Maestría en Explotación de Datos y Descubrimiento de Conocimiento
Aprendizaje Automático

1er cuatrimestre 2021

Trabajo Práctico nro 1

Integrantes:

López Malizia, Álvaro
Padula, Eliana Inés
Rossi, Fabiana Alejandra

Universidad de Buenos Aires

Facultad de Ciencias Exactas y Naturales
Facultad de Ingeniería

May 30, 2021

Resumen

El presente documento pretende explorar en profundidad el algoritmo de árboles de decisión utilizado en aprendizaje automático. A tal fin, se trabajó con una muestra de pacientes con atributos que pudieran estar relacionados con la ocurrencia de un accidente cerebrovascular (ACV). Se realizó un pre-procesamiento de la muestra recibida analizando sus datos faltantes, la relación de los atributos con la variable objetivo y se definió la importancia de los mismos. Se separó la base de datos en dos subconjuntos, uno para desarrollo y otro para testear el modelo con el fin de predecir un ACV (80%-20%, respectivamente). En la etapa de desarrollo se probaron diferentes algoritmos para definir la mejor poda y evaluar la *performance* de cada modelo. Este análisis se realizó con el conjunto de datos con las variables numéricas continuas agrupadas de forma dicotómica, y sin agrupar. Una vez validada la *performance* en el conjunto de entrenamiento, se evaluó el mejor modelo generado en el subconjunto de testeo. Se testeó la capacidad de predecir un ACV a partir de los datos. El código de cálculo del algoritmo y gráficos se encuentran en: https://colab.research.google.com/drive/1XE_Si8Nh1tvQKC7SYtCG3c2jOpLBLL-N.

Introducción

Los accidentes cerebrovasculares son una de las causas principales de discapacidad en adultos y ancianos, que puede resultar en numerosas dificultades socio-económicas, e incluso derivar en la muerte en ausencia de tratamiento [1]. En este escenario, predecir qué paciente será más propenso a desarrollar una lesión resulta de particular interés clínico [2]. En este sentido, se han aplicado numerosas técnicas de *machine learning* para abordar esta enfermedad [1-3]. El objetivo del presente trabajo práctico fue analizar el uso de algoritmos para la generación de árboles de decisión. Con este fin se utilizó una muestra de una población de individuos con diferentes atributos que podrían ser representativos e indicativos para predecir un ACV.

Datos

La base de datos presenta 10 atributos que pueden resultar significativos para predecir un ACV. En la siguiente tabla se resumen las características de los mismos. En anaranjado se indica el tratamiento de las variables numéricas continuas, cuando las mismas fueron agrupadas de manera dicotómica.

variable	tipo de dato		tratamiento VARIABLES CONTINUAS	tratamiento VARIABLES DICOTOMIZADAS	niveles (cantidad)	código
id	*int64*	*numérico discreto*	*drop*	*drop*	*no es informativa*	
gender	object	categórico nominal	drop 'Other' (1) label encoder	drop 'Other' (1) label encoder	Female(2994) Male(2115)	0 1
age	float64	numérico contínuo	sin encoding cuando son contínuas	gini split label encoder	< 68 años (4253) >= 68 años (856)	0 1
hypertension	int64	numérico discreto	label encoder	label encoder	0(4611) 1(498)	0 1
heart_disease	int64	numérico discreto	label encoder	label encoder	0(4833) 1(276)	0 1
ever_married	object	categórico nominal	label encoder	label encoder	No(1756) Yes(3353)	0 1
work_type	object	categórico nominal	dummies (cat no ordinal no binaria)	dummies (cat no ordinal no binaria)	Private(2924) Self-employed(819) children(687) Govt_job(657) Never_worked(22)	no (0) yes (1)
Residence_type	object	categórico nominal	label encoder	label encoder	Rural(2513) Urban(2596)	0 1
avg_glucose_level	float64	numérico contínuo	sin encoding cuando son contínuas	gini split label encoder	>= 162.14 (659) < 162.14(4450)	0 1
bmi	float64	numérico contínuo	imputación c media de dev (agrupado por sexo), sin encoding	imputación c media de dev (agrupado por sexo) gini split + label	< 26.1 >= 26.1	0 1
smoking_status	object	categórico nominal	dummies (cat no ordinal no binaria)	dummies (cat no ordinal no binaria)	never smoked(1892) Unknown(1544) formerly smoked(884) smokes(789)	no (0) yes (1)
stroke	int64	numérico discreto	label encoder	label encoder	0(4861) 1(249)	0 1

Metodología

Datos

Se generó un modelo en base a variables numéricas que favorecen la simplicidad del mismo. A partir de una base de datos con características de pacientes que sufrieron o no un ACV, se realizó, en primera instancia, una tarea de exploración de las variables. Se determinó la presencia de datos faltantes y la correlación entre los diferentes atributos.

