grandes cantidades de datos.

## 2.2. Flujo de trabajo en Machine Learning

Esta sección presenta un flujo de trabajo que detalla los pasos básicos a seguir en un proyecto de aprendizaje automático. Primero describimos los pasos del flujo de trabajo en términos generales y luego, en secciones separadas.

El "flujo de trabajo de aprendizaje automático" toma un conjunto de datos sin procesar como entrada y produce un modelo de ajuste como salida optimizado para producir buenas predicciones sobre datos futuros no vistos. A continuación enumeramos los pasos y los ilustramos en la **Figura 2.1**.

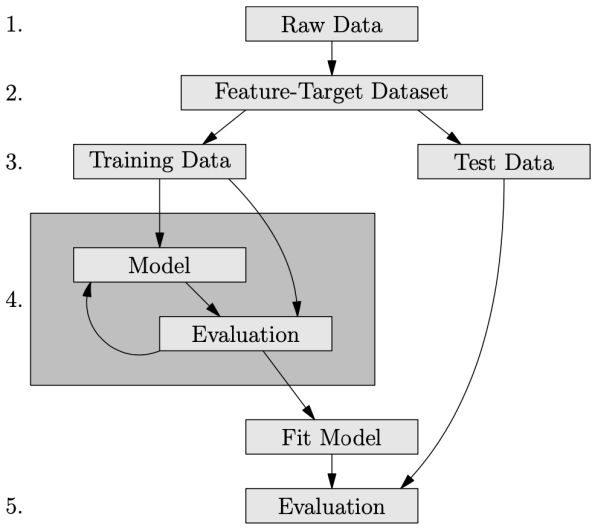



**Figura 2.1:** Un diagrama de flujo que ilustra un flujo de trabajo típico en Machine Learning. Los números a la izquierda de la figura corresponden a los pasos enumerados en el texto. [13]

1. Obtener el conjunto de datos sin procesar inicial.

2. Procesamiento previo: Se crea un conjunto de datos características - etiqueta (variables independientes - variable respuesta), a partir del conjunto de datos

inicial, lo que implica la limpieza de datos, el manejo de valores faltantes, la ingeniería de características y la selección de una variable de respuesta.

3. Se divide el conjunto de datos de características y respuesta en un conjunto de entrenamiento y un conjunto de prueba.

4. Modelado: Se crea un modelo a partir del conjunto de datos de entrenamiento.

   Mantendremos por fines de claridad este ejemplo de flujo de trabajo simple y formaremos un solo modelo sin buscar mejorarlo iterativamente. Pero, como muestra la **Figura 2.1**, este paso es típicamente un proceso iterativo, aprovechando herramientas como la validación cruzada o bootstrap.

5. Se utiliza el modelo para crear predicciones a partir de las características en el conjunto de datos de prueba y se puntúan esas predicciones comparándolas con los valores objetivos correspondientes en el conjunto de datos de prueba.

   La mayor parte del trabajo involucrado en los proyectos de aprendizaje automático se encuentra en el paso 2 (preprocesamiento) y el paso 4 (modelado) y, por lo tanto, esos pasos constituyen la mayor parte de esta sección.

### 2.2.1. Paso 1: Obtención del conjunto de datos inicial

El primer paso del flujo de trabajo es obtener el conjunto de datos sin procesar inicial. El conjunto de datos generalmente es fácil de obtener gratuitamente en Internet y muchos conjuntos de datos vienen ya integrados en varios sistemas de software de aprendizaje automático y estadístico populares. En otros casos, sin embargo, esta etapa implica leer y fusionar múltiples conjuntos de datos que pueden estar en diferentes formatos. Esto puede involucrar una amplia gama de técnicas de codificación.

El conjunto de datos inicial producido en esta etapa debe verificarse para asegurarse de que cumpla con las expectativas. Esta verificación puede ser informal y realizarse manualmente, lo que se denomina análisis exploratorio de datos, o puede ser un proceso de monitoreo automatizado que se utiliza para verificar la calidad del conjunto de datos con las especificaciones formales. En ambos casos, se comprueban los rangos y distribuciones de las variables y se identifican los valores faltantes. [13]

### 2.2.2. Paso 2: Preprocesamiento de datos

El paso de preprocesamiento en el flujo de trabajo toma como entrada el conjunto de datos inicial y produce un conjunto de datos características-respuesta utilizando una serie de transformaciones de datos. Este paso tiene dos objetivos. En primer lugar, el conjunto de datos características-respuesta resultante no debe tener valores perdidos. En segundo lugar, es posible que sea necesario crear o modificar algunas características, un proceso conocido como ingeniería de características. [13]

Muchos factores afectan el éxito del aprendizaje automático (ML) en una tarea determinada. La representación y la calidad de los datos es lo primero y más importante. Si hay mucha información irrelevante y redundante o datos ruidosos y poco fiables, el descubrimiento de conocimientos durante la fase de formación es más difícil. Es bien sabido que los pasos de preparación y filtrado de datos requieren una cantidad considerable de tiempo de procesamiento en los problemas de ML. El preprocesamiento de datos incluye limpieza, normalización, transformación, extracción y selección de características, etc. El producto del preprocesamiento de datos es el conjunto de formación final. [9]

#### 2.2.2.1. Valores perdidos y valores atípicos (outliers)

Muchos conjuntos de datos tienen entradas que no están especificadas (**valores perdidos**) o son **valores atípicos** (valores presentes pero extremos). Los valores atípicos (outliers) o faltantes pueden deberse a la naturaleza del dominio del que se derivan los datos o debido a problemas con la recopilación de datos. Para el manejo de estos valores hay varias opciones: eliminar observaciones con valores perdidos o atípicos, eliminar entidades con demasiados valores perdidos, reemplazar valores perdidos o atípicos con valores válidos, o usar un modelo que sea robusto a valores perdidos o atípicos.

Las dos primeras opciones requieren un umbral del número o la proporción de valores perdidos o atípicos. La eliminación de filas o columnas que exceden el umbral no debe hacerse a la ligera, ya que puede haber una "señal" en los valores faltantes o atípicos.

La tercera opción, reemplazar los valores perdidos o atípicos, se llama **imputación de datos** y debe hacerse con cuidado, para no afectar adversamente los resultados del modelo. Los datos faltantes pueden imputarse utilizando conocimientos especializados en el dominio o aprovechando herramientas estadísticas. Dos de las técnicas de imputación más comunes son reemplazar los valores perdidos con un valor específico basado en la comprensión del dominio o reemplazarlos con la media, la mediana o la moda de la variable. Otro enfoque consiste en reemplazar un valor faltante con un número aleatorio dentro de una desviación estándar del valor medio de la variable. Una forma más complicada de imputación es crear un modelo que prediga el valor de reemplazo en función de otras características del conjunto de datos.

Si se elige realizar la imputación de datos (tal como hacemos en el presente trabajo), es importante considerar medios sutiles de **fuga de datos**. Consideremos un estudio médico, en el que faltan los datos de la presión arterial de algunos pacientes, y optamos por imputar esos valores faltantes con las cifras medias de la presión arterial en todos los pacientes del conjunto de datos. Si esta acción se realiza, antes de que se hayan separado los conjuntos de entrenamiento y prueba, entonces las filas del conjunto de prueba han afectado este valor medio, que es probable que se impute en algunas filas

en los datos de entrenamiento, lo cual es un forma sutil de fuga de datos. Por lo tanto, cualquier imputación que agregue datos de una columna completa debe evitarse en el paso de preprocesamiento y realizarse una vez que se hayan separado los conjuntos de entrenamiento y prueba. En la **sección 2.2.6** hablaremos de forma extensa acerca de la fuga de datos y sus repercusiones en los modelos predictivos. [13]

La idea detrás del enfoque de imputación es reemplazar los valores faltantes con otros valores sensibles. Dado que siempre se pierde información con el enfoque de eliminación al descartar muestras (filas) o características completas (columnas), la imputación suele ser el enfoque preferido. [14]

Las múltiples técnicas de imputación se pueden dividir en dos subgrupos: imputación única o imputación múltiple.

En la imputación simple, se genera un valor de imputación simple para cada una de las observaciones faltantes. El valor imputado se trata como el valor real, ignorando el hecho de que ningún método de imputación puede proporcionar el valor exacto. Por lo tanto, la imputación única no refleja la incertidumbre de los valores faltantes.

En la imputación múltiple, se generan muchos valores imputados para cada una de las observaciones que faltan. Esto significa que se crean muchos conjuntos de datos completos con diferentes valores imputados. El análisis (p. ej., entrenar una regresión lineal para predecir una columna objetivo) se realiza en cada uno de estos conjuntos de datos y se sondean los resultados. La creación de imputaciones múltiples, en lugar de imputaciones únicas, explica la incertidumbre estadística en las imputaciones. [14]

La mayoría de los métodos de imputación son métodos de imputación simple, siguiendo tres estrategias principales: reemplazo por valores existentes, reemplazo por valores estadísticos y reemplazo por valores predichos. Dependiendo de los valores utilizados para cada una de estas estrategias, terminamos con métodos que funcionan solo con valores numéricos y métodos que funcionan tanto con columnas numéricas como nominales. [13]

#### 2.2.2.2. Imputación con la media, Imputación con la media por grupos e Imputación por regresión

En el presente trabajo se hará uso de las siguientes técnicas de imputación:

(1) **Mean Imputation** (Imputación con la media): En este método, se calcula la media de todos los valores dentro de la misma columna (característica o atributo) y luego se imputa en las celdas de datos que faltan. El método funciona solo si la característica no es nominal. Es decir, sea $X_i^j$ el atributo faltante de la instancia (una

instancia es cada uno de los datos de los que se disponen para hacer el análisis) $i$-ésima, que se imputa por:

$$X_i^j = \sum_{k \in I(complete)} \frac{X_k^j}{n_{|I(complete)|}} \tag{2.1}$$

donde I(complete) es un conjunto de índices de datos que no faltan en $X_i$ y $n_{|I(complete)|}$ es el número total de instancias en las que no falta el j-ésimo atributo.

(2) **Group Mean Imputation** (Imputación con la media por grupos): El proceso de este método es el mismo que el de la imputación media. Sin embargo, los valores que faltan se reemplazan con la media del grupo (o clase) de todos los valores conocidos de la característica. Cada grupo representa una clase objetivo de entre las instancias (registradas) que tienen valores faltantes. Sea $X_{m,i}^j$ el $j$-ésimo atributo faltante de la $i$-ésima instancia de la $m$-ésima clase, que se imputa mediante:

$$X_{m,i}^j = \sum_{k \in I(mth\ class\ incomplete)} \frac{X_{m,k}^j}{n_{|I(mth\ class\ incomplete)|}} \tag{2.2}$$

donde $I(mth\ class\ incomplete)$ es un conjunto de índices que no faltan en $X_{m,i}^j$ y $n_{|I(mth\ class\ incomplete)|}$ es el número total de instancias en las que no falta el j-ésimo atributo de la m-ésima clase.

(3) **Regression Imputation** (Imputación por regresión). Con la imputación por regresión, la información de otras variables se utiliza para predecir los valores faltantes en una variable mediante el uso de un modelo de regresión. Por lo general, primero se estima el modelo de regresión en los datos observados y, posteriormente, utilizando los pesos de regresión, se predicen y reemplazan los valores faltantes. [15]

#### 2.2.2.3. Método Z-score para identificar outliers

En el presente trabajo uno de los métodos empleados para identificar y remover outliers es el Método Z-score (los otros métodos empleados se basan en el algoritmo K-means, algoritmo al que hemos dedicado una sección más adelante).

El Z-score (o puntaje Z) son puntuaciones (scores) estándar que describen las diferencias de los valores individuales de la media en unidades de desviación estándar. Siempre que se conozcan la media y la desviación estándar en una distribución (o a través de sus valores estimados), todos los valores originales (también conocidos como puntajes brutos) de una variable se pueden convertir en puntajes Z. Los puntajes Z se pueden comparar con el mismo criterio. Se puede pensar en el criterio marcado por las puntuaciones Z como un criterio universal. Todos los puntajes Z se informan solo como

el número de desviación estándar por encima o por debajo de la media, sin la unidad de medida específica. Las puntuaciones Z proporcionan una regla general para identificar puntajes inusuales y puntajes extremos. La diferencia entre los valores inusuales y los valores extremos (outliers o valores atípicos) es que hay alrededor del 5 % de los valores que se consideran inusuales en una distribución, pero se sabe que existen. Pero los valores extremos no siempre existen en una distribución. Si existen, menos del 1 % de los valores de una distribución se consideran valores atípicos (outliers). Los valores atípicos tienen una influencia indebida en los resultados de los análisis estadísticos. Deben ser identificados y tratados con precaución. La regla general es que si un valor $X$ se convierte en una puntaje $Z$ y está a más de 2 desviaciones estándar de la media, $|Z| > 2$, entonces $X$ es un valor inusual. La regla general para identificar valores atípicos es más estricta. Si un valor $X$ se convierte en una puntuación $Z$ y está a más de 3 desviaciones estándar de la media $|Z| > 3$, entonces X es un valor atípico. Los valores de $Z$ entre $-2$ y $+2$ son valores ordinarios, como se muestra en la **Figura 2.2**.

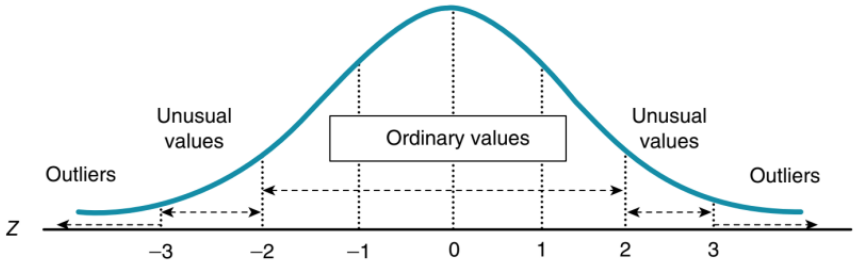



**Figura 2.2:** Valores ordinarios, inusuales y outliers. [16]

Los puntajes $Z$ estándar tienen tres atributos importantes:

**1.** El signo + o − describe el puntaje en relación con la media. Todas las puntuaciones Z positivas están por encima de la media y todas las puntuaciones $Z$ negativas superan la media.

**2.** La magnitud de la puntuación Z describe la distancia entre el valor y la media en número de unidades de desviación estándar.

**3.** Al transformar todos los puntajes sin procesar en puntajes $Z$, se está realizando una operación matemática en los valores sin procesar. Esta operación matemática no cambia la forma de la distribución. La forma de la distribución sigue siendo la misma que la distribución original. Los puntajes $Z$ tienen media = 0 y desviación estándar = 1.

Para calcular la puntuación Z de una observación, se toma la medida sin procesar, se resta la media ($\mu$) y se divide por la desviación estándar ($\sigma$). Matemáticamente, la

fórmula para este proceso es la siguiente:

$$Z = \frac{X - \mu}{\sigma} \tag{2.3}$$

De este modo, si la puntuación Z de un punto de datos es superior a 3 (debido a que se cubre el 99,7 % del área), el valor de los datos es bastante diferente a los demás valores y dicho dato es tomado como un valor atípico. [16]

#### 2.2.2.4. Ingeniería de características

La **ingeniería de características** es el proceso de tomar un conjunto de datos y construir variables explicativas (características) que se pueden usar para entrenar un modelo de aprendizaje automático para un problema de predicción. Estas técnicas se dividen en dos categorías amplias: utilizar los requisitos del modelo y utilizar nuestra comprensión del dominio del problema.

Por ejemplo, cuando se usa un modelo de regresión lineal, asumimos relaciones lineales entre las características del conjunto de datos y su variable objetivo. Pero podemos saber o sospechar que algunas características tienen una relación no lineal con el objetivo y, por lo tanto, podría ser apropiado agregar una nueva característica calculada transformando una o más de las existentes. Esto incluye términos de "interacción", es decir, el producto de dos características. Estos a veces se denominan características polinomiales por razones obvias y están motivados tanto por las limitaciones de un modelo lineal como por nuestro conocimiento del dominio.

Debido a que la mayoría de los modelos de aprendizaje automático asumen que las características son numéricas, las características categóricas deben convertirse en una o más características llamadas "binarias" o "ficticias", que son numéricas.

El enfoque más tradicional de la ingeniería de características es construir características una a la vez usando el conocimiento del dominio, el problema de este enfoque es que suele ser tardado y propenso a errores, este enfoque tradicional es conocido como ingeniería de funciones manuales.

Por otro lado, la ingeniería de características automatizada puede mejorar este flujo de trabajo estándar extrayendo automáticamente características útiles de un conjunto de datos, dando un marco que se puede aplicar a cualquier problema. [13]

En este sentido, como un objetivo central del presente trabajo, está la contrucción de un método de creación de características automatizado que nos permita mejorar el modelado predictivo de diabetes y pueda ser generalizado a otros conjuntos de datos.

### 2.2.3. Paso 3: Creación de conjuntos de datos de prueba y entrenamiento

Normalmente el 70 % de los datos se colocan en el conjunto de datos de entrenamiento y el 30 % restante en el conjunto de datos de prueba.

Para una variable objetivo numérica, asumiendo que no es secuencial (por ejemplo, el precio de una acción a intervalos de tiempo regulares), las observaciones generalmente se asignan al azar a los conjuntos de datos de entrenamiento y prueba. Para una variable objetivo categórica, es deseable asegurarse de que la distribución de clases del objetivo sea similar en los conjuntos de datos de entrenamiento y prueba, en este caso es pertinente realizar una división de datos estratificada (ver **sección 2.6.3**). [13]

### 2.2.4. Paso 4: Creación del modelo

La elección de qué estrategia utilizar para entrenar nuestro conjunto de datos de entrenamiento, se basa en la exploración y manipulación de datos en los pasos 1 y 2, y la elección del modelo también puede haber informado parte del trabajo de ingeniería de características, como se describe anteriormente.

Hay una distinción entre lo que es un **modelo** y un **modelo ajustado**. El científico de datos elige el modelo y el entorno informático entrena ese modelo en el conjunto de datos de entrenamiento, produciendo coeficientes (en el caso de regresión lineal, o parámetros en un modelo en general) que nos dan un modelo ajustado, listo para ser aplicado a nuevos puntos de datos. Ese proceso de entrenamiento ocurre en este paso.

Este es quizás el paso central en el flujo de trabajo del aprendizaje automático. Minimizar el error en el conjunto de entrenamiento no garantiza, en general, un buen rendimiento en datos no vistos. Este es el peligro llamado "sobreajuste". Este peligro es más probable si las hipótesis del modelo son demasiado complejas o específicas. Tal complejidad debe considerarse cuidadosamente en comparación con el tamaño del conjunto de entrenamiento; los conjuntos más pequeños son más propensos a sobreajustarse. Estas pautas pueden ayudarnos a seleccionar un modelo que tenga más probabilidades de funcionar bien con datos no vistos. [13]

#### 2.2.4.1. Escalado y normalización

Por lo general, la regresión lineal no requiere escalar las variables de características, ya que los coeficientes se pueden adaptar automáticamente a la escala de las características, pero este apartado es crucial en otros modelos.

Dos técnicas comunes para crear entidades con rangos comunes son:

**min-max scaling**: Donde cada valor ${x_j}^{(i)}$ de la característica $j$ es reemplazado por:

$$\frac{{x_j}^{(i)} - \text{mín}_i \, {x_j}^{(i)}}{\text{máx}_i \, {x_j}^{(i)} - \text{mín}_i \, {x_j}^{(i)}} \tag{2.4}$$

donde $i$ varía sobre todas las observaciones. Esto crea valores en el intervalo $[0, 1]$.

La segunda técnica comunmente empleada es:

**standard scaling**: En este caso cada valor ${x_j}^{(i)}$ de la característica $j$ es reemplazada por:

$$\frac{{x_j}^{(i)} - \mu}{\sigma} \tag{2.5}$$

donde $\mu$ y $\sigma$ son la media y la desviación estándar de la característica $j$ e $i$ varía sobre todas las observaciones. Esta transformación nos elimina la media y escala los datos a la varianza unitaria. [13]

La motivación para usar esta escala incluye la robustez a desviaciones estándar muy pequeñas de características y la preservación de entradas cero en datos escasos. [17]

#### 2.2.4.2. Importancia del escalado de características

El escalado de características a través de min-max scaling o standard scaling puede ser un paso de preprocesamiento importante para muchos algoritmos de aprendizaje automático.

La estandarización (standard scaling), implica volver a escalar las características de modo que tengan las propiedades de una distribución normal estándar con una media de cero y una desviación estándar de uno. Intuitivamente podemos pensar en el Análisis de Componentes Principales (PCA) (ver **sección 2.3**) como un excelente ejemplo de cuándo la estandarización es importante. En PCA estamos interesados en los componentes que maximizan la varianza. Si un componente (por ejemplo, la altura) varía menos que otro (por ejemplo, el peso) debido a sus respectivas escalas (metros frente a kilos), PCA podría determinar que la dirección de la variación máxima se corresponde más estrechamente con el eje del "peso", si esas características no están escaladas, ya que un cambio en la altura de un metro puede considerarse mucho más importante que el cambio en el peso de un kilogramo, lo cual es claramente incorrecto. [18]

Por otro lado, la normalización (min-max scaling), es muy importante para los métodos con regularización. Esto se debe a que la escala de las variables afecta la can-

tidad de regularización que se aplicará a una variable específica.

Por ejemplo, supongamos que una variable está en una escala muy grande, digamos del orden de millones y otra variable está entre 0 y 1. Entonces, podemos pensar que la regularización tendrá poco efecto en la primera variable. De este modo, la regularización hace que el predictor dependa de la escala de las características. [13]

En el caso de la regularización Ridge (ver **sección 2.5.4**), podemos expresar este concepto de una forma más formal diciendo que las soluciones de la regularización Ridge no son equivariantes bajo la escala de las entradas, por lo que normalmente se normalizan las entradas. [19]

Finalmente, recalcamos que cualquier transformación de escala que se aplique a los datos de entrenamiento también debe aplicarse sin cambios a los datos de prueba, o los datos de prueba no serán entradas sensibles para el modelo de ajuste más adelante. [13]

#### 2.2.4.3. Selección de características

La **selección de características**, es el proceso de identificar y seleccionar un subconjunto de características de entrada que son más relevantes para la variable de respuesta. La selección de características suele ser sencilla cuando se trabaja con datos de entrada y salida de valor real, como el uso del coeficiente de correlación de Pearson, pero puede ser un desafío cuando se trabaja con datos de entrada numéricos y una variable objetivo categórica. [20]

Suele ser común usar medidas estadísticas de tipo correlación entre las variables de entrada y salida como base para la selección de características. La elección de medidas estadísticas depende en gran medida de los tipos de datos. Los tipos de datos comunes incluyen numéricos (como la edad) y categóricos (como una etiqueta). Por otro lado, las **variables de entrada** son aquellas que se proporcionan como entrada a un modelo. En la selección de características, es este grupo de variables el que deseamos reducir de tamaño, mientras que las **variables de salida** son aquellas para las que un modelo pretende predecir, a menudo llamadas variables de respuesta.

El tipo de variable de respuesta normalmente indica el tipo de problema de modelado predictivo que se está realizando. Por ejemplo, una variable de salida numérica indica un problema de modelado predictivo de **regresión** y una variable de salida categórica indica un problema de modelado predictivo de **clasificación**.

Las medidas estadísticas utilizadas en la selección de características, generalmente calculan una variable de entrada a la vez con la variable de destino. Como tales, se les conoce como medidas estadísticas univariadas. Esto, puede significar que cualquier interacción entre las variables de entrada no se considera en el proceso de filtrado.

La selección de características, también está relacionada con las técnicas de reducción de dimensionalidad en el sentido de que ambos métodos buscan menos variables de entrada para un modelo predictivo. La diferencia, es que la selección de características selecciona características para mantenerlas o eliminarlas del conjunto de datos, mientras que la reducción de la dimensionalidad, crea una proyección de los datos que da como resultado características de entrada completamente nuevas. Como tal, la reducción de la dimensionalidad, es una alternativa a la selección de características en lugar de un tipo de selección de características.

El problema que nos concierne de modelado predictivo de diabetes, es un problema con entradas numéricas y variable objetivo de clasificación binaria. Hay dos técnicas populares de selección de características que se pueden usar para datos de entrada numéricos y una variable objetivo categórica. Ellos son: ANOVA F-Test y Mutual Information Statistics. [20]

En el presente proyecto, haremos uso de la prueba ANOVA F-Test para seleccionar las características numéricas más significativas en la predicción de la aparición de diabetes, características que nos permitirán crear un conjunto nuevo de predictores que servirán para mejorar el rendimiento de nuestro modelo de aprendizaje automático (detallaremos este proceso más adelante en la **sección 3.4.3.4**).

#### 2.2.4.4. ANOVA F-Test

ANOVA, es un acrónimo de análisis de varianza y es una prueba de hipótesis estadística paramétrica para determinar si las medias de dos o más muestras de datos provienen de la misma distribución o no. Una estadística F, o prueba F, es una clase de pruebas estadísticas que calculan la relación entre los valores de varianza, como la varianza de dos muestras diferentes o la varianza explicada y no explicada por una prueba estadística, como ANOVA. El método ANOVA es un tipo de estadística F que aquí se denomina ANOVA F-test.

Es importante destacar que ANOVA, se usa cuando una variable es numérica y la otra es categórica. Los resultados de esta prueba se pueden usar para la selección de características, donde aquellas características que son independientes de la variable de respuesta se pueden eliminar del conjunto de datos. [20]

La prueba ANOVA, utiliza la estadística F para la clasificación de características. Cuanto mayor sea el valor de la estadística F, mejor será la capacidad discriminativa de dicha característica.

El objetivo, al aplicar esta técnica en el presente trabajo, es realizar una prueba ANOVA F-test unidireccional para cada predictor continuo, que pruebe si todas las

diferentes clases de $\mathbf{Y}$ (diabetes o no-diabetes) tienen la misma media que $\mathbf{X}$. Se aplica la siguiente notación:

$N_j$ = El número de casos con $\mathbf{Y} = j$

$\bar{x}_j$ = La media muestral del predictor $\mathbf{X}$ para la clase objetivo $\mathbf{Y} = j$

$s_j^2 = \sum_{i=1}^{N_j} \frac{(x_{ij} - \bar{x}_j)^2}{N_j - 1}$ = La varianza muestral del predictor $\mathbf{X}$ para la clase objetivo $\mathbf{Y} = j$

$\bar{\bar{x}} = \sum_{j=1}^{J} \frac{N_j \bar{x}_j}{N}$ = La gran media del predictor $\mathbf{X}$, donde $N$ es el número total de pacientes y $J$ es el número total de clases.

De este modo el estadístico F podemos calcularlo con la siguiente expresión:

$$F = \frac{\frac{\sum_{j=1}^{J} N_j (\bar{x}_j - \bar{\bar{x}})^2}{J - 1}}{\frac{\sum_{j=1}^{J} (N_j - 1) s_j^2}{N - 1}} \tag{2.6}$$

donde $F$ es una variable aleatoria que sigue una distribución F con grados de libertad $(J - 1)$ y $(N - J)$.

De este modo, para todas las características numéricas del conjunto de datos, el valor F se calcula mediante la Ecuación (2.6), conservando finalmente las características con el mayor valor F. [21]

### 2.2.5. Paso 5: Predicción y evaluación

En este paso, se evalúa si el modelo ajustado que optimizamos usando datos de entrenamiento, funcionará bien en datos que no se han visto. Por lo tanto, ahora usamos el modelo ajustado para hacer predicciones a partir de características del conjunto de datos de prueba, este proceso es completamente mecánico, el software simplemente necesita aplicar la función obtenida a cada fila del conjunto de datos de prueba.

Por otro lado, para evaluar el rendimiento del modelo ajustado, se utilizan diferentes métricas, las cuales detallaremos más en la **sección 2.6** (enfocándolo en nuestro contexto del análisis predictivo de diabetes). [13]

### 2.2.6. Fuga de datos

Como vimos anteriormente, la preparación de datos, es el proceso de transformar datos sin procesar en una forma apropiada para el modelado. El enfoque más común, llamado también enfoque ingenuo, es aplicar la preparación de datos en todo el conjunto de datos antes de evaluar el rendimiento del modelo. Esto da como resultado un problema conocido como **fuga de datos** (data leakage), donde el conocimiento del conjunto de prueba se filtra al conjunto de datos utilizado para entrenar el modelo. Esto puede resultar en una estimación incorrecta del rendimiento del modelo al realizar predicciones sobre nuevos datos. Es requerido una aplicación cuidadosa de las técnicas de preparación de datos para evitar la fuga de datos, lo cual varía según el esquema de evaluación del modelo utilizado, tales como las divisiones train-test o la validación cruzada. [20]

**Problema con la preparación de datos ingenua**

Un enfoque común, es aplicar primero una o más transformaciones (imputación, eliminación de valores atípicos, PCA, etc.) a todo el conjunto de datos. A continuación dividir el marco de datos en conjuntos de prueba y entrenamiento, o utilizar una validación cruzada de $k$-veces para ajustar y evaluar un modelo de aprendizaje automático. Este enfoque ingenuo queda resumido del siguiente modo:

1. Preparar el conjunto de datos

2. Dividir los datos

3. Evaluar modelos

Aunque este es un enfoque común, suele ser incorrecto y peligroso en la mayoría de los casos. El problema de aplicar técnicas de preparación de datos antes de dividir los datos para la evaluación del modelo, es que puede conducir a una fuga de datos y, a su vez, probablemente resultará en una estimación incorrecta del rendimiento del modelo. La fuga de datos de forma más intuitiva se da cuando la información sobre el conjunto de datos de reserva (datos de prueba o validación), se pone a disposición del modelo en el conjunto de datos de entrenamiento. Esta fuga suele ser pequeña y sutil, pero puede tener un efecto marcado en el rendimiento. [20]

De este modo, Fuga significa que se revela información al modelo que le da una ventaja poco realista para hacer predicciones. Esto, podría suceder cuando los datos de prueba se filtran al conjunto de entrenamiento o cuando los datos del futuro se filtran al pasado. [22]

Obtenemos fugas de datos aplicando técnicas de preparación de datos a todo el

conjunto de datos. Este no es un tipo directo de fuga de datos, donde entrenaríamos el modelo en el conjunto de datos de prueba. En cambio, es un tipo indirecto de fuga de datos, donde algún conocimiento sobre el conjunto de datos de prueba, capturado en estadísticas resumidas, está disponible para el modelo durante el entrenamiento. Esto puede hacer que sea un tipo de fuga de datos más difícil de detectar. [20]

Como ejemplo, podemos considerar el caso donde queremos normalizar los datos, es decir, escalar las variables de entrada al rango $0 - 1$. De este modo para normalizar, es requerido que calculemos los valores mínimo y máximo para cada variable antes de usar estos valores para escalar las variables. Luego, el conjunto de datos se divide en conjuntos de datos de entrenamiento y de prueba, pero los ejemplos del conjunto de datos de entrenamiento ya saben algo sobre los datos del conjunto de datos de prueba, ya que han sido escalados por los valores mínimos y máximos globales, y de esta forma saben más sobre la distribución global de la variable de lo que deberían.

Obtenemos el mismo tipo de filtración con casi todas las técnicas de preparación de datos. También los modelos que imputan valores perdidos mediante un modelo o estadísticas resumidas se basarán en el conjunto de datos completo para completar los valores en el conjunto de datos de entrenamiento. De igual forma no sería válido ajustar el método de detección de valores atípicos en todo el conjunto de datos, ya que esto daría lugar también a una fuga de datos. Es decir, el modelo tendría acceso a los datos en el conjunto de prueba que no debería usarse para entrenar el modelo.

Para remediar esto, la preparación de datos deberá ajustarse únicamente al conjunto de datos de entrenamiento. Una vez ajustados, los algoritmos o modelos de preparación de datos se pueden aplicar al conjunto de datos de entrenamiento y al conjunto de datos de prueba. De este modo un flujo de trabajo más correcto sería el siguiente:

1. Dividir datos.

2. Ajustar la preparación de datos al conjunto de datos de entrenamiento.

3. Aplicar la preparación de datos para entrenar y probar conjuntos de datos.

4. Evaluar modelos.

De manera más general, toda la tubería de modelado debe prepararse solo en el conjunto de datos de entrenamiento para evitar la fuga de datos. Esto podría incluir transformaciones de datos, pero también otras técnicas como selección de características, reducción de dimensionalidad, ingeniería de características, imputación, etc.

Por otro lado, la preparación de datos sin fuga de datos cuando se utiliza la validación cruzada es un poco más desafiante, en este caso se requiere que los métodos de

preparación de datos se preparen en el conjunto de entrenamiento y se aplique a los conjuntos de entrenamiento y prueba dentro del procedimiento de validación cruzada, esto es, dentro de cada pliegue. Podemos lograr esto definiendo una tubería de modelado que define una secuencia de pasos de preparación de datos a realizar y terminar en el modelo ajustado y la evaluación. [20]

En las siguientes secciones, se desarrollan de forma detallada cuatro aspectos que son importantes en el desarrollo del presente proyecto, estos son; Análisis de Componentes Principales (**sección 2.3**), Análisis de Clústeres (**sección 2.4**), Modelo de Regresión Logística (**sección 2.5**) y finalmente Evaluación del Modelo (**sección 2.6**).

## 2.3. Análisis de Componentes Principales

### 2.3.1. Introducción

El Análisis de Componentes Principales (PCA por sus siglas en inglés), es una técnica utilizada para describir un conjunto de datos en términos de nuevas variables ("componentes") no correlacionadas mediante combinaciones lineales de las variables originales con una pérdida mínima de información en los datos observados. La información de interés se puede extraer de los datos a través de esta fusión de las múltiples variables que caracterizan a los individuos. El PCA, también se puede utilizar como una técnica para la comprensión visual de estructuras de datos, proyectando datos de alta dimensión en un número menor de variables, realizando una reducción de dimensión (reducción de dimensionalidad, también conocida como compresión dimensional) y proyectando los resultados en una línea unidimensional, plano bidimensional o espacio tridimensional. [23]

En el mejor de los casos, el resultado de un análisis de componentes principales sería la creación de una pequeña cantidad de nuevas variables que pueden usarse como sustitutos de una gran cantidad de variables originales y, en consecuencia, proporcionar una base más simple para, por ejemplo, graficar o resumir los datos, y realizar análisis multivariados de los datos.

### 2.3.2. Análisis de Componentes Principales (PCA)

La meta principal del análisis de componentes principales, es describir la varianza en un conjunto de variables correlacionadas $\mathbf{x}^T = (x_1, ..., x_p)$ en términos de un conjunto nuevo de variables no correlacionadas, $\mathbf{y}^T = (y_1, ..., y_p)$, cada una de las cuales es una combinación lineal de las $\mathbf{x}$ variables originales. Las nuevas variables están derivadas en orden decreciente de importancia en el sentido de que $y_1$ representa la mayor cantidad de varianza de los datos originales entre todas las combinaciones lineales de $\mathbf{x}$. Luego $y_2$ representa la componente con la segunda mayor cantidad de varianza, sujeto a no

estar correlacionado con $y_1$, etc.

Las nuevas componentes definidas por este proceso $y_1, ..., y_p$ son las componentes principales.

Para encontrar el eje de proyección que produce la varianza máxima, en PCA se usa una secuencia de ejes de proyección. Primero encontramos el eje de proyección, conocido como el primer componente principal, que maximiza la varianza total. A continuación, encontramos el eje de proyección que maximiza la varianza bajo la restricción de ortogonalidad al primer componente principal. Este eje se conoce como el segundo componente principal. Apliquemos esta idea a datos bidimensionales, para dejar más claro este proceso.

