ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE CIENCIAS

MODELO DE PREDICCIÓN DEL RENDIMIENTO ACADÉMICO PARA EL CURSO DE NIVELACIÓN DE LA ESCUELA POLITÉCNICA NACIONAL A PARTIR DE UN MODELO DE APRENDIZAJE SUPERVISADO AUTOMATIZADO EN R

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERÍA MATEMÁTICA

PROYECTO DE INVESTIGACIÓN

KAREN PRISCILLA CALVA YAGUANA
karen.calva@epn.edu.ec

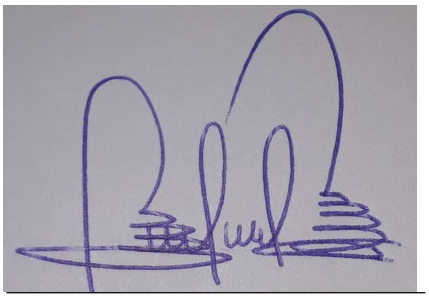
Director: MIGUEL ALFONSO FLORES, PH.D.
miguel.flores@epn.edu.ec

QUITO, FEBRERO 2020

DECLARACIÓN

Yo, KAREN PRISCILLA CALVA YAGUANA, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

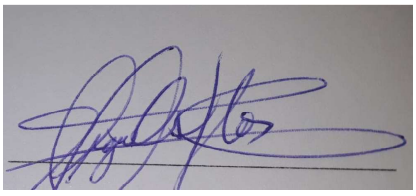
A través de la presente declaración cedo mis derechos de propiedad intelectual, correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



Karen Priscilla Calva Yaguana

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por KAREN PRISCILLA CALVA YAGUANA, bajo mi supervisión.



Miguel Alfonso Flores, Ph.D.

Director del Proyecto

Índice general

Resumen **1**

Abstract **2**

1. Introducción **3**

2. Marco Teórico **5**

2.1. Problemas de clasificación y su modelamiento 6

2.1.1. Modelos Aditivos Generalizados 6

2.1.2. Métodos basados en Árboles 7

2.1.3. Árboles de Clasificación 8

2.1.4. Modelos Boosting y Árboles Aditivos 12

2.1.5. Árboles Boosting 19

2.1.6. Optimización numérica a través del Gradient Boosting 21

2.1.7. Implementación por Gradient Boosting 24

2.1.8. Modelo de elección binaria: Regresión Logística 25

2.2. Evaluación y selección de los modelos 27

2.2.1. Sesgo, varianza y complejidad del modelo 28

2.2.2. Descomposición del sesgo y la varianza 31

2.2.3. Optimismo de la tasa de error de entrenamiento 33

2.2.4. Estimación del error de predicción en la muestra 35

2.2.5. Validación cruzada 37

2.2.6. Algoritmo genético para selección del mejor modelo lineal 42

2.3. Evaluación del desempeño de modelos de clasificación 44

2.3.1. Matriz de confusión 44
2.3.2. Curvas ROC 46

3. Metodología Analítica y Resultados **48**
3.1. Definición de las variables del modelo 48
3.1.1. Variable Dependiente 49
3.1.2. Variables Independientes 49
3.2. Construcción del modelo predictivo 50
3.2.1. Implementación del Algoritmo Gradient Boosting Machine. . . . 51
3.2.2. Resultados del Algoritmo Gradient Boosting Machine 53
3.3. Modelo Inferencial 57

4. Conclusiones y Recomendaciones **64**

Bibliografía **66**

A. Descripción del Curso de Nivelación **69**

B. Tasas de Aprobación y Reprobación **71**

C. Algoritmo GBM implementado en R **74**

D. Predicciones para los datos de prueba **78**

Índice de figuras

2.1. Particiones y el algoritmo CART 8
2.2. Medidas de impureza de los nodos para un problema de clasificación binaria. 11
2.3. Esquema de AdaBoost 13
2.4. Boosting con aumentos. 18
2.5. Error de la muestra de prueba y la muestra de entrenamiento. 29
2.6. Error de predicción, sesgo cuadrático y varianza. 32
2.7. AIC usado para la selección de modelos 37
2.8. Curva de aprendizaje hipotética para un clasificador en una tarea dada 39
2.9. Error de predicción y curva de validación cruzada 41
2.10. Manera correcta e incorrecta de hacer validación cruzada 42
2.11. Matriz de confusión binaria 44
2.12. Ejemplos de curvas ROC 46

3.1. Esquema de AdaBoost 53
3.2. Importancia de variables 54
3.3. Matriz de confusión 55
3.4. Curva ROC en la muestra de validación 55

Índice de tablas

2.1. Gradientes para funciones de pérdida comúnmente usadas 23

3.1. Descripción de variables 50
3.2. Estadísticos de la matriz de confusión 54
3.3. Predicciones para el conjunto de prueba (periodo 2019-A) por carrera de pregrado. 56
3.4. Predicciones para el conjunto de prueba (periodo 2019-A) por materia del curso de nivelación 57
3.5. Resultados del modelo inferencial 59

B.1. Indicadores de eficiencia interna del Curso de Nivelación 71
B.2. Indicadores de eficiencia interna por tipo de Curso de Nivelación 72
B.3. Indicadores de eficiencia interna por tipo de Curso de Nivelación y por Asignatura 73

D.1. Predicción por carrera para el periodo 2019-A 79
D.2. Predicción por materia para el periodo 2019-A 80

DEDICATORIA

Dedicada a mi hermana Raiza, te esperé once años para que cambiaras mi vida. Para ti hice de mi la mujer que soy ahora. Eres mi proyecto más amado y mi fuente de inspiración.

Resumen

En el presente proyecto se describe una metodología estadística basada en árboles de decisión y regresión logística donde el problema de aprendizaje se formula en términos de la minimización de la función de error mediante el método del descenso del gradiente, convirtiendo el problema de aprendizaje en un problema de optimización, como va siendo habitual. Para la metodología se toman en consideración variables socioeconómicas, demográficas, familiares, institucionales y de desempeño académico en la postulación y en el curso de nivelación que tiene un estudiante, con el fin de predecir la probabilidad de que dicho estudiante apruebe el curso en una ventana de tiempo anterior a la fecha de culminación del semestre. Adicionalmente, se implementa un algoritmo en el software estadístico R, el cual se encarga de realizar de manera automática cada uno de los pasos de la metodología descrita.

Palabras clave: Rendimiento académico, regresión logística, árboles de decisión, función de error, método del descenso del gradiente, programación en R.

Abstract

The purpose of this project is to describe a statistical methodology based on decision trees and logistic regression where the learning problem is formulated in terms of the error function minimization using the gradient descent method, turning the learning problem into an optimization problem. The considered variables for the methodology are socioeconomic, demographic, home-related, institutional and of academic performance in the admission exam and in the preparatory course, to predict the student's probability to approve the nivelation course in a time window prior to the end of the semester. Additionally, the algorithm is implemented in the statistical software R, which is responsible for automatically performing each of the steps of the described methodology.

Keywords: Academic performance, logistic regression, decision trees, error function, gradient descent method, R programming.

Capítulo 1

Introducción

En la actualidad, el rendimiento académico de los estudiantes representa uno de los indicadores de calidad más importantes de la labor académica de las universidades (Garbanzo, 2013), y de este, principalmente en los Cursos de Nivelación (CN) o Propedéutico, depende la oferta académica de los centros de estudio. Mientras más alta sea la tasa de reprobación, mayor será la cantidad de estudiantes que cursen nuevamente el CN en un siguiente semestre, llenando así cupos que podrían ser asignados a nuevos estudiantes. Tanto es así que, acorde a Augusto Barrera, antiguo Secretario de Educación Superior, en la actualidad nacional, la educación superior aumentó su oferta académica un 42 % para el primer semestre del 2018, ofertando cerca de 140.000 cupos para nuevos estudiantes (divididos en 90 000 cupos públicos y 40 000 privados, con 22 691 cupos extras respecto a 2017) [1]. Aunque la oferta creció, todavía no se equipara con la cantidad de postulantes, la cual está cerca de 270 000 jóvenes que semestralmente rinden la prueba Ser Bachiller (Paucar, 2018).

Para entrar en contexto, el rendimiento académico se define como el producto de la asimilación del contenido de los programas de estudio, expresado en calificaciones dentro de una escala convencional; es decir, se refiere al resultado cuantitativo que se obtiene en el proceso de aprendizaje de conocimientos, mediante evaluaciones y otras actividades que realiza el docente. Ya que es cuantificable, el rendimiento académico determina el nivel de conocimiento alcanzado por el estudiante, y es tomado como criterio para medir el éxito o fracaso a través de un sistema de calificaciones (Figueroa, 2014). El bajo rendimiento académico de los estudiantes en los primeros semestres universitarios es un problema que deben enfrentar las universidades en la actualidad. En

[1] http://www.uartes.edu.ec/SENESCYT-senala-incremento-del-42-en-la-oferta-academica-para-el-primer-periodo-academico-2018.php

este sentido, el alto porcentaje de reprobados por materias en semestres iniciales trae como consecuencia que se produzca una "barrera académica" que impide que la oferta de cupos sea la deseada, ya que no se puede en el corto plazo incrementar la planta docente para estas asignaturas, ni habilitar nuevos espacios físicos en las universidades (Valera y col., 2009).

El presente trabajo busca entender el rendimiento de los estudiantes en el CN de la Escuela Politécnica Nacional, ya que ellos son los principales perjudicados de la alta tasa de reprobación. A la vez, se propone un modelo automatizado en el software estadístico R para predecir su probabilidad de reprobación, que permita tomar acciones tempranas en su beneficio. Esto debido a que, mientras los estudiantes no adquieran los conocimientos y habilidades básicas para iniciar sus estudios universitarios, el problema de la reprobación se seguirá manifestando con igual o mayor tendencia en semestres posteriores.

Se sabe de estudios externos (e.g. Aina y col., 2018) que han tenido como objetivo determinar las variables explicativas del rendimiento académico de los estudiantes de educación superior, que estas parten de determinantes personales (aptitudes académicas, habilidades y comportamientos), bagaje y redes familiares (atención de los padres, contexto socioeconómico, etc.), sociodemográficos (género, edad, estado civil, etc.) e intrínsecos a la institución (estructura del semestre, pénsum de estudios, etc.). El determinar analíticamente los factores que influyen en el rendimiento académico de los estudiantes permitirá implementar medidas adecuadas para combatir la alta tasa de reprobación, y ayudará a predecir con antelación el número de estudiantes que aprobarán el CN y los que lo harán en segunda matrícula. Evidentemente, con esta información se podrá planificar de mejor manera los cupos del próximo periodo en el CN y en cada carrera.

Capítulo 2

Marco Teórico

En este capítulo se describen las nociones y definiciones teóricas necesarias para comprender la metodología utilizada en la construcción del modelo de predicción e inferencia del rendimiento académico. En su mayoría estas fueron tomadas de Hastie, Tibshirani y Friedman (2017), y serán constantemente utilizadas en este trabajo.

Para lograr un mejor entendimiento de las metodologías, es preciso que primero revisemos el concepto de aprendizaje estadístico o "statistical learning". Acorde a James y col. (2000), el aprendizaje estadístico se refiere a un amplio conjunto de herramientas para entender datos, que pueden incluir varias situaciones, tales como:

- Predecir el nivel de ventas de una empresa en base a su gasto en publicidad.
- Clasificar estudiantes que reprobarán o aprobarán un curso de nivelación.
- Descubrir patrones, por ejemplo, tipos de clientes según sus características socio-demográficas y de consumo.
- Realizar inferencia sobre variables que en teoría tienen efecto sobre un fenómeno.

Una vez que los datos han sido entendidos, las nuevas ideas generadas podrían ser utilizadas para ser monetizadas o incluso comunicadas para el uso de la comunidad en general.

Es así que para entender las ideas específicamente relacionadas al problema entre manos, es decir, para predecir la probabilidad de que un estudiante repruebe o no el curso de nivelación de la Escuela Politécnica Nacional y a la vez obtener conclusiones a manera de inferencia de los resultados de un modelo estadístico, hemos dividido este capítulo en tres secciones principales: problemas de clasificación y su modelamiento, selección del mejor modelo y evaluación de su desempeño.

2.1. Problemas de clasificación y su modelamiento

En el aprendizaje estadístico existen dos tipos de problemas supervisados (i.e. que poseen una variable dependiente que puede ser estimada): regresión y clasificación. En el problema de regresión, se estiman modelos que puedan predecir una variable de respuesta cuantitativa, mientras que en el problema de clasificación tenemos un conjunto de datos de entrenamiento $(x_1, y_1), ..., (x_n, y_n)$, usualmente el 70 % de los datos, que podemos usar para construir un clasificador, i.e. una variable dependiente y que posee varios grupos o clases. En este ejercicio desearíamos que dicho clasificador se comporte bien tanto en un conjunto de datos de entrenamiento, como en un conjunto de datos de prueba; es decir, en un grupo de datos que no fue considerado al momento de la estimación (alrededor del 30 % de datos). Para tal motivo existen varias técnicas de modelamiento, que serán descritas a continuación.

2.1.1. Modelos Aditivos Generalizados

Los modelos de regresión lineal juegan un papel importante en muchos análisis de datos proporcionando reglas de predicción y clasificación. Aunque es atractivo por su simpleza, el modelo de regresión lineal tradicional a menudo falla en situaciones de la vida real porque los efectos no suelen ser lineales. A continuación describiremos métodos estadísticos flexibles y automáticos que pueden usarse para identificar y caracterizar los efectos de regresión que no son lineales. Estos métodos se denominan "modelos aditivos generalizados". Un modelo aditivo generalizado tiene la siguiente forma:

$$E(Y|X_1, X_2, ..., X_p) = \alpha + f_1(X_1) + f_2(X_2) + ... + f_p(X_p) \tag{2.1}$$

Donde $X_1, X_2, ..., X_p$ representan a las variables predictoras o independientes e Y es la variable de salida o dependiente; los f_j's son funciones suaves no especificadas (no paramétricas).

Los modelos aditivos proporcionan una extensión útil de los modelos lineales, haciéndolos más flexibles y conservando gran parte de su capacidad de interpretación. Sin embargo, los modelos aditivos pueden tener limitaciones para grandes volúmenes de datos, ya que el algoritmo del modelo aditivo se adapta a todos los predictores, lo que no es deseable cuando hay un gran número de datos disponible. Para estos problemas un enfoque progresivo por etapas como el *boosting* es más efectivo, ya que a la vez permite que se incluyan interacciones en el modelo.

2.1.2. Métodos basados en Árboles

Los métodos basados en árboles dividen el espacio de variables en un conjunto de rectángulos y luego ajustan un modelo simple (como una constante) en cada uno. Son conceptualmente simples pero potentes. A manera de ilustración describiremos un método popular para la regresión y clasificación basada en árboles, denominado CART[1].

Consideremos un problema de regresión con la respuesta continua Y y las entradas X_1 y X_2, cada una tomando valores en el intervalo unitario. El panel superior izquierdo de la Figura 2.1 muestra el espacio de variables particionado por líneas que son paralelas a los ejes de coordenadas. En cada elemento de partición podemos modelar Y con una constante diferente. Sin embargo, hay un problema: aunque cada línea de partición tiene una descripción simple como $X_1 = c$, algunas de las regiones resultantes son complicadas de describir.

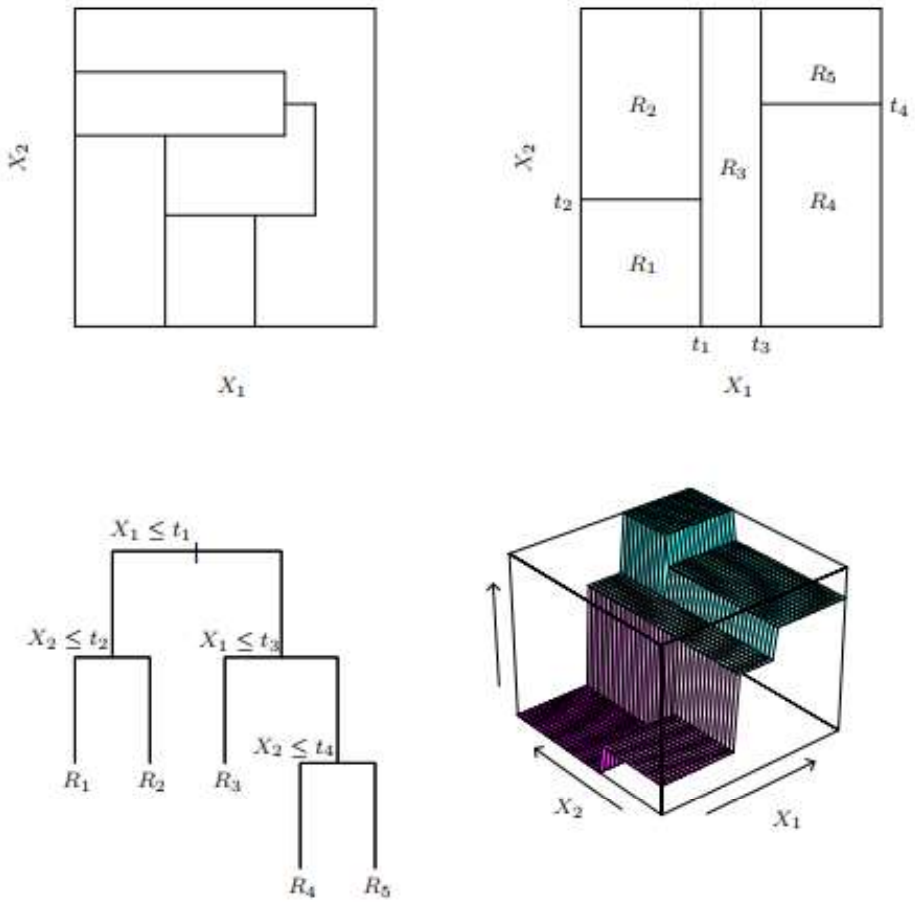
Para simplificar, restringimos la atención a particiones binarias recursivas como en el panel superior derecho de la Figura 2.1. Primero dividimos el espacio en dos regiones y modelamos la variable respuesta por la media de Y en cada región. Luego, una o ambas de estas regiones se dividen en dos regiones más, y este proceso continúa, hasta que se aplica alguna regla de detención. Por ejemplo, dividamos la figura en $X_1 = t_1$. Entonces la región $X_1 = t_1$ se divide en $X_2 = t_2$ y la región $X_1 > t_1$ se divide en $X_1 = t_3$. Finalmente, la región $X_1 > t_3$ se divide en $X_2 = t_4$. El resultado de este proceso es una partición en cinco regiones $R_1, R_2, ..., R_5$ como se muestra en la figura. El siguiente modelo de regresión predice Y con una constante c_m en la región R_m:

$$\hat{f}(X) = \sum_{m=1}^{5} c_m I(X_1, X_2) \in R_m \tag{2.2}$$

Una ventaja clave del árbol binario recursivo es su interpretabilidad. La partición del espacio de variables está completamente descrito por un solo árbol. Con más de dos entradas, las particiones como esa en el panel superior derecho de la Figura 2.1 son difíciles de dibujar, pero la representación del árbol binario funciona de la misma manera.

[1]Classification and Regression Trees.

Figura 2.1: Particiones y el algoritmo CART. El panel superior derecho muestra el espacio de variables bidimensional particionado mediante división binaria recursiva, como se usa en CART, aplicada a algunos datos simulados. El panel superior izquierdo muestra una partición general que no se puede obtener de la división binaria recursiva. El panel inferior izquierdo muestra el árbol correspondiente a la partición en el panel superior derecho, y en el panel inferior derecho aparece una gráfica de perspectiva de la superficie de predicción (Hastie, Tibshirani y Friedman, 2017, p.306).



2.1.3. Árboles de Clasificación

Para entender cómo realiza la estimación un árbol de clasificación, es necesario explicar primero cómo estimar un árbol de regresión. Entonces, recordemos que nuestros datos constan de p variables de entrada y una respuesta, para cada una de las N observaciones: es decir, (x_i, y_i) para $i = 1, 2, ..., N$, con $x_i = (x_{i1}, x_{i2}, ..., x_{ip})$. El algoritmo necesita decidir automáticamente en cómo dividir las variables y en qué puntos, además de la topología (o forma) que el árbol debería tener. Supongamos que primero tenemos una partición en M regiones $R_1, R_2, ..., R_M$ y que podemos modelar la respuesta como

una constante c_m en cada región:

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m) \tag{2.3}$$

Si adoptamos como criterio de minimización la suma de cuadrados $\sum(y_i - f(x_i))^2$, es fácil observar que el mejor $\hat{c_m}$ es equivalente a ponderar y_i en la región R_m:

$$\hat{c_m} = \text{promedio}(y_i | x_i \in R_m) \tag{2.4}$$

Sin embargo, encontrar la mejor partición binaria en términos de la suma de cuadrados mínima es computacionalmente inviable, y por ello se procede con otro tipo de algoritmo. Empezando con el conjunto de datos completo, se considera una variable j candidata a dividir el árbol y un punto de partición s, definiendo así el par de hiperplanos:

$$R_1(j,s) = \{X | X_j \leq s\} \text{ y } R_2(j,s) = \{X | X_j > s\} \tag{2.5}$$

Luego buscamos la variable divisora j y el punto de división s que resuelva:

$$\min_{j,s}[\min_{c1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2] \tag{2.6}$$

Para cualquier selección de j y s, la minimización interna es resuelta por:

$$\hat{c1} = \text{promedio}(y_i | x_i \in R_1(j,s)) \text{ y } \hat{c2} = \text{ave}(y_i | x_i \in R_2(j,s)) \tag{2.7}$$

Para cada variable divisora, la determinación del punto de división s puede ser hecha de manera rápida y por ende, revisar todas las variables de entrada y determinar si el mejor par (j,s) es viable.

Habiendo encontrado la mejor división, particionamos los datos en las dos regiones resultantes y repetimos el proceso de división en cada una de las dos regiones. Luego este proceso es repetido en todas las regiones resultantes.

¿Qué tanto debemos dejar que crezca el árbol? Es claro que un árbol muy grande podría sobreajustarse a los datos, mientras que un árbol muy pequeño podría no capturar la estructura importante.

El tamaño del árbol es un parámetro de ajuste que maneja la complejidad del

modelo, y por ende tal óptimo debe ser escogido de manera adaptativa de los datos. Un enfoque sería dividir los nodos del árbol solo si la caída en la suma de cuadrados excede algún umbral. Esta estrategia es sin embargo muy poco ambiciosa debido a que una división aparentemente innecesaria, podría llevar a otra en cambio muy buena.

