Introducción.

El objetivo de este reporte es desarrollar y comparar 2 algoritmos de modelos supervisados de clasificación: los árboles de decisiones y los K vecinos más cercanos (KNN por sus siglas en inglés), para predecir de mejor forma el riesgo crediticio según los datos aportados desde un banco alemán. Se trata de una matriz de datos con 17 variables y 1.000 registros donde la evaluación de riesgo crediticio (credit) se califica como **0** si es mala o **1** si es buena.

Para evaluar el rendimiento de los algoritmos generados se utilizarán 2 medidas derivadas desde la matriz de complejidad, considerando como *positivo* al valor de **credit** igual a 1:

- *La exactitud*, pues me interesa que mi modelo entregue la mayor probabilidad de realizar predicciones correctas, tanto para valores positivos como negativos.
- *La especificidad*, pues considero que la capacidad de identificar efectivamente a la mayor cantidad de personas con mala evaluación de riesgo crediticio (caso negativo) es más crítico para un sistema bancario, pues evitaría pérdidas que pueden llegar a ser considerables en la medida que se utilice la predicción para captar más casos.

Creación de los conjuntos de datos de entrenamiento y validación.

Un problema que debe tenerse siempre en cuenta al crear un modelo de análisis de datos es el *overfitting*: que el modelo se ajuste demasiado bien a los datos de entrenamiento, pero que no entregue un resultado adecuado al alimentarlo con otro conjunto de datos. Es por eso que una parte importante en el desarrollo de un modelo es tener unos datos para validarlo.

En este sentido, dado que sólo contamos con 1.000 registros, la mejor estrategia parece ser dividirlos en 2 grupos de trabajo seleccionados de forma aleatoria: uno más grande para entrenar al modelo, con 75% de los datos, y otro más pequeño para validar sus resultados, con el 25% restante. Los valores de las proporciones se eligieron arbitrariamente.

Para evaluar nuestras muestras generadas aleatoriamente podemos ir a mirar la variable de interés, **credit**: no se observan diferencias significativas, tanto en la probabilidad de que tenga valor **0** (todos cercanos al 30%), ni en las medidas de heterogeneidad por *Entropía de la información* (todas cercanos a 0,88) ni por el *Índice de impuridad de Gini* (todas cercanos a 0,42) (se hará comentario de estas 2 medidas más adelante en el texto).

Datos	Número_elementos	Probabilidad_credit_0	Heterogeneidad_por_entropía	Heterogeneidad_por_Gini
Datos originales	1000	0.300	0.8812909	0.420000
Datos de entrenamiento	750	0.296	0.8763462	0.416768
Datos de validación	250	0.312	0.8954686	0.429312

Árboles de decisión.

Los árboles de decisión son modelos supervisados que pueden ser usados para variables tanto categóricas, para clasificar, como numéricas continuas, para obtener una regresión. Tienen las ventajas de que los datos no requieren de tratamiento previo, además de que la comprensión del modelo y el cómo se genera el resultado es fácil, especialmente para las personas no especialistas, pues se explicita en los nodos los valores con los cuales se hace la división de los datos. En el siguiente ejercicio se generarán múltiples árboles de clasificación, manejando múltiples variables, para lograr predecir si una persona es de bajo o alto riesgo crediticio (la variable de interés o variable endógena) con la ayuda de 16 variables explicativas (o variables exógenas).

Para todos los casos se entrenará los modelos manipulando sólo el parámetro en discusión, dejando el resto con los valores por defecto. En todas las imágenes de los árboles se pueden ver los nodos como rectángulos ovalados con la siguiente información:

- El número arriba indica la clase ajustada, esto es, el valor más probable para ese nodo de la variable endógena (0 o 1).
- El número abajo indica la probabilidad de la clase ajustada para ese nodo (entre 0 y 1).
- La intensidad del color indica que tan homogénea es la muestra del nodo: mientras más intenso el verde, más homogénea con predominancia de **credit** como 1; mientras más intenso el azul, más homogénea con predominancia de **credit** como 0.