Modelo de datos Se realizó el trabajo práctico mediante dos metodologías de preprocesamiento y análisis. Por un lado, se dicotomizaron las variables numéricas continuas (se agruparon en 2 grupos) y se construyeron árboles a partir de esta simplificación. Por otro lado, se utilizaron los atributos numéricos continuos para las clasificaciones.

Preprocesamiento Se eliminó la columna correspondiente al atributo *id* por no resultar informativa para nuestro análisis, y las observaciones completas correspondientes al único individuo cuyo género presentaba un valor de 'other'.

Adecuación de valores continuos Los atributos continuos fueron discretizados realizando una partición a partir de un valor umbral que maximizara la reducción de impureza con el criterio de gini.

Valores faltantes De acuerdo al análisis de los valores faltantes, se encontraron 3.93% de datos nulos en la variable *bmi*. Se decidió imputar los mismos con la media de *bmi* del conjunto de desarrollo, agrupado por sexo. Este porcentaje no resulta sustancial para alterar la distribución de la variable.

Valores categóricos: Para los valores categóricos no ordinales, se transformaron en variables *dummies*, y cada una de estas se incorporaron como atributos al dataset. En el caso de las variables binarias, se les asignó uno o cero mediante *label encoder*.

Separación de datos: El conjunto de datos fue separado en Desarrollo y Held Out (o *test*). Dado que se observa que en nuestro conjunto de datos la variable *target* se encuentra desbalanceada, se realizó una estrategia de separación que considera este desbalance y distribuye en forma proporcional los datos *target* entre ambos conjuntos. A su vez, el conjunto de datos correspondiente a la categoría de desarrollo fue subdividida en los conjuntos de *train* y *validation*.

Algoritmos

Se definieron métricas de *performance*, se evaluaron los hiperparámetros, se simplifico el modelo y se definieron atributos más importantes con el objetivo de que el modelo prediga de manera eficiente y simple un ACV:

Elección de métricas y *performance* En cuanto a la selección de métricas utilizadas, se definió darle mayor importancia a *recall* que a la *precision*. Dado el objetivo de predecir un ACV, hayamos más importante detectar todos los casos positivos, aún existiendo falsos positivos. Consideramos importante la prevención de ACV. Con esta misma lógica, consideramos un beta = 2 en el F-beta score, otorgándole mayor peso a *recall*.

Selección *a priori* de los hiperparámetros Los hiperparámetros seleccionados *a priori* para desarrollar el modelo de árboles presentado fueron los siguientes: Impureza de gini, 5 valores mínimos en cada hoja, 15 niveles de profundidad como máximo y se le exigió al modelo que equilibre los pesos asociados a las clases de las variables.

Evaluación de los hiperparámetros definidos Se evaluó en el conjunto de desarrollo el modelo con los hiperparámetros definidos *a priori* a través de 50 semillas diferentes y a través de realizar una validación cruzada (*50k-fold cross validation*)

Poda del árbol A través del algoritmo de máxima complejidad de costos provisto por la librería de sklearn, se extrajo un rango de valores del hiperpárametro de alfa para evaluar un árbol con menor profundidad. Se evaluó cada alfa provisto mediante la técnica de *10k-fold*. Se seleccionó aquel alfa cuyo árbol predijo mejor el ACV con menor profundidad, bajo la *performance* de F-beta con el fin de no sobreajustar el modelo.

Selección de atributos Se utilizó la técnica de eliminación recursiva para determinar qué variables eran de mayor importancia para la construcción del modelo. A partir de los tres principales atributos, se construyó un nuevo árbol de decisión.