Denotamos los $n$ datos observados para las dos variables $\mathbf{x} = (x_1, x_2)^T$ como:

$$\mathbf{x}_1 = \begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix}, \mathbf{x}_2 = \begin{pmatrix} x_{21} \\ x_{22} \end{pmatrix}, \cdots, \mathbf{x}_n = \begin{pmatrix} x_{n1} \\ x_{n2} \end{pmatrix} \tag{2.7}$$

Estos $n$ datos bidimensionales se proyectan en $y = w_1x_1 + w_2x_2$, y luego se expresan como:

$$y_i = w_1x_{i1} + w_2x_{i2} = \mathbf{w}^T\mathbf{x}_i, \qquad i = 1, 2, \cdots, n \tag{2.8}$$

donde $\mathbf{w} = (w_1, w_2)^T$ representa el vector de coeficientes.

La media de los datos $y_1, y_2, \cdots, y_n$ que se proyectan en el eje de proyección es:

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i = \frac{1}{n}\sum_{i=1}^{n}(w_1x_{i1} + w_2x_{i2}) = w_1\bar{x_1} + w_2\bar{x_2} = \mathbf{w}^T\bar{\mathbf{x}} \tag{2.9}$$

donde $\bar{\mathbf{x}} = (\bar{x_1}, \bar{x_2})^T$ es el vector de media muestral que tiene como componentes la media muestral $\bar{x_j} = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$ $(j = 1, 2)$ de cada variable. La varianza se puede

expresar de la siguiente forma:

$$\begin{aligned} s_y^2 &= \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}\left[w_1\left(x_{i1} - \bar{x_1}\right) + w_2\left(x_{i2} - \bar{x_2}\right)\right]^2 \\ &= {w_1}^2\frac{1}{n}\sum_{i=1}^{n}\left(x_{i1} - \bar{x_1}\right)^2 + 2w_1w_2\frac{1}{n}\sum_{i=1}^{n}\left(x_{i1} - \bar{x_1}\right)\left(x_{i2} - \bar{x_2}\right) + {w_2}^2\frac{1}{n}\left(x_{12} - \bar{x_2}\right)^2 \\ &= {w_1}^2 s_{11} + 2w_1w_2s_{12} + {w_2}^2 s_{22} \\ &= \mathbf{w}^T S\mathbf{w} \end{aligned} \tag{2.10}$$

donde S es la matriz muestral de varianza-covarianza definida por:

$$S = \begin{pmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{pmatrix}, \quad s_{jk} = \frac{1}{n}\sum_{i=1}^{n}\left(x_{ij} - \bar{x_j}\right)\left(x_{ik} - \bar{x_k}\right), \quad j,k = 1,2. \tag{2.11}$$

El problema de encontrar el vector de coeficientes $\mathbf{w} = (w_1, w_2)^T$, que corresponde a la varianza máxima para los $n$ datos bidimensionales proyectados sobre $y = w_1x_1 + w_2x_2$ se convierte en el problema de maximización de la varianza $s_y^2 = \mathbf{w}^T S\mathbf{w}$ (en ecuación 2.10) bajo la restricción $\mathbf{w}^T\mathbf{w} = 1$. Esta restricción se aplica, ya que $||\mathbf{w}||$ sería infinitamente grande sin él y, por lo tanto, la varianza divergiría, esto debido a que la varianza podría ser incrementada sin límite, simplemente incrementando los coeficientes $\mathbf{w} = (w_1, w_2)^T$, y debido a esto se debe imponer alguna restricción. Una restricción razonable es requerir que la suma de los cuadrados de los coeficientes sean igual a uno, esto es lo mencionado anteriormente $\mathbf{w}^T\mathbf{w} = 1$.

El problema de la maximización de la varianza bajo esta restricción, puede resolverse mediante el método de Lagrange de multiplicadores indeterminados, encontrando el punto estacionario (en el que la derivada se convierte en 0) de la función de Lagrange.

El método de multiplicadores de Lagrange, es una técnica que permite encontrar los máximos y mínimos locales de una función sujeta a restricciones.

Consideremos el siguiente problema aplicado a dos funciones $f(x)$ y $g(x)$:

Se quiere $Maximizar\ f(x)\ restringido\ a\ g(x) = c$, el método de multiplicadores de Lagrange introduce una nueva variable $\lambda$, llamada multiplicador de Lagrange, la cual permite formar la llamada ecuación de Lagrange:

$$L(x, \lambda) = f(x) - \lambda\left[g\left(x\right) - c\right] \tag{2.12}$$

El signo de $\lambda$ puede ser positivo o negativo. Entonces se resuelve la ecuación para el punto estacionario de $L(x, \lambda)$, esto es:

$$\frac{\partial L\left(x, \lambda\right)}{\partial x} = 0 \tag{2.13}$$

En nuestro caso, podemos plantear el problema de optimización para la primera componente principal de la siguiente forma:

*Maximizar* $s_{y_1}^2 = \mathbf{w}^T S \mathbf{w}$, *restringido* a $\mathbf{w}^T \mathbf{w} = 1$, con lo cual planteamos la siguiente ecuación de Lagrange:

$$L(\mathbf{w}, \lambda) = \mathbf{w}^T S \mathbf{w} - \lambda \left(\mathbf{w}^T \mathbf{w} - 1\right) \tag{2.14}$$

Utilizando las siguientes identidades del cálculo matricial:

$$\frac{\partial}{\partial \mathbf{w}}(\mathbf{w}^T S \mathbf{w}) = 2S\mathbf{w}_1 \tag{2.15}$$

$$\frac{\partial}{\partial \mathbf{w}}(\mathbf{w}^T \mathbf{w}) = 2\mathbf{w} \tag{2.16}$$

Se deriva la función $L(\mathbf{w}, \lambda)$ respecto a $\mathbf{w}$ y se iguala a cero de este modo se obtiene:

$$\frac{\partial L(\mathbf{w}, \lambda)}{\partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}}[\mathbf{w}^T S \mathbf{w} - \lambda(\mathbf{w}^T \mathbf{w} - 1)] = 2S\mathbf{w} - 2\lambda\mathbf{w} = 0 \tag{2.17}$$

Despejando y reduciendo obtenemos:

$$S\mathbf{w} = \lambda\mathbf{w} \tag{2.18}$$

Podemos observar que se tiene una ecuación de valores propios, esto quiere decir que los coeficientes que estábamos buscando para escribir la primera componente principal como una combinación lineal de las variables originales, se corresponde al vector propio $\mathbf{w}_1$ de la matriz de covarianza de muestra $S$ correspondiente al valor propio máximo $\lambda_1$ obtenido al resolver la ecuación característica para la matriz muestral de varianza-covarianza $S$.

Esta solución es el vector propio $\mathbf{w}_1 = (\mathbf{w}_{11}, \mathbf{w}_{12})^T$ correspondiente al valor propio máximo $\lambda_1$ obtenido al resolver la ecuación característica para la matriz muestral de varianza-covarianza $S$. En consecuencia, la primera componente principal viene dada por:

$$y_1 = w_{11}x_1 + w_{12}x_2 = \mathbf{w_1}^T \mathbf{x} \tag{2.19}$$

Tomando la relación $S\mathbf{w}_1 = \lambda_1 \mathbf{w}_1$ y multiplicando ambos lados por $\mathbf{w_1}^T$, obtenemos:

$$\mathbf{w_1}^T S \mathbf{w}_1 = \mathbf{w_1}^T \lambda_1 \mathbf{w}_1 = \lambda_1 \mathbf{w_1}^T \mathbf{w}_1 \tag{2.20}$$

pero recordemos que ${\mathbf{w}_1}^T\mathbf{w}_1 = 1$ y ${\mathbf{w}_1}^T S\mathbf{w}_1 = {s_{y_1}}^2$, de este modo obtenemos que

$$s_{y_1}^2 = \lambda_1 \tag{2.21}$$

Vemos pues que la varianza de la primera componente principal $y_1$ es el mayor valor propio proveniente de la matriz muestral de varianza-covarianza.

Ahora para la segunda componente principal, que tiene los datos bidimensionales proyectados en $y = w_1x_1 + w_2x_2$ podemos plantear el siguiente problema:

La segunda componente principal $y_2$ será la componente con la segunda varianza más grande, esto es, ahora queremos encontrar los coeficientes que maximizan la varianza $s_{y_2}^2 = \mathbf{w}^T S\mathbf{w}$ restringido a las siguientes dos condiciones:

$${\mathbf{w}_1}^T\mathbf{w}_1 = 1 \tag{2.22}$$

$${\mathbf{w}_1}^T\mathbf{w} = 0 \tag{2.23}$$

Observemos que la condición (2.23) asegura que la primera componente principal $y_1$ y la segunda componente principal $y_2$ no estén correlacionadas, o dicho de otro modo, sean ortogonales.

La solución, al igual que para la primera componente principal, se da como el punto estacionario con respecto a $\mathbf{w}$ para la función de Lagrange:

$$L(\mathbf{w}, \lambda, \gamma) = \mathbf{w}^T S\mathbf{w} - \lambda\left(\mathbf{w}^T\mathbf{w} - 1\right) - \gamma\left(\mathbf{w}^T\mathbf{w}_1 - 0\right) = \mathbf{w}^T S\mathbf{w} - \lambda\left(\mathbf{w}^T\mathbf{w} - 1\right) - \gamma\left(\mathbf{w}^T\mathbf{w}_1\right) \tag{2.24}$$

donde $\lambda$, $\gamma$ son multiplicadores de Lagrange.

De forma análoga a la primera componente principal, ahora tomamos la derivada parcial respecto a $\mathbf{w}$ e igualamos a cero, de este modo se obtiene la siguiente igualdad:

$$\frac{\partial L\left(\mathbf{w}, \lambda, \gamma\right)}{\partial \mathbf{w}} = 2S\mathbf{w} - 2\lambda\mathbf{w} - \gamma\mathbf{w}_1 = 0 \tag{2.25}$$

Si multiplicamos por la izquierda el vector propio $\mathbf{w}_1$ correspondiente a el máximo valor propio $\lambda_1$, obtenemos:

$$2\mathbf{w}_1^T S\mathbf{w} - 2\lambda\mathbf{w}_1^T\mathbf{w} - \gamma\mathbf{w}_1^T\mathbf{w}_1 \tag{2.26}$$

Del requerimiento de ortogonalidad a la primera componente principal (ecuación 2.23) y la condición (2.22), se obtiene la siguiente expresión:

$$2\lambda_1{\mathbf{w}_1}^T\mathbf{w}_2 - \gamma = 0 \tag{2.27}$$

Despejando $\gamma$ se obtiene:

$$\gamma = 2\lambda_1 {\mathbf{w}_1}^T \mathbf{w} \tag{2.28}$$

pero usando la restricción ${\mathbf{w}_1}^T \mathbf{w} = 0$, entonces $\gamma = 0$, sustituyendo este valor en la ecuación (2.25), se obtiene:

$$2S\mathbf{w} - 2\lambda\mathbf{w} = 0 \tag{2.29}$$

Reordenando y reduciendo, llegamos finalmente a la siguiente expresión:

$$S\mathbf{w} = \lambda\mathbf{w} \tag{2.30}$$

La solución es el vector propio $\mathbf{w}_2 = (w_{21}, w_{22})^T$ correspondiente al segundo valor propio más grande de la matriz muestral de varianza-covarianza, esto es, $\lambda_2$.

De este modo la segunda componente principal queda dada por la siguiente ecuación:

$$y_2 = w_{21}x_1 + w_{22}x_2 = {\mathbf{w}_2}^T \mathbf{x} \tag{2.31}$$

Análogamente al caso anterior (ver obtención de ecuación 2.21), la varianza de la segunda componente principal, $y_2$, estará dada por:

$$s_{y_2}^2 = {\mathbf{w}_2}^T S \mathbf{w}_2 = \lambda_2 \tag{2.32}$$

En resumen, el PCA basado en los datos bidimensionales dados en (2.7) es esencialmente un problema de encontrar los valores propios y vectores propios de la matriz muestral de varianza-covarianza $S$. [23]

### 2.3.3. Proceso de derivar propiedades y componentes principales

En general, denotamos como $x = (x_1, x_2, \cdots, x_p)^T$ las $p$ variables que representan las características de los individuos. Con base en los datos $p-dimensionales$ $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n$ observados para estas $p$ variables, podemos obtener la matriz de varianza-covarianza de la siguiente fórmula:

$$S = (s_{jk}) = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \tag{2.33}$$

donde $\bar{\mathbf{x}}$ es el vector de media muestral y $s_{jk} = \sum_{i=1}^{n}(x_{ij} - \bar{x_j})(x_{ij} - \bar{x_k})/n$.

Siguiendo el concepto básico de la derivación de componentes principales como se describió anteriormente para datos bidimensionales, primero proyectamos los $n$ datos $p-dimensionales$ observados en el eje de proyección:

$$y = w_1x_1 + w_2x_2 + \cdots w_px_p = \mathbf{w}^T\mathbf{x} \tag{2.34}$$

entonces obtenemos los datos unidimensionales $y_i = \mathbf{w}^T\mathbf{x}_i$ con $i = 1, 2, \cdots, n$.

Ya que la media de los datos proyectados es $\bar{\mathbf{y}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{w}^T\mathbf{x}_i = \mathbf{w}^T\bar{\mathbf{x}}$, la varianza la podemos expresar del siguiente modo:

$$\begin{aligned} s_y^2 &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}(\mathbf{w}^T\mathbf{x}_i - \mathbf{w}^T\bar{\mathbf{x}})^2 \\ &= \mathbf{w}^T\frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T\mathbf{w} \\ &= \mathbf{w}^T S\mathbf{w} \end{aligned} \tag{2.35}$$

En consecuencia, de la misma manera que se aplicó para dos dimensiones, el vector de coeficientes que maximiza la varianza de los datos proyectados se puede dar como el vector propio $\mathbf{w}_1$ correspondiente al valor propio máximo $\lambda_1$ de la matriz muestral de varianza-covarianza $S$. El eje de proyección $y_1 = {\mathbf{w}_1}^T\mathbf{x}$ que tiene a $\mathbf{w}_1$ como vector de coeficientes es entonces el primer componente principal. De la misma manera, nuevamente, la varianza en este eje de proyección es el valor propio máximo $\lambda_1$.

El segundo componente principal es el eje que, bajo el requisito de ortogonalidad del primer componente principal, maximiza la varianza de los datos $p-dimensionales$ proyectados, y es por tanto el eje de proyección generado por el vector propio $\mathbf{w}_2$, que corresponde al segundo mayor valor propio $\lambda_2$ de la matriz $S$. Continuando de la misma manera, el tercer componente principal se define como el eje que, bajo el requisito de ortogonalidad al primer y segundo componentes principales, maximiza la varianza de los datos $p-dimensionales$ proyectados. Mediante sucesivas repeticiones de este proceso, podemos derivar $p$ componentes principales para las combinaciones lineales de las variables originales.

De este modo, PCA se convierte en un problema de encontrar los valores propios de la matriz muestral de varianza-covarianza $S$. [23]

### 2.3.4. Valores propios de la matriz muestral de varianza-covarianza y PCA

Sea $S$ una matriz muestral de varianza-covarianza basada en $n$ datos $p-dimensionales$ observados. Como se ve en la definición de (2.33), es una matriz simétrica de orden $p$.

Denotamos los $p$ valores propios como:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_i \geq \cdots \geq \lambda_p \geq 0 \tag{2.36}$$

dados como la solución de la ecuación característica de $S$, $\mid S - \lambda I_p \mid = 0$.

Además, denotamos los vectores propios $p - dimensionales$ normalizados a la longitud 1 correspondientes a estos valores propios como:

$$\mathbf{w}_1 = \begin{pmatrix} w_{11} \\ w_{12} \\ \vdots \\ w_{1p} \end{pmatrix}, \mathbf{w}_2 = \begin{pmatrix} w_{21} \\ w_{22} \\ \vdots \\ w_{2p} \end{pmatrix}, \cdots, \mathbf{w}_p = \begin{pmatrix} w_{p1} \\ w_{p2} \\ \vdots \\ w_{pp} \end{pmatrix} \tag{2.37}$$

Para estos vectores propios, se establece, la normalización ${\mathbf{w}_i}^T\mathbf{w}_i = 1$ a la longitud 1 y la ortogonalidad $\mathbf{w}_i^T\mathbf{w}_j = 0$ $(i \neq j)$. Las $p$ componentes principales y sus varianzas expresadas en términos de la combinación lineal de las variables originales ahora se pueden dar en el orden siguiente:

$$\begin{aligned} y_1 &= w_{11}x_1 + w_{12}x_2 + \cdots + w_{1p}x_p = {\mathbf{w}_1}^T\mathbf{x}, \quad var(y_1) = \lambda_1, \\ y_2 &= w_{21}x_1 + w_{22}x_2 + \cdots + w_{2p}x_p = {\mathbf{w}_2}^T\mathbf{x}, \quad var(y_2) = \lambda_2, \\ &\vdots \\ y_p &= w_{p1}x_1 + w_{p2}x_2 + \cdots + w_{pp}x_p = {\mathbf{w}_p}^T\mathbf{x}, \quad var(y_p) = \lambda_p \end{aligned} \tag{2.38}$$

Al aplicar PCA, es posible reducir la dimensionalidad de los $n$ datos $p-dimensionales$ observados $\{x_i = (x_{i1}, x_{12}, \cdots, x_{ip}); \quad i = 1, 2, \cdots, n\}$ para las $p$ variables originales a un número menor, usando solo los primeros componentes principales; por ejemplo, a los datos bidimensionales $\{(y_{i1}, y_{i2}); \quad i = 1, 2, \cdots, n\}$ utilizando sólo la primera y la segunda componente principal, donde $y_{i1} = {\mathbf{w}_1}^T\mathbf{x}_i$ y $y_{i2} = {\mathbf{w}_2}^T\mathbf{x}_i$.

Al proyectar así el conjunto de datos del espacio de dimensiones superiores en un plano bidimensional, podemos captar visualmente la estructura de datos. Además, al encontrar el significado de las nuevas variables combinadas como combinaciones lineales de las variables originales, podemos extraer información útil.

El significado de las componentes principales se puede entender en términos de la magnitud y el signo de los coeficientes $w_{ij}$ de cada variable. Asimismo, la correlación entre las componentes principales y las variables como indicador cuantitativo es de gran utilidad para identificar las variables que influyen en las componentes principales.

La correlación entre la $i$-ésima componente principal $y_i$ y la $j$-ésima variable $x_j$ está dada por:

$$r_{y_i,x_j} = \frac{cov(y_i, x_j)}{\sqrt{var(y_i)}\sqrt{var(x_j)}} = \frac{\lambda_i w_{ij}}{\sqrt{\lambda_i}\sqrt{s_{jj}}} = \frac{\sqrt{\lambda_i} w_{ij}}{\sqrt{s_{jj}}} \tag{2.39}$$

donde $\lambda_i$ es la varianza de la $i$-ésima componente principal, $w_{ij}$ es el coeficiente de la variable $x_j$ para la $i$-ésima componente principal y $s_{jj}$ es la varianza de la variable $x_j$.

La matriz de varianza-covarianza muestral $S$ dada por (2.33) es una matriz simétrica de orden $p$, con la siguiente relación entre sus valores propios y vectores propios:

$$Sw_i = \lambda_i w_i, \quad {w_i}^T w_i = 1, \quad {w_i}^T w_j = 0 \quad (1 \neq j) \tag{2.40}$$

para $i, j = 1, 2, \cdots, p$.

Denotamos como $W$ la matriz de orden $p$ que tiene $p$ vectores propios como columnas, y como $\Lambda$ la matriz de orden $p$ que tiene los valores propios como sus elementos diagonales, es decir:

$$W = (w_1, w_2, \cdots, w_p), \qquad \Lambda = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_p \end{pmatrix} \tag{2.41}$$

La relación entre los valores propios y los vectores propios de la matriz muestral de varianza-covarianza $S$ dada por (2.33) puede expresarse entonces de la siguiente forma:

1. $SW = W\Lambda, \quad W^T W = I_p$.
2. $W^T S W = \Lambda$.
3. $S = W\Lambda W^T = \lambda_1 \mathbf{w}_1 {\mathbf{w}_1}^T + \lambda_2 \mathbf{w}_2 {\mathbf{w}_2}^T + \cdots + \lambda_p \mathbf{w}_p {\mathbf{w}_p}^T$
4. $traza(S) = traza(W\Lambda W^T) = traza(\Lambda) = \lambda_1 + \lambda_2 + \cdots + \lambda_p$.

La ecuación (2) muestra que la matriz simétrica $S$ puede ser diagonalizada por la matriz ortogonal $W$.

La ecuación (3) se conoce como la descomposición espectral de la matriz simétrica $S$.

La ecuación (4) muestra que la suma, $traza(S) = s_{11} + s_{22} + \cdots + s_{pp}$ de las varianzas de las variables originales $x_1, x_2, \cdots, x_p$ es igual a la suma, $traza(\Lambda) = \lambda_1 + \lambda_2 + \cdots + \lambda_p$ de las varianzas de las $p$ componentes principales construidas.

Cuando las unidades en las que se expresan las diferentes características de los datos difieren sustancialmente, es necesario estandarizar los datos observados.

Para los $n$ datos $p-dimensionales$ observados:

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \cdots, x_{ip})^T, \quad i = 1, 2, \cdots, n \tag{2.42}$$

primero obtenemos el vector de media muestral $\bar{\mathbf{x}} = (\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \cdots, \bar{\mathbf{x}}_p)^T$ y la matriz muestral de varianza-covarianza $S = (s_{jk})$.

Normalizamos los datos $p-dimensionales$ en (2.42) de modo que:

$$\mathbf{z}_i = (z_{i1}, z_{i2}, \cdots, z_{ip})^T, \quad z_{ij} = \frac{x_{ij} - \bar{x}_j}{\sqrt{s_{jj}}}, \quad j = 1, 2, \cdots, p. \tag{2.43}$$

La varianza muestral $(s^*_{jj})$ y la covarianza muestral $(s^*_{jk})$ basadas en estos datos $p-dimensionales$ estandarizados están dadas por:

$$s^*_{jk} = \frac{1}{n}\sum_{i=1}^{n} z_{ij} z_{ik} = \frac{1}{n}\sum_{i=1}^{n} \frac{(x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)}{\sqrt{s_{jj}}\sqrt{s_{kk}}} \equiv r_{jk} \tag{2.44}$$

para $j, k = 1, 2, \cdots, p$.

En la matriz muestral de varianza-covarianza con los datos estandarizados, todos los elementos diagonales $r_{jj}$ son en consecuencia 1, y obtenemos la matriz $R$ de dimensión $p \times p$ con $r_{jk}$ como elementos no diagonales, los coeficientes de correlación muestral entre la $j$-ésima y la $k$-ésima variable se conoce como la matriz muestral de correlación.

El PCA que comienza con datos multidimensionales estandarizados se convierte así en un problema de encontrar valores y vectores propios de la matriz muestral de correlación $R$, que se realiza mediante el mismo proceso de derivación de componentes principales que el PCA comenzando con una matriz muestral de varianza-covarianza. [23]

### 2.3.5. Reducción de dimensiones y pérdida de información

Suele ocurrir en la reducción dimensional a traves de las componente principales, que haya perdida de información, por lo que es necesaria una medida cuantitativa de esta pérdida.

Para minimizar la pérdida de información en la reducción de dimensiones, es fundamental encontrar el eje de proyección que proporcione la mayor dispersión de los datos proyectados.

En el PCA, la varianza proporciona una medida de información y, por lo tanto, la pérdida de información se puede estimar cuantitativamente a partir de los tamaños

relativos de las varianzas de las componentes principales. Como se muestra en (2.38), la varianza de la componente principal viene dada por el valor propio de la matriz muestral de varianza-covarianza. En resumen, podemos utilizar $\frac{\lambda_1}{(\lambda_1+\lambda_2+\cdots+\lambda_p)}$ para evaluar qué proporción de la información contenida en las $p$ variables originales está presente en la primera componente principal $y_1$. En general, la siguiente ecuación se utiliza como medida de la información presente en la $i$-ésima componente principal $y_i$:

$$\frac{\lambda_i}{\lambda_1 + \lambda_2 + \cdots + \lambda_p} \tag{2.45}$$

De manera similar, el porcentaje de varianza explicada por las primeras $k$ componentes principales viene dada por:

$$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_k + \cdots + \lambda_p} \tag{2.46}$$

Retomando la relación dada anteriormente $traza(S) = traza(\Lambda)$ entre los valores y vectores propios de la matriz simétrica $S$, se tiene que las $p$ componentes principales $y_1, y_2, \cdots, y_p$ tomadas en conjunto contienen toda la información contenida en las variables originales $x_1, x_2, \cdots, x_p$ (es decir, la suma de sus varianzas) pero la pérdida de información ocurre en la reducción de dimensiones, lo que reduce el número de componentes principales que se utilizan realmente. En conclusión, la medida cuantitativa de la información contenida en los componentes principales que realmente se utilizan viene dada por la relación entre la suma de las varianzas de esos componentes principales y la suma de las varianzas de todos los componentes principales, que por lo tanto sirve como la medida de pérdida de información. [23]

## 2.4. Análisis de Clústeres

El análisis de clústeres, se refiere a una colección de métodos que están diseñados para descubrir grupos naturales, llamados clústeres, en los datos. La idea es que los grupos deben contener objetos similares entre sí, y los grupos deben ser lo más diferentes posible.

Los grupos se desconocen antes de aplicar el método, por lo que a menudo se denominan técnicas no supervisadas. Descubrir grupos naturales puede conducir a una mejor comprensión del conjunto de datos, mostrando las relaciones entre las observaciones o creando efectivamente una clasificación de las observaciones, y puede ser un objetivo final del análisis en sí mismo. En otros entornos, el agrupamiento es un paso previo al procesamiento y los grupos encontrados se investigan más a fondo como parte de algún otro análisis. La agrupación generalmente está diseñada para encontrar grupos de observaciones, pero también se puede usar para encontrar grupos de variables.

Las aplicaciones de la agrupación en clústeres son tan amplias como lo son los métodos; de hecho, los métodos de agrupamiento a menudo se adaptan a sus aplicaciones. [13]

### 2.4.1. Tipos de agrupamientos

Hay dos tipos de modelado de clústeres: **hard-clustering** (agrupamiento duro) y **soft-clustering** (agrupamiento suave). Un punto de datos puede pertenecer a un solo grupo en un modelo de agrupamiento duro. Un punto de datos puede pertenecer a varios clústeres con diferentes probabilidades en un modelo de agrupamiento suave. El modelo de agrupamiento suave es útil cuando los puntos de datos exhiben propiedades de más de un grupo. Un punto de datos se denomina valor atípico si no pertenece a ningún grupo.

La **Figura 2.3 (a)** describe un ejemplo de un agrupamiento duro y la Figura **Figura 2.3 (b)** da un ejemplo de un agrupamiento suave a través de un fuzzy cluster (agrupamiento difuso).

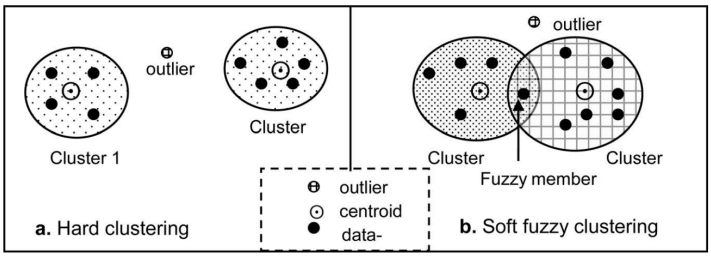



**Figura 2.3:** (a) Hard-clustering vs. (b) soft-clustering. [24]

En el agrupamiento suave, dos clústeres pueden superponerse. El agrupamiento suave se puede derivar usando una mezcla gaussiana de modelos, donde cada distribución gaussiana corresponde a un solo grupo, y dos distribuciones gaussianas puede superponerse. Un punto en las regiones superpuestas pertenece a ambos grupos. [24]

A continuación profundizaremos en dos de las técnicas de agrupamientos empleadas en el presente proyecto, estas son; K-means clustering y Fuzzy c-Means.

### 2.4.2. Introducción a k-means

El término "K-means" fue utilizado por primera vez por James MacQueen en 1967, aunque la idea se remonta a Hugo Steinhaus en 1956. El algoritmo estándar fue propuesto por primera vez por Stuart Lloyd de Bell Labs en 1957 como una técnica para la modulación de código de pulso, aunque no se publicó como artículo de revista hasta 1982. En 1965, Edward W. Forgy publicó esencialmente el mismo método, razón por

la cual a veces se lo conoce como el algoritmo de Lloyd-Forgy.

El algoritmo más común utiliza una técnica de refinamiento iterativo. Debido a su ubicuidad, a menudo se le llama "el algoritmo de K-means"; también se lo conoce como algoritmo de Lloyd, esto particularmente en la comunidad de ciencias de la computación. [25]

La agrupación en clústeres de K-means es el algoritmo de aprendizaje automático no supervisado más utilizado para dividir un conjunto de datos dado en un conjunto de $k$ grupos (es decir, $k$ clústeres), donde $k$ representa el número de grupos pre-especificados por el analista. Clasifica objetos en múltiples clústeres de modo que los objetos dentro del mismo clúster sean lo más similares posible (es decir, alta similitud intraclase), mientras que los objetos de diferentes clústeres sean lo más diferentes posible (es decir, baja inter-similitud de clase). En el agrupamiento de K-means, cada grupo está representado por su centro (centroide) que corresponde a la media de puntos asignados al grupo. La idea básica detrás de la agrupación de K-means consiste en definir agrupaciones de modo que se minimice la variación total dentro de la agrupación. [26]

La agrupación K-means, pertenece a técnicas basadas en particiones, estas se basan en la reubicación iterativa de puntos de datos entre agrupaciones. Se utiliza para dividir los casos o las variables de un conjunto de datos en grupos o conglomerados que no se superponen, según las características descubiertas. La técnica de agrupación de K-means también se puede describir como un modelo de centroide, ya que un vector que representa la media se utiliza para describir cada agrupación. Su facilidad de implementación, eficiencia computacional y bajo consumo de memoria ha hecho que el agrupamiento de K-means sea muy popular. [27]

**El algoritmo básico de K-means**

Comenzamos con una descripción del algoritmo básico. Primero elegimos $K$ centroides iniciales, donde $K$ es un parámetro especificado por el usuario, es decir, el número de conglomerados deseados. Luego, cada punto se asigna al centroide más cercano, y cada conjunto de puntos asignados a un centroide es un grupo. El centroide de cada grupo se actualiza luego en función de los puntos asignados al grupo. Repetimos la asignación y actualizamos los pasos hasta que ningún punto cambie de grupo, o de manera equivalente, hasta que los centroides permanezcan iguales. K-means se describe formalmente mediante el **Algoritmo 1**.

Ilustramos el funcionamiento de K-means en la **Figura 2.4**, que muestra cómo, a partir de tres centroides, los grupos finales se encuentran en cuatro pasos de actualización de asignación. Cada subfigura muestra (1) los centroides al comienzo de la iteración y (2) la asignación de los puntos a esos centroides. Los centroides están indicados por el símbolo (+); todos los puntos que pertenecen al mismo grupo tienen la misma forma

**Algoritmo 1:** Algoritmo básico de K-means

1 Seleccionar K puntos como centroides iniciales

2 **repetir**

3 Forme K clústeres asignando cada punto a su centroide más cercano

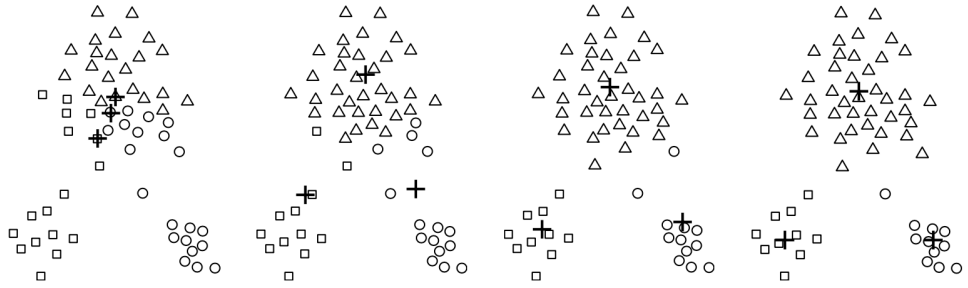
4 Vuelva a calcular el centroide de cada grupo

5 **mientras** que los centroides no cambien.

de marcador.

En el primer paso del Algoritmo 1, que se muestra en la Figura 2.4 (a), los puntos se asignan a los centroides iniciales, que están todos en el grupo más grande de puntos. Para este ejemplo, usamos la media como centroide. Una vez asignados los puntos a un centroide, el centroide se actualiza. Nuevamente, la figura de cada paso muestra el centroide al comienzo del paso y la asignación de puntos a esos centroides.

En el segundo paso, los puntos se asignan a los centroides actualizados y los centroides se actualizan nuevamente. En los pasos 2, 3 y 4, que se muestran en las Figuras 2.4 (b), (c) y (d), respectivamente, dos de los centroides se mueven a los dos pequeños grupos de puntos en la parte inferior de las figuras.



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4.

**Figura 2.4:** Uso del algoritmo de K-means para encontrar tres clústeres en datos de muestra. [27]

Cuando el algoritmo de K-means termina en la Figura 2.4 (d), debido a que no ocurren más cambios, los centroides han identificado las agrupaciones naturales de puntos.

Para algunas combinaciones de funciones de proximidad y tipos de centroides, *K*-

means siempre converge a una solución; es decir, K-means alcanza un estado en el que ningún punto se desplaza de un grupo a otro y, por tanto, los centroides no cambian. Sin embargo, debido a que la mayor parte de la convergencia ocurre en los primeros pasos, la condición en la línea 5 del Algoritmo 1 a menudo se reemplaza por una condición más débil, por ejemplo, repetir hasta que solo el 1 % de los puntos cambien de grupo. [27]

### 2.4.3. Criterio de K-means

El criterio de agrupación en clústeres particional más utilizado se basa en el **criterio de error al cuadrado**. El objetivo general es obtener esa partición que, para un número fijo de clústeres, minimice el error cuadrado, también llamado variación dentro del clúster (within-cluster variation), definimos a continuación este concepto.

La variación dentro del clúster (within-cluster variation) $C_k$ es una medida $W(C_k)$ de la cantidad en que las observaciones dentro de un clúster difieren entre sí.

Matemáticamente, queremos resolver el siguiente problema de optimización:

$$\min_{C_1,\ldots,C_k} \sum_{k=1}^{K} W(C_k) \tag{2.47}$$

En otras palabras, esto significa que queremos dividir los puntos de datos en clústeres de modo que la variación total dentro del clúster sumada en todos los $K$ clústeres sea lo más pequeña posible.

Hay muchas formas posibles de definir la variación dentro del clúster $W(C_k)$, pero, con mucho, la opción más común implica la distancia euclidiana al cuadrado.

Entonces definimos:

$$W(C_k) = \frac{1}{|C_l|} \sum_{i,j \in C_l} \|x_i - x_j\|^2 \tag{2.48}$$

donde $|C_l|$ denota el número de observaciones en el $k$-ésimo clúster. [28]

De este modo la variación dentro del clúster para el $k$-ésimo clúster es la suma de todas las distancias euclidianas cuadradas de las observaciones en el $k$-ésimo clúster, dividida por el número total de observaciones en el $k$-ésimo clúster.

Ahora volvamos a definir formalmente el problema de optimización [29]:

Dado $x_1, ..., x_n \in \mathbb{R}^d$, dividimos estos puntos en $k$ clústeres $C_1, ..., C_k$ basado en la función objetivo:

$$\min_{C_1,...,C_k} \sum_{l=1}^{k} W(C_k) = \min_{C_1,...,C_k} \sum_{l=1}^{k} \frac{1}{|C_l|} \sum_{i,j \in C_l} \|x_i - x_j\|^2 \tag{2.49}$$

Esta función de costo es un promedio ponderado de las varianzas de los clústeres, con ponderaciones proporcionales al tamaño del clúster en términos del número de observaciones $|C_l|$. Derivemos ahora una forma más tratable de ésta expresión.