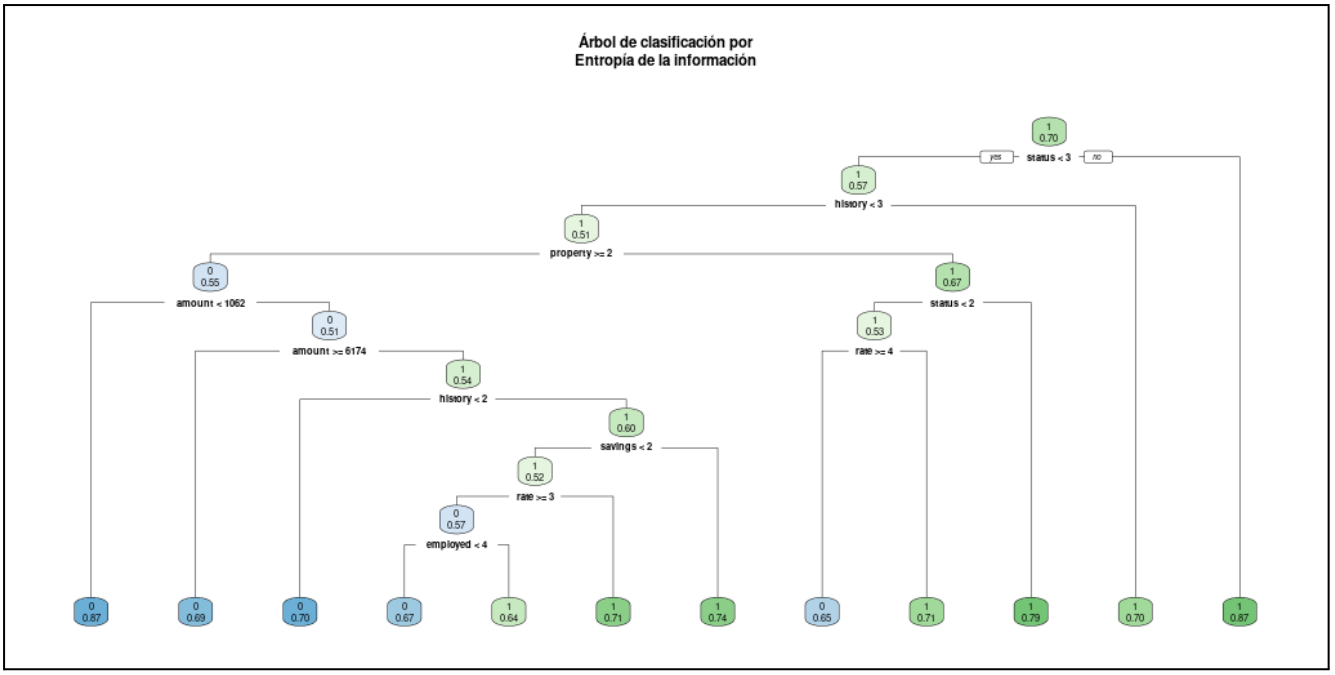
Debajo de los nodos se encuentran las respectivas variables exógenas con el valor calculado para hacer la división: el grupo de la rama que nace hacia la izquierda sí cumple la condición, el grupo de la rama que nace hacia la derecha no la cumple.

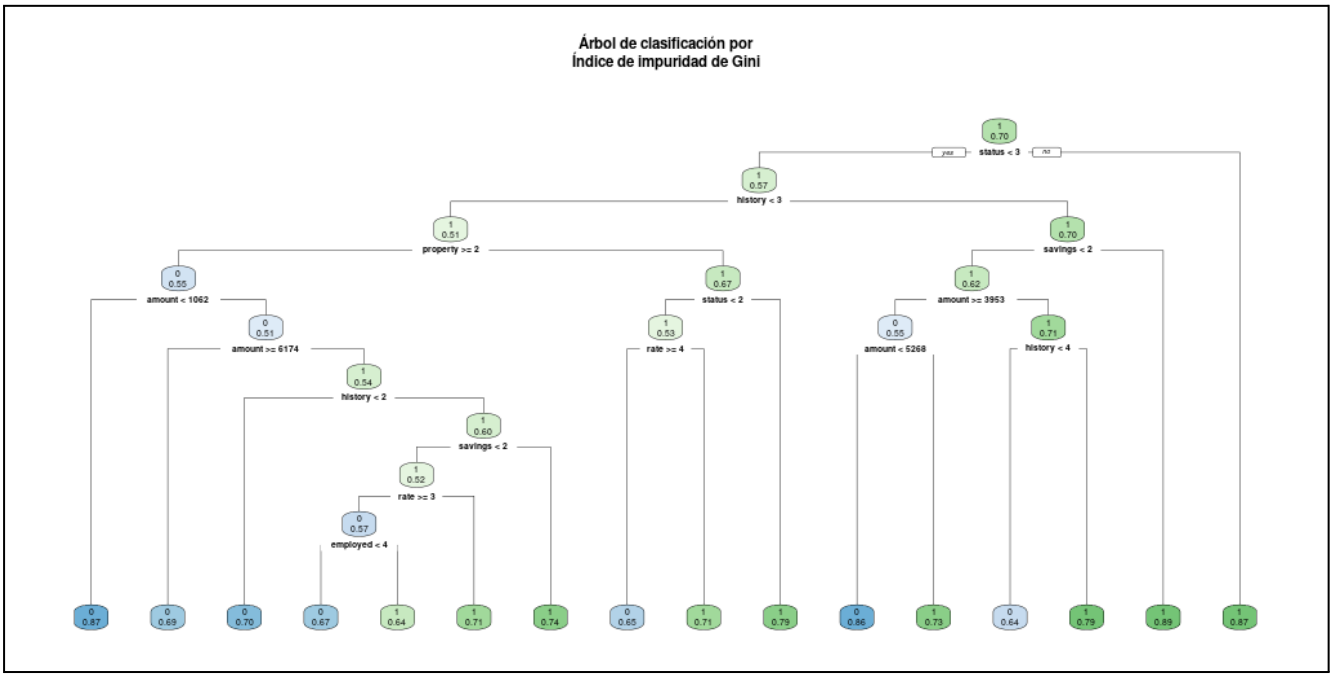
1. Influencia del método de división de nodos: *Entropía de la información* e *Índice de impuridad de Gini*.

