# COMPARACIÓN E INTERPRETACIÓN DE DIFERENTES MODELOS ESTADÍSTICOS Y DE APRENDIZAJE AUTOMÁTICO QUE PREDICEN TASAS DE DESERCIÓN EN LAS ESCUELAS PÚBLICAS DEL ESTADO DE NUEVA YORK

Resumen

Ha habido un continuo interés en los Estados Unidos y en todo el mundo en el tema de la deserción escolar secundaria porque los abandonos escolares tienen un alto costo para los individuos y para la sociedad. La investigación en diferentes países muestra que los abandonos son más propensos a quedarse sin empleo, quedarse sin empleo por más tiempo, tener menores ganancias, y durante su curso de vida, acumular menos riqueza. Este proyecto tiene como objetivo construir y comparar cuatro modelos diferentes que predicen las tasas de abandono escolar en las escuelas en el estado de Nueva York en el año de educación 2019-20-Regresión logística regularizada, Regresión de bosque aleatorio, Regresión de impulso de gradiente extremo, y Regresión basada en red neuronal artificial. Los modelos se comparan en términos de su precisión predictiva utilizando un error absoluto medio y R 2 como métricas de evaluación de rendimiento. Al mismo tiempo, el proyecto también tiene como objetivo entender por qué los modelos hacen una cierta predicción además de evaluar la precisión de su predicción. En este proyecto se utiliza la herramienta de interpretación modelo SHapley Additive exPlanations (o SHAP) para estimar la importancia de los predictores en cada modelo y evaluar la similitud de los predictores más importantes a través de los modelos.

# Introducción al problema

La graduación de la escuela secundaria se considera un nivel mínimo de educación para que las personas participen con éxito en el estudio y el trabajo. Esto se debe a que, en la mayoría de las naciones, la educación secundaria sirve como la base para la entrada a la universidad y otras oportunidades de educación y capacitación, así como la preparación para la entrada en el mercado laboral. Con el tiempo, cada vez es más importante decidir cómo se benefician los beneficios económicos y de otra índole, como la buena salud y el bienestar. . 1

Ha habido un continuo interés en los Estados Unidos y en todo el mundo en el tema de la deserción escolar secundaria porque los abandonos escolares tienen un alto costo para los individuos y para la sociedad. La investigación en diferentes países muestra que los abandonos son más propensos a quedarse sin empleo, quedarse sin empleo por más tiempo, tener menores ingresos, y durante su curso de vida, acumular menos riqueza (por ejemplo, ver Rumberger & Lamb, 2003; OCDE, 2001; Barro, 1997; Shavit & Mueller, 1998). Los abandonos también experimentan más a menudo una peor salud física y mental, tienen tasas de criminalidad más altas y menos a menudo se dedican a la ciudadanía activa (Owens, 2004; Rumberger, 1987). Además de los costes para las personas, también hay costes sociales asociados con el aumento de las necesidades de bienestar y la reducción de los ingresos fiscales (Owens, 2004). [1](#_bookmark0)

En los Estados Unidos, la tasa de abandono representa el porcentaje de jóvenes de 16 a 24 años que no están matriculados en la escuela y no han obtenido una credencial de secundaria (un diploma o una credencial de equivalencia). Las tasas de graduación de la escuela secundaria habían crecido dramáticamente en los EE.

cuando el 96% de todos los individuos de 18 años y más no habían completado la escuela secundaria. En la década de 1960, la tasa de no finalización era sólo de una cuarta parte de esta población. 2 Los datos de National Center for Education Statistics muestran que las tasas de abandono disminuyeron aún más al 8,3% en 2010 y al 5,1% por ciento en 2019. 3

Este proyecto tiene como objetivo construir y comparar cuatro modelos-Regresión logística regularizada, regresión de bosques aleatorios, regresión de aumento de gradiente extremo y regresión basada en la red neuronal artificial-que predicen las tasas de abandono escolar en las escuelas de Nueva York. En el año de educación 2019-20. La variable objetivo, la tasa de abandono, es el porcentaje de estudiantes de un género en particular que estudió como parte de una cohorte particular en una escuela en particular en el estado de Nueva York que, en la fecha de graduación de la cohorte, no están matriculados en la escuela y tienen no se ha ganado una credencial de secundaria. La predicción de las tasas de deserción se basa en varios factores sociales y económicos en el nivel de cohorte, escuela o condado descrito en la siguiente sección del proyecto.

Como parte de la comparación de modelos, el proyecto también tiene como objetivo identificar cuáles de los predictores están más fuertemente asociados con las tasas de deserción, utilizando herramientas de interpretación de modelos tales como ExPlanaciones Aditivas de SHapley descritas más adelante en el proyecto. Se puede suponer que la manipulación en los predictores que tienen la asociación más fuerte con la variable objetivo cambiaría las tasas de abandono en el futuro. Sin embargo, a menudo puede ser engañoso, ya que hay una diferencia fundamental entre la correlación y la causalidad. Este proyecto pretende transparentar las correlaciones recogidas por los modelos predictivos. Sin embargo, hacer que las correlaciones entre la variable objetivo y los predictores sean transparentes no las hace causales. Los modelos predictivos asumen implícitamente que los patrones de correlación se mantendrán constantes, y para entender lo que sucede en las variables de caso en el modelo cambian su comportamiento, se deben construir diferentes tipos de modelos. Estos modelos se llaman modelos causales, y construirlos requiere hacer suposiciones y usar las herramientas de análisis causal.

La construcción de un modelo causal está fuera del alcance de este proyecto.

# Recogida y preparación de datos

Los datos utilizados en el proyecto son una combinación de conjuntos de datos de diferentes orígenes.

La primera fuente es la información educativa disponible públicamente proporcionada por el Departamento de Educación del Estado de Nueva York (NYSED). Se puede encontrar aquí: [https://data.nysed.gov/downloads.php .](https://data.nysed.gov/downloads.php)  Los conjuntos de datos de interés son "Base de datos de tarifas de graduación", "Base de datos de tarjetas de informe" y "Base de datos de estudiantes y educadores" para el año de educación 2019-20 ya que es la última disponible.

**La base de datos de tasas de graduación contiene datos anuales de graduación y de abandono del estado desglosados por condados, distritos o escuelas. Los datos también incluyen el número de estudiantes matriculados. Para cada uno de los niveles de agregación antes mencionados (es decir, distrito, país, escuela), los datos anuales de graduación se incluyen para las cohortes actuales de cuatro años, cinco años y seis años. Una cohorte es un grupo de estudiantes que trabajan a través de un plan de estudios juntos para lograr el mismo grado académico juntos.** 4 La longitud de cohorte estándar en la escuela secundaria en los Estados Unidos es de cuatro años, pero también hay cohortes extendidas de 5 y 6 años.

Además, cada nivel de agregación puede ser desagregado aún más por subgrupos demográficos como género, raza u otros (por ejemplo, estudiantes económicamente desfavorecidos o no). Por ejemplo, es posible extraer los datos para estudiantes blancos o estudiantes femeninos, pero no para estudiantes mujeres blancas. Así que es

no es posible seleccionar el género y la raza, ya que estas observaciones no son independientes entre sí; las observaciones para los estudiantes blancos incluirán a todos los estudiantes blancos de todos los géneros, incluidas las mujeres, y las observaciones para las mujeres estudiantes incluirán mujeres estudiantes de todas las razas, incluyendo el blanco. Por lo tanto, es importante establecer un único nivel de agregación para que las observaciones sean independientes. Para este proyecto, consideré la agregación de abandonos por la escuela y el género. En consecuencia, las tasas de abandono en el conjunto de datos utilizado para el modelado son para estudiantes de un género determinado que estudian como parte de una cohorte particular en una escuela en particular.

**La base de datos de la tarjeta de informe contiene el estado, el distrito, la escuela pública y la responsabilidad de la escuela, la tasa de graduación secundaria, los gastos por alumno y los datos de las calificaciones del personal También se incluyen la rendición de cuentas y los datos de tasa de graduación secundaria por condado y el grupo de Capacidad de Recurso. Report Card Database fue originalmente almacenado en la base de datos de MS Access, y la información sobre el gasto por alumno y profesores sin experiencia fue extraída en 2 archivos CSV respectivos que fueron fusionados por el ID de la escuela.**

**La base de datos de estudiantes y educadores contiene información para cada tasa de asistencia escolar, número de consejeros y trabajadores sociales, porcentaje de suspensiones y porcentaje de alumnos que son elegibles para el almuerzo gratuito.**

Cabe señalar que para garantizar la confidencialidad de los estudiantes, NYSED no publica datos para grupos con menos de cinco estudiantes o datos que permitirían a los lectores determinar fácilmente el rendimiento de un grupo con menos de cinco estudiantes. Cuando había menos de cinco estudiantes en un grupo (por ejemplo, hispanos), los resultados para esos estudiantes fueron suprimidos para ese grupo y el siguiente grupo más pequeño. Los datos suprimidos se indican con un "-" en el conjunto de datos original. Estas observaciones fueron descartadas.

La segunda fuente de datos son los datos del censo de EE.UU. para cada condado en el estado de Nueva York como, pero no limitado a la mediana de los ingresos de los hogares, por ciento de las personas en la pobreza, por ciento de los hogares con un ordenador/Internet de banda ancha, el número de personas por hogar, etc. Esta información públicamente disponible se toma del sitio web de la Oficina del Censo de los Estados Unidos. En particular, la información se toma de la herramienta de acceso a datos de QuickFacts ( <https://www.census.gov/quickfacts/>) que proporciona a los usuarios un acceso fácil a las estadísticas básicas de población, negocios y geografía de todos los estados y condados de los Estados Unidos y de las ciudades y pueblos con más de 5.000 personas.

Después de extraer los datos del Censo NYSED y EE.UU. y dejar caer la información considerada no relevante para el proyecto (consulte la lista completa de las variables restantes en [Cuadro 1](#_bookmark2) ), he realizado la preparación de los conjuntos de datos finales. Para ello, primero fusioné la base de datos de tasas de graduación y la base de datos de tarjetas de informe por ID de escuela y luego fusioné el conjunto de datos resultante con el condado de Censo de EE.UU. establecido por nombre de condado.

## Valores perdidos

El conjunto de datos resultante contenía valores perdidos porque faltaban algunas escuelas en la base de datos de tarjetas de informe. De 1.275 escuelas públicas en la Base de Datos de Tasa de Graduación, 39 escuelas estaban desaparecidas en la Base de Datos de Tarjetas de Informe y en la Base de Datos de Estudiantes y Educadores. Estas escuelas comprenden el 1,48% de todos los estudiantes. Teniendo en cuenta el pequeño número de escuelas que faltan, las eliminé de la población.

En consecuencia, el proyecto cubre todas las escuelas del estado de Nueva York, que informan información detallada a NYSED.

## Preparación de datos

Después de recopilar los datos y eliminar los valores perdidos, preparé el conjunto de datos para el modelado.

En primer lugar, he realizado una codificación en caliente de las variables categóricas *Cohorte* y *Sexo* . Hay que tener en cuenta que los géneros reportados son sólo hombres y mujeres, por lo tanto, sólo me he mantenido *gender\_female* variable, donde 1 valores se relacionan con las hembras, mientras que 0 valores están relacionados con los hombres.

Luego, dividí el conjunto de datos en formación (50%), validación (25%), y test (25%). Para evitar un sobreajuste, los modelos se entrenarán en el conjunto de datos de entrenamiento y luego se marcarán en el conjunto de datos de validación. El mejor modelo se seleccionará en función de sus métricas de rendimiento de validación. El conjunto de datos de prueba se utilizará para evaluar el rendimiento del modelo final seleccionado. Si el error en el conjunto de pruebas es significativamente peor que el error en el conjunto de validación, se alertará de que el modelo no es tan bueno como se esperaba: el error de validación está subestimado y debe ser reinvestig ated 5 .

Después de dividir el conjunto de datos, las variables numéricas del conjunto de datos de entrenamiento fueron normalizadas (es decir, centradas y escaladas). La normalización de los conjuntos de validación y prueba se ha realizado utilizando sólo los datos del conjunto de formación. Sólo es necesario utilizar el conjunto de entrenamiento ya que la validación y los conjuntos de pruebas desempeñan el papel de nuevos datos que todavía no se han visto, por lo que no debe utilizarse en la formación de los modelos. El uso de cualquier información de la prueba o validación establecida antes o durante el entrenamiento introduce un sesgo potencial en la evaluación del rendimiento de los modelos. La normalización de los datos utilizando el conjunto de entrenamiento presupone que tanto los datos de prueba como los de validación tienen la misma media y desviación estándar que el conjunto de entrenamiento.

La normalización de datos es un procedimiento importante que ayuda a la regresión logística regularizada, y los algoritmos de descenso de gradiente convergen.

# Análisis exploratorio

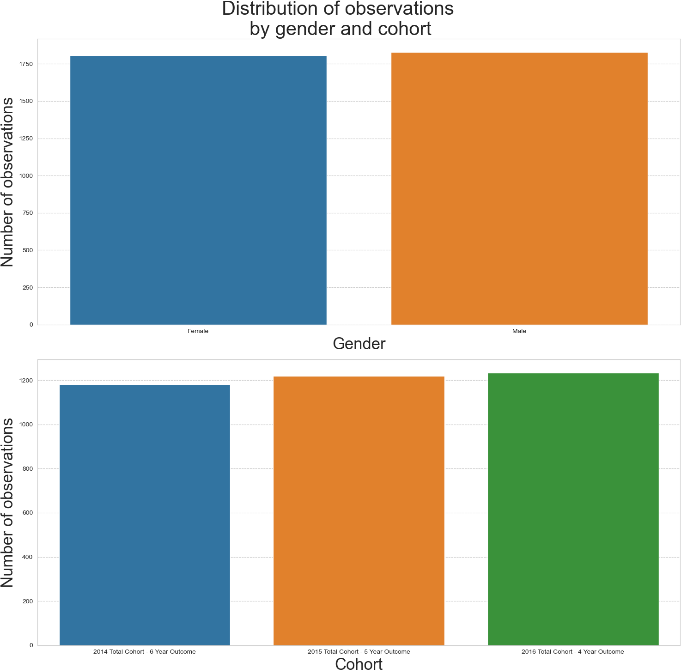
La población total utilizada en el proyecto consta de 7.269 observaciones, que representan 1.236 escuelas con hasta 3 cohortes en cada escuela y estudiantes de hasta 2 géneros en cada cohorte. [Cuadro 1](#_bookmark2)  La muestra las variables de destino y

