UNIVERSIDAD SAN FRANCISCO DE QUITO

Minería de Datos Aplicada a Credit Scoring

Mauricio Figueroa

Tesis de grado presentada como requisito para la obtención del título de Maestría en Matemáticas Aplicadas

Quito

Septiembre 2006

Universidad San Francisco de Quito

Colegio de Postgrados

HOJA DE APROBACION DE TESIS

Minería de Datos Aplicada a Credit Scoring

Mauricio Figueroa

Carlos Jiménez, Ph.D	.....
Director de Tesis	(firma)
Ximena Córdova, Ph.D.	.....
Miembro del Comité de Tesis	(firma)
Carlos Jiménez, Ph.D.	.....
Director de la Maestría	(firma)
Víctor Víteri, Ph.D.	.....
Decano del Colegio de Postgrados	(firma)

Quito, Septiembre 2006

© Derechos de Autor

Mauricio Figueroa

2006

Dedicatoria

A mi querida madre, mi hermana y sobrino.

A mis tíos y primos.

A mis amigos.

Agradecimiento

A todos los miembros de la Universidad San Francisco de Quito, en particular, a mis profesores de la Maestría en Matemáticas Aplicadas, que hicieron de la maestría una realidad.

Mauricio Figueroa
Quito, Septiembre 2006

Resumen

En el presente trabajo se explora las técnicas de clasificación conocidas como análisis discriminante, análisis de Fisher, regresión logística, árboles de clasificación, redes neuronales y support vectors machines y las particularidades de su aplicación al proceso de aprobación de créditos de consumo denominado credit scoring.

Abstract

Methods for classification like discriminant analysis, Fisher analysis, logistic regression, classification trees, neural networks and support vectors machines and its applications to credit scoring are revised in this paper.

Índice general

Índice de cuadros

Índice de figuras

Capítulo 1

Introducción

Uno de los principales riesgos que las instituciones financieras deben manejar es el riesgo de crédito.

Para tratar con riesgos de crédito de consumo se puede recurrir a la metodología denominada credit scoring, que es una aplicación de la mayoría de las técnicas de clasificación existentes. Estas técnicas permiten construir modelos de decisión que determinan los aplicantes a un crédito que serán beneficiados. Con el término crédito de consumo se hace referencia a "cualquiera de muchas formas de comercio bajo las cuales un individuo obtiene dinero, bienes o servicios con la condición de una promesa de reembolsar el dinero o pagar por los bienes y servicios junto con una cuota, el interés, en alguna fecha futura o fechas futuras".

Un prestamista debe tomar dos decisiones. La primera, si otorgar crédito a un nuevo aplicante, y la segunda, qué acciones se debe realizar con las personas que han obtenido el crédito solicitado, por ejemplo, se puede preguntar cómo ajustar las restricciones de crédito o cómo modificar el esfuerzo de marketing dirigido al cliente actual. Las técnicas utilizadas para enfrentar la primera inquietud se denominan *credit scoring* y las utilizadas con la segunda se denominan *behavioral scoring*. En ambos casos, cualquiera que sea la técnica utilizada, lo vital es contar con una muestra grande de clientes anteriores con sus respectivas aplicaciones bien detalladas y su historia de crédito. En esta tesis solamente se abordará el caso de credit scoring.

Algunas técnicas descritas en este trabajo producen un score o puntuación para un aplicante nuevo que permite al prestamista tomar o no el riesgo de conceder el crédito. Otras no producen tal score, en lugar de esto, indican directamente la probabilidad de que el nuevo aplicante sea un "buen" o un "mal" pagador. Y, una de estas técnicas en particular produce modelos de decisión en un lenguaje muy similar

a lo acostumbrado por los oficiales de crédito.

Como las técnicas de clasificación forman parte de las técnicas de minería de datos, el proceso de credit scoring puede ser visto como una aplicación de la minería de datos. La minería de datos o data mining es la exploración y análisis de datos con el propósito de descubrir patrones y relaciones en estos. “Cuando buscas minerales, para tener éxito se debe saber dónde buscar y reconocer lo que es importante cuando lo encuentras”. En este caso, se busca información que ayude a un prestamista decidir si otorgar o no un crédito solicitado.

La metodología de credit scoring ha ganado importancia gracias al New Basel Capital Accord, llamado Basel II, que se enfoca en técnicas que permiten a los bancos y supervisores evaluar adecuadamente los riesgos que un banco enfrenta. Así, credit scoring puede contribuir en los procesos de evaluación interna de una institución.

Esta tesis tiene el propósito de mostrar que con un conocimiento adecuado de las fortalezas, debilidades y supuestos de las técnicas clasificación y la utilización de la estructura de la minería de datos se pueden obtener modelos de decisión potentes que permitan significativamente reducir el riesgo al que se enfrenta un prestamista al otorgar un crédito.

La presente tesis está organizada de la siguiente manera. En este capítulo, se presenta una visión global de credit scoring tradicional desde el punto de vista de una aplicación en el ámbito financiero. En el Capítulo (2) se propone abordar la creación de un sistema de credit scoring bajo el enfoque de la minería de datos. En el Capítulo (3) se explica detalladamente las técnicas más utilizadas y las más recientes que podrían ser de utilidad en el desarrollo de sistemas de credit scoring. En el Capítulo (4) se describe el conjunto de datos utilizado en esta tesis, y se presenta el proceso mediante el cual se determina el mejor modelo de decisión para estos datos. El resumen de los modelos y las conclusiones se detalla en el Capítulo (5). En el Apéndice se menciona ciertos detalles técnicos concernientes a los modelos utilizados.

Como un dato adicional se debe mencionar que credit scoring es un área en la cual un pequeño mejoramiento en el desempeño de la clasificación puede significar un tremendo incremento en las ganancias del prestamista debido al volumen de préstamos otorgados gracias a la utilización de un sistema de credit scoring. Un descenso del 0,25 % en la proporción malos puede ahorrar millones a la institución financiera.

1.1. Un Resumen General de Credit Scoring

El principal tópico en las finanzas modernas es el pronóstico de riesgos. Aparte de la administración del portafolio, la valoración de opciones, bonos o de otros instrumentos financieros, credit scoring representa otro conjunto de procedimientos importante para estimar y reducir el riesgo de crédito. Involucra técnicas que ayudan a las organizaciones financieras decidir si otorgar o no un crédito a nuevos aplicantes. Básicamente, credit scoring trata de distinguir dos subgrupos diferentes en los datos históricos disponibles. El objetivo es seleccionar la técnica que haga pronósticos suficientemente precisos en tiempo real.

1.1.1. Un poco de historia

Credit scoring esencialmente es una forma de identificar grupos diferentes en una población cuando no se puede ver las características que definen a los grupos sino solamente características relacionadas. Los procedimientos de identificación de grupos se originaron alrededor del año 1936 con Fisher. En esta época algunas de las casas financieras y empresas de pedidos por correo tuvieron problemas con la gestión de créditos, principalmente porque la decisión de otorgar un crédito o enviar mercadería a los aplicantes fue no uniforme, subjetiva y dependiente de las reglas de la empresa y del conocimiento empírico del analista de crédito. A finales de los años 60, en Estados Unidos, el número de solicitudes de tarjetas de crédito aumentó considerablemente, de tal forma que los analistas de crédito no alcanzaban a lidiar con esta situación. Esto permitió la creación de sistemas automáticos de toma de decisiones que eran baratos, rápidos, objetivos y libres de juicios personales. Luego de la aplicación de los sistemas de credit scoring las tasas de incumplimiento bajaron en un 50 %, según Komorád [6]. El éxito de credit scoring en las tarjetas de crédito impulsó a las instituciones financieras a utilizar métodos de scoring en otros productos como préstamos personales, préstamos hipotecarios, préstamos para pequeños negocios, etc.

El evento que aseguró la completa aceptación de la metodología de credit scoring fue la Equal Credit Opportunity Acts y sus enmiendas en los Estados Unidos en 1975 y 1976. Esta declaró ilegal la discriminación al momento de otorgar créditos a menos que la discriminación “sea obtenida empíricamente y sea válida estadísticamente”.

1.1.2. Credit scoring en la actualidad

Al menos en Estados Unidos, las empresas de pedidos por correo, las de publicidad, los bancos y otras instituciones financieras utilizan los métodos de credit scoring para asignar un score a sus clientes, aplicantes y clientes potenciales, para estimar y minimizar el riesgo de crédito. Internacionalmente, The Basel Committee on Banking Supervision formula extensas directrices de supervisión para bancos y en general apela a que se precise las evaluaciones internas y anima a converger hacia metodologías universales. Los miembros de este comité son de Bélgica, Canadá, Francia, Alemania, Italia, Japón, Luxemburgo, Holanda, España, Suecia, Suiza, Reino Unido y Estados Unidos. En 1988, este comité decidió introducir un sistema de medida denominado The Basel Capital Accord. Esta estructura ha sido introducida progresivamente en los países miembros y en aquellos países con bancos internacionales activos. En junio de 1999, este comité propuso un nuevo acuerdo para reemplazar al de 1998 denominado Basel II que pone más énfasis en las metodologías internas propias de los bancos. Por lo tanto credit scoring y sus técnicas pueden ser materia de interés para los bancos cuando éstos traten de hacer sus evaluaciones internas más precisas y correctas como sea posible. La información del Basel II se encuentra en http://www.bis.org.

1.1.3. Aspectos filosóficos respecto al Credit Scoring

El objetivo del credit scoring es predecir el riesgo, no explicarlo. Por más de 50 años, el ánimo ha sido predecir el riesgo de que un consumidor caiga en mora. Recientemente, el método ha sido predecir el riesgo de que un consumidor no responda a una campaña de anuncios por medio de correo electrónico para una nuevo producto, el riesgo de que un consumidor no se decida por un producto a crédito, o aún el riesgo que un consumidor mueva su cuenta a otro prestamista. *Cualquiera que sea su uso, el punto vital es que credit scoring es un predictor de riesgo, y no es necesario que el modelo predictivo explique el por qué algunos consumidores caen en mora y otros no.*

"El pragmatismo y empirismo de credit scoring implica que cualquier característica del consumidor o del medio ambiente del consumidor que ayude a la predicción debería ser utilizada en el sistema de scoring". La mayoría de las variables utilizadas tienen conexiones obvias con el riesgo de mora. Algunas dan la idea de estabilidad (tiempo en la dirección, tiempo en el presente empleo), algunas reflejan la realidad financiera (tiene una cuenta, tiene tarjetas de crédito, tiempo en el banco actual), otras dan la fuentes del consumidor (estatus residencial, empleo, empleo de la esposa),

mientras que otras miran los posibles dependientes (número de hijos, número de dependientes). No hay necesidad de justificar la presencia de alguna característica, si ayuda a la predicción, debería ser utilizada. Se podría utilizar la información de las personas que han vivido en la misma dirección del consumidor, inclusive.

Los bancos americanos o ingleses tienen algunos problemas al momento de construir su modelo de decisión. La ley no les permite utilizar información sobre la raza, nacionalidad, religión, género o estado civil. La edad se puede utilizar siempre que las personas mayores de 62 años no sean discriminadas. Se debe mencionar que en el Ecuador no existen restricciones legales para las variables incluidas en modelos de credit scoring.

Algunos estudios han mostrado que si fuera posible utilizar género, entonces más mujeres conseguirían un crédito. Esto es porque otras variables como bajo ingreso y empleo a tiempo parcial son predictores de comportamiento de repago bueno en mujeres pero pobre comportamiento de repago en toda la población. No se utiliza porque se cree que se discriminizará en contra de las mujeres.

Otras características no son utilizadas para predecir el riesgo de mora porque son culturalmente inaceptadas. Así un historial de mala salud o encarcelamiento por infracciones de tránsito son predictores de un riesgo de mora creciente, pero los prestamistas no las utilizan por temor al qué dirán de la sociedad.

1.1.4. La evaluación de crédito antes del scoring

La evaluación tradicional de crédito depende del "buen tacto" que tenga el adminiustrador del banco y de la evaluación del carácter del posible solicitante de crédito y de su habilidad de repago. Esto significa que el posible solicitante no se acercaba al administrador del banco hasta que ha guardado dinero o utilizado otros servicios durante años. Entonces, se hacia una cita y el cliente, vistiendo de lo mejor, pediría prestado algo de dinero.

Luego, el administrador consideraría la proposición y, a pesar del tiempo de la relación con el cliente, ponderaría la probabilidad de repago y evaluaría la estabilidad y honestidad del individuo y su carácter. El administrador evaluaría el uso propuesto del dinero, y entonces probablemente averiguaba a una referencia independiente como un líder comunitario o empleador. Arreglaba probablemente una reunión con el cliente y entonces quizás tomaba una decisión y le informaba. Este proceso fue lento e inconsistente. Si se le preguntaba al administrador del banco sobre esto, él respondería que tal tarea

requiere muchos años de entrenamiento y experiencia.

Estas desventahas han existido desde hace algún tiempo, así qué paso para las cosas cambien?

Los bancos tienen que vender productos no solo a consumidores conocidos; el crecimiento fenomenal de las tarjetas de crédito obligaban a tener un mecanismo para tomar la decisión de prestar o no muy rápidamente porque no se tenía tiempo para lidiar con el volumen de aplicaciones y con las entrevistas; los bancos se enfocaban casi siempre exclusivamente en grandes préstamos y clientes corporativos. Ahora, los préstamos a clientes es una parte importante y creciente para el banco. Se dieron cuenta que en los créditos de consumo el animo no es evitar pérdidas sino maximizar beneficios. Mantener las pérdidas bajo control es parte de eso, pero uno puede maximizar ganancias teniendo un pequeño nivel controlado de malas deudas y así expandir la cartera de crédito de consumo.

1.1.5. La metodología de scoring para evaluar créditos

En general, el posible cliente presenta una proposición al prestamista. El prestamista considera la proposición y evalúa el riesgo asociado. Anteriormente, los banqueros ondeaban una barita mágica para saber si el riesgo está en un nivel bajo aceptable. Con scoring, el prestamista aplica una fórmula a los elementos importantes de la aplicación, y el resultado de esta fórmula da una cuantificación numérica del riesgo. Nuevamente, la proposición será aceptada si el riesgo es convenientemente bajo.

La forma en que credit scoring se acopla en la evaluación de crédito puede variar un poco de producto a producto. En estos días, el aplicante llena una forma, que puede estar en un papel o en una pantalla de computadora. Típicamente, los datos de la aplicación serán ranqueados. No todos los datos de la aplicación serán utilizados en calcular el score de crédito. Sin embargo, la información restante se necesita para varios propósitos, incluyendo identificación, seguridad, y futuros scorecards.

El cálculo del score de crédito puede incluir alguna información de un buró de crédito. En muchos casos y ambientes, el resultado del proceso de credit scoring hace una recomendación o decisión con respecto a la aplicación. El rol de la evaluación subjetiva humana ha sido reducido a un pequeño porcentaje de casos donde hay una genuina oportunidad para el prestamista humano de añadir valor.

