

Desarrollando Modelos de Análisis Predictivo Para Mejorar la Probabilidad de Graduación de Estudiantes Subgraduados

Jesús D. Hernández Londoño

Jurado

Wolfgang A. Rolke, Ph.D

Edgardo Lorenzo González, Ph.D

Carmen P. Pares Pares, MBA

Roberto Rivera Santiago, Ph.D - Consejero

Universidad de Puerto Rico Recinto Mayagüez
Departamento de Ciencias Matemáticas



Contenido

- 1 Introducción
- 2 Objetivo
- 3 Metodología
- 4 Resultados
- 5 Conclusiones
- 6 Referencias



Contenido

- 1 Introducción
- 2 Objetivo
- 3 Metodología
- 4 Resultados
- 5 Conclusiones
- 6 Referencias



Introducción

¿Qué beneficios brinda una población educada?

Una sociedad educada brinda a la nación seres humanos

- Capaces de luchar (comunicativos), generar cambio, crear y decidir.
- Capaces de respetar las diferencias.
- Un medio efectivo para poder acceder y ascender a los diferentes puestos de trabajo.
- La mejor garantía del progreso de una Nación.



Introducción

¿Cómo está la educación universitaria en Puerto Rico?

- En promedio solo el 45 % de estudiantes subgraduados obtienen su grado al 150 % de la duración de su programa académico.
- La Universidad de Puerto Rico Recinto de Río Piedras de 58 %, la Pontificia Universidad Católica de Puerto Rico de 35.3 %, la Universidad Interamericana de Puerto Rico de 33 %. La tasa de graduación es un poco mejor para el RUM, pero aún ronda por 50 %, muy por debajo del 59 % en Estados Unidos.



Introducción

Introducción

- Algunas universidades ofrecen talleres y consejería a estudiantes para facilitar que se gradúen.
- No existe una manera objetiva de predecir si una estudiante en particular completará su grado.
- Predecir si estudiantes particulares se graduarán permite desarrollar estrategias individualizadas para ayudar a estudiantes con pocas probabilidades de graduación.



Contenido

- 1 Introducción
- 2 **Objetivo**
- 3 Metodología
- 4 Resultados
- 5 Conclusiones
- 6 Referencias



Objetivo

Objetivo

Predecir si un estudiante subgraduado se gradúa al 150 % de la duración de su programa académico de la Universidad de Puerto Rico Recinto Mayagüez, usando métodos de aprendizaje automático según varias variables del estudiante.



Contenido

- 1 Introducción
- 2 Objetivo
- 3 Metodología**
- 4 Resultados
- 5 Conclusiones
- 6 Referencias



Metodología

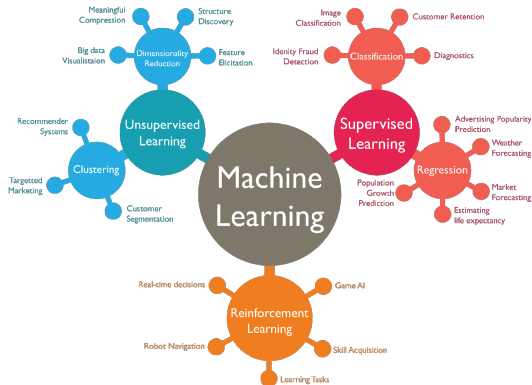


Figura 1: Tipos de aprendizaje (Extraída de Towards Data Science [1])



Metodología

- Se imputan datos usando la imputación multivariante por ecuaciones encadenadas (MICE).
- Exploración de los datos.
- Dos escenarios
 - **Escenario 1:** Predicción de graduación al 150 % de la duración de su programa académico *antes* de entrar a la universidad.
 - **Escenario 2:** Predicción de graduación al 150 % de la duración de su programa académico *luego* del primer año de universidad en el RUM.



Metodología

- Comparar Métodos para validar predicciones.
 - Árboles de Clasificación.
 - Bosque Aleatorio.
 - Gradient Boosting.
 - Naïve Bayes.
 - Regresión Logística.
 - TabNet. No se han confirmado sus componentes teóricos. En consecuencia no se ha publicado en algún jornal.