Sea $\mu_l = \frac{1}{|C_l|} \sum_{i \in C_l}^{n} x_i$ el centroide, esto es el centro de masa de las observaciones en el cluster $C_l$. Notemos que:

$$\begin{aligned}
\sum_{i,j \in C_l} \|x_i - x_j\|^2 &= \sum_{i,j \in C_l} \left( \|x_i{}^2\| + \|x_j\|^2 - 2\langle x_i, x_j \rangle \right) \\
&= \sum_{i \in C_l} \left( |C_l| \|x_i\|^2 + \sum_{j \in C_l} \|x_j\|^2 - 2|C_l| \langle x_i, \mu_l \rangle \right) \\
&= 2|C_l| \sum_{i \in C_l} \|x_i\|^2 - 2|C_l|^2 \|\mu_l\|^2
\end{aligned}$$

Por otro lado:

$$\begin{aligned}
\sum_{i \in C_l} \|x_i - \mu_l\|^2 &= \sum_{i \in C_l} \left( \|x_i{}^2\| + \|\mu_l\|^2 - 2\langle x_i, \mu_l \rangle \right) \\
&= \sum_{i \in C_l} \left( \|x_i\|^2 + |C_l| \|\mu_l\|^2 - 2|C_l| \|\mu_l\|^2 \right) \\
&= \sum_{i \in C_l} \|x_i\|^2 - |C_l| |\mu_l|^2
\end{aligned}$$

De este modo obtenemos:

$$\frac{1}{2} \sum_{i,j \in C_l} \|x_i - x_j\|^2 = |C_l| \sum_{i \in C_l} \|x_i - \mu_l\|^2$$

y por lo tanto minimizar la ecuación (2.49) es equivalente a resolver:

$$\min_{C_1,...,C_k} \sum_{j=1}^{k} \sum_{i \in C_j} \|x_i - \mu_j\|^2 \tag{2.50}$$

De este modo, vemos que tenemos el problema equivalente de minimizar la suma de errores cuadrados también conocida como SSE (sum of the squared error), esto es:

$$SSE = \sum_{j=1}^{k} \sum_{i \in C_j} \|x_i - \mu_j\|^2 \tag{2.51}$$

Ahora demostraremos la razón detrás de elegir la media de los puntos de datos en un clúster como el prototipo representativo de un clúster en el algoritmo de $k$-means. Denotemos $C_k$ como el $k$-ésimo clúster, $x_i$ es un punto en $C_k$ y $\mu_k$ es la media del $k$-ésimo clúster. Podemos resolver para el representante de $C_j$ que minimiza el SSE diferenciando el SSE con respecto a $\mu_j$ e igualandolo a cero. [30]

$$\begin{aligned}
\frac{\partial}{\partial \mu_j} SSE &= \frac{\partial}{\partial \mu_j} \sum_{k=1}^{K} \sum_{x_i \in C_k} (x_i - \mu_k)^2 \\
&= \sum_{k=1}^{K} \sum_{x_i \in C_j} \frac{\partial}{\partial \mu_j} (x_i - \mu_j)^2 \\
&= \sum_{x_i \in C_j} 2 * (x_i - \mu_j) = 0
\end{aligned}$$

De modo que:

$$\sum_{x_i \in C_j} 2 * (x_i - \mu_j) = 0 \Rightarrow |C_j| \cdot \mu_j = \sum_{x_i \in C_j} x_i \Rightarrow \mu_j = \frac{\sum_{x_i \in C_j} x_i}{|C_j|}$$

Por lo tanto, el mejor representante para minimizar la SSE de un clúster es la media de los puntos en el clúster. En $k$-means, el SSE disminuye monótonamente con cada iteración. Este comportamiento monótonamente decreciente eventualmente convergerá a un mínimo local.

Lo anteriormente expuesto, se formaliza en el siguiente Lemma. [29]

**Lemma 1.** Sea $x_1, ..., x_n$ un conjunto de puntos. La expresión $\sum_{i=1}^{n} \|x_i - \mu\|^2$ es minimizada cuando $\mu$ es el centroide, i.e, $\mu = \frac{1}{n} \sum_{i=1}^{n} x_i$.

Por lo tanto, si conociéramos los centros correctos $\mu_j$, podríamos asignar fácilmente cada $x_i$ a los grupos correctos resolviendo:

$$\min_j ||x_i - \mu_j||^2$$

Sin embargo, determinar los vectores $\mu_j$ es computacionalmente difícil y encontrar la solución a el problema de $k$-means planteado en la ecuación (2.50) es un problema

altamente no-convexo y encontrar su solución es un problema NP-hard. Debido a esto aunque el esquema de usar la función objetivo de $k$-means es bastante popular en aplicaciones prácticas de clustering, resulta que encontrar la solución óptima de $k$-means es a menudo computacionalmente inviable. Como alternativa, a menudo se usa el siguiente algoritmo iterativo simple, con tanta frecuencia que, en muchos casos, el término clustering por $k$-means se refiere al resultado de este algoritmo en lugar de al clustering que minimiza la función de costo de $k$-means. Este algoritmo alternativo sugerido por el Lemma 1, es llamado el algoritmo estándar o **algoritmo de Lloyd** (1982). [29]

A continuación se muestra el esquema del algoritmo [31]:

**Algoritmo 2:** Algoritmo de Lloyd

**Entrada:** $X \subset \mathbb{R}^n$ y número de clústeres $k$

**Inicialización:** Seleccionar aleatoriamente los centroides inciales $\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_k$

**repetir hasta convergencia**

    $\forall\, i \in [k]$ establece $C_i = \left\{\mathbf{x} \in X : i = argmin_j \parallel \mathbf{x} - \boldsymbol{\mu}_j \parallel\right\}$ (romper el ciclo de alguna manera arbitraria)

    $\forall\, i \in [k]$ actualizar $\boldsymbol{\mu}_i = \frac{1}{|C_i|} \sum_{\mathbf{x} \in C_i} \mathbf{x}$

Podemos ver de forma más intuitiva que el algoritmo consta de los siguientes pasos [29]:

**Algoritmo de Lloyd:**

1. Elegir $k$ centroides iniciales $\mu_1, ..., \mu_k$.
2. Asignar cada punto de datos a sus centroides de clúster más cercanos.
3. Actualizar el centroide como la media de todos los objetos en cada clúster.
4. Repetir los dos pasos anteriores hasta la convergencia a algún criterio de parada.

A continuación veamos algunas observaciones [32]:

1. El algoritmo de Lloyd es esencialmente un esquema de minimización alterno, haciendo también que la función de costo decrezca.
2. No hay garantía de que las iteraciones generadas a partir del algoritmo de Lloyd converjan en el mínimo global. Se garantiza un punto estacionario.
3. El algoritmo de Lloyd es una variante del algoritmo EM (maximización de expectativas) aplicado al modelo de mezcla gaussiana.
4. La principal ventaja del algoritmo de Lloyd es su eficiencia computacional.

### 2.4.4. Fuzzy C-Means

Realizar asignaciones estrictas de puntos a conglomerados no es factible en conjuntos de datos complejos donde hay clústeres superpuestos. Para extraer dichas estructuras superpuestas, se puede utilizar un algoritmo de agrupación difusa. Este método (desarrollado por Dunn en 1973 y mejorado por Bezdek en 1981) se usa con frecuencia en el reconocimiento de patrones. [30]

A continuación se muestran los aspectos matemáticos de esta técnica de agrupación [33]:

Sea $\mathbf{X} = \{x_{is} : i = 1, ..., n;\ s = 1, ..., p\} = \{\mathbf{x}_i = (x_{i1}, ..., x_{is}, ..., x_{ip})'\ i = 1, ..., n\}$ una matriz de datos, donde $x_{is}$ representa la $s$-ésima variable cuantitativa observada en el $i$-ésimo objeto y $\mathbf{x}_i$ representa el vector de la $i$-ésima observación. El método de agrupamiento FCM propuesto por Bezdek (1981) se formaliza de la siguiente manera:

$$min : \sum_{i=1}^{n}\sum_{k=1}^{c} u_{ik}^{m} d_{ik}^{2} = \sum_{i=1}^{n}\sum_{k=1}^{c} u_{ik}^{m} \parallel \mathbf{x}_i - \mathbf{h}_k \parallel^2 \quad s.t \sum_{k=1}^{c} u_{ik} = 1, u_{ik} \geq 0 \tag{2.52}$$

donde $u_{ik}$ denota el grado de pertenencia del $i$-ésimo objeto al $k$-ésimo clúster; $d_{ik}^2 = \parallel \mathbf{x}_i - \mathbf{h}_k \parallel^2$ es la distancia euclidiana al cuadrado entre el $i$-ésimo objeto y el centroide del $k$-ésimo clúster; $\mathbf{h}_k = (h_{k1}, ..., h_{ks}, ..., h_{kp})'$ representa el centroide $k$-ésimo, donde $h_{ks}$ indica el componente s-ésimo (variable s-ésima) del $k$-ésimo vector centroide; $m > 1$ es un parámetro que controla la borrosidad (fuzziness) de la partición.

Resolviendo el problema de optimización con restricciones (2.52) con el método de los multiplicadores de Lagrange, las soluciones iterativas óptimas condicionales son:

$$u_{ik} = \frac{1}{\sum_{k'=1}^{c} \left[ \frac{\parallel \mathbf{x}_i - \mathbf{h}_k \parallel}{\parallel \mathbf{x}_i - \mathbf{h}_{k'} \parallel} \right]^{\frac{2}{m-1}}} = \frac{\parallel \mathbf{x}_i - \mathbf{h}_k \parallel^{-\frac{2}{m-1}}}{\sum_{k'=1}^{c} \parallel \mathbf{x}_i - \mathbf{h}_{k'} \parallel^{-\frac{2}{m-1}}}, \quad \mathbf{h}_k = \frac{\sum_{i=1}^{n} u_{ik}^{m} \mathbf{x}_i}{\sum_{i=1}^{n} u_{ik}^{m}} \tag{2.53}$$

Cada centroide resume las características del clúster respectivo; en particular, cada centroide representa sintéticamente su clúster en el sentido de que representa un promedio ponderado apropiado de un conjunto de características observadas en los objetos. Por lo tanto, cada centroide puede utilizarse adecuadamente para interpretar cada clúster. Observemos que si aumenta la cohesión interna de los clústeres, aumenta el poder interpretativo/explicativo de los centroides y luego la incertidumbre del agrupamiento, es decir, la incertidumbre (fuzziness) relacionada con el proceso de agrupamiento y medido por medio de los grados de pertenencia de cada objeto a las disminuciones de los agrupamientos.

Para calcular las soluciones iterativas (2.53), se puede considerar el siguiente algoritmo (**Algoritmo 3**, Bezdek, 1981):

**Algoritmo 3:** Algoritmo FCM

1 Fijamos m y una matriz inicial de grados de pertenencia $u_{ik}^{(\alpha)}$ $(i = 1, ..., n;\ k = 1, ..., c)$ con $\alpha = 0$ .

2 Actualizamos los centroides por medio de $h_k^{(\alpha+1)} = \dfrac{\sum_{i=1}^{n} u_{ik}^{m(\alpha)} \mathbf{x}_i}{\sum_{i=1}^{n} u_{ik}^{m(\alpha)}}$ .

3 Actualizamos los grados de afiliación mediante $u_{ik}^{(\alpha+1)} = \sum_{k'=1}^{c} \left( \dfrac{\| \mathbf{x}_i - \mathbf{h}_k^{(\alpha+1)} \|}{\| \mathbf{x}_i - \mathbf{h}_{k'}^{(\alpha+1)} \|} \right)^{\frac{2}{m-1}}$ .

4 Comparamos $u_{ik}^{\alpha}$ con $u_{ik}^{\alpha+1}$ usando una norma matricial conveniente: Si $|u_{ik}^{(\alpha+1)} - u_{ik}^{(\alpha)}| < \tau$ $(i = 1, ..., n;\ \ k = 1, ..., c)$ (donde $\tau$ es un pequeño número positivo establecido por el investigador) paramos, de lo contrario, establecer $\alpha = \alpha + 1$ y volver al paso 2.

El parámetro de borrosidad (fuzziness parameter) $m$ juega un papel importante en el agrupamiento FCM. El valor de $m$ debe ser elegido de antemano. En la literatura se recomiendan diferentes estrategias heurísticas. Los valores demasiado cercanos a 1 darán como resultado una partición cercana con todas las pertenencias cercanas a 0 o 1. Los valores excesivamente grandes conducirán a una superposición desproporcionada con todas las pertenencias cercanas a $1/c$. En consecuencia, no se recomienda ninguno de estos tipos de $m$. Aunque ha habido algunos procedimientos heurísticos empíricos para determinar el valor de $m$, no parece existir una forma teóricamente justificable de seleccionar $m$. En la práctica, $m = 2$ es la opción más popular en el agrupamiento difuso. [33]

De hecho un valor de $m$ grande suprime los valores atípicos en los conjuntos de datos, es decir, cuanto más grande es $m$, más clústeres comparten sus objetos y viceversa. En el límite cundo $m \to 1$, el método se vuelve equivalente al agrupamiento de K-means, mientras que para $m \to \infty$ todos los objetos de datos tienen una pertenencia idéntica a cada clúster. Establecer $m = 2$ puede considerarse un compromiso entre una suposición a priori de una cierta cantidad de borrosidad en el conjunto de datos y la ventaja de evitar un cálculo de su valor que requiere mucho tiempo. Sin embargo, al ajustar cuidadosamente el fuzzificador, debería ser posible optimizar el algoritmo para tener en cuenta el ruido característico presente en el conjunto de datos. [34]

## 2.5. Modelo de Regresión Logística

Antes de profundizar en el modelo de regresión logística veamos un panorama general de los modelos de aprendizaje automático.

Hay tantos tipos diferentes de modelos de aprendizaje automático que es útil clasificarlos en categorías amplias basadas en [43]:

1. Si están capacitados o no con supervisión humana (supervisados, no-supervisados, semi-supervisados y aprendizaje reforzado).
2. Si pueden o no aprender de forma incremental sobre la marcha (aprendizaje en línea o por lotes).
3. Ya sea que funcionen simplemente comparando nuevos puntos de datos con puntos de datos conocidos o que, en su lugar, detecten patrones en los datos de entrenamiento y creen un modelo predictivo, como lo hacen los científicos (aprendizaje basado en instancias vs aprendizaje basado en modelos).
4. Obtener conocimientos sobre problemas complejos y grandes cantidades de datos.

Estos criterios no son exclusivos; pueden combinarse entre ellos. Examinemos más de cerca la categoría del Aprendizaje supervisado y no-supervisado.

**Aprendizaje supervisado y no-supervisado**

En el **aprendizaje supervisado**, los datos de entrenamiento que alimenta al algoritmo incluyen las soluciones deseadas, llamadas etiquetas. Una tarea típica de aprendizaje supervisado es la **clasificación**. Otra tarea típica es predecir un valor numérico objetivo, tal como el precio de un automóvil, dado un conjunto de características (kilometraje, edad, marca, etc...) llamadas predictores. Este tipo de tarea se llama de **regresión**.

Tengamos en cuenta que algunos algoritmos de regresión también se pueden utilizar para la clasificación y viceversa. Por ejemplo, la regresión logística se usa comúnmente para la clasificación, ya que puede generar un valor que corresponde a la probabilidad de pertenecer a una clase determinada.

Estos son algunos de los algoritmos de aprendizaje supervisado más importantes:

1. k-Nearest Neighbors.
2. Linear Regression.
3. Logistic Regression.

4. Support Vector Machines (SVMs)
5. Decision Trees and Random Forests
6. Neural Networks

Por otro lado, en el **aprendizaje no-supervisado**, los datos de entrenamiento no están etiquetados. El sistema intenta aprender sin un maestro.

Estos son algunos de los algoritmos de aprendizaje no-supervisado más importantes [43]:

1. **Clustering:** K-means, DBSCAN, Hierarchical Cluster Analysis (HCA).
2. **Anomaly detection and novelty detection:** One-class SVM, Isolation Forest.
3. **Visualization and dimensionality reduction:** Principal Component Analysis (PCA), Kernel PCA, Locally-Linear Embedding (LLE), t-distributed Stochastic Neighbor Embedding (t-SNE).
4. **Association rule learning:** Apriori, Eclat.

En el presente proyecto se emplea la regresión logística como un algoritmo de clasificación, en ese sentido veamos que los **algoritmos de clasificación** pueden agruparse en dos categorías que son: clasificadores **generativos** y **discriminativos**. [35]

**Clasificadores generativos:**

Consideremos un caso en el que tenemos una característica $x$ y una variable objetivo $y$. Estamos tratando de predecir $y$ basándonos en los valores de $x$.

Los clasificadores generativos aprenden la distribución de probabilidad conjunta $P(x, y)$. La atención se centra en cómo las características y la variable de respuesta ocurren juntas. El objetivo es poder explicar cómo se generan los datos.

Una vez que el modelo captura el proceso que generó los datos, se pueden hacer predicciones sobre los nuevos ejemplos (es decir, puntos de datos). Por lo tanto, el factor clave para los clasificadores generativos es poder aprender la distribución de datos subyacente.

Para hacer predicciones, los clasificadores generativos transforman la probabilidad conjunta ($P(x, y)$) en una probabilidad condicional ($P(y|x)$) utilizando la regla de Bayes. Los clasificadores de bayes ingenuos y los modelos de markov ocultos son ejemplos de clasificadores generativos. Dado que los modelos generativos aprenden la distribución de datos, también se pueden utilizar para generar nuevos datos.

**Clasificadores discriminativos:**

Los clasificadores discriminativos por otro lado, intentan encontrar límites que separen clases. Se comprueban todos los valores umbrales posibles para los límites y se selecciona el que da el error más bajo.

Estos límites pueden ser estrictos o suaves según el algoritmo. Límite suave significa permitir que algunos ejemplos se clasifiquen erróneamente.

La regresión logística, SVM (Support Vector Machine) y clasificadores basados en árboles (por ejemplo, Decision Trees) son ejemplos de clasificadores discriminativos.

Un modelo discriminativo aprende directamente la distribución de probabilidad condicional $P(y|x)$, mientras que recordemos el modelo generativo aprende la probabilidad conjunta $P(x, y)$ y luego la transforma en la condicional $P(y|x)$ usando la regla de Bayes.

Tanto el modelo generativo como el discriminativo tienen ventajas y desventajas. Por ejemplo, los modelos generativos necesitan más datos, se necesitan datos suficientes para poder representar las distribuciones con precisión. Los modelos generativos también son más costosos computacionalmente que los modelos discriminativos. [35]

Por otro lado los modelos discriminativos son más robustos frente a los valores atípicos, ya que los valores atípicos pueden tener un gran impacto en la distribución de datos, lo que afecta negativamente la precisión de los modelos generativos. [36]

Dado este panorama general de los modelos de aprendizaje automático, ahora profundicemos en el método de regresión logística, el cual es la piedra angular del presente proyecto.

### 2.5.1. Regresión Logística

La regresión logística, es un modelo estadístico que en su forma básica utiliza una función logística para modelar una variable dependiente binaria, aunque existen extensiones más complejas. Como vimos en la sección pasada la regresión logística se engloba dentro de la categoría de aprendizaje supervisado y dentro de la categoría de clasificador discriminativo. En el análisis de regresión, la regresión logística (o regresión logit) es la estimación de los parámetros de un modelo logístico (una forma de regresión binaria). Matemáticamente, un modelo logístico binario tiene una variable dependiente con dos valores posibles, como pasa/no pasa, que está representado por una variable indicadora , donde los dos valores están etiquetados como 0 y 1.

Consideremos una única observación de entrada $\mathbf{x}$, la cual representa un vector de

características $[x_1, x_2, ..., x_n]$. La salida del clasificador $y$ puede ser 1 (lo que significa que la observación es un miembro de la clase) o 0 (la observación no es un miembro de la clase). Queremos conocer la probabilidad $P(y = 1|\mathbf{x})$, de que esta observación sea un miembro de la clase. La regresión logística resuelve esta tarea aprendiendo, a partir de un conjunto de entrenamiento, un vector de pesos y un término de sesgo (bias). Cada peso $w_i$ es un número real y está asociado con una de las características de entrada $x_i$. El peso $w_i$ representa la importancia de esa característica de entrada para la decisión de clasificación y puede ser positiva (proporcionando evidencia de que la instancia que se clasifica pertenece a la clase positiva) o negativa (brindando evidencia de que la instancia que se clasifica pertenece a la clase negativa). El término de sesgo, también llamado intersección, es otro número real que se agrega a las entradas ponderadas.

Para tomar una decisión en un elemento del conjunto de prueba, después de haber aprendido los pesos en el entrenamiento, el clasificador primero multiplica cada $x_i$ por su peso $w_i$, suma las características ponderadas y agrega el término de sesgo $b$. El número único resultante $z$ expresa la suma ponderada de la evidencia para la clase. [37]

$$z = (\sum_{i=1}^{n} w_i x_i) + b \tag{2.54}$$

De forma equivalente podemos escribir:

$$z = \mathbf{w} \cdot \mathbf{x} + b \tag{2.55}$$

Pero observemos que nada en la ecuación (2.55) obliga a $z$ a ser una probabilidad permitida, es decir, a estar entre 0 y 1. De hecho, dado que los pesos son valores reales, la salida podría incluso ser negativa; $z$ varía de $-\infty$ a $\infty$.

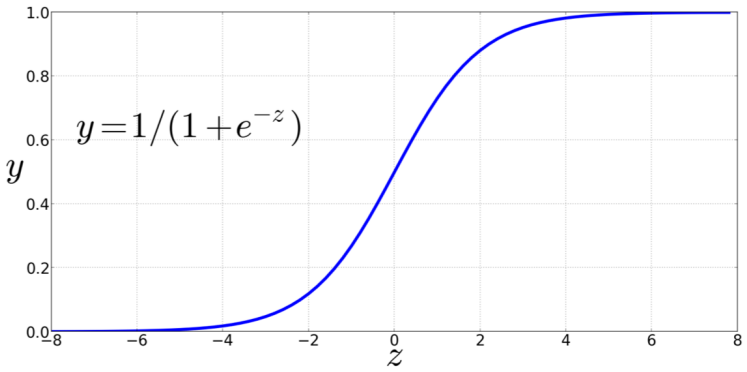



**Figura 2.5:** La función sigmoide $y = \frac{1}{1+e^{-z}}$ toma un valor real y lo asigna al rango $[0, 1]$.

Para obtener una probabilidad válida, pasaremos $z$ a través de la función sigmoide, $\boldsymbol{\sigma}(z)$. La función sigmoide también se llama función logística y le da su nombre a la

regresión logística. El sigmoide tiene la siguiente ecuación, que se muestra gráficamente en la **Figura 2.5**:

$$y = \boldsymbol{\sigma}(z) \tag{2.56}$$

El sigmoide tiene varias ventajas; en primer lugar, toma un número de valor real y lo asigna al rango $[0, 1]$, con lo cual ya tenemos probabilidades válidas, por otro lado, debido a que es casi lineal alrededor de 0 pero se aplana hacia los extremos, tiende a aplastar los valores atípicos hacia 0 o 1. Y finalmente, es diferenciable, lo que, como veremos más adelante será útil para el aprendizaje.

Si aplicamos el sigmoide a la suma de las características ponderadas, obtenemos un número entre 0 y 1. Para convertirlo en una probabilidad, solo necesitamos asegurarnos de que los dos casos, $P(y = 1)$ y $P(y = 0)$ sumen 1. Podemos hacer esto de la siguiente manera:

$$\begin{aligned} P(y=1) &= \boldsymbol{\sigma}(\mathbf{w}\cdot\mathbf{x}+b) \\ &= \frac{1}{1+e^{(-(\mathbf{w}\cdot\mathbf{x}+b))}} \\ P(y=0) &= 1-\boldsymbol{\sigma}(\mathbf{w}\cdot\mathbf{x}+b) \\ &= 1-\frac{1}{1+e^{(-(\mathbf{w}\cdot\mathbf{x}+b))}} \\ &= \frac{e^{(-(\mathbf{w}\cdot\mathbf{x}+b))}}{1+e^{(-(\mathbf{w}\cdot\mathbf{x}+b))}} \end{aligned} \tag{2.57}$$

Ahora tenemos un algoritmo que, dada una instancia $\mathbf{x}$, calcula la probabilidad $P(y = 1|\mathbf{x})$. Ahora podemos tomar la decisión de si se clasifica o no en la clase que estemos considerando, tomando un **límite de decisión**, por ejemplo si el límite de desición es de 0.5, entonces para una instancia de prueba $\mathbf{x}$, decimos que la instancia pertenece a la clase si $P(y = 1|\mathbf{x})$ es mayor que 0.5 y no en caso contrario, esto es:

$$\hat{y} = \begin{cases} 1 & si\ \ P(y=1|\mathbf{x}) > 0.5 \\ 0 & de\ \ lo\ \ contrario \end{cases}$$

Ahora veamos como la regresión logística aprende los parámetros del modelo, estos es, de que manera podemos obtener los pesos $\mathbf{w}$ y el sesgo $b$. La regresión logística es un método de clasificación supervisada en la que conocemos la etiqueta correcta $y$ (0 o 1) para cada observación $\mathbf{x}$. Lo que produce el sistema a través de la ecuación (2.57) es $\hat{y}$, la estimación del sistema de la $y$ verdadera. Queremos encontrar los parámetros ($\mathbf{w}$ y $b$) que hagan que $\hat{y}$ para cada observación de entrenamiento sea lo más cercano

posible a la verdadera $y$.

Esto requiere dos componentes. La primera es una métrica de qué tan cerca está la etiqueta actual $\hat{y}$ de la etiqueta verdadera $y$. En lugar de medir la similitud, generalmente hablamos de lo opuesto a esto: la distancia entre la salida del sistema y la salida verdadera, y llamamos a esta distancia la función de pérdida o la función de costo. En la siguiente sección, presentaremos la función de pérdida que se usa comúnmente para la regresión logística y también para las redes neuronales, la pérdida de entropía cruzada (cross-entropy loss).

Lo segundo que necesitamos es un algoritmo de optimización para actualizar iterativamente los pesos para minimizar esta función de pérdida. El algoritmo estándar para esto es el descenso de gradientes. [37]

### 2.5.2. The cross-entropy loss function

Necesitamos una función de pérdida que exprese, para una instancia $\mathbf{x}$, qué tan cerca está la salida del clasificador ($\hat{y}$) de la salida correcta $y$ (que es 0 o 1). Llamaremos a esto:

$$L(\hat{y}, y) = Cuánto\ difiere\ \hat{y}\ de\ la\ verdadera\ y \tag{2.59}$$

Hacemos esto a través de una función de pérdida que prefiere que las etiquetas de clase correctas de los ejemplos de entrenamiento sean más probables. Esto se denomina **estimación de máxima verosimilitud condicional**: elegimos los parámetros $\mathbf{w}$, $b$ que maximizan la probabilidad logarítmica de las etiquetas $y$ verdaderas en los datos de entrenamiento dadas las observaciones $\mathbf{x}$. La función de pérdida resultante es la pérdida de probabilidad logarítmica negativa, generalmente llamada pérdida de entropía cruzada.

Derivemos esta función de pérdida, aplicada a una sola observación $\mathbf{x}$. Nos gustaría obtener pesos que maximicen la probabilidad de la etiqueta correcta $P(y|\mathbf{x})$. Dado que solo hay dos resultados discretos (1 o 0), esta es una distribución de Bernoulli, y podemos expresar la probabilidad $P(y|\mathbf{x})$ que nuestro clasificador produce para una observación como sigue:

$$P(y|\mathbf{x}) = \hat{y}^y(1-\hat{y})^{1-y} \tag{2.60}$$

Ahora tomamos el logaritmo de ambos lados. Esto resultará útil matemáticamente y no afectará, ya que cualquier valor que maximice una probabilidad también maximizará el logaritmo de la probabilidad:

$$\begin{aligned} log[P(y|\mathbf{x})] &= log[\hat{y}^y(1-\hat{y})^{1-y}] \\ &= y\, log(\hat{y}) + (1-y) log(1-\hat{y}) \end{aligned} \tag{2.61}$$

La ecuación (2.61) describe una probabilidad logarítmica que debe maximizarse. Para convertir esto en una función de pérdida (algo que debemos minimizar), simplemente cambiaremos el signo de la ecuación (2.61). El resultado es la pérdida de entropía cruzada (cross-entropy loss) que denotaremos por $L_{CE}$:

$$L_{CE}(\hat{y}, y) = -log[P(y|\mathbf{x})] = -[y\ log(\hat{y}) + (1-y)\ log(1-\hat{y})] \tag{2.62}$$

Finalmente, podemos reemplazar la definición de $\hat{y} = \boldsymbol{\sigma}(\mathbf{w} \cdot \mathbf{x} + b)$:

$$L_{CE}(\hat{y}, y) = -[y\ log[\boldsymbol{\sigma}(\mathbf{w} \cdot \mathbf{x} + b)] + (1-y)\ log[1 - \boldsymbol{\sigma}(\mathbf{w} \cdot \mathbf{x} + b)]] \tag{2.63}$$

Ahora veamos porque minimizar esta probabilidad logarítmica negativa hace lo que queremos. Un clasificador perfecto asignaría probabilidad 1 al resultado correcto (y = 1 o y = 0) y probabilidad 0 al resultado incorrecto. Eso significa que cuanto mayor sea $\hat{y}$ (cuanto más cerca esté de 1), mejor será el clasificador; cuanto más bajo es $\hat{y}$ (cuanto más cerca está de 0), peor es el clasificador. El logaritmo negativo de esta probabilidad es una métrica de pérdida conveniente ya que va de 0 (logaritmo negativo de 1, sin pérdida) al infinito (logaritmo negativo de 0, pérdida infinita). Esta función de pérdida también asegura que a medida que se maximiza la probabilidad de la respuesta correcta, se minimiza la probabilidad de la respuesta incorrecta; dado que cualquier aumento en la probabilidad de la respuesta correcta se produce a expensas de la respuesta incorrecta. Se llama pérdida de entropía cruzada, porque la ecuación (2.61) es también la fórmula para la entropía cruzada entre la verdadera distribución de probabilidad $y$, y nuestra distribución estimada $\hat{y}$. [37]

### 2.5.3. Algoritmos de optimización

Ahora, el objetivo es encontrar los pesos óptimos: esto es, minimizar la función de pérdida que hemos definido para el modelo. En la ecuación (2.64) a continuación, representaremos explícitamente el hecho de que la función de pérdida $L$ está parametrizada por los pesos, a los que nos referiremos en el aprendizaje automático en general como $\boldsymbol{\theta}$ (en el caso de la regresión logística $\boldsymbol{\theta} = (\mathbf{w}, b)$). Entonces, debemos encontrar el conjunto de pesos que minimiza la función de pérdida, promediado en todos los ejemplos, ya que la regresión logística optimiza la pérdida para todas las observaciones en las que se entrena, que es lo mismo que optimizar la entropía cruzada promedio en la muestra.

$$\hat{\theta} = \underset{\theta}{\text{argmin}} \frac{1}{m} \sum_{i=1}^{m} L_{CE}(f(\mathbf{x}^{(i)}; \boldsymbol{\theta}),\ y^{(i)}) \tag{2.64}$$

Para actualizar $\boldsymbol{\theta}$, en algún algoritmo de optimización, necesitaremos una definición para el gradiente $\nabla L(f(\mathbf{x}; \boldsymbol{\theta}), y)$. A continuación mostramos la derivación de este gradiente. [37]

En primer lugar tenemos que la derivada de $ln(x)$ es:

$$\frac{d}{dx}ln(x) = \frac{1}{x}$$

Mientras que la derivada del sigmoide es:

$$\frac{d\boldsymbol{\sigma}(z)}{dz} = \boldsymbol{\sigma}(z)(1 - \boldsymbol{\sigma}(z))$$

Supongamos que estamos calculando la derivada de una función compuesta $f(x) = u(v(x))$. Por regla de la cadena, la derivada de $f(x)$ es la derivada de $u(x)$ con respecto a $v(x)$ multiplicada por la derivada de $v(x)$ con respecto a x:

$$\frac{df}{dx} = \frac{du}{dv}\frac{dv}{dx}$$

Primero, queremos conocer la derivada de la función de pérdida con respecto a un solo peso $w_j$ (necesitaremos calcularlo para cada peso y para el sesgo):

$$\begin{aligned}\frac{\partial L_{CE}}{\partial w_j} &= \frac{\partial}{\partial w_j} - [y\, log[\boldsymbol{\sigma}(\mathbf{w}\cdot\mathbf{x}+b)] + (1-y)\, log[1-\boldsymbol{\sigma}(\mathbf{w}\cdot\mathbf{x}+b)]] \\ &= -\left[\frac{\partial}{\partial w_j} y log[\boldsymbol{\sigma}(\mathbf{w}\cdot\mathbf{x}+b)] + \frac{\partial}{\partial w_j}(1-y)\, log[1-\boldsymbol{\sigma}(\mathbf{w}\cdot\mathbf{x}+b)]\right]\end{aligned}$$

A continuación, usando la regla de la cadena y tomando la derivada de log obtenemos:

$$\frac{\partial L_{CE}}{\partial w_j} = -\frac{y}{\boldsymbol{\sigma}(\mathbf{w}\cdot\mathbf{x}+b)}\frac{\partial}{\partial w_j}\boldsymbol{\sigma}(\mathbf{w}\cdot\mathbf{x}+b) - \frac{1-y}{1-\boldsymbol{\sigma}(\mathbf{w}\cdot\mathbf{x}+b)}\frac{\partial}{\partial w_j}1-\boldsymbol{\sigma}(\mathbf{w}\cdot\mathbf{x}+b)$$

Reordenando términos:

$$\frac{\partial L_{CE}}{\partial w_j} = -\left[\frac{y}{\boldsymbol{\sigma}(\mathbf{w}\cdot\mathbf{x}+b)} - \frac{1-y}{1-\boldsymbol{\sigma}(\mathbf{w}\cdot\mathbf{x}+b)}\right]\frac{\partial}{\partial w_j}\boldsymbol{\sigma}(\mathbf{w}\cdot\mathbf{x}+b)$$

Ahora usando la derivada del sigmoide, y usando la regla de la cadena una vez más, finalmente obtenemos:

$$\begin{aligned}\frac{\partial L_{CE}}{\partial w_j} &= -\left[\frac{y-\boldsymbol{\sigma}(\mathbf{w}\cdot\mathbf{x}+b)}{\boldsymbol{\sigma}(\mathbf{w}\cdot\mathbf{x}+b)[1-\boldsymbol{\sigma}(\mathbf{w}\cdot\mathbf{x}+b)]}\right]\boldsymbol{\sigma}(\mathbf{w}\cdot\mathbf{x}+b)[1-\boldsymbol{\sigma}(\mathbf{w}\cdot\mathbf{x}+b)]\frac{\partial(\mathbf{w}\cdot\mathbf{x}+b)}{\partial w_j} \\ &= -\left[\frac{y-\boldsymbol{\sigma}(\mathbf{w}\cdot\mathbf{x}+b)}{\boldsymbol{\sigma}(\mathbf{w}\cdot\mathbf{x}+b)[1-\boldsymbol{\sigma}(\mathbf{w}\cdot\mathbf{x}+b)]}\right]\boldsymbol{\sigma}(\mathbf{w}\cdot\mathbf{x}+b)[1-\boldsymbol{\sigma}(\mathbf{w}\cdot\mathbf{x}+b)]x_j \\ &= -\left[y-\boldsymbol{\sigma}(\mathbf{w}\cdot\mathbf{x}+b)\right]x_j \\ &= \left[\boldsymbol{\sigma}(\mathbf{w}\cdot\mathbf{x}+b)-y\right]x_j\end{aligned}$$

De este modo vemos que:

$$\frac{\partial L_{CE}}{\partial w_j} = [\boldsymbol{\sigma}(\mathbf{w} \cdot \mathbf{x} + b) - y]\, x_j \tag{2.65}$$

El gradiente es entonces definido como un vector de estas derivadas parciales.