*Cuadro 1. Lista de variables utilizadas en el proyecto*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Tipo variable** | **Nombre de variable** | **Descripción variable** |
| 1 | Objetivo  variable | Porcentaje de cuentagotas | Porcentaje de estudiantes que abandonaron (Variable de respuesta). |
| 2 | Predictores categóricos | Cohorte | Descripción de la pertenencia de la cohorte (el nombre original de la variable es MEMBERSHIP\_DESC). |
| 3 | Sexo | Género del grupo de estudiantes (Hombre/Mujer). El nombre original de la variable es SUBGROUP\_NAME. |
| 4 | Predictores numéricos de nivel escolar | PUPIL\_COUNT\_TOT | Número de alumnos en la escuela |
| 5 | PER\_FEDERAL\_EXP | Gasto por alumno utilizando fondos federales |
| 6 | PER\_STATE\_LOCAL\_EXP | Gasto por alumno utilizando fondos estatales y locales |
| 7 | PER\_TEACH\_INEXP | Número de profesores con menos de cuatro años de experiencia en sus puestos |
| 8 | ENSEÑANZA\_PER\_PUPILA | Número de profesores por 1 alumno en la escuela |
| 9 | ATTENDANCE\_RATE | Tasa anual de asistencia |
| 10 | PER\_SUSPENSIONES | Porcentaje de estudiantes suspendidos |
| 11 | PER\_FREE\_LUNCH | Porcentaje de estudiantes matriculados elegibles para el almuerzo gratuito |
| 12 | ACONSEJORS\_PER\_PUPILA | Número de consejeros por 1 alumno en la escuela |
| 13 | SOCIAL\_PER\_PUPILA | Número de trabajadores sociales por 1 alumno en la escuela |
| 14 | Predictores numéricos de nivel de condado | cty\_pop\_under\_18 | Personas menores de 18 años, |
| 15 | cty\_pop\_over\_65 | Personas de 65 años o más, por ciento |
| 16 | cty\_female | Mujeres, por ciento |
| 17 | cty\_foreign\_born | Personas nacidas en el extranjero, por ciento, 2016-2020 |
| 18 | cty\_owner\_copy | Tasa unitaria de vivienda ocupada por el propietario, 2016-2020 |
| 19 | cty\_housing\_unit\_median\_value | Valor medio de las unidades de vivienda ocupadas por el propietario, 2016-2020 |
| 20 | cty\_pers\_pers\_household | Personas por hogar, 2016-2020 |
| 21 | cty\_same\_house | Viviendo en la misma casa hace 1 año, por ciento de las personas de 1 año +, 2016-2020 |
| 22 | cty\_non\_english | Idioma distinto del inglés hablado en casa, por ciento de las personas edad 5  años +, 2016-2020 |
| 23 | cty\_with\_computer | Hogares con ordenador, porcentaje, 2016-2020 |
| 24 | cty\_with\_internet | Hogares con una suscripción a Internet de banda ancha, por ciento, 2016-2020 |
| 25 | cty\_hs\_grads | Graduado de secundaria o superior, por ciento de edad de 25 años +, 2016-2020 |
| 26 | cty\_uni\_degree | Licenciatura o superior, porcentaje de personas de 25 años +, 2016-  2020 |
| 27 | cty\_time\_to\_work | Tiempo medio de viaje al trabajo (minutos), edad de los trabajadores 16 años +, 2016-2020 |
| 28 | cty\_med\_household\_income | Renta familiar mediana (en dólares 2020), 2016-2020 |
| 29 | cty\_poverty | Personas en situación de pobreza, |
| 30 | cty\_pop\_density | Densidad demográfica, pop/sqft |

Los datos de capacitación constan de 3,634 observaciones, que representan el 50% de las observaciones seleccionadas al azar de la población total.

*Figura 1. Número de observaciones de cada género (superior) y cohorte (inferior)*



Mirando las variables categóricas en [Figura 1,](#_bookmark3)  podemos ver que los géneros son relativamente igualmente presentados en el conjunto de entrenamiento. Los varones (1.827) son algo más que Females (1.807), pero la diferencia es muy pequeña. Hay menos cohortes de 6 años en las escuelas (1.181) que 5 años (1.220) y 4 años (1.233). Aún así, las diferencias también son mínimas, y podemos suponer que el género y las cohortes se distribuyen por igual en la población.

[Figura 3](#_bookmark4)  La muestra la distribución de las tasas de abandono para cada variable categórica. Para todas las variables, podemos ver que la distribución de las tasas de abandono está sesgada hacia los valores cero tenía una cola larga, que se asemeja a la distribución de la ley de poder. Al mismo tiempo, también se puede ver que las tasas de abandono son más altas en promedio para los hombres que para las mujeres, ya que la distribución de las tasas de deserción para los hombres es

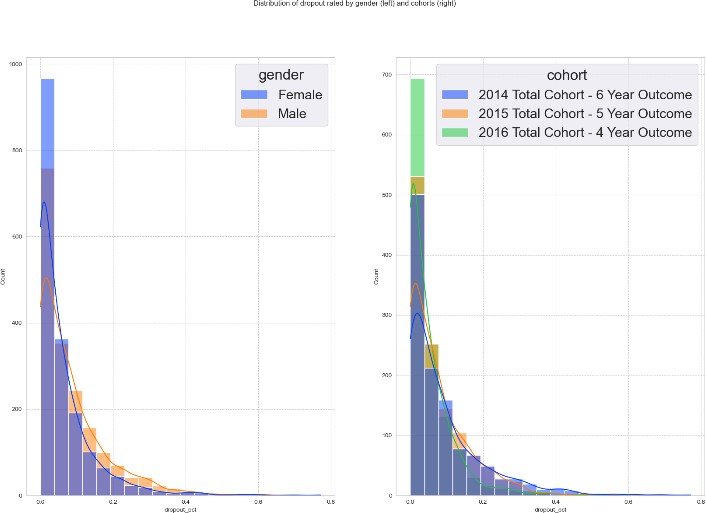
menos sesgado hacia cero, y la cola es más pesada. Lo mismo puede decirse de los estudiantes de cohortes de 6 años

en comparación con los estudiantes de 5 años y 4 años, y de 5 años en comparación con la cohorte de 4 años.

Como se muestra en [Figura 2 ,](#_bookmark5)  las tasas de abandono no se correlacionan con ninguno de los predictores continuos. La correlación positiva más fuerte es 0.48 con "PER\_FREE\_LUNCH" y la correlación negativa más fuerte de -0,58 es con "ATTENDANCE\_RATE". Si para observar la correlación entre cada par de variables predictoras continuas, 9 pares de variables tienen una correlación muy fuerte de más de 0,8 absolutos en valor ( [Cuadro 2](#_bookmark6) ). El diagrama de matriz de correlación para todos los predictores de continuidad se presenta en el Apéndice A.

*Figura 3. Distribución de tasas de abandono por sexo (izquierda) y cohortes (derecha)*

*Figura 2. Relación de la tasa de abandono con predictores numéricos*



La fuerte correlación es una indicación de un problema de multicolinealidad. La multicolinealidad ocurre cuando una o más de las variables independientes en el modelo pueden ser determinadas aproximadamente por algunas de las otras variables independientes. Cuando hay multicolinealidad, los coeficientes de regresión estimados del modelo ajustado pueden ser muy poco fiables. En consecuencia, cualquier estrategia de modelado debe comprobar la posible multicolor en varios pasos del proceso de selección de variables. 6

*Tabla 2. Predicciones continuas con correlación absoluta por encima de 0,8*

|  |  |  |
| --- | --- | --- |
| **Variable 1** | **variable 2** | **Correlación Pearson** |
| cty\_non\_english | cty\_foreign\_born | 0.97 |
| cty\_non\_english | cty\_owner\_copy | -0.82 |
| cty\_with\_internet | cty\_with\_computer | 0.91 |
| cty\_hs\_grads | cty\_non\_english | -0.82 |
| cty\_time\_to\_work | cty\_foreign\_born | 0.91 |
| cty\_time\_to\_work | cty\_non\_english | 0.92 |
| cty\_med\_household\_income | cty\_with\_internet | 0.86 |
| cty\_pop\_density | cty\_owner\_copy | -0.87 |
| cty\_pop\_density | cty\_housing\_unit\_median\_value | 0.89 |

En el valor de alta dimensión, el problema de la multicolinealidad es extremo: cualquier variable del modelo se puede escribir como una combinación lineal de todas las demás variables del modelo. Esencialmente, esto significa que nunca podemos saber exactamente qué variables (si las hay) son realmente predictivas del resultado, y nunca podemos identificar los mejores coeficientes para el uso en la regresión. 7

En resumen, la multicolinealidad generalmente no afectará el rendimiento de los modelos, especialmente de los modelos basados en árboles que utilizan el embolsado; sin embargo, no es posible interpretarlos. Esto es aplicable tanto para las estimaciones de parámetros en los modelos de regresión clásica como la regresión logística y el cálculo de importancia de características para técnicas complejas de aprendizaje automático, como los valores SHAP que se describen más adelante en el informe del proyecto.

La multicolinealidad no es un problema en el modelado predictivo, donde el objetivo es utilizar las asociaciones entre predictores y la variable de resultado para generar buenas predicciones para futuros resultados. El modelado predictivo tiene como objetivo maximizar la precisión predictiva, y las variables utilizadas en un modelo predictivo se basan en la asociación, no en la significación estadística ni en el significado científico.

En contraste, en el modelado explicativo, el enfoque principal es identificar las variables que tienen una relación científicamente significativa y estadísticamente significativa con un resultado. El objetivo principal es poner a prueba las hipótesis teóricas, enfatizando ambas relaciones teóricamente significativas y determinando si cada relación es estadísticamente significativa.

Aunque no realizo pruebas de hipótesis en el proyecto, sin embargo tengo el objetivo de arrojar luz sobre lo que hay detrás de los modelos de caja negra, como la red neuronal artificial basada en la red o la regresión de XGBoost.

Por lo tanto, para tener modelos interpretables, es importante abordar el problema de la multicolinealidad. En este proyecto, reduce el efecto de la multicolinealidad seleccionando variables utilizando la regularización, como se describe en la sección siguiente.

# Construir los modelos

### Definición de métricas de rendimiento y función de pérdida

Las técnicas de aprendizaje supervisado se utilizan cuando hay un conjunto de variables que pueden ser denotadas como entradas, que se miden o preestablecen, y estas entradas tienen cierta influencia en una o más salidas, mientras que el objetivo es utilizar las entradas para predecir los valores de las salidas. En la literatura estadística, los insumos son a menudo llamados predictores o variables independientes. En el aprendizaje automático, también se llaman características. Los tres términos anteriores se utilizan indistintamente en el informe. Las salidas se denominan variables de respuesta, dependientes o de destino.

Las respuestas varían en la naturaleza entre los ejemplos. En algunos casos, las respuestas son cuantitativas, como, por ejemplo, las tasas de deserción entre las diferentes escuelas. Mientras que en otras, las respuestas son cualitativas y pueden tomar sólo un número finito de valores, tales como, por ejemplo, si un estudiante particular abandonó o no. Esta distinción en el tipo de respuestas ha llevado a un convenio de denominación para las tareas de predicción: regresión cuando predigo salidas cuantitativas y clasificación cuando predigo salidas cualitativas. En otras palabras, la clasificación intenta predecir una clase de un elemento de datos, mientras que la regresión intenta predecir un valor.

Al desarrollar modelos de regresión y clasificación, la pregunta clave es estimar las estimaciones de los parámetros que proporcionarían la predicción más precisa. La estimación de parámetros es el proceso de identificar parámetros de modelo basados en un conjunto de entrenamiento dado que es probable que produzcan un buen rendimiento de predicción. Esto se puede ver como un proceso de optimización en el que se busca el espacio de posibles vectores de parámetros para uno que optimiza una medida de rendimiento adoptada.

La diferencia importante entre la clasificación y la regresión es la función de pérdida que se utiliza para elegir estos parámetros. La pérdida es la función de costo utilizada para evaluar errores, y así entrenar el predictor. La formación de un clasificador implica el uso de una pérdida que penaliza los errores en la predicción de clases de alguna manera, y la formación de un medio de regeneración significa utilizar una pérdida que penaliza los errores de predicción [. 9](#_bookmark8)

La variable de respuesta en este proyecto es la tasa de abandono. La tasa de deserción representa la proporción de estudiantes fuera del total que dejó la escuela antes de graduarse. Por un lado, la variable objetivo procede de la distribución binomial. La tasa de abandono es una manera de mostrar la proporción de éxitos (número de estudiantes que abandonaron) en el número total de ensayos (número total de estudiantes), por lo que encontrar las tasas de abandono puede considerarse un problema de clasificación. Sin embargo, por otro lado, la respuesta en este proyecto, las tasas de abandono, se modela como una variable continua que toma valores entre 0 y 1. Los problemas asociados a esto se describen en detalle en "Conclusión y sección de discusión".

Las diferentes funciones de pérdida se aplican a varios problemas de regresión. Las funciones de pérdida de clave incluyen el error cuadrático medio (MSE) y el error absoluto medio (MAE). La diferencia entre estas dos funciones de pérdida es que MSE pone mucho más énfasis en observaciones con grandes residuos absolutos

|y i -f (y i ) | durante el proceso de ajuste. Por lo tanto, es mucho menos robusto, y su rendimiento se degrada severamente para las distribuciones de errores de cola larga y especialmente para valores y valores muy mal medidos ("valores atípicos"). Otros criterios más robustos, como el MAE, funcionan mucho mejor en estas situaciones. En

la literatura de robustez estadística, se han propuesto una variedad de criterios de pérdida de regresión que proporcionan una fuerte resistencia (si no la inmunidad absoluta) a los valores brutos brutos mientras que son casi tan eficientes como los mínimos cuadrados para los errores de Gauss. A menudo son mejores que para las distribuciones de errores con

Colas moderadamente pesadas. Uno de estos criterios es el criterio de pérdida de Huber utilizado para la regresión de M (Huber, 1964). La función de pérdida de huber combina las buenas propiedades del error de predicción de error al cuadrado está cerca de cero, y el error de predicción de error absoluto es larg e 8 .

Teniendo en cuenta que las funciones de pérdida representan la diferencia entre las observaciones reales y los valores de observación previstos por el modelo, se utilizarán los valores de comparación para determinar el grado en que el modelo se ajusta a los datos, así como si la eliminación de algunas variables explicativas es posible sin dañar significativamente la capacidad predictiva del modelo.

En este informe, utilizaré la función de pérdida de MAE para la estimación del parámetro del modelo basado en la red neuronal artificial.

Debe hacerse otra distinción importante entre la función de pérdida y las métricas de evaluación del rendimiento. La función de pérdida la utiliza el modelo para aprender la relación entre la entrada y la salida. La métrica de evaluación se utiliza para evaluar qué tan buena es la relación aprendida.

Para que todos los modelos sean comparables, seleccioné la misma métrica para evaluar su rendimiento. Elegí MAE como la métrica de rendimiento principal debido a su ventaja descrita anteriormente (es decir, robustez a valores atípicos). Se seleccionará el modelo con el MAE más bajo.

Además de MAE, también usaré R-squared (R) 2 ) para comparar la bondad de ajuste de los modelos:

Suma residual de cuadrados

∑ n  (y i − y i ) 2

R 2  = 1 −

= 1 − i= 1

Suma total de plazas

∑ n  (y i − yacid) 2

i= 1

*Ecuación 1. Coeficiente de determinación (R 2 )*

Dónde y i -valor observado de la variable de respuesta,

N ° i -valor predicho de la variable de respuesta por el modelo, y

y valor predicho por valor de variable de la variable de respuesta por modelo nulo. Nulo en este proyecto es simplemente el valor medio de la variable de respuesta.

R-squared puede tomar cualquier valor entre 0 y 1. Teóricamente, R-squared puede incluso ser negativo en caso de que el modelo predice la variable objetivo peor que el modelo nulo. La r-squared cuantifica cuánto de una gota en la suma residual de cuadrados se contabiliza por el ajuste del modelo propuesto y se refiere a menudo como el coeficiente de determinación. Esto se expresa en una escala útil como una proporción de la variación total de los datos. Los valores de R-squared que se acercan a 1 indican que el modelo es un buen ajuste. Los valores de R-squared inferiores a 0,5 sugieren que el modelo da lugar a un mal ajuste a los datos.