MICE

MICE

- La imputación multivariante por ecuaciones encadenadas (MICE) es un método sofisticado que asume los valores faltantes de tipo MAR y predice el valor faltante usando un modelo donde la predictora a la que pertenece dicho valor faltante es la variable dependiente y las demás predictoras son independientes.
- Por ejemplo, las variables binarias se pueden modelar mediante regresión logística y las variables continuas mediante regresión lineal [1].

Nota: Este proyecto asume datos faltantes de tipo MAR.



MICE

Algoritmo. MICE

- ➊ Haga una imputación básica (media, moda u otra imputación) para cada valor faltante.
 - ➋ Tome una variable que contiene valores faltantes y vuelva a poner como NA sus valores faltantes originales. Será la variable dependiente a modelar.
 - ➌ Ajuste un modelo en el que la variable del paso 2 es la variable dependiente y las restantes las independientes.
 - ➍ Haga predicciones con el modelo del paso 3 y reemplace los valores faltantes con esas predicciones.
 - ➎ Repita del paso 2 al 4 hasta tomar todas las variables que tenían valores faltantes.
-

Cuadro 1: Algoritmo de MICE.(Adaptado de [1])



Árboles de Clasificación

- CART es una metodología no paramétrica (esto significa que no requiere establecer una estructura funcional paramétrica antes de usar los datos)
- Los árboles son gráficos con estructura de árbol en el que se ilustran reglas de decisión.
- La estructura del árbol consta de un solo nodo raíz (ubicado en la parte superior; es el primer nodo), nodos internos y nodos hoja (ubicados en la base del árbol), enlazados con ramas que contiene reglas de decisión desde la parte superior hasta la base del árbol.



Árboles de Clasificación

El procedimiento para desarrollar un árbol de clasificación usando CART consisten en los siguientes pasos

- Construcción del árbol.
- Poda del árbol y elección del árbol óptimo.



Árboles de Clasificación

Algoritmo. Construyendo un Árbol de Clasificación

- ➊ Divida los datos de las predictoras en un subconjunto de hipercubos.
- ➋ Calcule la medida de impureza, $G(m) = 2\hat{p}(1 - \hat{p})$ (Índice Gini)
Donde \hat{p} es la proporción de instancias que pertenecen a una de las clases.
- ➌ Calcule la expresión $I(m) = i(m) - P[m_L]i(m_L) - P[m_R]i(m_R)$.
Donde $i(m)$ es la medida de impureza del m-ésimo nodo "madre", $P[m_L]$ y $P[m_R]$ es la proporción de observaciones del nodo madre que quedan dentro del nodo izquierdo m_L y derecho m_R , respectivamente. $i(m_L)$ y $i(m_R)$ Es la medida de impureza del nodo hijo izquierdo y derecho, respectivamente.
- ➍ Elige la predictora y punto de división que maximice el paso anterior para asignarla como un nodo.
- ➎ Repite el paso del 1 al 4 en las particiones resultantes.

Cuadro 2: Construcción de un Árbol de Clasificación. (Adaptado de [1])



Árboles de Clasificación

Poda y elección del árbol óptimo

Cuando el árbol se poda, resultan una serie de diferentes árboles de tamaños decrecientes llamados subárboles, la pregunta que precede a esto es ¿cómo decido que subárbol usar? Para responder a esa pregunta la metodología CART usa una función costo-complejidad.

- 1 Aplique la función costo-complejidad $R_\alpha(T) = Resub(T) + \alpha|T|$ con distintos valores de α por cada subárbol.

Donde $Resub(T) = 1 - \max_k(\hat{p}_{mk})$ es la tasa de error de clasificación del árbol T en el m-ésimo hipercubo que pertenecen a la K-ésima clase (en el caso de un árbol de regresión se usa la suma de los cuadrados del error del árbol T), $|T|$ número de nodos del árbol T y el parámetro de complejidad α es un peso mayor o igual a cero que compensa el número de nodos de un árbol.