Para ser concretos, consideremos una operación de scoring simple. Suponga que tenemos un scorecard con cuatro variables o características: residencia, edad,

propósito del crédito, y valor de las multas por infracciones de tránsito. Los scores para los niveles de las variables consideradas se muestran en el Cuadro (1.1).

Residencia		Propósito	
Propietario	36	Carro nuevo	41
Inquilino	10	Carro de segunda	33
Vive con los padres	14	Mejoras a la casa	36
Otro	20	Feriado	19
No responde	16	Otro	25

Edad		Multa (en dólares)	
18-25	22	Ninguna	32
26-35	25	1-299	17
36-43	34	300-599	9
44-52	39	600-1199	-2
53+	49	1200+	-17

Cuadro 1.1: Un ejemplo simple de un scorecard

Un aplicante de 20 años, que vive con sus padres, que desea pedir dinero para obtener un carro de segunda y que nunca ha tenido un multa, obtendrá un score de 101 $(22 + 14 + 33 + 32)$. Para un aplicante de 55 años de edad, propietario de su vivienda, que ha tenido \$250 de multa y desea pedir dinero para la boda de su hija, obtendrá un store de 127 $(49 + 36 + 17 + 25)$.

Note que no se está diciendo que alguien 53+ tiene un score de 27 puntos más que alguien de 18-25. Para la edad este diferencial es verdadero. Sin embargo, hay correlaciones involucradas. Por ejemplo, alguien de 53+ es más probable que sea dueño de casa que alguien de 18-25, mientras que puede ser raro encontrar a alguien de 53+ que viva con sus padres. Así lo que podemos encontrar es que alguien en la categoría de mayor edad puede tener un score, en promedio, 40, 50 o 60 puntos más una vez que las otras características han sido tomadas en cuenta.

Al armar un sistema de credit scoring, se debe decidir la marca a pasar. Esta es una cosa simple de implementar pero no necesariamente es simple de decidir.

Supongamos que en el ejemplo anterior, la marca a pasar es de 100. Así, cualquier aplicación que obtenga un score de 100 o más tendrá la aprobación. Este será el caso cualquiera sea la respuesta de cada una de las cuatro variables. Lo que scoring permite, por tanto, es un balance, una debilidad en un factor puede ser compensada por una fortaleza en los otros factores.

Algunos prestamistas utilizarán la política de ser estrictos respecto de la marca elegida. Si el score es mayor o igual que la marca, la aplicación es aprobada.

Si el score es menor que la marca, la aplicación es rechazada. Para otros prestamistas aplica una variación simple de esto. Una banda de referencia o un área gris es creada. Esto puede ser 5 o 10 puntos en uno o los dos lados de la marca. Las aplicaciones que caen en el área gris son revisadas en detalle. Otros prestamistas utilizan políticas que obligan a que las aplicaciones potencialmente aceptadas a caer en una banda gris. Por ejemplo, esto puede ser el caso de una aplicación que alcanza la marca pero hay un evento adverso en la información del buró de crédito, por ejemplo, la quiebra de un negocio. En otras palabras, no se permite que las fortalezas de la aplicación compense automáticamente una debilidad. Por último, otro grupo de prestamistas operan lo que se conoce como categorías super-pass y super-fail. Obtener información de un buró de crédito tiene un costo. Por tanto, hay casos que obtiene un score sumamente pobre que el mejor reporte del burde crédito no llevará al score a la marca. Estos se consideran super-fail. En el otro extremo, hay casos que obtienen scores super buenos que el peor reporte del buró no reducirá el score bajo la marca. Estos son los super-pass. En efecto, estos prestamistas disponen de dos o tres marcas: una para definir super-fail, otra para definir super-pass, y otra, entre las dos, que se utiliza una vez que la información del buró es utilizada.

1.1.6. Selección de la marca

Asumamos por un momento que tenemos información perfecta del desempeño futuro de los grupos en términos de reembolso, reembolso temprano, atrasos, pérdida, y el tiempo de estos eventos. En tal caso, debemos ser capaces de evaluar cuánto dinero este grupo de aplicaciones harán para la organización si elegimos aceptarlos. Podemos producir estos datos en ingresos y pérdidas para cada posible score, o para cada score que es candidato a ser la marca. Una forma simple, pero no necesariamente incorrecta, es aceptar a todas las aplicaciones que generarán una ganancia, aunque pequeña. Se puede argumentar que se puede aceptar a las aplicaciones cuya ganancia sea cero también.

Esta estrategia podría fracasar principalmente por razones contables. Necesitamos expresar las ganancias y pérdidas en términos de su valor presente para considerar el retorno de la inversión. Lo que es importante desde el punto de vista de scoring es que se pueda obtener una marca utilizando retorno. Por tanto, pondríamos nuestra marca en un punto donde todas las aplicaciones cumplan un retorno mínimo requerido.

Un desafío que surge al evaluar el retorno de una aplicación considerada

es que necesitamos asumir que el futuro será como el pasado. No podemos saber si el préstamo será pagado antes de lo acordado o por adelantado. Solo podemos hacer algunas suposiciones basados en el desempeño pasado de préstamos similares. También necesitamos suponer el desempeño de recuperación de futuros casos problema; los casos problema surgen de aplicaciones que no se han aprobado todavía.

Por tanto, al evaluar cuál marca utilizar, varios problemas interrelacionados de crédito y contables deben ser considerados.

1.1.7. Definiciones de bueno y malo

Como parte del desarrollo de un sistema de credit scoring de aplicación, se necesita decidir cómo definir bueno y malo. Definir un caso como malo no necesariamente significa que los otros casos sean buenos. Frecuentemente en desarrollo de sistemas de scoring, al menos otros dos tipos de casos son identificados. El primero puede ser llamado "indeterminados", los casos que no son buenos ni malos. Los segundos pueden ser llamados "experiencia insuficiente".

Por ejemplo, en el desarrollo de un scorecard para un portafolio de tarjetas de crédito, una definición común de malo es un caso que en algún punto llega a tener tres pagos de atraso. Esto frecuentemente se conoce como "ever 3+ down"o "worst 3+ down". Los casos indeterminados pueden ser aquellos que tienen un estatus worst-ever de dos pagos atrasados. Así ellos han causado algunos problemas y algunas actividades adicionales de recaudación, quizás repetidamente si ellos han tenido dos pagos atrasados en algunas ocaciones, pero nunca han llegado a tener tres pagos atrasados. Entonces, podríamos identificar aquellos donde tenemos insuficiente experiencia. Suponga que tenemos una muestra de aplicaciones hechas hace 12 meses y un punto de observación un años después, así que los casos tienen 24 meses de exposición. Entonces podríamos etiquetar como insuficiente experiencia aquellos con tres o menos meses de actividad de ventas o avance de efectivo. En otras palabras, la cuenta no es mala, pero ha sido utilizada infrecuentemente y sería prematuro decir si es buena. Los restantes serían categorizados como cuentas buenas.

Esta clasificación solo es un ejemplo. Muchas variaciones son posibles. Uno podría definir como malo a un caso ever 3+ down o 2 down. Podríamos definir como "insuficiente experiencia" a aquellos que tienen menos de 6 meses de actividad. Podríamos incluir en el grupo indeterminado aquellos casos que han fallado en un pago.

Cualquier definición que se elija, no afecta a la metodología de scorecard.

(Esto asume que las definiciones crean una partición, es decir, todos los casos caen en exactamente una clase). Normalmente se descarta los indeterminados y los de insuficiente experiencia para construir el scorecard solo con los buenos y malos. Por supuesto, como se defina los buenos y malos se tendrá claramente un efecto en el resultado del desarrollo del scorecard. Definiciones diferentes pueden crear scorecards diferentes. Estas diferencias pueden ser en el peso de los scores o en las características que están en el scorecard. Sin embargo, esto no significa que los resultados serán muy diferentes. En realidad, diferentes definiciones de buenos y malos pueden generar diferentes scorecards pero resultan similares los casos que son aceptados y rechazados.

1.1.8. Problemas con la metodología

La mayoría de los problemas encontrados en credit scoring son de naturaleza técnica y no de naturaleza teórica. Se debe tener los datos necesarios para implementar el modelo de toma de decisiones, es decir, se tiene que incluir el número necesario de factores relevantes. Las instituciones financieras recolectan información por medio de aplicaciones pasadas, questionarios y entrevistas con los aplicantes. La información de un aplicante que usualmente se recolecta es la edad, género, estado civil, nacionalidad, educación, número de hijos, trabajo, ingreso, etc. Las variables que definen el modelo de decisión deben ser seleccionadas con especial cuidado para que no causen problemas computacionales. Por ejemplo, si en una aplicación en particular se dispone de una gran cantidad de variables categóricas al momento de generar las variables dummy correspondientes se obtiene una matriz de información con millones de elementos, pudiendo causar un problema muy serio.

Uno de los mayores problemas en el desarrollo de modelos de scoring de aplicación es que solo los casos que fueron aprobados en el pasado tiene datos de desempeño, lo que les permite ser clasificados como buenos o malos. Para los casos que fueron rechazados en el pasado, solo se dispone de los valores de sus características pero no su estado de bueno o malo. Si estos clientes son ignorados y descartados de la muestra, entonces no refleja la población verdadera que solicitará en el futuro un crédito. Esto causa sesgo en cualquier procedimiento de clasificación construido con esta muestra. Algunas técnicas han sido propuestas para superar este sesgo, y estas vienen bajo el nombre de “reject inference” y que se exponen en general en el Apéndice.

Las siguientes preguntas son cuán grande debe ser la muestra y cuál debería ser la división entre el número de buenos y malos en la muestra. Debería haber igual

número de buenos y malos en la muestra o debería la muestra reflejar la relación de buenos y malos en la población como un todo? Normalmente, lo último está fuertemente orientado a los buenos (digamos 20:1) que manteniendo la misma relación en la muestra significaría que no puede haber suficiente de la subpoblación de malos para identificar sus características. Por esta razón, la muestra tiende a ser 50:50 o algo como 50:50 y la verdadera proporción poblacional de buenos y malos. Si la distribución de buenos y malos en la muestra no es la misma distribución que en toda la población, entonces, se necesita ajustar los resultados obtenidos de la muestra. En los enfoques de regresión, esto se hace automáticamente puesto que las proporciones de buenos y malos en la verdadera oblación, p_G, p_B, se utilizan en los cálculos. En otros enfoques, se tiene que hacer a posteriori.

Para el tamaño de la muestra se ha sugerido que 1500 buenos y 1500 malos son los adecuados. En la práctica, muestras de tamaños muy superiores son utilizadas.

La dificultad real surge cuando muestras de tamaños relativamente grandes no pueden ser obtenidas. Es sensible poner aplicantes a diferentes productos o aplicantes a diferentes prestamistas juntos para tener una muestra grande? Con esto se deber tener cuidado.

Por otra parte, la decisión de las variables que deberían ser utilizadas en el sistema de scoring no es un inconveniente en la construcción de sistemas de scoring; el inconveniente surge cuando nos preguntamos cómo deberíamos utilizar a las variables.

Es de esperar aplicar las metodologías descritas en el Capítulo (3) a las variables que describen las características de los aplicantes. En la metodología de scorecard este no es el caso. Primero, uno necesita tomar cada característica y dividir las posibles respuestas en un número pequeño de clases, lo que se conoce con el nombre de "coarse classify the characteristic". Esto se debe hacer por dos diferentes razones, dependiendo si la variables es categórica (tiene un conjunto de respuestas discreto) o continua (tiene un conjunto infinito de posibles respuestas). Para las características categóricas, la razón es que puede haber demasiadas respuestas o atributos, así que no podría haber suficiente en la muestra con una respuesta particular para hacer el análisis robusto. Para características continuas, la razón es que credit scoring busca predecir riesgo mas que explicarlo, y así uno preferiría finalizar con un sistema en el cual el riesgo es no lineal en las variables continuas si eso una mejor predicción.

El desarrollo de los siguientes capítulos tienen un enfoque diferente a la metodología de credit scoring basada en un scorecard y su principal objetivo es presentar una alternativa para la construcción de sistemas de credit scoring. El principal motivo

para explorar una metodología diferente a la tradicional en el ambiente financiero es el último problema de la metodología presentado en el párrafo anterior. Puesto que considero que al categorizar las variables continuas se está, primero, perdiendo información del desempeño de los clientes y, segundo, limitando las herramientas de clasificación a solo aquellas que pueden lidiar con variables categóricas.

Hay que mencionar también que este enfoque de categorizar las variables continuas, desde mi punto de vista, se debe principalmente para obtener tablas o cuadros similares al Cuadro (1.1), para que los scorecard sean fácilmente interpretados. El enfoque propuesto en las siguienetes secciones no necesariamente terminará en tablas o cuadros como los mencionados y por tanto no necesitarán la definición de una marca, sino, mas bien, está enfocado para la creación de sistemas automáticos de credit scoring que pueden lidiar con modelos de estructura más compleja que un simple scorecard.

Capítulo 2

Minería de Datos y Credit scoring

El proceso de credit scoring puede ser abordado como un proceso de minería de datos que consiste de tres etapas. La Figura (2.1) muestra el proceso de minería de tres etapas para el desarrollo de credit scoring. Durante este proceso varias intervenciones humanas son necesarias con el afán de conseguir el modelo de decisión óptimo. En la primera etapa, expertos en minería de datos y expertos en el proceso crediticio de la institución deben cooperar para, definir el problema que debe ser resuelto con los modelos de clasificación y, para recolectar y preparar los datos relevantes. En la tercera etapa, luego de que los modelos scoring han sido construidos, se los debe utilizar en procesos de decisión prácticos. El desempeño de los modelos deben ser observados con el propósito de reconstruirlos si es necesario. En la segunda etapa, donde se construyen

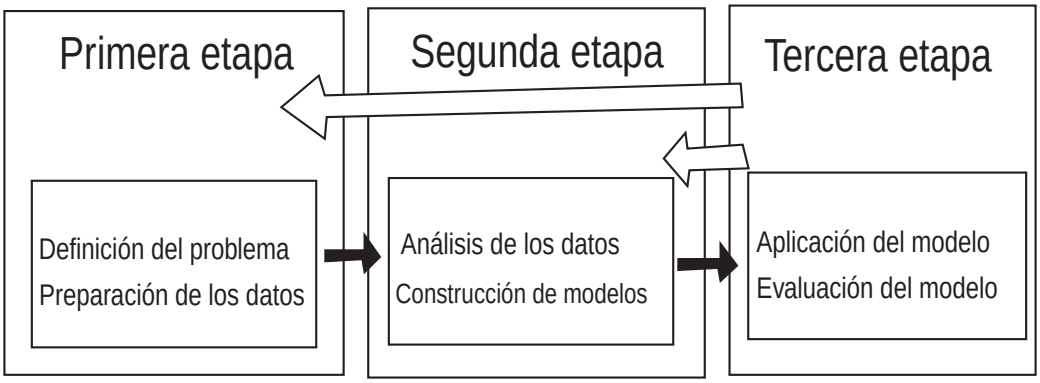


Figura 2.1: Minería de datos aplicada a credit scoring

los modelos, para conseguir resultados fidedignos se debe utilizar un proceso de varios pasos, entre los cuales constan los siguientes. Se debe, seleccionar el tamaño adecuado del conjunto de entrenamiento; decidir la técnica o técnicas de clasificación a ser

ejecutadas; seleccionar y calibrar los parámetros de los modelos; etc. Y finalmente, decidir la técnica más conveniente.