   Al momento de separar el conjunto de datos de un nodo se busca que los nodos resultantes tengan una mayor homogeneidad de la variable endógena (la variable de interés cuyo valor buscamos predecir con el modelo), esto es, que sus valores sean cada vez más iguales, que la probabilidad del valor más común sea cada vez mayor. Dos de las métodos de cálculo de heterogeneidad más comunes son:

   - La ***Entropía de la información***: siendo esta 0 cuando todos los valores de la variable endógena son iguales y es totalmente homogénea, y 1 cuando todos los valores son igualmente probables y es totalmente heterogénea.
   - El ***Índice de impuridad de Gini***: siendo esta 0 cuando no existe desigualdad de valores en la variable endógena y es totalmente homogénea, y 0,5 cuando la desigualdad de los valores es máxima y es totalmente heterogénea.

   Al observar los árboles de decisión generados llama la atención lo similares que son: los dos poseen **9 niveles** de profundidad y ambos utilizan las mismas 7 variables





para clasificar: el historial de cuenta corriente (**status**), el historial de créditos (**history**), el bien más valioso que la persona posee (**property**), el monto del crédito solicitado (**amount**), la tasa del monto del crédito por el sueldo del solicitante (**rate**), el historial de ahorros (**savings**) y el tiempo de empleo (**employed**). Sin embargo, la diferencia se produce con la ramificaciones, y es que al utilizar el método de Gini vemos que al tercer nivel de división nacen ramas desde un nodo que es considerado como terminal por el método de entropía: la variable **savings** con un valor menor a 2 genera 3 nodos divisorios más que, no obstante, ocupan variables que anteriormente usadas, pero con otros valores de corte. Salvo por estas ramas, el resto del árbol es el mismo para ambos métodos, incluso con los valores de división en sus variables donde coinciden las ramificaciones. Con esto se aumentó la complejidad del modelo desde los 11 nodos divisorios a los 15.

Al tener una mayor complejidad es de esperar que la exactitud usando el Índice de impuridad de Gini sea mayor que con el método de entropía de la información, lo cual se comprueba al comparar los resultados de exactitud de la matriz de confusión con los datos de entrenamiento: ***80,3%*** y ***78,5%*** de exactitud, respectivamente. De forma análoga los resultados son mejores para la especificidad: **46,4%** para el método de entropía y **54,6%** para Gini.

Ahora bien, al evaluar la capacidad predictiva de los modelos con el grupo de datos de validación se observa, como es de esperar, una disminución de la exactitud, pero con una leve mayor exactitud del método de entropía (***73,6%***) por sobre Gini (***73,2%***), lo cual puede explicarse por el fenómeno de *overfitting* asociado al aumento de complejidad que presentó este segundo modelo. En cambio, al evaluar la especificidad, si bien ambos bajan, Gini mantiene la ventaja con un **43,6%** contra un **42,3%** de especificidad para el método de entropía. Si bien cada método es superior en alguna métrica, al considerar ambos Gini presentaría mejor especificidad sin comprometer tanto la exactitud, como el método de entropía compromete la especificidad por una mejor exactitud, por lo cual en esta comparación el método de Gini representaría la mejor opción.

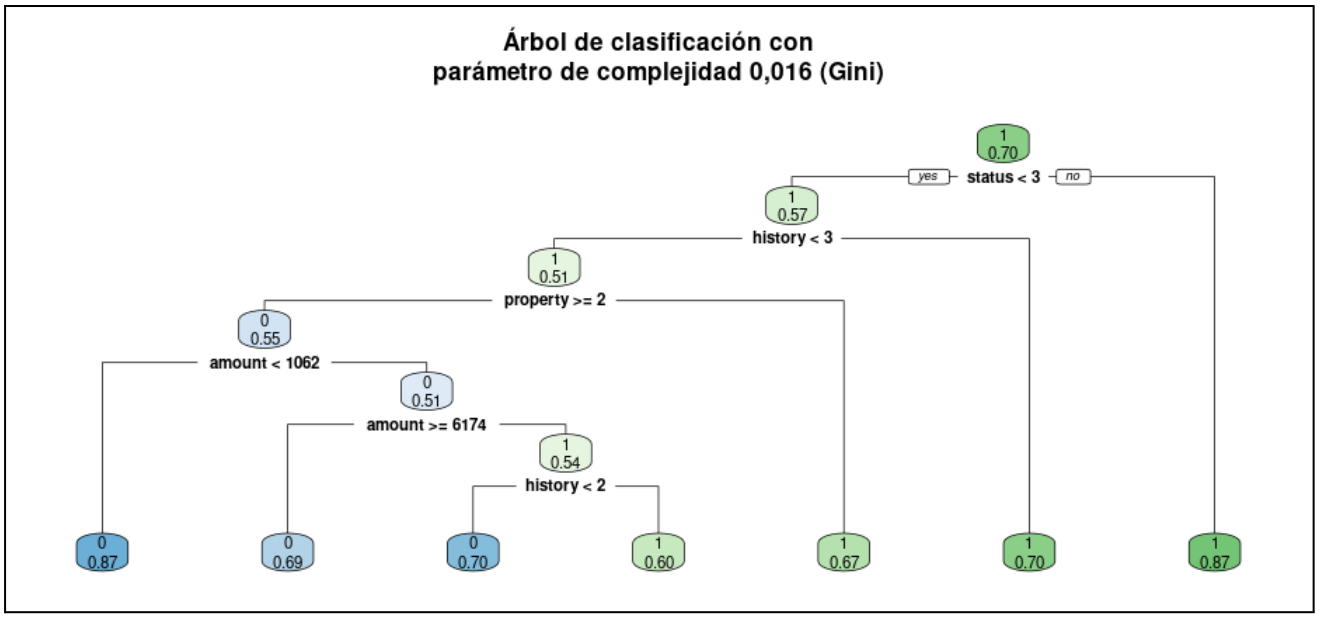
2. Influencia del costo asociado a la complejidad: el parámetro de complejidad.