Árboles de Clasificación

Poda y elección del árbol óptimo

- 1 Compare los resultados y elija los subárboles que minimicen $R_\alpha(T)$.
- 2 El árbol que obtenga en promedio un error de clasificación más bajo por validación cruzada en 2 será el árbol óptimo.



Árboles de Clasificación

Hiperparámetros

Los hiperparámetros principales usados para entrenar árboles de clasificación se describen en el cuadro 3. Se eligen los que consigan el mejor árbol óptimo de todas las combinaciones de hiperparámetros.

Hiperparámetros	Descripción
Parámetro de complejidad (CP)	Ayuda a la comparación entre diferentes tamaños de árboles
Min split	Número mínimo de observaciones que deben existir en un nodo para que se intente una división
Max depth	Número máximo de nodos entre un nodo hoja y el nodo raíz

Cuadro 3: Hiperparámetros Árboles de Clasificación. (Extraído de Rstudio)



Bosque Aleatorio

El nombre Bosque Aleatorio se dio en virtud de dos componentes que posee, muchos árboles ajustados en diferentes muestras Bootstrap (Bosque) y el elemento de aleatorización en el proceso de construcción de los árboles (Aleatorio).

Algoritmo. *Construyendo un Bosque Aleatorio*

- ❶ Elija 1 muestra bootstrap de tamaño N de los datos de entrenamiento.
 - ❷ Construya un árbol de clasificación en la muestra del paso 1 y en cada nodo seleccione una predictora de las $m = \sqrt{p}$ predictoras elegidas aleatoriamente.
 - ❸ Repita el paso 1 al 2.
-

Cuadro 4: Construcción de un Bosque Aleatorio. (Adaptado de [1])



Bosque Aleatorio

OOB (Out of Bag)

En promedio, cada muestra bootstrap selecciona alrededor de dos tercios de las observaciones para ajustar cada árbol del bosque aleatorio. La tercera parte restante que no se usa, se denomina observaciones fuera de la bolsa (de sus siglas en ingles OOB) [1]. Las OOB se usan para calcular el error de clasificación.

$$Error_{OOB} = \frac{1}{N} \sum_{i=1}^N I(Y_i \neq \hat{f}_{oob}(x))$$

El error de clasificación representa la proporción de observaciones que fueron incorrectamente clasificadas por el bosque aleatorio.



Bosque Aleatorio

Hiperparámetros

Los hiperparámetros principales usados para entrenar bosques aleatorios se describen en el cuadro 5. Se eligen los que consigan minimizar el error OBB de todas las combinaciones de hiperparámetros.

Hiperparámetros	Descripción
mtry	Número de variables seleccionadas aleatoriamente en cada división
Num trees	Número de árboles del bosque
Min node size	Número de observaciones mínimo en los nodos terminales de los árboles

Cuadro 5: Hiperparámetros Bosque Aleatorio. (Extraído de Rstudio)



Gradient Boosting

Gradient Boosting ajusta secuencialmente modelos llamados weak learners (que puede ser un árbol), a los residuales del modelo ajustado previamente, para que aprenda de los errores del modelo anterior y de esta manera impulse (como se refleja en el término "boosting") el rendimiento predictivo [1].

Algoritmo. *Construyendo Gradient Boosting*

- ➊ Dar un valor inicial como modelo constante.
 - ➋ Extraer una muestra de tamaño $n \leq N$ al azar de los datos de entrenamiento.
 - ➌ Calcule los residuales.
 - ➍ Ajuste un árbol de regresión al paso anterior.
 - ➎ Hacer predicciones con el modelo del paso anterior.
 - ➏ Repita el paso del 2 al 5.
-

Cuadro 6: Construcción de Gradient Boosting. (Adaptado de [2])



Gradient Boosting

Hiperparámetros

Los hiperparámetros principales usados para entrenar gradient boosting se describen en el cuadro 7. Se eligen los que consigan minimizar los residuales de todas las combinaciones de hiperparámetros.