Para buscar el mejor modelo de decisión, los modelos creados en este proceso deben ser evaluados; pero antes de presentar los métodos de evaluación, a continuación se introduce dos conceptos de vital importancia en data mining.

2.1. Muestras de entrenamiento y de prueba

En la segunda etapa del proceso de minería de datos es necesario dividir el conjunto global de datos disponibles en por lo menos dos conjuntos independientes denominados muestra de entrenamiento y muestra de prueba. Con la muestra de entrenamiento se obtienen los modelos de decisión y con la muestra de prueba se evalúan tales modelos. Se desea que el número de datos disponibles sea lo suficientemente grande para obtener modelos de decisión y porcentajes de error confiables. Tradicionalmente, un porcentaje fijo de objetos se utilizan para conformar la muestra de prueba y los restantes forman la muestra de entrenamiento. Con objetos insuficientes, los modelos no podrían ser construidos efectivamente, así que la mayoría de objetos se utilizan en la muestra de prueba. Las proporciones usuales son aproximadamente de 2/3 y de 1/3.

2.2. Razones para evaluar los modelos de scoring

La metodología de credit scoring es una aplicación de minería de datos en algo complicada. En la etapa donde se construyen los modelos, se dispone de varias técnicas de clasificación, algunas de ellas detalladas en el Capítulo (3). Para cada técnica se puede seleccionar varios valores para sus parámetros, una cantidad diferente de variables predictoras. Diferentes selecciones producen varios modelos de decisión. Por ejemplo, se puede comparar los modelos de decisión generados por diferentes técnicas de clasificación pero con el mismo conjunto de entrenamiento. Entonces, se debe evaluar los modelos de decisión para saber como afectan las diferentes selecciones a su desempeño. El objetivo de la evaluación es diferente en cada situación: encontrar los parámetros óptimos, la técnica óptima, etc.

2.3. Evaluación de los modelos de decisión

En credit scoring existen dos grupos que deben ser discriminados, los "buenos" y los "malos" pagadores. Los modelos de decisión ajustados en la etapa de construcción de modelos cometerán errores de clasificación, es decir, a algunos "buenos" pagadores los clasificarán como "malos" pagadores, y viceversa. La evaluación de los modelos nos indicará el modelo que se equivoque lo menos posible.

2.3.1. Exactitud de las clasificaciones

Para evaluar los modelos de decisión el camino más directo es evaluar cuantitativamente la exactitud de las clasificaciones obtenidas con estos modelos. El criterio equivalente es la porcentaje de clasificaciones incorrectas definido como

$$\text{porcentaje de error} = \frac{\text{Número de clasificaciones incorrectas}}{\text{Número total de casos}}.$$

Se debe recalcar que en aplicaciones prácticas de modelos de credit scoring no existe un modelo de decisión perfecto con porcentaje de error cero.

El porcentaje de error verdadero de un modelo es el error obtenido cuando el modelo de decisión actua sobre la distribución verdadera de los objetos de la población en consideración, es decir, sobre todos aquellos que necesitan un crédito. Pero por las observaciones echas en la sección (1.1.8) y porque no se puede conocer a todos los objetos de la población el instante mismo que se arma un modelo de decisión no se podrá obtener el valor verdadero del porcentaje de error. Entonces, necesariamente este porcentaje de error debe ser estimado.

El porcentaje de error de un modelo en la muestra de entrenamiento se denomina porcentaje de error aparente y en la muestra de prueba se denomina porcentaje de error en la muestra de prueba. Está comprobado que el porcentaje de error aparente no es un estimador confiable del porcentaje de error verdadero, pero por ventaja, el porcentaje de error en la muestra de prueba es un estimador adecuado del porcentaje de error verdadero [7]. Por tanto, el porcentaje de error en la muestra de prueba es utilizado como estimador del porcentaje de error verdadero. La comparación estadística del desempeño de dos modelos de decisión se da en el la sección (6.2).

La adecuada estimación del porcentaje de error supone que los objetos en la muestra de prueba son representativos de la población real futura. Para problemas de credit scoring reales, la población cambia con el tiempo. Así, el porcentaje de error en la muestra de prueba puede resultar un estimador no adecuado del porcentaje de

error verdadero para la población cambiante futura. Para superar este inconveniente se propone seleccionar la muestra de prueba de la siguiente manera. Si los objetos en la población tienen un atributo que permita ordenarlos por fecha de aplicación, la muestra de entrenamiento se toma en un periodo de tiempo anterior a una fecha determinada y la muestra de prueba se toma luego de esta fecha. La longitud de los dos periodos de tiempo depende del conocimiento de la institución.

Matriz de confusión

La matriz de confusión proporciona una comparación conveniente de las frecuencias de los riesgos reales y pronosticados en la muestra de prueba. El formato

	Riesgo actual			
	número		porcentaje	
Riesgo pronosticado	malo	bueno	malo	bueno
malo	a	b		$e_1 = \frac{c}{a+c}$
bueno	c	d	$e_2 = \frac{b}{b+d}$	

Cuadro 2.1: Matriz de confusión

de una matriz de confusión se muestra en el Cuadro (2.1), donde a es el número de "malos" individuos pronosticados correctamente, b el número de "buenos" individuos pronosticados incorrectamente, c el número de "malos" individuos pronosticados incorrectamente, a es el número de "malos" individuos pronosticados correctamente, d el número de "buenos" individuos pronosticados correctamente. Con esta notación tenemos que

$$\text{porcentaje de error} = \frac{b+c}{a+b+c+d}.$$

En problemas de credit scoring, dos tipos de porcentaje de errores deben ser considerados, el porcentaje de error de tipo I

$$e_1 = \frac{c}{a+c},$$

y el porcentaje de error tipo II

$$e_2 = \frac{b}{b+d}.$$

El error de tipo I y el error de tipo II también se denominan riesgo de crédito y riesgo comercial, respectivamente. Cuando la política de créditos es muy generosa, alto e_1, la institución está expuesta a riesgo de crédito. Cuando sucede e_2, la institución tiene un costo de oportunidad por la pérdida de "buenos" clientes.

Cuando se utiliza el porcentaje de error como medida de desempeño de un modelo de decisión, implícitamente se está suponiendo que los errores de tipo I y II tienen la misma importancia. En problemas de credit scoring, obviamente, éstos errores tienen diferente importancia puesto que la correcta predicción de "malos" clientes es más importante debido a los altos costos asociados. En este caso, el porcentaje de error no es un criterio apropiado para medir el desempeño de un modelo de decisión.

Curva ROC

Para problemas de clasificación con solamente dos grupos hay una importante técnica gráfica para evaluar el desempeño de las técnicas de clasificación que producen scores o puntuaciones numéricas, denominada curva ROC. Las iniciales ROC provienen de la técnica "relative operating characteristic" que se utiliza en detección de señales, otras veces conocida como "receiver operating characteristic".

La idea central de esta técnica gráfica es el umbral o cutoff level. Este umbral es un valor numérico que permite clasificar a los individuos como buenos y malos pagadores utilizando solamente los scores obtenidos mediante un modelo de clasificación. Por ejemplo, puede suceder que scores de valoración alta estén asociados con los malos pagadores, entonces la técnica decide que aquellos individuos con scores superiores a un determinado umbral se los clasifica como malos pagadores y a los restantes como buenos pagadores.

Suponga que los "buenos" y "malos" pagadores son evaluados por la regla de decisión determinada por un modelo de clasificación y que se obtienen resultados numéricos, denominados scores, para cada individuo. Las densidades hipotéticas correspondiente a estos scores para cada grupo se muestran en la Figura (2.2), en la cual se supone que valores altos de los valores numéricos corresponden a los malos pagadores y valores bajos a los buenos pagadores.

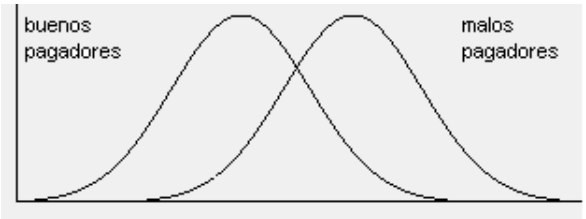


Figura 2.2: Densidades hipotéticas de los scores de los buenos y malos pagadores

Al considerar un umbral y las densidades de los buenos y malos pagadores

podemos observar gráficamente las probabilidades e_1 y e_2 correspondientes a los errores tipo I y II en la Figura (2.3).

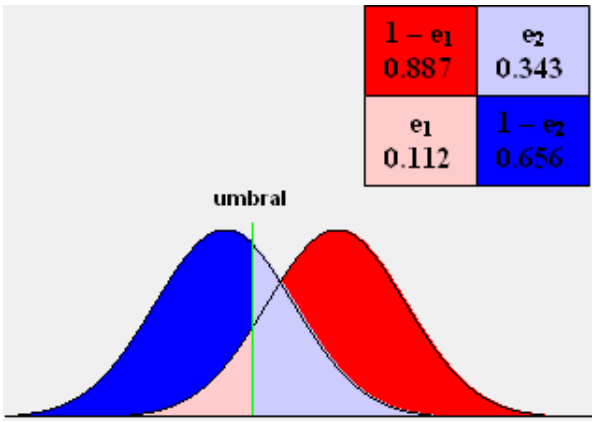


Figura 2.3: Umbral y y las probabilidades de error I y II

La curva ROC es una exploración de lo que sucede con $1 - e_1$ y e_2 mientras el umbral cambia. Luego de fijado el umbral, se puede obtener una estimación de los errores de tipo I y II. La curva ROC se obtiene graficando los diferentes valores de los errores del tipo I y II que se obtienen luego de variar el umbral. Una curva ROC hipotética se muestra en la Figura (2.4).

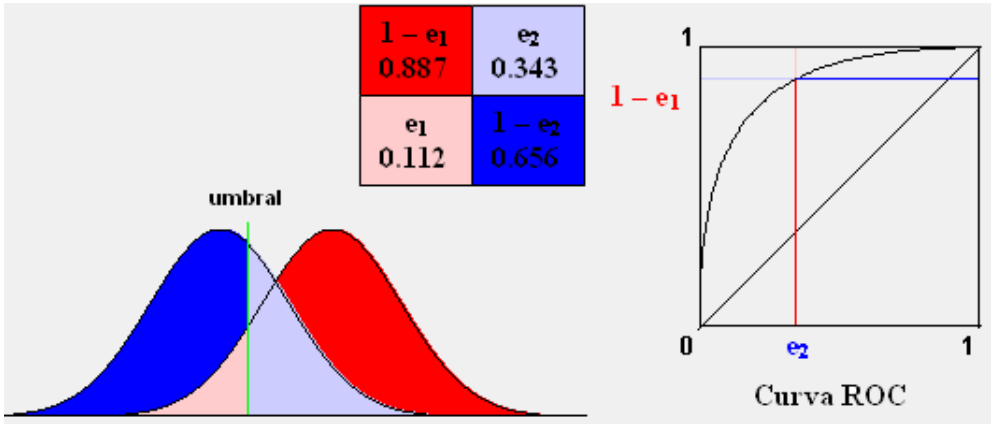
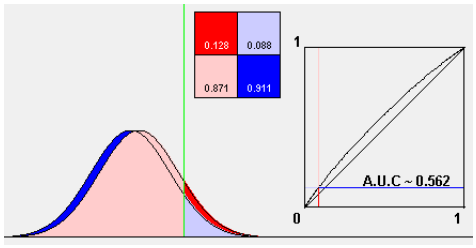
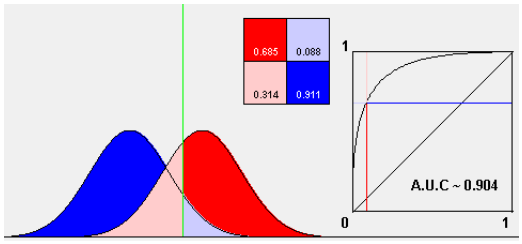


Figura 2.4: Curva ROC hipotética

En las Figuras (2.5(a)) y (2.5(b)) se puede observar el comportamiento de la curva ROC cuando la población de los buenos y malos pagadores son similares y diferentes. Cuando tienden a ser similares la curva ROC se comporta como una recta que pasa por el orígen con pendiente 1, y cuando tienden a ser diferentes la curva ROC



(a) Curva ROC para dos poblaciones similares



(b) Curva ROC para dos poblaciones diferentes

Figura 2.5: Comparación de curvas ROC

se aleja significativamente de la recta con pendiente 1. Entonces, este análisis gráfico no lleva a deducir que un clasificador es:

- **inútil** cuando no ha logrado separar a los buenos y malos pagadores produciendo una curva ROC que es una recta que pasa por el orígen de pendiente 1;
- **perfecto** cuando ha logrado separar a los buenos y malos pagadores produciendo una curva ROC que inicialmente es vertical y luego horizontal.

En la Figura (2.6) se muestra las curvas ROC correspondiente a un clasificador perfecto, a uno real y a un inútil. Pero la comparación gráfica puede resultar subjetiva al momento de juzgar la calidad de un clasificador. Una forma más precisa de caracterizar la lejanía o cercanía de la curva ROC a la diagonal es el área bajo la curva ROC (AUC). Para un clasificador inútil y perfecto es aproximadamente 0.5 y 1, respectivamente.

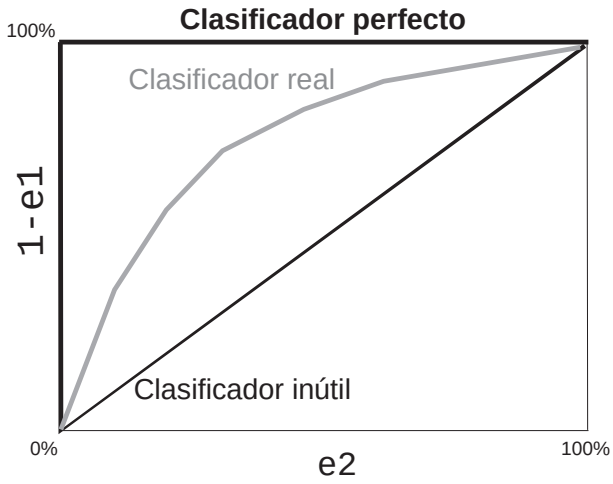


Figura 2.6: Ejemplo de una curva ROC

Mediante la comparación de curvas ROC se puede estudiar la capacidad de clasificación de dos clasificadores. Para un e_1 fijo, la curva que tenga el menor valor de e_2 tendrá una mayor capacidad de clasificación. Para un e_2 fijo, la curva con un mayor valor de $1 - e_1$ será la de mejor capacidad de clasificación. En resumen, el clasificador para el cual su curva ROC se encuentra más cerca del clasificador perfecto es la de mayor capacidad predictiva. Algunas veces, la gráfica de la curva ROC para un clasificador es superior a la de otro clasificador para todos los umbrales, en otras ocasiones, esto sucede solamente en ciertos intervalos del rango de los scores. Para este último caso, se tiene que fijar la atención en la curva que se encuentra más cerca del clasificador perfecto en el rango de valores que corresponde más fielmente al problema en cuestión.