Resultados

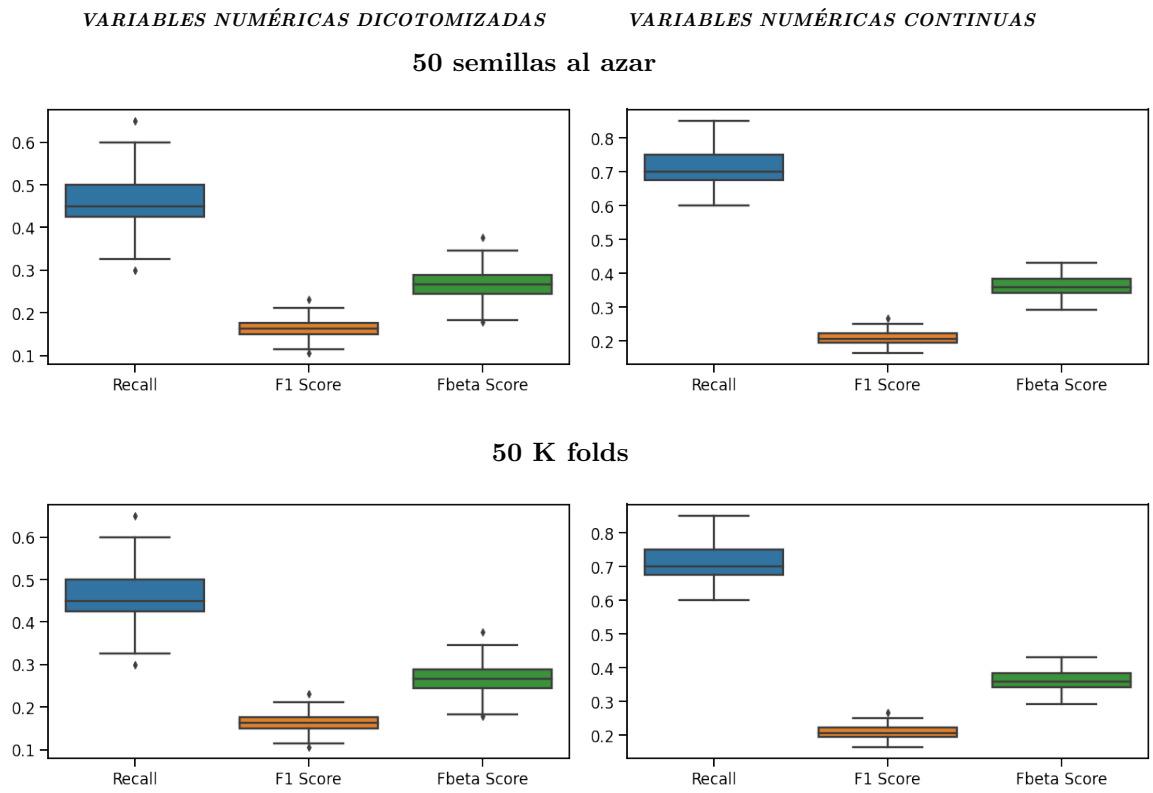
Correlaciones con *stroke* Para evaluar las variables en su relación con *stroke* se realizaron tablas de contingencia representadas en mapas de color, dado que la base de datos presentaba variables categóricas y numéricas. Se representan las relaciones con la variable objetivo según la cantidad de observaciones en cada clase.

Indicadores de mayor susceptibilidad Según el criterio de gini, las variables que aportarían mayor información

son: *age*, *avg glucose level*, *heart disease*, *hypertension*.