Hiperparámetros	Descripción
N trees	Número de arboles para ajustar
Interaction depth	Profundidad máxima de cada árbol
Shrinkage	Tasa de aprendizaje
N min obs in node	Número mínimo de observaciones en los nodos terminales de los árboles
Bag fraction	Fracción de observaciones seleccionadas al azar para proponer el siguiente árbol

Cuadro 7: Hiperparámetros Gradient Boosting. (Extraído de Rstudio)



Naïve Bayes

Este método se conoce como bayes ingenuo porque asume predictores independientes. Es simple y sencillo de aplicar. Dado un vector de entradas ($X_1 = u_1, X_2 = u_2, \dots, X_m = u_m$) el objetivo es elegir la clase c_i que maximice la probabilidad posterior de esa clase dadas las predictoras (1).

$$P(Y = c_i | X_1, X_2, \dots, X_m) = \frac{P(X_1, X_2, \dots, X_m | Y = c_i)P(Y = c_i)}{P(X_1, X_2, \dots, X_m)} \quad (1)$$

Corrección de Laplace: Es una técnica que resuelve el problema de probabilidad cero ($P(X_j = u_j | Y = c) = 0$), sumando a la probabilidad un valor de α que representa el parámetro de suavizado.



Naïve Bayes

Hiperparámetros

Los hiperparámetros principales usados para entrenar distintos Naïve Bayes se describen en el cuadro 8. Se eligen los que consigan maximizar la probabilidad posterior (expresión (1)) de todas las combinaciones de hiperparámetros.

Hiperparámetros	Descripción
Use kernel	Estimación de densidad
fL	Suavizado de Laplace
adjust	Ajustar el ancho de banda de la densidad del kernel (números más grandes significan una estimación de densidad más flexible)

Cuadro 8: Hiperparámetros Naïve Bayes. (Extraído de Rstudio)



Regresión Logística

La regresión logística es un modelo de regresión lineal que utiliza la función logística (2) para modelar la variable respuesta Y . Haciendo que la salida se interprete como una probabilidad, la probabilidad de que la variable y_i sea igual a la clase o categoría uno dada una combinación lineal de las predictoras (expresión (2)).

$$P(y_i = 1|\mathbf{X}) = \frac{e^{\beta^T \mathbf{X}}}{e^{\beta^T \mathbf{X}} + 1} \quad (2)$$

La combinación lineal de las predictoras se relacionado con la variable respuesta Y a través de una función de enlace, la función log-odds o función logit (expresión (3)).

$$\log \left(\frac{P(y_i = 1|\mathbf{X})}{1 - P(y_i = 1|\mathbf{X})} \right) = \beta^T \mathbf{X} \quad (3)$$



Regresión Logística

Parámetros

Como cada predictora se escala linealmente, cada predictora tendrá un coeficiente o parámetro β que debe ser estimado. La estimación de los β se hace mediante máxima verosimilitud.

$$\begin{aligned}
 l(\beta; \mathbf{Y}) = \log L(\beta; \mathbf{Y}) &= \sum_{i=1}^N \log f(y_i; p) \\
 &= \sum_{i=1}^N \{y_i \log(p(\mathbf{Y}; \beta)) + (1 - y_i) \log(1 - p(\mathbf{Y}; \beta))\}
 \end{aligned} \tag{4}$$

Entonces la estimación de los β 's se realiza maximizando (4).



TabNet

Es una arquitectura de red neuronal para datos tabulares. Su construcción permite un aprendizaje más eficiente a diferencia de la arquitectura de redes neuronales, ya que elige las predictoras más relevantes a través de la atención secuencial (transformador atento), impidiendo que en el aprendizaje se escojan variables irrelevantes.