Función de costo esperado

Un modelo de decisión óptimo es aquel que minimiza el costo esperado al clasificar a los aplicantes para un crédito. Cuando la discriminación es entre "buenos" y "malos" pagadores, la función de costo esperado es

$$C_e = \pi_m C_I \alpha + \pi_b C_{II} \alpha,$$

donde π_m, π_b, C_I y C_{II} es la proporción de "malos" pagadores, "buenos" pagadores, costo del error del tipo I y costo del error del tipo II, respectivamente.

Los supuestos tras esta función de costo son los siguientes. El costo de construir modelos de clasificación asignado a cada objeto es el mismo, y sin pérdida de generalidad, se considera que es cero, los objetos correctamente clasificados no incurren en costos adicionales y no hay diferencia en los costos individuales de los objetos mal clasificados. El último supuesto es usualmente violado en la realidad, sin embargo, la diferencia de los costos individuales puede ser ignorada puesto que los modelos de credit scoring se construyen separadamente para diferentes mercados de crédito. Por tanto, aunque el último supuesto no es del todo cierto en la realidad, es adoptado por investigadores y practicantes.

La elección de C_I y C_{II} tiene un efecto significativo en la evaluación de un modelo de decisión. Sin embargo, el cálculo de estos costos es de una complicación considerable, puesto que los factores que los afectan son de difícil cuanficación.

Por ejemplo, en un problema de credit scoring, C_I es el costo de otorgar un crédito a un "mal" pagador y C_{II} es el costo de oportunidad por rechazar una

aplicación de un "buen" pagador. Algunos componentes de C_I y C_{II} se muestran en el Cuadro (2.2).

C_I	+ pérdida del monto del crédito
	- algún ingreso recibido durante el proceso, como interés
	- valor de la propiedad asegurada al momento de la liquidación
	+ costos legales, costos administrativos, etc
C_{II}	+ pérdida de los intereses del "buen" pagador
	+ pérdida / - beneficios en aplicantes alternativos.

Cuadro 2.2: Componentes de los costos tipo I y II

El segundo componente de C_{II} se basa en el supuesto de que el prestamista no se quedará con el dinero que iba a prestar y que será prestado a otro aplicante. La ganancia y la pérdida en el aplicante alternativo disminuye y aumenta C_{II}, respectivamente.

El cálculo práctico de costos de mala clasificación puede ser más complicado. En la práctica, la administración y la decisión de otorgar créditos es un proceso complejo, en el cual existen varios factores que afectan los costos de mala clasificación. Algunos ejemplos complejos que pueden suceder son

- Cuando un aplicante es rechazado por el modelo de credit scoring, es reexaminado por analistas de crédito y el crédito es otorgado al aplicante, tal vez con nuevas condiciones. El costo de reexaminación y la probabilidad de la correcta decisión por el analista de crédito debería ser considerada en la función de costo esperado.

- Cuando un cliente no paga la deuda a tiempo, pero es finalmente pagada luego de una gestión de cobranza. El costo de la gestión de cobranza deberá ser considerado.

Hay otras situaciones complejas que añaden dificultad y incertidumbre al cálculo de los costos de mala clasificación. Debido a esto, usualmente las instituciones que otorgan crédito no pueden dar estimaciones precisas de estos costos. Además, estos costos probablemente cambien con el tiempo conforme cambia la economía. Por tanto, aunque es posible dar un rango en el cual probablemente estarán los costos, es usualmente improbable que los costos exactos sean calculados.

2.3.2. Criterios prácticos

Cuando se tiene que elegir entre varias alternativas un modelo de decisión como el modelo de credit scoring final, el criterio adecuado es la exactitud de las

clasificaciones; puesto que es una medida básica y directa del desempeño de un modelo. Este mejor modelo es aquel que puede predecir con precisión el grupo al que pertenece un nuevo objeto.

A pesar de esto, la exactitud de las clasificaciones no es el único criterio de selección en problemas de credit scoring. El modelo seleccionado debe ser práctico, de acuerdo a los siguientes criterios.

- El tiempo que se demora el modelo de decisión en decidir si una nueva aplicación se considera de riesgo aceptable o no es de crucial importancia.
- La interpretación del modelo de decisión es importante cuando tiene que ser expuesto al analista de crédito o cuando se tiene que explicar al cliente el porque de su aplicación rechazada.
- La simplicidad del modelo elegido es importante también. Si un modelo es simple, entonces, algunas veces es de fácil comprensión y de aplicación rápida.

En algunas situaciones, cuando el modelo de toma de decisiones proviene de una técnica de clasificación, la regla generada debe ser razonablemente justificada, y la justificación debe ser transparente e interpretable para el prestamista y el aplicante. En otras ocaciones, la velocidad y precisión de las decisiones son más importantes que la interpretabilidad.

Capítulo 3

Técnicas de Clasificación

En este capítulo se da una explicación detallada de las técnicas de clasificación frecuentemente utilizadas en credit scoring como son, análisis discriminante, regresión logística, árboles de clasificación, redes neuronales, y una considerada en esta tesis que no es muy frecuentemente encontrada en la literatura, denominada support vector machines. La presentación de cada una de estas técnicas es totalmente independiente, lo que permite concentrarse exclusivamente en la que sea de interes. Luego de haber cubierto este capítulo se estará en capacidad de reconocer los supuestos, las debilidades y fortalezas de cada una de las técnicas para credit scoring.

3.1. Análisis discriminante

El problema considerado por el análisis discriminante es el siguiente. Dado que un objeto pertenece a uno de g grupos distintos G_i, $i = 1, \ldots, g$, en una población $\mathbb{P}$, se desea asignar este objeto a uno de los g grupos utilizando d características, $X = (X_1, \ldots, X_d)$, asociadas con el objeto. La asignación debe ser óptima en algún sentido tal como minimizar el número de errores o el costo de cometer errores de clasificación.

Como $X \in \mathbb{R}^p$, se desea encontrar una partición $\{R_1, \ldots, R_g\}$ del espacio muestral $\mathbb{R}^p$ tal que un miembro de $\mathbb{P}$ sea asignado al grupo G_i si $X \in R_i$. Para esto, sea $f_i(X)$ la función de densidad de X dado que $X \in G_i$, entonces, la probabilidad de asignar a un miembro de $\mathbb{P}$ al grupo G_j cuando en realidad pertenece al grupo G_i es

$$P(j|i) = \int_{R_j} f_i(X)dX,$$

y la probabilidad de clasificar erróneamente un miembro de G_i es

$$P(i) = \sum_{j=1, j\neq i}^{g} P(j|i) = 1 - P(i|i).$$

Si π_i es la proporción de $\mathbb{P}$ en el grupo G_i, $\sum_i \pi_i = 1$, la probabilidad de clasificar incorrectamente es

$$P(CI) = \sum_{i=1}^{g} \pi_i P(i) = 1 - \sum_{i=1}^{g} \pi_i P(i|i).$$

Seber [11] muestra que $P(CI)$ es minimizada cuando $R_i = \{X : \pi_i f_i(X) \geq \pi_j f_j(X), j = 1, 2, \ldots, g\}$. Entonces, la regla de asignación óptima es,
asignar un miembro de $\mathbb{P}$ al grupo G_i si

$$\frac{f_i(X)}{f_j(X)} \geq \frac{\pi_j}{\pi_i}, \quad j = 1, 2, \ldots, g. \tag{3.1}$$

Poblaciones normales de igual varianza

Supongamos que las d características de cada grupo provienen de una distribución normal multivariada de vector de medias μ_i y matriz de varianza-covarianza Σ_i (la prueba de este supuesto se da en la Sección (6.3) del Apéndice), entonces

$$f_i(X) = (2\pi)^{-d/2} |\Sigma_i|^{-1/2} \exp\left\{ -\frac{1}{2}(X - \mu_i)^t \Sigma_i^{-1} (X - \mu_i) \right\}.$$

El análisis discriminante lineal surge cuando se supone que todas las matrices de varianza-covarianza son iguales, es decir, cuando $\Sigma_i = \Sigma, \forall i$ (la prueba de este supuesto se da en la Sección (6.7) del Apéndice). En este caso se tiene que

$$\begin{aligned} \frac{f_i(X)}{f_j(X)} &= \exp\left[-\tfrac{1}{2}(X - \mu_i)^t \Sigma^{-1} (X - \mu_i) + \tfrac{1}{2}(X - \mu_j)^t \Sigma^{-1} (X - \mu_j) \right] \\ &= \exp\left[X^t \Sigma^{-1} (\mu_i - \mu_j) - \tfrac{1}{2}(\mu_i + \mu_j)^t \Sigma^{-1} (\mu_i - \mu_j) \right] \\ &= \exp\left[\left\{ X^t - \tfrac{1}{2}(\mu_i + \mu_j)^t \right\} \Sigma^{-1} (\mu_i - \mu_j) \right]. \end{aligned}$$

Tomando logaritmo en (3.1), la regla para poblaciones normales es

asignar un miembro de $\mathbb{P}$ al grupo G_i si

$$\left[X^t - \frac{1}{2}(\mu_i + \mu_j)^t \right] \Sigma^{-1} (\mu_i - \mu_j) \geq \ln(\pi_j / \pi_i), \quad j = 1, 2, \ldots, g. \tag{3.2}$$

En el lenguaje de credit scoring, el valor $\ln(\pi_j/\pi_i)$ se conoce con el nombre de cuttof score. Reordenando la ecuación (3.2) se obtienen las funciones discriminantes lineales

$$\delta_j(X) = X^t \Sigma^{-1} \mu_j - \frac{1}{2} \mu_j^t \Sigma^{-1} \mu_j + \ln(\pi_j), \quad j = 1, 2, \ldots, g.$$

Con estas funciones discriminantes lineales la regla para poblaciones normales dada en (3.2) se convierte en asignar un miembro de $\mathbb{P}$ al grupo que corresponda a

$$\boxed{\begin{array}{c}\text{asignar un miembro de } \mathbb{P} \text{ al grupo que corresponda a} \\ G(X) = \arg\max_j \delta_j(X), \quad j = 1, 2, \ldots, g.\end{array}} \tag{3.3}$$

Además, utilizando estas funciones discriminantes lineales, la frontera entre dos grupos diferentes es descrita por la ecuación lineal $\{X : \delta_i(X) = \delta_j(X)\}$.

En el caso de que las características X de los objetos en los g grupos provengan de poblaciones normales, se pueden tener cálculos explícitos de las probabilidades de clasificación incorrectas $P(j|i)$. Dichos cálculos se presentan en la Sección (6.8) del Apéndice.

Cuando solamente hay dos grupos, G_1 y G_2, y $\pi_1 = \pi_2 = 1/2$, asignamos el miembro con características X al grupo G_1 si

$$X^t\Sigma^{-1}(\mu_1 - \mu_2) \geq \frac{1}{2}(\mu_1 + \mu_2)^t\Sigma^{-1}(\mu_1 - \mu_2), \tag{3.4}$$

con $P(1|2) = P(2|1) = \Phi(-\Delta/2)$. Esta regla de asignación fue encontrada por Fisher sin hacer suposiciones sobre la distribución de los grupos.

Poblaciones normales de varianza diferente

Ahora supongamos que las d características de cada grupo provienen de una distribución normal multivariada de vector de medias μ_i y matriz de varianza-covarianza Σ_i, con $\Sigma_i \neq \Sigma_j$ para $i \neq j$. En este caso tenemos que

$$\frac{f_i(X)}{f_j(X)} = \left(\frac{|\Sigma_i|}{|\Sigma_j|}\right)^{-1/2} \exp\left\{-\frac{1}{2}(X - \mu_i)^t\Sigma_i^{-1}(X - \mu_i) + \frac{1}{2}(X - \mu_j)^t\Sigma_j^{-1}(X - \mu_j)\right\}$$

Luego tomando logaritmo en (3.1) y definiendo las funciones discriminantes cuadráticas

$$\delta_i(X) = -\frac{1}{2}\ln|\Sigma_i| - \frac{1}{2}(X - \mu_i)^t\Sigma_i^{-1}(X - \mu_i) + \ln(\pi_i),$$

la regla para poblaciones normales (3.1) se convierte en

$$\boxed{\begin{array}{c}\text{asignar un miembro de } \mathbb{P} \text{ al grupo que corresponda a} \\ G(X) = \arg\max_j \delta_j(X), \quad j = 1, 2, \ldots, g.\end{array}} \tag{3.5}$$

Además, utilizando estas funciones discriminantes cuadráticas, la frontera entre dos grupos diferentes es descrita por la ecuación cuadrática $\{X : \delta_i(X) = \delta_j(X)\}$.

Costo esperado de clasificación erronea

La regla de asignación óptima (3.1) se obtuvo luego de minimizar la probabilidad de asignamientos erróneos. También, se puede obtener otras reglas de asignamiento equivalentes con (3.1) mediante la razón de verosimilitud, la maximización de las probabilidades posteriores y la minimización del costo de clasificación erroneo. Sin embargo, la maximización de probabilidades posteriores es la técnica común porque proporciona un buen preámbulo del análisis logístico que se presenta en la Sección (3.3).