Encontrar los pesos $\mathbf{w}$ minimizando la entropía cruzada es equivalente a encontrar los pesos que maximizan la función de verosimilitud evaluando que tan bien está haciendo nuestro modelo de regresión logística al aproximar la verdadera distribución de probabilidad de nuestra variable de Bernoulli. Sin embargo, en el caso más general, una función puede admitir múltiples mínimos y encontrar el mínimo global se considera un problema difícil. No obstante, se puede demostrar que minimizar la entropía cruzada binaria para la regresión logística es un problema convexo y, como tal, cualquier mínimo es global. [38]

En el caso de la técnica de descenso de gradiente, la ecuación final para actualizar $\boldsymbol{\theta}$ en función del gradiente es:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla L(f(\mathbf{x}; \boldsymbol{\theta}), y) \tag{2.66}$$

donde a $\eta$ se le conoce como tasa de aprendizaje (learning rate) o tamaño de paso. Representamos $\hat{y}$ como $f(\mathbf{x}; \boldsymbol{\theta})$ para hacer visible la dependencia de $\theta$.

Siendo la entropía cruzada una función convexa, cualquier técnica de optimización convexa está garantizada para encontrar el mínimo global. Algunas de los métodos de optimización más empleados suelen ser: Gradient Descent, Stochastic Gradient Descent y Newton-Raphon. [37]

En el caso del software Python (software empleado en el presente proyecto), este cuenta con los siguientes algoritmos de optimización: newton-cg, lbfgs, liblinear, sag y saga. En el presente proyecto dejamos el algoritmo de optimización que usa por default la regresión logística al hacer uso de la biblioteca scikit-learn, este es el algoritmo lbfgs. Podemos encontrar más información de cada uno de estos algoritmos de optimización en el repositorio de la biblioteca scikit-learn en [39].

### 2.5.4. Regularización

Hay un problema con los pesos de aprendizaje que hacen que el modelo coincida perfectamente con los datos de entrenamiento. Si una característica predice perfectamente el resultado porque ocurre solo en una clase, se le asignará un peso muy alto. Los pesos de las funciones intentarán encajar perfectamente con los detalles del conjunto de entrenamiento, de hecho demasiado perfectamente, modelando factores ruidosos que se correlacionan accidentalmente con la clase. Este problema se llama sobreajuste. Un

buen modelo debería poder generalizar bien desde los datos de entrenamiento hasta el conjunto de prueba, pero un modelo que se sobreajusta tendrá una generalización deficiente. [37]

Para evitar el sobreajuste, se agrega un nuevo término de regularización $R(\boldsymbol{\theta})$ a la función objetivo en la ecuación (2.64), lo que da como resultado el siguiente objetivo para un lote de $m$ ejemplos (modificamos ligeramente la Ecuación (2.64) para maximizar la probabilidad logarítmica en lugar de minimizar la pérdida, y eliminamos el término $\frac{1}{m}$ que no afecta el argmax):

$$\hat{\boldsymbol{\theta}} = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{m} log \left[ P\left( y^{(i)} | x^{(i)} \right) \right] - \alpha R(\boldsymbol{\theta}) \tag{2.67}$$

El nuevo término de regularización $R(\boldsymbol{\theta})$ se usa para penalizar pesos grandes. Por lo tanto, una configuración de los pesos que coincida perfectamente con los datos de entrenamiento, pero que utilice muchas ponderaciones con valores altos para hacerlo, se penalizará más que una configuración que coincida un poco menos con los datos, pero que lo haga utilizando pesos más pequeños. Hay dos formas comunes de calcular este término de regularización $R(\boldsymbol{\theta})$. La **regularización L2** es una función cuadrática de los valores de peso, nombrada así ya que usa el cuadrado de la norma $L2$ de los valores de peso. La norma $L2$, $\| \boldsymbol{\theta}\|_2$, es la misma que la distancia euclideana del vector $\boldsymbol{\theta}$ al origen. Si $\boldsymbol{\theta}$ consiste de $n$ pesos, entonces:

$$R(\boldsymbol{\theta}) = \| \boldsymbol{\theta}\|_2^{\,2} = \sum_{j=1}^{n} {\theta_j}^2 \tag{2.68}$$

De este modo la función objetivo regularizada con $L2$ se convierte en:

$$\hat{\boldsymbol{\theta}} = \underset{\theta}{\operatorname{argmax}} \left[ \sum_{i=1}^{m} log \left[ P\left( y^{(i)} | \mathbf{x}^{(i)} \right) \right] \right] - \alpha \sum_{j=1}^{n} {\theta_j}^2 \tag{2.69}$$

Por otro lado la **regularización L1** es una función lineal de los valores de peso, llamada así por la norma L1, $\| W \|_1$, la suma de los valores absolutos de los pesos, o la distancia de Manhattan:

$$R(\boldsymbol{\theta}) = \| \theta \|_1 = \sum_{i=1}^{n} |\theta_i| \tag{2.70}$$

De este modo la función objetivo regularizada con $L1$ se convierte en:

$$\hat{\boldsymbol{\theta}} = \underset{\theta}{\operatorname{argmax}} \left[ \sum_{i=1}^{m} log \left[ P\left( y^{(i)} | \mathbf{x}^{(i)} \right) \right] \right] - \alpha \sum_{j=1}^{n} |\theta_j| \tag{2.71}$$

Estos tipos de regularización provienen de las estadísticas, donde la regularización $L1$ se llama **regresión Lasso** y la regularización $L2$ se llama **regresión Ridge**. La

regularización $L2$ es más fácil de optimizar debido a que su derivada es más simple (la derivada de $\theta^2$ es solo $2\theta$), mientras que la regularización $L1$ es más compleja (la derivada de $|\theta|$ no es continua en cero). Pero donde L2 prefiere vectores de peso con muchos pesos pequeños, L1 prefiere soluciones dispersas con algunos pesos más grandes pero muchos más pesos establecidos en cero. Por lo tanto, la regularización de $L1$ conduce a vectores de peso mucho más dispersos, es decir, muchas menos características. [37]

## 2.6. Evaluación del Modelo

Esta sección presenta medidas para evaluar qué tan bueno o qué tan "preciso" es un clasificador para predecir la etiqueta de clase de las tuplas (o instancias). Consideraremos el caso en que las instancias de clase están distribuidas de manera más o menos uniforme, así como el caso en el que las clases están desequilibradas (por ejemplo, cuando una clase de interés importante es rara, como en las pruebas médicas). Las medidas de evaluación del clasificador presentadas en esta sección se resumen en la **Figura 2.6**. Incluyen accuracy (precisión) también conocida como tasa de reconocimiento, sensibilidad (o recall), especificidad, precisión, $F_1$ y $F_\beta$.

| *Measure* | *Formula* |
|---|---|
| accuracy, recognition rate | $\frac{TP+TN}{P+N}$ |
| error rate, misclassification rate | $\frac{FP+FN}{P+N}$ |
| sensitivity, true positive rate, recall | $\frac{TP}{P}$ |
| specificity, true negative rate | $\frac{TN}{N}$ |
| precision | $\frac{TP}{TP+FP}$ |
| $F$, $F_1$, $F$-score, harmonic mean of precision and recall | $\frac{2 \times precision \times recall}{precision + recall}$ |
| $F_\beta$, where $\beta$ is a non-negative real number | $\frac{(1+\beta^2) \times precision \times recall}{\beta^2 \times precision + recall}$ |

**Figura 2.6:** Medidas de evaluación de un clasificador. [40]

Antes de discutir las diversas medidas, definamos la terminología que nos será de utilidad en lo que sigue de la sección. [40]

En machine learning, podemos hablar en términos de instancias positivas (instancias de la clase principal de interés) e instancias negativas (todas las demás instancias). Dadas dos clases, por ejemplo, las instancias positivas pueden ser "Diabetes = si" mientras

que las instancias negativas son "Diabetes = no". Supongamos que usamos un conjunto de prueba de instancias etiquetadas. $P$ es el número de instancias positivas y $N$ es el número de instancias negativas. Para cada instancia, comparamos la predicción de la etiqueta de clase del clasificador con la etiqueta de clase conocida de las instancias.

Hay cuatro términos adicionales que necesitamos conocer, que son los "bloques de construcción" que se utilizan en el cálculo de muchas medidas de evaluación. Comprenderlos facilitará la comprensión del significado de las distintas medidas:

**Verdaderos Positivos ($TP$)**: se refieren a las instancias positivas que fueron etiquetadas correctamente por el clasificador. Descimos que $TP$ es el número de verdaderos positivos.

**Verdaderos Negativos ($TN$)**: son las instancias negativas que el clasificador etiquetó correctamente. Decimos que $TN$ es el número de verdaderos negativos.

**Falsos positivos ($FP$)**: son las instancias negativas que se etiquetaron incorrectamente como positivas. Decimos que $FP$ es el número de falsos positivos.

**Falsos negativos ($FN$)**: Son las instancias positivas que se etiquetaron erróneamente como negativas. Decimos que $FN$ es el número de falsos negativos.

Ahora analicemos algunas de las métricas de evaluación más utilizadas en machine learning.

### 2.6.1. Métricas de evaluación: Matriz de confusión, Accuracy, Error rate, sensitivity, specificity, precision, recall y F-score

**Matriz de confusión:**

La matriz de confusión (**Figura 2.7**) es una herramienta útil para analizar qué tan bien un clasificador puede reconocer instancias de diferentes clases. $TP$ y $TN$ nos dicen cuándo el clasificador está haciendo las cosas bien, mientras que $FP$ y $FN$ nos dicen cuándo el clasificador está haciendo las cosas mal (es decir, etiquetando incorrectamente). Dadas $m$ clases (donde $m \geq 2$), una matriz de confusión es una tabla de tamaño $m \times m$. Una entrada, $CM_{ij}$, en las primeras $m$ filas y $m$ columnas indican el número de instancias de clase $i$ que fueron etiquetadas por el clasificador como clase $j$. Para que un clasificador tenga una buena precisión, idealmente la mayoría de las instancias estarían representadas a lo largo de la diagonal de la matriz de confusión, desde la entrada $CM_{1,1}$ hasta la entrada $CM_{m,m}$, siendo el resto de las entradas cero o cercanas a cero. Es decir, idealmente, $FP$ y $FN$ estarían alrededor de cero.

| | | Predicted class | | |
|---|---|---|---|---|
| | | *yes* | *no* | Total |
| Actual class | *yes* | *TP* | *FN* | *P* |
| | *no* | *FP* | *TN* | *N* |
| | Total | $P'$ | $N'$ | $P+N$ |

**Figura 2.7:** Matriz de confusión. [40]

La tabla puede tener filas o columnas adicionales para proporcionar totales. Por ejemplo, en la matriz de confusión de la **Figura 2.7**, se muestran $P$ y $N$. Además, $P'$ es el número de instancias que se etiquetaron como positivas ($TP + FP$) y $N'$ es el número de instancias que se etiquetaron como negativas ($FN + TN$). El número total de instancias es $TP+TN+FP+TN$, o $P+N$, o bien $P'+N'$. Tengamos en cuenta que aunque la matriz de confusión que se muestra es para un problema de clasificación binaria, las matrices de confusión se pueden realizar para múltiples clases de manera similar.

Ahora veamos algunas de las medidas de evaluación más empleadas en machine learning. [40]

**Accuracy (Precisión):**

La precisión (accuracy) de un clasificador en un conjunto de prueba dado, es el porcentaje de instancias del conjunto de prueba que el clasificador clasifica correctamente. Es decir:

$$accuracy = \frac{TP + TN}{P + N} \tag{2.72}$$

En la literatura de reconocimiento de patrones, esto también se conoce como la tasa de reconocimiento general (recognition rate) del clasificador, es decir, refleja qué tan bien reconoce el clasificador las instancias de las diversas clases.

**Error rate:**

También podemos hablar de la tasa de error (error rate) o tasa de clasificación errónea (misclassification rate) de un clasificador $M$, el cual es simplemente $1-accuracy(M)$, donde accuracy($M$) es la precisión de $M$. Esto también se puede calcular como:

$$error\ \ rate = \frac{FP + FN}{P + N} \tag{2.73}$$

Si usáramos el conjunto de entrenamiento (en lugar de un conjunto de prueba) para estimar la tasa de error de un modelo, esta cantidad se conoce como error de resustitución. Esta estimación de error es optimista de la tasa de error real (y de manera

similar, la estimación de precisión correspondiente es optimista) porque el modelo no se prueba en ninguna muestra que no haya visto.

**Sensitivity y specificity:**

Ahora consideraremos el problema del desequilibrio de clases, donde la clase principal de interés es rara. Es decir, la distribución del conjunto de datos refleja una mayoría significativa de la clase negativa y una minoría de la clase positiva.

En los datos médicos, es frecuente una clase poco común, como "diabetes". Supongamos que se ha entrenado a un clasificador para clasificar instancias de datos médicos, donde el atributo de etiqueta de clase es "diabetes" y los posibles valores de clase son "sí" y "no". Una tasa de precisión de, digamos, 97 % puede hacer que el clasificador parezca bastante preciso, pero si el 3 % de las instancias de entrenamiento son en realidad diabetes, entonces claramente una tasa de precisión del 97 % puede no ser aceptable; el clasificador podría etiquetar correctamente solo las instancias no diabéticas, por ejemplo, y clasificar erróneamente todas las instancias diabéticas. En cambio, necesitamos otras medidas, que accedan a qué tan bien el clasificador puede reconocer las instancias positivas (Diabetes = sí) y qué tan bien puede reconocer las instancias negativas (Diabetes = no). Veamos pues algunas de estas métricas.

Las medidas de sensibilidad (sensitivity) y especificidad (specificity), se pueden utilizar, respectivamente, para este propósito. La sensibilidad también se conoce como true positivite rate (tasa verdadera positiva), es decir, la proporción de instancias positivas que se identifican correctamente, mientras que la especificidad es la true negative rate (tasa negativa verdadera), esto es, la proporción de instancias negativas que se identifican correctamente. Estas medidas se definen como:

$$sensitivity = \frac{TP}{P} \tag{2.74}$$

$$specificity = \frac{TN}{N} \tag{2.75}$$

Por otro lado, se puede demostrar que la precisión (accuracy) es función de la sensibilidad y la especificidad:

$$accuracy = sensitivity\frac{P}{(P+N)} + specificity\frac{N}{(P+N)} \tag{2.76}$$

**Precision y recall:**

Las medidas de precisión (precision) y exhaustividad (recall) también se utilizan ampliamente en la clasificación. Precision, se puede considerar como una medida de exactitud (es decir, qué porcentaje de instancias etiquetadas como positivas son realmente tales), mientras que recall es una medida de integridad (qué porcentaje de ins-

tancias positivas están etiquetadas como tales). Recall realmente es lo mismo que la sensibilidad (o la verdadera tasa positiva). Estas medidas se pueden calcular como:

$$precision = \frac{TP}{TP + FP} \tag{2.77}$$

$$recall = \frac{TP}{TP + FN} = \frac{TP}{P} \tag{2.78}$$

Una puntuación de precision perfecta de 1.0 para una clase $C$ significa que cada instancia que el clasificador etiquetó como perteneciente a la clase $C$ pertenece de hecho a la clase $C$. Sin embargo, no nos dice nada sobre el número de instancias de clase $C$ que el clasificador etiquetó incorrectamente. Una puntuación de recall perfecto de 1.0 para $C$ significa que todos los elementos de la clase $C$ fueron etiquetados como tales, pero no nos dice cuántas otras instancias fueron etiquetadas incorrectamente como pertenecientes a la clase $C$. Tiende a haber una relación inversa entre precision y recall, donde es posible aumentar uno a costa de reducir el otro. Por ejemplo, nuestro clasificador médico puede lograr un alto precision al etiquetar todas las instancias de diabetes que se presentan de cierta manera como diabetes, pero puede tener poco recall si etiqueta erróneamente muchas otras instancias de instancias de diabetes. Las puntuaciones de precision y recall se suelen utilizar juntas, donde los valores de precisión se comparan para un valor fijo del recall, o viceversa.

**F-score:**

Una forma alternativa de utilizar precision y recall es combinarlos en una sola medida. Este es el enfoque de la medida $F$ (también conocida como puntuación $F_1$ o $F - score$) y la medida $F_\beta$. Estas se definen como:

$$F = \frac{2 \times precision \times recall}{precision + recall} \tag{2.79}$$

$$F_\beta = \frac{\left(1 + \beta^2\right) \times precision \times recall}{\beta^2 \times precision + recall} \tag{2.80}$$

donde $\beta$ es un número real no-negativo. La medida $F$ es la media armónica de precision y recall. Da igual importancia a precision y a recall. Por otro lado la medida $F_\beta$ es una medida ponderada de precision y recall. Asigna $\beta$ veces más peso para recall que para precision. Las medidas de $F_\beta$ comúnmente utilizadas son $F_2$ (cuyos pesos en recall son el doble que en precision) y $F_{0.5}$ (que pondera precision dos veces más que recall).

En resumen, hemos presentado varias medidas de evaluación. La medida accuracy funciona mejor cuando las clases de datos están distribuidas de manera bastante uniforme. Otras medidas, como la sensibilidad (o el recall), la especificidad, la precisión, $F$ y $F_\beta$, se adaptan mejor al problema de desequilibrio de clases, donde la clase principal de interés es rara. [40]

### 2.6.2. Curva ROC

Los verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos también son útiles para evaluar los costos y beneficios (o riesgos y ganancias) asociados con un modelo de clasificación. El costo asociado con un falso negativo (como predecir incorrectamente que un paciente diabético no es diabético) es mucho mayor que el de un falso positivo (etiquetar de manera incorrecta pero conservadora a un paciente no diabético como diabético). En tales casos, podemos compensar un tipo de error sobre otro asignando un costo diferente a cada uno. Estos costos pueden considerar el peligro para el paciente, los costos financieros de las terapias resultantes y otros costos hospitalarios. De manera similar, los beneficios asociados con una verdadera decisión positiva pueden ser diferentes a los de una verdadera negativa. Hasta ahora, para calcular la precisión del clasificador, hemos asumido costos iguales y esencialmente dividimos la suma de verdaderos positivos y verdaderos negativos por el número total de instancias de prueba.

Las curvas de características operativas del receptor (Receiver operating characteristic curves), son una herramienta visual útil para comparar dos modelos de clasificación. Las curvas ROC provienen de la teoría de detección de señales que se desarrolló durante la Segunda Guerra Mundial para el análisis de imágenes de radar. Una curva ROC para un modelo dado, muestra la compensación entre la tasa de verdaderos positivos (TPR) y la tasa de falsos positivos (FPR). Dado un conjunto de pruebas y un modelo, TPR es la proporción de instancias positivas (o "sí") que están correctamente etiquetadas por el modelo; FPR es la proporción de instancias negativas (o "no") que están mal etiquetadas como positivas. Recordemos que TP, FP, P y N son; el número de instancias verdaderos positivos, falsos positivos, positivos y negativos, respectivamente, y $TPR = \frac{TP}{P}$ es la sensibilidad. Además, $FPR = \frac{FP}{N}$, lo cual es equivalente a $1 - specificity$.

Para un problema de dos clases, una curva ROC nos permite visualizar la compensación entre la tasa a la que el modelo puede reconocer con precisión los casos positivos y la tasa a la que identifica erróneamente los casos negativos como positivos para diferentes partes del conjunto de prueba. Cualquier aumento de TPR se produce a costa de un aumento de FPR. El área bajo la curva ROC es una medida de la precisión del modelo.

Para trazar una curva ROC, para un modelo de clasificación dado, $M$, el modelo debe poder devolver una probabilidad de la clase predicha para cada instancia de prueba. Con esta información, clasificamos y ordenamos las instancias de modo que la instancia que tenga más probabilidades de pertenecer a la clase positiva o "sí" aparezca en la parte superior de la lista, y la instancia que tenga menos probabilidades de pertenecer a la clase positiva aparezca al final de la lista. Para un problema binario, normalmente se selecciona un umbral t de modo que las instancias donde $f(X) \geq t$ se consideren

positivas y todas las demás instancias se consideren negativas. Tengamos en cuenta que el número de verdaderos positivos y el número de falsos positivos son funciones de $t$, por lo que podríamos escribir $TP(t)$ y $FP(t)$. Ambas son funciones descendentes monótonas.

Primero describimos la idea general detrás de trazar una curva ROC y luego seguimos con un ejemplo. El eje vertical de una curva ROC representa $TPR$. El eje horizontal representa $FPR$. Para trazar una curva ROC para un modelo $M$, se trabaja de la siguiente manera, primero, en la esquina inferior izquierda (donde $TPR = FPR = 0$), verificamos la etiqueta de clase real de la instancia en la parte superior de la lista. Si tenemos un verdadero positivo (es decir, una instancia positiva que se clasificó correctamente), entonces $TP$ y, por lo tanto, $TPR$ aumentan. En el gráfico, nos movemos hacia arriba y trazamos un punto. Si, en cambio, el modelo clasifica una instancia negativa como positiva, tenemos un falso positivo y, por lo tanto, tanto $FP$ como $FPR$ aumentan. En el gráfico, nos movemos hacia la derecha y trazamos un punto. Este proceso se repite para cada una de las instancias de prueba en orden de clasificación, cada vez que se mueve hacia arriba en el gráfico para un verdadero positivo o hacia la derecha para un falso positivo. Veamos esto con más claridad dando un ejemplo. [40]

| Tuple # | Class | Prob. | TP | FP | TN | FN | TPR | FPR |
|---|---|---|---|---|---|---|---|---|
| 1 | *P* | 0.90 | 1 | 0 | 5 | 4 | 0.2 | 0 |
| 2 | *P* | 0.80 | 2 | 0 | 5 | 3 | 0.4 | 0 |
| 3 | *N* | 0.70 | 2 | 1 | 4 | 3 | 0.4 | 0.2 |
| 4 | *P* | 0.60 | 3 | 1 | 4 | 2 | 0.6 | 0.2 |
| 5 | *P* | 0.55 | 4 | 1 | 4 | 1 | 0.8 | 0.2 |
| 6 | *N* | 0.54 | 4 | 2 | 3 | 1 | 0.8 | 0.4 |
| 7 | *N* | 0.53 | 4 | 3 | 2 | 1 | 0.8 | 0.6 |
| 8 | *N* | 0.51 | 4 | 4 | 1 | 1 | 0.8 | 0.8 |
| 9 | *P* | 0.50 | 5 | 4 | 0 | 1 | 1.0 | 0.8 |
| 10 | *N* | 0.40 | 5 | 5 | 0 | 0 | 1.0 | 1.0 |

**Tabla 2.** Instancias ordenadas por score decreciente, donde el score es el valor devuelto por un clasificador probabilístico. [40]

La **Tabla 2** muestra el valor de probabilidad (columna 3) devuelto por un clasificador probabilístico para cada una de las 10 instancias en un conjunto de prueba, ordenadas por orden de probabilidad decreciente. La columna 1 es simplemente un número de identificación de la instancia (o tupla). La columna 2 es la etiqueta de clase real de la instancia. Hay cinco instancias positivas y cinco instancias negativas, por lo que $P = 5$ y $N = 5$. A medida que examinamos la etiqueta de clase conocida de cada instancia, podemos determinar los valores de las columnas restantes, $TP$, $FP$, $TN$,

$FN$, $TPR$, y $FPR$. Comenzamos con la instancia 1, que tiene la puntuación de probabilidad más alta, y tomamos esa puntuación como nuestro umbral, es decir, $t = 0.9$. Por lo tanto, el clasificador considera que la instancia 1 es positiva y todas las demás instancias se consideran negativas. Dado que la etiqueta de clase real de la instancia 1 es positiva, tenemos un verdadero positivo, por lo tanto, $TP = 1$ y $FP = 0$. Entre las nueve instancias restantes, que están todas clasificadas como negativas, cinco en realidad son negativas (por lo tanto, $TN = 5$) . Los cuatro restantes son todos realmente positivos, por lo tanto, $FN = 4$. Por lo tanto, podemos calcular $TPR = \frac{TP}{P} = \frac{1}{5} = 0.2$, mientras que $FPR = 0$. De este modo, tenemos el punto $(0.2, 0)$ para la curva ROC.

A continuación, el umbral $t$ se establece en 0.8, el valor de probabilidad para la instancia 2, por lo que esta instancia ahora también se considera positiva, mientras que las instancias 3 a 10 se consideran negativas. La etiqueta de clase real de la instancia 2 es positiva, por lo que ahora $TP = 2$. El resto de la fila se puede calcular de forma análoga al caso anterior, lo que da como resultado el punto $(0.4, 0)$. A continuación, examinamos la etiqueta de clase de la instancia 3, hacemos $t = 0.7$, por lo tanto, la instancia 3 se considera positiva, pero su etiqueta real es negativa y, por lo tanto, es un falso positivo. Por lo tanto, $TP$ permanece igual y $FP$ se incrementa de modo que $FP = 1$. Calculando el resto de los valores en la fila obtenemos el punto $(0.4, 0.2)$. El gráfico ROC resultante, de examinar cada instancia, es la línea dentada que se muestra en la **Figura 2.8**.

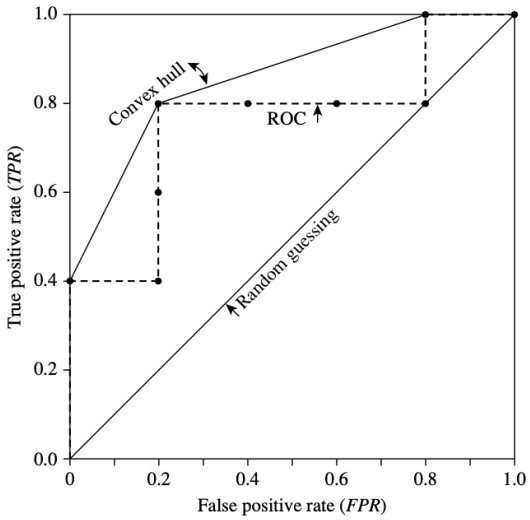



**Figura 2.8:** Curva ROC para los datos dados en la **Tabla 2**. [40]

Existen muchos métodos para obtener una curva a partir de estos puntos, el más común de los cuales es utilizar un convex hull (envolvente convexa). El gráfico también muestra una línea diagonal en la que por cada verdadero positivo de dicho modelo, es muy probable que encontremos un falso positivo. A modo de comparación, esta línea

representa una adivinación aleatoria. [40]

Hay varios puntos críticos a lo largo de una curva ROC que tienen interpretaciones bien conocidas:

- $(TPR = 0, FPR = 0)$: El modelo predice que cada instancia será una clase negativa.
- $(TPR = 1, FPR = 1)$: El modelo predice que cada instancia será una clase positiva.
- $(TPR = 1, FPR = 0)$: El modelo ideal.

Un buen modelo de clasificación debe ubicarse lo más cerca posible de la esquina superior izquierda del diagrama, mientras que un modelo que hace conjeturas aleatorias debe residir a lo largo de la diagonal principal, conectando los puntos (TPR = 0, FPR = 0) y ( TPR = 1, FPR = 1). Random guessing significa que un registro se clasifica como una clase positiva con una probabilidad fija $p$, independientemente de su conjunto de atributos. Por ejemplo, consideremos un conjunto de datos que contiene $n_+$ instancias positivas y $n_-$ instancias negativas. Se espera que el clasificador aleatorio clasifique correctamente $pn_+$ de las instancias positivas y clasifique erróneamente $pn_-$ de las instancias negativas. Por lo tanto, el $TPR$ del clasificador es $(pn_+)/n_+ = p$, mientras que su $FPR$ es $(pn_-)/p = p$. Dado que $TPR$ y $FPR$ son idénticos, la curva ROC para un clasificador aleatorio siempre reside a lo largo de la diagonal principal. [27]

La **Figura 2.9** muestra las curvas ROC de dos modelos de clasificación. También se muestra la línea diagonal que representa el random guessing. Por lo tanto, cuanto más cerca esté la curva ROC de un modelo de la línea diagonal, menos preciso será el modelo. Si el modelo es realmente bueno, inicialmente es más probable que encontremos verdaderos positivos a medida que avanzamos en la lista de clasificación. Por tanto, la curva sube abruptamente desde cero. Más tarde, a medida que comenzamos a encontrar cada vez menos verdaderos positivos y más y más falsos positivos, la curva se suaviza y se vuelve más horizontal. Para evaluar la precisión de un modelo, podemos medir el área bajo la curva.

El **área bajo la curva ROC (AUC)**, proporciona otro enfoque para evaluar qué modelo es mejor en promedio. AUC proporciona una medida agregada de rendimiento en todos los umbrales de clasificación posibles. Una forma de interpretar AUC es como la probabilidad de que el modelo clasifique un ejemplo positivo aleatorio más alto que un ejemplo negativo aleatorio. Si el modelo es perfecto, entonces su área bajo la curva ROC sería igual a 1. Si el modelo simplemente realiza adivinaciones al azar, entonces su área bajo la curva ROC sería igual a 0.5. Un modelo que es estrictamente mejor que otro tendría un área más grande bajo la curva ROC. [40]

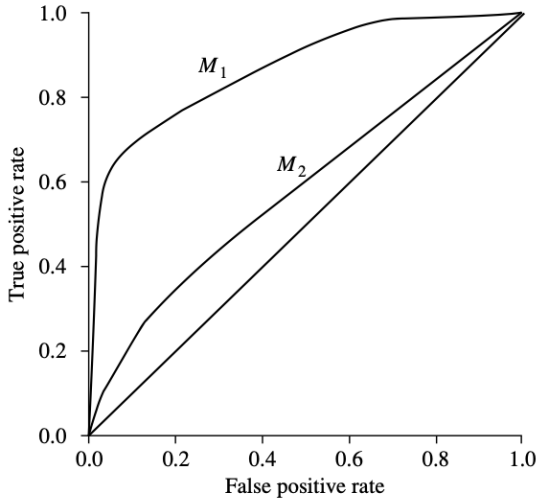



**Figura 2.9:** Curvas ROC de dos modelos de clasificación, $M_1$ y $M_2$. La diagonal muestra dónde, por cada verdadero positivo, es igualmente probable que encontremos un falso positivo. Cuanto más cerca esté una curva ROC de la línea diagonal, menos preciso será el modelo. Por tanto, M1 es más preciso aquí. [40]

### 2.6.3. Validación cruzada

En el método de retención (**Holdout Method**), los datos proporcionados se dividen aleatoriamente en dos conjuntos independientes, un conjunto de entrenamiento y un conjunto de prueba. Normalmente, dos tercios de los datos se asignan al conjunto de entrenamiento y el tercio restante se asigna al conjunto de prueba. El conjunto de entrenamiento se utiliza para obtener el modelo, luego, se estima la precisión del modelo con el conjunto de prueba (**Figura 2.10**).

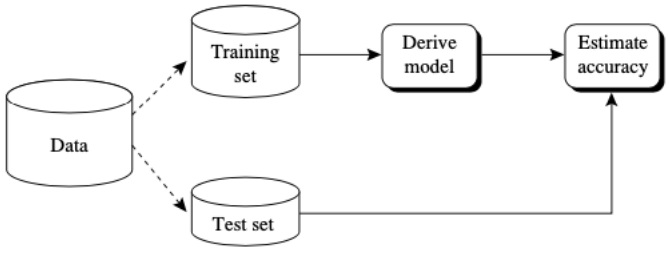



**Figura 2.10:** Estimación de la precisión con el Holdout Method. [40]

Por otro lado, el submuestreo aleatorio (**Random Subsampling**) es una variación

del método de retención en el que el método de retención se repite $k$ veces. La estimación de la precisión general se toma como el promedio de las precisiones obtenidas de cada iteración.

En **$k$-fold cross validation** (validación cruzada $k$ veces), los datos iniciales se dividen aleatoriamente en $k$ subconjuntos mutuamente excluyentes o "folds" (pliegues), $D_1, D_2, ..., D_k$, cada uno de aproximadamente el mismo tamaño, entonces el entrenamiento y las pruebas se realizan $k$ veces. En la iteración $i$, la partición $D_i$ se reserva como el conjunto de prueba y las particiones restantes se usan colectivamente para entrenar el modelo. Es decir, en la primera iteración, los subconjuntos $D_2, ..., D_k$ sirven colectivamente como el conjunto de entrenamiento para obtener un primer modelo, que se prueba en $D_1$; la segunda iteración se entrena en los subconjuntos $D_1, D_3, ..., D_k$ y se prueba en $D_2$; etcétera, podemos observar el diagrama de k-fold cross-validation en la **Figura 2.11**.

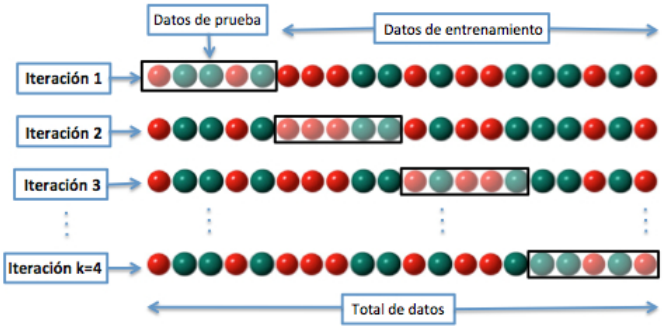



**Figura 2.11:** Validación cruzada de $k$ iteraciones con $k = 4$. [41]

A diferencia de los métodos de retención y submuestreo aleatorio, aquí cada muestra se usa la misma cantidad de veces para entrenamiento y una vez para prueba. Para la clasificación, la estimación de precisión es el número total de clasificaciones correctas de las $k$ iteraciones, dividido por el número total de instancias en los datos iniciales. [40]

En nuestro proyecto, debido al desbalance de datos, emplearemos un método estrechamente relacionado a la validación cruzada, este método es conocido como **stratified cross-validation** (validación cruzada estratificada), esta técnica es un tipo de validación cruzada en la que los pliegues se estratifican de modo que la distribución de clases de las instancias en cada pliegue sea aproximadamente la misma que en los datos iniciales. Esto garantiza que ningún valor esté sobre-representado o sub-representado en los conjuntos de entrenamiento y prueba, lo que brinda una estimación más precisa del rendimiento del modelo.

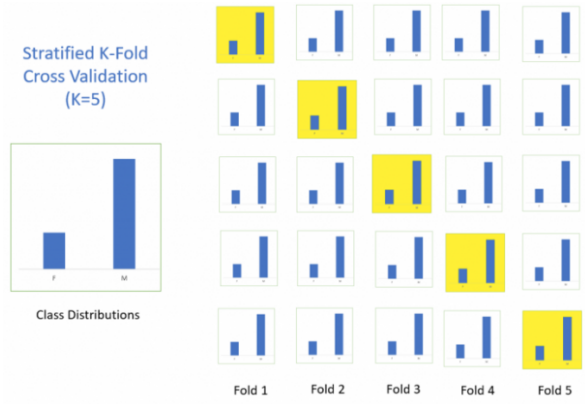



**Figura 2.12:** Stratified cross-validation con $k = 5$, se puede observar como en cada pliegue la distribución de clase se conserva. [42]

Podemos observar este tipo de validación cruzada en la **Figura 2.12**. Una validación cruzada $k$ veces, puede fallar fácilmente en el caso de desequilibrios de clase. La razón es que los datos se dividen en $k$ pliegues con una distribución de probabilidad uniforme. Esto podría funcionar bien para datos con una distribución de clase equilibrada, pero cuando la distribución está muy sesgada, es probable que uno o más pliegues tengan pocos o ningún ejemplo de la clase minoritaria. Esto, significa que algunas o quizás muchas de las evaluaciones del modelo serán engañosas, ya que el modelo solo necesita predecir correctamente la clase mayoritaria.