La estrategia usual es hacer crecer un árbol grande T_0, deteniendo la división solo cuando algún nodo de tamaño mínimo es alcanzado. Luego este árbol grande es podado usando el *podado costo-complejidad*, ahora descrito.

Definimos un sub-árbol $T \subset T_0$ a ser cualquier árbol que pueda ser obtenido podando T_0, es decir, contrayendo cualquier número de sus nodos internos no terminales. Por notación, indexaremos los nodos terminales como m, con el nodo m representando la región R_m. $|T|$ denota el número de nodos terminales en T, dejando que:

$$\begin{aligned} N_m &= \#\{x_i \in R_m\} \\ \hat{c_m} &= \frac{1}{N_m} \sum_{x_i \in R_m} y_i \\ Q_m(T) &= \frac{1}{N_m} \sum_{x_i \in R_m} (y_i - \hat{c_m}^2) \end{aligned} \tag{2.8}$$

para definir el criterio costo complejidad:

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T| \tag{2.9}$$

La idea es encontrar, para cada α, el sub-árbol $T_\alpha \subseteq T_0$ que minimice $C_\alpha(T)$. El parámetro de ajuste $\alpha \geq 0$ domina el intercambio entre el tamaño del árbol y su calidad de ajuste a los datos. Valores muy grandes de α resultan en árboles más pequeños T_α, y por el contrario, para valores pequeños de α. Como la notación lo sugiere, con $\alpha = 0$ la solución es el árbol completo T_0.

Para cada α se puede mostrar que existe un único sub-árbol T_α que minimiza $C_\alpha(T)$. Para encontrar T_α usaremos el método del *podado del vínculo más débil (weakest link prunning)*: contraemos sucesivamente el nodo interno que produzca el menor incremento por nodo de $\sum_m N_m Q_m(T)$, y continuamos hasta que se produzca un árbol raíz (i.e. con un solo nodo). Esto nos dará una secuencia finita de sub-árboles. Se puede mostrar que en esta secuencia constará T_α. Así, la estimación de α se logra a través de validación cruzada quíntuple, donde escogemos el valor $\hat{\alpha}$ que minimiza la suma de cuadrados de validación cruzada. Nuestro árbol final viene denotado por $T_{\hat{\alpha}}$.

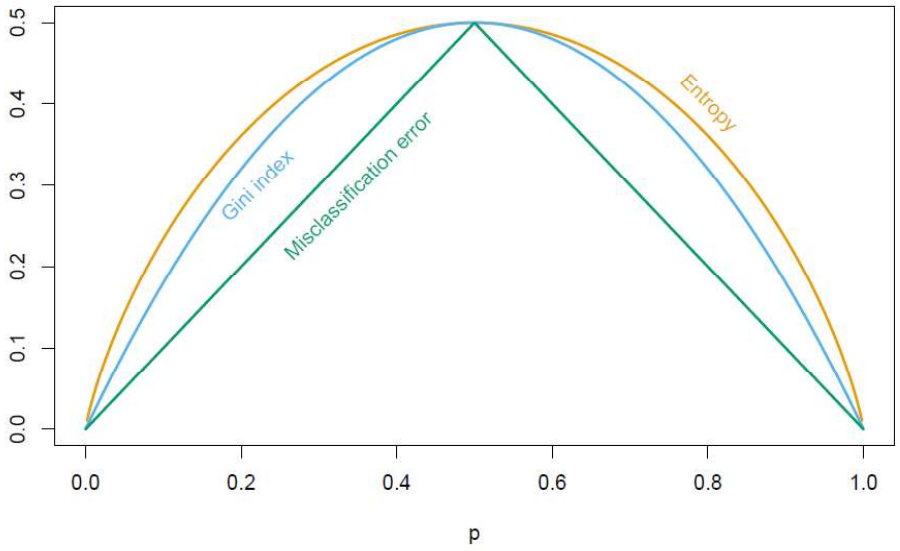
Cuando la variable objetivo es un categoría con valores $1, ..., K$, lo único que se debe cambiar en el algoritmo de árbol anteriormente descrito es el criterio para dividir los nodos y la poda del árbol. Mientras que para árboles de regresión se usa la medida de impureza del error cuadrado Q_m, en árboles de clasificación, en un nodo m, representando a una región R_m con N_m observaciones, se usa:

$$\hat{p_{mk}} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k) \tag{2.10}$$

que es la proporción de observaciones clase k en un nodo m. Clasificaremos las observaciones en un nodo m a la clase $k(m) = \text{argmax}_k \hat{p_{mk}}$, i.e. la clase mayoritaria en el nodo m. Las medidas $Q_m(T)$ de impureza incluyen:

- *Error de clasificación:* $\frac{1}{N_m} \sum_{i \in R_m} I(y_i \neq k(m)) = 1 - \hat{p_{mk(m)}}$
- *Índice de Gini:* $\sum_{k \neq k'} \hat{p_{mk}} \hat{p_{mk'}} = \sum_{k=1}^{K} \hat{p_{mk}}(1 - \hat{p_{mk}})$
- *Entropía cruzada o desviación:* $-\sum_{k=1}^{K} \hat{p_{mk}} \log \hat{p_{mk}}$

Figura 2.2: Medidas de impureza de los nodos para un problema de clasificación binaria, como una función de la proporción p en la clase 2. La entropía cruzada ha sido reescalada para pasar a través de (0.5,0.5) (Hastie, Tibshirani y Friedman, 2017, p.309).



Para dos categorías, si p representa la proporción en la segunda clase, estas tres medidas son $1 - \text{máx}(p, 1-p)$, $2p(1-p)$ y $-p \log p - (1-p) \log(1-p)$ respectivamente. Estas se muestran en la figura 2.2. Las tres son similares, pero la entropía

cruzada y el índice de Gini son diferenciables, y por ende más susceptibles de optimización numérica. Al observar los criterios de división de una variable entre el árbol de regresión y clasificación, podemos observar que necesitamos ponderar las medidas de impureza en un nodo por el número N_{mL} y N_{mR} de observaciones en los dos nodos hijos creados por el nodo dividido m.

Adicionalmente, la entropía cruzada y el índice de Gini son más sensibles a cambios en las probabilidades en un nodo en comparación al error de clasificación. Por ejemplo, en un problema con 400 observaciones en cada clase $(400, 400)$, supongamos que tenemos un nodo que creó la división $(300, 100)$ y $(100, 300)$, mientras que otros nodos crearon $(200, 400)$ y $(200, 0)$. Ambas divisionnes tienen un error de clasificación de $0{,}25$, pero la segunda división produce un nodo puro y probablemente es preferido. Tanto el índice de Gini como la entropía cruzada son menores para este último caso. Por este motivo, tanto el índice Gini como la entropía cruzada deben ser usadas cuando se estima el árbol. Para guíar el podado costo-complejidad en cambio, cualquiera de las 3 medidas puede ser utilizada, aunque de manera general, se usa el error de clasificación.

El índice de Gini puede ser interpretado de dos maneras interesantes. En lugar de clasificar observaciones a la clase predominante en el nodo, podríamos clasificarlas a la clase k con probabilidad $\hat{p_{mk}}$. Entonces el error de entrenamiento esperado de esta regla en el nodo es $\sum_{k \neq k'} \hat{p_{mk}} \hat{p_{mk'}}$. De manera similar, si codificamos cada observación como 1 para la categoría k y 0 en caso contrario, la varianza del nodo con esta respuesta $0 - 1$ es $\hat{p_{mk}}(1 - \hat{p_{mk}})$. Sumando sobre las clases k de nuevo, nos devuelve como resultado el índice de Gini.

2.1.4. Modelos Boosting y Árboles Aditivos

Boosting es uno de los modelos de aprendizaje para clasificación más poderosos introducidas en los últimos veinte años. La motivación para su impulso fue usar un procedimiento que combine los resultados de muchos clasificadores "débiles" para producir una "poderosa combinación".

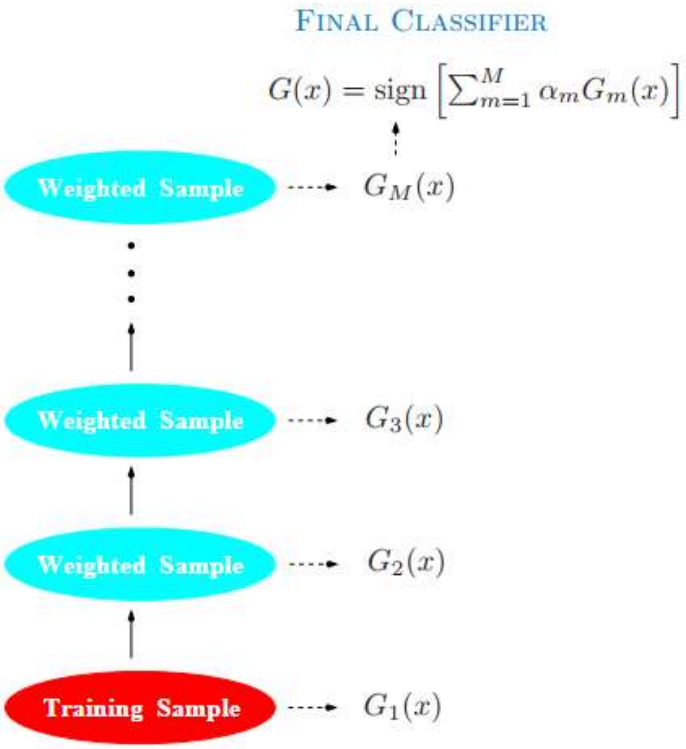
Comenzamos describiendo el algoritmo boosting más popular, introducido por Freund y Schapire (1997), llamado "AdaBoost.M1". Considere un problema de dos clases, con la variable de salida codificada como $Y \in \{-1, 1\}$. Dado un conjunto de variables predictoras X, un clasificador $G(X)$ produce una predicción tomando uno de los dos

valores $\{-1, 1\}$. La tasa de error en la muestra de entrenamiento es:

$$\overline{err} = \frac{1}{N} \sum_{i=1}^{N} I(y_i \neq G(x_i)) \tag{2.11}$$

y la tasa de error esperada en predicciones futuras es $E_{XY} I(Y \neq G(X))$. Un *clasificador débil* es aquel cuya tasa de error es solo ligeramente mejor que la suposición aleatoria. El propósito del boosting es aplicar secuencialmente el algoritmo de clasificación débil a versiones de los datos modificadas repetidamente, produciendo así una secuencia de clasificadores débiles $G_m(x),\ m = 1, 2, .., M$.

Figura 2.3: Esquema de AdaBoost. Los clasificadores se entrenan en versiones ponderadas del conjunto de datos y luego se combinan para producir una predicción final (Hastie, Tibshirani y Friedman, 2017, p.338).



Las predicciones de todos ellos se combinan mediante un voto mayoritario ponderado para producir la predicción final:

$$G(x) = sign(\sum_{m=1}^{M} \alpha_m G_m(x)) \tag{2.12}$$

Donde $\alpha_1, \alpha_2, ..., \alpha_M$ son calculados con el algortimos boosting, y ponderan la contribución respectiva de cada $G_m(x)$. Su efecto es dar mayor influencia a los clasificadores

más precisos en la secuencia. La figura 3.2 muestra un esquema del procedimiento AdaBoost.

Las modificaciones de los datos en cada paso del boosting consisten en aplicar pesos $w_1, w_2, ..., w_N$ para cada una de las observaciones de entrenamiento (x_i, y_i), $i = 1, 2, ..., N$. Inicialmente, todos los pesos se establecen en $wi = 1/N$, de modo que el primer paso simplemente entrena un clasificador en los datos de la manera habitual. Para cada iteración sucesiva $m = 2, 3, ..., M$ los pesos de observación se modifican individualmente y el algoritmo de clasificación se vuelve a aplicar a las observaciones ponderadas. En el paso m, las observaciones que fueron clasificadas erróneamente por el clasificador $G_{m-1}(x)$ inducidas en el paso anterior aumentarán sus pesos, mientras que los pesos disminuirán para aquellos que se clasificaron correctamente. Así, a medida que avanzan las iteraciones, las observaciones que son difíciles de clasificar correctamente reciben una influencia cada vez mayor.

Estimación Boosting de un Modelo Aditivo

El éxito del boosting realmente no es tan misterioso. La clave yace en la expresión 2.12. El boosting es una manera de estimar una expansión aditiva en un conjunto de funciones "base" elementales. Aquí las funciones bases son los clasificadores individuales $G_m(x) \in \{-1, 1\}$. Más generalmente, las expansiones de las funciones base toman la forma:

$$f(x) = \sum_{m=1}^{M} \beta_m b(x; \gamma_m) \tag{2.13}$$

donde $\beta_m, m = 1, 2, ..., M$ son los coeficientes de expansión, y $\underline{(}x; \gamma) \in \mathbb{R}$ son usualmente funcionales del argumento multivariado x, caracterizado por un conjunto de parámetros γ. Las expansiones aditivas como esta son el corazón de muchas técnicas de aprendizaje.

Típicamente estos modelos se estiman minimizando una función de pérdida promedio sobre los datos de entrenamiento, tal como la función de pérdida de error cuadrático o basada en verosimilitud:

$$\min_{\{\beta_m, \gamma_m\}_1^M} \sum_{i=1}^{N} L\left(y_i, \sum_{m=1}^{M} \beta_m b(x_i; \gamma_m)\right) \tag{2.14}$$

Para muchas funciones de pérdida $L(y, f(x))$ y/o funciones base $b(x; \gamma)$, esto re-

quiere técnicas numéricas de optimización computacionalmente intensivas. De cualquier manera, una alternativa simple puede ser encontrada cuando es viable resolver rápidamente el subproblema de ajustar una única función base:

$$\min_{\beta,\gamma} \sum_{i=1}^{N} L(y_I, \beta b(x_i; \gamma)) \tag{2.15}$$

Modelado aditivo progresivo por etapas

El modelado aditivo progresivo por etapas aproxima la solución a 2.14 añadiendo secuencialmente nuevas funciones base a la expansión sin ajustar los parámetros y coeficientes de aquellos que ya han sido añadidos. Esto es delineado en el Algoritmo 1:

Algoritmo 1: Modelado aditivo progresivo por etapas

1. Inicializar $f_0(x) = 0$.
2. Para $m = 1$ hasta M:
   a) Computar
   $$(\beta_m, \gamma_m) = \arg\min_{\beta,\gamma} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + \beta b(x_i, \gamma))$$
   b) Fijar $f_m(x) = f_{m-1}(x) + \beta_m b(x; \gamma_m)$

En cada iteración m, se resuelve la función base óptima $b(x; \gamma_m)$ y su correspondiente coeficiente β_m para añadir a la expresión actual $f_{m-1}(x)$. Esto produce $f_m(x)$ y el proceso es repetido. Los términos añadidos previamente no son modificados.

Para la función de pérdida con error cuadrático:

$$L(y, f(x)) = (y - f(x))^2 \tag{2.16}$$

uno tiene que:

$$\begin{aligned} L(y_i, f_{m-1}(x_i) + \beta b(x_i, \gamma)) &= (y_i - f_{m-1}(x_i) - \beta b(xi; \gamma))^2 \\ &= (r_{im} - \beta b(x_i; \gamma))^2 \end{aligned} \tag{2.17}$$

donde $r_{im} = y_i - f_{m-1}(x_i)$ es simplemente el residuo del modelo actual para la i-ésima observación. Entonces, para una función de pérdida de error cuadrático, el término $\beta_m b(x; \gamma_m)$ que mejor se ajusta a los residuos actuales es añadido a la expansión

en cada paso. Sin embargo, en problemas de clasificación, la función de pérdida de error cuadrático no es una buena opción, y es necesario considerar otro criterio de pérdida.

Pérdida exponencial y AdaBoost

Algoritmo 2: AdaBoost.M1.

1. Inicializar los pesos de cada observación $w_i = 1/N, i = 1, 2, ..., N$
2. Para $m = 1$ hasta M:
   - *a*) Ajustar un clasificador $G_m(x)$ a los datos de entrenamiento usando pesos w_i
   - *b*) Computar
   $$err_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i}$$
   - *c*) Computar $\alpha_m = \log((1 - err_m)/err_m)$
   - *d*) Fijar $w_i \leftarrow w_i \exp[\alpha_m I(y_i \neq G_m(x_i))]$ con $i = 1, 2, ..., N$.
   - *e*) Devolver el resultado $G(x) = sign\left[\sum_(m = 1)^{M} \alpha_m G_m(x)\right]$

Ahora se muestra que el algoritmo 2 AdaBoost.M1 es equivalente al modelado aditivo progresivo por etapas (algoritmo 1) usando la siguiente función de pérdida:

$$L(Y, f(x)) = \exp(-yf(x)) \tag{2.18}$$

Para el AdaBoost las funciones base son los clasificadores individuales $G_m(x) \in \{-1, 1\}$. Usando la función exponencial de pérdida, se debe resolver:

$$(\beta_m, G_m) = arg \min_{\beta, G} \sum_{i=1}^{N} \exp\left[-y_i(f_{m-1}(x_i) + \beta G(x_i))\right] \tag{2.19}$$

para el clasificador G_m y su coeficiente correspondiente β_m que deben ser añadidos en cada paso. Esto puede ser expresado como:

$$(\beta_m, G_m) = arg \min_{\beta, G} \sum_{i=1}^{N} w_i^{(m)} \exp(-\beta y_i G(x_i)) \tag{2.20}$$

con $w_i^{(m)} = \exp(-y_i f_{m-1}(x_i))$. Debido a que cada $w_i^{(m)}$ no depende de β ni de

$G(x)$, puede considerarse como un peso que se aplica a cada observación. Este peso depende de $f_{m-1}(x_i)$, y por tanto cada peso individual cambia en cada iteración m.

La solución a la ecuación 2.20 puede ser obtenida en dos pasos. Primero, para cualquier valor de $\beta > 0$, la solución para $G_m(x)$ es:

$$G_m = arg\min_{G} \sum_{i=1}^{N} w_i^{(m)} I(y_i \neq G(x_i)) \tag{2.21}$$

el cual es el clasificador que minimza la tasa de error ponderada al predecir y. Esto puede ser fácilmente observado al expresar el criterio de 2.20 como:

$$e^{-\beta} \sum_{y_i = G(x_i)} w_i^{(m)} + e^{\beta} \sum_{y_i \neq G(x_i)} w_i^{(m)} \tag{2.22}$$

que puede ser escrita de otra manera como:

$$(e^{\beta} - e^{-\beta}) \sum_{i=1}^{N} w_i^{(m)} I(y_i \neq G(x_i)) + e^{-\beta} \sum_{i_1}^{N} w_i^{(m)} \tag{2.23}$$

Uniendo el término G_m a la expresión 2.20 y resolviendo para β tenemos:

$$\beta_m = \frac{1}{2} \log \frac{1 - err_m}{err_m} \tag{2.24}$$

donde err_m es la tasa de error ponderada minimizada:

$$err_m = \frac{\sum_{i=1}^{N} w_i^{(m) I(y_i \neq G_m(x_i))}}{\sum_{i=1}^{N} w_i^{(m)}} \tag{2.25}$$

La aproximación es por ello actualizada:

$$f_m(x) = f_{m-1}(x) + \beta_m G_m(x) \tag{2.26}$$

lo que causa que los pesos para la siguiente iteración sean:

$$w_i^{m+1} = w_i^m e^{-\beta_m y_i G_m(x_i)} \tag{2.27}$$

Usando el hecho de que $-y_i G_m(x_i) = 2I(y_i \neq G_m(x_i)) - 1$, la ecuación anterior se

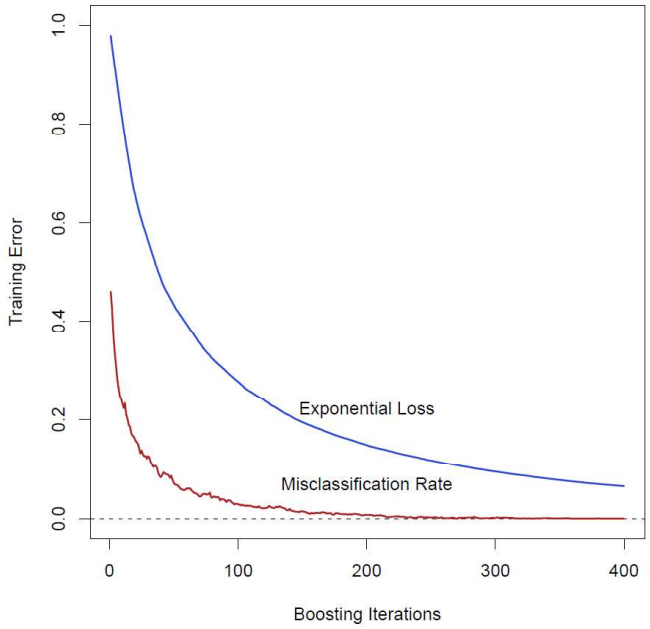
convierte en:

$$w_i^{m+1} = w_i^m e^{\alpha_m I(y_i \neq G_m(x_i))} e^{-\beta m} \tag{2.28}$$

donde $\alpha_m = 2\beta_m$ es la cantidad definida en la línea 2.c del algoritmo 2 AdaBoost.M1. El factor $e^{-\beta m}$ en 2.28 multiplica a todos los pesos por el mismo valor, perdiendo así cualquier efecto, entonces 2.28 es equivalente a la línea 2.d del algoritmo 2.

Uno puede observar la línea 2.a del algoritmo AdaBoost.M1 como un métoodo para resolver aproximadamente la minimización en las ecuaciones 2.21 y 2.23. Por tal motivo se puede concluir que el algoritmo AdaBoost.M1 minimiza el criterio de pérdida exponencial a través de un enfoque de modelado aditivo progresivo.

Figura 2.4: Datos simulados, boosting con aumentos: error de clasificación en los datos de entrenamiento, con una función de pérdida exponencial promedio: $(1/N)\sum_{i=1}^{N} \exp(-y_i f(x_i))$. Después de 250 iteraciones, el error de clasificación es cero, mientras que la función de pérdida exponencial continúa decreciendo (Hastie, Tibshirani y Friedman, 2017, p.345).