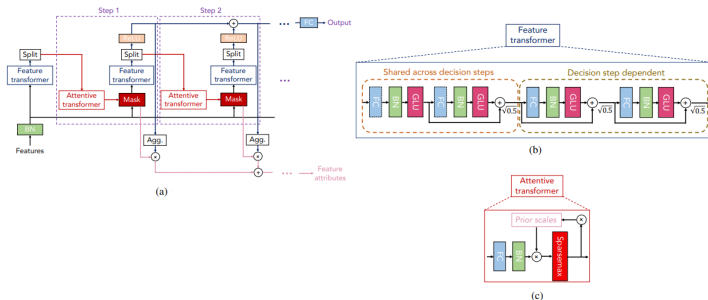


Figura 2: (a) Arquitectura TabNet compuesta por un (b) transformador de predictoras y (c) un transformador atento. (Extraída de [2])



TabNet

Hiperparámetros

Los hiperparámetros principales usados para entrenar TabNet se describen en el cuadro 9. Se eligen los que consigan maximizar la sensibilidad de todas las combinaciones de hiperparámetros.

Hiperparámetros	Descripción
Max epochs	Número máximo de épocas para el entrenamiento
Batch size	Tamaño del lote a normalizar. Se recomiendan lotes grandes
Virtual batch size	Tamaño del lote a normalizar en el Feature Transformer y el Attentive Transformer
Patience	Número de épocas consecutivas sin mejora antes de realizar una parada anticipada

Cuadro 9: Hiperparámetros TabNet. (Extraído de [2])



Métricas de Evaluación

	Actual Positive	Actual Negative
Predicted Positive	True Positive (TP)	False Positive (FP) Type II Error
Predicted Negative	False Negative (FN) Type I Error	True Negative (TN)

Cuadro 10: Matriz de Confusión para métodos de aprendizaje automatizado.

- Sensibilidad = Recall = $\frac{TP}{TP+FN}$.
- $P1 = \frac{FN}{FN+TP} = 1 - \text{Recall}$.
- $P2 = \frac{FN}{FN+TP+FP+TN}$.
- AUC es la proporción de casos positivos y negativos predichos correctamente por el modelo.



Contenido

- 1 Introducción
- 2 Objetivo
- 3 Metodología
- 4 Resultados**
- 5 Conclusiones
- 6 Referencias



Resultados

Datos

- Cohorte hasta el año de comienzo 2010.
- Observaciones 24,432. Predictoras 17.
- De los 24,432 resultan 19,546 para entrenamiento (80 %) y 4,886 (20 %) de prueba.

Predictoras

Año, Facultad, Programa de admisión, Apt Verbal, Aprov Matemáticas, Apt Matemáticas, Aprov Español, Aprov Inglés, Ingreso Familiar, Educación Padre, Educación Madre, Genero, Tipo de Escuela, GPA Primer año, Rel Estudiante GPA, Rel Escuela GPA y Graduación.



Resultados

Rel Estudiante GPA

- Se obtiene al dividir el GPA de escuela superior del estudiante al solicitar admisión, entre el promedio del GPA de todos los estudiantes admitidos a la universidad que provienen de la misma escuela superior.
- Explica qué tan preparados están los estudiantes para la universidad en relación a otros estudiantes admitidos.

Rel Escuela GPA

- Se obtiene al dividir el promedio del GPA de primer año de universidad de los estudiantes de una misma escuela superior entre el promedio del GPA de la escuela de esos mismos estudiantes.
- Pretende medir la calidad de una escuela superior en relación a otras escuelas superiores.



Análisis Exploratorio

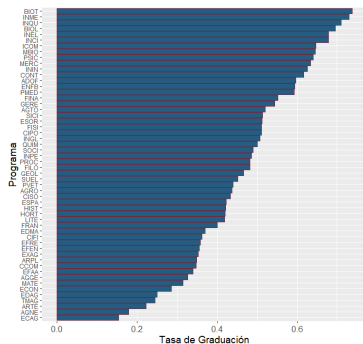


Figura 3: Tasa de Graduación exitosa por programa.

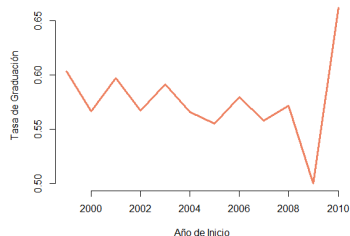


Figura 4: Tasa de Graduación por Año de Ingreso.