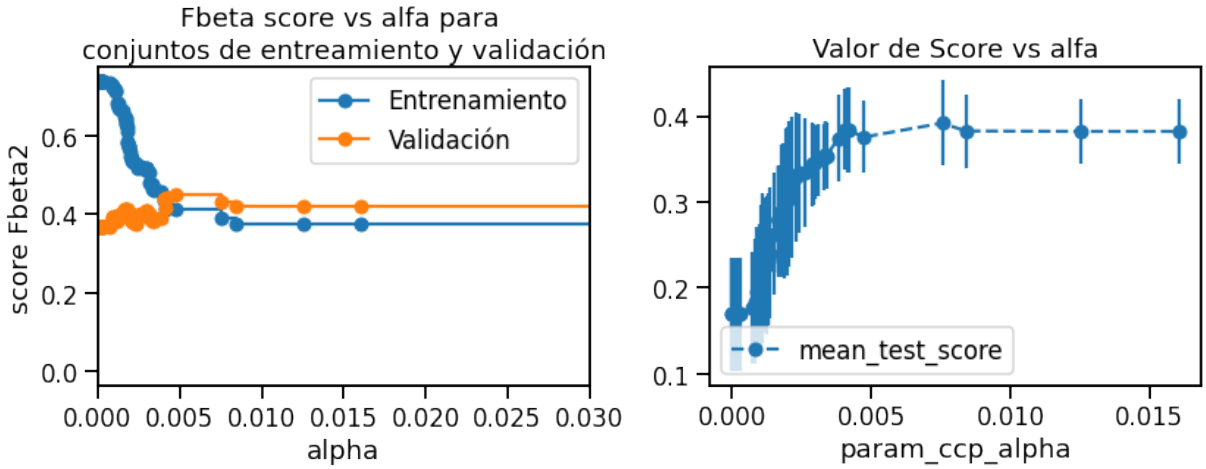
Balance de variable objetivo La variable objetivo (*stroke* o ACV), se encuentra desbalanceada, dado que sólo un 4.87% de los datos resultaron positivos. En consecuencia, se utilizó el parámetro *stratify*, al crear los subconjuntos de Held Out y desarrollo, de forma de garantizar la proporción de datos positivos en cada caso. Por otro lado, se utilizó el parámetro *class weight* para compensar el desbalance en la implementación de los modelos.

Validación del modelo con hiperparámetros *a priori*



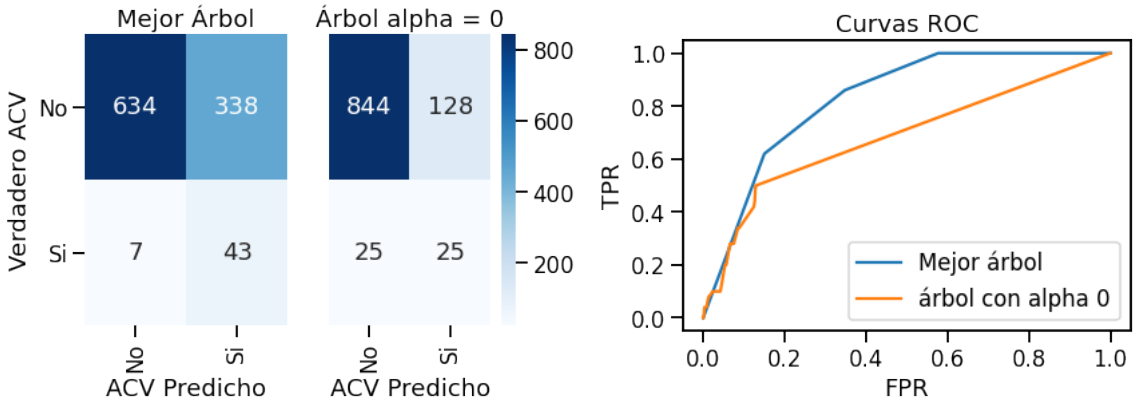
Se evaluaron los hiperparámetros definidos *a priori* con los datos con variables "dicotomizadas" o continuas. Nuestros resultados indicaron que las métricas de *performance* en los modelos de *50k fold* y 50 semillas para las variables en su forma binaria eran menores que aquellas de los modelos que utilizaron las variables continuas

Poda del árbol Los resultados de re-entrenar un árbol iterando sobre distintos valores de alfas de mínima complejidad utilizando *10k folds* muestran que la *performance* se incrementa en el subconjunto de validación conforme aumenta el alfa.