La figura 2.4 muestra el error de clasificación y el valor de la pérdida exponencial promedio para el conjunto de datos de entrenamiento de datos simulados. El error de clasificación reduce a cero alrededor de las 250 iteraciones (y se queda allí), mientras que la función de pérdida exponencial sigue mejorando. Claramente el algoritmo AdaBoost no está optimizando el error de clasificación de entrenamiento. Por otro lado, la pérdida exponencial muestra ser más sensible a cambios en las probabilidades de clase estimadas.

2.1.5. Árboles Boosting

Recordemos que los árboles de regresión y clasificación particionan el espacio de todos los valores de las variables predictoras en regiones disjuntas $R_j, j = 1, 2, ..., J$, representadas por los nodos terminales del árbol. Una constante γ_j es asignada a cada región y la regla predictiva es:

$$x \in R_j \rightarrow f(x) = \gamma_j \tag{2.29}$$

Así, un árbol puede ser formalmente descrito como:

$$T(x; \Theta) = \sum_{j=1}^{J} \gamma_j I(x \in R_j) \tag{2.30}$$

Con parámetros $\Theta = \{R_j, \gamma_j\}_1^J$. J es usualmente interpretado como un meta-parámetro. Los parámetros se encuentran minimizando el riesgo empírico:

$$\hat{\Theta} = arg \min_{\Theta} \sum_{j=1}^{J} \sum_{x_i \in R_j} L(y_i, \gamma_j) \tag{2.31}$$

el cual es un problema de optimización combinatorial formidable y usualmente se conforma con soluciones aproximadas subóptimas. Por ello, es útil dividir el problema de minimización en dos partes:

- **Encontrar γ_j dado R_j:** Dado R_j, estimar γ_j es típicamente trivial, y usualmente $\hat{\gamma}_j = \overline{y_j}$, la media de y_i cae en la región R_j. Para la pérdida por mala clasificación, $\hat{\gamma}_j$ es la clase modal de las observaciones que caen en la región R_j.

- **Encontrar R_j:** Esta es la tarea difícil, para la cual se encuentran soluciones aproximadas. Nótese que encontrar R_j significa estimar γ_j también. Una estrategia típica es usar un algoritmo de partición recursiva exhaustiva, de arriba hacia abajo para encontrar R_j. Adicionalmente, a veces es necesario aproximar 2.31 con un criterio más suave y conveniente para la optimización de R_j:

  $$\tilde{\Theta} = arg \min_{\Theta} \sum_{j=1}^{N} \tilde{L}(y_i, T(x_i, \Theta)) \tag{2.32}$$

  Entonces dado $\hat{R}_j = \tilde{R}_j$, γ_j puede ser estimado de manera más precisa usando el criterio original.

Para árboles de clasificación se usa dicha estrategia. El índice de Gini reemplaza la pérdida por mala clasificación en el crecimiento de árbol (identificando R_j).

El modelo de árbol por boosting entonces es la suma de tales árboles,

$$f_M(x) = \sum_{m=1}^{M} Tx; \Theta_m \tag{2.33}$$

inducido de atrás hacia adelante de manera escalonada (como en el algoritmo 1). En cada paso de dicho procedimiento se debe resolver:

$$\hat{\Theta_m} = arg\min_{\Theta_m} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + T(x_i; \Theta_M)) \tag{2.34}$$

para el conjunto de la región y las constantes $\Theta_m = \{R_{jm}, \gamma_{jm}\}_1^{I_m}$ del siguiente árbol, dado el actual modelo $f_{m-1}(x)$.

Dadas las regiones R_{jm}, encontrar las constantes óptimas γ_{jm} en cada región se logra resolviendo:

$$\hat{\gamma_{jm}} = arg\min_{\gamma_{jm}} \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i + \gamma_{jm})) \tag{2.35}$$

Encontrar las regiones es difícil, y aún más difícil para un único árbol. Para algunos casos especiales, el problema se simplifica.

Para la pérdida del error cuadrático, la solución a 2.34 no es más difícil que un único árbol. Simplemente es el árbol de regresión que mejor predice los residuos $y_i - f_{m-1}(x_i)$, y $\hat{\gamma_{jm}}$ es la media de estos residuos en cada región.

Para el problema de clasificación binaria con pérdida exponencial, el enfoque escalonado da paso al algoritmo 2 para boosting de árboles de clasificación. En particular, si los árboles $T(x; \Theta_m)$ se restringen a ser árboles de clasificación escalados, es decir, restringimos $\beta_m T(x; \Theta_m)$ con $\gamma_{jm} \in \{-1, 1\}$, la solución a la ecuación 2.34 es el árbol que minimiza la tasa de error ponderada $\sum_{i=1}^{N} w_i^{(m)} I(y_i \neq T(x_i; \Theta_m))$ con los pesos $w_i^{(m)} = e^{-y_i f_{m-1}(x_i)}$.

Sin esta restricción, 2.34 aún se simplifica de una pérdida exponencial a un criterio ponderado exponencial para el nuevo árbol:

$$\hat{\Theta_m} = arg\min_{\Theta_m} \sum_{i=1}^{N} w_i^{(m)} exp[-y_i T(x_i; \Theta_m)] \tag{2.36}$$

Es sencillo implementar un algoritmo de partición recursiva utilizando esta pérdida exponencial ponderada como criterio de división. Dada la región R_{jm}, uno puede demostrar que la solución a 2.35 es el log-odds ponderado en cada región:

$$\hat{\gamma_{jm}} = \frac{1}{2} \log \frac{\sum_{x_i \in R_{jm}} w_i^{(m)} I(y_i = 1)}{\sum_{x_i \in R_{jm}} w_i^{(m)} I(y_i = -1)} \tag{2.37}$$

Esto requiere un algoritmo especializado en el crecimiento del árbol. En la práctica se prefiere la aproximación que usa un árbol de regresión con mínimos cuadrados ponderados.

2.1.6. Optimización numérica a través del Gradient Boosting

Un algoritmo rápido aproximado para resolver la ecuación 2.34 con cualquier función de pérdida diferenciable puede ser derivado por analogía a la optimización numérica. La función de pérdida al usar $f(x)$ para predecir y en los datos de entrenamiento es:

$$L(f) = \sum_{i=1}^{N} L(y_i, f(x_i)) \tag{2.38}$$

El objetivo es minimizar $L(f)$ con respecto a f, donde $f(x)$ está restringida a ser una suma de árboles. Ignorando esta restricción, la optimización de 2.38 puede ser vista como optimización numérica:

$$\hat{\mathbf{f}} = arg \min_{\mathbf{f}} L(\mathbf{f}) \tag{2.39}$$

donde los "parámetros" $\mathbf{f} \in \mathbb{R}^N$ son los valores de aproximar la función $f(x_i)$ en cada uno de los N puntos x_i:

$$\mathbf{f} = \{f(x_1), f(x_2), ..., f(x_N)\}^T \tag{2.40}$$

La optimización numérica resuelve 2.39 como la suma de vectores componentes:

$$\mathbf{f}_M = \sum_{m=0}^{M} \mathbf{h}_m, \mathbf{h}_m \in \mathbb{R}^N \tag{2.41}$$

donde $\mathbf{f}_0 = \mathbf{h}_0$ es el primer intento, y cada $\mathbf{f}_m$ sucesivo es inducido basado en el actual vector parámetro $\mathbf{f}_{m-1}$, el cual es la suma de las actualizaciones inducidas

anteriormente. La optimización numérica difiere en su prescripción al computar cada vector de incremento $\mathbf{h}_m$ (también llamado "paso").

Descenso empinado

El descenso empinado escoge $\mathbf{h}_m = -\rho_m g_m$ donde ρ_m es un escalar y $g_m \in \mathbb{R}^N$ es el gradiente de $L(f)$ evaluado en $\mathbf{f} = \mathbf{f}_{m-1}$. Los componentes del gradiente g_m son:

$$\mathrm{g}_{im} = \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x_i)=f_{m-1}(x_i)} \tag{2.42}$$

El *largo del paso* ρ_m es la solución a:

$$\rho_m = arg\,\min_{\rho} L(\mathbf{f}_{m-1} - \rho g_m) \tag{2.43}$$

La solución es luego actualizada a :

$$\mathbf{f}_m = \mathbf{f}_{m-1} - \rho_m g_m \tag{2.44}$$

y el proceso se repite en la siguiente iteración. El descenso empinado puede ser visto como una estrategia muy codiciosa, debido a que $-g_m$ es la dirección local en $\mathbb{R}^N$ para la cual $L(\mathbf{f})$ decrece más rápido en $\mathbf{f} = \mathbf{f}_{m-1}$.

Gradient Boosting

Las estrategias escalonadas hacia adelante siempre producen un camino muy codicioso. En cada paso el árbol solución es aquel que reduzca al máximo 2.34, dado el modelo actual f_{m-1} y sus ajustes $f_{m-1}(x_i)$. Entonces, las prediccionens del árbol $T(x_i;\Theta_m)$ son análogos a las componentes del gradiente negativo (ecuación 2.42). La principal diferencia entre ellos es que las componentes del árbol $t_m = \{T(x_1;\Theta_m), ..., T(x_N;\Theta_m)\}^T$ no son independientes. Están restringidas a ser los predictores de un nodo terminal J_m, mientras que el gradiente negativo es la dirección no restringida máxima de descenso.

La solución a 2.35 en el enfoque escalonado es análogo a la búsqueda de línea de 2.43 en el descenso empinado. La diferencia es que 2.35 realiza una búsqueda de línea separada para todas aquellas componentes de $\mathbf{t}_m$ que correspondan a cada región terminal separada $\{T(x_i;\Theta_m)\}_{x_i \in R_{jm}}$.

Si el minimizar la función de pérdida en los datos de entrenamiento fuese el único

Tabla 2.1: Gradientes para funciones de pérdida comúnmente usadas

Configuración	Función de Pérdida	$\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}$
Regresión	$\frac{1}{2}[y_i - f(x_i)]^2$	$y_i - f(x_i)$
Regresión	$\lvert y_i - f(x_i)\rvert$	$sign[y_i - f(x_i)]$
Regresión	Huber	$y_i - f(x_i)$ para $\lvert y_i - f(x_i)\rvert \leq \lambda_m$ $\lambda_m sign[y_i - f(x_i)]$ para $\lvert y_i - f(x_i)\rvert > \lambda_m$ donde $\lambda_m = \alpha$-ésimo cuantil $\{\lvert y_i - f(x_i)\rvert\}$
Clasificación	Desviación	k-ésima componente: $I(y_i = G_k) - p_k(x_i)$

objetivo, el descenso empinado sería la estrategia utilizada. El gradiente 2.42 es trivial de calcular para cualquier función diferenciable $L(y, f(x))$, mientras que resolver 2.34 es difícil para algunos criterios. Desafortunadamente el gradiente solamente está definido para los datos de entrenamiento x_i, mientras que el objetivo principal es generalizar $f_M(x)$ a un nuevo conjunto de datos no representado en el conjunto de entrenamiento.

Una posible resolución a este dilema es introducir un árbol $T(x; \Theta_m)$ en la m-ésima iteración cuyos predictores $\mathbf{t}_m$ estén tan cerca como sea posible del gradiente negativo. Usando el error cuadrático para medir cercanía, llegamos a:

$$\tilde{\Theta}_m = arg \min_{\Theta} \sum_{i=1}^{N} (-g_{im} - T(x_i; \Theta))^2 \tag{2.45}$$

Es decir, se estima el árbol T para los valores del gradiente negativo por mínimos cuadrados. A pesar de que las regiones de solución $\tilde{R}_{jm}$ para 2.45 no tenderán a regiones idénticas a las de la solución para 2.34, son lo suficientemente similares para servir a su propósito. En cualquier caso, el boosting escalonado hacia adelante, y la inducción de árboles de arriba hacia abajo son procedimientos de aproximación. Después de construir el árbol en 2.45, las correspondientes constantes en cada región están dadas por 2.35.

La tabla 2.1 resume los gradientes de las funciones de pérdida comúnmente utilizadas. Para la pérdida del error cuadrático, el gradiente negativo es tan solo el residuo ordinario $-g_{im} = y_i - f_{m-1}(x_i)$ de tal manera que 2.45 por su cuenta sea equivalente al boosting de mínimos cuadrados. Con la pérdida del error absoluto, el gradiente negativo es el signo de los residuos, para que en cada iteración 2.45 ajuste al árbol al signo de los residuos actuales por mínimos cuadrados. Para la regresión de Huber, el gradiente negativo es un compromiso entre estos dos.

Para problemas de clasificación, la función de pérdida es la desviación multinomial y K árboles por mínimos cuadrados son construidos en cada iteración. Cada árbol T_{km}

es ajustado a su respectivo vector de gradiente negativo g_{km}:

$$\begin{aligned} -g_{ikm} &= \left[\frac{\partial L(y_i, f_1(x_i), ..., f_k(x_i))}{\partial f_k(x_i)}\right]_{\mathbf{f}(x_i)=\mathbf{f}_{m-1}(x_i)} \\ &= I(y_i = G_k) - p_k(x_i) \end{aligned} \tag{2.46}$$

Para problemas de clasificación binaria ($K = 2$) solo se necesita un árbol.

2.1.7. Implementación por Gradient Boosting

Algoritmo 3: Algoritmo Gradient Boosting

1. Inicializar $f_0(x) = arg\,\text{mín}_\gamma \sum i = 1^N L(y_i, \gamma)$.
2. Para $m = 1$ hasta M:
   a) Para $i = 1, 2, ..., N$ computar:
   $$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}}$$
   b) Se estima un árbol a los objetivos r_{im} que dan regiones terminales R_{jm}, $j = 1, 2, ..., J_m$.
   c) Para $j = 1, 2, ..., J_m$ se computa:
   $$\gamma_{jm} = arg\,\min_\gamma \sum_{x_i \in R_{jm}} L(y_i, f_{m-1} + \gamma)$$
   d) Actualizar $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$
3. Devolver $\hat{f}(x) = f_M(x)$.

El algoritmo 3 presenta el algoritmo genérico del gradiente para árboles de regresión con boosting. Algoritmos específicos se obtienen al escoger distintas funciones de pérdida $L(y, f(x))$. La primera línea del algoritmo inicializa el modelo constante óptimo, que es un árbol con un solo nodo terminal. Las componentes del gradiente negativo computado en la línea 2.a son referidas como pseudo-residuos o residuos generalizados. Los gradientes más usados para distintas funciones de perdida son resumidas en la tabla 2.1.

El algoritmo para el árbol de clasificación es similar. Las líneas 2.a a 2.d son repetidas K veces en cada iteración m, una vez por cada clase, utilizando 2.46. El re-

sultado de la línea 3 pertenece a tres expansiones distintas de K árboles diferentes $f_{kM}(x), k = 1, 2, ..., K$. Para estos cálculos, dos parámetros de ajuste básicos son el número de iteraciones M y los tamaños de cada uno de los árboles constituyentes $J_m, m = 1, 2, ..., M$.

2.1.8. Modelo de elección binaria: Regresión Logística

Finalmente, además de los modelos basados en árboles tenemos que, acorde a James y col. (2000), cuando se da el caso en que la variable dependiente Y de nuestro modelo cae dentro de una de dos (o más) categorías, como es el caso, podemos utilizar un modelo de regresión logística para estimar la probabilidad de que Y pertenezca a una categoría en particular, asegurándonos de que los valores de esta probabilidad se encuentren entre 0 y 1. Los modelos de regresión logística son usados en su mayoría para el análisis de datos y la inferencia, donde su objetivo es entender el rol de variables de entrada importantes al explicar la variable respuesta Y.

Para lograr este cometido debemos modelar $Pr(Y = 1|X) = pr(X)$ usando una función que de como resultado un valor entre 0 y 1 para cualquier vector X. Existen varias funciones que cumplen esta condición, pero quizás la más utilizada es la *función logística*, dada por:

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+...+\beta_p X_p}}{1 + e^{\beta_0+\beta_1 X_1+...+\beta_p X_p}} \tag{2.47}$$

La función logística siempre producirá una curva en forma de S para que independientemente de los valores que tomen las variables de entrada podamos obtener una predicción sensible. Si manipulamos un poco la expresión anterior podemos llegar a:

$$\frac{p(X)}{1 - p(X)} = e\beta_0 + \beta_1 X_1 + ... + \beta_p X_p \tag{2.48}$$

donde la cantidad $\frac{p(X)}{1-p(X)}$ es llamado *odds*, y puede tomar cualquier valor entre 0 e ∞. Valores cercanos a 0 e ∞ indican muy bajas y muy altas probabilidades de que suceda el evento de interés, respectivamente. Si tomamos el logaritmo de ambos lados obtenemos:

$$\log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p \tag{2.49}$$

donde el lado izquierdo es llamado *log-odds* o *logit*. Se puede observar entonces que

el modelo de regresión tiene un logit lineal para el vector X. Así, en un modelo logístico, el incrementar una variable de entrada j en una unidad, incrementa el log-odds en β_j, o equivalentemente, multiplica las posibilidades (odds) en e^{β_j}. En todo caso, debido a que la relación entre $p(X)$ y X no es una línea recta, β_j no corresponde al cambio en $p(X)$ asociado a incremento de una unidad en X_j. La cantidad en la que $p(X)$ cambie debido a la variación de X_j siempre dependerá del valor actual de X. Sin embargo, sin importar el valor de $p(X)$ y si el signo de β_j es positivo (negativo), siempre el aumento de X_j significará el aumento (la disminución) de $p(X)$.

Para ajustar este modelo y encontrar los valores de $\beta_0, \beta_1, ..., \beta_p$, se usa el método de máxima verosimilitud, usando la verosimilitud de Y dado X. Debido a que Pr(Y=1|X) especifica completamente la distribución condicional, la distribución binomial es apropiada, y la función de logaritmo de versimilitud viene dada por:

$$\begin{aligned} \ell(\beta) &= \sum_{i=1}^{N} \{y_i \log p(x_i;\beta) + (1-y_i)\log(1-p(x_i;\beta))\} \\ &= \sum_{i=1}^{N} \{y_i \beta^T x_i + \log(1+e^{\beta^T X_i})\} \end{aligned} \tag{2.50}$$

Para maximizar el logaritmo de verosimilitud debemos igualar las derivadas parciales de esta expresión a cero. Estas derivadas, llamadas ecuaciones de *score* son:

$$\frac{\partial \ell(\beta)}{\partial \beta} = \sum_{i=1}^{N} x_i (y_i - p(X_i;\beta)) = 0 \tag{2.51}$$

que son $p+1$ ecuaciones no lineales en β. Para resolver estas ecuaciones se utiliza el algoritmo de Newton-Raphson, que requiere la segunda derivada o matriz Hessiana:

$$\frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^T} = -\sum_{i=1}^{N} x_i x_i^T p(x_i;\beta)(1-p(x_i;\beta)) \tag{2.52}$$

El algoritmo empieza con β^{old}, y en una iteración actualiza β:

$$\beta^{new} = \beta^{old} - \left(\frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^T} \right)^{-1} \frac{\partial \ell(\beta)}{\partial \beta} \tag{2.53}$$

donde las derivadas son evaluadas en β^{old}.

Si pasamos estas ecuaciones a notación matricial, tenemos que $\mathbf{y}$ denotará un vector de valores de y_i, $\mathbf{X}$ la matriz $N \times (p+1)$ de valores x_i, $\mathbf{p}$ el vector de probabilidades ajustadas con el i-ésimo elemento $p(x_i;\beta^{old})$ y $\mathbf{W}$ una matriz diagonal $N \times N$ de pesos

con el i-ésimo elemento diagonal $p(x_i;\beta^{old})(1-p(x_i;\beta^{old}))$. Así tenemos ahora que:

$$\frac{\partial^2 \ell(\beta)}{\partial\beta\partial\beta^T} = \mathbf{X}(\mathbf{y}-\mathbf{p}) \tag{2.54}$$

$$\frac{\partial^2 \ell(\beta)}{\partial\beta\partial\beta^T} = -\mathbf{X}^T\mathbf{W}\mathbf{X} \tag{2.55}$$

Donde la iteración de Newton es:

$$\begin{aligned} \beta^{new} &= \beta^{old} + (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{y}-\mathbf{p}) \\ &= (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}(\mathbf{X}\beta^{old} + \mathbf{W}^{-1}(\mathbf{y}-\mathbf{p})) \\ &= (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\mathbf{z} \end{aligned} \tag{2.56}$$

En esta expresión básicamente se ha notado la iteración de Newton como una interación de mínimos cuadrados ponderados, con la respuesta:

$$\mathbf{z} = \mathbf{X}\beta^{old} + \mathbf{W}^{-1}(\mathbf{y}-\mathbf{p}) \tag{2.57}$$

conocida muchas veces como *respuesta ajustada*. Estas ecuaciones son solucionadas repetitivamente debido a que en cada iteración $\mathbf{p}$ cambia, al igual que $\mathbf{W}$ y $\mathbf{z}$. Este algoritmo es referido como IRLS o *mínimos cuadrados ponderandos iterativamente - iteratively reweighted least squares*, debido a que en cada iteración se resuelve el problema de mínimos cuadrados ponderados.

Al momento de resolverlo se comienza con $\beta = 0$. Típicamente el algoritmo converge debido a que la función del logaritmo de verosimilitud es cóncavo.

2.2. Evaluación y selección de los modelos

Una vez que hemos terminado de modelar nuestros datos, debemos decidir cómo escoger el mejor modelo a ser utilizado, lo cual se consigue a través de varias técnicas de evaluación, descritas en esta sección.

Comencemos entonces con el error de generalización de un modelo de aprendizaje, el cual se relaciona con su capacidad de predicción en datos de prueba independientes. La evaluación de este desempeño es importante en la práctica ya que guía a la elección del modelo y nos da una medida de la calidad del modelo elegido.

2.2.1. Sesgo, varianza y complejidad del modelo

El sesgo[2] y varianza[3] son dos conceptos importantes a la hora de medir el error en los modelos de aprendizaje automático. Por eso es necesario comprender su significado para evaluar correctamente lo que nos dicen. La figura 2.5 ilustra la importancia de evaluar el error de generalización de un método de aprendizaje. Consideremos primero el caso de una variable cuantitativa objetivo Y, un vector de variables explicativas X y un modelo de predicción $\hat{f}(X)$ que se ha estimado a partir de un conjunto de entrenamiento τ. La función de pérdida para medir errores entre Y y $\hat{f}(X)$ se denota por $L(Y, \hat{f}(X))$. Las opciones típicas de función de pérdida son:

$$L(Y, \hat{f}(X)) = \begin{cases} (Y - \hat{f}(X))^2 & \text{error cuadrático} \\ |Y - \hat{f}(X)| & \text{error absoluto} \end{cases} \tag{2.58}$$

El *error de prueba* (error de generalización) es el error de predicción calculado sobre una muestra independiente.

$$Err_\tau = E[L(Y, \hat{f}(X))|\tau] \tag{2.59}$$

donde X e Y son tomadas aleatoriamente de su distribución conjunta. El conjunto de entrenamiento τ es fijo, y el error de prueba se refiere al error para este conjunto de entrenamiento específico. De igual forma, la esperanza del error de predicción o prueba es el siguiente:

$$Err = E[L(Y, \hat{f}(X))] = E[Err_\tau] \tag{2.60}$$

En la figura 2.5 la curva sólida roja es el error promedio y por tanto una estimación del error de prueba esperado (Err).