Análisis Exploratorio

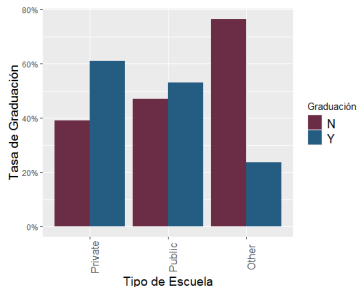


Figura 5: Tasa de Graduación por Tipo de Escuela.

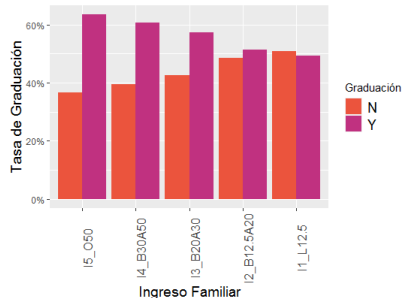


Figura 6: Tasa de Graduación por Ingreso Familiar.



Análisis Exploratorio

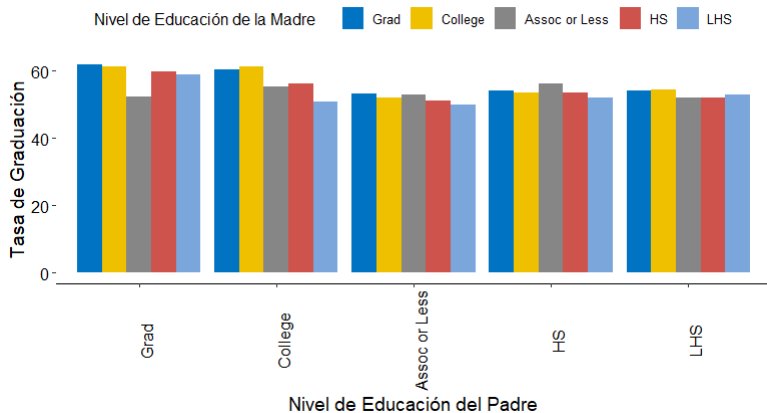


Figura 7: Efecto de la Educación del Padre con la Educación de la Madre - None a NA



Análisis

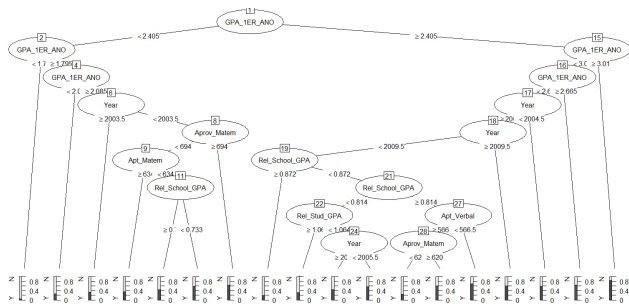
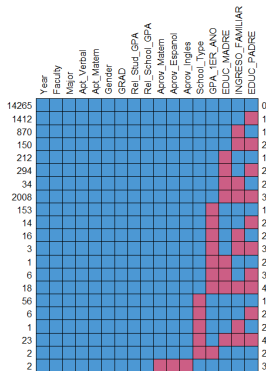


Figura 9: Ejemplo de Árbol decisional.

Figura 8: Patrones
Valores Faltantes



Rendimiento Predictivo

Métrica	Árbol de Clasificación	Bosque Aleatorio	Gradient Boosting	Naive Bayes	Regresión Logística	TabNet
Sensibilidad	0.5905	0.6294	0.7842	0.5757	0.5952	0.6331
AUC	0.6705	0.7261	0.6868	0.6792	0.7248	0.7370
P(Modelo predice si se gradúa Realmente no se gradúa)	0.4095	0.3706	0.2158	0.4242	0.4048	0.1019
P(Modelo predice incorrectamente que el estudiante si se gradúa)	0.1858	0.1681	0.0979	0.1925	0.1836	0.1687

Cuadro 11: Modelos para escenario 1: Predicción de graduación al 150 % de la duración de su programa académico antes de entrar a la universidad - datos no disponibles son removidos.