Para minimizar el costo de clasificación erróneo, sea $C(j|i)$ el costo de clasificar un miembro de $\mathbb{P}$ como perteneciente al grupo G_j dado que pertenece al grupo G_i, entonces, el costo esperado de asignar erróneamente un miembro del grupo G_i es

$$C(i) = \sum_{j=1, j\neq i}^{g} C(j|i)P(j|i),$$

y el costo esperado de clasificación errónea es

$$C_T = \sum_{i=1}^{g} \pi_i C(i).$$

Seber [11] muestra que C_T es minimizado cuando

$$R_i = \{X : C(j|i)\pi_i f_i(X) \geq C(i|j)\pi_j f_j(X), j = 1, 2, \ldots, g\}.$$

Entonces, la regla de asignación óptima es

asignar un miembro de $\mathbb{P}$ al grupo G_i si

$$\frac{f_i(X)}{f_j(X)} \geq \frac{C(i|j)\pi_j}{C(j|i)\pi_i}, \quad j = 1, 2, \ldots, g \tag{3.6}$$

Selección de variables en el análisis discriminante

Seber [11] menciona que a pesar de que las probabilidades de clasificación errónea disminuirán cuando d aumente, la precisión de las estimaciones y la robustes de las funciones discriminantes disminuirá. Entonces, se puede buscar un subconjunto $X^{(1)}$ de k de las d variables en X que discrimine tan bien como lo hacen las d variables. Para datos normales multivariados con matriz de varianza-covarianza Σ se puede utilizar pruebas debidas a Rao. Sea $\delta = (\mu_1 - \mu_2)$ y $\delta_1 = (\mu_1^{(1)} - \mu_2^{(1)})$. Si $\Delta_d^2 = \delta'\Sigma^{-1}\delta$ y $\Delta_k^2 = \delta_1'\Sigma_{11}^{-1}\delta_1$ son las distancias de Mahalanobis al cuadrado para d y k $(k < d)$

variables, respectivamente, entonces la prueba de hipótesis $H_0 : \Delta_k^2 = \Delta_d^2$ está dada por el estadístico

$$F = \frac{n_1 + n_2 - d - 1}{d - k} \left\{ \frac{D_d^2 - D_k^2}{c + D_k^2} \right\}, \tag{3.7}$$

donde $c = (n_1 + n_2)(n_1 + n_2 - 2)/n_1 n_2$, D_d^2 y D_k^2 son las distancias de Mahalanobis muestrales, por ejemplo, $D_d^2 = (\overline{X}_{1.} - \overline{X}_{2.})' S_p^{-1} (\overline{X}_{1.} - \overline{X}_{2.})$. Cuando H_0 es verdad, $F \sim F_{d-k, n_1+n_2-d-1}$.

Como las funciones discriminantes lineales pueden tomar la forma de $\alpha + \beta' X$, entonces $\beta' X = \beta_1' X^{(1)} + \beta_2' X^{(2)}$. Así, la prueba H_0 es equivalente a probar $\beta_2 = 0$ y las técnicas se selección de variables aplicadas a la regresión pueden utilizarce. En el procedimiento de incluir una variable o descartarla, el estadístico (3.7) prueba $H_0 : \Delta_k^2 = \Delta_{k+1}^2$. En cualquier etapa la variable con el estadístico F más grande se añade al subconjunto actual de variables si su valor F supera al valor F_{IN}, un umbral específico. Luego de que una variable ha sido añadida, todas las variables en el subconjunto deben ser reexaminadas y aquella con valor F más pequeño debe ser eliminada del subconjunto si su valor F es menor que F_{OUT}, un segundo umbral. Se sugiere utilizar niveles de significancia en el intervalo $0{,}10 \leq \alpha \leq 0{,}25$ puesto que los convencionales ($\alpha \leq 0{,}10$) detienen el proceso prematuramente.

Análisis discriminante en la práctica

En la práctica no se conoce los parámetros verdaderos θ_i de las funciones de densidad $f_i(X, \theta_i)$ de las características de los grupos, entonces, necesariamente se los debe estimar utilizando un estimador óptimo, como por ejemplo, uno de máxima verosimilitud. Para esto se recurre al conjunto de datos que serán utilizados en la creación de las reglas de clasificación, conjunto que recibe el nombre de datos de entrenamiento. Para el caso de las funciones discriminantes lineales y cuadráticas se tiene que $\widehat{\pi}_i = n_i/n$, donde n_i es el número de observaciones en el grupo G_i; $\widehat{\mu}_i = \frac{1}{n_i} X_j^t 1_{n_i}$, donde 1_{n_i} es el vector de n_i unos y X_j son las características del grupo G_i; $\widehat{\Sigma}_i = \frac{1}{n_j} X_j^t X_j - \widehat{\mu}_i \widehat{\mu}_i^t$, y la varianza común es estimada con $\widehat{\Sigma} = \sum_{j=1}^{g} \left(\frac{n_j - 1}{n - g} \right) \widehat{\Sigma}_j$.

En lo que tiene que ver con el desempeño de las funciones discriminantes, Seber [11] afirma que para datos no normales las funciones discriminantes lineales y las funciones discriminantes cuadráticas no proporcionan reglas de clasificación óptimas y que las funciones discriminantes lineales pueden tener propiedades pobres cuando se utilizan características continuas y discretas. Sin embargo, Johnson [5] afirma que las reglas de clasificación obtenidas por el análisis discriminante con frecuencia se aplican

a datos no normales. Dice que los investigadores tienen la ventaja de ver cuán bien funcionan las funciones discriminantes al utilizarlas. “Si las reglas funcionan bien con datos no normales, entonces no hay razón para preocuparse demasiado por el echo de que los datos sean de esta clase. De hecho, yo he usado este tipo de reglas sobre variables categóricas, mediante la introducción de variables ficticias y utilizando estas últimas en los programas de discriminación”. Al final de la Sección (3.3) también se realizan comentarios respecto a la conveniencia o no de utilizar análisis discriminante.

3.2. Funciones discriminantes canónicas

Un método alternativo de discriminación fue propuesto por Fisher en 1936. Dado que un objeto pertenece a uno de g grupos distintos G_i, $i = 1, \ldots, g$, en una población $\mathbb{P}$, se desea asignar este objeto a uno de los g grupos utilizando d características, $X^t = (X_1, \ldots, X_d)$, asociadas con el objeto. La asignación se considera óptima cuando se encuentra una combinación lineal de las características que produzca la máxima separación de los g grupos y la menor varianza dentro de los mismos.

En la Figura (3.1) se muestra un caso hipotético de dos distribuciones en las que la componente principal no es capaz de distinguirlas, mientras que la función discriminante canónica hace un buen trabajo.

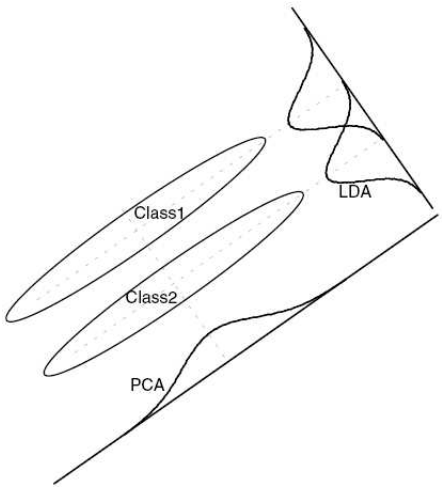


Figura 3.1: Dos poblaciones hipotéticas proyectadas sobre una función discriminante canónica

Para lograr este objetivo, la metodología recurre a la descomposición de la suma de cuadrados de la variable aleatoria X,

$$\begin{aligned}\sum_x (x-\overline{x})(x-\overline{x})^t &= \sum_{j=1}^g \sum_{x\in G_j}(x-\overline{x})(x-\overline{x})^t \\ &= \sum_{j=1}^g \sum_{x\in G_j}(x-\overline{x}_j+\overline{x}_j-\overline{x})(x-\overline{x}_j+\overline{x}_j-\overline{x})^t \\ &= \sum_{j=1}^g \sum_{x\in G_j}(x-\overline{x}_j)(x-\overline{x}_j)^t + \sum_{j=1}^g \sum_{x\in G_j}(\overline{x}_j-\overline{x})(\overline{x}_j-\overline{x})^t \\ &\quad + \sum_{j=1}^g \sum_{x\in G_j}[(x-\overline{x}_j)(\overline{x}_j-\overline{x})^t + (\overline{x}_j-\overline{x})(x-\overline{x}_j)^t],\end{aligned}$$

como,

$$\sum_{x\in G_j}[(x-\overline{x}_j)(\overline{x}_j-\overline{x})^t + (\overline{x}_j-\overline{x})(x-\overline{x}_j)^t] = 0,$$

entonces,

$$\sum_x (x-\overline{x})(x-\overline{x})^t = \sum_{j=1}^g \sum_{x\in G_j}(x-\overline{x}_j)(x-\overline{x}_j)^t + \sum_{j=1}^g n_j(\overline{x}_j-\overline{x})(\overline{x}_j-\overline{x})^t.$$

Las matrices

$$\begin{aligned}S_T &= \textstyle\sum_x (x-\overline{x})(x-\overline{x})^t, \\ S_B &= \textstyle\sum_{j=1}^g \sum_{x\in G_j}(x-\overline{x}_j)(x-\overline{x}_j)^t, y \\ S_W &= \textstyle\sum_{j=1}^g n_j(\overline{x}_j-\overline{x})(\overline{x}_j-\overline{x})^t,\end{aligned}$$

se conocen con el nombre de suma de cuadrados total, dentro de los grupos y entre grupos, respectivamente.

Para la combinación lineal $y = a^t X$, la descomposición de la suma de cuadrados es,

$$a^t S_T a = a^t S_W a + a^t S_B a.$$

Así, Fisher propone que $\widehat{a}$ sea el vector que maximiza la función

$$J(a) = \frac{a^t S_B a}{a^t S_W a}. \tag{3.8}$$

Algunas veces, cuando se tiene dos grupos, $g = 2$, se define a S_B como

$$S'_B = (\overline{x}_1 - \overline{x}_2)(\overline{x}_1 - \overline{x}_2)^t.$$

Se muestra que $S_B = \frac{n_1 n_2}{n} S'_B$, es decir, básicamente se está multiplicando el objetivo por una constante que no afecta al resultado final.

Como $J(\alpha a) = J(a)$, se puede requerir que $a^t S_W a = 1$, lo que modifica el problema de maximización (3.8) a

$$\begin{cases} \min\limits_a & -\frac{1}{2} a^t S_B a \\ \text{s.a.} & a^t S_W a = 1. \end{cases}$$

Para este problema de optimización, el lagragiano es

$$L(a) = -\frac{1}{2}a^t S_B a + \frac{1}{2}\lambda(a^t S_W a - 1),$$

(los valores 1/2 se añaden por conveniencia). Las KKT condiciones implican que la ecuación (3.9) debe darse en la solución,

$$S_B a = \lambda S_W a, \tag{3.9}$$

que puede escribirse como

$$S_W^{-1} S_B a = \lambda a.$$

Esta última ecuación correspondería un problema de valores y vectores propios si la matriz $S_W^{-1} S_B$ fuera simétrica. Sin embargo, como S_B es una matriz simétrica definida positiva, podemos descomponerla como

$$S_B = U \Lambda U^t,$$

para obtener

$$S_B^{\frac{1}{2}} = U \Lambda^{\frac{1}{2}} U^t,$$

lo que nos permite escribir

$$S_B = S_B^{\frac{1}{2}} S_B^{\frac{1}{2}},$$

y así definir

$$b = S_B^{\frac{1}{2}} a,$$

para obtener el problema de valores y vectores propios

$$S_B^{\frac{1}{2}} S_W^{-1} S_B^{\frac{1}{2}} b = \lambda b,$$

puesto que $S_B^{\frac{1}{2}} S_W^{-1} S_B^{\frac{1}{2}}$ es una matriz simétrica definida positiva, para la cual se puede encontrar la solución λ_k y b_k que correspondería a la solución

$$a_k = S_B^{-\frac{1}{2}} b_k.$$

Reemplazando (3.9) en (3.8) se obtiene

$$J(a) = \frac{a^t S_B a}{a^t S_W a} = \lambda_k \frac{a_k^t S_W a_k}{a_k^t S_W a_k} = \lambda_k,$$

así, se debe elegir el valor propio más grande para que (3.8) sea maximizada. Este valor propio se nota como λ_1 y su vector propio asociado se nota por b_1.

La combinación lineal $y_1 = b_1^t X$ es la función discriminante lineal única que proporciona la separación máxima entre los g vectores de medias. En el caso de dos grupos, una sola función discriminante canónica es adecuada; sin embargo, cuando $g > 2$, una sola función en general no es adecuada, a menos que todas las medias de los grupos se encuentren sobre una recta del espacio $\mathbb{R}^d$.

En la primera variable discriminante canónica, la regla de asignación es

$$\boxed{\begin{array}{c}\text{asignar un miembro de } \mathbb{P} \text{ al grupo que corresponda a} \\ G(X) = \arg\max_i |b_1^t X - b_1^t \widehat{\mu}_i|, \quad i = 1, \ldots, g.\end{array}} \tag{3.10}$$

Si se tiene dos grupos, $y_1 = b_1^t X$ es equivalente a la función discriminante lineal (3.4).

Si los vectores de medias de los grupos no caen sobre una recta sino sobre todo un plano se necesitan dos funciones discriminantes canónicas.

Similarmente, se puede encontrar la dirección b_2, ortogonal en S_W a b_1 tal que J es máximo. Esta dirección es el vector propio de $S_B^{\frac{1}{2}} S_W^{-1} S_B^{\frac{1}{2}}$ asociado con el segundo valor propio λ_2 más grande. Ahora, se tiene la función discriminante $y_2 = b_2^t X$ ortogonal a y_1. En el nuevo plano definido por y_1 y y_2 la regla de asignación es

$$\boxed{\begin{array}{c}\text{asignar un miembro de } \mathbb{P} \text{ al grupo que corresponda a} \\ G(X) = \arg\max_i \sqrt{(b_1^t X - b_1^t \widehat{\mu}_i)^2 + (b_2^t X - b_2^t \widehat{\mu}_i)^2}, \quad i = 1, \ldots, g.\end{array}} \tag{3.11}$$

La determinación de la dimensión del espacio canónico, aquel generado por las y_i, se la realiza analizando los $\min(d, g-1)$ valores propios λ_i, de manera similar al análisis de componentes principales. Hay que tener presente que los valores propios λ_i son la proporción de la varianza entre grupos explicada por las combinaciones lineales y_i.

En unos cuantos casos, un investigador puede ser capaz de interpretar las variables canónicas, lo que incrementa su utilidad. Una ventaja que tienen esta funciones, sin importar si se pueden interpretar, es que a menudo permiten que un investigador visualice las distancias reales entre los grupos que se están investigando en un espacio de dimensión reducido.