Uso dos métricas de rendimiento para la comparación de modelos para comprobar si están alineadas entre sí. Validación cruzada y división de datos

El objetivo de los modelos construidos en este proyecto es conseguir un regressor que funcione bien en futuros datos para los que podríamos nunca conocer la verdadera tasa de abandono, utilizando un conjunto de ejemplos de formación que incluyen valores de tasa de abandono observados. En otras palabras, el modelo de regresión debe generalizarse a partir de los datos de entrenamiento para probar datos. Aquí, es importante hacer una distinción entre error de entrenamiento y error de prueba. El error de entrenamiento de un modelo de regresión es la tasa de error (MAE seleccionada para este proyecto) en ejemplos utilizados para entrenar al regressor. Por el contrario, el error de prueba es un error en los ejemplos no utilizados para entrenar al regressor. Con las técnicas avanzadas de modelado que están disponibles en el mercado,

Los modelos podrían perfectamente ajustarse perfectamente a los datos de entrenamiento. Sin embargo, es posible que estos modelos no tengan errores de prueba pequeños porque el procedimiento de regresión se elige para hacerlo bien en los datos de entrenamiento. Este efecto a menudo se llama sobreajuste . 9 Se produce cuando la generalización de los datos de formación se realiza de forma deficiente en el conjunto de datos de prueba. Es bastante probable que un procedimiento de entrenamiento eficiente encuentre propiedades especiales del conjunto de datos de entrenamiento que no son representativos del conjunto de datos de prueba porque el conjunto de datos de entrenamiento no es el mismo que el conjunto de datos de prueba.

Para evitar un exceso de ajuste, no puedo solamente estimar el MAE para los datos utilizados para entrenar el modelo. Por lo tanto, necesito separar los datos en conjuntos de formación y validación, formar el modelo en un conjunto de formación y evaluar su rendimiento en un conjunto de validación. Un error de validación es el que más importa, como en este caso, obtendré una estimación de error imparcial, lo que significa que el valor estimado de error estará cerca del verdadero valor del error.

Puedo resolver este problema con la validación cruzada, que implica repetidamente: dividir los datos en ***k*** la formación y la validación se establecen de forma uniforme y aleatoria, formando un modelo en el conjunto de formación, evaluándolo en el conjunto de validación y, a continuación, promediando el error sobre todas las divisiones. Cada división diferente se suele llamar un pliegue, por lo que la validación cruzada a menudo se llama k-fold cross-validation. Por ejemplo, cuando elijo una k de 10, dividiré los datos en diez partes. Entonces cada una de las diez partes servirá como datos de validación una vez. Esto significa que las nueve partes restantes se utilizan como datos de entrenamiento. Este procedimiento arroja una estimación del probable rendimiento futuro de un modelo a expensas de un cálculo sustancial. A continuación, calcularé el error de validación final en los datos de validación que dividí por separado para este propósito. También utilizaré la validación cruzada para optimizar los parámetros de ajuste de los modelos, tales como: [Ecuación 7 ,](#_bookmark13)  e hiperparámetros para los modelos Random Forest y XGBoost listados en [Cuadro 5](#_bookmark18)  y [Cuadro 6](#_bookmark22)  respectivamente).

Al comparar diferentes modelos de este proyecto, también dividiré los datos en el conjunto de pruebas. Por consiguiente, utilizo la división de prueba de validación de trenes. Formo modelos en los datos de entrenamiento y, a continuación, referencia los modelos en los datos de validación, que es la base para seleccionar el modelo. Luego, por último, utilizo el modelo seleccionado y computo un error en el conjunto de pruebas. Un error de prueba cercano al error de validación confirmará que obtuve una estimación de error imparcial [5 .](#_bookmark1)

**Regresión logística penalizada**

## Base teórica

En primer lugar, debe tenerse en cuenta que la variable de respuesta, la tasa de abandono, es un dato proporcional, cuya característica principal es que está limitada por el intervalo [0, 1]. Los datos proporcionales se estudian frecuentemente en áreas como, por ejemplo, los ensayos de prevención.

Como datos continuos por su naturaleza, los datos proporcionales se pueden modelar utilizando modelos de regresión lineal.

= Y:i:X T Artículo

*Ecuación 2. Predicción de la regresión lineal*

Donde Yfying es el valor predicho de la variable de destino, X T -matriz de diseño y vector-vector de los parámetros del modelo (coeficientes).

Sin embargo, el modelado de los datos proporcionales utilizando la regresión lineal crea ciertos problemas. Algunos de los supuestos estadísticos subyacentes a los modelos de regresión lineal, como linealidad, normalidad y errores homogéneos, podrían ser violados debido a estas características especiales de datos de proporción. Otra cuestión importante es que las predicciones de regresión lineal no están limitadas para que los valores previstos puedan estar fuera del rango entre 0 y 1. Para superar parcialmente algunos de los problemas de datos mencionados anteriormente, una posibilidad es transformar los datos. Por ejemplo, en algunos campos, como la ecología, la transformación de la raíz cuadrada de arcsina ha sido durante mucho tiempo una práctica común para transformar datos proporcionales.

Una alternativa a los métodos de transformación, el modelo lineal generalizado, se ha vuelto más frecuente en el análisis de datos de proporción en estudios clínicos. El modelo maneja naturalmente problemas de no normalidad y heterogeneidad, y el uso de una función de enlace garantiza que los valores ajustados estarán exactamente dentro del rango deseado [0, 1]. Por ejemplo, la regresión logística, un modelo lineal generalizado común, es adecuado para un resultado binomial, donde la proporción se calcula como la proporción del número de eventos objetivo en el número total de ensayos, " n y de n ". El modelo logístico ajustado es a menudo más interpretable que un modelo transformado, y refleja las características subyacentes de los datos mejor que el regressi lineal obre. 10

Un GLM generaliza el modelo lineal normal al permitir una variable de respuesta con una distribución distinta a la normal, pero un miembro de la familia exponencial de distribuciones y una relación entre la respuesta y el componente lineal de la forma g (Y) = X T Donde g () es la función de enlace. La regresión logística es un ejemplo de GLM, donde la variable de respuesta está enlazada a los predictores a través de la función de enlace logit.

Y

T

logit (Y) = log () = X

1 − Y

*Ecuación 3. Función de enlace de Logit*

Resolviendo para Y, la ecuación anterior se puede reorganizar de la siguiente manera:

e X T Artículo 1

Y = 1 + e XTexc  = 1 + e −XTrep

*Ecuación 4. Función de enlace de Logit (transformada)*

La función anterior se llama la función Sigmoid, matemáticamente denotada como f (x) = 1

1 + e −x

.

convierte cualquier número real en un número entre 0 y 1. La regresión logística es generalmente

utilizado cuando la variable dependiente es binaria en la naturaleza, pero como el uso de una función de enlace logit garantiza que los valores ajustados estarán exactamente dentro del rango deseado entre 0 y 1, puede proporcionar mejores resultados al problema de regresión para datos proporcionales. La regresión logística es un modelo lineal para una transformación de la probabilidad.

Otros modelos de regresión clásica *beta* o *tobit* La regresión se puede utilizar para predecir los datos proporcionales. Sin embargo, utilizo la regresión logística y sus variaciones, como la regresión logística regularizada o la capa de activación sigmoide en la regresión basada en la red de neuronas artificiales (ANN) en este proyecto, ya que las diferentes variaciones de la transformación del logit son implementado en muchos conocidos módulos de Python (keras, sklearn, xgboost) y paquetes R (glmnet).

Los supuestos básicos para la regresión logística incluyen la independencia de los errores, la linealidad en la logit para las variables continuas, la ausencia de multicolinealidad y la falta de valores atípicos fuertemente influyentes. 11

En este proyecto, utilizaremos técnicas de regresión regularizada, que tienen como objetivo hacer una estimación sesgada de los parámetros de regresión. Cuando se estima en los datos de entrenamiento, las técnicas de regresión regularizadas aceptan errores más grandes para obtener mejores resultados en los nuevos datos de prueba, lo que se llama el comercio de desviación de sesgo. En otras palabras, el rendimiento de la formación se reduce introduciendo sesgos. Este sesgo reduce la varianza en las predicciones de nuevos datos. Estas técnicas también penalizan las variables correlacionadas, por lo que las suposiciones subyacentes de la regresión logística son en parte eludidas por los métodos de regresión logística penalizada, como Lasso, regresión de Ridge, y Elastic Net.

Además, todos estos métodos incluyen uno o varios parámetros de regularización descritos anteriormente, y ninguno de ellos tiene una regla definida para seleccionar los valores de estos parámetros. El valor óptimo se encuentra a través de un procedimiento de validación cruzada, pero existen varios métodos de validación cruzada, y pueden producir resultados diferentes. Como resultado, el resultado real de los parámetros de modelo de cualquiera de estos métodos de regresión penalizados no está completamente definido por el método, pero puede depender de las opciones del modelador. 12

Por lo tanto, no voy a hacer ningún supuesto y declaraciones sobre la optimalidad teórica de las estimaciones de los parámetros del modelo de regresión logística penalizada y otros modelos utilizados en este proyecto. En su lugar, seleccionaremos el modelo óptimo que realiza el mejor rendimiento en datos de validación a través de un procedimiento de validación cruzada.

### Estimación de parámetros de modelo

En un modelo lineal generalizado, estamos interesados en los parámetros del modelo 1 , ..., j que describen cómo la respuesta depende de las variables explicativas. Usamos el y observado y 1 , ..., y i para maximizar la función de log-verosimilitud (o minimizar el log-verosimilitud negativo). Para la regresión logística, la función log-verosimilitud será la siguiente:

n

l (valor, y) = ∑ y i ln (p i ) + (y i − 1) ln (1 − p i )

i= 1

*Ecuación 5. Función log-verosimilitud de la regresión logística*

### Regularización

Generalmente, la regresión por etapas hacia adelante y hacia atrás fueron estrategias para la selección de predictores con el fin de evitar que el modelo se ajustara. Un enfoque alternativo y muy poderoso es penalizar los coeficientes si son iguales a cero. El modelo resultante ignora las variables independientes correspondientes. Los modelos construidos de esta manera a menudo se llaman modelos dispersos porque (una espera) que muchas variables independientes tendrán coeficientes cero. Por lo tanto, el modelo utiliza un subconjunto disperso de los posibles predictores [. 9](#_bookmark8)

El paquete glmnet proporciona una manera automatizada de penalizar las estimaciones de los parámetros. Se ajusta a un modelo lineal generalizado (GLM) denominado vía de máxima verosimilitud penalizada. Concretamente, resuelve el siguiente problema:

1

argminmin (−

N

l (valor, y) + r (valor)

*Ecuación 6. Función objetiva de la regresión logística penalizada*

Donde l (es decir, y)-son las funciones de log-verosimilitud descritas anteriormente.

r (valor)-función que introduce la penalización del parámetro. N-número total de observaciones de formación.

Puesto que esta fórmula utiliza un argmin, es una búsqueda numérica de los valores óptimos para las betas. Esto significa que el ajuste del modelo se realiza simplemente mediante la búsqueda iterativa a través de diferentes valores para las betas e identificar qué valores para las tetas conducen al valor más bajo para el resto de la función objetivo.

Para minimizar la función objetivo, el paquete glmnet utiliza un algoritmo de "Newton proximal". Este algoritmo hace un uso repetido de una aproximación cuadrática a la probabilidad de registro, y luego el descenso de coordenadas ponderadas en el problema de mínimos cuadrados ponderado resultante. Estos constituyen un bucle externo e interno, también conocido como un cuadrado menos penalizado de reponderada iterativamente s. 13

La expresión general de la función de penalización r (en inglés) para Elastic Nets es la siguiente:

p p

(1 − 1)

() 2

R:

= número

2 ∑ j

j= 1

+ alc. j |)

j= 1

*Ecuación 7. Función de penalización para regresiones regularizadas*

La regularización elástica de Net combina las penalizaciones de cresta y lasso para conseguir lo mejor de ambos mundos. En la expresión anterior, el parámetro es un parámetro de "mezcla" que equilibra las regularizaciones de R1 y R2. Si el valor es cero, obtenemos la regularización L2, y si el valor es uno, entonces obtenemos la regularización L1.

El valor > 0 es un hiperparámetro conocido como parámetro de regularización (también conocido como parámetro de ajuste), que determina la fuerza global de la penalización. La trayectoria de regularizacion se calcula para la lasa o penalidad neta elastica en una rejilla de valores (en la escala de registro) para el parametro de regularizacion.

Se sabe que la penalidad de cresta encoge los coeficientes de predictores correlacionados entre sí mientras que el lasso tiende a escoger uno de ellos y desechar los otros. La penalidad de la red elástica mezcla estos dos: si los predictores están correlacionados en grupos, un valor = 0.5 tiende a seleccionar o dejar fuera todo el grupo de características. Se trata de un parámetro de nivel superior, y los usuarios pueden elegir un valor inicial o un experimento con unos pocos valores diferentes [. 13](#_bookmark12)

La regresión de Ridge, al penalizar las estimaciones de los parámetros del modelo, los establece muy cerca de cero, pero no igual a cero, lo que significa que no elimina los predictores del modelo. Como el principal objetivo de la regularización para este proyecto es realizar la selección de variables, voy a construir sólo Lasso y Elastic Nets, pero no la regresión de Ridge.

## Aplicación en el proyecto

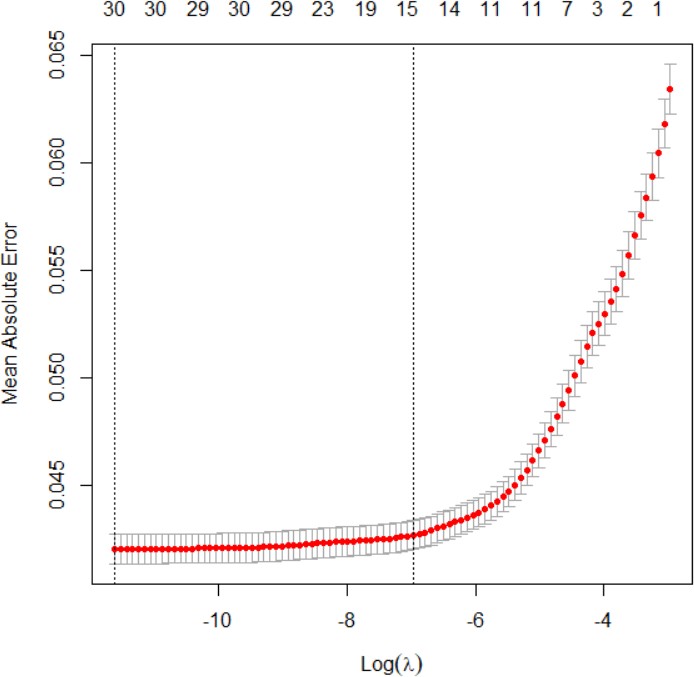
Los modelos de red Lasso y Elastic se construyen utilizando el paquete R glmnet.

Como se ha mencionado anteriormente, sólo es necesario ajustar el hiperparámetro de la regresión en la regresión de Lasso, ya que otro hiperparámetro se fija en 0. El valor del hiperparámetro de regularización óptimo se estimó a través de la validación cruzada de 10 pliegues que es la función de construcción-en cv.glmnet del paquete glmnet.

[Figura 4](#_bookmark14)  A continuación se muestra cómo el MAE del modelo regularizado L1 cambió con diferentes valores de

hiperparámetro.

*Figura 4. MAE contra log-λ en regresión logística regularizada L1*



El error absoluto medio es muy bajo desde el principio, lo que significa que el modelo va un buen trabajo incluso sin regularización. Sin embargo, nuestro objetivo es la selección de variables, y podemos ver que para la regularización L-1, el valor MAE se mantiene casi igual hasta que quedan 15 variables en el modelo (valores en la parte superior de la trama) y sólo entonces comienza a subir. Esto indica que incluso la regularización al eliminar casi la mitad de las 31 variables iniciales del modelo no resulta en una peor calidad de predicción. También hay dos líneas verticales en la trama. El valor de MAE mínimo correspondiente a la anotación cronológica ("log" en este informe significa registro natural) de -11.618. La otra línea vertical está dentro de un error estándar del MAE mínimo (es decir, la llamada regla "one-standard-error") correspondiente al valor log-λ de -6.966. La segunda línea es un poco

modelo más restringido (puesto que se refiere a un valor más grande para el sistema). La trama muestra que no hay mejora en el MAE con el valor creciente del parámetro de regularización. Sin embargo, el valor MAE es casi el mismo en el modelo con 15 predictores que en el modelo que incluye todas las variables. Sobre esta base, seleccionaré el modelo final de L1-regularizado con valor de referencia dentro de un error estándar de uno con el MAE mínimo.