Un modelo tipo árbol de decisiones que aumenta en complejidad, es decir, que tiene mayor cantidad de nodos de división, puede presentar nodos finales cada vez más precisos, con un mayor nivel de homogeneidad. Sin embargo, esto puede ser un problema pues es posible que al aumentar la complejidad no exista una ganancia significativa de homogeneidad, pero sí un aumento en la necesidad de cómputo. Por otra parte, se aumenta el riesgo de *overfitting*, esto es, que el modelo se ajuste demasiado a los datos de entrenamiento, pero que empeore su capacidad predictiva al aportar nuevos datos.

Para evitar estos problemas se puede definir un parámetro de complejidad o valor *lambda* para penalizar el exceso de complejidad de forma tal que si no existe una ganancia significativa de homogeneidad no se proceda con mayores divisiones. Al revisar la tabla de parámetros de complejidad generada en ambos modelos del apartado anterior resalta el valor delta de ***0,016*** al generar **6** nodos de división, pues en manos casos hasta este nivel se logran cambios que me parecieron significativos en el error relativo (***79,73%*** del error original en comparación al ***72,52%*** que se lograba con los originales 11 nodos divisorios). Para fines prácticos se comparará el modelo sin este valor fijado contra uno con el parámetro de complejidad fijado en **0,016** usando el método de división de Gini, pues tuvo mejor rendimiento.

Lo primero que llama la atención es la evidente menor complejidad: los esperables 6 nodos divisorios con 6 niveles de profundidad, en donde las variables **status**, **history**, **property** y **amount** siguen siendo protagonistas. Llama la atención que para definir las 6 divisiones más significativas desde el nodo con la división **property** mayor o igual que 2 se omitiera la ramificación hacia la derecha (no cumple la condición), decantándose por aumentar la profundidad hacia la izquierda, lo cual puede ser por la cantidad de datos de cada nodo: desde este punto a la derecha se agrupa el 10% de los registros originales, mientras que al llegar a la división final por

la rama izquierda (**history** menor que 2) aún se conserva un grupo con el 19% de los registros originales, dando más peso y relevancia a esta rama.



Ahora bien, al momento de comparar el rendimiento del modelo con el parámetro de complejidad fijado con el mismo grupo de datos con los cuales se entrenó se observa una disminución en la exactitud obtenida con la matriz de confusión, lo cual es esperable dado la menor complejidad de este modelo (***76,4%*** vs ***80,3%*** del original), pero es mucho más notoria la baja en la especificidad (desde un **54,6%** a **31,5%** usando el parámetro de complejidad). Sin embargo, al evaluar con los datos de validación se observa una leve disminución en la exactitud de la matriz de confusión (***75,6%***), pero que es llamativamente mayor al valor de exactitud del modelo sin parámetro de complejidad fijado (***73,2%***). En cuanto a la especificidad, esta mejora un poco al usar el valor *lambda* (***32,1%***), pero sigue siendo inferior al modelo original con el mismo grupo de datos (**43,6%**). Si bien la exactitud mejora 2,4 puntos porcentuales al fijar un parámetro de complejidad, la especificidad se ve muy castigada, con 11,5 puntos porcentuales menos, por lo que el modelo original con método de división de Gini sigue siendo la mejor opción.

3. Influencia de los criterios de parada relativos: observaciones predivisión, observaciones posdivisión y profundidad máxima.

Fijar un parámetro de complejidad no es la única estrategia para definir límites al crecimiento de los árboles de decisión. A continuación se presentan otras tres estrategias:

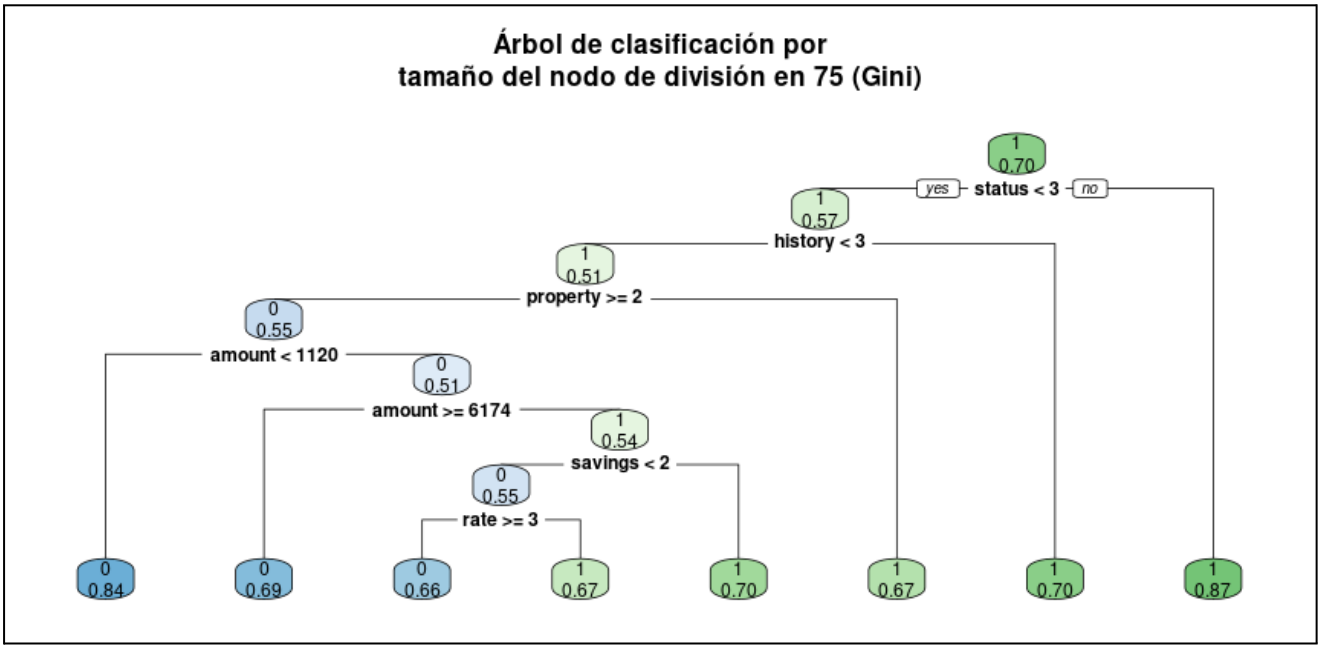
- *Límite de tamaño del nodo divisible*: es posible definir un mínimo de observaciones que deba tener un nodo para poder dividirse, con lo cual si no las cumple éste queda como nodo terminal. Se escogió el valor de ***75*** pues corresponde al 10% de la muestra de entrenamiento y 5 de los 15 nodos divisorios del modelo original no alcanzan esa proporción,pareciéndome un valor interesante para hacer el corte.

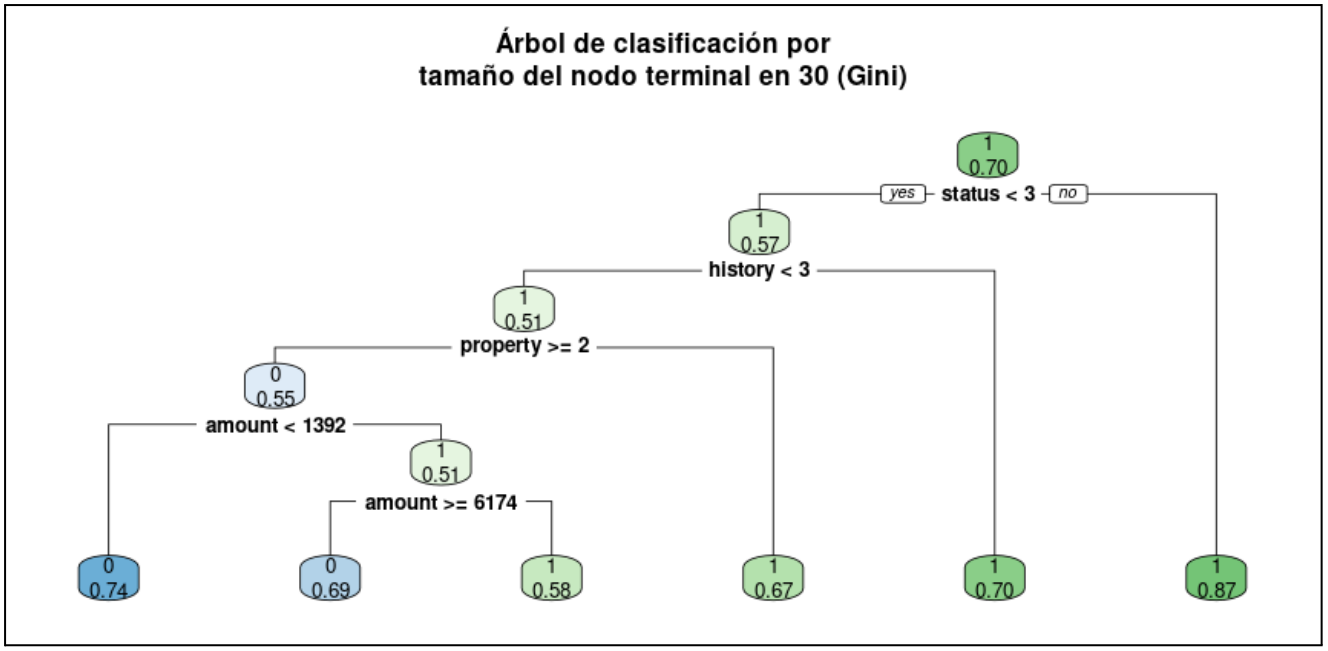
- *Límite de tamaño del nodo terminal*: es posible definir un mínimo de observaciones que deba tener un nodo después de una división con lo cual, si no las cumple se deshace la división y el nodo *padre* queda como nodo terminal. Se escogió el valor de ***30*** pues corresponde al 4% de la muestra de entrenamiento y si bien todos los nodos divisorios tienen una mayor porcentaje de datos, 8 de 15 nodos terminales tienen una proporción menor, por lo que me pareció interesante de evaluar.