3.2.1. Funciones discriminantes canónicas en la práctica

En esta sección se muestra la estructura matricial de las matrices S_B y S_W para efectos de cálculo. Sea $\mathbb{X}$ la matriz $n \times d$ de observaciones

$$\mathbb{X} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1d} \\ x_{21} & x_{22} & \dots & x_{2d} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nd} \end{pmatrix},$$

entonces, $\widehat{\mu} = \frac{\mathbf{1}^t \mathbb{X}}{n} = (\widehat{\mu}_1\, \widehat{\mu}_2 \quad \widehat{\mu}_d)$ es el vector fila de medias de las d características observadas, donde $\mathbf{1}$ es el vector $n \times 1$ de unos. Sean, M la matriz $g \times p$ de medias de los grupos

$$\mathbb{M} = \begin{pmatrix} \widehat{\mu}_{11} & \widehat{\mu}_{12} & \dots & \widehat{\mu}_{1d} \\ \widehat{\mu}_{21} & \widehat{\mu}_{22} & \dots & \widehat{\mu}_{2d} \\ \dots & \dots & \dots & \dots \\ \widehat{\mu}_{g1} & \widehat{\mu}_{g2} & \dots & \widehat{\mu}_{gd} \end{pmatrix},$$

donde $\widehat{\mu}_{ij}$ es la media de la j-ésima característica en el i-ésimo grupo y G la matriz $n \times g$ de indicadores de grupo (es decir, $g_{ij} = 1$ si y solo si la i-ésima observación es asignada al j-ésimo grupo), entonces la matriz de suma de cuadrados de las variables centradas respecto a los vectores de medias de los grupos, etiquetada por S_W es

$$S_W = (\mathbb{X} - GM)^t(\mathbb{X} - GM),$$

y la matriz de suma de cuadrados de los vectores de medias de los grupos centradas respecto a las medias globales, etiquetada por S_B es

$$S_B = (GM - \mathbf{1}\widehat{\mu})^t(GM - \mathbf{1}\widehat{\mu}).$$

3.3. Regresión logística

En regresión logística, dado que un objeto pertenece a uno de g grupos distintos G_i, $i = 1, \dots, g$, en una población $\mathbb{P}$, se desea asignar este objeto a uno de los g grupos utilizando d características, $X = (X_1, \dots, X_d)$, asociadas con el objeto. La asignación es óptima al maximizar la verosimilitud de las probabilidades posteriores de los grupos.

Los datos de entrenamiento de la regresión logística consiste de n pares (x_1, y_1), (x_2, y_2), $\dots$, (x_n, y_n), con $x_i \in \mathbb{R}^d$ y $y_i = (0, 0, \dots, 0, 1, 0 \dots, 0) \in \mathbb{R}^g$, es decir,

y_i tiene el valor de uno solo en la i-ésima coordenada y las restantes coordenadas toman el valor de cero. Así, el par (x_i, y_i) pertenecen al grupo G_i.

Utilizando el teorema de Bayes, la probabilidad posterior de G_i es

$$\begin{aligned} q_i(X) &= P[X \in G_i|X] \\ &= \frac{P[X|X \in G_i]P[X \in G_i]}{\sum_{j=1}^{g} P[X|X \in G_j]P[X \in G_j]} \\ &= \frac{f_i(X)\pi_i}{\sum_{j=1}^{g} f_j(X)\pi_j}. \end{aligned} \tag{3.12}$$

El modelo de regresión logística surge del deseo de modelar las probabilidades posteriores de los g grupos por medio de funciones lineales en X, asegurando que sumen uno y permanezcan en el intervalo $[0, 1]$.

Como las reglas de clasificación desarrolladas en las Sección (3.1) dependen de la razón de funciones de densidad (3.1), se puede expresar esta razón sin especificar las densidades individuales $f_i(X)$ asumiendo que

$$\ln\left[\frac{f_i(X)}{f_g(X)}\right] = \alpha_i + \beta_i^t X, \quad \text{donde } i = 1, \ldots, g-1. \tag{3.13}$$

Utilizando las probabilidades posteriores (3.12) y el supuesto (3.13) se obtiene que

$$\begin{aligned} \ln\left(\frac{q_i(X)}{q_g(X)}\right) &= \ln\left(\frac{f_i(X)\pi_i}{f_g(X)\pi_g}\right) \\ &= \ln\left(\frac{f_i(X)}{f_g(X)}\right) + \ln\left(\frac{\pi_i}{\pi_g}\right) \\ &= \alpha_i + \beta_i^t X + \log\left(\frac{\pi_i}{\pi_g}\right) \\ &= \beta_{0i} + \beta_i^t X \end{aligned} \tag{3.14}$$

El modelo (3.14) se conoce con el nombre de modelo de las probabilidades posteriores o de las transformaciones logísticas. El modelo está especificado en función de las $g-1$ transformaciones logísticas. Aunque el modelo utiliza el último grupo como denominador en las razones de las probabilidades posteriores, la elección del denominador es arbitraria puesto que los estimadores son equivalentes bajo esta elección. Una de las ventajas de este modelo es que solamente se necesita estimar $(g-1)(d+1)$ parámetros sin tener que especificar las densidades f_i. Otra ventaja es que la familia de distribuciones que satisfacen (3.13) es muy amplia.

Un simple cálculo muestra que

$$\begin{aligned} q_i(X) = P[X \in G_i|X] &= \frac{e^{\beta_{0i}+\beta_i^t X}}{1+\sum_{l=1}^{g-1} e^{\beta_{0l}+\beta_l^t X}} \quad i = 1\ldots, g-1, \\ q_g(X) = P[X \in G_g|X] &= \frac{1}{1+\sum_{l=1}^{g-1} e^{\beta_{0l}+\beta_l^t X}}, \end{aligned} \tag{3.15}$$

y ellas claramente suman uno. Para enfatizar la dependencia en el conjunto de parámetros $\theta = \{\beta_{01}, \beta_1, \ldots, \beta_{0(g-1)}, \beta_{g-1}\}$, escribimos las probabilidades como $q_i(X) = P[X \in G_i|X] = p_i(X;\theta)$. Note que β_{0i} es un número real, mientras que β_i es un vector de $\mathbb{R}^d$.

Los modelos de regresión logística son usualmente estimados utilizando el método de máxima verosimilitud, utilizando la verosimilitud condicional de G dado X. Como $P[G|X]$ especifica completamente la distribución condicional, la distribución multinomial es apropiada. El logaritmo de la verosimilitud para n observaciones es

$$l(\theta) = \sum_{i=1}^{n} \ln p_i(x_i;\theta). \tag{3.16}$$

Como cada objeto en el conjunto de prueba debe ser asignado a un solo grupo de los g posibles, la probabilidad de que sea asignado al j-ésimo grupo es

$$q_1(x_i)^{y_{i1}}, q_2(x_i)^{y_{i2}}, \ldots, q_j(x_i)^{y_{ij}}, \ldots, q_g(x_i)^{y_{ig}},$$

y la ecuación (3.16) toma la forma

$$l(\theta) = \sum_{i=1}^{n}\sum_{j=1}^{g} y_{ij} \ln q_j(x_i).$$

Si se sabe que las variables predictoras X tiene una distribución normal multivariada, entonces las reglas discriminantes descritas en la Sección (3.1) son las mejores en la discriminanción. El método de regresión logística se puede considerar en situaciones en las que las variables predictoras no estén distribuidas normalmente y en las que algunas o todas esas variables sean discretas o categóricas. La regresión logística se asemeja a la regresión múltiple, es lineal en los parámetros y la diferencia principal es que, en la regresión logística, la variable dependiente es categórica, en tanto que en la múltiple, es variable dependiente es continua. Mucho de lo que se acostumbra hacer en el modelado por regresión tiene su contraparte en la regresión logística, los métodos de selección de variables, como hacia atrás y por pasos, se pueden utilizar en ambos tipos de modelos.

Para emplear el modelo de regresión logística con fines de discriminación, se calcula una estimación de las probabilidades $q_i(X_0)$ por medio de las ecuaciones (3.15) para un nuevo individuo X_0 y se lo asigna al grupo para el cual se obtenga la $q_i(X_0)$ más grande, es decir, la regla inducida por la regresión logística es

$$\boxed{\begin{array}{c}\text{asignar un miembro de } \mathbb{P} \text{ al grupo que corresponda a} \\ G(X) = \arg\max_i \widehat{q}_i(X), \quad i = 1, \ldots, g.\end{array}} \tag{3.17}$$

Tibshirani et al ([4]) afirman que en la práctica los supuestos para aplicar el análisis discriminante usualmente no son logrados, y frecuentemente algunos de los componentes de las características X son cualitativas. Mencionan que a pesar de que es comúnmente pensado que la regresión logística es más robusta que el análisis discriminante, la experiencia de estos investigadores ha mostrado que los dos modelos producen resultados muy similares aún cuando el análisis discriminante es utilizado inadecuadamente, por ejemplo, utilizando variables cualitativas.

3.4. Árboles de clasificación

En el método denominado árboles de clasificación, dado que un objeto pertenece a uno de g grupos distintos G_i, $i = 1, \ldots, g$, en una población $\mathbb{P}$, se desea asignar este objeto a uno de los g grupos utilizando d características continuas o categóricas, $X = (X_1, \ldots, X_d)$, asociadas con el objeto. La asignación debe ser óptima en algún sentido tal como minimizar el número de errores o minimizar algún índice adecuado.

Para simplificar la introducción centremos la atención en el siguiente procedimiento recursivo de particionamiento binario. En primer lugar se divide el espacio de las características X en dos regiones, R_1 y R_2. Es decir, $X = R_1 \cup R_2$. Luego en cada una de éstas dos regiones calculamos la proporción de objetos pertenecientes a los g grupos

$$\widehat{p}_{mj} = \frac{1}{n_m} \sum_{x_i \in R_m} \mathbb{I}(y_i = j), \ m = 1, 2, \ j = 1, \ldots, g,$$

donde n_m es el número de objetos en la región R_m. Entonces, decimos que las observaciones en el nodo m pertenecen al grupo para el cual

$$j(m) = \arg\max_j \widehat{p}_{mj},$$

el grupo que es mayoría en la región. La variable seleccionada y el punto de división deben ser seleccionados de tal manera que se obtenga el mejor ajuste. Luego, la región

R_1 o la R_2 o ambas se dividen en dos regiones más, y este proceso se repite hasta que alguna regla de parada sea alcanzada. Por ejemplo, la primera división se realiza en $X_1 = t_1$. Luego la región $X_1 \leq t_1$ se divide en $X_2 = t_2$ y la región $X_1 > t_1$ se divide en $X_1 = t_3$. Finalmente, la región $X_1 > t_3$ se divide en $X_2 = t_4$. El resultado de este proceso es una partición del espacio de las características en cinco regiones, $R_1, \ldots, R_5$, que se muestran en el gráfico izquierdo de la Figura (3.2).

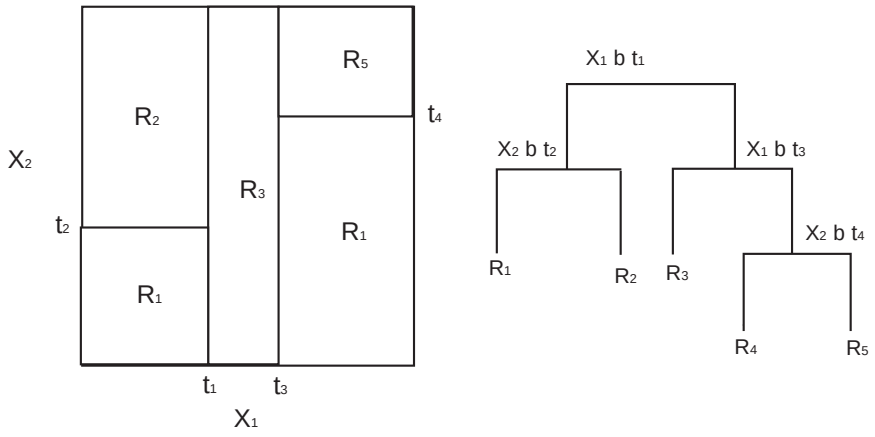


Figura 3.2: Ejemplo de un árbol de clasificación

Este mismo modelo puede representarse por el árbol binario mostrado en el gráfico derecho de la Figura (3.2). Todos los datos de entrenamiento están situados en la parte mas alta del árbol. Los datos que satisfacen la condición en cada unión se asignan a la rama izquierda y las otras a la rama derecha. Los nodos terminales o hojas del árbol corresponden a las regiones $R_1, \ldots, R_5$. La principal ventaja de un árbol binario recursivo es que es fácil de interpretar.

Para construir un árbol de clasificación se dispone de los datos de entrenamiento que consiste de n pares ordenados (x_i, y_i), $i = 1, \ldots, n$, con $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ y $y_i \in \{1, 2, \ldots, g\}$. El algoritmo necesita conocer la regla que se utiliza para dividir el conjunto en dos (la regla de división) y cómo decidir que un conjunto es un nodo terminal (la regla de parada).

Las reglas de división más simples son aquellas que buscan solamente un paso hacia delante el resultado de la división propuesta. Estas reglas hacen esto encontrando la mejor división para cada característica a la vez teniendo una medida de cuan buena una división es. Estas reglas deciden la división de cada característica es la mejor bajo esta medida. Para una variable continua X_i, se busca en las divisiones $x_i < s$, $x_i \geq s$ para todos los valores de s y encuentran el valor de s donde la medida es la mejor. Si

X_i es una variable categórica, entonces se busca en todas las divisiones posibles de las categorías en dos y verifican la medida bajo estas dos diferentes divisiones. Así que, qué medida es utilizada? Las medidas más comunes son el índice de impureza básico, el índice de Gini, el índice de entropía y la suma-media de cuadrados.

Hay una clase completa de índices de impureza que tratan de evaluar cuán impuro es cada nodo m del árbol, donde pureza corresponde a un nodo con una sola clase de elementos. Si se divide el nodo m en los nodos izquierdo l y derecho r con proporciones $p(l)$ y $p(r)$, respectivamente, se puede medir el cambio en la impureza que provocó la división por medio de

$$I = i(m) - p(l)i(l) - p(r)i(r),$$

donde la función $i(\cdot)$ es una medidad de impureza. Mientras más grande es esta diferencia, mayor es el cambio en la impureza, lo que significa que los nuevos nodos son más puros. Esto es lo que se desea, así que se elige la división que maximiza esta expresión. Esto es equivalente a minimizar $p(l)i(l) + p(r)i(r)$. Obviamente, si no hay una división con una diferencia positiva, no se debería dividir el nodo.

El índice de impureza más básico es aquel que cumple con

$$i(m) = \begin{cases} \widehat{p}_{Gm} & \text{si } \widehat{p}_{Gm} \leq 0{,}5 \\ \widehat{p}_{Bm} & \text{si } \widehat{p}_{Bm} < 0{,}5, \end{cases}$$

la proporción del grupo más pequeño en el nodo m.

En lugar de que la probabilidad de impureza sea lineal, el índice de Gini es cuadrático para dar más énfasis a los nodos puros. Se define por

$$i(m) = \widehat{p}_{Gm}\widehat{p}_{Bm},$$

así que la diferencia ahora es

$$\text{Gi} = \widehat{p}_{Gm}\widehat{p}_{Bm} - p(l)\widehat{p}_{Gl}\widehat{p}_{Bl} - p(r)\widehat{p}_{Gr}\widehat{p}_{Br}.$$

Otro índice no lineal es el índice de entropía o deviance,

$$i(m) = -\widehat{p}_{Gm}\ln(\widehat{p}_{Gm}) - \widehat{p}_{Bm}\ln(\widehat{p}_{Bm}).$$

Como su nombre lo sugiere, está relacionado a la entropía o a la cantidad de información en la división de buenos y malos en el nodo. Es una medida de cuántas formas diferentes pueden dar lugar a la división actual de buenos y malos en el nodo.