*Cuadro 3. Rendimiento de validación de Lasso y Elastic Net*

|  |  |  |
| --- | --- | --- |
| **Métrica** | **Regresión lasso** | **Redes elásticas** |
| MAE | 0.042 | 0.047 |
| R 2 | 0.50 | 0.45 |

La red elástica también se construyó utilizando el paquete glmnet. A diferencia de la regresión de Lasso, Elastic net tiene dos hiperparámetros de ajuste. El paquete glmnet no tiene una función estándar en la función para ajustar tanto los hiperparámetros como los de la versión de los datos, por lo que hemos utilizado el paquete caret. El paquete Caret ejecuta una validación cruzada de 10 pliegues de varios modelos con diferentes ajustes

valores de parámetros y deriva los valores que minimizan el MAE. Después de realizar la validación cruzada del modelo de redes elásticas, obtuve valores óptimos de hiperparámetro = 0.1 y de = 0.0446.

Después de crear ambos modelos con hiperparámetros óptimos, calculé sus métricas de rendimiento (MAE, R 2 ), que se presentan en [Cuadro 3 .](#_bookmark15)

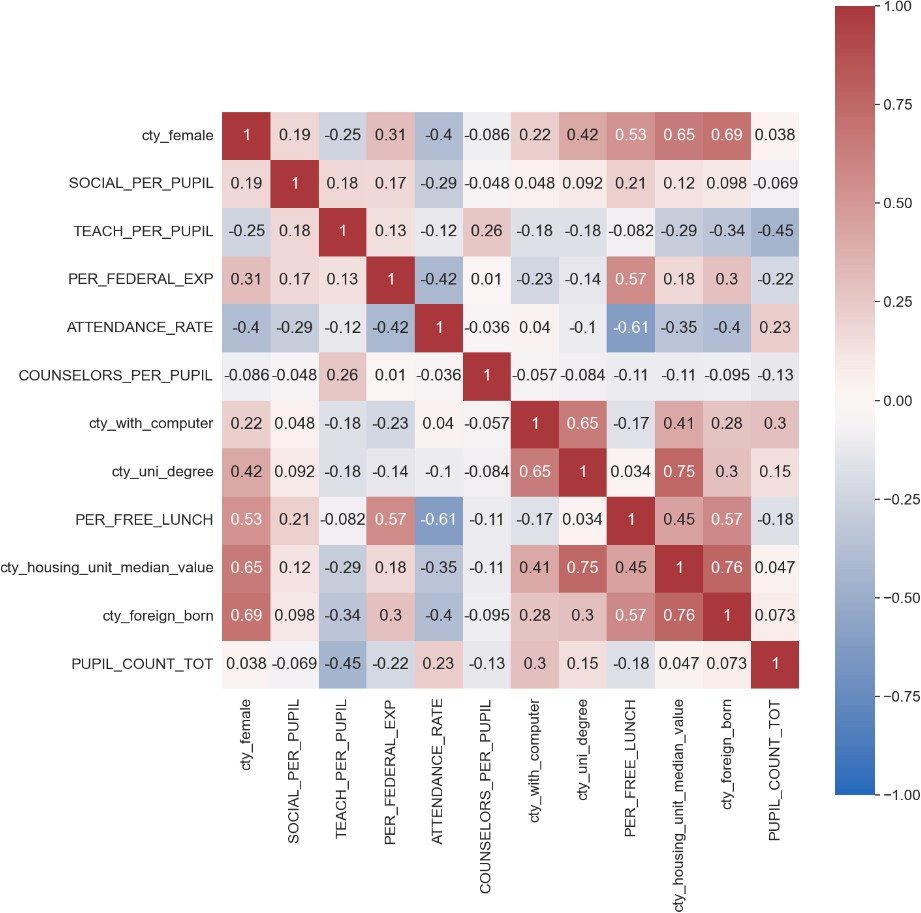
La tabla muestra que la regresión de Lasso funciona mejor que la red Elastic. Por lo tanto, seleccioné la regresión de Lasso para la selección de variables y la comparación con otros modelos más complejos analizados más adelante en el informe. Los predictores que se han mantenido y eliminado del modelo se presentan en [Cuadro 4 .](#_bookmark16)

*Cuadro 4. Predictores que se han mantenido y eliminado del modelo*

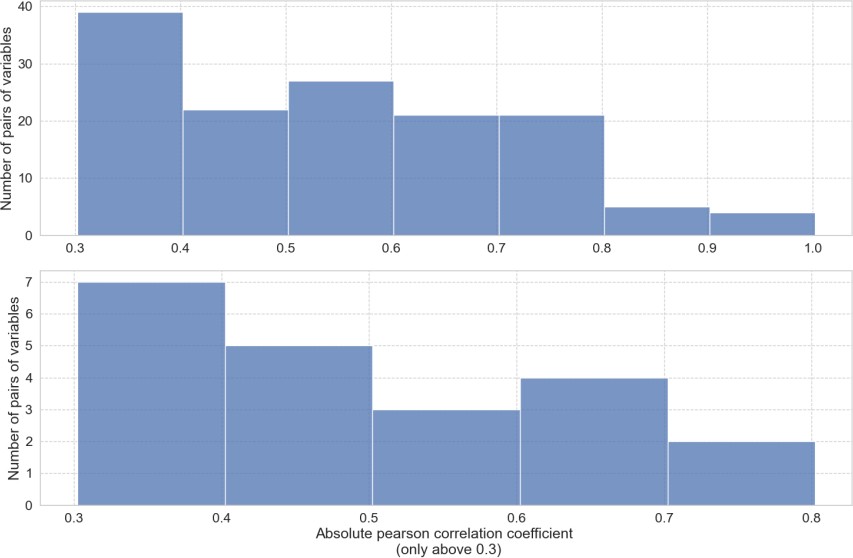
|  |  |  |  |
| --- | --- | --- | --- |
| **Los predictores permanecen en el modelo** | **coeficiente** |  | **Predictores eliminados del modelo** |
| PER\_FREE\_LUNCH | 0.6234 | "cohort\_2015 Total Cohort-Resultado De 5 Años" |
| ATTENDANCE\_RATE | -0.3110 | PER\_STATE\_LOCAL\_EXP |
| "cohort\_2016 Total Cohort-Resultado De 4 Años" | -0.2028 | PER\_TEACH\_INEXP |
| cty\_foreign\_born | -0.1832 | PER\_SUSPENSIONES |
| gender\_Female | -0.1800 | cty\_pop\_under\_18 |
| cty\_uni\_degree | -0.1001 | cty\_pop\_over\_65 |
| ENSEÑANZA\_PER\_PUPILA | 0.0928 | cty\_owner\_copy |
| "cohort\_2014 Total Cohort-Resultado De 6 Años" | 0.0699 | cty\_pers\_pers\_household |
| ACONSEJORS\_PER\_PUPILA | 0.0618 | cty\_same\_house |
| PER\_FEDERAL\_EXP | 0.0611 | cty\_non\_english |
| cty\_female | -0.0562 | cty\_with\_internet |
| cty\_with\_computer | 0.0553 | cty\_hs\_grads |
| PUPIL\_COUNT\_TOT | 0.0444 | cty\_time\_to\_work |
| cty\_housing\_unit\_median\_value | -0.0392 | cty\_med\_household\_income |
| SOCIAL\_PER\_PUPILA | 0.0085 | cty\_poverty |
|  | | | cty\_pop\_density |

La matriz de correlación presentada en [Figura 5](#_bookmark17)  y la distribución de coeficientes de correlación absolutos para pares de variables antes y después de la selección de variables muestran que no hay variables con correlación de Pearson absolutas por encima de 0,8 entre el conjunto de variables restantes, y sólo dos pares tiene un coeficiente de correlación absoluto de más en el rango entre 0,7 y 0,8.

*Figura 5. Matriz de correlación de los predictores continuos restantes*



*Figura 6. Distribución del coeficiente de correlación de Pearson absoluto de pares de variables antes del conjunto inicial (gráfico anterior) y después de la selección (gráfico a continuación)*



Después de la selección de variables, construyo otros modelos de regresión más complejos-bosque aleatorio, impulso de gradiente extremo y modelos basados en redes neuronales artificiales.

**Bosque aleatorio**

## Base teórica

El modelo de bosque aleatorio es un modelo fácil de usar, y se sabe que muestra un alto rendimiento tanto para la regresión como para los problemas de clasificación.

El Bosque Aleatorio está fuertemente basado en el modelo de Árbol de Decisiones. La idea detrás del modelo de regresión del árbol de decisión se puede entender intuitivamente como una larga lista de sentencias if-else. Cada nodo divide el árbol en dos nodos nuevos y representa la condición. Dos bordes procedentes del nodo representan los posibles resultados, es decir, si la condición está satisfecha o no, y están llevando a un nodo de decisión adicional o a un nodo terminal que muestra el valor predicho de la variable de respuesta. Por lo tanto, podemos decir que el Árbol de decisiones es una ordenación jerárquica de múltiples divisiones de decisión (llamadas hojas). La pregunta lógica es de dónde vienen esas decisiones. A diferencia de los modos de árbol de decisión de clasificación, en los que la división se basa en la minimización del índice de entropía o Gini de la información, la calidad de una división en los árboles de regresión se evalúa en función de los errores cuadrados o absolutos entre el valor promedio de la observaciones dentro del nodo y sus valores reales. Como se ha mencionado anteriormente, he seleccionado errores absolutos como criterio de evaluación de rendimiento para este proyecto. Las predicciones finales del nodo terminal son la media del subconjunto de formación de respuestas dentro del nodo.

Random Forest añade más complejidad al modelo de árbol de decisiones. Como su nombre indica, un Bosque Aleatorio consiste en un gran número de árboles de decisión, cada uno de ellos con una ligera variación.

El bosque aleatorio utiliza el embolsado para crear un conjunto de árboles de decisión. Generalmente, un bosque aleatorio puede combinar cientos o incluso miles de modelos de árboles de decisión. Además, al dividir un árbol, los bosques aleatorios sólo considerarán un subconjunto aleatorio de posibles predictores. Es decir, ignora intencionadamente un conjunto aleatorio de variables. Cada vez que se considera una nueva división, un nuevo aleatorio

se dibuja un subconjunto de predictores. Si hay un único predictor dominante en el conjunto de datos, la mayoría de los árboles utilizarán el mismo predictor para la primera división y garantizarán la correlación y la similitud entre los árboles, por lo que el uso de un subconjunto aleatorio de predictores en cada árbol de decisión aleatorio asegura que el bosque aleatorio se compone de árboles no correlacionados.

Por lo tanto, cada uno de estos modelos se instalará en un subconjunto aleatorio de datos de entrenamiento (dibujado con reemplazo) y un subconjunto aleatorio de predictores para no ser totalmente iguales. Entonces la predicción de una nueva instancia se basa en el promedio entre la predicción de todo el conjunto de estos modelos. Cuando un modelo de aprendizaje automático a veces puede estar equivocado, la predicción promedio de un gran número de modelos de aprendizaje automático es menos probable que sea errónea. Esta idea es la base del aprendizaje de conjunto.

A continuación se muestra el algoritmo para construir un bosque aleatorio para problemas de regresión [8](#_bookmark7) : Para b = 1 a B:

1. Extraer una muestra de rutina de carga Z **\*** de tamaño *N* de los datos de formación.
2. Cultivar un árbol de bosque aleatorio *T b* a los datos de arranque, repitiendo recursivamente los pasos siguientes para cada nodo terminal del árbol, hasta el tamaño mínimo del nodo *n min* se alcanza.
   1. Seleccione *m* variables al azar de la *p* variables.
   2. Elija la mejor variable/punto de división entre *m* .
   3. Divida el nodo en dos nodos hija.
3. Salida del conjunto de árboles {T b } B) .

1

La predicción de un nuevo punto se calcula como una predicción promedio de todos los árboles en el bosque:

Artículo B) 1 B)

rf = ∑ T b (x)

Bf

b= 1

*Ecuación 8. Predicción realizada por regresión de bosques aleatorios*

El modelo de bosque aleatorio se construye utilizando las funciones de RandomForestRegressor en el paquete scikit-learn de Python. El modelo tiene muchos hiperparámetros que se pueden ajustar, como por ejemplo, pero sin limitarse a la profundidad máxima del árbol, el número mínimo de muestras necesarias para dividir un nodo interno, el número mínimo de muestras necesario para estar en un nodo de hoja, el número de características a tener en cuenta a la hora de buscar la mejor división.

Random Forest es una técnica computacionalmente eficiente que puede operar rápidamente sobre grandes conjuntos de datos. En general, más árboles proporcionan un mejor resultado. Sin embargo, la mejora disminuye a medida que aumenta el número de árboles. Por lo tanto, en un determinado momento, el beneficio en el rendimiento de la predicción de aprender más árboles será más bajo que el costo en el tiempo de cómputo para aprender estos árboles adicionales. Aunque los modelos de bosque aleatorio se han utilizado en muchos proyectos de investigación y aplicaciones del mundo real, la literatura asociada no proporciona casi ninguna dirección sobre cuántos árboles se deben utilizar para componer un bosque aleatorio. Basé mi decisión en el trabajo de T.M. Oshiro, P.S. Pérez, y J.A. Baranauskas, quien analizó el impacto del número de árboles en 29 conjuntos de datos para problemas de clasificación. Su análisis mostró que de 128 árboles, no hay diferencia más significativa entre los bosques usando 256, 512, 1024, 2048 y 4096 árboles. La media y la mediana de los valores de las AUC no presentan grandes cambios en 64 árboles. Por lo tanto, los investigadores concluyeron que, con base en los experimentos, un rango razonable para estar en un bosque aleatorio es entre 64

y 128, sin mayores mejoras traídas por un mayor número de árboles. 14 Por lo tanto, seleccioné 100 árboles, el número predeterminado de árboles en el paquete scikit-learn.

## Aplicación en este proyecto

Comprobé por prueba y error que los cambios en la profundidad máxima del árbol (max\_depth) y los nodos de hoja máxima (max\_leaf\_nodes) tuvieron el mayor impacto en el rendimiento del modelo. Así que seleccioné estos hiperparámetros para sintonizar a través de la validación cruzada de 5 pliegues.

Tanto la profundidad máxima de un árbol como el número máximo de nodos de hoja controlan la complejidad de los árboles. Más nodos equivalen a árboles más profundos, más complejos y menos nodos dan como resultado árboles más superficiales.