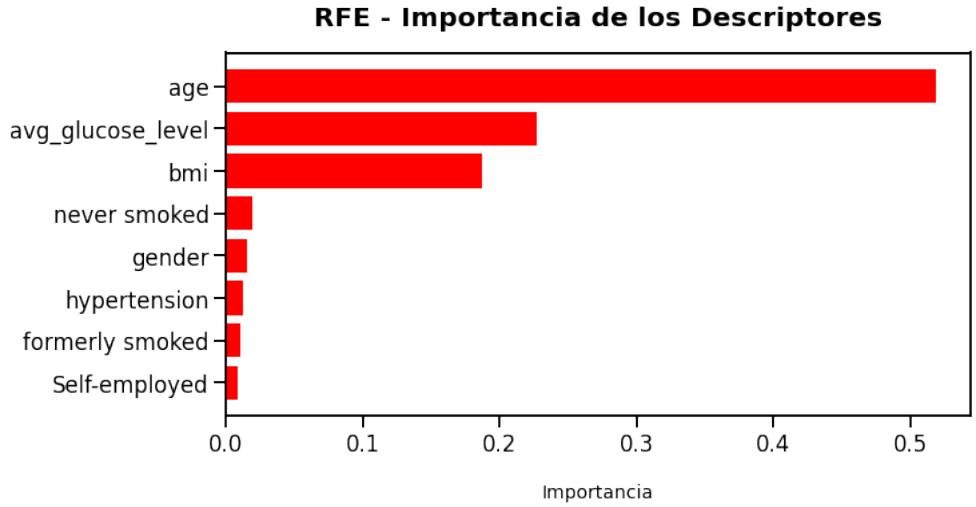
En este caso, establecer alfa = 0.0075 maximiza la medida de Fbeta de la prueba.

Evaluación en conjunto de test - Árbol podado *versus* sin poda Se evaluó en el set de datos de testeo el árbol podado y sin podar. Los resultados indicaron que la poda mejoraba la *performance* del modelo en el *scoring* utilizado. Se obtuvieron valores de *score* de 0,37 y 0,35 para el árbol con y sin poda, respectivamente.



En cuanto a la matriz de confusión, la misma indicó que el modelo podado mejora la cantidad de casos positivos tal cual se ha definido el objetivo de *performance*. En otras palabras, la curva de ROC presenta un ratio de casos positivos mayor.

Selección de atributos a través de eliminación recursiva Para el árbol sin poda, se utilizó la técnica de eliminación recursiva que indicó que sólo 7 de las 14 variables analizadas eran significativas para modelar.



Se encontró que la edad, el promedio de nivel de glucosa en sangre y *bmi* eran las variables más importantes respecto de las seleccionadas. Se re-entrenó el árbol de decisión utilizando dichas variables, con un alfa=0 e iterando sobre distintos valores de profundidad. Se obtuvo un valor de F-beta2 mayor, de 0,41.

Conclusión

La comparación entre los tipos de procesamiento (variables continuas o "continuas dicotomizadas"), demostró que transformar variables numéricas continuas en binarias produce una reducción en la capacidad predictiva de los árboles generados. Este resultado es atribuible a la pérdida de información como consecuencia del agrupamiento de los valores. La selección de las variables más representativas, así como la poda del árbol, impactan en la calidad del modelo a desarrollar. La predicción de tener un ACV mejora cuando sólo se utilizan las variables más representativas, y se ejecuta un árbol de menor profundidad. Estas consideraciones seguramente permitan diseñar modelos más flexibles con mayor capacidad para realizar conjeturas. Se concluye que es importante validar nuestros modelos en los hiperparametros seleccionados y analizar qué variables son representativas del análisis. Finalmente, dado que la variable objetivo se encontraba desbalanceada (con pocos casos positivos), a pesar de las metodologías aplicadas, el modelo sólo alcanzara un Fbeta *score* de 0,41.

Bibliografía

[1] Yu, J.; Park, S.; Kwon, S.-H.; Ho, C.M.B.; Pyo, C.-S.; Lee, H. AI-Based Stroke Disease Prediction System Using Real-Time Electromyography Signals. Appl. Sci. 2020, 10, 6791. https://doi.org/10.3390/app10196791

[2] Truelsen T, Beggs S, Mathers CD (2006) The global burden of cerebrovascular disease. Geneva, Switzerland: WHO. https://www.who.int/healthinfo/statistics/bod_cerebrovasculardiseasestroke.pdf

[3] Amini L, Azarpazhouh R, Farzadfar MT, et al. Prediction and control of stroke by data mining. Int J Prev Med. 2013;4(Suppl 2):S245-S249.–

[4] Miroslav Kubat,An Introduction to Machine Learning. Second Edition 2007. Capítulo 6

[5] StatQuest with Josh Starmer,Decision Trees in Python from Start to Finish https://www.youtube.com/watch?v=q90UDEgYqeI&t=270s