El *error de entrenamiento* es la pérdida promedio esperada de la muestra de entrenamiento:

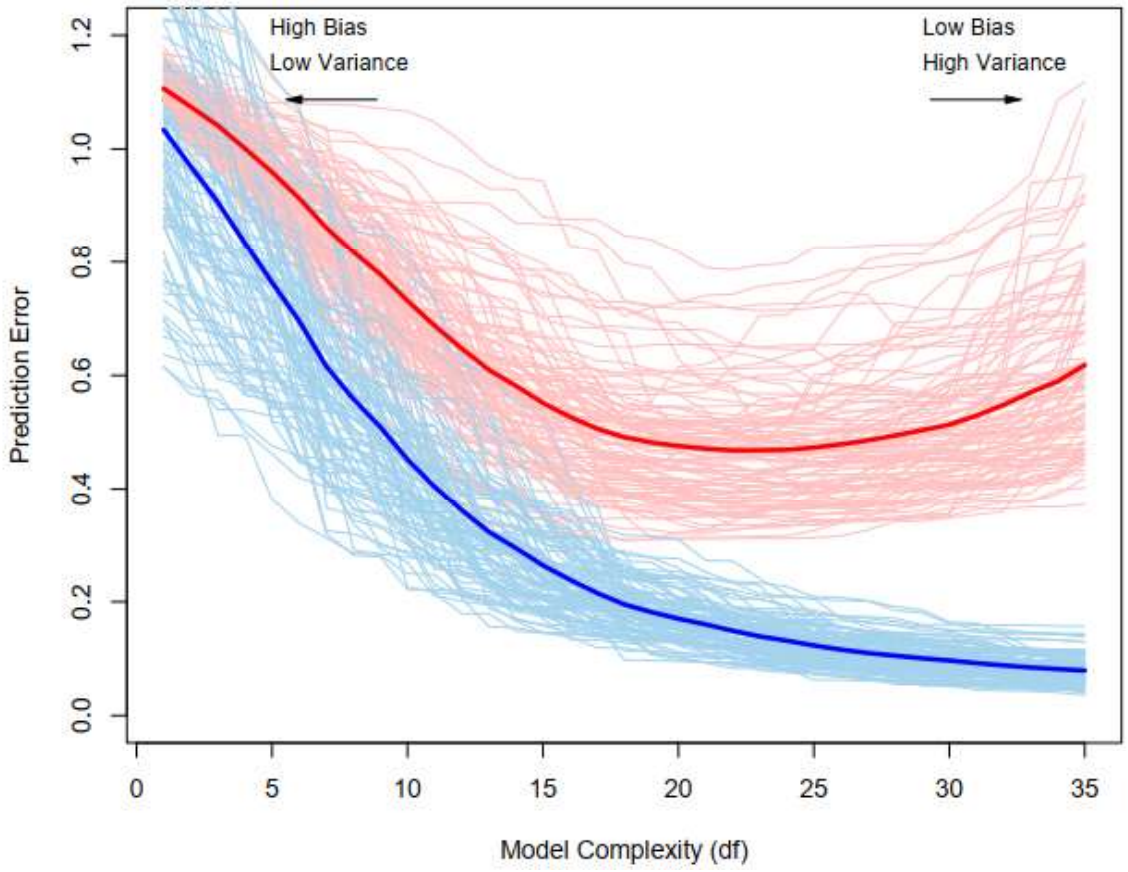
$$\overline{err} = \frac{1}{N} \sum_{i=1}^{N} L(y_i, \hat{f}(x_i)) \tag{2.61}$$

Aquí necesitamos saber la esperanza del error de prueba para el modelo estimado

[2]El sesgo mide lo lejos que se encuentra el valor estimado respecto al real de la población.

[3]La varianza se refiere a la cantidad que la estimación de la función objetivo cambiará si se utiliza diferentes datos de entrenamiento.

Figura 2.5: Comportamiento del error de la muestra de prueba y la muestra de entrenamiento a medida que aumenta la complejidad del modelo. Las curvas de color azul claro muestran el error de entrenamiento $\overline{err}$, mientras que las curvas de color rojo claro muestran el error condicional de prueba Err_{τ} para 100 conjuntos de entrenamiento de tamaño 50 cada uno. Las curvas continuas muestran el error de prueba esperado Err y el error de entrenamiento esperado $E[\overline{err}]$ (Hastie, Tibshirani y Friedman, 2017, p.220).



$\hat{f}$. A medida que el modelo se vuelve más complejo, usa más datos de entrenamiento y se adapta a estructuras subyacentes más complicadas. Entonces, el sesgo decrece, a cambio de un incremento en la varianza; por tanto, debe existir un punto donde el error de prueba esperado alcance su mínimo.

Desafortunadamente, el error de entrenamiento no es un buen estimador del error de prueba, como se puede observar en la figura 2.5. El error de entrenamiento decrece constantemente, hasta tender a cero, a medida que la complejidad del modelo aumenta; sin embargo, un modelo cuyo error de entrenamiento es cero estaría sobre ajustado a la muestra de entrenamiento, lo cual significa que no sería generalizable a un conjunto de datos diferente.

Con este concepto claro, extendible a todas las metodologías de aprendizaje, analicemos el caso donde la variable dependiente no es continua. Si la variable de res-

puesta Y es binaria, estimamos la probabilidad $p_1(X) = Pr(Y = 1|X)$, entonces $\hat{Y}(X) = arg\ max_1 \hat{p}_1(X)$. Usualmente, las funciones de pérdida son las siguientes:

$$L(Y, \hat{Y}(X)) = I(Y \neq \hat{Y}(X)) \text{ donde I es una función indicatriz.} \tag{2.62}$$

$$\begin{aligned} L(Y, \hat{p}(X)) &= -2 \sum_{k=0}^{1} I(Y = k) \log \hat{p}_k(X) \\ &= -2 \log \hat{p}_Y(X) \end{aligned} \tag{2.63}$$

donde $\log \hat{p}_Y(X)$ es el logaritmo de verosimilitud y $-2 \times \log \hat{p}_Y(X)$ suele ser referido como la desviación. De nuevo, el error de prueba aquí es $Err_\tau = E[L(Y, \hat{Y}(X))|\tau]$, el error de clasificación del modelo entrenado en τ de la población, y Err es la esperanza del error de clasificación.

El error de entrenamiento en una muestra es análogo al logaritmo de verosimilitud del modelo:

$$\overline{err} = -\frac{2}{N} \sum_{i=1}^{N} \log \hat{p}yi(x_i) \tag{2.64}$$

El logaritmo de verosimilitud puede ser usado como una función de pérdida para distintas distribuciones de probabilidad como Poisson, Gamma, exponencial, log-normal y otras. Por lo general nuestro modelo tendrá uno o varios parámetros de ajuste α, a partir de los cuales podemos escribir nuestras predicciones como $\hat{f}_\alpha(x)$.

El parámetro varía según la complejidad del modelo y es nuestro objetivo encontrar el valor α que minimiza el error; es decir, encontrar el punto donde la curva del error de prueba promedio alcanza su mínimo en la figura 2.5.

En el presente estudio destacaremos la importancia de los siguientes dos objetivos:

- **Selección del Modelo:** Estimar el desempeño de distintos modelos para elegir el mejor.
- **Evaluación del Modelo:** Una vez escogido el modelo, estimar el error de predicción (error de generalización) en un nuevo conjunto de datos.

El mejor enfoque para ambos problemas es dividir aleatoriamente el conjunto de datos en tres partes: un conjunto de entrenamiento, un conjunto de validación y un conjunto de prueba. El conjunto de entrenamiento se utiliza para ajustar los modelos; el conjunto de validación se usa para estimar el error de predicción para la selección

del modelo y el conjunto de prueba se utiliza para evaluar el error de generalización del modelo final elegido. Idealmente, el conjunto de prueba debe mantenerse en una "bóveda" y aparecer solo al final del análisis de datos.

2.2.2. Descomposición del sesgo y la varianza

Si asumimos que $Y = f(X) + \epsilon$ donde $E(\epsilon) = 0$ y $Var(\epsilon) = \sigma_\epsilon^2$, podemos derivar una expresión para la esperanza del error de predicción del modelo ajustado $\hat{f}(X)$ en un punto arbitrario $X = x_0$, usando la ecuación del error cuadrático:

$$\begin{aligned} Err(x_0) &= E[(Y - \hat{f}(x_0))^2 | X = x_0] \\ &= \sigma_\epsilon^2 + [E\hat{f}(x_0) - f(x_0)]^2 + [Ef(x_0) - \hat{f}(x_0)]^2 \\ &= \sigma_\epsilon^2 + \text{Sesgo}^2(\hat{f}(x_0) + \text{Varianza}(\hat{f}(x_0) \\ &= \text{Error aleatorio} + \text{Sesgo}^2 + \text{Varianza} \end{aligned} \tag{2.65}$$

El primer término es la varianza del punto alrededor de su media verdadera $f(x_0)$ y no puede ser evadido sin importar que tan bien estimemos $f(x_0)$, a menos que $\sigma_\epsilon^2 = 0$. El segundo término es el cuadrado del sesgo, que representa la cantidad por la cual el promedio de nuestro valor estimado difiere de su media verdadera; el último término es la varianza, la desviación cuadrada esperada de $\hat{f}(x_0)$ alrededor de su media. Generalmente, mientras más complejo sea el modelo $\hat{f}$, más bajo será el sesgo, pero más alta la varianza.

Ejemplo: Compensación Sesgo-Varianza

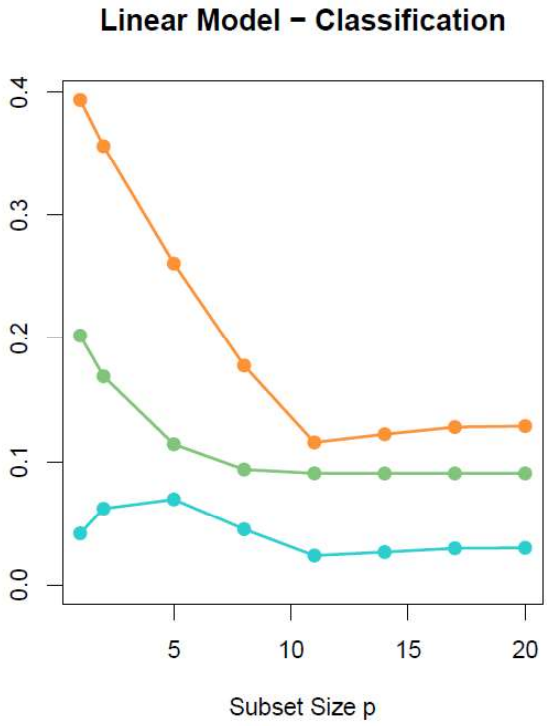
La figura 2.6 muestra el intercambio para el sesgo y la varianza para datos simulados con ochenta observaciones y veinte predicciones, uniformemente distribuidas en el hipercubo $[0,1]^{20}$. La fórmula que genera la simulación es la siguiente:

$$Y = \begin{cases} 1 & \text{si } \sum_{j=1}^{10} X_j \leq 5 \\ 0 & \text{otro caso} \end{cases} \tag{2.66}$$

Donde usamos el mejor subconjunto de tamaño p de variables significativas.

A este error de predicción lo denominaremos tasa de mal clasificados y no es obtenido como la suma del cuadrado del sesgo y la varianza. En la gráfica anterior podemos observar los siguientes fenómenos:

Figura 2.6: Esperanza del error de predicción (naranja), sesgo cuadrático (verde) y varianza (azul) para datos simulados; error de clasificación 0-1 (Hastie, Tibshirani y Friedman, 2017, p.227).



- El error de clasificación alcanza su mínimo para $p \geq 10$, aunque las mejoras más notables se encuentran cuando $p > 1$.
- Se puede observar que el sesgo y la varianza parecen interactuar al determinar el error de predicción.

¿Por qué sucede esto? Existe una explicación simple para el primer fenómeno. Supongamos que para un punto dado la probabilidad verdadera de que este sea 1 es 0.9, mientras que el valor esperado de nuestra estimación es 0.6. Entonces el sesgo al cuadrado es $(0{,}6 - 0{,}9)^2$, el cual es considerablemente grande, pero el error de predicción es cero ya que hemos tomado la decisión correcta clasificándolo como 1 (considerando como umbral el valor de 0.5). En otras palabras, el error de estimación nos deja en el lado correcto de la decisión y por tanto no nos afecta.

La idea principal es que la compensación entre sesgo y varianza se comporta diferente para funciones de pérdida 0 y 1, en comparación a la función de pérdida cuadrática del error.

2.2.3. Optimismo de la tasa de error de entrenamiento

La estimación de la tasa de error puede ser confusa porque debemos aclarar qué conjuntos son fijos y cuáles son aleatorios. Antes de continuar necesitamos algunas definiciones adicionales. Dado el conjunto de entrenamiento $\tau = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$, el error de generalización de un modelo $\hat{f}$ es:

$$Err_\tau = E_{X_0,Y_0}[L(Y^0, \hat{f}(X^0))|\tau] \tag{2.67}$$

donde el conjunto de entrenamiento τ es fijo. El punto (X^0, Y^0) es un nuevo punto de prueba, obtenido de la distribución conjunta F. Promediar el error sobre conjuntos de entrenamiento τ nos lleva a la esperanza del error:

$$Err = E_\tau E_{X_0,Y_0}[L(Y^0, \hat{f}(X^0))|\tau] \tag{2.68}$$

el cual es más susceptible al análisis estadístico. El error de entrenamiento:

$$\overline{err} = \frac{1}{N}\sum_{i=1}^{N} L(y_i,, \hat{f}(x_i)) \tag{2.69}$$

será menor que el verdadero error Err_τ ya que los mismos datos están siendo usados para estimar el modelo y calcular su error. Un método de estimación generalmente ajusta los datos de entrenamiento, y por ende el error de entrenamiento $\overline{err}$ será una estimación muy optimista del error de generalización Err_τ.

La cantidad Err_τ puede ser pensada como un error de muestra adicional ya que los datos de prueba no coinciden necesariamente con los de entrenamiento. La naturaleza del optimismo en el error $\overline{err}$ es fácilmente entendido cuando nos enfocamos en el error dentro de la muestra:

$$Err_{in} = \frac{1}{N}\sum_{i=1}^{N} E_{Y^0}[L(Y_i^0, \hat{f}(x_i))|\tau] \tag{2.70}$$

La notación Y^0 indica que observamos N nuevos valores de respuesta de cada uno de los puntos de entrenamiento x_i con $i = 1, 2, ..., N$. Definimos el optimismo como la diferencia entre Err_{in} y el error de entrenamiento $\overline{err}$:

$$op \equiv Err_{in} - \overline{err} \tag{2.71}$$

Esto es generalmente positivo ya que $\overline{err}$ está usualmente sesgado hacia abajo como un estimador del error de predicción. Finalmente, el error optimista promedio es la esperanza del error optimista sobre el conjunto de entrenamiento:

$$\omega \equiv E_y(op) \tag{2.72}$$

Aquí los predictores en los datos de entrenamiento son fijos, y la esperanza se calcula sobre los valores de salida del conjunto de entrenamiento, por tanto vamos a usar la notación E_y en lugar de E_τ. Usualmente solo podemos estimar el valor esperado ω en lugar de op, en la misma manera en la que podemos estimar la esperanza del error Err en lugar del error condicional Err_τ.

Se puede demostrar que para las funciones de pérdida en general, y en particular para la función de pérdida 0-1:

$$\omega = \frac{2}{N}\sum_{i=1}^{N} Cov(\hat{y}_i, y_i) \tag{2.73}$$

donde Cov representa la covarianza. Así la cantidad por la cual $\overline{err}$ subestima el verdadero valor del error depende de qué tan fuerte es la relación entre y_i y los efectos sobre su propia predicción $\hat{y}_i$. Mientras más ajustados estén los datos, mayor será la covarianza entre esos datos, y por tanto mayor será el optimismo. Para la función de pérdida 0-1, $\hat{y}_i \in \{0, 1\}$ es la clasificación en x_i, y para la pérdida de entropía, $\hat{y}_i \in [0, 1]$es la probabilidad de clasificar como clase 1 a x_i.

En resumen, tenemos la relación importante:

$$E_y(Err_{in}) = E_y(\overline{err}) + \frac{2}{N}\sum_{i=1}^{N} Cov(\hat{y}_i, y_i) \tag{2.74}$$

Esta expresión se simplifica si $\hat{y}_i$ es obtenido por una regresión lineal con d variables de entrada o funciones base:

$$\sum_{i=1}^{N} Cov(\hat{y}_i, y_i) = d\sigma_\epsilon^2 \tag{2.75}$$

para el modelo con error aditivo $Y = f(X) + \epsilon$, y así:

$$E_y(Err_{in}) = E_y(\overline{err}) + 2\frac{d}{N}\sigma_\epsilon^2 \tag{2.76}$$

Esta última ecuación es la base de la definición para el número efectivo de parámetros. El optimismo crece linealmente con el número d de variables de entrada que usemos, pero decrece según como el número de datos de entrenamiento incremente. Esta función tiene una forma similar cuando la variable respuesta Y es binaria.

Una manera obvia de estimar el error de predicción es estimar el optimismo y luego sumarlo al error de entrenamiento $\overline{err}$. El método descrito en la siguiente sección (Criterio de Información de Akaike, AIC) trabaja de esta manera para una clase especial de estimaciones que son lineales en sus parámetros.

En contraste, los métodos de validación cruzada y bootstrap son estimaciones directas del error de muestras adicionales Err. Estas herramientas generalizadas pueden ser usadas con cualquier función de pérdida y con técnicas de estimación no lineales.

El error en la muestra no es usualmente de interés ya que futuros valores de las variables explicativas no necesariamente coincidirán con sus valores en los datos de entrenamiento. Sin embargo, para comparación entre modelos, el error en la muestra es conveniente y usualmente nos guía a una selección de modelo eficaz. La razón yace en que el tamaño relativo del error es lo que importa.

2.2.4. Estimación del error de predicción en la muestra

El artículo seminal del criterio de información de Akaike (AIC) fue propuesto en Akaike (1973). La forma general para la estimación del error en la muestra es:

$$\hat{Err}_{in} = \overline{err} + \hat{\omega} \tag{2.77}$$

donde $\hat{\omega}$ es la estimación del optimismo promedio.

Usando la expresión 2.76, aplicable cuando se estiman d parámetros bajo una función de pérdida de error cuadrático, se llega a una versión del estadístico C_p:

$$C_p = \overline{err} + 2\frac{d}{N}\hat{\sigma_\epsilon^2} \tag{2.78}$$

donde $\hat{\sigma_\epsilon^2}$ es un estimador de la varianza del ruido, obtenida del error cuadrático medio de un modelo con bajo sesgo. Usando este criterio ajustamos el error de entrenamiento por un factor proporcional al número de funciones base usadas.

El criterio de información de Akaike es un estimador similar pero más generalmente aplicable de Err_{in} cuando se usa una función de pérdida del logaritmo de la verosi-

militud. Este se basa en una relación similar a 2.76 que se mantiene asintóticamente cuando $N \to \infty$:

$$-2E[\log Pr_{\hat{\theta}}(Y)] \approx -\frac{2}{N}E[\text{loglik}] + 2\frac{d}{N} \tag{2.79}$$

Aqui $Pr_\theta(Y)$ es una familia de densidades para Y (conteniendo la "verdadera" densidad), $\hat{\theta}$ es el estimador de máxima verosimilitud de θ, y loglik es el logaritmo de verosimilitud maximizado:

$$\text{loglik} = \sum_{i=1}^{N} \log Pr_{\hat{\theta}}(y_i) \tag{2.80}$$

Para el modelo de regresión logística, usando el logaritmo de verosimilitud binomial tenemos:

$$AIC = -2\frac{2}{N}\text{loglik} + 2\frac{d}{N} \tag{2.81}$$

Para usar este criterio en la selección de un modelo, simplemente escogemos aquel modelo que nos de el menor AIC sobre el conjunto de modelos considerados.

Dado un conjunto de modelos $f_\alpha(x)$ indexados por un parámetro de ajuste α, denotados por $\overline{err}(\alpha)$ y $d(\alpha)$ el error de entrenamiento y el número de parámetros para cada modelo. Entonces para este conjunto de modelos definimos:

$$AIC(\alpha) = \overline{err}(\alpha) + 2\frac{d(\alpha)}{N}\hat{\sigma_\epsilon^2} \tag{2.82}$$

La función $AIC(\alpha)$ provee un estimador de la curva del error de prueba, y podemos encontrar el parámetro de ajuste α que la minimice. Nuestro modelo final elegido es $f_\alpha(x)$. Nótese que si las funciones base son escogidas de forma adaptativa, 2.75 no se mantienen. Por ejemplo, si tenemos un total de p variables de entrada, y escogemos el mejor modelo lineal con $d < p$ variables, el optimismo sobrepasará ($\frac{2d}{N}\sigma_\epsilon^2$). Puesto de otra manera, al escoger el modelo mejor ajustado con d variables, el ajuste del número efectivo de parámetros es mayor a d.