También calculé el número máximo óptimo de la característica m del total de p a través de la validación cruzada. Aunque algo de literatura recomienda m = p/3 o m = registro cronológico de datos 2 p − 1, en la práctica, los mejores valores para estos parámetros dependerán del problema y deben tratarse como parámetros de ajuste [. 8](#_bookmark7)

I explícitamente especificaba valores para cada hiperparámetro. Los valores óptimos se resaltan en **rojo** :

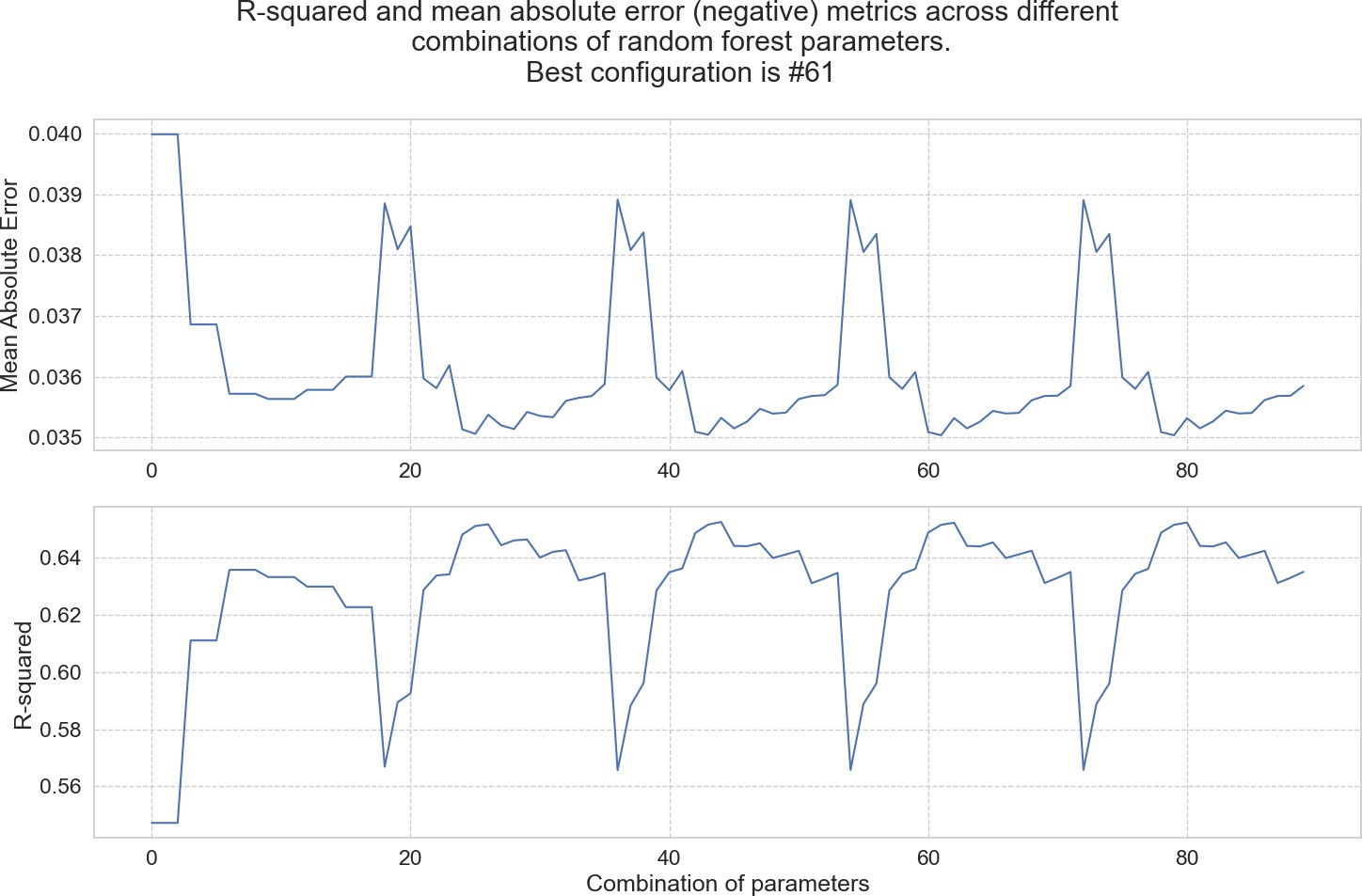
*Cuadro 5. Cuadrícula de hiperparámetros sintonizados a través de la validación cruzada en el bosque aleatorio (los valores óptimos están en rojo)*

|  |  |  |
| --- | --- | --- |
| **Nombre de hiperparámetro** | **Descripción del hiperparámetro** | **Valores** |
| max\_feature | Número de características en cada árbol | [1, 3, **6** , 9, 12, 15] |
| max\_leaf\_nodes | Número máximo de nodos hoja | [500, **1000** , 1500] |
| max\_depth | Profundidad máxima de los árboles | [10, 17, 25, **32** , 40] |

Los valores de todos los otros hiperparámetros no mencionados anteriormente siguen siendo el valor predeterminado.

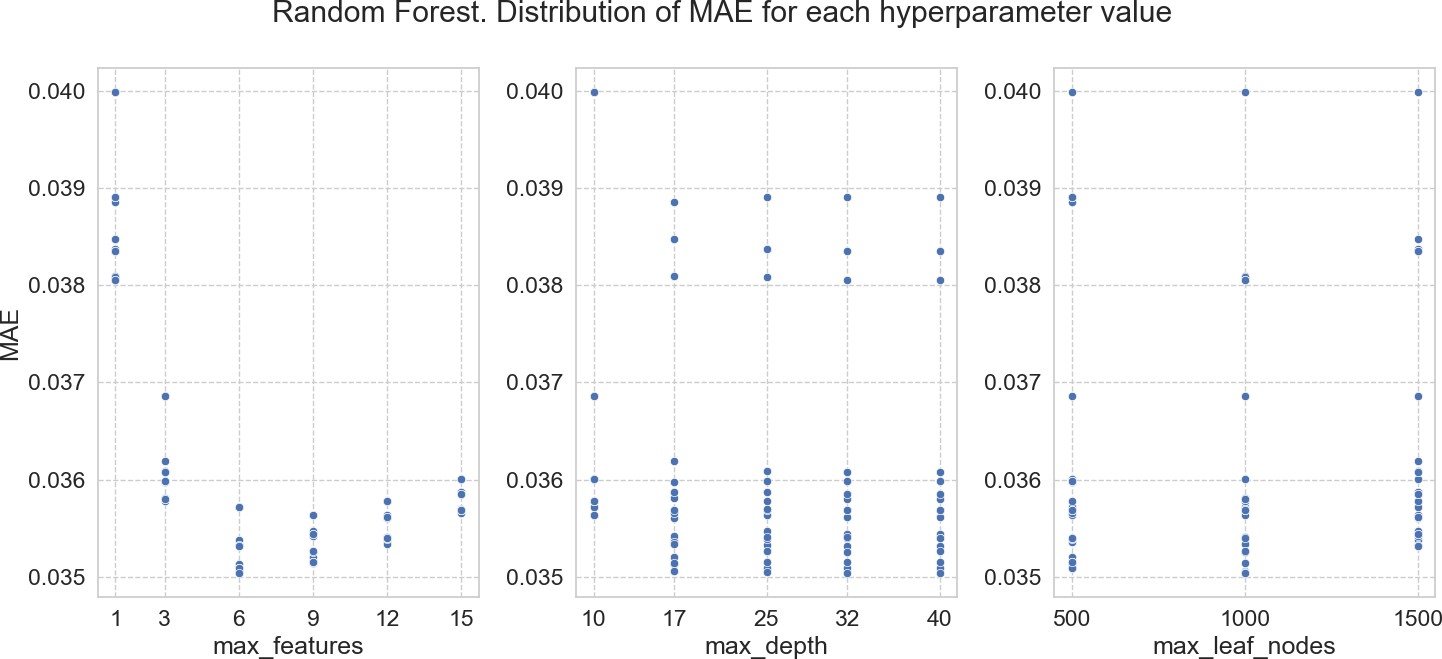
He realizado una búsqueda exhaustiva a través de todas las combinaciones posibles para valores óptimos utilizando la función GridSearchCV del paquete scikit-learn.

*Figura 7. Medidas de rendimiento de validación para toda la combinación de valores de hiperparámetros del modelo de bosque aleatorio en la cuadrícula de validación cruzada*



[Figura 7](#_bookmark19)  arriba muestra cómo el MAE y R 2 cambiar con diferentes combinaciones de hiperparámetros en la cuadrícula de búsqueda. Gráfico de MAE casi espejos R 2 , es decir, el valor más bajo de MAE corresponde al R más alto 2 y viceversa. La trama también muestra un patrón en el que 5 "colinas" son seguidas por 5 "valles". Indica que algún hiperparámetro en el conjunto domina los demás. Para comprobar esto, hice tres diagramas de dispersión a continuación, cada uno mostrando el cambio en MAE del modelo con el valor de un hiperparámetro fijo mientras que otros están cambiando. Se presentan en [Figura 8](#_bookmark20)  a continuación. La figura muestra que el número de características es el hiperparámetro más importante ya que el patrón de cambio en MAE se pronuncia más con un cambio en el número de características independientemente de los valores de otros hiperparámetros. En cuanto a la profundidad máxima, el MAE correspondiente al valor más bajo de "max\_depth" igual a 10 es mayor en promedio que el MAE correspondiente a los valores más grandes de "max\_depth" (es decir, 17, 25, 32, y 40), lo que hace que la profundidad de 10 sea demasiado baja como resultado del ajuste del modelo. Al mismo tiempo, no hay diferencia en MAE para valores de "max\_depth" superiores a 10 que pueden ser observables en la trama. Además, la trama no muestra una gran diferencia en el rendimiento del modelo para distintos valores de hiperparámetro "max\_leaf\_nodes".

*Figura 8. Bosque aleatorio. Cambio en MAE del modelo para cada valor de hiperparámetro fijo y todos los demás valores de otros hiperparámetros.*



En función de los resultados de validación cruzada, el valor óptimo de max\_depth es 32, max\_leaf\_nodes es 1000 y max\_feature es 6.

**Aumento de gradiente extremo (XGBoost)**

## Base teórica

Como el Bosque Random, XGBoost también utiliza conjuntos de árboles de decisión. Sin embargo, mientras que el bosque aleatorio utiliza un algoritmo de embolsado para crear un conjunto de árboles de decisión, XGBoost utiliza la técnica alternativa llamada impulso de gradiente. En cuanto al embolsado, el impulso de gradiente combina numerosos modelos pequeños de Decision Tree para hacer predicciones. La distinción crítica con el embolsado es cómo esos árboles de decisión vienen a ser diferentes entre sí.

El impulso es una técnica de conjunto donde se añaden nuevos modelos para corregir los errores que cometen los modelos existentes. Los modelos se añaden secuencialmente hasta que no se puedan realizar más mejoras.

El impulso es un proceso iterativo. Añade cada vez más estudiantes débiles al modelo de conjunto de una manera inteligente. En cada paso, se ponderan los puntos de datos individuales. Los puntos de datos que ya se preven bien no serán importantes para que el alumno sea agregado. Por lo tanto, los nuevos estudiantes débiles se centrarán en aprender las cosas que aún no se entienden y, por lo tanto, mejorar el conjunto.

Actualmente, uno de los algoritmos de impulso más populares es XGBoost. En lugar de estimar una predicción como un promedio sobre todos los árboles de decisión lista como el caso en Random Forest, la predicción de XGBoost es una suma de la predicción de todos los árboles que fueron entrenados a través de entrenamiento aditivo que puede ser escrito matemáticamente como en [Ecuación 9 .](#_bookmark21)  Una descripción detallada del algoritmo de XGBoost a continuación se toma de su página web oficial [https://xgboost.readthedocs.io](https://xgboost.readthedocs.io/) : 15

K

N ° i = ∑ f k (x i ), f k FUE

k= 1

*Ecuación 9. Predicción realizada por la regresión de XGBoost*

donde *K* es el número de árboles (también conocido como funciones aditivas), f es una función en el espacio funcional *F* , y F = {f (x) = w q (x) } (q: R m  → T, w r T ) es el espacio de árboles de regresión.

Aquí *q* representa la estructura de cada árbol que correlaciona un ejemplo con el índice de hoja correspondiente. *T* es el número de hojas en el árbol. Cada f k corresponde a una estructura de árbol independiente *q* y pesos de hoja *w* . A diferencia de los árboles de decisión, cada árbol de regresión contiene una puntuación continua en cada una de las hojas, usamos w i para representar la puntuación en la hoja i-th.

Para aprender el conjunto de funciones utilizadas en el modelo, el algoritmo minimiza la siguiente función objetivo regularizada.

n t

obj = ∑ l (y i , y de i ) + ∑ (f) k )

i= 1

k= 1

T

donde, (f) = (f) = T + 1 ∑ ∑ w 2

2 j

j= 1

*Ecuación 10. Fórmula general de funciones objetivo (obj) y regularización (Ω (f)) en la regresión de XGBoost*

Aquí l es una función diferenciable de pérdida convexa que mide la diferencia entre la predicción y la i y el destino y i El segundo término penaliza la complejidad del modelo (es decir, las funciones del árbol de regresión).

Para este proyecto, la función de pérdida se selecciona como "reg:logistic", que es la pérdida de registro y las predicciones de redondeo con

0,5 umbral, ya que la respuesta es continua. i

El término de regularización adicional ayuda a suavizar las ponderaciones finales aprendidas para evitar el exceso de ajuste. El control de hiperparámetro controla la fuerza de la regularización y el valor de referencia se refiere a una reducción de pérdida mínima necesaria para hacer una partición adicional en un nodo de hoja del árbol.

El modelo de conjunto de árboles en la ecuación anterior incluye funciones como parámetros y no se puede optimizar utilizando métodos de optimización tradicionales en el espacio euclidiano. En cambio, el modelo se entrena de manera aditiva, es decir, en cada paso, arreglamos lo que hemos aprendido y agregamos un árbol nuevo a la vez. Nosotros

escribir el valor de predicción en el paso t as y (t) . Entonces tenemos:

i

N ° (0)  = 0

i

N ° (1)  = f (x) = y (0)  + f (x)

i 1 i i i

1 i

N ° (2)  = f (x) + f (x) = y (1)  + f (x)

i 1 i

2 i i

...

2 i

i <https://stackoverflow.com/questions/53530189/the-loss-function-and-evaluation-metric-of-xgboost>

https://github.com/dmlc/xgboost/issues/521

t

N ° (t)  = ∑ f (x) = y (t− 1)  + f (x)

i k i i

k= 1

t i

*Ecuación 11. Predicción por un árbol en cada paso del bosque XGBoost*

Queda por preguntar: ¿qué árbol queremos en cada paso? Lo natural es añadir la que optimice nuestro objetivo.

n t n

obj (t)  = ∑ l (y, y (t) ) + ∑ (f) = ∑ l (y, y (t− 1)  + f (x)) + faxes (f) + constante

i= 1

i i

k

k= 1

i= 1

i i

t i t

*Ecuación 12. Función objetivo de la regresión de XGBoost en cada paso*

Más iteraciones de aumento de gradiente de número (es decir, número de estimadores) reduce los errores del conjunto de entrenamiento. Aumentar el número de iteraciones demasiado alto aumenta el sobreajuste. En este caso, la supervisión del error de predicción utilizando un conjunto de datos de validación separado ayudará a seleccionar el número óptimo de gradientes que aumentan las iteraciones. Además de sintonizar el número de estimadores, también es posible utilizar la profundidad de los árboles como un eficiente parámetro de regularización. El aumento de la profundidad de los árboles conduce al modelo que probablemente va a sobreajustarse a los datos de entrenamiento.

## Aplicación en este proyecto

Para evitar el sobreajuste y asegurar que el modelo mostrará un fuerte desempeño de nuevos datos, hice lo siguiente:

* Valores óptimos estimados de los siguientes hiperparámetros utilizando una validación cruzada de 5 veces. En cuanto a Random Forest, he realizado una búsqueda exhaustiva a través de todas las combinaciones posibles para valores óptimos utilizando la función GridSearchCV del paquete scikit-learn.