En general, se recomienda la validación cruzada estratificada 10 veces para estimar la precisión (incluso si el poder de cálculo permite usar más pliegues) debido a su sesgo y varianza relativamente bajos. [40]

Existen varios beneficios al usar técnicas de tipo validación cruzada en lugar de una sola división en un conjunto de entrenamiento y prueba. Primero, el método de sólo dividir en prueba y test realiza una división aleatoria de los datos. Imaginemos que tenemos "suerte" cuando dividimos los datos al azar, y todos los ejemplos que son difíciles de clasificar terminan en el conjunto de entrenamiento. En ese caso, el conjunto de prueba solo contendrá ejemplos "fáciles" y la precisión de nuestro conjunto de prueba será irrealmente alta. Por el contrario, si tenemos "mala suerte", es posible que hayamos colocado al azar todos los ejemplos difíciles de clasificar en el conjunto de pruebas y, en consecuencia, obtengamos una puntuación irrealmente baja. Sin embargo, al usar la validación cruzada (o variaciones), cada ejemplo estará en el conjunto de entrenamiento exactamente una vez: cada ejemplo está en uno de los pliegues y cada pliegue es el conjunto de prueba una vez. Por lo tanto, el modelo debe generalizarse bien a todas las muestras del conjunto de datos para que todas las puntuaciones de validación cruzada (y su media) sean altas.

La principal desventaja de la validación cruzada es un mayor costo computacional. Como ahora estamos entrenando $k$ modelos en lugar de un solo modelo, la validación cruzada será aproximadamente $k$ veces más lenta que hacer una sola división de los datos.

Es importante tener en cuenta que la validación cruzada no es una forma de construir un modelo que se pueda aplicar a nuevos datos. La validación cruzada no devuelve un modelo. Al llamar a cross validation, se construyen varios modelos internamente, pero el propósito de la validación cruzada es solo evaluar qué tan bien se generalizará un algoritmo dado cuando se entrene en un conjunto de datos específico. [44]

Capítulo 3

# Diseño del Modelo de Predicción de Diabetes en base al clasificador de Regresión Logística

En esta sección, se construirán modelos predictivos enfocados en la predicción de diabetes. El modelado predictivo utiliza la extracción de datos, el aprendizaje automático y las estadísticas para identificar patrones en los datos y reconocer las posibilidades de que se produzcan los resultados. Como principal algoritmo de clasificación se hará uso de la regresión logística a través del lenguaje de programación Python.

La regresión logística, es un modelo de clasificación en aprendizaje automático, ampliamente utilizado en análisis clínicos. Utiliza estimaciones probabilísticas que ayudan a comprender la relación entre la variable dependiente y una o más variables independientes (ver **sección 2.5.1**). La diabetes, siendo una de las enfermedades más comunes en todo el mundo, cuando se detecta a tiempo, puede prevenir la progresión de la enfermedad y evitar otras complicaciones. En este trabajo, diseñamos un modelo de predicción que predice si un paciente tiene diabetes, en función de ciertas medidas de diagnóstico incluidas en el conjunto de datos, y exploramos varias técnicas para mejorar el rendimiento y la precisión. Este capítulo explora diversos trabajos relacionados, da la descripción de los conjuntos de datos utilizados y explica el proceso de construcción y evaluación de los modelos diseñados.

## 3.1. Introducción

Varios investigadores han utilizado el aprendizaje automático (ML) para predecir la diabetes utilizando el conjunto de datos PID (Pima Indians Diabetes). En esta sección se analizan algunos trabajos relacionados.

En [45] se realiza una recopilacion de 20 diferentes estudios relacionados con la predicción temprana de la diabetes. Para una comparación justa, todos los estudios previos los seleccionaron utilizando el mismo conjunto de datos PID. En la **Tabla 3** se muestran 10 de los 20 estudios recopilados, estos estudios son: Guldogan et al [46], Alam et al [47], Wang et al [48], Woldemichael and Menaria [49], Wu et al [50], Vaishali et al [51], AlJarullah [52], Marcano-Cedeño et al [53], Patil et al [54] y Han et al [55].

| Estudio | Enfoque de preprocesamiento | Algoritmo | Accuracy |
|---|---|---|---|
| Guldogan et al. | Missing data deleted | Multilayer perceptron (MLP) and radial basis function (RBF) | MLP giving an accuracy of 78.1% |
| Alam et al. | Median value imputation | ANN, random forest, K-means clustering | ANN giving an accuracy of 75.7% |
| Wang et al. | naïve Bayes imputation, ADASYN oversampling | Random forest | 87.10% accuracy |
| Woldemichael and Menaria. | Mean value imputation | Backpropagation, support vector machine, J48, naïve Bayes | Backpropagation giving an accuracy of 83.11% |
| Wu et al. | Mean value imputation | K-means + logistic regression | 95.42% accuracy |
| Vaishali et al. | Missing data kept unchanged | Naïve Bayes, J48, MLP, MOE fuzzy classifier | MOE fuzzy giving accuracy of 83.04% |
| AlJarullah . | Missing data deleted | J48 | 78.17% accuracy |
| Marcano-Cedeño et al. | Missing data deleted | Artificial metaplasticity on multilayer perceptron algorithm (AMMLP) | 89.93% accuracy |
| Patil et al. | Missing data deleted | K-means + C4.5 | 92.38% accuracy |
| Han et al. | Missing data deleted | ID3, decision tree | ID3 giving an accuracy of 80.0% |

**Tabla 3.** Estudios de aprendizaje automático para la clasificación de la diabetes en el conjunto de datos PID.

Los autores en [45] informan que la Red Neural Artificial (ANN) fue el clasificador más popular donde 11 de 20 estudios utilizaron diferentes tipos de ANN en sus experimentos. Más de la mitad de los estudios dan como resultado una precisión de predicción dentro del rango del 80 % al 90 %. De igual forma reportan que dentro de los 20 estudios analizados Wu et al [50], el cual utilizó un modelo predictivo híbrido, obtuvo una precisión del 95.42 % en la predicción, la más alta de los estudios revisados.

De igual forma en [45] los autores realizan una investigación que propone desarrollar un modelo predictivo que pueda lograr una alta precisión de clasificación de la diabetes tipo 2, para esto se valen del conjunto de datos PID. El estudio consta de dos partes fundamentales. En primer lugar, los autores investigan el manejo de los datos faltantes adoptando la imputación de datos, en este apartado utilizan; la imputación de la mediana por grupos (median value imputation), la imputación del vecino más cercano (K-nearest neighbor) y la imputación iterativa. En consecuencia, el estudio validó las implicaciones de estas imputaciones utilizando varios algoritmos de clasificación (algoritmos lineales, basados en árboles y conjuntos) para ver cómo cada método afectó la precisión de la clasificación. En segundo lugar, emplean una red neuronal artificial (ANN por sus siglas en inglés) para modelar los datos imputados de mejor rendimiento, equilibrando los datos usando SMOTETomek, lo que garantiza que cada clase esté representada de manera justa. Para la primera fase (análisis de las técnicas de imputación) los autores realizan en primer lugar la imputación, en el siguiente paso de preprocesamiento manejan los valores atípicos en los tres conjuntos de datos imputados estableciendo un umbral para decidir qué puntos de datos constituyen valores atípicos, finalmente los autores proceden a utilizar varias métricas de evaluación para evaluar el rendimiento de cada técnica de imputación aplicada a través de una validación cruzada de 10 veces para siete clasificadores de aprendizaje automático. En este apartado concluyen que la imputación del Método I (imputación de la mediana por grupos) fue la que obtuvo el mejor rendimiento. Para validar este hecho, observan también las métricas de evaluación: F1 score, Precision y Recall, considerando un promedio ponderado (weighted average) para la puntuación.

Varios otros estudios han demostrado que la regresión logística funciona tan bien como las técnicas de aprendizaje automático para la predicción del riesgo de enfermedades ( [56], [57], por ejemplo).

En [58] los autores utilizaron varios algoritmos de aprendizaje automático, tales como Support Vector Machines, Random Forest Classifier, Decision Tree Classifier, Extra Tree Classifier, Ada Boost algorithm, Perceptron, Linear Discriminant Analysis algorithm, Logistic Regression, K-NN, Gaussian Naïve Bayes, Bagging algorithm y Gradient Boost Classifier. Utilizaron dos conjuntos de datos diferentes: el PID y otro conjunto de datos de Diabetes para probar los diversos modelos. En el caso del conjunto de datos PID, la regresión logística les dio un valor de precisión del 76 % mientras que en el otro conjunto de datos reportan que la regresión logística les dio la más alta precisión con un 96 %. En este trabajo se imputaron los valores perdidos sin embargo no mencionan el método de imputación.

Analicemos más profundamente los dos estudios que en la **Tabla 3** reportan las precisiones más altas, estos son Patil et al [54] y Wu et al [50].

En Patil et al [54] se propone un enfoque híbrido al combinar el algoritmo de agrupamiento de K-means y C4.5 para clasificar el conjunto de datos de diabetes de los indios Pima (PID). Su sistema propuesto tiene tres pasos. En primer lugar, los datos se han preprocesado mediante la eliminación de datos inapropiados e inconsistentes. Debido a los valores 0 asociados en el conjunto de datos PID, los investigadores eliminaron dos características, serum-insulin y triceps skin fold, además eliminaron 143 instancias del conjunto de datos. Después del preprocesamiento, el conjunto de datos PID se reduce de 768 a 625 instancias y de 8 a 6 características. Se aplicó el método de z-score para normalizar el conjunto PID reducido. En segundo lugar, los patrones se extrajeron utilizando el algoritmo de agrupamiento en clústeres K-means, a través de K-means eliminaron los patrones agrupados incorrectamente; por lo tanto, el conjunto de datos se redujo a 433 instancias. En tercer lugar, se ha construido un modelo de árbol de decisión (C4.5) utilizando los patrones extraídos. Han logrado una precisión de clasificación del 92.38 % utilizando una validación cruzada de diez veces (10-fold cross validation).

Por otro lado, en Wu et al [50] el modelo se compone de dos partes, el algoritmo mejorado de K-means y la regresión logística. En el paso de preprocesamiento los autores determinan que el número de embarazos tiene poca relación con la Diabetes Mellitus y transforman este atributo numérico en un atributo nominal. El valor 0 indica no embarazada y 1 indica embarazada. En segundo lugar, los datos faltantes fueron imputados usando las medias de los datos de entrenamiento para reemplazar todos los valores faltantes. A continuación utilizaron el método de z-score para normalizar el conjunto PID imputado. Finalmente, el modelo consta de algoritmos de doble nivel. En el primer nivel, usan un algoritmo K-means mejorado para eliminar los datos agrupados incorrectamente, a continuación este conjunto de datos optimizado lo usan como entrada para el siguiente nivel, en el cual utilizan el algoritmo de regresión logística para clasificar los datos restantes. Después del procedimiento de extracción usando k-means, obtienen 589 pacientes correctamente clasificados, que sirvieron como entrada para el algoritmo de regresión logística. Finalmente como método de validación reportan haber usado k-fold cross validation, con una precisión (accuracy) de hasta el 95.42 %.

Otro trabajo más reciente con métodos similares a los dos anteriores, se encuentra en Zhu et al [6]. Aquí se han utilizado PCA y regresión logística junto con el agrupamiento K-means. En el preprocesamiento deciden aplicar la misma técnica utilizada en [50], al transformar el atributo numérico embarazos en un atributo nominal de valor 0 y 1. A continuación los valores faltantes en cada atributo se imputan con la media de dicho atributo y utilizaron el método de z-score para normalizar el conjunto PID imputado. Finalmente, el algoritmo modelo está compuesto por 3 etapas. En la primera etapa del diseño, realizan una reducción de dimensionalidad en el conjunto de datos ya procesado (usando PCA). Luego, agrupan el conjunto de datos reducido utilizando K-means para abordar los valores atípicos y eliminar cualquier dato clasificado incorrectamente. Por último, los datos correctamente agrupados y clasificados se utilizan como entrada para

la clasificación supervisada mediante regresión logística. Después de limpiar los datos agrupados a través de k-means, informan haber obtenido 614 pacientes correctamente agrupados, que se utilizan como entrada para entrenar el algoritmo de regresión logística. En conclusión, para la evaluación se informa haber usado validación cruzada de diez veces y haber obtenido una precisión (accuracy) de 97.39 %.

En [59] usando los datos PID crean un modelo predictivo con solo el algoritmo de regresión logística después de preprocesar los valores faltantes (imputando con los valores medios de las columnas). A continuación, se emplean varios métodos para mejorar la precisión. Los métodos incluyen: creación de nuevas características (estas características se crean en base a una investigación de ciertas medidas de diagnóstico aplicables a pacientes diabéticos) y selección de características (utilizando la prueba de chi-cuadrado). Después de la selección de características obtienen un accuracy de 0.7532. Por otro lado, para un segundo conjunto de datos (datos de afroamericanos rurales en Virginia), se empleó la selección de características univariante mediante la prueba de chi-cuadrado, y se eligieron ocho características con la puntuación más alta, después de la selección de características para el segundo conjunto de datos obtienen un accuracy de 0.8889. Posteriormente, métodos de conjunto se utilizan para intentar aumentar el rendimiento.

En varios de los artículos analizados, pudimos notar que el paso de la preparación de datos lo realizan antes de la división de datos en train y test, ya sea a través del Holdout Method o de la validación cruzada. Tal como se analizó en la **sección 2.2.6**, el aplicar transformaciones a todo el conjunto de datos (imputación, normalización, eliminación de valores atípicos, PCA, etc...) conduce a la Fuga de Datos. Recordemos que fuga, es cuando se revela información al modelo que le da una ventaja poco realista para hacer predicciones, cada vez que un modelo recibe información a la que no debería tener acceso hay una fuga de datos. Debido a esto, en este trabajo se realiza de forma rigurosa la evalución de los diferentes modelos analizados usando el método Stratified cross validation (ver **sección 2.6.2**) cuidando que toda la tubería de modelado se prepare sólo en el conjunto de entrenamiento. En [60] realizan una búsqueda sistemática en las bases de datos PubMed y EMBASE para identificar estudios publicados antes de mayo de 2011 que describen el desarrollo de modelos que combinan dos o más variables para predecir el riesgo de diabetes tipo 2 prevalente o incidente. Los autores encuentran que muchos estudios se caracterizaron por un nivel generalmente deficiente de informes, con muchos detalles clave para juzgar objetivamente la utilidad de los modelos a menudo omitidos. En este contexto, en el presente trabajo trataremos de ser rigurosos en la presentación exhaustiva de los métodos empleados.

En suma, varios de los trabajos revisados (por ejemplo; [60], [59] y [6]) utilizan como métricas de rendimiento; Accuracy, Precision, Recall, F1-score y AUC para evaluar el rendimiento del modelo, estas serán las métricas que tomaremos en consideración para la evaluación de nuestro modelo propuesto y tener una base de comparación con otros modelos.

## 3.2. Conjuntos de Datos

El proyecto utiliza principalmente dos conjuntos de datos: uno es el conjunto de datos Pima Indians Diabetes Database (PIDD), que es originalmente del National Institute of Diabetes and Digestive and Kidney Diseases (Instituto Nacional de Diabetes y Enfermedades Digestivas y Renales), el conjunto de datos se puede encontrar en el sitio web de Kaggle [61] y otro conjunto de datos proveniente de la Universidad de Vanderbilt [62]. Con fines de simplicidad nos referiremos al conjunto de datos de los Indios Pima como "datos PID", mientras que al segundo conjunto de la Universidad de Vanderbilt como "datos Vanderbilt".

Para la construcción de nuestro modelo de predicción, se utilizó el conjunto de datos PID, mientras que para evaluar la consistencia del modelo en un conjunto de datos diferente, se decidió utilizar los datos Vanderbilt, que aunque no es análogo al conjunto PID, si permite evaluar métodos claves del modelo principal.

El conjunto de **datos PID**, está compuesto por 768 pacientes de muestra de una población cercana a Phoenix, Arizona, EE. UU. Se trabajará con este conjunto de datos, debido a que en los Indios Pima es alta la prevalencia de diabetes tipo 2. Este grupo ha sobrevivido con una dieta pobre en carbohidratos durante años debido a la predisposición genética a la diabetes. En los últimos años, el grupo Pima obtuvo un alto índice de diabetes debido al repentino cambio de cultivos tradicionales a alimentos procesados.[63]

En particular, todos los pacientes son mujeres de al menos 21 años de edad de origen Indio Pima. Hay un total de nueve características/variables, entre las cuales ocho son variables predictoras y una es la variable objetivo. Las características son las siguientes:

**1. Number of times pregnant:** Número de veces que la paciente estuvo embarazada.

**2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test**: Concentración de glucosa en plasma durante dos horas en una prueba de tolerancia oral a la glucosa.

**3. Diastolic blood pressure**: Presión arterial diastólica (mm Hg).

**4. Triceps skin fold thickness**: Grosor del pliegue cutáneo del tríceps (mm).

**5. 2-Hour serum insulin**: Insulina sérica de dos horas (mu U/ml).

**6. Body mass index**: Índice de masa corporal ($kg/m^2$).

**7. Diabetes Pedigree Function**: Función del pedigrí de la diabetes, una función que califica la probabilidad de diabetes en función de los antecedentes familiares.

**8. Age**: Edad en años.

**9. Class variable**: Variable objetivo, donde la clase 1 indica que una persona tiene diabetes y 0 indica sin diabetes.

En el conjunto de datos, hay un total de 268 casos positivos evaluados y 500 casos negativos evaluados. En la **Tabla 4** podemos ver un resumen estadístico general de las variables predictoras.

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insuline | BMI | PedigreeFunction | Age |
|---|---|---|---|---|---|---|---|---|
| count | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 |
| mean | 3.845052 | 120.894531 | 69.105469 | 20.536458 | 79.799479 | 31.992578 | 0.471876 | 33.240885 |
| std | 3.369578 | 31.972618 | 19.355807 | 15.952218 | 115.244002 | 7.884160 | 0.331329 | 11.760232 |
| min | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.078000 | 21.000000 |
| 25% | 1.000000 | 99.000000 | 62.000000 | 0.000000 | 0.000000 | 27.300000 | 0.243750 | 24.000000 |
| 50% | 3.000000 | 117.000000 | 72.000000 | 23.000000 | 30.500000 | 32.000000 | 0.372500 | 29.000000 |
| 75% | 6.000000 | 140.250000 | 80.000000 | 32.000000 | 127.250000 | 36.600000 | 0.626250 | 41.000000 |
| max | 17.000000 | 199.000000 | 122.000000 | 99.000000 | 846.000000 | 67.100000 | 2.420000 | 81.000000 |

**Tabla 4.** Resumen estadístico general del conjunto de datos PID.

Por otro lado, en la **Tabla 5** observamos las 6 filas iniciales del marco de datos.

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insuline | BMI | PedigreeFunction | Age | ClassVariable |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 6 | 148 | 72 | 35 | 0 | 33.6 | 0.627 | 50 | 1 |
| 1 | 1 | 85 | 66 | 29 | 0 | 26.6 | 0.351 | 31 | 0 |
| 2 | 8 | 183 | 64 | 0 | 0 | 23.3 | 0.672 | 32 | 1 |
| 3 | 1 | 89 | 66 | 23 | 94 | 28.1 | 0.167 | 21 | 0 |
| 4 | 0 | 137 | 40 | 35 | 168 | 43.1 | 2.288 | 33 | 1 |
| 5 | 5 | 116 | 74 | 0 | 0 | 25.6 | 0.201 | 30 | 0 |

**Tabla 5.** Visualización de primeras 6 filas del conjunto de datos PID.

El segundo conjunto de datos (**datos Vanderbilt**), consta de 16 características, contiene tanto pacientes femeninos como masculinos. Los individuos son pacientes afro-americanos rurales en Virginia. En este conjunto, se tienen 60 pacientes diabéticos y 330 que no lo son. Las características son las siguientes:

**1. Patient number:** Identifica a los pacientes por número.

**2. Cholesterol**: Colesterol total.

**3. Glucose**: Azúcar en sangre en ayunas.

**4. HDL**: HDL o colesterol bueno.

**5. Chol/HDL**: Relación entre el colesterol total y el colesterol bueno. El resultado deseable es $< 5$.

**6. Age**: Edad del paciente.

**7. Gender**: Género: 162 hombres, 228 mujeres.

**8. Height**: Altura en pulgadas.

**9. Weight**: Peso: En libras.

**10. BMI**: Índice de masa corporal. ($kg/m^2$).

**11. Systolic BP**: Número superior de la presión arterial.

**12. Diastolic BP**: Número inferior de la presión arterial.

**13. Waist**: Cintura, medida en pulgadas.

**14. Hip**: Cadera, medida en pulgadas.

**15. Waist/hip**: Relación cintura/cadera.

**16. Diabetes**: Variable de clase (Diabetes, No diabetes).

Debido a que la primera característica (Patient number) no aporta ningún tipo de información médica, la eliminamos de nuestro marco de datos, finalmente nos quedamos con 15 variables, 14 variables predictoras y una variable de clase.

En la **Tabla 6** observamos las 6 filas iniciales del marco de datos.

## 3.3. Evaluación de modelos

En este proyecto, como técnica de validación de modelos, haremos uso del método stratified cross-validation de 10 veces. En la **sección 2.6.2**, se da una explicación de

| | Cholesterol | Glucose | HDL Chol | Chol/HDL ratio | Age | Gender | Height | Weight | BMI | Systolic BP | Diastolic BP | waist | hip | Waist/hip ratio | Diabetes |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 193 | 77 | 49 | 3.9 | 19 | 0 | 61 | 119 | 22.5 | 118 | 70 | 32 | 38 | 0.84 | No diabetes |
| 1 | 146 | 79 | 41 | 3.6 | 19 | 0 | 60 | 135 | 26.4 | 108 | 58 | 33 | 40 | 0.83 | No diabetes |
| 2 | 217 | 75 | 54 | 4.0 | 20 | 0 | 67 | 187 | 29.3 | 110 | 72 | 40 | 45 | 0.89 | No diabetes |
| 3 | 226 | 97 | 70 | 3.2 | 20 | 0 | 64 | 114 | 19.6 | 122 | 64 | 31 | 39 | 0.79 | No diabetes |
| 4 | 164 | 91 | 67 | 2.4 | 20 | 0 | 70 | 141 | 20.2 | 122 | 86 | 32 | 39 | 0.82 | No diabetes |
| 5 | 170 | 69 | 64 | 2.7 | 20 | 0 | 64 | 161 | 27.6 | 108 | 70 | 37 | 40 | 0.93 | No diabetes |

**Tabla 6.** Visualización de primeras 6 filas del conjunto de datos Vanderbilt.

esta técnica, se analizan los principales beneficios que tiene este método en comparación de usar una sola división en un conjunto de entrenamiento y prueba, y se ven las ventajas que tiene respecto a la validación cruzada, cuando hay desequilibrio de clases.

Por otro lado, las métricas de evaluación que usaremos a lo largo de este estudio son: Accuracy (ver **ecuación 2.72**), Precision (ver **ecuación 2.77**), Recall (ver **ecuación 2.78**), F1-score (ver **ecuación 2.79**) y AUC (ver **sección 2.6.2**). De igual forma informaremos el valor del pliegue que obtuvo el mínimo y máximo Accuracy, también informaremos el tiempo promedio de ejecución para cada modelo implementado y finalmente desplegaremos la matriz de confusión promedio y la gráfica de las curvas ROC.

Recalcamos, que cada una de estas métricas se calculan sobre cada fold (o pliegue) generado por stratified cross-validation, de este modo la medida final que tomaremos en cuenta será el promedio de las puntuaciones.

A lo largo de todo el proyecto, se usara como software de trabajo el entorno de programación Anaconda haciendo uso del lenguaje Python.

## 3.4. Construcción de modelos predictivos

En esta sección se diseña el Modelo Base, el Modelo 1 (el cual está basado en el modelo propuesto en [6]) y se diseña un modelo predictivo a través de técnicas de preprocesamiento de datos.

### 3.4.1. Modelo Base

Como modelo base, con el cual comparar los subsiguientes modelos implementados, tomaremos un flujo de trabajo muy simple, el cual se muestra en la **Figura 3.1**.

En esta figura podemos observar que para cada pliegue (fold) creado por stratified cross validation, se escalan los datos de entrenamiento (Train) y los datos de prueba (Test) usando la técnica Min-max scaling (ver **ecuación 2.4**). La línea dentada nos indicará a partir de ahora, que el proceso correspondiente en Test, usa algún parámetro

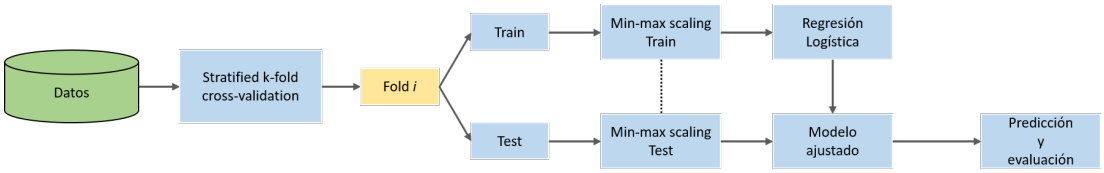



**Figura 3.1:** Flujo de trabajo del Modelo Base.

del conjunto Train. En el caso de la técnica Min-max scaling se hace uso del mínimo y el máximo valor de cada característica en el conjunto Train, con este mínimo y máximo en Train escalamos el conjunto Test, esto es debido a que las transformaciones de datos (tales como la estandarización, la selección de características, PCA, etc...) deben "aprenderse" (o ajustarse) del conjunto de entrenamiento y aplicarse a los datos retenidos para la predicción. En el siguiente paso del flujo de trabajo, se aplica el clasificador de regresión logística para obtener el modelo ajustado, en este modelo y los subsiguientes modelos que se analizarán, usaremos la regresión logística penalizada, usando la regularización L2 o regresión de Ridge (ver sección **2.5.4**) haciendo uso de los mismos parámetros, para tener una comparación justa, este clasificador lo implementamos haciendo uso de la biblioteca scikit-learn de Python. Finalmente como último paso del modelo base, pasamos los datos Test escalados por Min-max scaling por el modelo ajustado para obtener las predicciones y las métricas de evaluación.

| | Precision (Precision ± std) | Recall (Recall ± std) | F1 score (F1 score ± std) |
|---|---|---|---|
| 0 | 0.8390 ± 0.0480 | 0.7720 ± 0.0976 | 0.8010 ± 0.0591 |
| 1 | 0.6434 ± 0.1123 | 0.7198 ± 0.0990 | 0.6725 ± 0.0754 |
| Macro Avg. | 0.7412 ± 0.0689 | 0.7459 ± 0.0617 | 0.7368 ± 0.0648 |
| Weighted Avg. | 0.7708 ± 0.0583 | 0.7538 ± 0.0655 | 0.7562 ± 0.0623 |

| Minimum Accuracy | Maximum Accuracy | Overall Accuracy (Accuracy ± std) | Execution Time (t ± std) s |
|---|---|---|---|
| 0.6579 | 0.8571 | 0.7538 ± 0.0655 | 0.00981 ± 0.00081 |

**Tabla 7.** Métricas de evaluación del Modelo Base.

En la **Tabla 7**, se observan los valores promedios de las métricas: **Precision**, **Recall** y **F1 score**, se muestra el valor de cada una de estas métricas por clase (indicadas por 0 y 1) y de igual forma se da el promedio macro (**Macro avg.**) y el promedio ponderado (**Weighted Avg.**), cada una con sus respectivas desviaciones estándar (std). El promedio macro calcula el promedio sin considerar la proporción de las etiquetas de clase, mientras que en el promedio ponderado se utiliza una ponderación que depende del número de etiquetas verdaderas de cada clase. De igual forma, también se muestran los valores mínimo y máximo, de entre los 10 pliegues de Stratified cross validation (**Minimum Accuracy** y **Maximum Accuracy** respectivamente), mostramos el Accuracy promedio (**Overall Accuracy**) con su desviación estándar y finalmente damos el tiempo de ejecución promedio (**Execution Time**) con su respectiva desviación estándar.

Por otro lado, en la **Figura 3.2**, a la izquierda se muestra la matriz de confusión promedio y a la derecha un gráfico de las 10 curvas ROC obtenidas en cada pliegue, la curva ROC promedio y el valor del área bajo la curva ROC (AUC) con su desviación estándar.

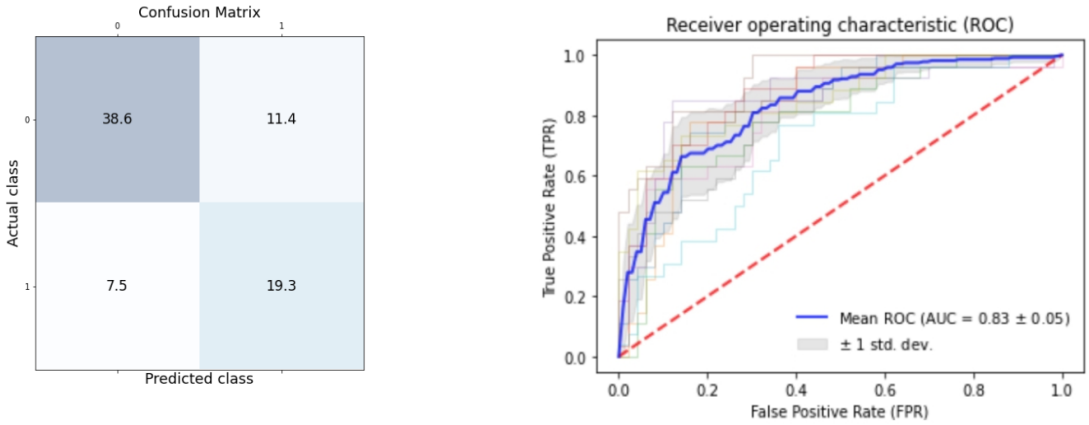



**Figura 3.2:** Matriz de confusión y curva ROC del Modelo Base.

En los modelos que se presentarán a continuación, las métricas de evaluación completas así como la matriz de confusión y la curva ROC se anexarán en el **Apéndice A**, esto para hacer más fluido el contenido de la presente sección. Para comparar el cambio en el rendimiento de los modelos que se irán implementando, se presentaran tablas haciendo uso de las métricas Macro promedio (Macro Avg.) para Precision, Recall y F1-score, y se presentará la medida Accuracy, el área bajo la curva ROC (AUC) y el tiempo de ejecución (Execution Time).

### 3.4.2. Modelo 1

Al observar que los modelos que en la **Tabla 3** reportan las precisiones más altas (modelos Patil et al. [54] y Wu et al. [50]), comparten el uso de k-means para extraer patrones, decidimos explorar la implementación del modelo propuesto en [6], el cual es un trabajo más reciente con métodos similares a los dos anteriores. Esta implementación la realizamos para verificar que también se generalizan estos enfoques al ser rigurosos en que no haya ningún tipo de Fuga de Datos y evitar el sobreajuste del modelo.

Podemos observar nuestra implementación del algoritmo propuesto en [6], en la **Figura 3.3**, para fines de simplicidad nos referiremos a este flujo de trabajo como **Modelo 1**.

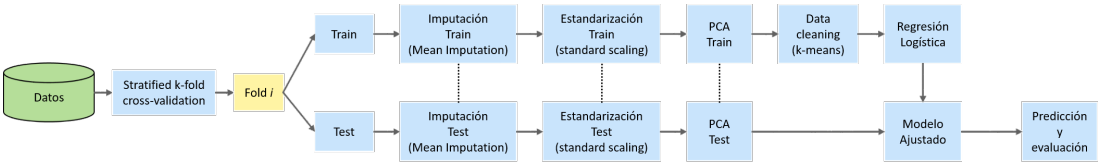



**Figura 3.3:** Flujo de trabajo del Modelo 1.

En este flujo de trabajo se observa que para cada uno de los 10 pliegues, comenzamos imputando los valores perdidos tanto de Train como de Test, usando la técnica Mean Imputation (ver **ecuación 2.1**), a continuación se estandarizan los datos usando standard scaling (ver **ecuación 2.5**), seguido a esto realizamos una reducción dimensional usando PCA (ver **sección 2.3**) quedándonos con el número de componentes principales que hagan que la varianza explicada sea mayor o igual al 65 % (ver **ecuación 2.46**).

A continuación, hacemos una limpieza de datos (Data cleaning), con el objetivo de validar las clases elegidas, haciendo uso del algoritmo k-means (ver **Algoritmo 2**: Algoritmo de Lloyd, **sección 2.4**). En este apartado, se crea un bucle en el cual se realiza la agrupación de k-means (con una sola iteración y con k=2 debido a que la variable "class" tiene dos resultados) variando el valor de la semilla 1000 veces, al variar el valor de la semilla los centroides iniciales van cambiando, de este modo para cada valor semilla se comparan los clústeres obtenidos con las etiquetas reales, nos quedamos con las agrupaciones cuyos centroides proporcionan el mayor porcentaje de etiquetas correctamente clasificadas, finalmente eliminamos los datos con etiquetas incorrectas dentro de cada clúster, quedándonos solo con datos con las etiquetas validadas.

Después del procedimiento de extracción a través de k-means, se obtienen aproximadamente entre el 72 % y 74 % (dependiendo de cada pliegue) de pacientes correctamente clasificados, los cuales sirven como entrada para el algoritmo de regresión logística, con el cual obtenemos nuestro modelo ajustado. Finalmente, pasamos los datos Test reducidos dimensionalmente por PCA por el modelo ajustado para obtener las predicciones

y las métricas de evaluación.

En la **sección A.2**, se muestran las métricas de evaluación completas, así como la matriz de confusión y la curva ROC obtenida del Modelo 1.

### 3.4.3. Modelo Predictivo a través de técnicas de Preprocesamiento de Datos

En esta sección, desarrollaremos paso a paso la forma en que se fueron explorando diferentes vías de preprocesamiento de datos, hasta llegar al Modelo Propuesto en el presente trabajo. En esta fase, llevamos a cabo una investigación sobre métodos de imputación de datos, limpieza de datos y creación de características.

#### 3.4.3.1. Técnicas de Imputación

En la **Figura 3.4**, podemos observar el numero de valores cero que tienen las variables: Glucose, BloodPressure, SkinThickness, Insuline y BMI. El conocimiento médico explica que tales atributos no pueden ser 0, por lo tanto, sugiere que el conjunto de datos contiene valores perdidos, los cuales si no se manejan apropiadamente puede afectar la calidad del resultado y la precisión de nuestro modelo. Para el manejo de estos valores compararemos tres técnicas de imputación: La técnica **Imputación-I** se refiere a la imputación **Mean Imputation** (ver **ecuación 2.1**) en la que se calcula la media de cada atributo con valores perdidos y luego se imputa en las celdas de datos que faltan, **Imputación-II** se refiere a la técnica de **Group Mean Imputation** (ver **ecuación 2.2**) en donde los valores faltantes en cada columna se reemplazan con la media del grupo (o clase) de todos los valores conocidos de la columna e **Imputación-III** se refiere a la técnica de **Regression Imputation**, en esta técnica usaremos como variables predictivas a las columnas que no tiene valores perdidos (Pregancies, PedigreeFunction y Age) y se ajustará un modelo de regresión lineal tomando como variables de respuesta a cada una de las columnas con valores perdidos.

```
Pregnancies              0
Glucose                  5
BloodPressure           35
SkinThickness          227
Insuline               374
BMI                     11
PedigreeFunction         0
Age                      0
```

**Figura 3.4:** Número de valores perdidos en el conjunto de datos PID.

En la **Figura 3.5** se muestra el flujo de trabajo implementado para comparar las tecnicas de imputación.