Métrica	Árbol de Clasificación	Bosque Aleatorio	Gradient Boosting	Naive Bayes	Regresión Logística	TabNet
Sensibilidad	0.6682	0.7024	0.8090	0.6609	0.7259	0.7338
AUC	0.8128	0.8330	0.8436	0.7801	0.8442	0.8266
P(Modelo predice si se gradúa Realmente no se gradúa)	0.3317	0.2975	0.1910	0.3391	0.2741	0.2661
P(Modelo predice incorrectamente que el estudiante si se gradúa)	0.1505	0.1350	0.0866	0.1538	0.1243	0.1237

Cuadro 12: Modelos para escenario 2: Predicción de graduación al 150 % de la duración de su programa académico luego del primer año de universidad en el RUM - datos no disponibles son removidos.



Rendimiento Predictivo

Métrica	Árbol de Clasificación	Bosque Aleatorio	Gradient Boosting	Naive Bayes	Regresión Logística	TabNet
Sensibilidad	0.5657	0.6414	0.7949	0.5664	0.5972	0.7023
AUC	0.6554	0.7058	0.7303	0.6797	0.7251	0.7370
P(Modelo predice si se gradúa Realmente no se gradúa)	0.4343	0.3585	0.2050	0.4336	0.4028	0.2976
P(Modelo predice incorrectamente que el estudiante si se gradúa)	0.1970	0.1627	0.0930	0.1967	0.1827	0.1443

Cuadro 13: Modelos para escenario 1: Predicción de graduación al 150 % de la duración de su programa académico antes de entrar a la universidad - datos no disponibles son imputados.

Métrica	Árbol de Clasificación	Bosque Aleatorio	Gradient Boosting	Naive Bayes	Regresión Logística	TabNet
Sensibilidad	0.6917	0.7078	0.8224	0.6495	0.7299	0.7744
AUC	0.7713	0.8457	0.8307	0.8449	0.8312	0.8352
P(Modelo predice si se gradúa Realmente no se gradúa)	0.3083	0.2922	0.1776	0.3505	0.2701	0.2255
P(Modelo predice incorrectamente que el estudiante si se gradúa)	0.1399	0.1326	0.0805	0.1590	0.1225	0.1072

Cuadro 14: Modelos para escenario 2: Predicción de graduación al 150 % de la duración de su programa académico luego del primer año de universidad en el RUM - datos no disponibles son imputados.



Gradient Boosting - Modelo para el escenario 1

Hiperparámetros

Finalmente Gradient Boosting elige N trees (número de árboles)= 2147, Interaction depth (profundidad máxima de cada árbol)= 3, Shrinkage (tasa de aprendizaje)= 0.01, N min obs in node (mínimo de observaciones en los nodos terminales)= 10 y Bag fraction (fracción de observaciones)= 1.

	Actual N	Actual Y
Predicted N	1186	855
Predicted Y	306	941

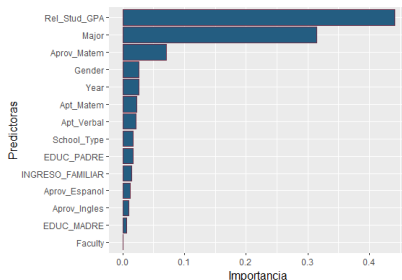
Cuadro 15: Matriz de confusión Gradient Boosting.



Gradient Boosting

Métricas de Evaluación

- Sensibilidad = $\frac{1186}{1186+306} = 0.7949$
- $P(\text{Modelo predice si se gradúa} \mid \text{Realmente no se gradúa}) = \frac{306}{1186+306} = 0.2050$
- $P(\text{Modelo predice incorrectamente que el estudiante si se gradúa}) = \frac{306}{1186+306+855+941} = 0.0930$



Gradient Boosting - Modelo para el escenario 2

Hiperparámetros

Finalmente Gradient Boosting elige N trees (número de árboles)= 2470, Interaction depth (profundidad máxima de cada árbol)= 3, Shrinkage (tasa de aprendizaje)= 0.01, N min obs in node (mínimo de observaciones en los nodos terminales)= 5 y Bag fraction (fracción de observaciones)= 0.65.