La última medida que no es un índice, pero proviene del estadístico Ji-cuadrado, que prueba si la proporción de buenos es la misma en los dos nodos hijos. Si este estadístico Ji-cuadrado es grande, se dice que no hay suficiente información para aceptar la hipótesis, es decir, las dos proporciones no son las mismas. Si $n(l)$ y $n(r)$ son el número total de casos en los nodos izquierdo y derecho, entonces el estadístico Ji proviene de maximizar

$$\text{Ji} = n(l)n(r) - \frac{(\widehat{p}_{Gl} - \widehat{p}_{Gr})^2}{n(l) + n(r)}.$$

Para cuando se consideran g grupos diferentes, el índice de Gini adopta la forma

$$i(m) = \sum_{k \neq k'} \widehat{p}_{mk}\widehat{p}_{mk'} = \sum_{k=1}^{g} \widehat{p}_{mk}(1 - \widehat{p}_{mk'}),$$

y el índice de entropía adopta la forma

$$i(m) = \sum_{k=1}^{g} \widehat{p}_{mk} \ln(\widehat{p}_{mk}).$$

Si se obtiene un árbol donde cada nodo terminal tiene solamente un caso del conjunto de entrenamiento, entonces el árbol será un discriminador perfecto en el conjunto de entrenamiento pero será un pobre clasificador con otro conjunto. Así, se crea nodos terminales debido a que si el número de casos e un nodo es pequeño entonces no tiene sentido dividirlos en el futuro. Este caso se da cuando hay menos de 10 casos en el nodo.

Cuán grande debería ser un árbol? Claramente un árbol muy grande puede sobre-estimar los datos, mientras que un pequeño puede no capturar la estructura que realmente es importante. El tamaño de un árbol es un parámetro de calibración que govierna la complejidad del modelo, y el tamaño óptimo del árbol debería ser de los datos de entrenamiento. La estrategia preferida es formar un árbol grande T_0 y detener el proceso de división solo cuando un número de nodos determinado es logrado. Luego este árbol grande es podado utilizando lo que se conoce como cost-complexity pruning, que se describe a continuación.

Se define un sub-árbol $T \subset T_0$ como cualquier árbol obtenido luego de podar T_0, esto es, colapsando cualquier número de sus nodos internos (no los nodos terminales). Los nodos son indexados con la letra m, con el nodo m representando a la región R_m. Sea $|T|$ el número de nodos terminales en T. Se define

$$C_\alpha(T) = \sum_{m=1}^{|T|} n_m i(m) + \alpha|T|.$$

La idea es encontrar, para cada α, el sub-árbol $T_\alpha \subset T_0$ que minimice $C_\alpha(T)$. El parámetro de calibración $\alpha \geq 0$ govierna el compromiso entre el tamaño del árbol y su capacidad de ajuste a los datos. Grandes valores de α producen árboles pequeños, y árboles grandes si sucede lo opuesto. Cuando $\alpha = 0$ se optiene el árbol completo T_0.

Un problema significante de los árboles de clasificación es que tienen alta varianza. Frecuentemente un pequeño cambio en los datos de entrenamiento puede producir divisiones diferentes, luego las interpretaciones son inadecuadas. Este comportamiento puede afectar el proceso de evaluación del árbol ajustado con el conjunto de prueba. Para lidiar con este inconveniente se puede promediar varias árboles en la etapa de entrenamiento para reducir la varianza.

3.5. Redes Neuronales

Una red neuronal es un modelo estadístico no lineal de clasificación o de regresión que consta de dos etapas, típicamente representado por una diagrama de red como se ilustra en la Figura (3.3). El nombre de redes neuronales se debe a que en sus inicios fueron modelos del cerebro humano. En esta sección describimos la red neuronal feed forward de una sola capa oculta con M unidades.

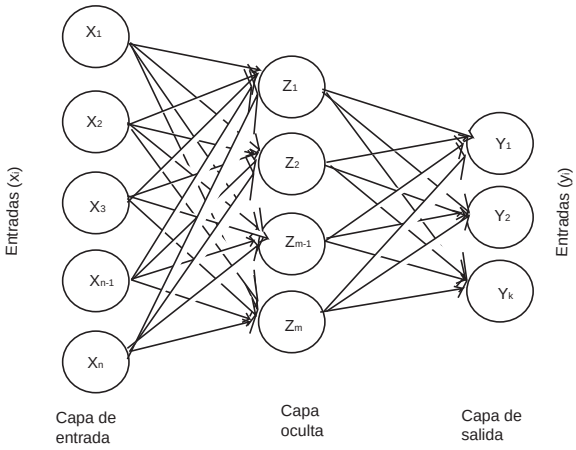


Figura 3.3: Ejemplo de un diagrama de red neuronal

Los datos de entrenamiento de la red neuronal consiste de n pares (x_1, y_1), (x_2, y_2), $\ldots$, (x_n, y_n), con $x_i \in \mathbb{R}^d$ y $y_i = (0, 0, \ldots, 0, 1, 0 \ldots, 0) \in \mathbb{R}^g$, es decir, y_i tiene el valor de uno solo en la i-ésima coordenada y las restantes coordenadas toman el valor de cero. Así, el par (x_i, y_i) pertenecen al grupo G_i.

Las redes neuronales toman como datos de entrada las características $X = (X_1, \ldots, X_d)$ de los n objetos en el conjunto de entrenamiento y obtienen M características derivadas,

$$Z_m = \sigma(\alpha_{0m} + \alpha_m^t X),\ m = 1, \ldots, M,$$

por medio de combinaciones lineales de las entradas X. Esto supone que la capa oculta de la red neuronal tiene M unidades. Luego las variables y_i son estimadas por medio de combinaciones lineales de las variables Z_m,

$$T_i = \beta_{0i} + \beta_i^t Z,\ i = 1, \ldots, g,$$
$$f_i(X) = g_i(T),\ i = 1, \ldots, g,$$

donde $Z = (Z_1, Z_2, \ldots, Z_M)$ y $T = (T_1, T_2, \ldots, T_M)$.

La función de activación $\sigma(\cdot)$ usualmente es la función sigmoidal

$$\sigma(\nu) = \frac{1}{(1 + e^{-\nu})}.$$

La función de salida $g_i(\cdot)$, para clasificación, es la función softmax

$$g_i(T) = \frac{e^{T_i}}{\sum_{l=1}^{g} e^{T_l}}.$$

Esta transformación es la que se utiliza en la sección (3.3), y produce estimaciones positivas que suman uno. Entonces, $f_i(X)$ es la probabilidad del grupo G_i.

Los parámetros desconocidos de las redes neuronales se denominan pesos y deben ser calculados de tal forma que el modelo se ajuste a los datos de entrenamiento. Sea θ el conjunto de todos los pesos

$$\{\alpha_{0m}, \alpha_m,\ m = 1, \ldots, M\}\ \ M(p+1) \text{ pesos},$$
$$\{\beta_{0i}, \beta_i,\ i = 1, \ldots, g\}\ \ g(M+1) \text{ pesos}.$$

Para clasificación se utiliza como medida de ajuste la función de error cross-entropy

$$R(\theta) = -\sum_{i=1}^{n}\sum_{k=1}^{g} y_{ik} \log f_k(x_i),$$

y la regla de clasificación es

$$\boxed{\begin{array}{c}\text{asignar un miembro de } \mathbb{P} \text{ al grupo que corresponda a} \\ G(X) = \arg\max_k f_k(X).\end{array}} \tag{3.18}$$

Típicamente no se requiere el mínimo global de $R(\theta)$ puesto que se obtiene soluciones sobre-ajustadas. Para superar este inconveniente se necesita una forma de

regularización denominada weight decay. Esta regularización añade una penalización a la función de error, $R(\theta) + \lambda J(\theta)$, donde

$$J(\theta) = \sum_{km} \beta_{km}^2 + \sum_{ml} \alpha_{ml}^2,$$

y $\lambda \geq 0$ es un parámeton de calibración. Usualmente λ es estimado via validación cruzada.

Se recomienda que los pesos iniciales de la red neuronal sean elegidos al azar, los datos de entrada sean estandarizados y que las unidades en la capa oculta estén entre 5 y 100. Además, se recomienda hacer varias corridas del modelo con varios pesos elegidos al azar porque existen varios mínimos locales de $R(\theta)$, y luego seleccionar la solución que produce el menor error penalizado.

Desde un punto de vista práctico, las redes neuronales aplicadas a credit scoring son utilizadas en procesos de post-aprovación más que en el momento de decidir otorgar o no un crédito, puesto que no es posible explicar las variables que producen una decisión adversa.

3.6. Support Vector Machines

Supongamos que existen dos grupos, G_1 y G_2, en una población $\mathbb{P}$. Nuestra intención es encontrar una frontera en el espacio de las características,
$X = (X_1, X_2, \ldots, X_p)$, que separe a estos dos grupos con el propósito de obtener una regla de clasificación para nuevos objetos.

Estos dos grupos pueden estar completamente separados o solapados en el espacio X, entonces la frontera que nos interesa puede ser lineal o no lineal dependiendo del grado de separación que nos interese. La metodología que determina esta frontera se denomina Support Vector Machines.

En la practica, para encontrar la frontera que separará a los dos grupos, se dispone de dos conjuntos de datos, el conjunto de entrenamiento y el conjunto de prueba. Con el primero ajustamos la frontera y con el segundo evaluamos la capacidad discriminatoria de ésta. Los datos de entrenamiento consiste de n pares
$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, con $x_i \in \mathbb{R}^d$ y $y_i \in \{-1, 1\}$. Hay que notar que para aplicar este método de clasificación se deben etiquetar los grupos G_1 y G_2 con los valores 1 y -1.

3.6.1. Grupos separables

Cuando los dos grupos son separables (no se solapan) en el espacio X, es suficiente buscar una frontera lineal que los separará completamente. Tales fronteras reciben el nombre de hiperplanos y son simplemente combinaciones lineales de las d características. Formalmente, definimos a un hiperplano como

$$\{X : \ X^t\beta + \beta_0 = 0\},$$

donde β es un vector unitario, $\|\beta\| = 1$. Un ejemplo de esta situación se muestra en la Figura (3.4).

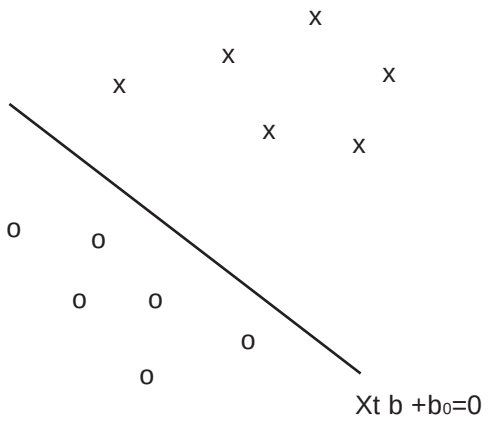


Figura 3.4: Dos grupos separables por cualquier hiperplano

Una característica relevante en el desarrollo subsiguiente es la distancia que existe desde un punto $x \in \mathbb{R}^d$ al hiperplano $H = \{X : \ f(X) = X^t\beta + \beta_0 = 0\}$. Si $X_0 \in H$, la distancia con signo de X al hiperplano H es

$$\frac{\beta^t}{\|\beta\|}(X - X_0) = \beta^t X - \beta^t X_0 = \beta^t X + \beta_0 = f(X).$$

Como un solo hiperplano separador deja a los objetos etiquetados positivamente separados de aquellos etiquetados negativamente, podemos suponer que para aquellos objetos con $y_i = 1$ se tenga que $f(x_i) = x_i^t\beta + \beta_0 > 0$ y para aquellos con $y_i = -1$ se tenga que $f(x_i) = x_i^t\beta + \beta_0 < 0$; entonces la regla de clasificación inducida por $f(X)$ es

asignar un miembro de $\mathbb{P}$ al grupo que corresponda a

$$G(X) = \text{sign}(X^t\beta + \beta_0). \tag{3.19}$$

Debe de ser evidente que existen varios hiperplanos separadores, así, el algoritmo support vector machine busca el hiperplano que produzca el mayor margen entre los puntos de entrenamiento para los grupos G_1 y G_2. En concreto, queremos el hiperplano tal que la distancia de todos los puntos x_i a este sea mayor que un valor C, es decir, si $y_i = 1$, $x_i^t\beta + \beta_0 \geq C$ o si $y_i = -1$, $x_i^t\beta + \beta_0 \leq -C$. Esta situación es ilustrada en la Figura (3.5).

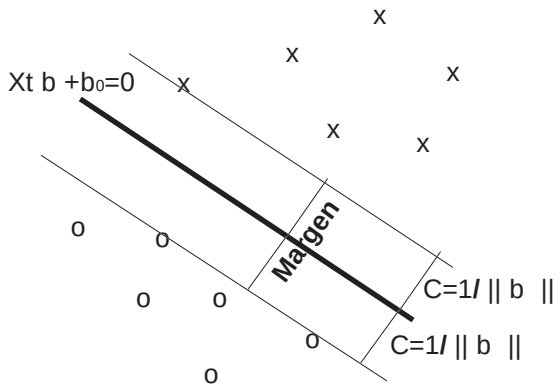


Figura 3.5: Dos grupos separables por el hiperplano que logra la separación máxima

El problema de optimización que captura este objetivo es

$$\begin{aligned} &\max_{\beta,\beta_0,\|\beta\|=1} C \\ &\text{s.a.} \quad y_i(x_i^t\beta + \beta_0) \geq C, i = 1, \ldots, n. \end{aligned} \tag{3.20}$$

Para no incluir la restricción $\|\beta\| = 1$ en el problema (3.20) se hace que las restricciones sean

$$\frac{1}{\|\beta\|} y_i(x_i^t\beta + \beta_0) \geq C, i = 1, \ldots, n,$$

o equivalentemente

$$y_i(x_i^t\beta + \beta_0) \geq C\|\beta\|, i = 1, \ldots, n.$$

Como para cualquier β y β_0 que satisfacen estas ecuaciones, cualquier múltiplo positivo escalado también las satisface, se puede hacer arbitrariamente que $\|\beta\| = 1/C$. Entonces, (3.20) es equivalente a

$$\begin{aligned} &\min_{\beta,\beta_0} \frac{1}{2}\|\beta\|^2 \\ &\text{s.a.} \quad y_i(x_i^t\beta + \beta_0) \geq 1, i = 1, \ldots, n. \end{aligned} \tag{3.21}$$

En (3.21) se elevó la norma de β al cuadrado porque es equivalente minimizar las funciones $g(t) = \sqrt{t}$ o $g(t) = t$; y se incluyó el término $1/2$ para que al derivar la función objetivo desaparezcan las constantes.