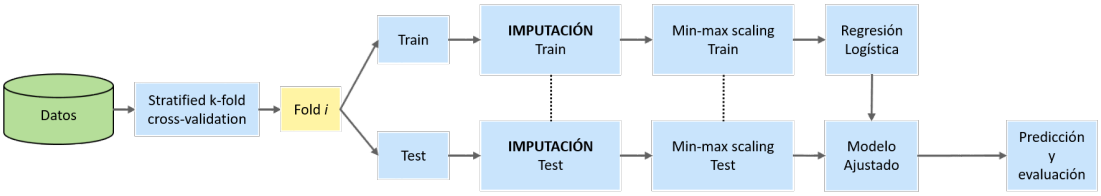



**Figura 3.5:** Flujo de trabajo implementado para la comparación de las técnicas de imputación.

En este flujo de trabajo, podemos observar que para cada pliegue (fold) generado por Stratified k-fold cross validation, el primer paso es imputar los valores perdidos (aplicando por separado cada una de las técnicas de imputación antes mencionadas) tanto del conjunto Train como de Test, a continuación se escalan los datos aplicando la técnica Min-max scaling, seguido a esto, obtenemos el modelo ajustado aplicando el clasificador de regresión logística y finalmente pasamos los datos Test escalados al modelo ajustado para obtener las predicciones y las métricas de evaluación.

En la **Tabla 8** para el Modelo Base, Imputación I, Imputación II e Imputación III, se muestra el promedio macro (Macro Avg) de las métricas: Precision, Recall y F1-score. También se informan el Accuracy promedio, el área bajo la curva ROC (AUC) y el tiempo de ejecución (Execution Time) en segundos.

| Modelos | Accuracy | Precision | Recall | F1 score | ROC (AUC) | Execution Time [s] |
|---|---|---|---|---|---|---|
| **Modelo Base** | 0.7538 | 0.7412 | 0.7459 | 0.7368 | 0.83 | 0.00981 |
| **Imputación I** | 0.7552 | 0.7421 | 0.7452 | 0.7376 | 0.84 | 0.025 |
| **Imputación II** | 0.7812 | 0.7679 | 0.7755 | 0.7663 | 0.86 | 0.035 |
| **Imputación III** | 0.7526 | 0.7393 | 0.7423 | 0.7347 | 0.84 | 0.161 |

**Tabla 8.** Métricas de evaluación del modelo base y de las técnicas de imputación.

Analizando estas métricas podemos observar que respecto al Modelo Base, la técnica Imputación II (Group Mean Imputation) es la que mejora apreciablemente el rendimiento del modelo, debido a esto decidimos quedarnos con esta técnica de imputación en la siguiente etapa, donde se exploran técnicas de limpieza de datos.

En la **sección A.3**, se muestran las métricas de evaluación completas, así como la matriz de confusión y la curva ROC obtenida para cada una de las técnicas de imputación implementadas.

#### 3.4.3.2. Técnicas de limpieza de datos (Data cleaning)

Quedándonos con la imputación Group mean, a continuación probamos tres técnicas de limpieza de datos, estas técnicas por simplicidad las hemos llamado: **Limpieza I**, **Limpieza II** y **Limpieza III**. A continuación se detalla cada una de estas técnicas.

**Limpieza I**: Esta técnica de limpieza de datos, la cual en sí, es una técnica de validación de etiquetas, es la misma que se ha aplicado en el modelo 1. Creamos un bucle en el cual se realiza la agrupación de k-means (con una sola iteración y con k=2 debido a que la variable "class" tiene dos resultados) variando el valor de la semilla 1000 veces, al variar el valor de la semilla los centroides iniciales van cambiando, de este modo para cada valor semilla se comparan los clústeres obtenidos con las etiquetas reales y nos quedamos con las agrupaciones cuyos centroides proporcionan el mayor porcentaje de etiquetas correctamente clasificadas, finalmente eliminamos los datos con etiquetas incorrectas dentro de cada clúster, quedandonos solo con datos con las etiquetas validadas.

**Limpieza II**: En esta técnica nuevamente aplicamos el algoritmo de agrupación k-means, esta técnica es empleada para remover valores atípicos que se mantiene alejados de los clústeres creados, para esto, debemos definir un valor umbral y los datos cuyas distancia al centro de su clúster (centroide) quedan fuera del umbral se contarán como valores atípicos.[65]

Podemos ver de forma más clara este proceso en la **Figura 3.6**.

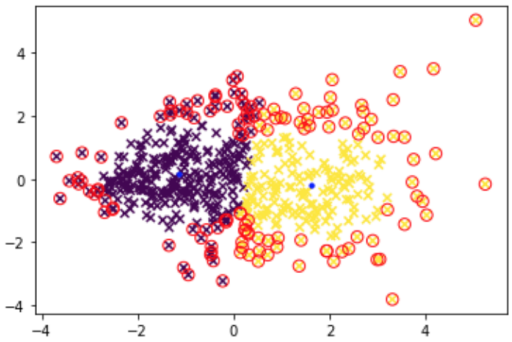



**Figura 3.6:** Técnica Limpieza II. En morado y amarillo se observan los clústeres creados por el algoritmo k-means, en azul se observan los centroides de cada clúster y remarcados en un círculo rojo los valores atípicos removidos por la técnica Limpieza II.

En esta figura, para fines de ilustración, hemos proyectado los datos imputados por Imputación II (Group Mean Imputation), a las dos primeras componentes principales aplicando la técnica de reducción de dimensionalidad PCA. El primer paso para remover los valores atípicos, será encontrar los centros de los clústeres y luego calcular las distancias entre cada punto al centro de su clúster, en nuestro caso como métrica de distancia elegimos la distancia euclideana (aunque otras métricas pueden ser exploradas). A continuación se decide una valor umbral, llamemosle $U$ y se ordenan todas las distancias de todos los puntos (a sus propios centros), de este modo se define como valor atípico a los datos cuya distancia a su clúster está por encima del percentil (umbral) $U$ elegido. En nuestro caso este valor umbral lo hemos establecido en 75 %. En la **Figura 3.6**, podemos ver a los valores atípicos removidos remarcados en un círculo rojo.

**Limpieza III**: Una de las técnicas más simples y más empleada en la identificación de valores atípicos es el método Z-score, debido a la simplicidad de este método es que hemos decidimos emplearlo para compararlo con los métodos basados en K-means. En la **sección 2.2.2.3** se ha descrito en detalle la implementación de este método.

Cada una de estas técnicas de limpieza de datos se aplica de acuerdo al flujo de trabajo mostrado en la **Figura 3.7**.

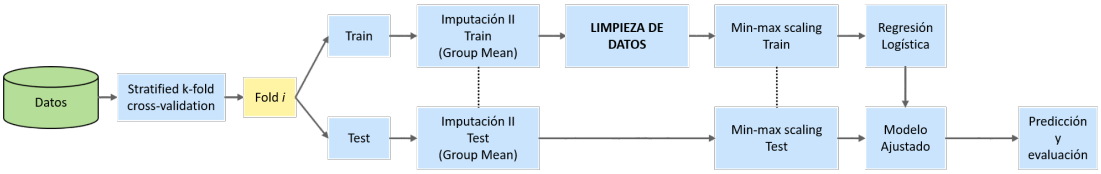



**Figura 3.7:** Flujo de trabajo implementado para la comparación de las técnicas de limpieza de datos.

En este flujo de trabajo, podemos ver que para cada pliegue, se realiza en primer lugar la imputación de valores perdidos aplicando la técnica Imputación II (Group Mean), a continuación se realiza la limpieza de datos en el conjunto de datos de entrenamiento (Train) ya imputado, seguido a esto se escalan los datos usando el algoritmo Min-max scaling, y se obtiene el modelo ajustado usando el clasificador de regresión logítica, finalmente se pasan los datos de prueba por el modelo ajustado para obtener las predicciones y las métricas de evaluación.

En la **Tabla 9** podemos observar las métricas de evaluación obtenidas para cada una de las técnicas de limpieza de datos (Limpieza I, Limpieza II y Limpieza III), y para ver de forma más clara el cambio en el rendimiento de los modelos, hemos incluido las métricas del Modelo Base y las métricas de la técnica de Imputación II.

| Modelos | Accuracy | Precision | Recall | F1 score | ROC (AUC) | Execution Time [s] |
|---|---|---|---|---|---|---|
| **Modelo Base** | 0.7538 | 0.7412 | 0.7459 | 0.7368 | 0.83 | 0.00981 |
| **Imputación II** | 0.7812 | 0.7679 | 0.7755 | 0.7663 | 0.86 | 0.035 |
| **Limpieza I** | 0.8112 | 0.7987 | 0.8021 | 0.7961 | 0.88 | 2.986 |
| **Limpieza II** | 0.7773 | 0.7633 | 0.7699 | 0.7610 | 0.86 | 0.058 |
| **Limpieza III** | 0.8060 | 0.7936 | 0.8094 | 0.7952 | 0.88 | 0.035 |

**Tabla 9.** Métricas de evaluación de las técnicas de limpieza de datos.

Analizando las métricas de evaluación en la **Tabla 9**, podemos observar que la técnica de validación de etiquetas Limpieza I, es la que obtiene un mejor rendimiento de las técnicas exploradas (respecto al modelo base y a Imputación II), sin embargo la técnica Limpieza III mantiene un rendimiento cercano, con la ventaja de que el tiempo de ejecución de esta técnica es mucho menor que el de Limpieza I (85 veces menor), por esta razón para el siguiente apartado que es la creación de características, decidimos quedarnos con la técnica Limpieza III (método Z-score).

En la **sección A.4**, se muestran las métricas de evaluación completas, así como la matriz de confusión y la curva ROC obtenida para cada una de las técnicas de limpieza de datos exploradas.

#### 3.4.3.3. Creación automatizada de características

Teniendo en cuenta que el modelo de regresión logística, es un modelo lineal generalizado, o en otras palabras, la variable de salida del modelo está relacionada con las variables de entrada a través de una "función de enlace", es decir, la función sigmoidea de una combinación lineal de las características, la regresión logística no puede capturar una relación no lineal más compleja con las características. De este modo para mejorar el rendimiento de nuestro modelo de regresión logística, decidimos aplicar la ingeniería de características a través de la creación de características, que como vimos en la **sección 2.2.2.4**, es una técnica donde se construyen variables explicativas que permiten mejorar el rendimiento del modelo.

Al revisar la literatura, se pudo encontrar un trabajo ([59]) en donde los autores ejecutan un modelo de predicción de diabetes, haciendo uso del conjunto de datos PID y empleando el diseño de características (en la **sección 3.1** se da una reseña de este trabajo). En este modelo los autores crean nuevas características en base a una investigación de ciertas medidas de diagnóstico aplicables a pacientes diabéticos. Las

características que los autores crean son las siguientes:

NF1 - Edad menor o igual a 30 y Valor de Glucosa menor o igual a 140.

NF2 - IMC menor o igual a 30.

NF3 - Edad menor o igual a 30 años y Embarazos menor o igual a 3.

NF4 - Valor de glucosa menor o igual a 140 y Presión Arterial menor o igual a 80.

NF5 - Valor de glucosa menor o igual a 140 e IMC menor o igual a 45.

Como mencionamos, en la **sección 2.2.2.4** la creación manual de características suele ser un procedimiento complicado, que requiere una gran cantidad de tiempo, por otro, lado la creación automatizada de características suele ahorrar tiempo, y tener la ventaja de ser generalizable a otros conjuntos de datos. De este modo, uno de los objetivos centrales de nuestro proyecto fue el diseñar una técnica de creación automatizada de características que permitiera mejorar el rendimiento del clasificador de regresión logística.

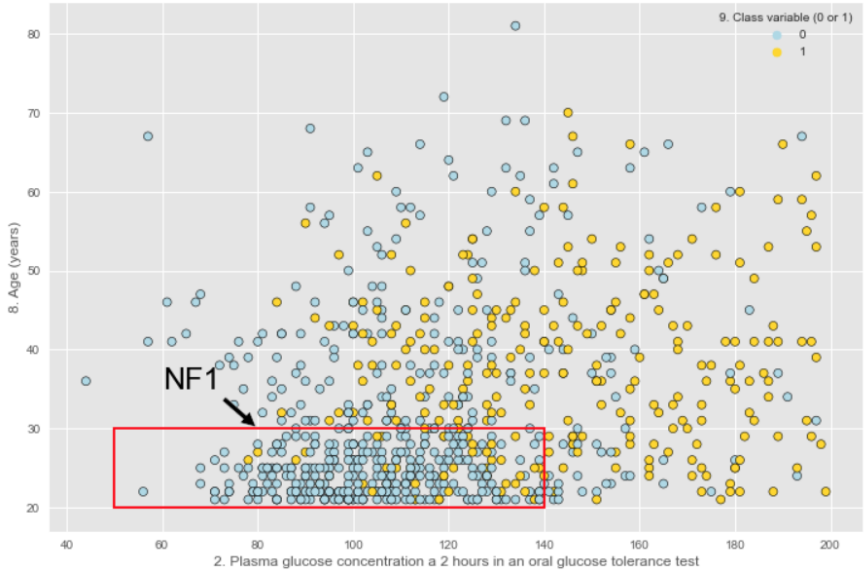



**Figura 3.8:** Diagrama de dispersión Glucosa vs Edad. En azul se muestran los pacientes pertenecientes a la clase 0 (no diabetes) y en amarillo los pertenecientes a la clase 1 (diebetes). Se encierra en el rectángulo rojo los pacientes contenidos en los umbrales de la característica NF1.

Podemos observar en primera instancia que las características NF1, NF3, NF4 y

NF5, pueden verse como características creadas al elegir ciertos umbrales en un diagrama de dispersión generado por dos características originales, por ejemplo; para NF1, si realizamos el diagrama de dispersión Glucosa vs Edad (ver **Figura 3.8**), podemos observar en el rectangulo rojo los valores que respetan los umbrales de Edad menor o igual a 30 y Valor de Glucosa menor o igual a 140. Se observa que en esta región hay una alta densidad de pacientes no diabeticos, de modo que la nueva característica se crea dando un valor de 1 a los datos que se encuentran dentro de los umbrales y dando un valor 0 en caso contrario.

Motivados por esta idea, para automatizar el proceso de diseñar nuevas características, decidimos aplicar tres diferentes técnicas, al construir todos los diagramas de dispersión posibles sobre el conjunto de datos. Al tener nuestro conjunto de datos 8 características podemos usar la fórmula de combinaciones sin repetición para obtener el número de diagramas de dispersión que se estarán generando:

$$C_{n,x} = \frac{n!}{x!(n-x)!} = \frac{8!}{2!(8-2)!} = 28$$

De este modo, se generarán 28 diagramas de dispersión y por lo tanto 28 nuevas características en nuestro marco de datos.

La primera técnica de creación automatizada de características, a la cual llamaremos por simplicidad **Creación I**, está guiada por la idea de obtener sobre cada uno de los 28 diagramas de dispersión la región de mayor densidad datos, para esto dividimos en un marco rectangular de $3 \times 3$ el diagrama de dispersión y sobre cada cuadrante se cuenta cuantos datos hay, de modo que la nueva característica se crea tomando como valores umbrales el cuadrante sobre el que se encuentra el mayor número de datos, esto es, si el dato pertenece al cuadrante con mayor densidad de datos, a ese dato le damos el valor de 1 y en caso contrario el valor de 0, en el diagrama izquierdo de la **Figura 3.9** podemos observar este procedimiento. En la figura se muestra el diagrama de dispersión Insulina vs Glucosa, se puede observar que el cuadrante con mayor densidad contiene 258 pacientes, de modo que la nueva característica se crea dando un valor de 1 a los individuos contenidos en el cudrante y un valor de 0 en caso contrario. Para la ejecución de esta técnica fue modificado el código fuente del paquete "QStatistic" (Quadrat analysis of point pattern) de la librería "pointpats" de Python (código fuente en [64]).

La segunda técnica de creación automátizada de características que fue diseñada, a la cual llamaremos **Creación II**, es la aplicación de la técnica de agrupamiento K-means (tomando $k = 2$) sobre cada diagrama de dispersión, de modo que las nuevas característica se contruyen en base a la pertenencia de los datos al clúster generado, esto es, daremos el valor de 0 si los datos pertenecen a un clúster y el valor de 1 si pertenecen al otro clúster. En el diagrama central de la **Figura 3.9** se puede observar esta implementación.

Debido al bajo desempeño mostrado por la técnica Creación II, se decidió implementar una variante del agrupamiento k-means llamado fuzzy C-means, técnica que se analiza en detalle en la **sección 2.4.4**. Guiados por el hecho de que varios diagramas de dispersión tiene datos superpuestos y es sabido que el agrupamiento difuso de c-means brinda resultados comparativamente mejores para conjuntos de datos superpuestos, es que se decidió explorar esta técnica para la creación automatizada de características. A esta técnica la denominaremos **Creación III**. En este caso para crear nuevas características se implementa el agrupamiento fuzzy C-means (con $k = 2$ y un parámetro de borrosidad $m = 2$) sobre cada diagrama de dispersión, de modo que las nuevas características, son características binarias de valor 0 y 1, dependiendo de la pertenencia a los clústeres generados por fuzzy C-means. En el diagrama a la derecha de la **Figura 3.9** se muestra este método de creación de características.

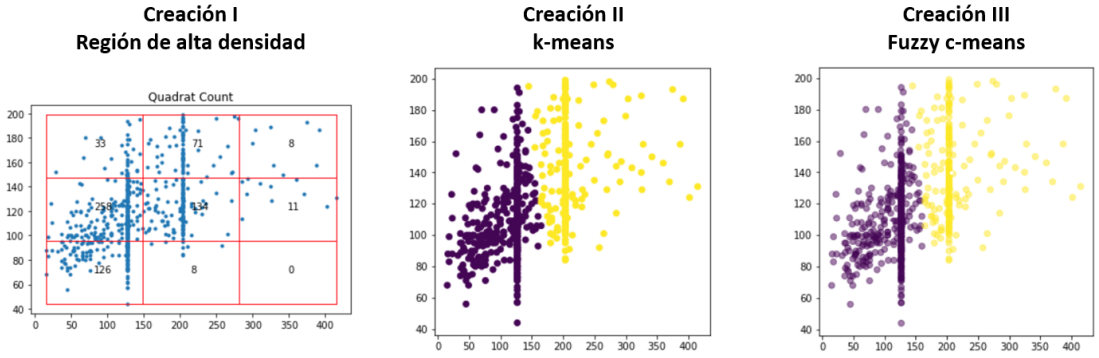



**Figura 3.9:** Técnicas de creación de características. En esta figura podemos observar sobre el diagrama de dispersión Insulina vs Glucosa, la aplicación de las tres diferentes técnicas de creación automatizada de características diseñadas en el presente proyecto.

El flujo de trabajo utilizado para realizar la comparación de rendimientos de las técnicas de creación aumatizadas de características, se muestra en la **Figura 3.10**.

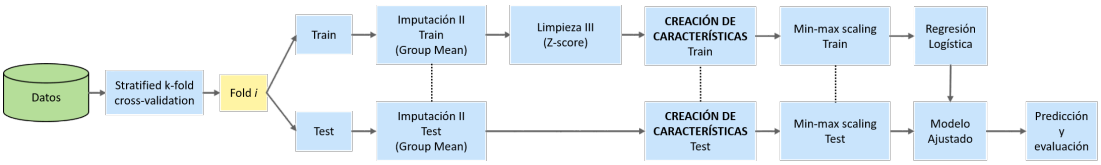



**Figura 3.10:** Flujo de trabajo implementado para realizar la comparación de las técnicas de creación automatizadas de características.

En este flujo podemos observar, que para cada uno de los 10 pliegues dados por stratified k-fold cross-validation, se ejecuta en primer lugar la técnica de Imputación II (Group Mean), a continuación sobre los datos de entrenamiento se aplica la técnica

de limpieza de datos Z-score y seguido a esto se realiza la Creación de Características. La creación de características se realiza primero sobre los datos de entrenamiento y con los parámetros ajustados de cada técnica de creación (es decir, los umbrales para Creación I o la pertenencia al clúster en Creación II y III) es que se realiza la creación de características en los datos de prueba, recordemos que no es correcto que se fugue ningún tipo de información de los datos de prueba a cualquier tipo de transformación de datos. Seguido a la creación de características, aplicamos la técnica Min-max scaling, obtenemos el modelo ajustado con el clasificador de regresión logística y finalmente pasamos los datos de prueba por el modelo ajustado para obtener las predicciones y las métricas de evaluación.

En la **Tabla 10**, se puede observar que la técnica Creación I, es la que mejora en mayor medida las métricas de rendimiento, sin embargo la técnica Creación III tiene un rendimiento similar con la ventaja de que el tiempo de ejecución es aproximadamente dos veces más rápido que en Creación I, debido a esto para el siguiente apartado decidimos continuar con la técnica Creación III.

En la **sección A.5**, se muestran las métricas de evaluación completas, así como la matriz de confusión y la curva ROC obtenida para cada una de las técnicas de creación automatizada de características.

#### 3.4.3.4. Modelo ANOVA F-Test + Creación III

Crear características si mejora de forma considerable el rendimiento del modelo, sin embargo, el pasar de un marco de datos de 8 variables explicativas a uno de 36 variables puede tener ciertas desventajas (tales como un gran costo computacional o dificultar la interpretación del modelo), y ya que en este proyecto se busca crear un modelo que pueda ser generalizable a otros conjuntos de datos, es que se decidió explorar la alternativa de crear nuevas características usando solamente la mitad de variables, y ver si es posible obtener un rendimiento similar que en el paso anterior, al usar todas las características.

De este modo, en este apartado se usa la técnica de selección de características ANOVA F-Test (ver **sección 2.2.4.4**) eligiendo las cuatro características cuyo estadístico $F$ tengan el mayor valor, o dicho de otro modo desechando cuatro características que son independientes de la variable de respuesta según la prueba ANOVA F-Test. Al crear las nuevas características utilizando únicamente las cuatro características seleccionadas por la prueba ANOVA F-Test, se estarán generando 6 nuevas características, de este modo pasamos de un marco de datos de 8 variables explicativas a uno de 14 variables, es decir 22 variables menos que en el paso anterior al usar todas las características.

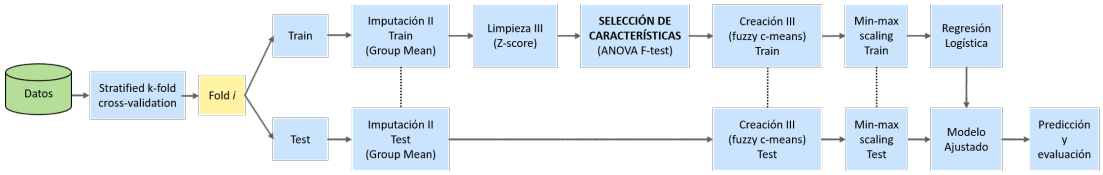



**Figura 3.11:** Flujo de trabajo al realizar Selección de Características a través de la técnica ANOVA F-Test previo a la creación de características.

En la **Figura 3.11**, podemos observar el flujo de trabajo seguido para este análisis. En este flujo vemos que para cada uno de los 10 pliegues dados por stratified k-fold cross-validation, se ejecuta en primer lugar la técnica de Imputación II (Group Mean), a continuación sobre los datos de entrenamiento se aplica la técnica de limpieza de datos Z-score y seguido a esto se realiza la Selección de Características usando la técnica ANOVA F-Test, con las cuatro "mejores" características seleccionadas se procede a crear las nuevas variables usando la técnica Creación III, a continuación se escalan los datos usando Min-max scaling y se obtiene el modelo ajustado usando la regresión logística. Finalmente, los datos de prueba se pasan por el modelo ajustado para realizar las predicciones y obtener las métricas de evaluación.

En la **Tabla 10**, se puede observar que el presente modelo (ANOVA + Creación III), mejora todas las métricas evaluadas respecto a Creación III (exceptuando la métrica AUC que permanece igual) y se acerca al rendimiento mostrado por Creación I, con la ventaja de que el tiempo de ejecución es nueve veces menor que el de Creación I y cuatro veces menor que el de Creación III.

En la **sección A.6**, se muestran las métricas de evaluación completas, así como la matriz de confusión y la curva ROC obtenida para el presente modelo (ANOVA F-Test + Creación III).

#### 3.4.3.5. Modelo ANOVA F-Test + Creación III + PCA

Por último, como paso final, decidimos aplicar el Análisis de Componentes Principales (o PCA por sus siglas en inglés) el cual analizamos en detalle en la **sección 2.3**.

Debido a que en PCA, los componentes principales son ortogonales y esto puede eliminar los factores de influencia mutua entre las variables de un marco de datos, es que se decidió explorar si esta técnica permite mejorar aún más el rendimiento del modelo. Esto debido que al estar creando nuevas características basadas en diagramas de dispersión de las variables originales, puede haber variables altamente correlacionadas, de modo que esperamos que PCA pueda reducir cierto "ruido" en los datos que permita mejorar las métricas de evaluación.

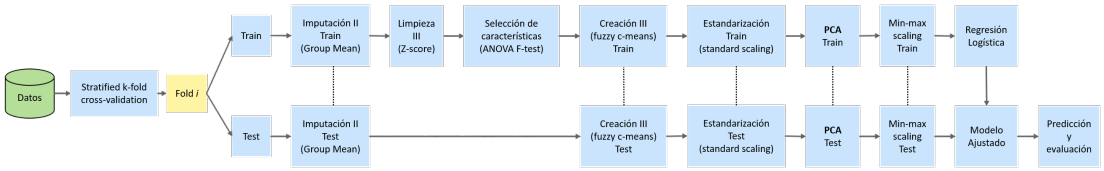



**Figura 3.12:** Flujo de trabajo al realizar Selección de Características a través de la técnica ANOVA F-Test previo a la creación de características y la aplicación posterior del Análisis de Componentes Principales (PCA).

En la **Figura 3.12**, se observa el flujo de trabajo implementado. Este flujo es análogo al presentado en la sección anterior hasta la creación de características, seguido a la creación, debido a que se implementará PCA, estandarizamos los datos usando la técnica standard scaling, esto debido a que la estandarización se debe realizar cuando las unidades en las que se expresan las diferentes características de los datos difieren sustancialmente (tal como vimos en la **sección 2.2.4.2**). Seguido al proceso de estandarización, aplicamos la reducción dimensional PCA, en este caso la elección de cuantas componentes principales escoger fue elegida de forma empírica, ya que no hay un criterio que pueda ser implementado de forma general, debido a que la construcción de las componentes principales dependen de la forma específica del contenido de información en un conjunto de datos. De este modos nos quedamos con las componentes donde las métricas promedio del modelo mostraron un mejor rendimiento, esto sucedio al tomar 8 componentes principales, con estas componentes para cada pliegue se obtiene aproximadamente el 94 % de varianza explicada. En la **Figura 3.13** se observa un gráfico de la varianza individual explicada por cada una de las 8 componentes principales y la varianza acumulada. Despúes de reducir las dimensiones del marco de datos, procedemos a aplicar la técnica Min-max scaling para finalmente al aplicar la regresión logística obtener el modelo ajustado; como paso final, pasamos los datos de prueba por el modelo ajustado para obtener las predicciones y las métricas de evaluación.

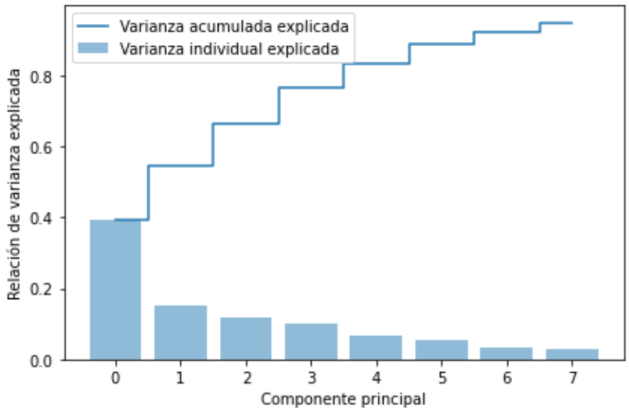



**Figura 3.13:** Varianza individual explicada y varianza acumulada explicada por las primeras 8 componentes principales. El gráfico fue obtenido al realizar una sola división Train-Test (70 % datos de entrenamiento y 30 % datos de prueba), sin embargo para los pliegues generados por stratified k-fold cross-validation la varianza acumulada e individual no difieren significativamente a este gráfico.

En la **Tabla 10** podemos observar que al aplicar PCA, las métricas; Accuracy, Precision, Recall y F1-score mostraron un aumento respecto al paso anterior sin la reducción dimensional y aunque el área bajo la curva ROC (AUC) descendió un punto, podemos ver de forma más clara comparando las Tablas **A.12** y **A.13** que el Recall para la clase diabética (etiqueta 1) muestra un aumento significativo. De este modo, vemos que en este caso, PCA puede hacer que con la elección correcta de componentes principales, las predicciones se sesguen menos hacia la clase mayoritaria.

| Modelos | Accuracy | Precision | Recall | F1 score | ROC (AUC) | Execution Time [s] |
|---|---|---|---|---|---|---|
| Modelo Base | 0.7538 | 0.7412 | 0.7459 | 0.7368 | 0.83 | 0.00981 |
| Limpieza III | 0.8060 | 0.7936 | 0.8094 | 0.7952 | 0.88 | 0.035 |
| Creación I | 0.8451 | 0.8307 | 0.8422 | 0.8333 | 0.91 | 1.189 |
| Creación II | 0.5796 | 0.5973 | 0.5941 | 0.5581 | 0.54 | 3.024 |
| Creación III | 0.8386 | 0.8271 | 0.8388 | 0.8281 | 0.91 | 0.539 |
| ANOVA + Creación III | 0.8411 | 0.8297 | 0.8408 | 0.8305 | 0.91 | 0.130 |
| ANOVA + Creación III + PCA (Modelo final) | 0.8490 | 0.8372 | 0.8503 | 0.8389 | 0.90 | 0.135 |

**Tabla 10.** Métricas de evaluación de técnicas de creación automatizada de características.

#### 3.4.3.6. Aplicación del Modelo Propuesto al conjunto de datos Vanderbilt

Para ver el comportamiento del modelo final propuesto (**sección 3.4.3.5**), en un conjunto de datos externo, decidimos utilizar el conjunto de datos Vanderbilt (ver descripción del marco de datos en **sección 3.2**).

El flujo de trabajo implementado, se observa en la **Figura 3.12**, con la omisión del paso de imputación (ya que en este marco de datos no encontramos valores perdidos). De este modo, comenzamos creando nuestros pliegues usando Stratified cross-validation, debido a los pocos datos, únicamente usamos 5 pliegues, de este modo para cada pliegue comenzamos aplicando la técnica Limpieza III (Z-score) para eliminar valores atípicos, a continuación usando la técnica ANOVA F-Test se eligen las 7 "mejores" características (la mitad de las características originales que son 14), con estas 7 características procedemos a realizar la creación de características usando la técnica Creación III, de este modo se nos crean 21 nuevas características, teniendo finalmente nuestro marco de datos ampliado 35 características. Con el conjunto de datos ampliado, se estandarizan los datos usando standard scaling para aplicar la reducción dimensional a través de PCA, en este caso elegimos las primeras 19 componentes principales, con el conjunto proyectado a estas componentes principales procedemos a escalar los datos usando Min-max scaling, para finalmente obtener el modelo ajustado con el algoritmo de regresión logística; como paso final, pasamos los datos de prueba por el modelo ajustado para realizar predicciones y obtener las métricas de evaluación.

En la **Tabla 11**, observamos las métricas del Modelo Base (construido empleando

el mismo flujo de trabajo que en los datos PID, este flujo se observa en la **Figura 3.1**), el Modelo Propuesto y un modelo propuesto en el trabajo [59], donde los autores usan el conjunto Vanderbilt y aplican 'SelectKBest' y 'chi2' de la biblioteca sklearn para la selección de características univariante en Python, seleccionan las ocho características con la puntuación más alta y se lleva a cabo la predicción mediante la regresión logística con las características seleccionadas, en este caso los autores realizan una división Train-Test con 70 % para train y 30 % para test.

En esta tabla se observa que el modelo propuesto mejora la métrica Accuracy, Precision, Recall y F1 score respecto al Modelo Base, y respecto al Modelo [59] hay una mejora sustancial en la métrica Recall.

| Modelos | Accuracy | Precision | Recall | F1 score | ROC (AUC) | Execution Time [s] |
|---|---|---|---|---|---|---|
| **Modelo Base** | 0.8821 | 0.7861 | 0.8485 | 0.8037 | 0.93 | 0.012 |
| **Modelo Propuesto** | 0.9026 | 0.8105 | 0.8879 | 0.8365 | 0.93 | 0.349 |
| **Modelo [59]** | 0.8889 | 0.88 | 0.76 | 0.80 | --- | 0.005 |

**Tabla 11.** Métricas de evaluación del Modelo Base, Modelo Propuesto y Modelo del artículo [59].

Para comparar de forma más directa con el modelo dado en [59] decidimos aplicar el modelo propuesto en una sóla división Train-Test usando la proporción 70 % para train y 30 % para test. En la **Tabla 12** se muestran las métricas de evaluación dadas en [59] y en la **Tabla 13**, se muestran las métricas obtenidas al aplicar nuestro modelo final propuesto.

| | Precision | Recall | F1 score | Support |
|---|---|---|---|---|
| 0 | 0.89 | 0.98 | 0.93 | 93 |
| 1 | 0.87 | 0.54 | 0.67 | 24 |
| Accuracy | | | 0.89 | 117 |
| Macro Avg | 0.88 | 0.76 | 0.80 | 117 |
| Weighted Avg | 0.89 | 0.89 | 0.88 | 117 |

**Tabla 12.** Métricas de evaluación dadas por el modelo propuesto en [59] para el conjunto de datos Vanderbilt.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.95 | 0.91 | 0.92 | 95 |
| 1 | 0.65 | 0.77 | 0.71 | 22 |
| accuracy | | | 0.88 | 117 |
| macro avg | 0.80 | 0.84 | 0.82 | 117 |
| weighted avg | 0.89 | 0.88 | 0.88 | 117 |

**Tabla 13.** Métricas de evaluación obtenidas al aplicar el Modelo Propuesto en una sola división Train-Test con una proporcion 70-30.

Podemos observar que para la clase de interés 1 (diabéticos), la métrica Precision obtenida en **Tabla 12** es de 0.87 mientras que en nuestro modelo propuesto es de 0.65, sin embargo, el Recall observado en la **Tabla 12** es de 0.54 mientras que en **Tabla 13** es de 0.77, de este modo vemos que aunque el modelo propuesto reduce Precision, aumenta el Recall respecto al modelo dado en [59].

Cuando se contruyen modelos enfocados en el diagnóstico de enfermedades, es mejor el tener falsos positivos; esto es, está bien el afirmar falsamente que un paciente tiene una enfermedad y luego darse cuenta que la decisión fue incorrecta, sin embargo, el falso negativo significa que alguien que tiene realmente la enfermedad no recibe el tratamiento adecuado debido a la errónea predicción, lo cual es mucho más delicado.

Recordemos que la métrica Precision es una medida de cuántos de los pacientes predichos por el clasificador como positivos son realmente positivos, esto nos da una medida de qué tan seguro estamos de que nuestros pacientes predichos como positivos sean realmente positivos; o dicho de otro modo, Precision responde la pregunta: ¿Cuántos de los que etiquetamos como diabéticos son en realidad diabéticos?.

Por otro lado, la métrica Recall responde a la pregunta: De todas las personas que son diabéticas, ¿cuántas de ellas predecimos correctamente?, esto es, Recall es la proporción de los positivos correctamente etiquetados por nuestro programa respecto a todos los que son en realidad diabéticos.