	Actual N	Actual Y
Predicted N	1227	573
Predicted Y	265	1223

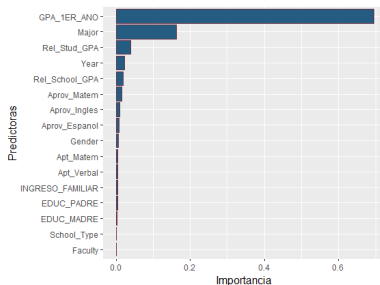
Cuadro 16: Matriz de confusión para Gradient Boosting.



Gradient Boosting

Métricas de Evaluación

- Sensibilidad = $\frac{1227}{1227+265} = 0.8224$
- $P(\text{Modelo predice si se gradúa} \mid \text{Realmente no se gradúa}) = \frac{265}{1227+265} = 0.1776$
- $P(\text{Modelo predice incorrectamente que el estudiante si se gradúa}) = \frac{265}{1227+265+573+1223} = 0.0805$



Contenido

- 1 Introducción
- 2 Objetivo
- 3 Metodología
- 4 Resultados
- 5 Conclusiones**
- 6 Referencias



Conclusiones

Conclusiones

- La imputación de los datos mejoró las predicciones solo para TabNet en ambos escenarios.
- Gradient Boosting (para el escenario 1) identifica correctamente el 79.49 % de los estudiantes que no se graduaron y la probabilidad de predecir incorrectamente que el estudiante si se gradúa es del 9.3 %.
- Gradient Boosting (para el escenario 2) identifica correctamente el 82.24 % de los estudiantes que no se graduaron y la probabilidad de predecir incorrectamente que el estudiante si se gradúa es del 8.05 %.



Conclusiones

Conclusiones

- Este proyecto propone una herramienta para predecir la posibilidad de graduación para estudiantes ya admitidos al RUM, no es una herramienta para decidir si se admite un estudiante o no.
- Que se logre detectar estudiantes con bajas probabilidades de graduación, dependerá de la implementación efectiva de un programa de intervención. Ese programa puede contar con
 - Consejería académica.
 - Asistencia en manejo de tiempo y estrés.
 - Orientación de programas en el RUM que ayudan al estudiante.



Conclusiones

Trabajo Futuro

- Es importante resaltar que como los métodos tienen que ser eficientes la intervención de los oficiales también lo debe ser.
- Considerar XGBoost un método de ensemble de aprendizaje automático que ha superado a otros métodos en diferentes competiciones de Kaggle.
- Planificar nuevos modelos ML a implementar una vez intervención rinda frutos.
- Los modelos desarrollados
 - Se pueden entrenar añadiendo el recinto de la UPR al que pertenece el estudiante.
 - Se pueden construir intervalos de confianza para la predicción de graduación.



Contenido

- 1 Introducción
- 2 Objetivo
- 3 Metodología
- 4 Resultados
- 5 Conclusiones
- 6 Referencias



Referencias



Lorberfeld A. 2019.

Machine Learning Algorithms In Layman's Terms, Part 1 [En Línea].

Disponible:

[Towards data science](#)



Tibshirani R., T. Hastie, y F. Jerome.

The Elements of Statistical Learning, Data Mining, Inference, and Prediction.
Second Edition, Springer New York, 745 pp.



Referencias



Tibshirani R., T. Hastie, D. Witten, y G. James.

An Introduction to Statistical Learning with applications in R.

First Edition, Springer New York Heidelberg Dordrecht London, 441 pp.



Arık, S. O. y T. Pfister. 2020.

TabNet: Attentive Interpretable Tabular Learning.

arXiv, [cs.LG], V. 4. <https://arxiv.org/pdf/1908.07442.pdf>



Referencias



Mera-Gaona M., Neumann Ú., Vargas-Canas R., López D. 2021.
Evaluating the impact of multivariate imputation by MICE in feature selection.

Front. Neurorobot [En Línea]. Disponible: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0254720>