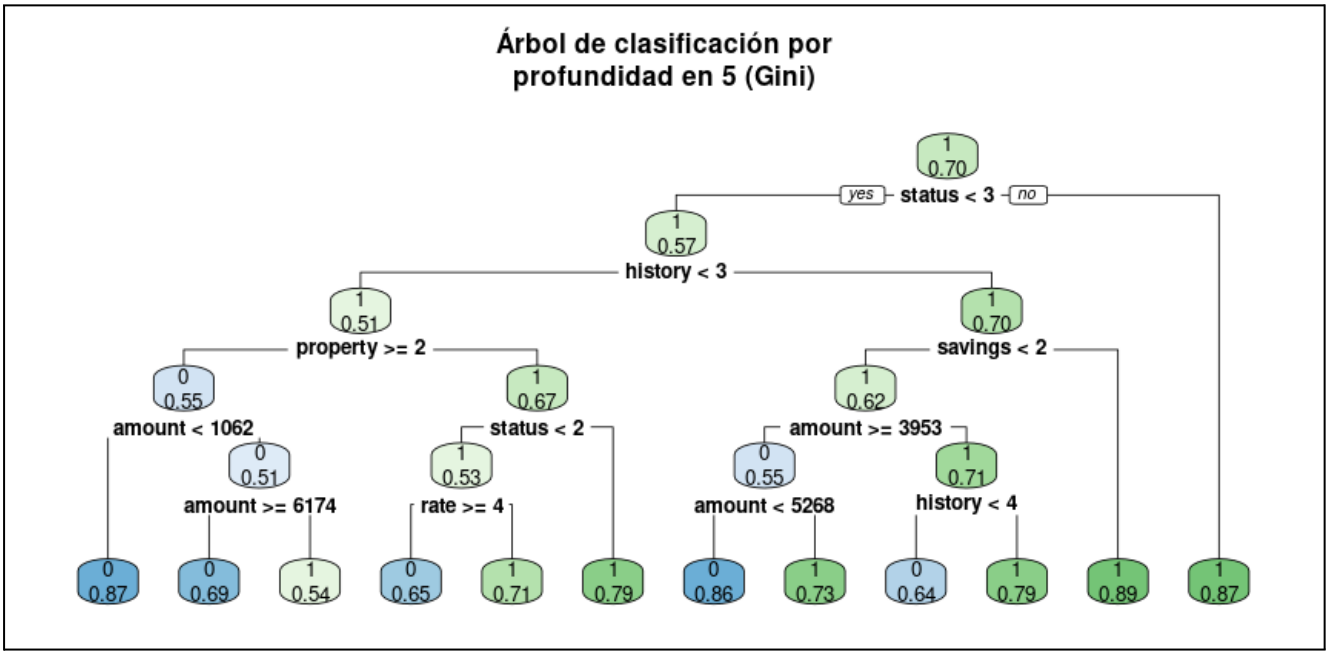
- *Límite de niveles del árbol*: es posible definir un máximo de niveles de profundidad para el modelo con lo cual, si se alcanzan los niveles divisorios indicados, los eventuales hijos quedan como nodos terminales. Se escogió el valor de ***5*** por ser la mitad de la profundidad del modelo original (9 niveles de división más 1 de nodos terminales).

En este caso el desarrollo de los 3 modelos se hará utilizando el método de división basado en el índice de impuridad de Gini, al igual que en el apartado anterior y por las razones ahí expuestas.

Lo primero que salta a la vista es que, además de la esperable menor complejidad en comparación al árbol del modelo sin criterios de parada fijos, los árboles comparten similitud sólo hasta el segundo nivel de división, donde se usan las variables **status** e **history**. De aquí en adelante están las diferencias: el árbol con profundidad fija es el único que conserva la rama derecha debajo de **history**, esa que hace diferente al árbol del método de división de Gini en comparación al del método de entropía. En los otros árboles se puede ver que si bien comparten el nodo de **property** mayor o igual a 2, en el cuarto nivel presentan un nodo para la variable **amount**, pero ambos con valores de corte distintos: el modelo con el nodo predivisión fijado tiene un corte en menor que 1.120, mientras que el modelo con el nodo posdivisión fijado lo tiene en menor a 1.392. Esto incide en que para el primer árbol se agregan las variables **savings** y **rate**, dejando 7 nodos divisorios y 8 nodos terminales, mientras que en el árbol con el nodo posdivisión fijado no se agreguen nuevas variables exógenas y se tiene una menor complejidad (5 nodos de división y 6 nodos terminales). Finalmente, el nodo con la profundidad fija termina agregando las variables **property**, **amount**, **savings** y **rate**, como si un fino cuchillo hubiese pasado por el modelo original.