Se le da más importancia al Recall, cuando la idea de obtener falsos positivos es mucho mejor que la de los falsos negativos, en otras palabras, si la ocurrencia de falsos negativos es inaceptable; como en nuestro caso al predecir diabetes. Dicho de otro modo, preferimos etiquetar a algunas personas sanas como diabéticas antes que dejar a una persona diabética etiquetada como sana. En este sentido, vemos que el aumento del Recall dado por el modelo propuesto a expensas de la métrica Precision, es un aspecto positivo del modelo.

En la **sección A.8** se muestran las métricas completas de evaluación del Modelo Base y del Modelo Propuesto, aplicados al conjunto de datos Vanderbilt.

Capítulo 4

# Discusión y conclusiones

La primera fase del presente proyecto, fue analizar de forma rigurosa (esto es, evitando todo tipo de fuga de datos y a través de la técnica Stratified k-fold cross-validation), el enfoque de utilizar el algoritmo de k-means para extraer patrones; para esto, se implementó el flujo de trabajo propuesto en [6], el cual aplicado con rigor se ejemplifica con el Modelo 1 (ver **sección 3.4.2**).

Al realizar el análisis del modelo propuesto en [6], encontramos que el preprocesamiento de datos adolece del problema de la fuga de datos, y por otro lado, al ejecutarse la validación de etiquetas a través del algoritmo k-means sobre todo el conjunto de datos y pasar los datos correctamente etiquetados al algoritmo de regresión logística, realmente se le está pasando al algoritmo de regresión, datos que ya fueron previamente clasificados de forma correcta por el algoritmo de k-means, de modo que es de esperar que el algoritmo de regresión logística clasifique también de forma correcta la mayoría de los datos, lo cual explica que en este enfoque se tengan métricas extremadamente altas, por ejemplo, algunas métricas reportadas en [6] son: un Accuracy del 97.40 %, Recall para la clase negativa (0 o no-diabéticos) del 0.99, Recall para la clase positiva (1 o diabéticos) del 0.94 y un AUC del 0.967. Sin embargo, al aplicar este enfoque de forma rigurosa a través del Modelo 1, pudimos constatar que realmente no se mejoran las métricas de rendimiento de la regresión logística respecto al Modelo Base.

Por otro lado, en la segunda fase del proyecto nos enfocamos en diseñar un modelo que a través de técnicas de prepocesamiento de datos permitiera mejorar el rendimiento del clasificador de regresión logística. Para esto, fueron analizadas técnicas de imputación y limpieza de datos, y fueron diseñadas diversas técnicas de creación automatizada de características. Finalmente, el Modelo Propuesto es el observado en la **sección 3.4.3.5**. A continuación analizaremos algunos detalles de este modelo.

Destacamos en primer lugar que estamos haciendo uso de una regresión logística penalizada, ya que la idea es construir un modelo que tenga una buena generalización, y el uso de la regularización (ver **sección 2.5.4**) permite minimizar la complejidad del

modelo a la vez que minimiza la función de costo. Esto resulta en modelos más simples que tienden a generalizar mejor. Los modelos que son excesivamente complejos tienden a sobreajustar. La regularización Ridge, que es la que estamos aplicando, sirve cuando se sopecha que varias de las variables de entrada están correlacionadas. Ridge hace que los coeficientes acaben siendo más pequeños. Esta disminución de los coeficientes minimiza el efecto de la correlación entre las variables predictoras y hace que el modelo generalice mejor. Ridge funciona mejor cuando la mayoría de las características son relevantes, lo cual resulta beneficioso para un modelo más generalizable, debido a que, para otros marcos de datos debemos mantener el supuesto de que todas las características son relevantes.

Por otro lado, la construcción automatizada de características, fue la técnica clave que nos permitió mejorar el rendimiento de la clasificación por regresión logística de forma más apreciable. A pesar de que la técnica Creación I (la cual se basa en encontrar regiones de alta densidad para los diagramas de dispersión generados), fue la que mostró el mejor desempeño en el conjunto PID, encontramos que es una técnica que depende de forma más directa de la estructura de datos, esto lo comprobamos al implementar la técnica Creación I en los datos Vanderbilt; en este caso, encontramos que debido a la dispersión de datos de este conjunto, era más conveniente tomar una división rectangular de $2x2$, lo cual es un inconveniente al querer que el modelo construido pueda ser generalizado a otros conjuntos de datos. Por otro lado, la técnica Creación III basada en el algoritmo Fuzzy C-means, es una técnica de creación que se puede generalizar de forma más directa a otros conjuntos de datos con variables numéricas, ya que a pesar de haber datos superpuestos en los diagramas de dispersion, el algoritmo fuzzy es capaz de brindar información relevante que permite mejorar de forma significativa la clasificación por regresión logística.

Al construir nuevas características usando el algoritmo de clustering fuzzy C-means, usando las 4 mejores características dadas por ANOVA F-Test, las características diseñadas fueron construidas en base a los siguientes diagramas de dispersión:

$N1 \longrightarrow$ Insuline vs Glucose

$N2 \longrightarrow$ Insuline vs BMI

$N3 \longrightarrow$ Insuline vs SkinThickness

$N4 \longrightarrow$ Glucose vs BMI

$N5 \longrightarrow$ Glucose vs SkinThickness

$N6 \longrightarrow$ BMI vs SkinThickness

En la **Figura 4.1**, se muestran los F-score al aplicar la prueba ANOVA F-Test a las 14 variables del marco de datos, para el conjunto de entrenamiento del primer pliegue generado por Stratified k-fold cross validation. En esta Figura, podemos observar que el método de creación automatizada de características, sí está generando características con una fuerte dependencia a la variable independiente.

| Feature_Name | Score |
|---|---|
| N1 | 568.090047 |
| N2 | 522.605866 |
| N3 | 514.010437 |
| 5. 2-Hour serum insulin (mu U/ml) | 235.239652 |
| 2. Plasma glucose concentration a 2 hours in a... | 198.586125 |
| N6 | 144.561109 |
| N5 | 136.132624 |
| N4 | 132.906344 |
| 6. Body mass index (weight in kg/(height in m)^2) | 68.722574 |
| 4. Triceps skin fold thickness (mm) | 61.025098 |
| 8. Age (years) | 44.470827 |
| 1. Number of times pregnant | 36.593110 |
| 7. Diabetes pedigree function | 28.884493 |
| 3. Diastolic blood pressure (mm Hg) | 22.705157 |

**Figura 4.1:** Selección de las mejores características sobre las 14 variables del marco de datos usando ANOVA F-Test, los F-score están ordenados de mayor a menor, donde un valor más alto del Score indica mayor capacidad discriminativa de la variable.

Por otro lado, en la **Tabla 14**, se muestra una compilación de métricas de evaluación para diferentes modelos aplicados al conjunto de datos PID, esto con el fin de observar de forma clara como se compara el Modelo Propuesto con otros modelos. En esta tabla, se han incluido el Modelo Base, Modelo 1, Modelo Propuesto, Modelo [59] (que hemos incluido debido a que es un modelo que emplea la creación de características, ver **sección 3.1**), Modelo (LR) [45] (se ha incluido este modelo de regresión logística (LR) debido a que los autores se quedan con una técnica de imputación por grupos utilizando la mediana, mientras que en nuestro caso usamos la imputación por grupos usando la media, ver **sección 3.1**) y Modelo (GB) [45] (este modelo lo hemos incluido debido a que los autores informan que con la imputación con la mediana por grupos el algoritmo Gradient Boosting (GB) es el que obtuvo el mejor rendimiento).

| Modelos | Accuracy | Precision | Recall | F1 score | ROC (AUC) |
|---|---|---|---|---|---|
| Modelo Base | 0.7538 | 0.7412 | 0.7459 | 0.7368 | 0.83 |
| Modelo 1 | 0.7473 | 0.7252 | 0.7080 | 0.7135 | 0.81 |
| Modelo Propuesto | 0.8490 | 0.8372 | 0.8503 | 0.8389 | 0.90 |
| Modelo [59] | 0.7532 | 0.73 | 0.72 | 0.73 | --- |
| Modelo (LR) [45] | 0.82 | 0.76 | 0.76 | 0.76 | 0.73 |
| Modelo (GB) [45] | 0.9106 | 0.86 | 0.87 | 0.87 | 0.85 |

**Tabla 14.** Recopilación de Métricas de evaluación de diferentes modelos.

Aclaramos que en el Modelo [59], las métricas proporcionadas son dadas en una sola división Train-Test con la proporción $70\,\% - 30\,\%$. Para el Modelo(LR) [45] y Modelo (GB) [45] los autores utilizan una validación cruzada de 10 veces para evaluar el rendimiento del conjunto de entrenamiento y para las métricas F1-score, Precision, and Recall consideran el promedio ponderado (weighted average). En nuestro caso, recordemos que estamos mostrando los valores promedios de estas métricas haciendo uso del promedio Macro (Macro average). Debido a que el promedio Macro calcula las puntuaciones (ya sea de Precision, Recall o F1-score) separado por clase pero sin usar pesos para la agregación, resulta en una penalización mayor cuando el modelo no funciona bien con las clases minoritarias (que es exactamente lo que se quiere cuando hay desequilibrio de clases), por otro lado, cuando se usa el promedio Weighted, se utiliza una ponderación que depende del número de etiquetas verdaderas de cada clase, por lo tanto, favoreciendo a la clase mayoritaria (que normalmente no es lo deseado).

En la **Tabla 14**, podemos observar que el Modelo Propuesto mejora significativamente el redimiento del clasificador de regresión logística respecto al Modelo Base. Observamos que el Modelo 1 no presenta una mejora respecto al Modelo Base. Por otro lado, el Modelo[59], el cual construye nuevas características de forma manual, no presenta unas métricas de rendimiento que mejoren la clasificación respecto a nuestro Modelo Base. El Modelo(LR)[45] si presenta un mejor rendimiento respecto al Modelo Base, sin embargo el Modelo Propuesto mantiene de forma clara un mejor desempeño en la clasificación. Finalmente podemos ver que el Modelo (GB) [45] tiene mejores métricas de evaluación, exceptuando el área bajo la curva ROC; en este caso, debido al desequilibrio de clases y que los autores utilizan una validación cruzada sin estratificar, es difícil saber si realmente su modelo tiene un mejor rendimiento, recordemos que cuando se aplica una validación cruzada sin estratificar en un conjunto de datos desbalanceados, es probable que uno o más pliegues tengan pocos o ningún ejemplo de la clase minoritaria y esto significa que algunas evaluaciones del modelo pueden ser

engañosas, ya que el modelo solo necesita predecir correctamente la clase mayoritaria. Dejando esto a un lado, se observa que de todas formas el modelo propuesto de regresión logística se mantiene en un rango comparable a un algoritmo de clasificación tan sofisticado como Gradient Boosting, el cual al combinar múltiples árboles, pierde interpretabilidad, lo cual puede no ser desable en problemas médicos.

Al evaluar el modelo predictivo construido (Modelo Propuesto), en el segundo conjunto de datos (Datos Vanderbilt), en la **Tabla 11**, podemos observar que la curva ROC no presenta ningún cambio, para una clasificación desequilibrada con un sesgo severo y pocos ejemplos de la clase minoritaria, el ROC AUC puede ser engañoso. Una alternativa común es la curva de Recall-Precision y el área bajo la curva. Una curva de Precision-Recall (o Curva PR) es un gráfico de la precisión (eje y) y la recuperación (eje x) para diferentes umbrales de probabilidad. Las curvas PR se recomiendan para dominios muy sesgados donde las curvas ROC pueden proporcionar una visión excesivamente optimista del rendimiento. El AUC de Precision-Recall es como el AUC de ROC, ya que resume la curva con un rango de valores de umbral como una sola puntuación. La puntuación se puede utilizar como punto de comparación entre diferentes modelos en un problema de clasificación binaria donde una puntuación de 1 representa un modelo con una habilidad perfecta.[66]

En este sentido, el **PRAUC** (área bajo la curva Precision-Recall) aplicado a los datos Vanderbilt, dio un valor promedio (promediado de los pliegues dados en Stratified k-fold cross-validation) de 0.7456 para el Modelo Base y un valor de 0.795 para el Modelo Propuesto, con esto podemos verificar de forma más fiable que con la curva ROC, que el Modelo Propuesto es un mejor modelo predictivo que el Modelo Base.

Como conclusiones y observaciones finales se enfatiza lo siguiente:

- Es importante aplicar en datos desbalanceados, el método Stratified k-fold cross validation, ya que este método garantiza que ningún valor esté sobrerepresentado o subrepresentado y de este modo obtener métricas de rendimiento más realistas.
- Debido a que el promedio Macro calcula las puntuaciones (ya sea de Precision, Recall o F1-score) separado por clase, pero sin usar pesos para la agregación, resulta en una penalización mayor cuando el modelo no funciona bien con las clases minoritarias (que es exactamente lo que se quiere cuando hay desequilibrio de clases), por otro lado, cuando se usa el promedio Weighted, se utiliza una ponderación que depende del número de etiquetas verdaderas de cada clase, por lo tanto, favoreciendo a la clase mayoritaria (que normalmente no es lo deseado).
- Es importante evitar, al aplicar las técnicas de preprocesamiento y transformación de datos, el problema de la fuga de datos, ya que esto puede resultar en una estimación incorrecta del rendimiento del modelo al realizar predicciones sobre

nuevos datos. Al aplicar cualquier método de tipo validación cruzada, es importante en la codificación, ser sumamente cuidadosos de que todas las técnicas de preprocesamiento o transformaciones de datos, respeten la independencia del conjunto de prueba; esto es, que en ningún pliegue, información del conjunto de prueba se filtre en la realización del modelo ajustado.

- La técnica de agrupación fuzzy C-means, es una técnica que permite extraer información útil al emplearla para construir características, usando agrupaciones sobre los diagramas de dispersión generados por las variables numéricas de un conjunto de datos, esta técnica tiene la ventaja de ser altamente generalizable a otros conjuntos de datos, esto al dar un buen rendimiento con datos superpuestos, con lo cual el presente trabajo ofrece una técnica sólida de creación automatizada de características.

- Se pudo observar que el rendimiento del clasificador de regresión logística puede ser mejorado aplicando técnicas de preprocesamiento y transformación de datos. La imputación, la limpieza de datos, la creación de características y el escalar de forma correcta los datos (tal como normalizar antes de PCA o aplicar la técnica min-max scaling al usar penalización en la regresión logística) permite mejorar de forma sustancial el rendimiento del modelo.

- Para la regresión logística penalizada y el algoritmo fuzzy C-means, fueron utilizados los hiperparámetros dados por default por las librerías de Python, mientras que para el método Z-score (para la identificación de valores atípicos) se tomó la regla general de tomar el puntaje $|Z|$ mayor a 3. Al querer construir un flujo de trabajo que pueda ser generalizable a otros conjuntos de datos, es importante basarse en suposiciones genéricas para evitar el sobreajuste del modelo, ya que de otro modo se corre el riesgo de crear un flujo que solo funcione sobre el conjunto de datos sobre el cual se está construyendo.

- Una vía futura a explorar pueden ser técnicas automatizadas de ajuste de hiperparámetros que permitan mejorar el rendimiento del modelo propuesto, evitando la fuga de datos.

- Todo el flujo de trabajo propuesto es un proceso automático basado en suposiciones genéricas, exceptuando la aplicación del Análisis de Componentes Principales, en el cual el número de componentes fue seleccionado de forma manual, de este modo, otro camino a explorar pueden ser técnicas automatizadas para la selección de componentes principales.

Apéndice A

# Métricas completas de evaluación

En este apartado mostraremos las métricas de evaluación completas de los modelos predictivos diseñados en el proyecto.

La presentación de estas métricas de evaluación quedará estructurada de la siguiente forma:

- Para cada modelo construido se presentará una Tabla donde se incluirán las métricas: **Precisión**, **Recall** y **F1 score**, estas métricas se mostrarán separadas por clases (donde la clase 0 indica sano y la clase 1 diabetes) también se informarán el promedio Macro (**Macro Avg**) y el promedio ponderado (**Weighted Avg**). Abajo de la Tabla principal se anexará una tabla informando la precisión mínima (**Minimum Accuracy**), la precisión máxima (Maximum Accuracy), la precisión promedio o general (Overall Accuracy) y finalmente el tiempo de ejecución (**Execution Time**).
- Por otro lado, para cada modelo, en una misma figura se incluirán dos imágenes, donde en la imagen izquierda se mostrará la matriz de confusión promedio y en la imágen derecha la curva ROC promedio.

En la **sección 3.4.1**, en la **Tabla 7** y **Figura 3.2**, se explica a detalle las Métricas de evaluación y las figuras de la Matriz de confusión y Curva ROC del Modelo Base, esta explicación más detallada es análoga y generalizable para interpretar las Tablas y Figuras presentadas en este Apéndice.

## A.1. Modelo Base

| | Precision (Precision ± std) | Recall (Recall ± std) | F1 score (F1 score ± std) |
|---|---|---|---|
| 0 | 0.8390 ± 0.0480 | 0.7720 ± 0.0976 | 0.8010 ± 0.0591 |
| 1 | 0.6434 ± 0.1123 | 0.7198 ± 0.0990 | 0.6725 ± 0.0754 |
| Macro Avg. | 0.7412 ± 0.0689 | 0.7459 ± 0.0617 | 0.7368 ± 0.0648 |
| Weighted Avg. | 0.7708 ± 0.0583 | 0.7538 ± 0.0655 | 0.7562 ± 0.0623 |

| Minimum Accuracy | Maximum Accuracy | Overall Accuracy (Accuracy ± std) | Execution Time (t ± std) s |
|---|---|---|---|
| 0.6579 | 0.8571 | 0.7538 ± 0.0655 | 0.00981 ± 0.00081 |

**Tabla A.1.** Métricas de evaluación del Modelo Base.

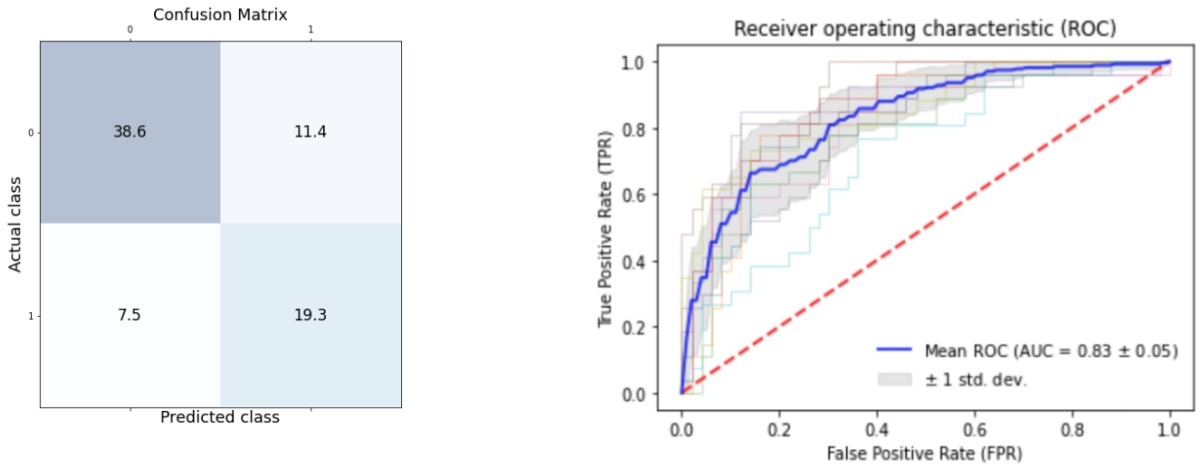



**Figura A.1:** Matriz de confusión y curva ROC del Modelo Base.

## A.2. Modelo 1

| | Precision (Precision ± std) | Recall (Recall ± std) | F1 score (F1 score ± std) |
|---|---|---|---|
| 0 | 0.7875 ± 0.0326 | 0.8380 ± 0.0577 | 0.8115 ± 0.0403 |
| 1 | 0.6629 ± 0.1104 | 0.5779 ± 0.0717 | 0.6155 ± 0.0788 |
| Macro Avg. | 0.7252 ± 0.0693 | 0.7080 ± 0.0545 | 0.7135 ± 0.0586 |
| Weighted Avg. | 0.7441 ± 0.0572 | 0.7473 ± 0.0526 | 0.7432 ± 0.0525 |

| Minimum Accuracy | Maximum Accuracy | Overall Accuracy (Accuracy ± std) | Execution Time (t ± std) s |
|---|---|---|---|
| 0.6711 | 0.8442 | 0.7473 ± 0.0526 | 1.6739 ± 0.049 |

**Tabla A.2.** Métricas de evaluación del Modelo 1.

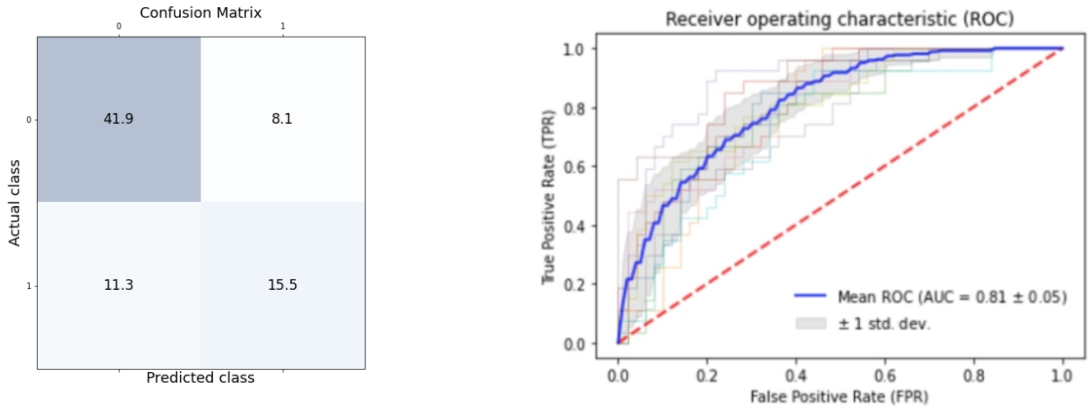



**Figura A.2:** Matriz de confusión y curva ROC del Modelo 1.

## A.3. Técnicas de Imputación

### A.3.1. Imputación I (Mean Imputation)

| | Precision (Precision ± std) | Recall (Recall ± std) | F1 score (F1 score ± std) |
|---|---|---|---|
| 0 | 0.8357 ± 0.0431 | 0.7780 ± 0.0977 | 0.8031 ± 0.0594 |
| 1 | 0.6485 ± 0.1194 | 0.7124 ± 0.0910 | 0.6721 ± 0.0758 |
| Macro Avg. | 0.7421 ± 0.0716 | 0.7452 ± 0.0608 | 0.7376 ± 0.0653 |
| Weighted Avg. | 0.7704 ± 0.0592 | 0.7552 ± 0.0659 | 0.7574 ± 0.0628 |

| Minimum Accuracy | Maximum Accuracy | Overall Accuracy (Accuracy ± std) | Execution Time (t ± std) s |
|---|---|---|---|
| 0.6711 | 0.8442 | 0.7552 ± 0.0659 | 0.025 ± 0.002 |

**Tabla A.3.** Métricas de evaluación de Imputación I.

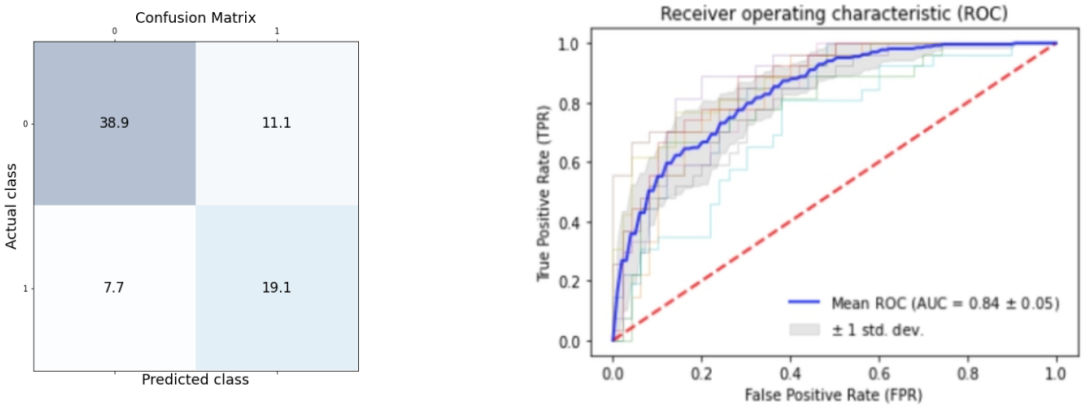



**Figura A.3:** Matriz de confusión y curva ROC de Imputación I.

### A.3.2. Imputación II (Group Mean Imputation)

| | Precision (Precision ± std) | Recall (Recall ± std) | F1 score (F1 score ± std) |
|---|---|---|---|
| 0 | 0.8601 ± 0.0371 | 0.7940 ± 0.0833 | 0.8237 ± 0.0499 |
| 1 | 0.6757 ± 0.1067 | 0.7570 ± 0.0772 | 0.7088 ± 0.0662 |
| Macro Avg. | 0.7679 ± 0.0623 | 0.7755 ± 0.0522 | 0.7663 ± 0.0565 |
| Weighted Avg. | 0.7958 ± 0.0508 | 0.7812 ± 0.0565 | 0.7837 ± 0.0540 |

| Minimum Accuracy | Maximum Accuracy | Overall Accuracy (Accuracy ± std) | Execution Time (t ± std) s |
|---|---|---|---|
| 0.6974 | 0.8701 | 0.7812 ± 0.0565 | 0.035 ± 0.002 |

**Tabla A.4.** Métricas de evaluación de Imputación II.

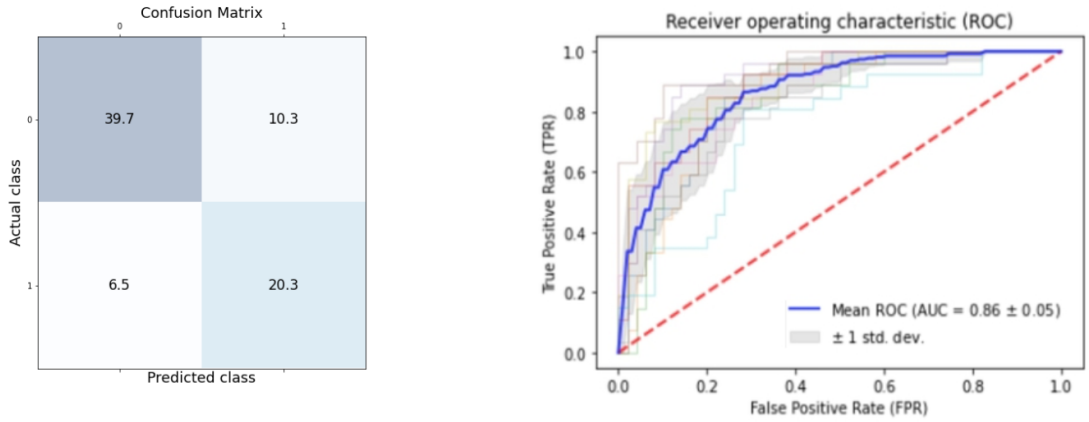



**Figura A.4:** Matriz de confusión y curva ROC de Imputación II.

### A.3.3. Imputación III (Regression Imputation)

| | Precision (Precision ± std) | Recall (Recall ± std) | F1 score (F1 score ± std) |
|---|---|---|---|
| 0 | 0.8337 ± 0.0412 | 0.7760 ± 0.0961 | 0.8010 ± 0.0568 |
| 1 | 0.6450 ± 0.1162 | 0.7085 ± 0.0891 | 0.6683 ± 0.0709 |
| Macro Avg. | 0.7393 ± 0.0685 | 0.7423 ± 0.0569 | 0.7347 ± 0.0614 |
| Weighted Avg. | 0.7679 ± 0.0562 | 0.7526 ± 0.0625 | 0.7548 ± 0.0593 |

| Minimum Accuracy | Maximum Accuracy | Overall Accuracy (Accuracy ± std) | Execution Time (t ± std) s |
|---|---|---|---|
| 0.6711 | 0.8442 | 0.7526 ± 0.0625 | 0.161 ± 0.055 |

**Tabla A.5.** Métricas de evaluación de Imputación III.

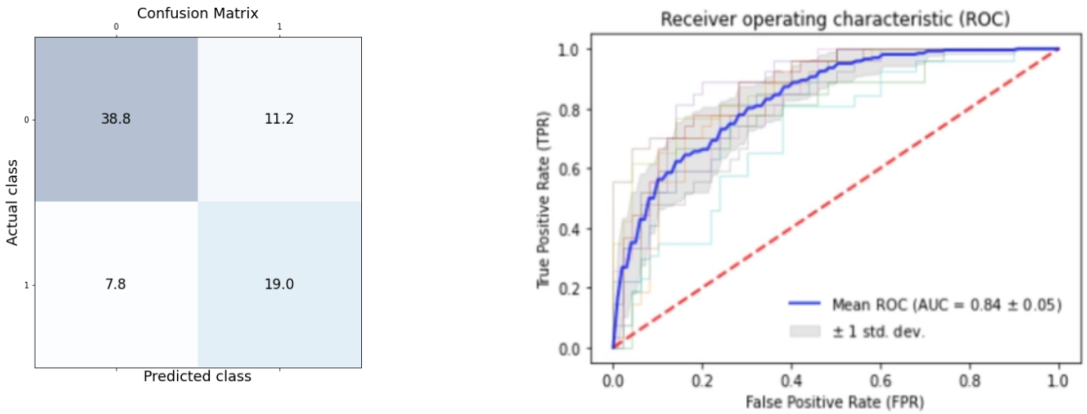



**Figura A.5:** Matriz de confusión y curva ROC de Imputación III.

## A.4. Técnicas de Limpieza de Datos

### A.4.1. Limpieza I

| | Precision (Precision ± std) | Recall (Recall ± std) | F1 score (F1 score ± std) |
|---|---|---|---|
| 0 | 0.8734 ± 0.0310 | 0.8320 ± 0.0732 | 0.8504 ± 0.0382 |
| 1 | 0.7240 ± 0.1009 | 0.7722 ± 0.0670 | 0.7419 ± 0.0491 |
| Macro Avg. | 0.7987 ± 0.0528 | 0.8021 ± 0.0378 | 0.7961 ± 0.0426 |
| Weighted Avg. | 0.8213 ± 0.0405 | 0.8112 ± 0.0433 | 0.8125 ± 0.0409 |

| Minimum Accuracy | Maximum Accuracy | Overall Accuracy (Accuracy ± std) | Execution Time (t ± std) s |
|---|---|---|---|
| 0.75 | 0.8701 | 0.8112 ± 0.0433 | 2.986 ± 0.176 |

**Tabla A.6.** Métricas de evaluación de Limpieza I.

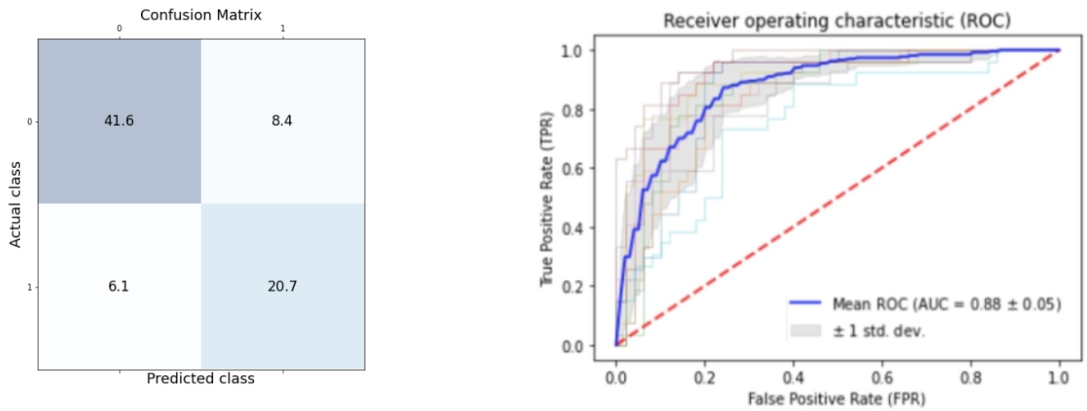



**Figura A.6:** Matriz de confusión y curva ROC de Limpieza I.

### A.4.2. Limpieza II

| | Precision (Precision ± std) | Recall (Recall ± std) | F1 score (F1 score ± std) |
|---|---|---|---|
| 0 | 0.8560 ± 0.0429 | 0.7940 ± 0.0783 | 0.8215 ± 0.0444 |
| 1 | 0.6705 ± 0.0938 | 0.7459 ± 0.0939 | 0.7006 ± 0.0637 |
| Macro Avg. | 0.7633 ± 0.0557 | 0.7699 ± 0.0504 | 0.7610 ± 0.0516 |
| Weighted Avg. | 0.7913 ± 0.0473 | 0.7773 ± 0.0507 | 0.7793 ± 0.0487 |

| Minimum Accuracy | Maximum Accuracy | Overall Accuracy (Accuracy ± std) | Execution Time (t ± std) s |
|---|---|---|---|
| 0.6974 | 0.8571 | 0.7773 ± 0.0507 | 0.058 ± 0.004 |

**Tabla A.7.** Métricas de evaluación de Limpieza II.

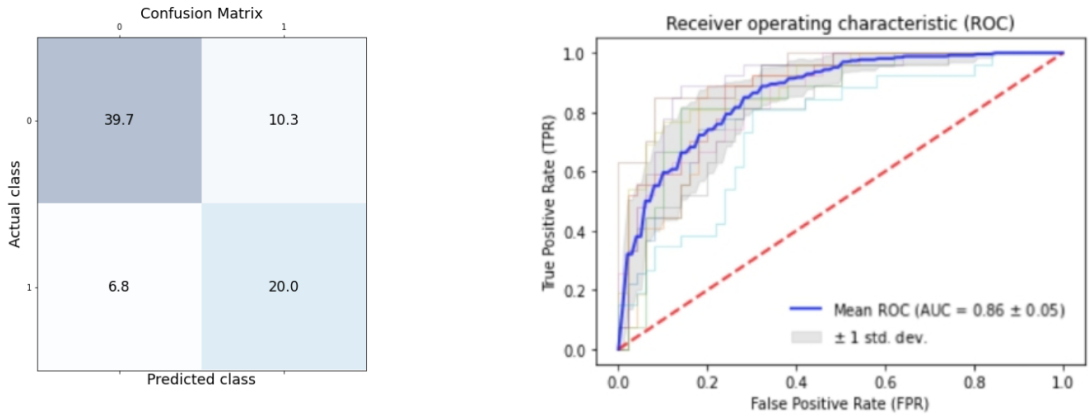



**Figura A.7:** Matriz de confusión y curva ROC de Limpieza II.

### A.4.3. Limpieza III

| | Precision (Precision ± std) | Recall (Recall ± std) | F1 score (F1 score ± std) |
|---|---|---|---|
| 0 | 0.8928 ± 0.0227 | 0.7980 ± 0.0745 | 0.8412 ± 0.0438 |
| 1 | 0.6945 ± 0.0850 | 0.8208 ± 0.0427 | 0.7492 ± 0.0490 |
| Macro Avg. | 0.7936 ± 0.0468 | 0.8094 ± 0.0386 | 0.7952 ± 0.0459 |
| Weighted Avg. | 0.8236 ± 0.0364 | 0.8060 ± 0.0474 | 0.8091 ± 0.0452 |

| Minimum Accuracy | Maximum Accuracy | Overall Accuracy (Accuracy ± std) | Execution Time (t ± std) s |
|---|---|---|---|
| 0.75 | 0.8701 | 0.8060 ± 0.0474 | 0.035 ± 0.005 |