La figura 2.7 muestra al criterio AIC en acción, un modelo de regresión logística es usado para predecir la variable dependiente con función de coeficientes $\beta(f) = \sum_{m=1}^{M} h_m(f)\theta_m$, una expansión en M funciones base por splines. Para cualquier valor de M, una base natural de splines cúbicas es usada para h_m, con nudos escogidos uniformemente sobre el rango de frecuencias (así $d(\alpha) = d(M) = M$). Usando el AIC

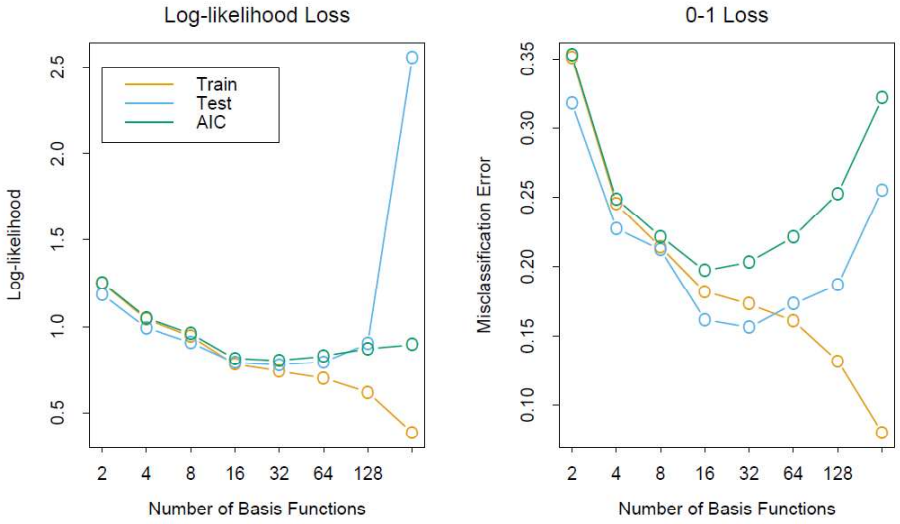
se selecciona el número de funciones bases que aproximadamente minimizarán $Err(M)$ para la función de pérdida 0-1.

La fórmula:

$$\frac{2}{N}\sum_{i=1}^{N} Cov(\hat{y}_i, y_i) = \frac{2d}{N}\sigma_\epsilon^2 \tag{2.83}$$

se manntiene para modelos lineales con errores aditivos y pérdida de error cuadrático, y aproximadamente para modelos lineales y con logaritmo de verosimilitud.

Figura 2.7: AIC usado para la selección de modelos. Los coeficientes de la función de regresión logística $\beta(f) = \sum_{m=1}^{M} h_m(f)\theta_m$ está modelada con una expansión en M funciones base por splines. En el panel de la izquierda podemos ver el estadístico AIC usado para estimar Err_{in} usando la función de pérdida del logaritmo de verosimilitud. Está incluido un estimador de Err en base a la independencia de la muestra de prueba. El desempeño es bueno excepto para el caso sobre parametrizado ($M = 266$ parámetros para $N = 1000$ observaciones). En el panel derecho se muestra lo mismo para la función de pérdida 0-1 (Hastie, Tibshirani y Friedman, 2017, p.232).



2.2.5. Validación cruzada

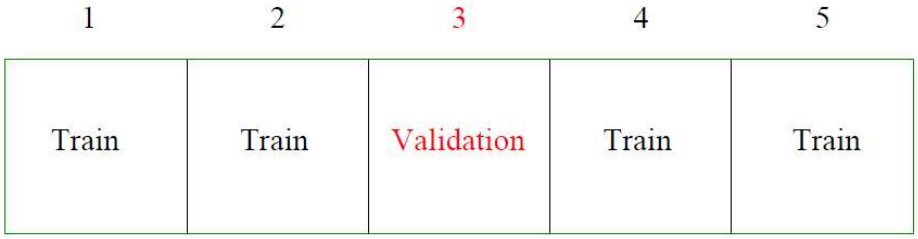
Las principales referencias para validación cruzada son Stone (1974), Stone (1977) y Allen (1974).

Probablemente la forma más simple y ampliamente utilizada para estimar el error de predicción es la validación cruzada. Este método estima directamente el error en varias muestras $Err = E[L(Y, \hat{f}(X))]$, el error de generalización promedio cuando el modelo $\hat{f}(X)$ es aplicado a una muestra de prueba independiente de la distribución

conjunta de X e Y. Se espera que la validación cruzada estime el error condicional con el conjunto de datos de entrenamiento τ fijo; sin embargo, las estimaciones de la validación cruzada estiman bien únicamente la esperanza del error de predicción.

Validación cruzada K-Fold

La Validación cruzada K-Fold usa parte de los datos disponibles para ajustar el modelo, y una parte diferente para probarlo. Para ello, dividimos los datos en K partes aproximadamente iguales; por ejemplo, cuando $K = 5$ la división es la siguiente:



Para la k-ésima parte (tercera de arriba) ajustamos el modelo en las otras $K-1$ partes de los datos y calculamos el error de predicción del modelo ajustado cuando predecimos sobre la k-ésima parte de los datos. Hacemos esto para $k = 1, 2, ..., K$ y combinamos las k estimaciones del error de predicción.

Sean $k : \{1, ..., N\} \to \{1, ..., K\}$ una función indexadora que indica la partición a donde la observación i es asignada aleatoriamente y $\hat{f}^{-k}(x)$ la función de ajuste calculada con la k-ésima parte de los datos removidos, entonces el estimador de validación cruzada para el error de predicción es el siguiente:

$$CV(\hat{(f)}) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}^{-k(i)}(x_i)) \tag{2.84}$$

Usualmente se escoge el valor $K = 5$ o $K = 10$. El caso $K = N$ es conocido como validación cruzada *dejar-uno-afuera* (leave-one-out). En este caso $k(i) = i$, y para la i-ésima observación, el ajuste del modelo es calculado en todos los datos excepto el i-ésimo.

Dado un conjunto de modelos $f(x, \alpha)$ indexados por un parámetro de ajuste α, denotemos como $\hat{f}^{-k}(x, \alpha)$ al α-ésimo modelo ajustado con la k-ésima parte de los datos removidos. Entonces para este conjunto de modelos definimos:

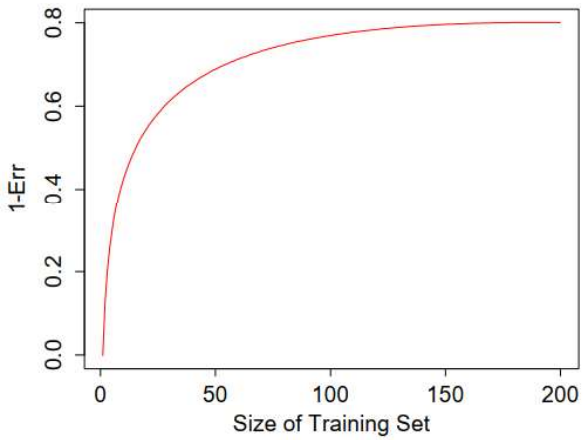
$$CV(\hat{(f)}, \alpha) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}^{-k(i)}(x_i, \alpha)) \tag{2.85}$$

La función $CV(\hat{(}f), \alpha)$ provee un estimador de la curva del error de prueba, y encontraremos el parámetro de ajuste α que la minimice. Nuestro modelo final escogido es $\hat{f}(x, \alpha)$, el cual usaremos para ajustar todos los datos.

Deberíamos preguntarnos qué cantidad de validaciones cruzadas K-fold estimar. Con $K = 5$ o $K = 10$ podríamos pensar que estima la esperanza del error Err, dado que los conjuntos de entrenamiento en cada fold son bastante diferentes del conjunto de entrenamiento original. Por otro lado, si $K = N$ podríamos pensar que la validación cruzada estima el error condicional Err_τ. Resulta que la validación cruzada únicamente estima bien la esperanza del error Err, como lo veremos más adelante.

¿Qué valor debemos escoger para K? Con $K = N$, el estimador de la validación cruzada es aproximadamente insesgado por el verdadero error de predicción (esperado), pero puede tener alta varianza porque los N "conjuntos de entrenamiento" son muy similares uno con otro. El esfuerzo computacional también es considerable, ya que requiere N aplicaciones del modelo de aprendizaje.

Figura 2.8: Curva de aprendizaje hipotética para un clasificador en una tarea dada: una gráfica de 1-Err versus el tamaño del conjunto de entrenamiento N. Con un conjunto de datos de 200 observaciones, 5 validaciones cruzadas usarían conjuntos de entrenamiento de tamaño 160, que se comportarían de manera muy similar al conjunto completo. Sin embargo, con un conjunto de datos de 50 observaciones, la validación cruzada quíntuple (K=5) usaría conjuntos de entrenamiento de tamaño 40, y esto resultaría en una sobre estimación del error de predicción (Hastie, Tibshirani y Friedman, 2017, p.243).



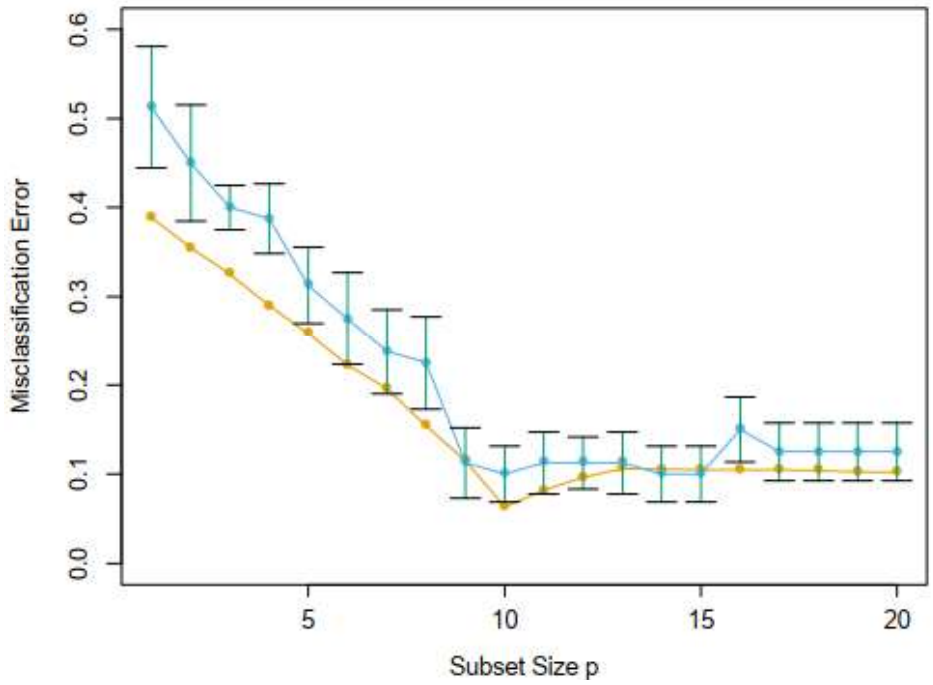
Por otro lado, con K = 5, por ejemplo, la validación cruzada tiene menor varianza. Pero el sesgo podría ser un problema, dependiendo de cómo el desempeño del modelo de aprendizaje varía con respecto al tamaño del conjunto de entrenamiento. La figura 2.8 muestra una "curva de aprendizaje" hipotética para un clasificador en una tarea dada. El desempeño del clasificador mejora según incrementa el tamaño del conjunto

de entrenamiento hacia 100 observaciones; incrementar el número a 200 solo trae un pequeño beneficio. Si nuestro conjunto de entrenamiento tuviera 200 observaciones, la validación cruzada quíntuple (K=5) estimaría el desempeño del clasificador sobre conjuntos de entrenamiento de tamaño 160, que de la figura 2.8 es prácticamente el mismo que el desempeño para el conjunto de entrenamiento de tamaño 200. Entonces la validación cruzada no sufrirá de mucho sesgo. De cualquier forma si el conjunto de entrenamiento tuviera 50 observaciones, la validación cruzada quíntuple estimaría el desempeño de nuestro clasificador sobre conjuntos de entrenamiento de tamaño 40, y de la figura, eso sería una sub estimación de $1 - Err$. Por tanto, como un estimador de Err la validación cruzada estará sesgada hacia arriba.

Para resumir, si la curva de aprendizaje tuviera una pendiente considerable en el tamaño del conjunto de entrenamiento dado, la validación cruzada con $K = 5$ o $K = 10$ sobre estimará el verdadero error de predicción. Si este sesgo es un inconveniente en la práctica depende del objetivo. Por otro lado, la validación cruzada dejando uno afuera tiene bajo sesgo pero puede tener alta varianza.

La figura 2.9 muestra el error de predicción y la curva de validación cruzada con $K = 10$ estimadas a partir de un único conjunto de entrenamiento, del escenario planteado en la figura 2.6. Este es un problema de clasificación binario, usando un modelo lineal con las mejores variables regresoras de un subconjunto de tamaño p. Se muestran las barras de los errores estándar, que son los errores estándar de las tasas de clasificación individuales para cada una de las 10 partes. Ambas curvas alcanzan el mínimo en $p = 10$, aunque la curva de validación cruzada es más plana más allá de $p = 10$. Usualmente la regla del "un error estándar" se usa con la validación cruzada, en la cual escogemos el modelo más parsimonioso cuyo error no sea mayor que una desviación estándar sobre el error del mejor modelo. Aquí parece que el mejor modelo, con $p = 9$ predictores será escogido, mientras el verdadero modelo usa $p = 10$.

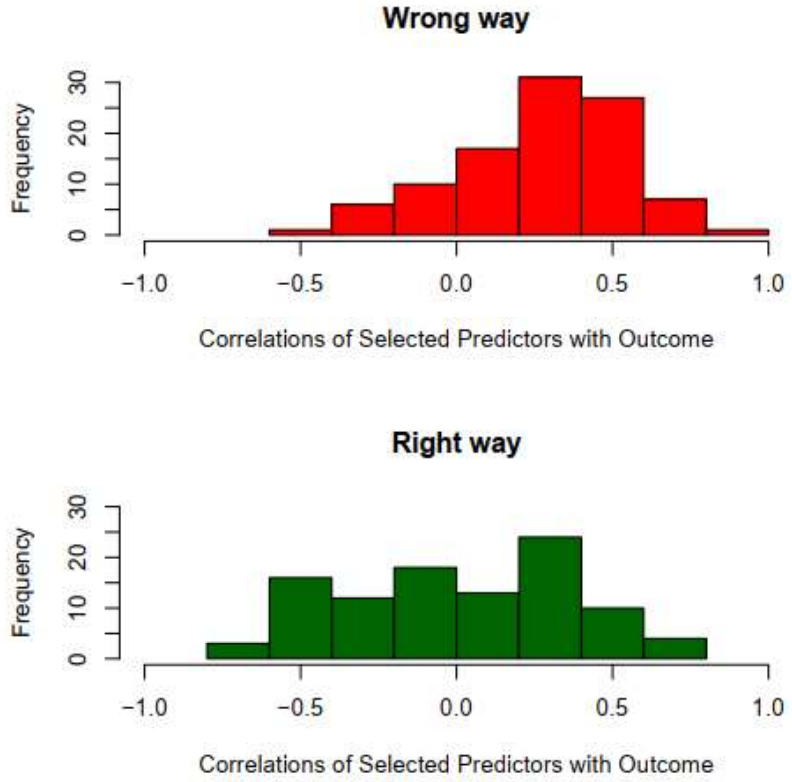
Figura 2.9: Error de predicción (naranja) y la curva de validación cruzada con $K = 10$ (azul) estimadas en un solo conjunto de entrenamiento, del escenario planteado en la figura 2.6 (Hastie, Tibshirani y Friedman, 2017, p.244).



A continuación se muestra la manera correcta de llevar a cabo la validación cruzada.

1. Dividir aleatoriamente la muestra en $K = 10$ grupos de validación cruzada.
2. Para cada grupo $k = 1, 2, ..., K$:
   - *a*) Encontrar un conjunto de "buenos" predictores que muestren una correlación univariada considerable con respecto a la variable dependiente usando todas las observaciones, excepto las que están en el grupo k.
   - *b*) Usando solo el subconjunto de predictores, construir un clasificador multivariado, usando todas las observaciones excepto las que están en el grupo k.
   - *c*) Usar el clasificador para predecir la variable dependiente en las observaciones del grupo k.

Figura 2.10: Los histogramas muestran la correlación de las etiquetas de clase, en 10 muestras escogidas aleatoriamente, con 100 predictores escogidos usando la forma incorrecta (rojo) y correcta (verde) de la validación cruzada (Hastie, Tibshirani y Friedman, 2017, p.246).



Las estimaciones del error del paso 2.c son acumulados sobre los 10 grupos, para producir el estimador de validación cruzada del error de predicción. El panel inferior de la figura 2.10 muestra las correlaciones de las etiquetas de clase con los 100 predictores escogidos en el paso 2.a del procedimiento correcto, sobre las muestras de un grupo k. Vemos que las correlaciones se promedian alrededor de cero, como deberían.

2.2.6. Algoritmo genético para selección del mejor modelo lineal

En otro tema, y por el lado de la regresión logística, cuando se modelan este tipo de problemas, se estiman varios modelos en búsqueda de un modelo parsimonioso que envuelva un subconjunto de variables de entrada adecuado. Es por ello que para los objetivos de este estudio y para realizar inferencia sobre los determinantes de la probabilidad de reprobación del curso de nivelación de la Escuela Politécnica Nacional, se ha decidido utilizar una aproximación a través de un *algoritmo genético* disponible en la librería *glmulti* y explicada por Calcagno (2019).

Cuando pensamos en la selección del mejor modelo, la primera idea en mente sería una aproximación a "fuerza bruta", es decir, estimar todos los modelos posibles y luego seleccionar de entre ellos, el mejor, a partir de algún criterio de información. Lamentablemente este ejercicio puede ser computacionalmente inviable debido a que el número de modelos a estimar crece a razón de 2^p, donde p es el número de variables de entrada (sin tomar en cuenta posibles términos de interacción). Es decir, con $p = 20$ variables tendríamos que estimar $2^{20} = 1'048{,}576$ modelos, lo que nos tomaría alrededor de 72 horas en una computadora personal. Para ello, los autores proponen un algoritmo genético (genetic algorithm - GA) que explora solo un subconjunto de todos los posibles modelos, con sesgo hacia los mejores, que gracias a un criterio de selección se vuelve mucho más rápido (para las mismas 20 variables el procedimiento toma alrededor de 4 horas). Este algoritmo genético es eficiente explorando espacios discretos y puede converger aún con problemas muy complicados.

Específicamente, lo que hace el algoritmo es codificar vectores de ceros y unos para indicar qué términos están presentes y cuáles no dentro de la regresión. A este vector se lo conoce como *cromosoma* y atravesará un proceso de *evolución adaptativa*.

Entonces, el algoritmo genético mantiene una población de modelos, y en cada generación (i.e. iteración) estimará modelos con su respectivo criterio de información (para este caso, el criterio de información de Akaike AIC), que luego usará para computar la *aptitud* de cada uno, denotada por ω:

$$\omega_i = \exp(-(AIC_i - AIC_{mejor})) \tag{2.86}$$

donde AIC_{mejor} es el mejor AIC en la actual población de modelos. Un AIC mayor significa menor aptitud del modelo y por términos de rendimiento, se mantiene solo una lista finita de los mejores modelos encontrados hasta el momento.

En cada generación se producen modelos de tres maneras: (1) reproducción asexual, (2) reproducción sexual e (3) inmigración. Un modelo producido por reproducción asexual es simplemente una copia de su padre (tomado aleatoriamente de la generación previa con una probabilidad proporcional a su aptitud). En el modelo por reproducción sexual cada modelo tendrá madre y padre con sus "cromosomas" combinados. Por último, el modelo producido por inmigración tiene en cambio un estado asignado aleatoriamente, lo cual evita que el algoritmo quede atascado en un mínimo local.

Finalmente, el algoritmo genético se detendrá cuando el AIC caiga bajo un umbral específico, lo cual se revisa cada veinte generaciones. Si tal condición se cumple, el algoritmo converge y se toman los resultados.

2.3. Evaluación del desempeño de modelos de clasificación

Finalmente, en esta sección analizaremos las medidas existentes para evaluar los resultados de un proceso de modelización para un problema de clasificación. El objetivo es cuantificar de alguna manera la calidad del ajuste de la solución que hayamos encontrado y hacer posible la comparación entre varios modelos, sean de la misma metodología o no.

Cuando se evalúan modelos de clasificación, las medidas de desempeño se calculan comparando las predicciones generadas por este para la muestra de prueba o validación, contra las clases verdaderas del mismo conjunto. Con esta consideración, disponemos de las siguientes medidas, sugeridas por Gironés Roig y col. (2017).

2.3.1. Matriz de confusión

La matriz de confusión es una tabla de doble entrada que permite observar los errores cometidos por el modelo de clasificación entrenado. Esta matriz es conocida además como la matriz de errores.

La figura 2.11 presenta la matriz de confusión para un caso de clasificación binaria, donde P representa la clase 1 y N la clase 0.

Figura 2.11: Matriz de confusión binaria

		Clase predicha	
		P	N
Clase verdadera	P	TP	FN
	N	FP	TN

Esta matriz nos muestra el número de observaciones correcta e incorrectamente clasificadas, donde a cada celda se le asigna una etiqueta distinta:

- Verdaderos positivos (True Positive, TP): es el número de clasificaciones correctas para la clase 1.
- Verdaderos negativos (True Negative, TN): es el número de clasificaciones correctas para la clase 0.

- Falsos negativos (Falso Negative, FN): es el número de clasificaciones incorrectas para la clase 1 ya que fueron clasificadas como 0.

- Falsos positivos (Falso Positive, FP): es el número de clasificaciones incorrectas para la clase 0 ya que fueron clasificadas como 1.

De estas celdas se derivan algunas métricas de desempeño que permiten cuantificar la bondad de ajuste del modelo, y son:

- Error de clasificación (Err): Ya revisada anteriormente, es la suma de predicciones incorrectas sobre el número total de predicciones:

$$Err = \frac{FP + FN}{FP + FN + TP + TN} \tag{2.87}$$

- Exactitud (Accuracy, Acc): Es el número de predicciones correctas sobre el número total de predicciones:

$$Acc = \frac{TP + TN}{FP + FN + TP + TN} = 1 - Err \tag{2.88}$$

.

- Tasa de verdaderos positivos (True Positive Rate, TPR): Es una medida del error en los falsos positivos, dada por:

$$TPR = \frac{TP}{FN + TP} \tag{2.89}$$

.

- Tasa de veraderos negativos (False Positive Rate, FPR): Es una medida del error en los falsos negativos, dada por:

$$FPR = \frac{FP}{FP + TN} \tag{2.90}$$

.

- Precisión (Pre): Mide el rendimiento relacionado con las tasas de verdaderos positivos y negativos, y expresa la proporción de puntos que nuestro modelo dice que son relevantes, y realmente lo son:

$$Pre = \frac{TP}{TP + FP} \tag{2.91}$$

.