Al evaluar la exactitud de los modelos con los datos de entrenamiento, ninguno logra superar al modelo original (**80,3%**), esperable por la menor complejidad, teniendo un empate los modelos con el nodo predivisión fijado y con profundamente fijada (***77,3%*** de exactitud), seguidos del modelo con nodo posdivisión fijado (***75,9%***); para la especificidad el modelo sin parámetros fijados también presenta la mejor puntuación (**54,6%**), seguidos por los árboles con nodo predivisión fijo, profundidad fija y nodo posdivisión fijo: **40,5%**, **36,5%** y **30,6%** de especificidad, respectivamente.

Al comparar la exactitud de la matriz de confusión usando los datos de validación los modelos, como es de esper, bajan su porcentaje: ***72,8%*** para los árboles con valor del nodo posdivisión fijo y con profundidad fija, y ***72,4%*** para el árbol con nodo predivisión fijado, ninguno de los cuales alcanza la exactitud del modelo original con

los datos de validación (***73,2%***). Por su parte, la especificidad baja de manera importante: **34,6%** para el modelo con valor del nodo predivisión fijo, **26,9%** para el modelo con la profundidad fija, y **24,4%** para el modelo con el valor del nodo posdivisión fijo, todos lejos del **43,6%** de especificidad que tiene el modelo original con método de división de Gini. Ante este escenario el modelo original sigue siendo la mejor opción de todos los árboles generados.

K vecinos cercanos.

El algoritmo de árbol de decisión no es el único modelo supervisado que nos sirve para clasificar datos. Otra alternativa es usar el modelo KNN, el cual será usado a continuación. La idea de este modelo es clasificar el valor de la variable endógena para un registro de acuerdo al valor que para esta variable endógena tienen un determinado número K de vecinos cercanos. Para escoger a los vecinos más cercanos se calcula la *distancia* para cada una de las variables exógenas con respecto a la misma variable del dato sin clasificar, se combinan las distancias de todas estas y se escogen los K registros considerados con menor distancia; posteriormente se suma el valor de la variable endógena de dichos vecinos y se escoge como predicción el valor más frecuente, el valor más *votado*. Para determinar la distancia existen diversas estrategias, siendo la *distancia Euclidiana* una de las más usadas, y el método que se utilizará en este ejercicio.

Para poder generar este modelo el conjunto de datos de entrenamiento se tomó como el conjunto de registros en que la variable endógena es conocida y definirán el valor para **credit** de los registros en que la variable endógena es desconocida, rol que asumió el grupo de datos de validación. Es importante hacer notar que se probaron múltiples combinaciones en el valor de K, la elección de las variables a considerar y el tratamiento de estas, evaluando su rendimiento mediante el cálculo de la exactitud y la especificidad. El modelo que se presenta a continuación es el que consideré mejor para este grupo de datos.

Variables.

Antes de la selección fue necesario hacer un tratamiento a las variables. Por un lado las variables categóricas fueron divididas en **N** variables binarias, tantas como categorías tuviese la variable original, con un valor de **1** cuando correspondía al de la categoría y **0** si no; si bien podrían haberse creado **N - 1** variables binarias, entendiéndose la última como aquella en que todas las otras tuviesen valor **0**, el rendimiento del modelo fue levemente peor, por lo que se prefirió crear las **N** variables binarias. Por otra parte, las variables contínuas fueron normalizadas, asignándoles un valor entre **0** y **1**.

Al momento de seleccionar las variables del modelo quise trabajar sin preconcepciones teóricas: obviamente el conocer el negocio financiero es vital para poder trabajar con datos de créditos, pero personalmente no cuento con la formación ni la experiencia para poder tomar algún enfoque inicial; además, me gusta dejar que los datos hablen y, de acuerdo a las métricas seleccionadas, expliquen de mejor forma los fenómenos. Es por esto que para poder empezar a trabajar se tomaron 2 enfoques iniciales: probar todas las variables resultantes de las transformaciones anteriormente descritas (quedando en total 50 variables

exógenas) y probar sólo las variables derivadas de aquellas que fueron comunes a todos los árboles anteriormente diseñados (**status**, **history**, **property** y **amount**, con un total de 13 variables exógenas). Puesto que el rendimiento fue mejor con este último enfoque se agregó variables de entre aquellas que aparecieran por lo menos 1 vez en alguno de los árboles anteriores (**savings**, **employed** y **rate**), jugando con distintas combinaciones. Finalmente, tomando en cuenta maximizar los valores de exactitud y especificidad, estas variables fueron máximas al trabajar con **status**, **history**, **property**, **savings** y **amount**, para un total de 18 variables exógenas.

Valor de K.