**Tabla A.8.** Métricas de evaluación de Limpieza III.

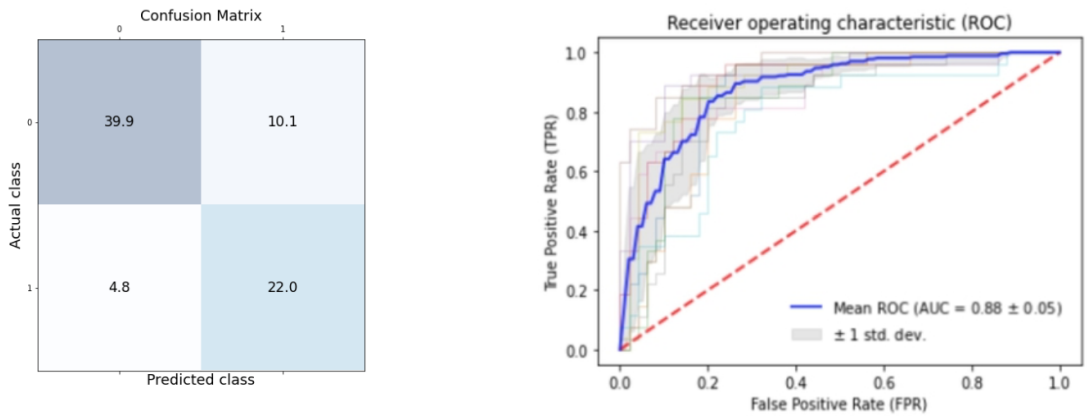



**Figura A.8:** Matriz de confusión y curva ROC de Limpieza III.

## A.5. Técnicas de Creación Automatizada de Características

### A.5.1. Creación I

| | Precision (Precision ± std) | Recall (Recall ± std) | F1 score (F1 score ± std) |
|---|---|---|---|
| 0 | 0.9057 ± 0.0309 | 0.8520 ± 0.0492 | 0.8770 ± 0.0280 |
| 1 | 0.7557 ± 0.0637 | 0.8323 ± 0.0630 | 0.7897 ± 0.0426 |
| Macro Avg. | 0.8307 ± 0.0361 | 0.8422 ± 0.0339 | 0.8333 ± 0.0346 |
| Weighted Avg. | 0.8534 ± 0.0308 | 0.8451 ± 0.0332 | 0.8465 ± 0.0324 |

| Minimum Accuracy | Maximum Accuracy | Overall Accuracy (Accuracy ± std) | Execution Time (t ± std) s |
|---|---|---|---|
| 0.7792 | 0.8831 | 0.8451 ± 0.0332 | 1.189 ± 0.031 |

**Tabla A.9.** Métricas de evaluación de Creación I.

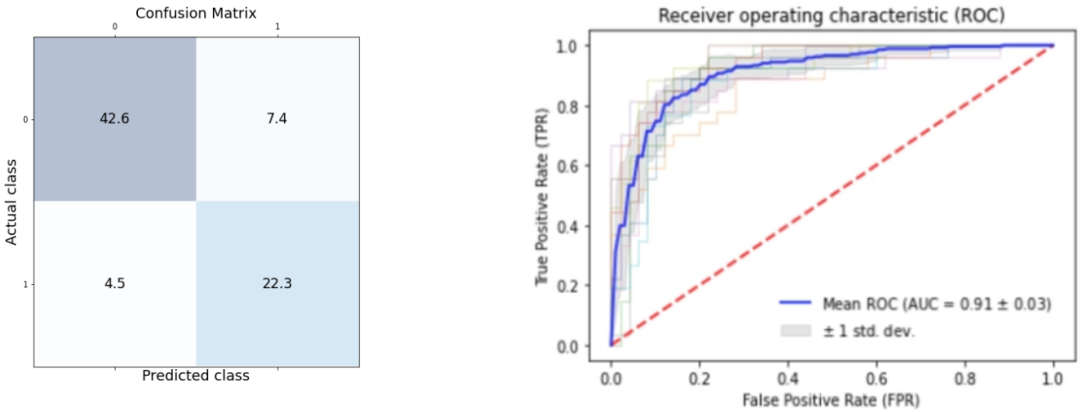



**Figura A.9:** Matriz de confusión y curva ROC de Creación I.

### A.5.2. Creación II

| | Precision (Precision ± std) | Recall (Recall ± std) | F1 score (F1 score ± std) |
|---|---|---|---|
| 0 | 0.7290 ± 0.0842 | 0.5460 ± 0.2480 | 0.5992 ± 0.1880 |
| 1 | 0.4657 ± 0.1082 | 0.6422 ± 0.1625 | 0.5171 ± 0.0641 |
| Macro Avg. | 0.5973 ± 0.0851 | 0.5941 ± 0.0798 | 0.5581 ± 0.1104 |
| Weighted Avg. | 0.6372 ± 0.0824 | 0.5796 ± 0.1223 | 0.5706 ± 0.1325 |

| Minimum Accuracy | Maximum Accuracy | Overall Accuracy (Accuracy ± std) | Execution Time (t ± std) s |
|---|---|---|---|
| 0.3766 | 0.7143 | 0.5796 ± 0.1223 | 3.024 ± 0.133 |

**Tabla A.10.** Métricas de evaluación de Creación II.

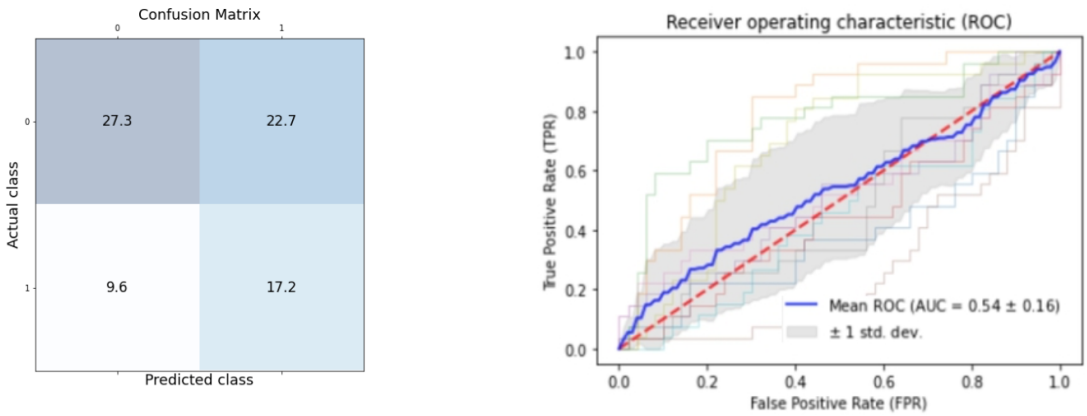



**Figura A.10:** Matriz de confusión y curva ROC de Creación II.

### A.5.3. Creación III

| | Precision (Precision ± std) | Recall (Recall ± std) | F1 score (F1 score ± std) |
|---|---|---|---|
| 0 | 0.9077 ± 0.0298 | 0.8380 ± 0.0786 | 0.8696 ± 0.0452 |
| 1 | 0.7466 ± 0.0986 | 0.8396 ± 0.0551 | 0.7866 ± 0.0584 |
| Macro Avg. | 0.8271 ± 0.0544 | 0.8388 ± 0.0437 | 0.8281 ± 0.0512 |
| Weighted Avg. | 0.8514 ± 0.0431 | 0.8386 ± 0.0512 | 0.8406 ± 0.0493 |

| Minimum Accuracy | Maximum Accuracy | Overall Accuracy (Accuracy ± std) | Execution Time (t ± std) s |
|---|---|---|---|
| 0.7662 | 0.9091 | 0.8386 ± 0.0512 | 0.539 ± 0.025 |

**Tabla A.11.** Métricas de evaluación de Creación III.

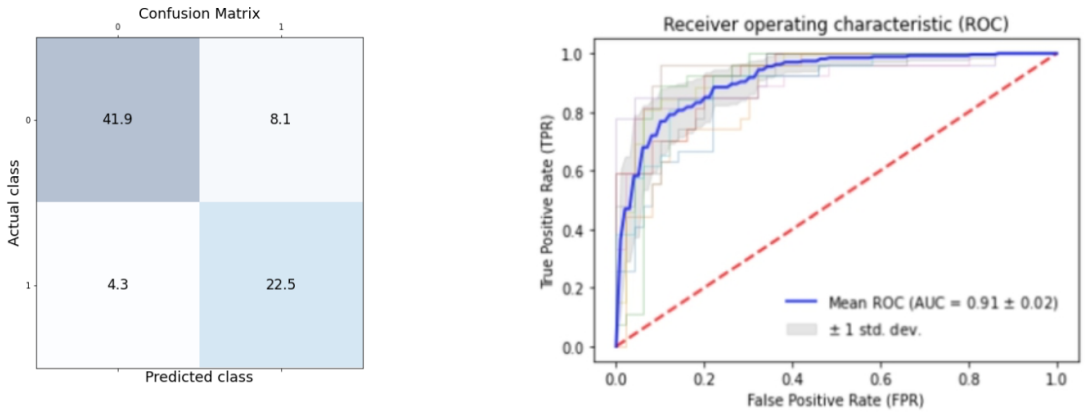



**Figura A.11:** Matriz de confusión y curva ROC de Creación III.

## A.6. Modelo ANOVA F-Test + Creación III

| | Precision (Precision ± std) | Recall (Recall ± std) | F1 score (F1 score ± std) |
|---|---|---|---|
| 0 | 0.9083 ± 0.0360 | 0.8420 ± 0.0769 | 0.8721 ± 0.0454 |
| 1 | 0.7511 ± 0.1032 | 0.8396 ± 0.0676 | 0.7888 ± 0.0640 |
| Macro Avg. | 0.8297 ± 0.0586 | 0.8408 ± 0.0485 | 0.8305 ± 0.0541 |
| Weighted Avg. | 0.8535 ± 0.0475 | 0.8411 ± 0.0530 | 0.8431 ± 0.0514 |

| Minimum Accuracy | Maximum Accuracy | Overall Accuracy (Accuracy ± std) | Execution Time (t ± std) s |
|---|---|---|---|
| 0.7532 | 0.9221 | 0.8411 ± 0.0530 | 0.130 ± 0.007 |

**Tabla A.12.** Métricas de evaluación del modelo ANOVA F-Test + Creación III.

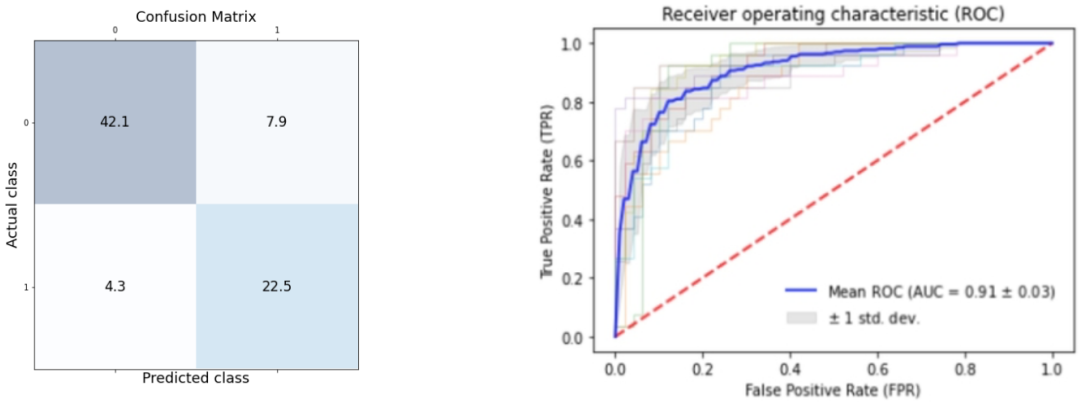



**Figura A.12:** Matriz de confusión y curva ROC del modelo ANOVA F-Test + Creación III.

## A.7. Modelo ANOVA F-test + Creación III + PCA

| | Precision (Precision ± std) | Recall (Recall ± std) | F1 score (F1 score ± std) |
|---|---|---|---|
| 0 | 0.9168 ± 0.0350 | 0.8460 ± 0.0712 | 0.8783 ± 0.0411 |
| 1 | 0.7576 ± 0.0885 | 0.8546 ± 0.0639 | 0.7996 ± 0.0552 |
| Macro Avg. | 0.8372 ± 0.0496 | 0.8503 ± 0.0428 | 0.8389 ± 0.0477 |
| Weighted Avg. | 0.8613 ± 0.0406 | 0.8490 ± 0.0472 | 0.8508 ± 0.0456 |

| Minimum Accuracy | Maximum Accuracy | Overall Accuracy (Accuracy ± std) | Execution Time (t ± std) s |
|---|---|---|---|
| 0.7662 | 0.8961 | 0.8490 ± 0.0472 | 0.135 ± 0.008 |

**Tabla A.13.** Métricas de evaluación del modelo ANOVA F-test + Creación III + PCA.

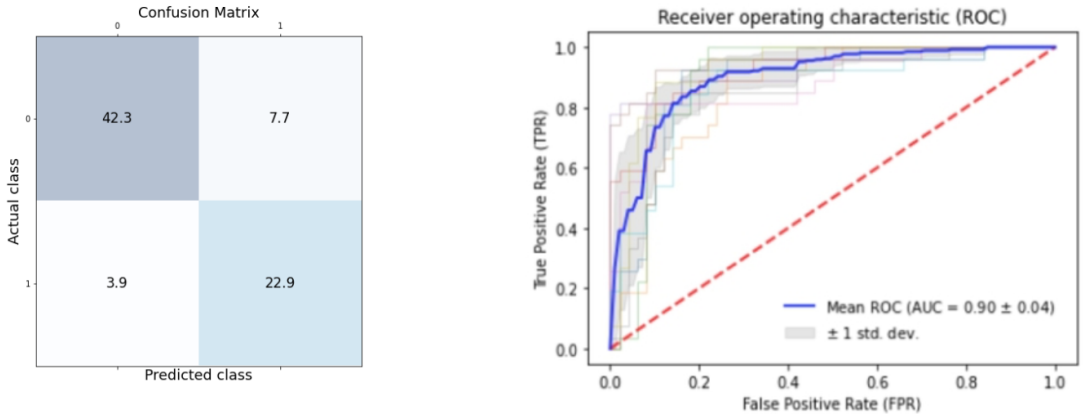



**Figura A.13:** Matriz de confusión y curva ROC del modelo ANOVA F-test + Creación III + PCA.

## A.8. Métricas completas de evaluación de los modelos aplicados al conjunto de datos Vanderbilt

### A.8.1. Modelo Base

| | Precision (Precision ± std) | Recall (Recall ± std) | F1 score (F1 score ± std) |
|---|---|---|---|
| 0 | 0.9619 ± 0.0157 | 0.8970 ± 0.0518 | 0.9273 ± 0.0211 |
| 1 | 0.6103 ± 0.1178 | 0.8000 ± 0.0950 | 0.6801 ± 0.0384 |
| Macro Avg. | 0.7861 ± 0.0523 | 0.8485 ± 0.0250 | 0.8037 ± 0.0293 |
| Weighted Avg. | 0.9078 ± 0.0096 | 0.8821 ± 0.0306 | 0.8893 ± 0.0235 |

| Minimum Accuracy | Maximum Accuracy | Overall Accuracy (Accuracy ± std) | Execution Time (t ± std) s |
|---|---|---|---|
| 0.8462 | 0.9231 | 0.8821 ± 0.0306 | 0.012 ± 0.003 |

**Tabla A.14.** Métricas de evaluación del Modelo Base aplicado al conjunto de datos Vanderbilt.

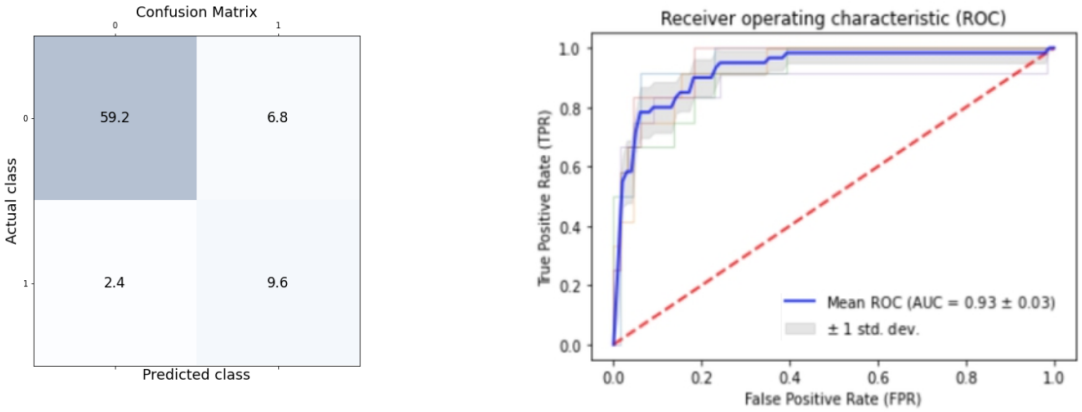



**Figura A.14:** Matriz de confusión y curva ROC del Modelo Base aplicado al conjunto de datos Vanderbilt.

### A.8.2. Modelo Propuesto

| | Precision (Precision ± std) | Recall (Recall ± std) | F1 score (F1 score ± std) |
|---|---|---|---|
| 0 | 0.9751 ± 0.0229 | 0.9091 ± 0.0429 | 0.9401 ± 0.0183 |
| 1 | 0.6460 ± 0.0702 | 0.8667 ± 0.1264 | 0.7329 ± 0.0553 |
| Macro Avg. | 0.8105 ± 0.0310 | 0.8879 ± 0.0506 | 0.8365 ± 0.0349 |
| Weighted Avg. | 0.9245 ± 0.0168 | 0.9026 ± 0.0266 | 0.9082 ± 0.0224 |

| Minimum Accuracy | Maximum Accuracy | Overall Accuracy (Accuracy ± std) | Execution Time (t ± std) s |
|---|---|---|---|
| 0.859 | 0.9231 | 0.9026 ± 0.0266 | 0.349 ± 0.063 |

**Tabla A.15.** Métricas de evaluación del Modelo Propuesto aplicado al conjunto de datos Vanderbilt.

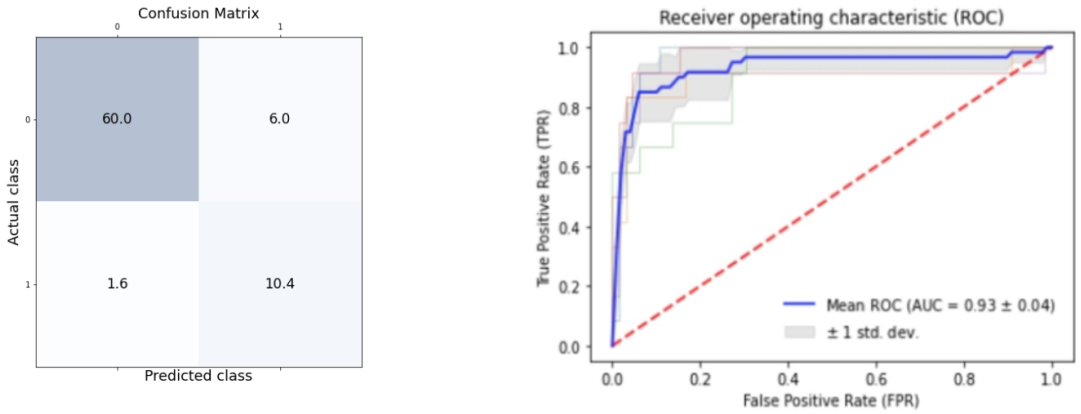



**Figura A.15:** Matriz de confusión y curva ROC del Modelo Propuesto aplicado al conjunto de datos Vanderbilt.

# Bibliografía


[1] Antonio Lozano, José. *Diabetes Mellitus*. Offarm, Vol.25, Núm.10, 66-78. España. Elsevier. 2006. 1

[2] Oficina de Prensa y Colaboradores. *LA PANDEMIA DE DIABETES EN MÉXICO*. CIAD. 2020. Consultado el 15 de Marzo de 2022. https://www.ciad.mx/notas/item/2450-la-pandemia-de-diabetes-en-mexico 1, 2

[3] M.M. Lima-Martínez, C. Carrera Boada, M.D. Madera-Silva, W. Marín, M. Contreras. *COVID-19 and diabetes: A bidirectional relationship*. Clínica e Investigación en Arteriosclerosis (English Edition), Volume 33, Issue 3, Pages 151-157. May–June 2021. 2

[4] Li, Xiaohua et al. *Improving the Accuracy of Diabetes Diagnosis Applications through a Hybrid Feature Selection Algorithm*. Neural processing letters, 1-17. 27 Mar, 2021. 2

[5] Zou, Quan et al. *Predicting Diabetes Mellitus With Machine Learning Techniques*. Frontiers in genetics, Vol.9, 515. 6 Nov, 2018. 3

[6] Changsheng Zhu, Christian Uwa Idemudia, Wenfang Feng. *Improved logistic regression model for diabetes prediction by integrating PCA and K-means techniques*. Informatics in Medicine Unlocked, Science Direct, April 2019. 4, 70, 71, 75, 78, 95

https://doi.org/10.1016/j.imu.2019.100179

[7] Nilashi M, Bin Ibrahim O, Mardani A, Ahani A, Jusoh A. *A soft computing approach for diabetes disease classification*. Health Informatics Journal. December 2018:379-393. 3

[8] Wikipedia, Wikimedia Foundation. *Data science*. Consultado el 15 de Marzo de 2022. 5

https://en.wikipedia.org/wiki/Data_science

[9] Brian Everitt, Torsten Hothorn. *An Introduction to Applied Multivariate Analysis with R*. Springer Science & Business Media. 2011. 5, 7, 8, 11

[10] Géron Aurélien. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. 2nd Edition. O'Reilly Media. 2019. 43, 44

[11] Dobson, A.J. and Barnett, A.G. *An Introduction to Generalized Linear Models*. 3rd Edition, CRC Press, Chapman & Hall, Boca Raton. 2008. 7, 8

[12] Velleman, P. and Wilkinson, L. *Nominal, ordinal, interval, and ratio typologies are misleading*. The American Statistician, 47, 65–72. 8

[13] Carter, Nathan (book editor). *Data Science for Mathematicians*. First edition. Chapman & Hall/CRC The R Series. 2020. 5, 8, 9, 10, 12, 15, 16, 17, 18, 20, 34

[14] Melcher, Kathrin. Silipo, Rosaria. *Missing Value Imputation–A Review*. KDnuggets. Consultado el 15 de Marzo de 2022. https://www.kdnuggets.com/2020/09/missing-value-imputation-review.html 12

[15] Jaemun Sim, Jonathan Sangyun Lee, Ohbyung Kwon. *Missing Values and Optimal Selection of an Imputation Method and Classification Algorithm to Improve the Accuracy of Ubiquitous Computing Applications*. Mathematical Problems in Engineering, vol. 2015, Article ID 538613. 2015. 13

[16] Bowen Chieh-Chen. *Straightforward Statistics*. SAGE Publications. 2015. 14, 15

[17] *Preprocessing data*. scikit-learn. Consultado el 15 de Marzo de 2022. 17
https://scikit-learn.org/stable/modules/preprocessing.html#preprocessing-scaler

[18] *Importance of Feature Scaling*. scikit-learn. Consultado el 15 de Marzo de 2022. 17
https://scikit-learn.org/stable/auto_examples/preprocessing/plot_scaling_importance.html

[19] Hastie, Trevor, Robert Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. New York: Springer, 2009. 18

[20] Brownlee Jason. *Data Preparation for Machine Learning: Data Cleaning, Feature Selection, and Data Transforms in Python*. Machine Learning Mastery. 2020. 18, 19, 21, 22, 23

[21] Elssied, Nadir Omer Fadl, Othman Ibrahim and Ahmed Hamza Osman. *A Novel Feature Selection Based on One-Way ANOVA F-Test for E-Mail Spam Classification*. Research Journal of Applied Sciences, Engineering and Technology7, 625-638. 2014. 20

[22] Zheng Alice, Casari Amanda. *Feature Engineering for Machine Learning*. First Edition. O'Reilly Media. 2014. 21

[23] Sadanori Konishi. *Introduction to Multivariate Analysis: Linear and Nonlinear Modeling*. CRC Press. 2014. 23, 28, 29, 32, 33

[24] Bansal, A.K., Khan, J.I., & Alam, S.K. (Eds). *Introduction to Computational Health Informatics*. 1st ed. Chapman and Hall/CRC. 2020. 34

[25] Wikipedia, Wikimedia Foundation. *k-means clustering*. Consultado el 15 de Marzo de 2022. 35

https://en.wikipedia.org/wiki/K-means_clustering

[26] Alboukadel Kassambara. *Practical Guide to Cluster Analysis in R: Unsupervised Machine Learning*. 1st ed. STHDA. 2017. 35

[27] Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, and Vipin Kumar. *Introduction to Data Mining*. 1st ed. Pearson. 2014. 35, 36, 37, 61

[28] Steorts Rebecca. *K-means Clustering*. Duke University. Consultado el 15 de Marzo de 2022. 37

http://www2.stat.duke.edu/∼rcs46/lectures_2017/10-unsupervise/10-kmeans_v2.pdf

[29] *Clustering*. UC Davis. Consultado el 15 de Marzo de 2022. 37, 39, 40

https://www.math.ucdavis.edu/∼strohmer/courses/180BigData/180lecture_kmeans.pdf

[30] Charu & Reddy, Chandan (book editors). *DATA CLUSTERING Algorithms and Applications*. Chapman and Hall/CRC. 2014. 39, 41

[31] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press. 2014. 40

[32] Shuyang Ling. *k-means Clustering*. NYU. Consultado el 15 de Marzo de 2022. 40

https://cims.nyu.edu/∼sling/MATH-SHU-236-2020-SPRING/MATH-SHU-236-Lecture-6-kmeans.pdf

[33] Hennig, C., Meila, M., Murtagh, F., & Rocci, R. (Eds). *Handbook of Cluster Analysis*. 1st ed. Chapman and Hall/CRC. 2015. 41, 42

[34] Veit Schwämmle, Ole Nørregaard Jensen. *A simple and fast method to determine the parameters for fuzzy c–means cluster analysis*. Bioinformatics, Volume 26, Issue 22, Pages 2841–2848. November 2010. 42

[35] Andrew Y. Ng and Michael I. Jordan. *On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes*. In Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic (NIPS'01). MIT Press, Cambridge, MA, USA, 841–848. 2001. 44, 45

[36] Soner Yildirim. *Generative vs Discriminative Classifiers in Machine Learning*. Towards Data Science. Consultado el 15 de Marzo de 2022. 45

https://towardsdatascience.com/generative-vs-discriminative-classifiers-in-machine-learning-9ee265be859e

[37] Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. 3rd Edition. Prentice-Hall, Inc. USA. 2020. 46, 48, 49, 51, 52, 53

[38] Jean-Christophe B. Loiseau. *Binary cross-entropy and logistic regression*. Towards Data Science. Consultado el 15 de Marzo de 2022. 51

https://towardsdatascience.com/binary-cross-entropy-and-logistic-regression-bf7098e75559

[39] *sklearn.linear_model.LogisticRegression*. scikit-learn. Consultado el 15 de Marzo de 2022. 51

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[40] Jiawei Han, Micheline Kamber, Jian Pei. *Data Mining: Concepts and Techniques*. Third Edition. Morgan Kaufmann, Elsevier. 2012. 53, 55, 57, 59, 60, 61, 62, 63, 64

[41] Wikipedia, Wikimedia Foundation. *Cross-validation (statistics)*. Consultado el 15 de Marzo de 2022. 63

https://en.wikipedia.org/wiki/Cross-validation_(statistics)

[42] Towards Data Science. *Cross Validation Explained: Evaluating estimator performance*. Consultado el 15 de Marzo de 2022. 64

https://towardsdatascience.com/cross-validation-explained-evaluating-estimator-performance-e51e5430ff85

[43] Géron Aurélien. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. 2nd Edition. O'Reilly Media. 2019. 43, 44

[44] Müller, Andreas C., and Sarah Guido. *Introduction to machine learning with Python*. First Edition. O'Reilly Media. 2017. 65

[45] Kumarmangal Roy, Muneer Ahmad, Kinza Waqar, Kirthanaah Priyaah, Jamel Nebhen, Sultan S Alshamrani, Muhammad Ahsan Raza, Ihsan Ali. *An Enhanced Machine Learning Framework for Type 2 Diabetes Classification Using Imbalanced Data with Missing Values*. Complexity, vol. 2021, Article ID 9953314. 2021. 68, 69, 97, 98

https://doi.org/10.1155/2021/9953314

[46] E. Guldogan, Z. Tunc, A. Acet, and C. Colak. *Performance evaluation of different artificial neural network models in the classification of type 2 diabetes mellitus*. The Journal of Cognitive Systems, vol. 5, no. 1, pp. 5–9. 2020. 68

[47] T. M. Alam, M. A. Iqbal, Y. Ali et al. *A model for early prediction of diabetes*. Informatics in Medicine Unlocked, vol. 16. 2019. 68

[48] Q. Wang, W. Cao, J. Guo, J. Ren, Y. Cheng, and D. N. Davis. *DMP_MI: an effective diabetes mellitus classification algorithm on imbalanced data with missing values*. IEEE Access, vol. 7, pp. 102232–102238. 2019. 68

[49] F. G. Woldemichael and S. Menaria. *Prediction of diabetes using data mining techniques*. Proceedings of the International Conference on Trends in Electronics and Informatics (ICOEI), pp. 414–418, IEEE, Tirunelveli, India. May 2018. 68

[50] H. Wu, S. Yang, Z. Huang, J. He, and X. Wang. *Type 2 diabetes mellitus prediction model based on data mining*. Informatics in Medicine Unlocked, vol. 10, pp. 100–107. 2018. 68, 69, 70, 78

https://doi.org/10.1016/j.imu.2017.12.006

[51] R. Vaishali, R. Sasikala, S. Ramasubbareddy, S. Remya, and S. Nalluri. *Genetic algorithm based feature selection and MOE Fuzzy classification algorithm on Pima Indians Diabetes dataset*. Proceedings of the International Conference on Computing Networking and Informatics (ICCNI), pp. 1–5, IEEE, Lagos, Nigeria. October 2017. 68

[52] A. A. AlJarullah. *Decision tree discovery for the diagnosis of type II diabetes*. Proceedings of the 2011 International Conference on Innovations in Information Technology, pp. 303–307, Abu Dhabi, UAE. April 2011. 68

[53] A. Marcano-Cedeño, J. Torres, and D. Andina. *A prediction model to diabetes using Artificial Metaplasticity*. New Challenges on Bioinspired Applications, pp. 418–425, Springer, Berlin, Germany. 2011. 68

[54] B. M. Patil, R. C. Joshi, and D. Toshniwal. *Hybrid prediction model for Type-2 diabetic patients*. Expert Systems with Applications, vol. 37, no. 12, pp. 8102–8108. 2010. 68, 69, 70, 78

https://doi.org/10.1016/j.eswa.2010.05.078

[55] J. Han, J. C. Rodriguze, and M. Beheshti. *Diabetes data analysis and prediction model discovery using RapidMiner*. Proceedings of the 2008 Second International Conference on Future Generation Communication and Networking, pp. 96–99, IEEE, Hainan, China. December 2008. 68

[56] Christodoulou, E.; Ma, J.; Collins, G.S.; Steyerberg, E.W.; Verbakel, J.Y.; Van Calster, B. *A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models*. J. Clin. Epidemiol, 110, 12–22. 2019. 69

[57] Nusinovici, S.; Tham, Y.C.; Yan, M.Y.C.; Ting, D.S.W.; Li, J.; Sabanayagam, C.; Wong, T.Y.; Cheng, C.Y. *Logistic regression was as good as machine learning for predicting major chronic diseases*. J. Clin. Epidemiol. 122, 56–69. 2020. 69

[58] Aishwarya Mujumdar, V Vaidehi. *Diabetes Prediction using Machine Learning Algorithms*. Procedia Computer Science, Volume 165, Pages 292-299. 2019. 69

https://doi.org/10.1016/j.procs.2020.01.047

[59] Priyanka Rajendra, Shahram Latifi. *Prediction of diabetes using logistic regression and ensemble techniques*. Computer Methods and Programs in Biomedicine Update, Volume 1, 100032, ISSN 2666-9900. 2021. 71, 83, 92, 93, 97, 98

https://doi.org/10.1016/j.cmpbup.2021.100032

[60] Collins, G. S., Mallett, S., Omar, O., & Yu, L. M. *Developing risk prediction models for type 2 diabetes: a systematic review of methodology and reporting*. BMC medicine, 9, 103. 2011. 71

https://doi.org/10.1186/1741-7015-9-103

[61] Kaggle. *Pima Indians Diabetes Database*. Consultado el 15 de Marzo de 2022. 72

https://www.kaggle.com/uciml/pima-indians-diabetes-database

[62] data.world. *Diabetes Prediction*. Consultado el 15 de Marzo de 2022. 72

https://data.world/informatics-edu/diabetes-prediction

[63] Naz, Huma, and Sachin Ahuja. *Deep learning approach for diabetes prediction using PIMA Indian dataset*. Journal of diabetes and metabolic disorders vol. 19,1 391-403. 14 Apr. 2020. 72

https://doi.org/10.1007/s40200-020-00520-5

[64] Quadrat analysis of point pattern. *QStatistic*. Consultado el 15 de Marzo de 2022. 85

https://geographicdata.science/book/notebooks/08_point_pattern_analysis.html

[65] medium. *Outlier Detection with K-means Clustering in Python*. Ayşe Kübra Kuyucu. Consultado el 15 de Marzo de 2022. 81

https://medium.datadriveninvestor.com/outlier-detection-with-k-means-clustering-in-python-ee3ac1826fb0

[66] machinelearningmastery. *ROC Curves and Precision-Recall Curves for Imbalanced Classification*. Jason Brownlee. Consultado el 9 de Mayo de 2022. 99

https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-imbalanced-classification/

Casa abierta al tiempo
UNIVERSIDAD AUTÓNOMA METROPOLITANA

# ACTA DE EXAMEN DE GRADO

No. 00228
**Matrícula: 2202800086**

LA CIENCIA DE DATOS Y ANÁLISIS MULTIVARIANTE APLICADOS AL ANÁLISIS Y PREDICCIÓN DE DIABETES.

En la Ciudad de México, se presentaron a las 16:00 horas del día 14 del mes de diciembre del año 2022 en la Unidad Iztapalapa de la Universidad Autónoma Metropolitana, los suscritos miembros del jurado:

DR. JOAQUIN DELGADO FERNANDEZ
DRA. BLANCA ROSA PEREZ SALVADOR
DRA. LIZBETH NARANJO ALBARRAN
DR. ASAEL FABIAN MARTINEZ MARTINEZ

Bajo la Presidencia del primero y con carácter de Secretario el último, se reunieron para proceder al Examen de Grado cuya denominación aparece al margen, para la obtención del grado de:

MAESTRO EN CIENCIAS (MATEMÁTICAS APLICADAS E INDUSTRIALES)

DE: EDUARDO ANTONIO SANTIAGO TOLEDO

EDUARDO ANTONIO SANTIAGO TOLEDO
ALUMNO

y de acuerdo con el artículo 78 fracción III del Reglamento de Estudios Superiores de la Universidad Autónoma Metropolitana, los miembros del jurado resolvieron:

Aprobar

Acto continuo, el presidente del jurado comunicó al interesado el resultado de la evaluación y, en caso aprobatorio, le fue tomada la protesta.

REVISÓ

MTRA. ROSALIA SERRANO DE LA PAZ
DIRECTORA DE SISTEMAS ESCOLARES

DIRECTOR DE LA DIVISIÓN DE CBI

Román Linares Romero
DR. ROMAN LINARES ROMERO

PRESIDENTE

DR. JOAQUIN DELGADO FERNANDEZ

VOCAL

DRA. BLANCA ROSA PEREZ SALVADOR

VOCAL

DRA. LIZBETH NARANJO ALBARRAN

SECRETARIO

Asael

DR. ASAEL FABIAN MARTINEZ MARTINEZ