- Recall o Sensibilidad (Rec, Sen): Se corresponden con la tasa de verdaderos positivos y expresan la capacidad del modelo de encontrar todos los puntos de interés en un conjunto de datos:

$$Rec = Sen = \frac{TP}{FN + TP} \tag{2.92}$$

.

- Especificidad (Spe): Se define como la tasa de observaciones correctamente clasificadas como clase 0 respecto a todas las instancias de clase 0. Viene dada por:

$$Spe = \frac{TN}{TN + FP} = 1 - FPR \tag{2.93}$$

.

- F1 Score (F1): Este indicador se obtiene al combinar las medidas de precisión y recall, que crece y tiende a 1 a medida que ambas métricas se vuelven "perfectas". Está dada por:

$$F1 = 2 \times \frac{Pre \times Rec}{Pre + Rec} \tag{2.94}$$

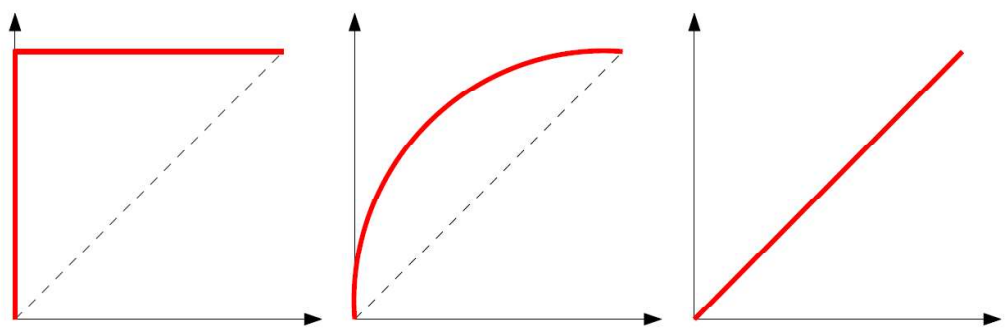
.

El uso de estas medidas dependerá del problema que estemos analizando.

2.3.2. Curvas ROC

Una curva *Receiver Operating Characteristic* o ROC mide el rendimiento respecto a los falsos positivos FP y verdaderos positivos TP. Su diagonal se interpreta como un modelo generado aleatoriamente, mientras que valores inferiores a ella se consideran peores que una estimación aleatoria de nuevos datos.

Figura 2.12: Ejemplos de curvas ROC



(a) AUC = 1 **(b)** AUC ≈ 0.8 **(c)** AUC = 0.5

Bajo esta métrica, en la figura 2.12 se muestran tres casos. Un clasificador perfecto ocuparía el primer panel de la gráfica, con una tasa de verdaderos positivos (TP) de uno y una tasa de falsos positivos (FP) igual a cero. A partir de la curva ROC se calcula el área bajo la curva o (*area under de curve*, AUC) que permite caracterizar el rendimiento de nuestro modelo. Así, en el segundo panel se presenta una curva ROC con un rendimiento bueno ya que no alcanza los niveles de un clasificador perfecto, pero tampoco es igual al modelo generado completamente aleatorio. En la práctica por lo general se dice que con un AUC entre 0.5 y 0.6 se tiene un rendimiento malo, de 0.6 a 0.75 un rendimiento regular, de 0.75 a 0.9 un rendimiento bueno, de 0.9 a 0.97 un rendimiento muy bueno, y superior a 0.97, un rendimiento excelente.

Capítulo 3

Metodología Analítica y Resultados

En este capítulo nos centraremos en describir la metodología empleada en la construcción del modelo, el cual deberá suplir dos necesidades; la de predecir el estado de aprobación o reprobación de los estudiantes del Curso de Nivelación y presentar características descriptivas y factores determinantes del rendimiento académico de los mismos. Es decir, que esta sección constará de un modelo predictivo, en el cual se conocen todos los valores de los atributos de los estudiantes, a excepción del resultado del rendimiento académico; y de un modelo inferencial, deseable desde el punto de vista de la extracción de conocimiento.

3.1. Definición de las variables del modelo

La Dirección de Gestión de la Información y Procesos (DGIP) almacena, gestiona y administra los recursos informáticos y tecnológicos de la Escuela Politécnica Nacional a través del Sistema de Administración e Información Estudiantil (SAEW); ésta es la fuente subyacente que sirve como insumo de las variables que se considerarán en el presente estudio.

Como se mencionó en el Capítulo 2, dividiremos aleatoriamente el conjunto de datos en tres partes: un conjunto de entrenamiento, un conjunto de validación y un conjunto de prueba como se describe a continuación:

70 % de los años 2017 y 2018, semestres A y B

30 % de los años 2017 y 2018, semestres A y B | Periodo 2019-A

Es decir, para el conjunto de entrenamiento y validación se tomará aleatoriamente

el 70 % y 30 % de los periodos 2017-A, 2017-B, 2018-A y 2018-B respectivamente; y para el conjunto de prueba se tomará el periodo 2019-A.

3.1.1. Variable Dependiente

La variable dependiente Y representará el rendimiento académico del estudiante en el Curso de Nivelación (CN), por tanto, será una variable binaria que tomará el valor de 1 para los individuos etiquetados como Reprueba y 0 para los estudiantes etiquetados como Aprueba:

$$Reprueba = \begin{cases} 1 & \text{si el estudiante reprueba,} \\ 0 & \text{si no.} \end{cases} \tag{3.1}$$

3.1.2. Variables Independientes

En el rendimiento académico interactúan elementos multicausales, tanto sociodemográficos, psicosociales, pedagógicos, institucionales y socioeconómicos; la combinación y ponderación de estos factores no siempre es la misma, razón por la cual este es un tema que amerita constante investigación (Montero, Villalobos y Valverde, 2007). Para el presente análisis, consideraremos como el conjunto de variables explicativas o independientes a las mostradas en la Tabla 3.1. Podemos agruparlas de la siguiente forma:

- **Sociodemográficas**: Sexo (2), Estado civil (4), Etnia(5) y Edad (10 y 22) .
- **Bagaje y familiares:** Número de miembros en el núcleo familiar (20), Ingreso mensual (21), Tipo colegio (7) y Residencia (9).
- **Académicas**: Promedio ponderado del primer bimestre (17), Calificación de postulación (19), Calificación del primer bimestre (18), Número de materias tomadas (16), Número de matrícula (12), Segmento poblacional (8) y Número de créditos[1] por materia (3).
- **Institucionales:** Curso de nivelación (11), Semestre del año (13), Jornada (6), Materia (15) y Carrera a la que aspira (14).

[1]Un crédito equivale a dos horas de clases.

Tabla 3.1: Descripción de variables

N	Variable	Descripcion
1	Reprueba	A: Aprueba o F: Reprueba
2	Sexo	Femenino o Masculino
3	NumeroCreditos	4, 6 u 8 créditos
4	EstadoCivil	S: Solero/a. C: Casado/a, U: Unión libre o D: divorciado/a
5	Etnia	Blanco/a, Indígena. Mestizo/a, Montubio/a, Mulato/a, Negro/a u Otra
6	Jornada	Matutina o Vespertina
7	TipoColegio	Fiscal, Fiscomisional, Municipal, Particular o Extranjero
8	SegmentoPoblacional	Acción afirmativa, Grupo de alto rendimiento. Merito Territorial o Población general
9	Residencia	Quito, Otra o Extranjero/a
10	Edad	Menor de 18. Entre 18 y 23 o Mayor de 23
11	CursoNivelacion	Ingeniería, Ciencias y Ciencias Administrativas o Nivel Técnológico Superior
12	NumeroMatricula	Matrícula 1, 2 o 3
13	Semestre	Semestre A o B
14	CarreraAspira	Oferta académica del EPN
15	Materia	Asignatura que cursa el estudiante
16	NumeroMaterias	Número de materias que está cursando el estudiante
17	PromedioPonderadoBimestre1	Calificaciones del primer bimestre promediadas y ponderadas por el número de créditos
18	Calificacion1	La calificación del primer bimestre por materia
19	CalificacionPostulacion	Puntaje obtenido en la prueba Ser Bachiller
20	NumeroMiembrosFamilia	Número de miembros en el núcleo familiar del estudiante.
21	IngresoMensual	Ingreso mensual familiar reportado por el estudiante
22	edad	Edad del estudiante en el periodo cursado

Note:
Fuente: Sistema de Administración e Información Estudiantil (SAEW). 2019.

3.2. Construcción del modelo predictivo

La construcción del modelo se realizará mediante aprendizaje supervisado, el cual parte de casos particulares (experiencias) y obtiene casos generales (modelos o reglas) (Hunt, 1993). El aprendizaje supervisado no depende de un experto para "deducir" una regla (modelo o hipótesis) que sirva para describir el conocimiento; por tanto, la ventaja del aprendizaje supervisado es que puede automatizarse (Kononenko, Bratko y Kukar, 1998). Ya que la variable a predecir *Reprueba* es binaria, emplearemos un modelo de aprendizaje supervisado para clasificación.

El algoritmo de *Gradient Boosting* (GB) del Capítulo 2 es el escogido como modelo de predicción de la variable *Reprueba*. El GB es un algoritmo popular del aprendizaje automático que ha demostrado ser exitoso en muchos campos y es uno de los métodos líderes para ganar las competencias Kaggle[2]. La idea principal del GB es que construye un conjunto de árboles sucesivos poco profundos y débiles, donde cada árbol está aprendiendo y mejorando con respecto al anterior. Cuando se combinan estos árboles sucesivos producen un poderoso clasificador que a menudo es difícil de superar con otros algoritmos.

[2]Kaggle es una plataforma con recursos para aprender Machine Learning y Ciencia de Datos. Cuenta con varias competiciones de Machine Learning que tienen más de 1 millón de dólares en premios y cientos de competidores.

Ventajas y desventajas del algoritmo Gradient Boosting Machine

De la implementación del algortimo GB se pueden destacar los siguiente:

Ventajas:

- El algoritmo proporciona una buena precisión predictiva.
- Mucha flexibilidad: se puede optimizar las diferentes funciones de pérdida y los hiperparámetros para que la función se ajuste de mejor forma.
- No se requiere procesamiento previo de datos: a menudo funciona muy bien con valores categóricos y numéricos tal como están.
- Maneja datos faltantes: no se requiere imputación. Es recomendado en bases de datos con datos atípicos.

Desventajas:

- El algoritmo GB continuará mejorando para minimizar todos los errores. Esto puede enfatizar a los valores atípicos y causar un sobreajuste. Se incorporó validación cruzada para neutralizar esta desventaja.
- Computacionalmente caro: el algoritmo GB a menudo requiere muchos árboles (>1000) que pueden ser exhaustivos en tiempo y memoria.
- La alta flexibilidad da como resultado muchos parámetros que interactúan e influyen fuertemente en el comportamiento del enfoque (número de iteraciones, profundidad del árbol, parámetros de regularización, etc.). Esto requiere una búsqueda exhaustiva durante el ajuste.
- No es intuitivamente interpretable, para lo cual ajustaremos otro modelo para este objetivo.

3.2.1. Implementación del Algoritmo Gradient Boosting Machine.

Haciendo uso del software estadístico R[3] (Team, 2019) y la librería *gbm* (Greenwell y col., 2019) implementamos el modelo de predicción.

[3]R es un entorno de software libre para computación estadística y gráficos.

Gbm

La librería *gbm* de R es una implementación para el algoritmo AdaBoost de Freund y Schapire, y la Gradient Boosting Machine de Friedman. Las características de la librería son las siguientes:

- Admite hasta 1024 niveles de variables categóricas.
- Soporta árboles de clasificación y regresión.
- Puede incorporar muchas funciones de pérdida.
- Se proporciona un estimador del número óptimo de iteraciones.
- La validación cruzada interna se puede paralelizar a todos los núcleos de la máquina.

Consideraciones de la Implementación:

El código de la implementación se encuentra dsponible en el Apéndice B, a continuación se muestran los detalles y obervaciones del mismo.

La librería *gbm* tiene dos funciones principales de entrenamiento: *gbm::gbm* y *gbm:: gbm.fit*. La diferencia principal es que *gbm::gbm* usa la interfaz de fórmula para especificar su modelo, mientras que *gbm:: gbm.fit* requiere las matrices X e Y separadas. Cuando se trabaja con muchas variables, es más eficiente usar la matriz en lugar de la interfaz de la fórmula.

Para determinar los parámetros óptimos primero se ha ejecutado un loop para encontrar los que hacen mínima la tasa de mal clasificados. La configuración que vamos a utilizar del *gbm* tiene una tasa de aprendizaje (*shrinkage*) de 0.001. Esta es una tasa de aprendizaje muy pequeña y generalmente requiere una gran cantidad de árboles para encontrar el MSE mínimo o la menor tasa de mal clasificados. Sin embargo, *gbm* usa un número predeterminado de árboles de 100, que rara vez es suficiente. En consecuencia, esta se subió hasta 10,000 árboles. La profundidad predeterminada de cada árbol (*interaction.depth*) es 5, lo que significa que estamos ensamblando árboles con profundidad máxima de 5. Se especificó el número mínimo de observaciones en el nodo final (*n.minobsinnode*) como 15. La distribución (*distribution*) escogida es "bernoulli" para usar la función de pérdida 0-1. Para la división de los datos (*train.fraction*) se fija el 70 % de los datos para entrenamiento y el restante 30 % para el conjunto de validación.

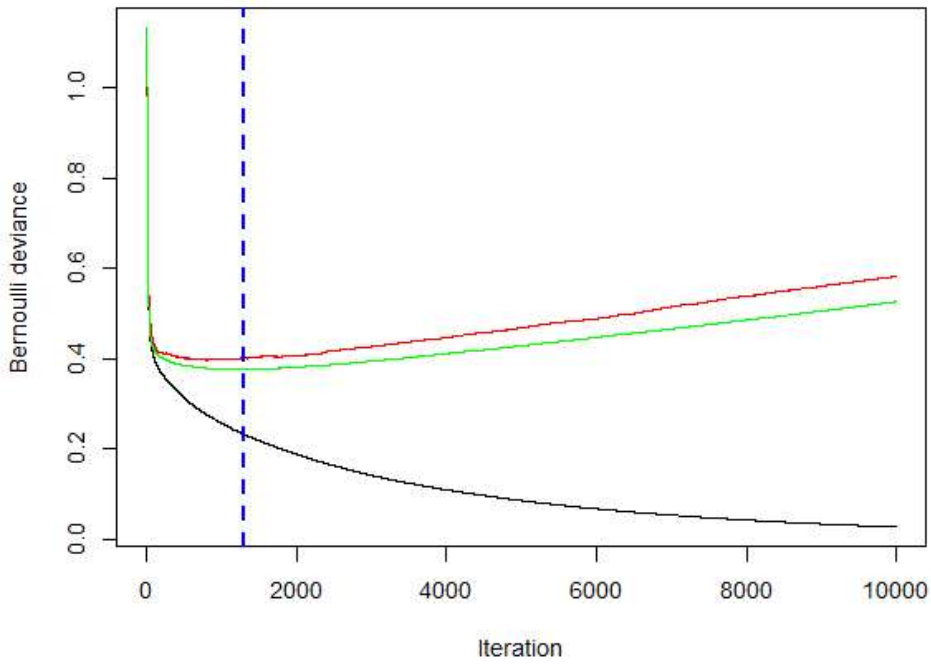
Por último, también se incluye *cv.folds* para realizar una validación cruzada de $K = 5$ partes.

El modelo tardó aproximadamente 20 minutos en ejecutarse usando una computadora con siete núcleos de procesamiento y 24 GB de RAM; los resultados muestran que nuestra función de pérdida 0-1 se minimiza con 10,000 árboles.

3.2.2. Resultados del Algoritmo Gradient Boosting Machine

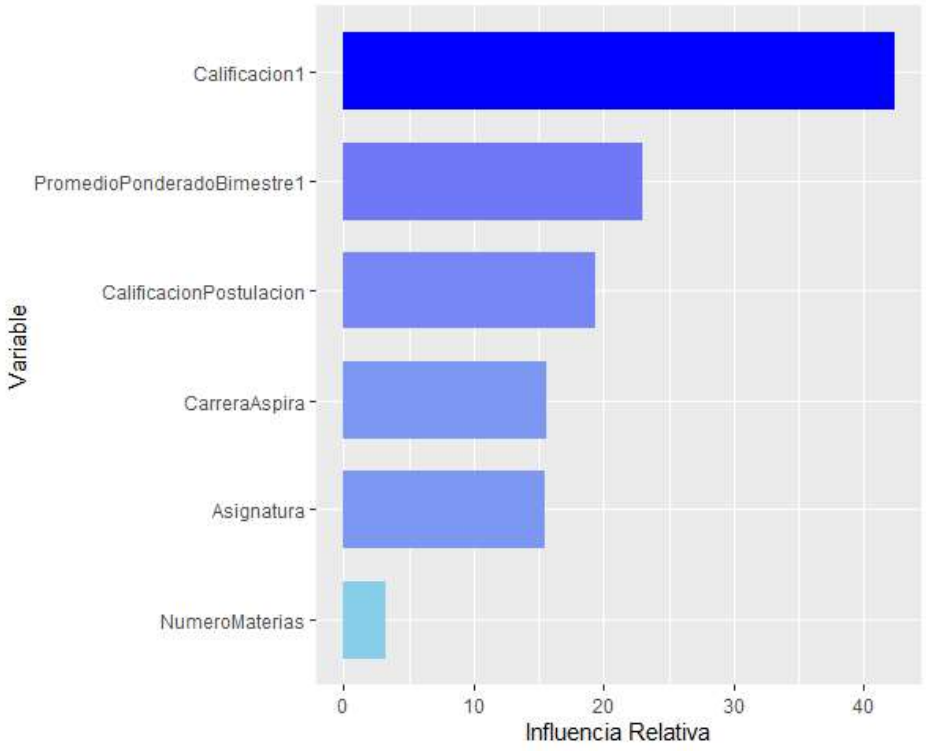
El objeto *gbm* de salida es una lista que contiene varios modelos e información de los resultados. En la siguiente figura, la línea vertical indica el número de árboles que se necesitan para que la función de perdida alcance su punto mínimo.

Figura 3.1: Error de entrenamiento (negro), error de validación (rojo) y error de prueba (verde) con validación cruzada a medida que se agregan n árboles al algoritmo GBM. El número óptimo de árboles es 1288.



Una característica fundamental en el modelado de GBM es la importancia de las variables. En la siguiente figura se muestran las variables en función de su influencia relativa, que es una medida que indica la importancia relativa de cada variable en el entrenamiento del modelo. Las variables con la mayor disminución promedio en el error de clasificación se consideran las más importantes.

Figura 3.2: Las variables con la mayor disminución promedio en el error de clasificación se consideran las más importantes.



Respecto al desempeño del modelo, evaluemos primero los resultados de la matriz de confusión. Comenzando con los estadísticos de la Tabla 3.2 para los datos de entrenamiento y de prueba. El punto óptimo de corte nos sugiere que la tasa de estudiantes que reprueban es mayor a la de los que aprueban. El valor del punto se escogió en función de minimizar el error de clasificación.

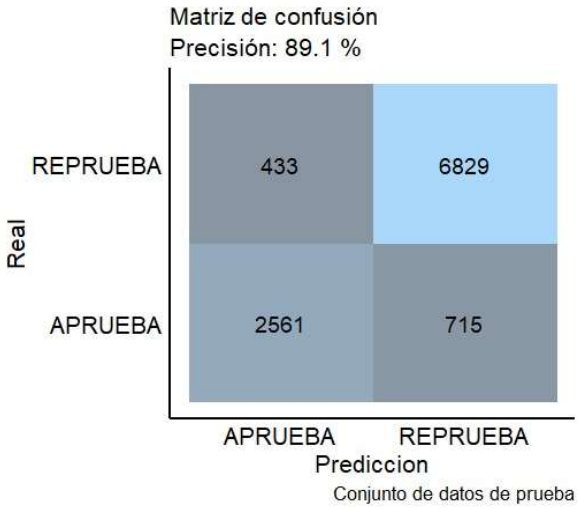
	OptimalCutOff	MissClassificationError	TPR	FPR	Specificity
Entrenamiento	0.69	0.02	0.99	0.04	0.96
Prueba		0.10	0.96	0.09	0.94

Tabla 3.2: Estadísticos de la matriz de confusión

Para el error de clasificación (Err) de la tabla anterior tenemos que este es ligeramente menor en el conjunto de entrenamiento que en el conjunto de prueba; ya que en el conjunto de prueba los datos son no procesados. Sin embargo, el desempeño es bueno y podemos decir que el modelo no esta sobre ajustado al conjunto de datos de entrenamiento. Por tanto, el error de entrenamiento y el error de generalización (E_τ) no son tan distintos.

Los indicadores para los verdaderos positivos (TPR) y los verdaderos negativos (FPR) nos muestran que hay más tendencia a cometer el error de predecir que *Reprueba* cuando en realidad no es así, es por esto que la tasa de los correctamente clasificados como *Reprueba* (Especificidad) es tan cercana a uno.

Figura 3.3: Matriz de confusión



A continuación, tenemos la curva ROC para los datos de prueba, la cual vamos a interpretar. Ya que el área bajo la curva es cercana a uno, podemos decir que el rendimiento del modelo es bueno con respecto a los TPR y los FPR. Así, hemos encontrado un clasificador con un rendimiento muy bueno sin que se sobreajuste.

Figura 3.4: Curva ROC en la muestra de validación

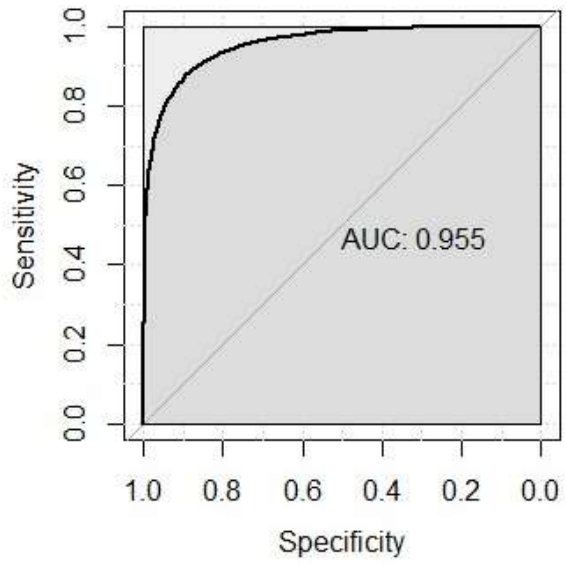


Tabla 3.3: Predicciones para el conjunto de prueba (periodo 2019-A) por carrera de pregrado.