*Cuadro 6. Cuadrícula de hiperparámetros sintonizados a través de la validación cruzada en XGBoost (los valores óptimos están en rojo)*

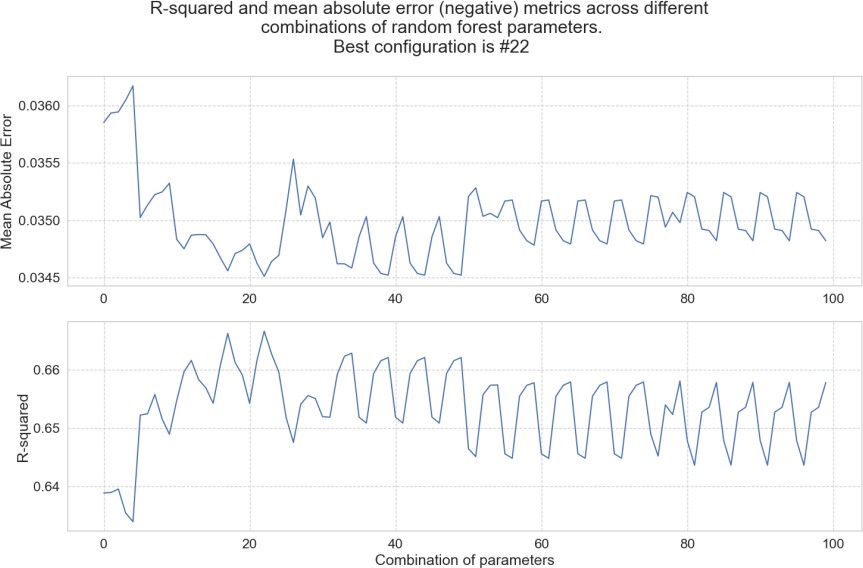
|  |  |  |
| --- | --- | --- |
| **Nombre de hiperparámetro** | **Descripción del hiperparámetro** | **Valores** |
| n\_estimadores | Número de árboles potenciados con gradiente.  Equivalente al número de rondas de impulso. | [100, 200, 300, 400, **500** ] |
| reg\_lambda | Resistencia de la regularización L2 | [4, 6, **8** , 10, 12] |
| max\_depth | Profundidad máxima de los árboles | [ **3** , 5, 7, 9] |

* Se ha activado la detención temprana al entrenar el modelo estableciendo que la métrica de validación necesita mejorar al menos una vez en cada 20 rondas para continuar con la formación. De lo contrario, la formación se detiene. Por lo general, es una buena idea seleccionar las rondas de detención como una función razonable del número total de épocas de entrenamiento (10% en este caso) o intentar corresponder al periodo de puntos de inflexión como se puede observar en las parcelas de aprendizaje. curvas. 16 Es importante tener en cuenta que la detención temprana se incluyó en el modelo XGBoost y en el neural artificial

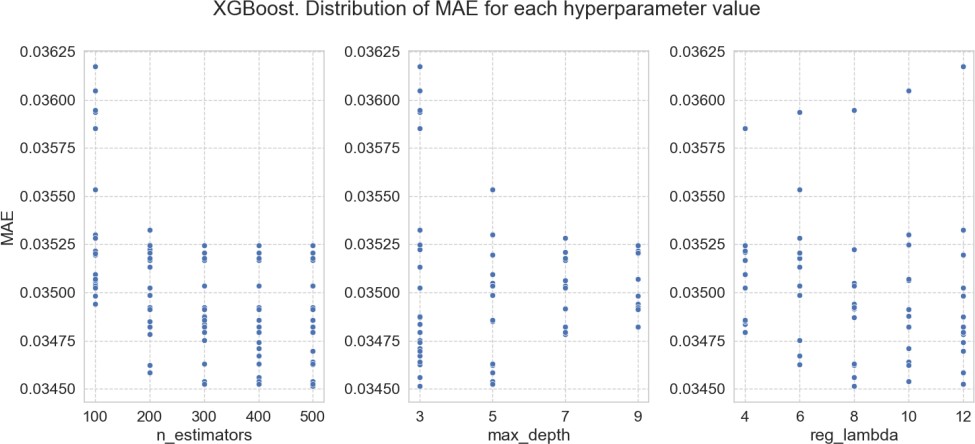
Modelo basado en la red a continuación, ya que estos modelos están haciendo que la solución sea más compleja con más iteraciones y es más probable que se sobreajusten. Como se mencionó anteriormente, el bosque aleatorio, a diferencia de estos modelos, hace una predicción como el promedio a través de un "bosque" de árboles de decisión construidos independientemente uno del otro, por lo que no requiere detenerse temprano.

[Figura 9](#_bookmark23)  La muestra el cambio en las métricas de rendimiento (MAE y R 2 ) con diferentes combinaciones de valores de hiperparámetros. En cuanto al bosque aleatorio, la trama MAE refleja la R 2 solar. Al mismo tiempo, el patrón de las parcelas no es tan claro como para el bosque aleatorio. Significa que hay más de un hiperparámetro dominante en XGBoost. [Figura 10](#_bookmark24)  A continuación se muestra el cambio en el rendimiento del modelo (MAE) sobre los valores del hiperparámetro del modelo. El patrón para la trama "n\_estimators" indica que el MAE es más bajo para 100 estimadores, mientras que con el creciente número de árboles, no difiere mucho, lo que significa que no hay signos de sobreajuste cuando el número de estimadores crece. También podemos observar el mismo patrón en la esquina superior izquierda de "n\_estimators" y "max\_depth", lo que significa que la profundidad de 3 tiene valores MAE altos sólo cuando el número de estimadores es igual a 100. Cuando el número de estimadores crece a 200 y por encima, el MAE se vuelve ligeramente inferior para la profundidad máxima igual a 3 que la profundidad máxima por encima de 3. La trama de regularización de datos no muestra ningún patrón en particular, lo que significa que es el hiperparámetro menos importante del modelo.

*Figura 9. Métricas de rendimiento de validación para todas las combinaciones de valores de hiperparámetros del modelo XGBoost en la cuadrícula de validación cruzada*



*Figura 10. XGBoost. Cambio en MAE del modelo para cada valor de hiperparámetro fijo y todos los demás valores de otros hiperparámetros.*



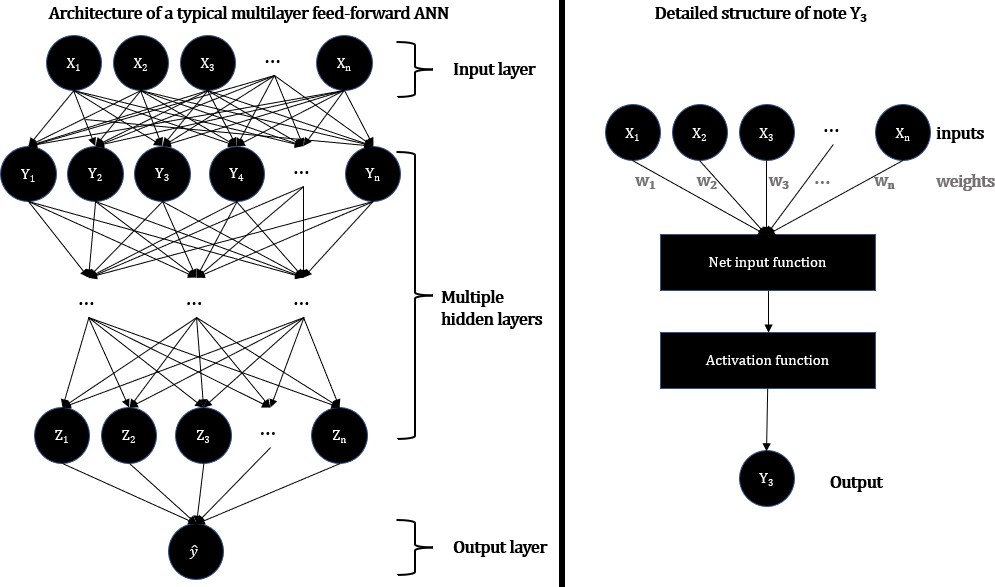
La mejor combinación es de 500 árboles, la regularización de 8, y la profundidad máxima de cada árbol de 3. Redes neuronales artificiales

## Base teórica y aplicación en este proyecto

El modelo final que voy a construir en este proyecto es la regresión basada en una red neuronal artificial (ANN). El término redes neuronales artificiales se refiere a una gran clase de modelos y métodos de aprendizaje. Una red neuronal es un modelo de regresión de varias etapas, normalmente representado por un diagrama de red.

Los ANN están compuestos de capas de nodo, y en particular de una capa de entrada, una o más capas ocultas, y una capa de salida. Cada nodo en capas ocultas y capa de salida se puede considerar como su propio modelo de regresión lineal. Todos los nodos de una ANN están interconectados y tienen pesos y umbrales asociados. Si el valor de salida de un nodo individual está por encima del umbral especificado establecido por una función de activación llamada, este nodo se activa y los datos de esta nota se envían a la siguiente capa de la ANN. De lo contrario, no se pasan datos. Como resultado, la salida de un nodo pasa a ser la entrada del siguiente nodo. Este proceso de pasar datos de una capa a la siguiente capa define esta red neuronal como una red de alimentacion hacia delante.

*Figura 11. Arquitectura de una red neuronal (izquierda) y un nodo (derecha)*



[Figura 11](#_bookmark25)  En la se muestra la arquitectura de una red de comedforward multicapa típica y cómo puede verse un único nodo en esta ANN. Además de alimentar las redes neuronales, existen otros tipos más complejos de ANN, como las redes neuronales convolucionales y las redes neuronales recurrentes. Las redes neuronales convolucionales se aplican más comúnmente para analizar imágenes visuales y se utilizan para el reconocimiento de escritura o escritura no segmentado, conectado. Como el proyecto tenía como objetivo predecir las tasas de abandono, consideré la red de alimentación como la más apropiada para este problema de regresión.

Los pesos de cada nodo en ANN se estiman en el momento del ajuste del modelo a través del algoritmo de retropropagación. El algoritmo de retropropagación calcula el gradiente de la función de pérdida con respecto a los pesos de la red, una capa a la vez. A continuación, se itera hacia atrás desde la última capa, por lo que se llama retropropagación. Es una forma eficiente de ajustar el número masivo de pesos necesarios para una red neuronal. Sin embargo, todavía es un algoritmo complejo, y se necesita tiempo para calcular. Las prestaciones del algoritmo de retropropagación dependen de la elección del optimizador.

Al construir la red neuronal artificial, el modelador necesita hacer una elección con respecto a lo siguiente:

* Arquitectura de red neuronal
* Función de pérdida
* Optimizador

## Arquitectura de red neuronal

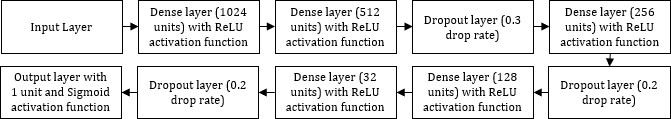
La arquitectura de la red neuronal se basa principalmente en la elección del modelador. En términos generales, es mejor tener demasiadas unidades ocultas que muy pocas. Con muy pocas unidades ocultas, es posible que el modelo no tenga suficiente flexibilidad para capturar las no linealidades en los datos; con demasiadas unidades ocultas, los pesos adicionales se pueden reducir hacia cero si se utiliza la regularización adecuada. Por lo general, el número de unidades ocultas está en algún lugar del rango de 5 a 100, con el número creciente con el número de entradas y el número de casos de entrenamiento [. 8](#_bookmark7)

En este proyecto, evaluo tres redes diferentes:

* Modelo de 5 capas:
* Modelo de 7 capas:



* Modelo de 10 capas:



Todos los modelos consisten en capas Dense y Dropout y difieren entre sí solo por el número de capas. La capa densa es la capa de red neuronal profundamente conectada. Es la capa más común y frecuentemente utilizada. La capa densa hace la operación siguiente en la entrada y devuelve la salida. 17

= activación de salida (punto (entrada, kernel) + sesgo)

*Ecuación 13. Operaciones realizadas en la Capa Dense*

donde,

* *entrada* representar los datos de entrada
* *kernel* representar la matriz de ponderaciones creada por la capa
* *Punto* representar el producto de punto numpy de todas las entradas y sus ponderaciones correspondientes
* *sesgo* representan un valor sesgado utilizado en el aprendizaje automático para optimizar el modelo
* *activación* representan la función de activación de elementos.

Consideré que la capa de Dense sería la más apropiada para el propósito de la regresión. Mientras que otras capas, como la convolución, la puesta en común, y las capas recurrentes, se utilizan en redes neuronales convolucionales y recurrentes, cuyo propósito es diferente del objetivo de este proyecto como se discutió anteriormente.

He utilizado la función de activación lineal rectificada (ReLU) para todas las capas densas ocultas y la activación de SoftMax para la capa densa de salida. La activación de RELU conserva el valor máximo de 0 y el producto de punto de la capa de entrada y las ponderaciones de modelo. La elección de RELU se basa en las razones descritas en los párrafos siguientes.

Otras funciones de activación, como sigmoid y tahn, también se utilizan comúnmente en el aprendizaje profundo. Sin embargo, la función de activación sigmoide transforma un espacio de entrada grande en un espacio de entrada pequeño entre 0 y 1. Por lo tanto, un cambio grande en la entrada de la función sigmoid causará un pequeño cambio en la salida, lo que hace que el derivado de la función sigmoide sea muy pequeño. Como gradientes de redes neuronales se encuentran utilizando la retropropagación multiplicando los derivados de la final a la capa inicial utilizando la regla de la cadena, a continuación, la multiplicación de los derivados de las funciones de sigmoid en redes neuronales de múltiples capas resultará en un un pequeño gradiente general de la red. Así que cada paso de descenso de gradiente hará sólo un pequeño cambio a los pesos, llevando a una convergencia lenta o, en otras palabras, al problema del gradiente de desaparición. ReLU ayuda a superar el problema del gradiente de desaparición, ya que no tiene un pequeño derivado. Tiene gradiente 1 cuando la salida está por encima de cero y 0 de lo contrario. Por lo tanto, estos derivados devolverán 0 o 1 cuando se multipliquen juntos en el proceso de retropropagación. Por lo tanto, la actualización no es nada o toma contribuciones enteramente de los otros pesos y sesgos. Otra ventaja para ReLU, aparte de evitar el problema de desaparición de gradientes, es que tiene un tiempo de ejecución mucho más bajo que cualquier función sigmoide, que utiliza un exponente que es computacionalmente lento. 18

Para la capa de salida, por otro lado, he considerado que sigmoid es la función de activación más adecuada, ya que transforma los valores de la capa de entrada ( *X* ) y leights *β* en un único valor en el rango entre 0 y 1 utilizando la fórmula de [Ecuación 3](#_bookmark10)  o [Ecuación 4 .](#_bookmark11)  Por lo tanto, el valor predicho está en el mismo rango que la variable de destino.

La capa de Dropout establece de forma aleatoria unidades de entrada a 0 con una tasa fija en cada paso durante el entrenamiento. La capa de abandono se utiliza entre las capas Dense para regularizar la red neuronal para evitar su sobreajuste.

## Función de pérdida

El error absoluto medio se ha seleccionado como una función de pérdida en línea con otros modelos. Su elección se describe previamente en el memorándum del proyecto. Además de la función de pérdida de MAE, seleccioné R 2 como medida de rendimiento para otros modelos.

## Optimizador

La consideración crítica en la construcción de la red neuronal es qué optimizador basado en gradiente de gradiente a elegir. Existen diferentes tipos de optimizadores basados en el descenso de gradiente, pero todos tienen el objetivo de encontrar la función mínima global de pérdida utilizando el gradiente de esta función. Sin embargo, estos optimizadores difieren en las formas en que responden a este gradiente. En primer lugar, observaré los tipos más comunes de optimizadores basados en descenso de gradiente y, a continuación, elije el más adecuado para esta tarea:

* Descenso de gradiente por lotes. Este algoritmo calcula el gradiente instantáneo de la función de coste con respecto a los parámetros del modelo para todo el conjunto de datos de entrenamiento. Teniendo en cuenta que este algoritmo utiliza todos los datos de entrenamiento a la vez, es muy caro computacionalmente; por lo tanto, su uso no es práctico. Además, como este algoritmo utiliza todo el conjunto de datos de entrenamiento, no introduce ruido en las estimaciones de los parámetros del modelo, lo que hace más probable que me quede atascado en el mínimo local de la función de pérdida en lugar de su mínimo global.