La selección del número de vecinos cercanos, el valor K, fue otro punto crucial, pues no existe un número o expresión que defina la mejor opción. Ante esto, se decidió una aproximación *de fuerza bruta*: se generó para cada una de las combinaciones de variables con la que se trabajó 20 modelos KNN probando valores K desde el 1 al 100. Se decidió trabajar con 20 repeticiones para cada valor de K pues el cálculo del modelo puede presentar variaciones en caso de empate de vecinos, pudiendo tomar valores al azar en algunas ocasiones, por lo que tomar varios intentos y promediar sus resultados para exactitud y especificidad parece ser lo más adecuado para disminuir la incertidumbre de lo aleatorio. Por otra parte, al decidir generar modelos con valores K hasta el 100, si bien la cifra fue escogida desde la arbitrariedad, se esperaba abarcar un gran número de posibles soluciones en que se pudiera optimizar el modelamiento de la información. Esto significa que para cada combinación de posibles variables a utilizar se realizaron 2.000 cálculos de modelos KNN para buscar la mejor exactitud, y otros 2.000 cálculos para encontrar la mejor precisión. Luego de toda esta combinación de variables y valores de K, se fijó como óptimo un valor de K igual a **7**.

Resultados del modelo.

Como se extrae de los párrafos anteriores, el modelo KNN se realizó con un valor K de **7** y usando 18 variables: **status**, **history**, **property**, **savings** en sus transformaciones binarias y **amount** normalizada. El resultado, evaluando los 250 registros del grupo de datos de validación se refleja en estas 2 matrices de confusión:

Matiz de confusión 1

		Valores de referencia		
		1	0	
Valores predichos	1	***146***	***42***	**188**
	0	***26***	***36***	**62**
		172	**78**	**250**

Matiz de confusión 2

		Valores de referencia		
		0	1	
Valores predichos	0	***147***	***42***	**189**
	1	***25***	***36***	**61**
		172	**78**	**250**

Primero que todo quisiera aclarar por qué muestro 2 matrices de confusión. La razón es que, probablemente, para uno de los casos de validación se produce un empate entre clasificarlo como 0 o 1, situación que es perceptible gracias a las múltiples repeticiones en el cálculo de la precisión, no afectando esta situación a la especificidad. Por convención, se ha dejado al valor *positivo* en la columna de la izquierda (referencia) y en la primera fila (predicho), y al valor *negativo* en la columna de la derecha (referencia) y en la segunda fila (predicho).

Para la matriz de la izquierda el valor de exactitud es de 72,8%, mientras que para la matriz de la derecha la exactitud es de un 73,2%, por lo que consideraré el promedio de ambos

para evaluar el modelo con un ***73%*** de precisión. Como mencioné anteriormente la especificidad no se ve afectada por el azar y queda en un ***46,2%***. Si bien estos no son los mejores valores de todos los obtenidos en la fase de experimentación esta sí es la mejor combinación en que se puede maximizar la precisión sin castigar demasiado la exactitud (la mejor exactitud obtenida fue de ***74,9%***, pero con un valor de la especificidad del 38,9%; mientras que la mejor especificidad obtenida fue de ***47,4%***, pero con una exactitud muy castigada del 65,2%).

Comparación de ambos algoritmos.

Después de todo este análisis podemos, finalmente, comparar los dos modelos de aprendizaje estadístico supervisados de clasificación analizados acá:

	Modelo de árbol	***Modelo KNN***
Variables exógenas	*status*	*status*
	history	*history*
	property	*property*
	amount	*amount*
	savings	*savings*
	rate	-
	employed	-
Parámetro definitorio	*Indice de impuridad de Gini*	*K = 7*
Exactitud	*73,2%*	*73%*
Especificidad	*43,6%*	*46,2%*

- El árbol de decisiones con método de división basado en el **Índice de impuridad de Gini**, usando 7 variables exógenas: **status**, **history**, **property**, **amount**, **savings**, **rate** y **employed**.

- El modelo KNN con valor de K igual a **7** y con 18 variables exógenas derivadas de la transformación de **status**, **history**, **property**, **savings** y **amount** (creación de variables binarias para cada una de las categorías de las 4 primeras variables y normalización para la última).

Al momento de evaluar el rendimiento de ambos modelos, recordando que el valor de **credit** igual a **1** fue definido como positivo, podemos ver un comportamiento sin grandes diferencias: la exactitud, la capacidad que tiene el modelo para identificar correctamente tanto a casos *positivos* como *negativos* del total de casos de la muestra de validación es de ***73,2%*** para el árbol y de ***73%*** para KNN, unos escasos 0,2 puntos porcentuales de diferencia; en cambio para la especificidad, la capacidad del modelo para predecir cuáles casos son *negativos* de entre todos los casos efectivamente *negativos*, si bien los valores son cercanos, la diferencia se hace notar en favor del **46,2%** del modelo KNN contra el **43,6%** del modelo del árbol de decisiones. Como ha ocurrido durante la mayor parte del desarrollo de este reporte, la especificidad ayuda a definir la elección del modelo, decantando como mejor alternativa para subir a producción el modelo KNN.

Es de notar, para terminar, que los datos utilizados en este ensayo, por lo menos de la forma en que se usaron, no parecen ser los mejores para identificar con mayor seguridad a quienes no tienen buena salud crediticia. Se puede hipotetizar que el desbalance en la proporción de valores **0** y **1**, 30% y 70% para todos los grupos, aproximadamente, esté afectando para que los modelos tiendan a buscar mejores métricas prediciendo el valor más frecuente en desmedro del más raro. Quedará para otra ocasión intentar generar un modelo con datos más balanceados.

Evaluación 2
Curso Minería de datos
Néstor Patricio Rojas Ríos