CarreraAspira	Estudiantes	ApruebaNivelacion	REAL	ESTIMADO	PorcentajeError
AGROINDUSTRIA	82	A	16	16	0.0
		F	66	66	
COMPUTACION	165	A	30	30	
		F	135	135	
ECONOMIA	71	A	20	17	4.2
		F	51	54	
ELECTRICIDAD	103	A	31	25	5.8
		F	72	78	
ELECTRONICA Y AUTOMATIZACION	106	A	43	44	0.9
		F	63	62	
FISICA	29	A	15	14	3.4
		F	14	15	
GEOLOGIA	36	A	5	4	2.8
		F	31	32	
INGENIERIA AMBIENTAL	123	A	22	17	4.1
		F	101	106	
INGENIERIA CIVIL	133	A	43	39	3.0
		F	90	94	
INGENIERIA DE LA PRODUCCION	114	A	28	28	0.0
		F	86	86	
INGENIERIA QUIMICA	91	A	38	42	4.4
		F	53	49	
MATEMATICA	36	A	13	15	5.6
		F	23	21	
MATEMATICA APLICADA	75	A	24	28	5.3
		F	51	47	
MECANICA	298	A	92	101	3.0
		F	206	197	
PETROLEOS	54	A	12	8	7.4
		F	42	46	
SOFTWARE	167	A	46	48	1.2
		F	121	119	
TECNOLOGIA SUPERIOR EN AGUA Y SANEAMIENTO AMBIENTAL	103	A	5	12	6.8
		F	98	91	
TECNOLOGIA SUPERIOR EN DESARROLLO DE SOFTWARE	127	A	15	21	4.7
		F	112	106	
TECNOLOGIA SUPERIOR EN ELECTROMECANICA	135	A	19	24	3.7
		F	116	111	
TECNOLOGIA SUPERIOR EN REDES Y TELECOMUNICACIONES	141	A	18	23	3.5
		F	123	118	
TECNOLOGIAS DE LA INFORMACION	100	A	16	18	2.0
		F	84	82	
TELECOMUNICACIONES	98	A	31	35	4.1
		F	67	63	

Tabla 3.4: Predicciones para el conjunto de prueba (periodo 2019-A) por materia del curso de nivelación

CursoNivelacion	Materia	Estudiantes	Reprueba	REAL	ESTIMADO	PorcentajeError
INGENIERIA, CIENCIAS Y CIENCIAS ADMINISTRATIVAS	FISICA	1855	0	608	644	1.9
		1855	1	1247	1212	1.9
	FUNDAMENTOS DE MATEMATICA	1695	0	532	384	8.7
		1695	1	1163	1312	8.8
	FUNDAMENTOS DE QUIMICA	1545	0	478	285	12.5
		1545	1	1067	1260	12.5
	GEOMETRIA Y TRIGONOMETRIA	1775	0	542	467	4.2
		1775	1	1233	1309	4.3
	LENGUAJE Y COMUNICACION	1312	0	649	237	31.4
		1312	1	663	1076	31.5
NIVEL TECNOLOGICO SUPERIOR	FISICA	505	0	77	87	2.0
		505	1	428	419	1.8
	FUNDAMENTOS DE MATEMATICA	485	0	102	114	2.5
		485	1	383	371	2.5
	FUNDAMENTOS DE QUIMICA	393	0	58	56	0.5
		393	1	335	337	0.5
	GEOMETRIA Y TRIGONOMETRIA	501	0	70	109	7.8
		501	1	431	392	7.8
	LENGUAJE Y COMUNICACION	360	0	107	36	19.7
		360	1	253	324	19.7

3.3. Modelo Inferencial

El modelo inferencial busca mejorar el entendimiento de los factores que influyen en el rendimiento académico de los estudiantes del Curso de Nivelación de la Escuela Politécnica Nacional. La regresión logística es la técnica que se ha seleccionado para cumplir con este objetivo.

El ajuste del modelo logit permite crear un perfil de los estudiantes en base a las variables predictivas; tal que todas en una sola ecuación conjunta explican la probabilidad de aprobar o reprobar de cada estudiante. En este modelo los coeficientes pueden ser interpretados (al ser exponenciados y luego restados en uno). El conocimiento de los coeficientes y su ponderación es muy importante para conocer los factores que influyen en la reprobación y con ello poder recomendar acciones que ayuden a reducirla.

Por ello, en la tabla 3.5 se muestra el mejor modelo inferencial de todos los posibles, estimado a partir de una regresión logística y escogido de un pool de estimaciones por tener el mejor criterio de información de Akaike, a través de un algoritmo de selección genético. Este método explora aleatoriamente (pero con un sesgo hacia los mejores modelos) varios subconjuntos de todos los posibles, haciéndolo computacionalmente rápido, eficiente en espacios discretos grandes y logrando converger aún con problemas muy complicados (Calcagno, 2019).

Para la interpretación de este modelo inferencial, partiremos de varios puntos de vista de la literatura para estudiar los efectos de cada variable utilizada sobre la probabilidad de reprobar el curso de nivelación. Al final, el estudio de estas características puede ayudar desde dos enfoques. A nivel micro, sus conclusiones pueden ser utilizadas para mejorar los planes de apoyo y contingencia de las instituciones de educación superior en temas de logística, admisión y cuidado de sus estudiantes; y desde el punto de vista macro, a formular políticas de educación superior que permitan mejorar la calidad de la educación, tener mayores tasas de aprobación y, por ende, mejores resultados macroeconómicos, a la par de cumplimiento de objetivos de desarrollo social.

Así, para entender la problemática desde un espectro más amplio, citaremos a Aina y col. (2018), quienes explican en uno de sus más recientes estudios que la decisión de una persona sobre invertir en la educación terciaria, desde el punto de vista económico, es un proceso secuencial que se va haciendo sobre niveles descendientes de incertidumbre sobre los costos de educación y sus retornos futuros, debido a que los estudiantes actualizan su información disponible y a la vez, su decisión, con cada semestre que pasa. Es decir, un estudiante acabará sus estudios sí y solo sí el valor presente neto de la inversión en su educación (tanto de fuentes pecuniarias como no) es superior a cero. Evidentemente, los costos pecuniarios tienen alta relevancia en esta decisión. Acorde a Larose y col. (1998) y Pascarella, Terenzini y Wolfle (1986), un estudiante tendrá éxito en la universidad si logra integración académica y social, obedeciendo a características de su pasado escolar y contexto social. Al final, todos estos costos y determinantes, podrían en algún momento, causar que el estudiante falle uno u otro curso, y los determinantes de esta situación son los que tomaremos como guía para interpretar nuestros resultados.

Así, primero analicemos las variables sociodemográficas de los estudiantes y su efecto sobre su probabilidad de reprobar.

La edad por su cuenta presenta un signo significativo y positivo, lo cual sugiere que personas mayores en el curso de nivelación tienen una mayor probabilidad de reprobar. En línea con Alexander y Woodruff (1940) y Aina y col. (2018), esto puede deberse a que personas con más edad pueden sentir la obsolescencia de su conocimiento previo, lo que ocasionará que se incremente su dificultad de estudio. Estudiantes más jóvenes en cambio son más conscientes de la actualidad de sus propias habilidades y aptitudes, lo que les permitirá decidir mejor informados al momento de cursar la nivelación. Sin embargo, cabe la posibilidad de que esta variable interactúe con otros factores sociodemográficos.

Tabla 3.5: Resultados del modelo inferencial

Variable	Estimador	Error estándar	Estadístico de Prueba	Pr(>|z|)
(Intercept)	8.3894	0.7189	11.67	0.0000
Sociodemografía				
Edad	0.0239	0.0108	2.22	0.0266
Lugar de residencia: Otras provincias	.	.	.	.
Extranjero	0.5565	0.2050	2.72	0.0066
Quito	-0.1004	0.0448	-2.24	0.0249
Estado civil: Casado	.	.	.	.
Divorciado	-3.2269	0.9546	-3.38	0.0007
Soltero	-1.7953	0.4347	-4.13	0.0000
Unión libre	-1.9019	0.6443	-2.95	0.0032
Bagaje y redes familiares				
Numero Miembros Familia	0.1391	0.0119	11.66	0.0000
Tipo de colegio: Extranjero	.	.	.	.
Fiscal	0.6011	0.2895	2.08	0.0379
Fiscomisional	0.7508	0.2991	2.51	0.0121
Municipal	0.6684	0.3002	2.23	0.0260
Particular	0.7588	0.2911	2.61	0.0092
Características de los estudiantes, habilidades y comportamientos				
Promedio ponderado del primer bimestre	-0.9649	0.0259	-37.25	0.0000
Calificación de postulación	0.0002	0.0000	11.94	0.0000
Calificación del primer bimestre	-0.8633	0.0207	-41.69	0.0000
Número de materias tomadas: Una	.	.	.	.
Dos	0.8765	0.3371	2.60	0.0093
Tres	1.4357	0.3280	4.38	0.0000
Cuatro	2.3101	0.3268	7.07	0.0000
Cinco	2.9183	0.3363	8.68	0.0000
Número de matrícula: Primera	.	.	.	.
Segunda	0.8121	0.0979	8.29	0.0000
Tercera	-0.0533	0.1822	-0.29	0.7697
Segmento poblacional: Acción afirmativa	.	.	.	.
GAR	-0.6059	0.2258	-2.68	0.0073
Mérito Territorial	-0.4715	0.1752	-2.69	0.0071
Población general	-0.2332	0.0914	-2.55	0.0107
Características intrínsecas de la institución				
Curso de nivelación: Ingeniería y Ciencias	.	.	.	.
Nivel Tecnológico Superior	0.1611	0.0526	3.06	0.0022
Semestre del año: A	.	.	.	.
B	-0.3436	0.0426	-8.07	0.0000
Jornada: Matutina	.	.	.	.
Vespertina	-0.1314	0.0413	-3.18	0.0015
Materia: Fundamentos de Física	.	.	.	.
Fundamentos de Matemática	-0.6350	0.0684	-9.28	0.0000
Fundamentos de Química	-1.6994	0.0722	-23.52	0.0000
Geometría y Trigonometría	-0.2912	0.0664	-4.38	0.0000
Lenguaje y Comunicación	-2.8853	0.0941	-30.67	0.0000

Cuando revisamos el lugar de residencia del estudiante, podemos observar que el residir en Quito, reduce la probabilidad de reprobar el curso de nivelación, con respecto a personas de otras ciudades. Este resultado va acorde a la literatura económica, ya que según Aina (2010) en un estudio para universidades italianas, se sugiere que personas que provienen de lugares lejanos al de la ubicación de la universidad tienen mayor probabilidad de fallar en sus cursos debido a las dificultades que estos atraviesan para ajustarse a nuevos ambientes con más responsabilidades.

Respecto al estado civil, nuestros resultados muestran que estudiantes divorciados, solteros y en unión libre presentan una menor probabilidad de reprobar que estudiantes casados. Esto podría deberse a que el estudiante presenta, al estar casado, mayores responsabilidades y por ende, le dedica menos tiempo a sus estudios, factor que será explicado más adelante como comportamiento y motivación del estudiante.

En segundo lugar analicemos las variables relacionadas al bagaje de los padres y las redes familiares.

Empezando por el tipo de colegio donde estudió el individuo previo a ingresar a la universidad, podemos observar que, ceteris paribus, el estudiar en un colegio fiscal, fiscomisional, municipal o particular, respecto a uno extranjero, presenta en promedio mayor probabilidad de reprobar. Este resultado podría venir de varios factores, siendo los más relacionados, aquellos que acorde a Aina y col. (2018), encajan dentro del bagaje educativo y económico de los padres o familia a cargo del estudiante. En efecto, se argumenta que en familias con mayores recursos económicos y cuyos miembros han alcanzado mayores niveles de educación, las nuevas generaciones tienen mayor probabilidad de terminar sus carreras, debido a que estos últimos pueden tener como beneficios, guía y asesoría al momento de la postulación y estudios, además de que con mayores niveles de ingreso, la carga de costos será también más leve. Alexander y Woodruff (1940) por su lado también mencionan que aquellos estudiantes que tengan el apoyo de sus padres en su elección de carrera, tendrán una probabilidad mayor de tener éxito en comparación a aquellos que soolo toman una carrera por complacer a sus familias.

Por otro lado, la variable de número de miembros de la familia muestra que a medida que esta incrementa, lo hace también la probabilidad de reprobar el curso de nivelación, manteniendo todo lo demás constante. Aina y col. (2018) sugiere que familias con un mayor número de miembros genera que sus estudiantes disminuyan su probabilidad de aprobar debido a que los recursos del hogar (tanto pecuniarios como no, e.g. ingreso disponible y atención de los padres) se diluyen con más miembros.

Pasemos ahora a analizar las variables de características de los estudiantes, sus

habilidades y comportamientos.

La variable de número de materias nos otorga información para interpretar que aquellos individuos que toman un mayor número de ellas son más propensos a fallar. Este efecto está correlacionado con el tiempo dedicado al estudio, ya que, al ser un curso propedéutico presencial, por lo general no permite que sus estudiantes asistan a clases y trabajen al mismo tiempo. Entonces es plausible asumir que quienes tienen menos materias le dedican más tiempo a estudiarlas. En efecto, Babcock y Marks (2011) muestran evidencia empírica de universidades estadounidenses que respaldan esta conclusión. Sin embargo, es preciso recalcar que este tiempo dedicado al estudio podría tener interacciones adicionales con variables relacionadas principalmente a la motivación del estudiante, las cuales deberían ser estudiadas más a fondo.

Es así que, al topar la temática de la motivación del estudiante, es prudente analizar el efecto de la variable de número de matrícula ya que se puede observar que aquellos individuos que están cursando su segunda o tercera matrícula tienen menor probabilidad de reprobar que aquellos que se encuentran cursando la primera. Este efecto, en línea con Larose y col. (1998) y Aina y col. (2018) podría estar representando la motivación del estudiante, en especial de aquellos que a pesar de haber reprobado ya una vez el curso de nivelación, lo siguen intentando. Quienes ya lo han repetido dos veces, no presentan diferencias estadísticamente significativas con aquellos que van en su primera matrícula.

Los resultados de la variable segmento poblacional[4], que indican si el estudiante clasifica en algún grupo de los incluidos en su examen de admisión, sugieren que individuos que ingresaron a la universidad por el grupo de alto rendimiento, mérito territorial y población general tienen una menor probabilidad de reprobar que aquellos que ingresaron por acción afirmativa. A este respecto existen dos interpretaciones desde la literatura. La primera corresponde a los grupos de alto rendimiento y mérito territorial, cuya menor probabilidad de falla proviene tanto del argumento de motivación del estudiante (ya que estos estudiantes son recompensados con una beca o cupo por sus buenos resultados) como del de desempeño previo (por haber destacado en sus estudios de segundo nivel). Por otro lado, el hecho de que el grupo de población general presente

[4]**Población general:** Sin aplicación de un criterio adicional.
Acción afirmativa / política de cuotas: Se otorgan de 5 a 15 puntos adicionales en los exámenes de admisión a los aspirantes que encajen dentro de ciertos criterios de condición socioeconómica, ruralidad, territorialidad y vulnerabilidad.
GAR: Aspirantes que accedieron a los mejores puntajes en el examen de admisión en sus respectivas provincias.
Mérito Territorial: Aspirantes que accedieron a los mejores puntajes al momento de graduarse en colegios fiscales, fiscomisionales y municipales (Acuerdo No. Senescyt -2019-30).

una menor probabilidad de reprobar que el grupo de acción afirmativa, hace referencia a la problemática enunciada por Aina (2010), quien da paso a argumentos en contra de la aplicación de estas últimas ya que por lo general estas ocasionan que se apliquen criterios menos severos al seleccionar estudiantes de grupos históricamente discriminados[5]. Adicionalmente, Sandoval y col. (2018) muestran evidencia empírica para la Escuela Politécnica Nacional que soporta la hipótesis de que personas que se acogen a acciones afirmativas y política de cuotas, tienen un menor rendimiento académico que la población general.

Al analizar la calificación del primer bimestre en cada materia, podemos concluir que mientras mayores sean esta calificaciones, menor será la probabilidad del individuo de reprobar. Desde los años noventa hasta la actualidad, autores como Van Overwalle (1989), Larose y col. (1998) y Aina y col. (2018) han sugerido que, independientemente de los resultados obtenidos en el colegio, aquellos individuos con buenos resultados en sus primeros momentos de universidad tienen mayor probabilidad de aprobar debido a que existe un mayor disfrute de sus actividades académicas y perciben menos riesgo sobre el resultado de las mismas, a la vez que ven disminuidos sus costes pecuniarios (e.g. contratación de profesores particulares) para pasar.

Como última en este grupo de variables, la calificación de postulación posee un signo negativo y significativo, de la cual podemos interpretar que estudiantes con una mayor calificación de postulación tendrán una menor probabilidad de reprobar debido a la motivación que tienen para estudiar, y a su vez, a un posible buen pasado académico en el colegio. Larose y col. (1998), Lassibille y Gómez (2011) y Aina y col. (2018) muestran de manera empírica que estudiantes que se gradúan con altas calificaciones del colegio tienen mayor probabilidad de aprobar, siempre y cuando el estudiante no esté en disonancia con su elección.

Finalmente, cabe notar los efectos de algunas variables relacionadas a características intrínsecas de la institución.

La variable materia nos muestra que los estudiantes tienen menor probabilidad de reprobar en todas ellas, con respecto a física. Este fenómeno puede explicarse debido a que generalmente los estudiantes que ingresan al sistema de educación superior se dan cuenta de que el conocimiento básico esperado en el curso propedéutico no es con el que suelen llegar, especialmente en física y geometría.

Respecto a la jornada, se puede observar que quienes estudian en la jornada ves-

[5]Es preciso destacar que este es un tema sensible que requiere de un análisis aislado y detallado, fuera del alcance de este estudio.

pertina tienen menor probabilidad de reprobar que aquellos estudiantes de la jornada matutina. Además, la época del año, representada a través de si el estudiante cursaba el semestre A o B, también nos indica que existe menor probabilidad de reprobar en el último. Esto puede deberse por un lado, a que estudiantes de la jornada vespertina tienen menor probabilidad de llegar tarde a clases por la naturaleza de sus horarios. Así mismo, la mayor tasa de reprobación en el semestre A puede deberse a su correlación con la llegada de estudiantes de otras provincias, cuyo resultado está en línea con la teoría revisada anteriormente.

Capítulo 4

Conclusiones y Recomendaciones

En este trabajo se ha mostrado la utilidad de combinar modelos con objetivos de predicción e inferencia utilizando técnicas de aprendizaje supervisado, para entender a fondo un problema de alta relevancia social.

Por un lado, el uso de Gradient Boosting Machine (GBM) tiene buenos resultados en la predicción de si un estudiante aprobará o no el curso de nivelación, potenciado a través de validación cruzada. Por ello también su preferencia de uso en varias ramas de la ciencia. Este algoritmo predice con una tasa de aciertos del 89 % a aquellos estudiantes que reprobarán el curso de nivelación, logrando un área bajo la ROC en el conjunto de datos de validación de 0.95, la cual nos indica un buen desempeño de la estimación realizada.

Por otro lado, en el modelo de inferencia se muestran algunos puntos. La selección del mejor modelo logit a través del algoritmo de selección genético ha sido útil para determinar qué variables afectan a la probabilidad de reprobar. Factores como la calificación ponderada del primer bimestre, la calificación con la que postuló, su jornada de estudios, su ubicación geográfica de origen, entre otras, afectan de una u otra manera a la probabilidad del estudiante, de aprobar el curso de nivelación.

En definitiva, el uso de estas técnicas estadísticas permite el análisis de políticas relacionadas tanto a la situación del estudiante (e.g. política de cuotas) como al manejo del curso de nivelación (e.g. jornadas), que permitan obtener mejores resultados en la aprobación del mismo. Por ejemplo, es importante dar soporte al problema de pasado académico de los nuevos estudiantes universitarios y generar políticas en la educación secundaria que permitan que los bachilleres lleguen con el menor número de vacíos académicos a su primer año de universidad en pos de que este sea exitoso. Por otro lado, se puede también dar mayor aporte socioeconómico y psicológico a los estudiantes

para que puedan aprobar sus cursos sin problema, sea con apoyo respecto a la elección de su carrera, ayuda económica para él y su familia (si aplicase), tutorías y guías para alivianar la carga de estudios, mejor planeación de las jornadas académicas para evitar agotamientos, entre otras medidas a nivel institucional. A nivel más general, el solucionar el problema de infraestructura para acoger un mayor número de estudiantes y la potencialización de universidades alejadas de las ciudades más grandes son también posibles recomendaciones para evitar que estudiantes de provincia se queden sin cupo.

Así mismo, se recomienda continuar con mayor profundidad el estudio de los efectos de las acciones afirmativas y políticas de cuotas. Esto debido a que Sandoval y col. (2018) y los resultados de este trabajo muestran que los estudiantes acogidos a estas medidas son más propensos a reprobar el Curso de Nivelación y abandonar la universidad; donde algunos de sus principales factores de reprobación asociados son el ingreso, la nota de postulación, la provincia de procedencia, falta de acompañamiento académico, entre otros.

Con ello en mente, y regresando al aspecto pragmático de este trabajo, se recomienda darle seguimiento a los resultados del modelo GBM para analizar posibles cambios debido a nuevos efectos que puedan surgir con el paso del tiempo, sea debido a fenómenos externos o propios al curso de nivelación. Se sugiere, además, sociabilizar el modelo para permitir que otras facultades, e incluso, otras instituciones puedan hacer de esta herramienta y así aportar a la mejora de las condiciones de aprobación en instituciones de educación superior.

En línea con todo este análisis, podemos llegar a un nivel más arriba de la discusión y poner sobre la mesa la coyuntura actual: la inminente intersección entre la ciencia de datos y la política pública. Varios artículos publicados por académicos y profesionales de la industria[1] sugieren que esta es necesaria debido a que el análisis provisto por los algoritmos de la ciencia de datos pueden ayudarnos a entender y quizás resolver problemas complejos, siempre y cuando hayamos entendido su trasfondo histórico, legal y socioeconómico. De esta manera, es oportuno quedarnos con la idea de Sosa Escudero (2018), quien menciona que el aparecimiento de la revolución tecnológica, el big data y el aprendizaje automático implican tanto el aprovechamiento de los datos para una mejor toma de decisiones, así como un cambio de paradigma dentro de las ciencias sociales, haciendo posible la aceleración de los procesos de investigación y desarrollo, enfocándolos en una aplicación práctica (entre otras cosas) a la política pública.