*Figura 12. Cambio en las métricas de rendimiento de validación durante el ciclo de formación de modelos de regresión basados en ANN de diferentes arquitecturas.*

* Descenso de gradiente estocástico (SGD). El descenso de gradiente estocástico tiene como objetivo superar los inconvenientes del descenso de gradiente por lotes. En lugar de realizar cálculos en todo el conjunto de datos-que es redundante e ineficiente-SGD sólo calcula en un pequeño subconjunto de selección aleatoria

|  |
| --- |
| **Modelo de 5 capas** |
| **Modelo de 7 capas** |
| **Modelo de 10 capas** |

de ejemplos de datos, introduciendo así el ruido en las estimaciones de los parámetros del modelo.

* Estimación del momento adaptativo (Adam). Incluye una tasa de aprendizaje cambiante y un término de impulso. El optimizador mantiene un promedio de funcionamiento de los gradientes recientes y gradientes al cuadrado; efectivamente, recuerda los valores de los gradientes experimentados, con los hiperparámetros

controlar la longitud de su memoria. En comparación con los dos optimizadores de gradiente de gradiente descritos anteriormente, Adam computa no instantáneo sino el gradiente promedio de la historia reciente. Esto permite a Adam introducir el impulso en el descenso de gradiente y permitir que los parámetros cambien rápidamente y cruzar los mínimos locales sin quedar atrapados. Además, la tasa de aprendizaje en el optimizador de Adam se divide por la raíz cuadrada del gradiente cuadrado promedio, lo que asegura que cada paso tenga aproximadamente el mismo tamaño. Si el segundo momento es grande, este término reducirá el tamaño del paso.

El optimizador de Adam adopta mejores estrategias para evitar los mínimos locales. Además, Adán adapta la tasa de aprendizaje a la tasa del descenso. Hay otros optimizadores, como RMSprop y Adadelta, pero la forma en que funcionan es similar al optimizador de Adam. Adam es uno de los optimizadores más utilizados y prácticos para la formación de modelos de aprendizaje profundo por lo que considera que es la mejor opción general.

El desempeño de cada modelo de ANN se presenta en [Figura 12.](#_bookmark26)  Los modelos con 7 capas y 5 capas muestran casi el mismo rendimiento (el MAE más bajo y el mayor R 2 ), mientras que el modelo con 10 capas muestra peor desempeño, y su trama indica sobreajuste. Teniendo en cuenta que el modelo con 5 capas es menos complejo que con 7 capas mientras muestra el mismo rendimiento, lo seleccioné para comparación con otros modelos no-ANN descritos anteriormente.

### Comparación del rendimiento de validación de los modelos

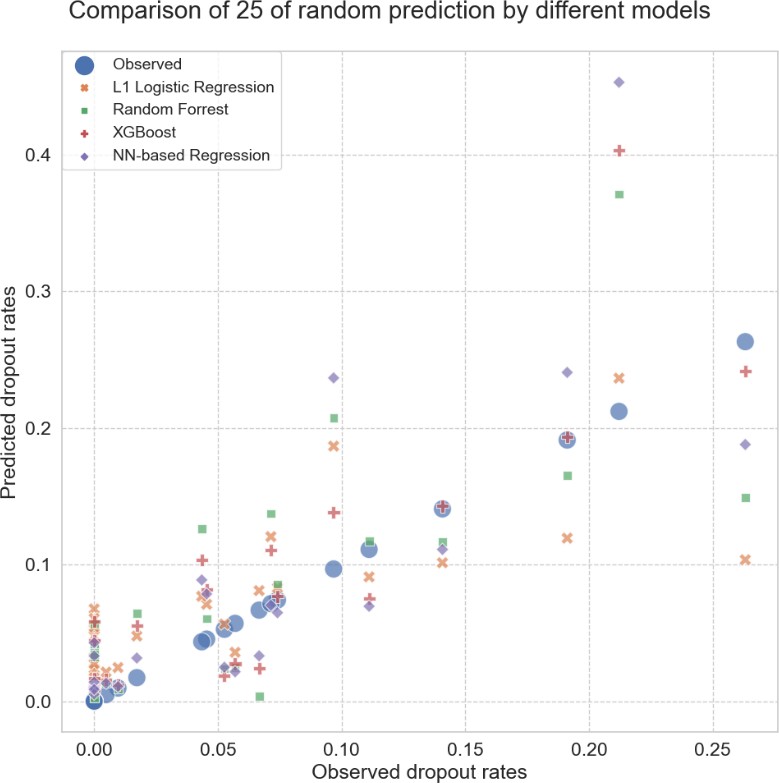
El rendimiento de validación del modelo mejor seleccionado de cada tipo se presenta en [Cuadro 7 .](#_bookmark27)  Las métricas se calculan en todo el conjunto de datos de validación. Pero para estimar la varianza de MAE y R 2 , tomé 5.000 sorteos de 300 observaciones aleatorias del conjunto de datos de validación y calculé 5.000 métricas, y calculé la desviación estándar de MAE y R 2 . Considerando la incertidumbre sobre el verdadero MAE y R 2 valor, la regresión de XGBoost muestra el mejor rendimiento, mientras que la regresión del bosque aleatorio está ligeramente por detrás. En contraste, la regresión basada en la ANN y la regresión logística penalizada son mucho peores.

*Cuadro 7. Métricas de rendimiento de validación de los mejores modelos de cada tipo*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Para todo el conjunto de datos de validación** | | **Bootstrap (5000 empates con 300 observaciones cada uno)** | | | |
| Modelo | Error absoluto medio | R2 | Error absoluto medio (media) | R2 (media) | Error absoluto medio (std) | R2 (std) |
| R1 regresión logística regularizada | 0.042 | 0.50 | 0.042 | 0.50 | 0.002 | 0.06 |
| Regresión forestal aleatoria | 0.033 | 0.67 | 0.033 | 0.67 | 0.002 | 0.05 |
| Regresión de XGBoost | 0.033 | 0.70 | 0.032 | 0.69 | 0.002 | 0.05 |
| Regresión basada en ANN | 0.037 | 0.57 | 0.037 | 0.56 | 0.002 | 0.06 |

[Figura 13](#_bookmark28)  La muestra el gráfico de los valores pronosticados frente a los observados para las 25 observaciones aleatorias del conjunto de validación. En mi, pero no todos, casos, XGBoost (cruz roja) está más cerca de los valores observados (ronda azul) que otros modelos.

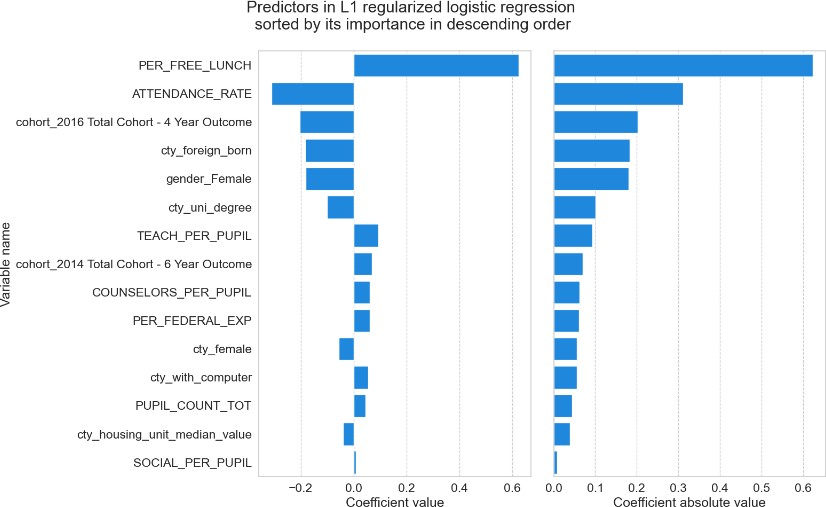
*Figura 13. Comparación de las predicciones de 25 observaciones aleatorias realizadas por modelos seleccionados de cada tipo*



### Importancia de características

La tabla de rendimiento de validación anterior muestra que los modelos basados en árboles de decisiones más complejos y en la red neuronal muestran un mejor rendimiento que una regresión logística penalizada menos compleja. Sin embargo, la regresión logística penalizada tiene una ventaja significativa sobre otros modelos-puede ser fácilmente interpretable. Existe una relación directa entre los predictores y la variable de destino. β coeficiente en la regresión logística se puede interpretar como la cantidad de log odds de un aumento de la tasa de abandono (si β es positivo) o disminución (si β es negativo) cuando el valor del predictor aumenta en 1.

Y como todos los valores del predictor se normalizan, podemos comparar los coeficientes de β directamente entre sí.

*Figura 14. Predictores en la regresión logística regularizada L1 clasificados por importancia en orden descendente*

[Figura 14](#_bookmark29)  La muestra el valor y el valor absoluto del coeficiente β de cada predictor. Podemos ver que PER\_FREE\_LUNCH y

ATTENDANCE\_RATE tiene más impacto que otras variables.

Sin embargo, en modelos más complejos, no existe una relación directa entre los predictores y las variables de destino. Pero existen enfoques para extraer la importancia de la característica para dichos modelos.

**Importancia de la característica-Explicaciones de aditivo Shapley**

## Base teórica

El enfoque utilizado en este proyecto se llama SHapley Explanations Additive (SHAP). SHAP es un enfoque de juego teórico para explicar la salida de cualquier modelo de aprendizaje automático. Conecta la asignación óptima de crédito con las explicaciones locales utilizando los valores de Shapley clásico de la teoría del juego y sus extensiones relacionadas. 19 SHAP es un enfoque de extracción de características agnósticas modelo, por lo que puede explicar la salida de cualquier modelo de aprendizaje automático. Utilizo el paquete shap en Python para calcular los valores de SHAP para los modelos de regresión basados en XGBoost, Random Forest y ANN.

Los valores de Shapley son un enfoque ampliamente utilizado de la teoría del juego cooperativo que vienen con las propiedades deseables. En la teoría del juego del carbón, también conocida como teoría del juego cooperativo, las diferentes combinaciones de jugadores son coaliciones, las diferencias en las puntuaciones son contribuciones marginales de cada jugador, y el valor Shapley es la media de estas contribuciones sobre muchas simulaciones. Para un modelo, las características son los jugadores, diferentes subconjuntos de características son las coaliciones de los jugadores, las diferencias en el error predictivo son las contribuciones marginales. 20

Las matemáticas involucradas en el cálculo de los valores de Shapley para un modelo implican conjuntos y factorials y están fuera del alcance de este proyecto. Sin embargo, todos los detalles algorítmicos descritos en los papeles que adaptaron los valores de Shapley al aprendizaje automático. 21

Hablando de consistencia, los valores de Shapley tienen varias propiedades derivadas de la teoría del juego coalítica que lo hacen ideal como un método de importancia de característica:

* **Maniquí: Si una característica nunca aporta ningún valor marginal, su valor de Shapley es cero**

Shapley i = 0.

* **Sustituibilidad: Si dos características dadas i y j contribuyen por igual a todos sus posibles subconjuntos,**

Shapley i = Shapley j

* **Aditividad: Si un modelo p es un conjunto de submodelos k, las contribuciones de una característica i en los submodelos deben agregarse.**
* **Eficiencia: Asimismo, todos los valores de Shapley deben agregarse como la diferencia entre las predicciones y los valores esperados; o en otras palabras el resultado predicho de una observación es la suma del valor base y los valores de Shapley para todos los predictores. Mientras que el valor base es la media de la variable de destino prevista.**

n

Predicción = Valor base + ∑ Shapley i

i= 1

*Ecuación 14. Relación entre los valores pronosticados y SHAP*

donde n es el número de características.

Como se ha descrito anteriormente, en teoría, los valores de Shapley (también conocidos como valores SHAP con respecto a su aplicación al aprendizaje automático) de una característica representan el promedio de las contribuciones marginales de esta característica al resultado predicho en todos los casos posibles. permutaciones de características en el modelo. En la práctica, sin embargo, el tiempo de cálculo para los valores de Shapley debe invariablemente crecer exponencialmente a medida que las características aumentan de modo que un enfoque de fuerza bruta sería muy intensivo en recursos. Hay varias estrategias para minimizar el cómputo.

El método de permutación se utiliza en este proyecto. Funciona al iterar sobre permutaciones completas de las características hacia adelante y la inversa. Al hacer esto, cambiar una característica a la vez, podemos minimizar el número de evaluaciones de modelo que son necesarias y siempre garantizar que satisfacemos la eficiencia sin importar cuántas ejecuciones del modelo original que elegimos utilizar para la aproximación de la característica valores de atribución. Por lo tanto, los valores de SHAP calculados, mientras se aproximan, hacen exactamente la suma de la diferencia entre el valor base del modelo y la salida del modelo para cada instancia explicada. 22

## Aplicación en este proyecto

Para este proyecto, se calcularon los valores de SHAP para el conjunto de datos de validación, que consta de 1.817 observaciones y 15 predictores. Por consiguiente, la matriz de valores SHAP tendrá una dimensión [1817, 15], ya que los valores se calculan para cada predictor en cada observación. Considerando la fórmula en la propiedad de eficiencia, podemos decir que el valor SHAP de la característica i en observación *j* Muestra hasta qué punto la característica i empuja la predicción en observación *j* lejos de la media de predicción en todas las observaciones.

[Figura 15](#_bookmark30)  La muestra la media de los valores de SHAP absolutos para cada modelo (lado izquierdo de la figura), la distribución de valores SHAP y su impacto en la variable de destino (lado derecho de la figura). La media de los valores de SHAP absolutos se puede interpretar como una puntuación de importancia de característica. Y como podemos ver, las 5 principales características son las mismas en todos los modelos, aunque algunas están en diferentes lugares. Además, las características menos importantes que se encuentran en la parte inferior de cada una de las tramas del lado izquierdo son similares entre los modelos.

El lado derecho de la trama proporciona más detalles sobre cómo afectan las características al valor predicho. Las tramas de violín muestran la distribución de los valores SHAP para cada característica en todas las observaciones. Cuanto más se distribuyen los valores de SHAP a partir del 0, mayor será el impacto. Al mismo tiempo, el color de los valores SHAP en la trama corresponde al valor de la predicción-rojo para valores altos y azul para valores bajos. Por ejemplo, si tomamos los valores SHAP para ATTENDANCE\_RATE que se distribuyen de forma similar en todos los modelos, podemos ver que la mayoría de los valores SHAP están entre -0,05 y 0 son rojos. A la misma larga cola de valores SHAP altos de 0 a 0.2, y estos valores son azules. Esto significa que la alta asistencia (roja) tiende a disminuir las tasas de deserción pronosticadas hasta aproximadamente

0.05. Mientras que en unos pocos casos, las bajas tasas de asistencia (azul) aumentan la deserción en más de 0,2. PER\_FREE\_LUCH, por otro lado, tiene el impacto opuesto en las predicciones de tasas de abandono. Los tres modelos tienden a predecir tasas de deserción más altas para las escuelas con un alto porcentaje de estudiantes elegibles para el almuerzo gratis y viceversa.

Los modelos de aprendizaje de máquinas predictivas como los modelos XGBoost o ANN se vuelven más transparentes cuando se combinan con herramientas de interpretación como SHAP. SHAP identifica las relaciones más informativas entre los predictores del modelo y el resultado, lo cual es útil para explicar lo que está haciendo el modelo. Sin embargo, como ya se mencionó en la sección de Introducción, es importante recordar que la correlación no implica causalidad. SHAP hace transparentes las correlaciones recogidas por modelos ML predictivos. Pero hacer que las correlaciones sean transparentes no las hace causales. La búsqueda de enlaces causales entre predictores y tasas de abandono está fuera del ámbito de este proyecto.

*Figura 15. Resultados de SHAP. Puntuaciones de importancia de características (izquierda) y distribuciones de valores SHAP (derecha) para Random Forest, XGBoost y regresión basada en ANN*

|  |  |
| --- | --- |
| Puntuación de importancia de característica-media absoluta de valores SHAP | Distribución de valores SHAP |
| **Regresión de XGBoost** | |
|  |  |
| **Regresión forestal aleatoria** | |
|  |  |
| **Regresión basada en ANN** | |
|  |  |

Mirando hacia [Figura 14](#_bookmark29)  y [Figura 15 ,](#_bookmark30)  todos los modelos tienen subconjuntos similares de las características más y menos importantes, las cinco principales características son las mismas, aunque están en un orden diferente. Calculé la correlación de Spearman de rango de importancia de característica para evaluar la similitud entre los modelos más precisamente. Los resultados se presentan en [Cuadro 8 .](#_bookmark31)

*Cuadro 8. Matriz de correlación de función*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Modelo** | **Regresión logística regularizada** | **Regresión forestal aleatoria** | **Regresión de XGBoost** | **NN**  **Regresión** |
| **Regresión logística regularizada** | 1.00 | 0.66 | 0.54 | 0.67 |
| **Regresión forestal aleatoria** | 0.66 | 1.00 | 0.92 | 0.77 |
| **Regresión de XGBoost** | 0.54 | 0.92 | 1.00 | 0.83 |
| **Regresión NN** | 0.67 | 0.77 | 0.83 | 1.00 |

La tabla muestra que los rangos de características en todos los modelos tienen una correlación positiva de moderada a fuerte. Los rangos de características en Regresión logística regularizada se correlacionan al menos con otros modelos, ya que todos los coeficientes de correlación están por debajo de 0.7. Hay una correlación más fuerte entre la regresión basada en ANN y ambos modelos basados en árboles-0.77 con Random Forest y 0.83 con XGBoost.

Al mismo tiempo, la correlación de rango de características más fuerte (0.92) se encuentra entre los modelos basados en árboles (XGBoost y Random Forest). Teniendo en cuenta que estos modelos mostraron el mayor rendimiento, analizé más importancia de características de estos modelos.

El paquete scikit-learn proporciona métricas de importancia de características alternativas para XGBoost y Random Forest. Una de estas métricas se basa en la disminución media de la impureza (MDI). En el caso de los árboles de regresión, una disminución de la impureza será equivalente a una disminución de MSE en cada árbol dividido. Sin embargo, este método puede ser muy sesgado y favorecer características de alta cardinalidad (características típicamente numéricas) sobre características de baja cardinalidad tales como características binarias o variables categóricas con un pequeño número de categorías posibles.

Otro método proporcionado por el paquete scikit-learn es la función de función basada en la permutación. Las importaciones de características basadas en permutación no exhiben tal sesgo. Además, la importancia de la característica de permutación puede ser métrica de rendimiento calculado en las predicciones del modelo y se puede utilizar para analizar cualquier clase de modelo (no sólo los modelos basados en árboles). 23

Aunque la importancia de la característica basada en la permutación se considera menos sesgada, comparé la correlación de Pearson entre las puntuaciones de rendimiento de las características derivadas del paquete SHAP y las puntuaciones basadas en permutación y MDI del paquete scikit-learn. Los resultados de las 5 principales características SHAP más importantes de XGBoost se presentan en [Cuadro 9 .](#_bookmark32)

*Cuadro 9. Correlación de Pearson de 5 características principales del modelo XGBoost (puntuaciones basadas en SHAP) con las mismas características de XGBoost y Random Forest derivadas por SHAP y MDI-y Permutation-based scores from scikit-learn package.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Modelo** | Importancia de característica de RF (MDI) | Importancia de la característica de RF (permutación) | Importancia de característica de RF (SHAP) | Importancia de característica de XGB (MDI) | Importancia de la característica de XGB (permutación) | Importancia de característica de XGB (SHAP) |
| Importancia de característica de RF (MDI) | 1.00 | 0.97 | 0.98 | 0.71 | 1.00 | 0.99 |
| Importancia de característica de RF  (permutación) | 0.97 | 1.00 | 0.99 | 0.85 | 0.99 | 0.93 |
| Importancia de característica de RF (SHAP) | 0.98 | 0.99 | 1.00 | 0.82 | 0.99 | 0.96 |
| Importancia de característica de XGB (MDI) | 0.71 | 0.85 | 0.82 | 1.00 | 0.75 | 0.62 |
| XGB de importancia de característica  (permutación) | 1.00 | 0.99 | 0.99 | 0.75 | 1.00 | 0.98 |
| Importancia de característica de XGB (SHAP) | 0.99 | 0.93 | 0.96 | 0.62 | 0.98 | 1.00 |

Como se esperaba, existe una fuerte correlación entre las puntuaciones basadas en permutación del cikit y (entre 0,93 y 1). Las puntuaciones basadas en XGBoost MDI tienen una correlación mucho más débil con otras

métodos básicos de importancia de características (entre 0,62 y 0,85). Se esperan diferentes resultados proporcionados por el método de puntuación de importancia de la característica basada en MDI debido a las razones explicadas anteriormente.

### Conjunto de modelos

Antes de decidir sobre el modelo final, también es útil comprobar si el conjunto de los modelos mostrará un mejor resultado que un único modelo. En el aprendizaje automático, el promedio de conjuntos está creando múltiples modelos y combinándolos para producir la salida deseada, en lugar de crear un solo modelo. Con frecuencia un conjunto de modelos funciona mejor que cualquier modelo individual, porque los diversos errores de los modelos "average out". 24

Este proyecto utiliza un conjunto medio ponderado de modelos, que es una combinación de las predicciones de los modelos. El conjunto medio ponderado asume que algunos modelos son mejores que otros y se les debe dar más peso en la predicción general. La contribución de cada modelo se pondera proporcionalmente a su capacidad o habilidad.

Una predicción promedio ponderada involucra primero asignar un coeficiente de peso fijo a cada miembro del ensamble. Puede ser un valor de coma flotante entre 0 y 1, que representa un porcentaje del peso. También podría ser un entero a partir de 1, lo que representa el número de votos para dar a cada modelo. 25

*Cuadro 10. Predicción media ponderada de conjuntos de modelos (basado en el conjunto de validación)*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Combinación** | **XGB\_Weight** | **RF\_Weight** | **NN\_Weight** | **GLM\_Weight** | **MAE** | **R2** |
| **XGB + RF + NN + GLM** | 0.65 | 0.25 | 0.05 | 0.05 | 0.0321 | 0.705 |
| **XGB + RF + GLM** | 0.65 | 0.30 | - | 0.05 | 0.0321 | 0.706 |
| **XGB + RF + NN** | 0.65 | 0.30 | 0.05 | - | 0.0319 | 0.706 |
| **XGB + RF** | 0.65 | 0.35 | - | - | 0.0319 | 0.707 |
| **XGB + GLM** | 0.90 | - | - | 0.10 | 0.0327 | 0.701 |
| **XGB + NN + GLM** | 0.90 | - | 0.05 | 0.05 | 0.0325 | 0.701 |
| **XGB + NN** | 0.90 | - | 0.10 | - | 0.0324 | 0.700 |
| **XGB** | 1.00 | - | - | - | 0.0325 | 0.699 |

Teniendo en cuenta que el modelo XGBoost mostró el mejor resultado, calculé la predicción media ponderada con este modelo y todos los demás modelos. El cálculo se realizó para todos los pesos posibles con un paso de 0,05 para que la suma de todos los pesos sea igual a 1. La siguiente tabla muestra las mejores predicciones de diferentes combinaciones de modelos. Considerando que la desviación estándar del MAE es de aproximadamente 0,002 y de R 2 es aproximadamente 0,05 (ver [Cuadro 7](#_bookmark27) ), las ganancias de rendimiento de la mejor combinación (XGB + RF) son una disminución de 0,0006 en MAE y un aumento de 0,008 en R 2 , que es inferior a 1/3 de la desviación estándar de ambas métricas de rendimiento. Consideré que las ganancias de rendimiento no son lo suficientemente sustanciales como para considerar un modelo más complejo y, por lo tanto, seleccioné el modelo de regresión XGBoost como una elección final.

### Estimación del rendimiento del modelo final en el conjunto de datos de prueba

El rendimiento del modelo de regresión de XGBoost se presenta en [Cuadro 11 .](#_bookmark34)  Tanto error absoluto medio como R 2 se ha descartado en comparación con las métricas de validación; sin embargo, la caída está dentro de una desviación estándar, y lo considero aceptable.

*Cuadro 11. Rendimiento de XGBoost en el conjunto de pruebas*

|  |  |  |
| --- | --- | --- |
| **Modelo** | **Error absoluto medio** | **R2** |
| Regresión de XGBoost | 0.035 | 0.66 |

# Conclusión y debate

En este proyecto, aggregé la información sobre la deserción escolar pública en el estado de Nueva York junto con otros datos sociales y económicos en escuelas públicas y condados del estado.

El objetivo del proyecto era construir 4 modelos diferentes, la regresión logística regularizada, el bosque aleatorio, el impulso de gradiente, y los modelos de red neuronal artificial que predicen las tasas de abandono escolar y comparar el rendimiento de estos modelos. Los modelos basados en árboles y las redes neuronales artificiales construidas en este proyecto son comúnmente conocidos como modelos "black-box" que hacen predicciones sin revelar ninguna información sobre su funcionamiento interno. Por lo tanto, otro objetivo de este proyecto era utilizar herramientas agnósticas modelo que harían que estos modelos fueran interpretables para entender lo que está detrás de la "caja negra". SHAP, o ExPlanaciones de Shapley Additive, fue empleado para este propósito en el proyecto. Después de interpretar los modelos, los comparé no solo en cuanto a su rendimiento sino también en términos de qué variables en cada modelo impactan más la predicción.

Como se presenta en [Cuadro 7](#_bookmark27)  Los métodos de los conjuntos de árboles, Random Forest y XGBoost superaron el modelo basado en ANN y regularizaron la regresión logística, mientras que XGBoost fue marginalmente mejor que el Random Forest.

Mientras que los modelos de conjunto de árboles proporcionan una menor interpretación que los modelos de regresión logística, pueden capturar relaciones no lineales y más complejas, por lo que inicialmente esperaba que superaran la regresión logística.

Sin embargo, no esperaba que Random Forest y XGBoost mostraran mejor medio de error absoluto y R2 que el modelo de ANN. Puede haber varias razones para ello:

* En general, las redes profundas necesitan muchos datos para batir más algoritmos tradicionales de aprendizaje automático, por lo que el conjunto de datos de este proyecto puede no tener suficientes características que el modelo de ANN podría haber utilizado para batir los modelos de conjunto.
* Las redes neuronales profundas son más útiles para los problemas para los que las características no llevan casi ninguna información individualmente. Estos modelos se utilizan más a menudo para los problemas con datos no estructurados (imágenes, texto, grabaciones de audio), como la visión de la computadora y el procesamiento del lenguaje natural. Al mismo tiempo, el problema de este proyecto es un problema de regresión utilizando un conjunto de datos de entrada estructurado (tabular). Los modelos de conjunto son modelos mucho más sencillos diseñados para trabajar con datos estructurados.

Mirando la contribución de las variables a las predicciones finales, me enteré de que todos los modelos tenían las mismas 2 características más importantes, "ATTENANCE\_RATE" y "PER\_FREE\_LUNCH", y esa importancia absoluta marca estas características (valor de coeficiente β para La regresión logística regularizada y la media de los valores SHAP para otros modelos) fueron mucho más grandes que otras características (ver [Figura 14](#_bookmark29)  y [Figura 15](#_bookmark30) ), lo que significa que estas dos características contribuyen más a la predicción de la tasa de abandono. Pero debo subrayar que las puntuaciones de alta importancia de la característica no deben confundirse con indicar un vínculo causal fuerte entre los predictores y el resultado debido a las razones ya mencionadas en la introducción de este proyecto.

La correlación de rango más fuerte fue entre XGBoost y Random Forest, por lo que comparé la correlación de Pearson de 5 variables más importantes derivadas por diferentes enfoques de la evaluación de la importancia de la característica (SHAP y scikit-aprender MDI-and Permutation-based Puntuación de importancia). El

correlación fue casi perfecta (por encima de 0,9) para la mayoría de los enfoques, excepto para las puntuaciones basadas en MDI, como se presenta en [Cuadro 9 .](#_bookmark32)

Como paso final, evalué MAE y R2 de las predicciones medias ponderadas de XGBoost con otros modelos (ver [Cuadro 10](#_bookmark33) ). Sin embargo, el aumento en el rendimiento estuvo muy por debajo de la desviación estándar de las métricas de rendimiento, por lo que consideré mantener un modelo menos complejo que consiste solo en XGBoost.

El problema que afronté al hacer este proyecto es que algunos paquetes estándar de R y Python pueden entender que la variable de destino proporcional se extrae de la distribución binomial, y estos paquetes pueden aplicar la pérdida cruzada de entropía, que se utiliza para problemas de clasificación, para entrenar el modelo para predecir las tasas de deserción. Un ejemplo de tales paquetes es glmnet utilizado para entrenar la regresión logística regularizada descrita más adelante en este informe. Glmnet utiliza familias de distribución del paquete stats, en el que, para la familia binomial, la respuesta se puede especificar de una de las tres maneras siguientes: 26 :

* Como factor: el "éxito" se interpreta como el factor que no tiene el primer nivel (y por lo tanto generalmente de tener el segundo nivel).
* Como vector numérico con valores entre 0 y 1, interpretado como la proporción de casos exitosos (con el número total de casos dados por los pesos).
* Como matriz de enteros de dos columnas: la primera columna proporciona el número de aciertos y el segundo el número de anomalías.

**El paquete xgboost en Python también permite utilizar la entropía cruzada como una función objetivo para la variable de destino proporcional.**

Sin embargo, otros paquetes estándar como Keras de TensorFlow que se utilizó para entrenar modelos basados en redes neuronales artificiales no permiten usar la pérdida cruzada de entropía con una respuesta numérica. Por lo tanto, construí estos modelos de aprendizaje automático supervisados utilizando funciones de pérdida para problemas de regresión. La pérdida cruzada de entropía (igual a la probabilidad de registro negativa) se describe en la sección "Regresión logística penalizada" de este informe, y la razón de la selección Se describe el error absoluto como una pérdida para los problemas de regresión en " Definición de la función de pérdida y sección de métricas de rendimiento ".

Debido al tiempo limitado, sólo me centré en la regresión logística regularizada/penalizada, el bosque aleatorio, el XGBoost y los modelos de regresión basados en ANN en este proyecto. Sin embargo, hay muchas otras formas de modelar los datos proporcionales. Por ejemplo, para el tipo de datos proporcionales, donde los valores están entre 0 y 1 pero no son exactamente binomiales, se puede utilizar un modelo de regresión logística o beta sobre-disperso, permitiendo estructuras de varianza más flexibles [10 .](#_bookmark9)

Como comentario final, me gustaría enfatizar que, como ya mencioné en la sección de introducción, el Centro Nacional de Datos de Estadísticas Educativas muestra que en los Estados Unidos, la tasa de abandono nacional disminuyó de 8.3 por ciento en 2010 a 5.1. por ciento en 2019. Los modelos construidos en este proyecto se basan en datos estáticos de 2020 y pueden no mostrar la misma precisión cuando se aplican a datos futuros.

Apéndice A. Matriz de correlación antes de la selección de variables