[1] https://towardsdatascience.com/musings-at-the-intersection-of-data-science-and-public-policy-cf0bb2fadc01

Bibliografía

[1] Carmen Aina. "Success and failure of Italian university students . Evidence from administrative data". En: (2010), págs. 1-51.

[2] Carmen Aina y col. "DISCUSSION PAPER SERIES The Economics of University Dropouts and Delayed Graduation : A Survey The Economics of University Dropouts and Delayed Graduation : A Survey". En: 11421 (2018).

[3] H. Akaike. "Information theory and the maximum likelihood principle". En: *2nd International Symposium on Information Theory*. 1973.

[4] Norman Alexander y Ruth Woodruff. "Determinants of College Success". En: *The Journal of Higher Education* 11.9 (1940), págs. 479-485. ISSN: 0022-1546. DOI: 10.1080/00221546.1940.11773738.

[5] David M. Allen. "The Relationship Between Variable Selection and Data Agumentation and a Method for Prediction". En: *Technometrics* (1974). ISSN: 15372723. DOI: 10.1080/00401706.1974.10489157.

[6] Philip Babcock y Mindy Marks. "The falling time cost of college: Evidence from half a century of time use data". En: *Review of Economics and Statistics* (2011). ISSN: 00346535. DOI: 10.1162/REST_a_00093.

[7] Vincent Calcagno. *glmulti: Model Selection and Multimodel Inference Made Easy*. R package version 1.0.7.1. 2019. URL: https://CRAN.R-project.org/package=glmulti.

[8] Carlos Figueroa. *Sistemas de evaluación académica*. El Salvador: Editorial Universitaria, 2014.

[9] Guiselle Garbanzo. *Factores asociados al rendimiento académico en estudiantes universitarios desde el nivel socioeconómico: Un estudio en la Universidad de Costa Rica*. El Salvador: Revista Electrónica Educare Vol. 17, 2013. URL: https://www.redalyc.org/html/1941/194128798005/.

[10] Jordi Gironés Roig y col. *Minería de datos: modelos y algoritmos*. 2017, pág. 274. ISBN: 9788491169048.

[11] Brandon Greenwell y col. *gbm: Generalized Boosted Regression Models*. R package version 2.1.5. 2019. URL: https://CRAN.R-project.org/package=gbm.

[12] Trevor Hastie, Robert Tibshirani y Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. New York: Springer, 2017.

[13] J Hunt. *Classification by induction: Aplications to modelling and control of non linear dynamic systems*. Intelligent Systems Engineering, 1993.

[14] Gareth James y col. *An introduction to Statistical Learning*. 2000. ISBN: 978-1-4614-7137-0. DOI: 10.1007/978-1-4614-7138-7. arXiv: arXiv:1011.1669v3.

[15] I Kononenko, I Bratko y M Kukar. *Machine Learning and Data MIning: Methods and Aplications*. John Wiley & Sons Ltd, 1998.

[16] Simon Larose y col. "Nonintellectual learning factors as determinants for success in college". En: *Research in Higher Education* 39.3 (1998), págs. 275-297. ISSN: 03610365. DOI: 10.1023/A:1018776917403.

[17] Gérard Lassibille y Ma Lucía Navarro Gómez. "How Long Does it Take to Earn a Higher Education Degree in Spain?" En: *Research in Higher Education* 52.1 (2011), págs. 63-80. ISSN: 03610365. DOI: 10.1007/s11162-010-9186-z. URL: http://dx.doi.org/10.1016/j.sbspro.2010.03.544.

[18] Eiliana Montero, Jeannette Villalobos y Astrid Valverde. *Factores Institucionales, Pedagógicos, Psicosociales y Sociodemográficos asociados al rendimiento académico en la Universidad de Costa Rica: Un análisis Multinivel*. Costa Rica: Revista Electrónica de Investigación y Evaluación Educativa, 2007.

[19] Ernest T. Pascarella, Patrick T. Terenzini y Lee M. Wolfle. "Orientation to College and Freshman Year Persistence/Withdrawal Decisions". En: *The Journal of Higher Education* 57.2 (1986), pág. 155. ISSN: 00221546. DOI: 10.2307/1981479.

[20] Elena Paucar. *Cerca de 140 000 cupos ofertan las universidades para el primer semestre del 2018*. Ecuador: El Comercio, 2018. URL: https://www.elcomercio.com/actualidad/cupos-oferta-universidades-estudiantes-educacion.html.

[21] Iván Sandoval y col. "Factores Asociados Al Abandono En Estudiantes De Grupos Vulnerables. Caso Escuela Politécnica Nacional". En: *Congresos CLABES* (2018), págs. 132-141. URL: https://revistas.utp.ac.pa/index.php/clabes/article/view/1907.

[22] Walter Sosa Escudero. "Big data y aprendizaje automático: Ideas y desafíos para economistas". En: *Una nueva econometría*. 2018. ISBN: 978-987-655-201-1.

[23] M. Stone. "Asymptotics for and against cross-validation". En: *Biometrika* (1977). ISSN: 00063444. DOI: 10.1093/biomet/64.1.29.

[24] M. Stone. "Cross-validation and multinomial prediction". En: *Biometrika* (1974). ISSN: 00063444. DOI: 10.1093/biomet/61.3.509.

[25] R Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing, 2019. URL: https://www.R-project.org/.

[26] Jorge Valera y col. *Una explicación del rendimiento estudiantil universitario mediante modelos de regresión logística*. Venezuela: Visión Gerencial, 2009.

[27] Frank Van Overwalle. "Success and failure of freshmen at university: a search for determinants". En: *Higher Education* 18.3 (1989), págs. 287-308. ISSN: 00181560. DOI: 10.1007/BF00138185.

Apéndice A

Descripción del Curso de Nivelación

La Escuela Politécnica Nacional (EPN), es una universidad pública de grado y posgrado ubicada en Quito (Ecuador); reconocida por ser un referente en investigación y educación en ciencias básicas, ingenierías y tecnología. La EPN ofrece un curso Propedéutico o de Nivelación (CN) con diferentes programas académicos, según los grupos de carreras. A estos cursos pueden acceder los aspirantes que aprueben el examen Ser Bachiller y a los cuales la Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT) les asigne un cupo en la EPN.

El objetivo del CN es nivelar los conocimientos de los bachilleres para que puedan continuar con éxito su formación profesional en las diferentes carreras que ofrece la EPN. En la institución se ofrecen dos tipos de CN: uno de Ingeniería, Ciencias y Ciencias Administrativas y otro en Tecnologías, respetando así la diversidad de los perfiles y niveles profesionales. De esta forma los estudiantes que aspiran a la carrera de Física, Matemática e Ingenierías (Agroindustrial, Ambiental, Civil, Matemática, Producción, Eléctrica, Electrónica y Control, Electrónica y Redes de Información, Electrónica y Telecomunicaciones, Geología, Mecánica, Petróleos, Química, Software y Computación) toman el CN para Ingeniería, Ciencias y Ciencias Administrativas; y los estudiantes que aspiran a una Tecnología (Agua y Saneamiento Ambiental, Análisis de Sistemas Informáticos, Electromecánica, Electrónica y Telecomunicaciones) toman el CN Tecnológico Superior. Cada CN tiene sus respectivas materias y número de créditos [1].

Curso de Nivelación de Ingeniería, Ciencias y Ciencias Administrativas

[1] Cada crédito equivale a dos horas normales.

No.	Código	Materia	Créditos
1	CNIC010	FUNDAMENTOS DE MATEMÁTICA	8
2	CNIC020	GEOMETRÍA Y TRIGONOMETRÍA	6
3	CNIC030	FÍSICA	8
4	CNIC040	FUNDAMENTOS DE QUÍMICA	6
5	CNIC050	LENGUAJE Y COMUNICACIÓN	4
		TOTAL	32

Curso de Nivelación Tecnológico Superior

No.	Código	Materia	Créditos
1	CNTS010	FUNDAMENTOS DE MATEMÁTICA	8
2	CNTS020	GEOMETRÍA Y TRIGONOMETRÍA	6
3	CNTS030	FÍSICA	8
4	CNTS040	QUÍMICA	4
5	CNTS050	LENGUAJE Y COMUNICACIÓN	4
		TOTAL	30

Apéndice B

Tasas de Aprobación y Reprobación

Tabla B.1: Indicadores de eficiencia interna del Curso de Nivelación

Periodo	MatriculaInincial	Aprobados	Reprobados	TasaAprobacion	TasaReprobacion
2017-A	985	87	898	8.8 %	91.2 %
2017-B	2127	428	1699	20.1 %	79.9 %
2018-A	2423	490	1933	20.2 %	79.8 %
2018-B	2392	313	2079	13.1 %	86.9 %
2019-A	2410	590	1820	24.5 %	75.5 %
2019-B	2252	2	2250	0.1 %	99.9 %

Note:
Fuente: SAEW, 2019.

Tabla B.2: Indicadores de eficiencia interna por tipo de Curso de Nivelación

Periodo	CursoNivelacion	MatriculaInincial	Aprobados	Reprobados	TasaAprobacion	TasaReprobacion
2017-A	INGENIERIA, CIENCIAS Y CIENCIAS ADMINISTRATIVAS	806	79	727	9.8 %	90.2 %
	NIVEL TECNOLOGICO SUPERIOR	179	8	171	4.5 %	95.5 %
2017-B	INGENIERIA, CIENCIAS Y CIENCIAS ADMINISTRATIVAS	1572	381	1191	24.2 %	75.8 %
	NIVEL TECNOLOGICO SUPERIOR	555	47	508	8.5 %	91.5 %
2018-A	INGENIERIA, CIENCIAS Y CIENCIAS ADMINISTRATIVAS	1847	413	1434	22.4 %	77.6 %
	NIVEL TECNOLOGICO SUPERIOR	576	77	499	13.4 %	86.6 %
2018-B	INGENIERIA, CIENCIAS Y CIENCIAS ADMINISTRATIVAS	1868	300	1568	16.1 %	83.9 %
	NIVEL TECNOLOGICO SUPERIOR	524	13	511	2.5 %	97.5 %
2019-A	INGENIERIA, CIENCIAS Y CIENCIAS ADMINISTRATIVAS	1897	531	1366	28 %	72 %
	NIVEL TECNOLOGICO SUPERIOR	513	59	454	11.5 %	88.5 %
2019-B		477	2	475	0.4 %	99.6 %

Note:
Fuente: SAEW, 2019.

Tabla B.3: Indicadores de eficiencia interna por tipo de Curso de Nivelación y por Asignatura

Periodo	CursoNivelacion	Materia	MatriculaInincial	Aprobados	Reprobados	TasaAprobacion	TasaReprobacion
		FISICA	1186	180	1006	15.2 %	84.8 %
		FUNDAMENTOS DE MATEMATICA	1074	102	972	9.5 %	90.5 %
	INGENIERIA, CIENCIAS Y CIENCIAS ADMINISTRATIVAS	FUNDAMENTOS DE QUIMICA	1028	96	932	9.3 %	90.7 %
		GEOMETRIA Y TRIGONOMETRIA	1152	152	1000	13.2 %	86.8 %
2017-A		LENGUAJE Y COMUNICACION	780	60	720	7.7 %	92.3 %
		FISICA	250	12	238	4.8 %	95.2 %
		FUNDAMENTOS DE MATEMATICA	267	16	251	6 %	94 %
	NIVEL TECNOLOGICO SUPERIOR	GEOMETRIA Y TRIGONOMETRIA	260	15	245	5.8 %	94.2 %
		LENGUAJE Y COMUNICACION	180	3	177	1.7 %	98.3 %
		QUIMICA	165	3	162	1.8 %	98.2 %
		FISICA	1650	402	1248	24.4 %	75.6 %
		FUNDAMENTOS DE MATEMATICA	1482	289	1193	19.5 %	80.5 %
	INGENIERIA, CIENCIAS Y CIENCIAS ADMINISTRATIVAS	FUNDAMENTOS DE QUIMICA	1411	255	1156	18.1 %	81.9 %
		GEOMETRIA Y TRIGONOMETRIA	1703	449	1254	26.4 %	73.6 %
2017-B		LENGUAJE Y COMUNICACION	1073	187	886	17.4 %	82.6 %
		FISICA	556	37	519	6.7 %	93.3 %
		FUNDAMENTOS DE MATEMATICA	544	35	509	6.4 %	93.6 %
	NIVEL TECNOLOGICO SUPERIOR	GEOMETRIA Y TRIGONOMETRIA	576	49	527	8.5 %	91.5 %
		LENGUAJE Y COMUNICACION	428	21	407	4.9 %	95.1 %
		QUIMICA	473	22	451	4.7 %	95.3 %
		FISICA	1740	333	1407	19.1 %	80.9 %
		FUNDAMENTOS DE MATEMATICA	1588	231	1357	14.5 %	85.5 %
	INGENIERIA, CIENCIAS Y CIENCIAS ADMINISTRATIVAS	FUNDAMENTOS DE QUIMICA	1487	163	1324	11 %	89 %
		GEOMETRIA Y TRIGONOMETRIA	1737	328	1409	18.9 %	81.1 %
2018-A		LENGUAJE Y COMUNICACION	1204	90	1114	7.5 %	92.5 %
		FISICA	541	64	477	11.8 %	88.2 %
		FUNDAMENTOS DE MATEMATICA	482	28	454	5.8 %	94.2 %
	NIVEL TECNOLOGICO SUPERIOR	GEOMETRIA Y TRIGONOMETRIA	537	60	477	11.2 %	88.8 %
		LENGUAJE Y COMUNICACION	328	11	317	3.4 %	96.6 %
		QUIMICA	411	14	397	3.4 %	96.6 %
		FISICA	1829	283	1546	15.5 %	84.5 %
		FUNDAMENTOS DE MATEMATICA	1707	190	1517	11.1 %	88.9 %
	INGENIERIA, CIENCIAS Y CIENCIAS ADMINISTRATIVAS	FUNDAMENTOS DE QUIMICA	1596	153	1443	9.6 %	90.4 %
		GEOMETRIA Y TRIGONOMETRIA	1737	212	1525	12.2 %	87.8 %
2018-B		LENGUAJE Y COMUNICACION	1181	110	1071	9.3 %	90.7 %
		FISICA	506	8	498	1.6 %	98.4 %
		FUNDAMENTOS DE MATEMATICA	506	8	498	1.6 %	98.4 %
	NIVEL TECNOLOGICO SUPERIOR	GEOMETRIA Y TRIGONOMETRIA	515	12	503	2.3 %	97.7 %
		LENGUAJE Y COMUNICACION	390	2	388	0.5 %	99.5 %
		QUIMICA	465	2	463	0.4 %	99.6 %
		FISICA	1855	500	1355	27 %	73 %
		FUNDAMENTOS DE MATEMATICA	1695	347	1348	20.5 %	79.5 %
	INGENIERIA, CIENCIAS Y CIENCIAS ADMINISTRATIVAS	FUNDAMENTOS DE QUIMICA	1545	219	1326	14.2 %	85.8 %
		GEOMETRIA Y TRIGONOMETRIA	1775	420	1355	23.7 %	76.3 %
2019-A		LENGUAJE Y COMUNICACION	1312	124	1188	9.5 %	90.5 %
		FISICA	505	56	449	11.1 %	88.9 %
		FUNDAMENTOS DE MATEMATICA	485	41	444	8.5 %	91.5 %
	NIVEL TECNOLOGICO SUPERIOR	GEOMETRIA Y TRIGONOMETRIA	501	52	449	10.4 %	89.6 %
		LENGUAJE Y COMUNICACION	360	10	350	2.8 %	97.2 %
		QUIMICA	393	11	382	2.8 %	97.2 %

Note:
Fuente: SAEW, 2019.

Apéndice C

Algoritmo GBM implementado en R

```
#####################################################################
####                Gradient Boosting Machine
#####################################################################
#### EPN
#### AUTOR: KAREN CALVA
#####################################################################
#### DIRECTORIO DE TRABAJO
#####################################################################

rm(list=ls())
options(java.parameters="-Xmx8000m")
memory.size(max=T)

#####################################################################
#### LIBRERIAS NECESARIAS
#####################################################################

require(gbm)
require(ggplot2)
require(e1071)
require(dplyr)
require(InformationValue)
require(ROCR)
```

```
require(pROC)
require(parallel)
require(sqldf)
require(purrr)

######################################################################
#### CARGANDO BASE ORIGINAL DE TRABAJO
######################################################################

wd <- "Direccion donde esta la base"
setwd(wd)
load("BaseCursoNivelacion.RData" )

PeriodosEntrenamiento = c("2017-A","2017-B","2018-A","2018-B")
PeriodoPrueba= "2019-A"

Train = BaseCursoNivelacion %>% filter(Periodo %in% PeriodosEntrenamiento)

Prueba = BaseCursoNivelacion %>% filter(Periodo == PeriodoPrueba)

######################################################################
#### ENCONTRAR LOS PARÁMETROS ÓPTIMOS
######################################################################

MatrizParametros <- expand.grid(
  shrinkage = c(.01, .1, .3),
  interaction.depth = c(1, 3, 5),
  n.minobsinnode = c(5, 10, 15),
  bag.fraction = c(.65, .8, 1),
  optimal_trees = 0,
  min_RMSE = 0
)

# numero total de combinaciones
nrow(MatrizParametros)
## [1] 81
```

```
# Busqueda exhaustiva
for(i in 1:nrow(MatrizParametros)) {

  # reproducibilidad
  set.seed(123)

  # entrenamiento del modelo
  gbm.tune <- gbm(
    formula = Reprueba ~ .,
    distribution = "bernoulli",
    data = Train,
    n.trees = 10000,
    interaction.depth = MatrizParametros$interaction.depth[i],
    shrinkage = MatrizParametros$shrinkage[i],
    n.minobsinnode = MatrizParametros$n.minobsinnode[i],
    bag.fraction = MatrizParametros$bag.fraction[i],
    train.fraction = .7,
    n.cores = NULL, # usará todos los cores disponibles
    verbose = TRUE,
    keep.data = FALSE
  )

  MatrizParametros$optimal_trees[i] <- which.min(gbm.tune$valid.error)
  MatrizParametros$min_RMSE[i] <- sqrt(min(gbm.tune$valid.error))
}

MatrizParametros %>%
  dplyr::arrange(min_RMSE) %>%
  head(1)

#####################################################################
#### ENTRENAMIENTO DEL MODELO GBM
#####################################################################

set.seed(123)
```

```
# train GBM model
gbm.fit.final <- gbm(
  formula = Reprueba ~ .,
  distribution = "bernoulli",
  data = Train,
  n.trees = 10000,
  interaction.depth = 5,
  shrinkage = 0.01,
  n.minobsinnode = 15,
  bag.fraction = .85,
  train.fraction = 1,
  n.cores = NULL,
  verbose = FALSE,
  keep.data = FALSE,
  cv.folds=5
  )

print(gbm.fit)
```

Apéndice D

Predicciones para los datos de prueba

Tabla D.1: Predicción por carrera para el periodo 2019-A

CarreraAspira	Estudiantes	ApruebaNivelacion	REAL	ESTIMADO	PorcentajeError
AGROINDUSTRIA	82	A	16	11	6.1
		F	66	71	
COMPUTACION	165	A	30	21	5.5
		F	135	144	
ECONOMIA	71	A	20	11	12.7
		F	51	60	
ELECTRICIDAD	103	A	31	21	9.7
		F	72	82	
ELECTRONICA Y AUTOMATIZACION	106	A	43	44	0.9
		F	63	62	
FISICA	29	A	15	12	10.3
		F	14	17	
GEOLOGIA	36	A	5	3	5.6
		F	31	33	
INGENIERIA AMBIENTAL	123	A	22	17	4.1
		F	101	106	
INGENIERIA CIVIL	133	A	43	33	7.5
		F	90	100	
INGENIERIA DE LA PRODUCCION	114	A	28	25	2.6
		F	86	89	
INGENIERIA QUIMICA	91	A	38	39	1.1
		F	53	52	
MATEMATICA	36	A	13	9	11.1
		F	23	27	
MATEMATICA APLICADA	75	A	24	15	12.0
		F	51	60	
MECANICA	298	A	92	81	3.7
		F	206	217	
PETROLEOS	54	A	12	8	7.4
		F	42	46	
SOFTWARE	167	A	46	51	3.0
		F	121	116	
TECNOLOGIA SUPERIOR EN AGUA Y SANEAMIENTO AMBIENTAL	103	A	5	10	4.9
		F	98	93	
TECNOLOGIA SUPERIOR EN DESARROLLO DE SOFTWARE	127	A	15	13	1.6
		F	112	114	
TECNOLOGIA SUPERIOR EN ELECTROMECANICA	135	A	19	21	1.5
		F	116	114	
TECNOLOGIA SUPERIOR EN REDES Y TELECOMUNICACIONES	141	A	18	16	1.4
		F	123	125	
TECNOLOGIAS DE LA INFORMACION	100	A	16	7	9.0
		F	84	93	
TELECOMUNICACIONES	98	A	31	28	3.1
		F	67	70	

Tabla D.2: Predicción por materia para el periodo 2019-A

CursoNivelacion	Materia	Estudiantes	Reprueba	REAL	ESTIMADO	PorcentajeError
INGENIERIA, CIENCIAS Y CIENCIAS ADMINISTRATIVAS	FISICA	1855	A	608	442	8.9
			F	1247	1414	9.0
	FUNDAMENTOS DE MATEMATICA	1695	A	532	411	7.1
			F	1163	1286	7.3
	FUNDAMENTOS DE QUIMICA	1545	A	478	481	0.2
			F	1067	1065	0.1
	GEOMETRIA Y TRIGONOMETRIA	1775	A	542	555	0.7
			F	1233	1223	0.6
	LENGUAJE Y COMUNICACION	1312	A	649	671	1.7
			F	663	642	1.6
NIVEL TECNOLOGICO SUPERIOR	FISICA	505	A	77	80	0.6
			F	428	425	
	FUNDAMENTOS DE MATEMATICA	485	A	102	96	1.2
			F	383	390	1.4
	GEOMETRIA Y TRIGONOMETRIA	501	A	70	76	1.2
			F	431	425	
	LENGUAJE Y COMUNICACION	360	A	107	77	8.3
			F	253	283	
	QUIMICA	393	A	58	61	1.5
			F	335	329	