El problema (3.21) es un problema de optimización convexo, y para resolverlo hacemos uso del método de los multiplicadores de Lagrange. La principal ventaja de este enfoque es que los datos de entrenamiento apareceran en forma de producto punto entre vectores, característica que será crucial al momento de generalizar el método support vector machine al caso no lineal. La función de Lagrange es

$$L = \frac{1}{2}\|\beta\|^2 + \sum_{i=1}^{n} \alpha_i[y_i(x_i^t\beta + \beta_0) - 1]. \tag{3.22}$$

Ahora debemos calcular $\min_{\beta,\beta_0} L$. Para eso, igualamos las derivadas $\partial L/\partial\beta$ y $\partial L/\partial\beta_0$ a cero y obtenemos

$$\beta = \sum_{i=1}^{n} \alpha_i y_i x_i, \tag{3.23}$$

$$0 = \sum_{i=1}^{n} \alpha_i y_i. \tag{3.24}$$

La solución de (3.21) debe satisfacer las condiciones de Karush-Kuhn-Tucker

$$\frac{\partial L}{\partial \beta_\nu} = \beta_\nu - \sum_{i=1}^{n} \alpha_i y_i x_{i\nu} = 0,\ \nu = 1, \ldots, d \tag{3.25}$$

$$\frac{\partial L}{\partial \beta_0} = -\sum_{i=1}^{n} \alpha_i y_i = 0, \tag{3.26}$$

$$y_i(x_i^t\beta + \beta_0) - 1 \geq 0,\ i = 1, \ldots, n \tag{3.27}$$

$$\alpha_i \geq 0\ \forall i, \tag{3.28}$$

$$\alpha_i[y_i(x_i^t\beta + \beta_0) - 1] = 0\ \forall i. \tag{3.29}$$

De la ecuación (3.29) se puede observar que

- si $\alpha_i > 0$, entonces $y_i(x_i^t\beta + \beta_0) = 1$, es decir, los x_i están en los límites del margen;
- si $y_i(x_i^t\beta + \beta_0) - 1 > 0$, es decir, los x_i no estan en los límites de la banda, $\alpha_i = 0$.

Así, podemos observar en (3.23) que el vector β es una combinación lineal de los x_i que se encuentran en los límites del margen. Estos x_i se conocen como vectores soporte (support vectors). Para determinar β_0 utilizamos la ecuación (3.29) y elegimos cualquier i tal que $\alpha_i \neq 0$ y calculamos β_0. Entonces, el hiperplano separador óptimo o la frontera lineal óptima produce la función $\widehat{f}(X) = X^t\widehat{\beta} + \widehat{\beta}_0$ para clasificar nuevas observaciones de acuerdo a la regla

asignar un miembro de $\mathbb{P}$ al grupo que corresponda a

$$G(X) = \text{sign}\widehat{f}(X). \tag{3.30}$$

Las condiciones de Karush-Kuhn-Tucker son satisfechas en la solución de cualquier problema de optimización con restricciones, convexo o no, con cualquier clase de restricciones, en particular con restricciones lineales como es el caso del problema (3.21). Además, como el problema (3.21) es convexo, las condiciones de Karush-Kuhn-Tucker son necesarias y suficientes para decir que $\widehat{\beta}$, $\widehat{\beta}_0$ y $\widehat{\alpha}$ es una solución de (3.21).

La teoría de optimización establece que todo problema tiene un dual si la función a optimizar y las restricciones son estrictamente convexas. Este es el caso del problema (3.21). Las condiciones de Karush-Kuhn-Tucker se emplean para obtener el problema dual, que tiene la ventaja de que su resolución tiene una complejidad que crece con el número de datos n y no con la dimensión de los mismos, d. Así, se pueden resolver problemas con un número moderado de muestras, por ejemplo, 10000.

Entonces, sustituyendo (3.23) y (3.24) en (3.22) se obtiene la función de Lagrange del problema dual

$$\begin{aligned} L_D &= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \alpha_i \alpha_k y_i y_k x_i^t x_k \\ &\text{sujeto a } \alpha_i \geq 0. \end{aligned}$$

La solución se obtiene maximizando L_D en la región definida por $\alpha_i \geq 0$. Este es un simple problema de optimización convexa que software estandar lo puede resolver.

3.6.2. Grupos solapados

Cuando los dos grupos se solapan en el espacio de las características X, se puede buscar una frontera lineal o no lineal que los separe, permitiendo que la regla de clasificación inducida por esta frontera realice malas clasificaciones en cierto grado. Naturalmente, es de esperar que la frontera no lineal realice malas clasifcaciones en un grado menor al realizado por una frontera lineal.

Una forma de tratar este solapamiento es maximizar $\|C\|$, pero permitiendo que ciertos puntos se ubiquen en el lado incorrecto del margen; esta estrategia provoca tener costos adicionales que se veran reflejados en la función objetivo. Sean las variables de holgura $\xi = (\xi_1, \xi_2, \ldots, \xi_N)$, entonces una forma natural de modificar las restricciones

en (3.20) es haciendo que

$$y_i(x_i^t\beta + \beta_0) \geq C(1 - \xi_i),\ \forall i,\ \xi_i \geq 0,$$
$$\sum_{i=1}^{N} \xi_i \leq k.$$

La restricción $y_i(x_i^t\beta + \beta_0) \geq C(1 - \xi_i)$ asegura que cuando $0 \leq \xi_i < 0$ el dato de entrenamiento x_i ya no está a una distancia C del hiperplano separador sino que está a una distancia $C(1 - \xi_i) < C$, es decir, que x_i está dentro del margen. Cuando $\xi_i > 1$, el dato x_i se encuentra en el lado incorrecto, es decir, está mal clasificado. Por tanto, si la suma $\sum \xi$ es acotada por k, el número total de malas clasificaciones está acotado por k. Esta situación se ilustar en la Figura (3.6).

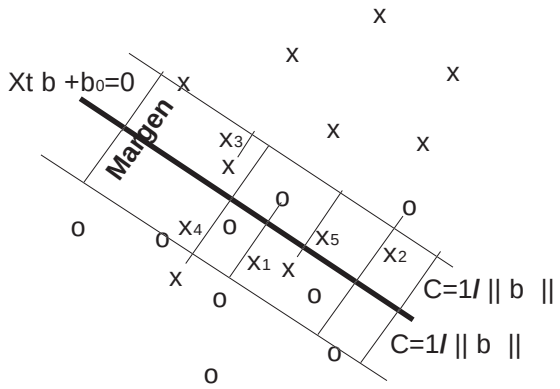


Figura 3.6: Dos grupos solapados separados por el hiperplano que logra el error especificado

Si $C = 1/\|\beta\|$, la restricción en la norma de β ya no es considerada y (3.20) se convierte en

$$\begin{aligned} &\min_{\beta,\beta_0} \frac{1}{2}\|\beta\|^2 \\ \text{s.a.}\quad &y_i(x_i^t\beta + \beta_0) \geq 1 - \xi_i, i = 1, \ldots, n, \\ &\sum_{i=1}^{n} \xi_i \leq K, \\ &\xi_i \geq 0. \end{aligned} \tag{3.31}$$

Por los errores de clasificación que se permiten se debe añadir un costo extra a la función objetivo, y un costo natural es $\gamma \sum_{i=1}^{n} \xi_i$, donde γ debe ser elegido. Además,

computacionalmente es conveniente expresar (3.31) como

$$\begin{aligned} &\min_{\beta,\beta_0} \frac{1}{2}\|\beta\|^2 + \gamma \sum_{i=1}^{n} \xi_i \\ \text{s.a.} \quad &y_i(x_i^t\beta + \beta_0) \geq 1 - \xi_i, i = 1, \ldots, n, \\ &\xi_i \geq 0. \end{aligned} \tag{3.32}$$

El caso separable corresponde cuando $\gamma = \infty$.

La función de Lagrange es

$$L = \frac{1}{2}\|\beta\|^2 + \gamma \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i[y_i(x_i^t\beta + \beta_0) - (1 - \xi_i)] - \sum_{i=1}^{n} \mu_i\xi_i, \tag{3.33}$$

que debe ser maximizada con respecto a β, β_0 y ξ_i. Igualando a cero las derivadas $\partial L/\partial\beta$, $\partial L/\partial\beta_0$ y $\partial L/\partial\xi_i$ se tiene que

$$\beta = \sum_{i=1}^{n} \alpha_i y_i x_i, \tag{3.34}$$

$$0 = \sum_{i=1}^{n} \alpha_i y_i, \tag{3.35}$$

$$\alpha_i = \gamma - \mu_i, \ \forall i, \tag{3.36}$$

$$\alpha_i, \mu_i, \xi_i \geq 0 \ \forall i. \tag{3.37}$$

Las condiciones de Karush-Kuhn-Tucker para el problema (3.32) son

$$\frac{\partial L}{\partial \beta_\nu} = \beta_\nu - \sum_{i=1}^{n} \alpha_i y_i x_{i\nu} = 0, \quad \nu = 1, \ldots, p, \tag{3.38}$$

$$\frac{\partial L}{\partial \beta_0} = -\sum_{i=1}^{n} \alpha_i y_i = 0, \tag{3.39}$$

$$\frac{\partial L}{\partial \xi_i} = \gamma - \alpha_i - \mu_i = 0, \tag{3.40}$$

$$y_i(x_i^t\beta + \beta_0) - 1 + \xi_i \geq 0, \quad i = 1, \ldots, n, \tag{3.41}$$

$$\xi_i \geq 0, \tag{3.42}$$

$$\alpha_i \geq 0, \tag{3.43}$$

$$\mu_i \geq 0, \tag{3.44}$$

$$\alpha_i[y_i(x_i^t\beta + \beta_0) - 1 + \xi_i] = 0, \tag{3.45}$$

$$\mu_i\xi_i = 0. \tag{3.46}$$

De las ecuación (3.34) se tiene que

$$\widehat{\beta} = \sum_{i=1}^{n} \widehat{\alpha}_i y_i x_i, \tag{3.47}$$

con $\alpha_i \neq 0$ solo para aquellos datos de entrenamiento que satisfacen exactamente la ecuación (3.41) debido a la ecuación (3.45). Estos datos se denominan vectores soporte (support vectors), puesto que β está representado en términos de éstos. Con las ecuaciones (3.45) y (3.46) se determina β_0. Note que la ecuación (3.40) combinada con la ecuación (3.46) provoca que $\xi_i = 0$ si $\alpha_i < \gamma$. Así, se toma cualquier dato de entrenamiento para el cual $0 < \alpha_i < \gamma$ y con la ecuación (3.45), con $\xi_i = 0$, se calcula β_0. Es prudente calcular el promedio de todos los β_0 producidos con todos los α_i que satisfacen la condición.

Dadas las soluciones de $\widehat{\beta}$ y $\widehat{\beta}_0$, la regla de asignación es

asignar un miembro de $\mathbb{P}$ al grupo que corresponda a

$$G(X) = \text{sign}[X^t\widehat{\beta} + \widehat{\beta}_0]. \tag{3.48}$$

El parámetro de calibración de este procedimiento es γ.

Sustituyendo (3.34) hasta (3.37) en (3.33) se obtiene la función de Lagrange para el problema dual

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j x_i^t x_j, \tag{3.49}$$

que da una cota inferior para la función objetivo en (3.32) para cualquier vector factible. Entonces, se debe resolver

$$\begin{aligned} &\max_{\alpha_i} L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j x_i^t x_j \\ \text{s.a.}\quad &0 \leq \alpha_i < \gamma \\ &\sum_{i=1}^{n} \alpha_i y_i = 0. \end{aligned} \tag{3.50}$$

El problema dual (3.50) es cuadrático convexo y más simple que el problema dual (3.33), y puede ser resuelto con técnicas estandars.

3.6.3. Support vector machines no lineales

El clasificador en base a vectores (support vector classifier) descrito hasta ahora en las secciones (3.6.1) y (3.6.2) encuentra fronteras lineales en el espacio de las características X. Cómo se puede generalizar estos métodos para cuando la frontera que separa a los grupos G_1 y G_2 no es lineal? Para esto note que la única manera en que intervienen las características de entrenamiento en las ecuaciones (3.50) es en

forma de producto punto $x_i^t x_j$. Ahora, suponga que transformamos las características de entrenamiento $h(X)$, entonces las características transformadas también intervendran en forma de producto punto, $h(x_i)^t h(x_j)$. Una vez que las transformaciones $h_m(X), m = 1, \ldots, M$ son seleccionadas, el procedimiento es similar a lo anterior, es decir, se utiliza el clasificador en base a vectores con las características $h(x_i) = [h_1(x_i), h_2(x_i), \ldots, h_M(x_i)]$, $i = 1, \ldots, n$, y se obtiene la función no lineal $\widehat{f}(X) = h(X)^t \widehat{\beta} + \widehat{\beta}_0$. Ahora la regla de asignación es

asignar un miembro de $\mathbb{P}$ al grupo que corresponda a

$$G(X) = \text{sign}[h(X)^t \widehat{\beta} + \widehat{\beta}_0]. \tag{3.51}$$

Esta metodología se conocen con el nombre de support vector machines.

La función de Lagrange (3.49) para el problema dual tiene la forma

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle h(x_i), h(x_j) \rangle. \tag{3.52}$$

De la ecuación (3.34) la función solución $f(x)$ puede escribirse como

$$\begin{aligned} f(X) &= h(X)^t \beta + \beta_0 \\ &= \sum_{i=1}^{n} \alpha_i y_i \langle h(X), h(x_i) \rangle + \beta_0. \end{aligned} \tag{3.53}$$

Dado el α_i, β_0 se determina resolviendo $f(x_i) = 0$ en (3.53) para cualquier (o todos) x_i tal que $0 < \alpha_i < \gamma$. Así, (3.52) y (3.53) consideran a $h(X)$ solamente a traves de productos internos.

Ahora, si es que existiese una función núcleo K tal que $K(X, X') = \langle h(X), h(X') \rangle$, solo se necesitaría utilizar K en el algoritmo y nunca necesitaríamos decir explícitamente la forma de $h(\cdot)$. Los primeros núcleos que nos permiten hacer esta simplificación son

Polinomio de grado d: $K(x, y) = (1 + \langle x, y \rangle)^d$,

Bases radiales: $K(x, y) = \exp(-\|x - y\|^2 / c)$,

Red neuronal: $K(x, y) = \tanh(k_1 \langle x, y \rangle + k_2)$.

Por suerte, todas las consideraciones de las secciones anteriores se mantienen, puesto que todavía se está realizando una separación lineal, pero en el espacio de las características transformado. De la ecuación (3.53), la frontera no lineal que separa a los grupos G_1 y G_2 es

$$\widehat{f}(X) = \sum_{i=1}^{n} \widehat{\alpha}_i y_i K(X, x_i) + \widehat{\beta}_0.$$

La regla de asignación no lineal es

$$\boxed{\begin{array}{c}\text{asignar un miembro de } \mathbb{P} \text{ al grupo que corresponda a} \\ G(X) = \text{sign}[\widehat{f}(X)].\end{array}} \tag{3.54}$$

Capítulo 4

Aplicación

En este capítulo se utiliza las técnicas de clasificación desarrolladas en el Capítulo 3 para crear un modelo de credit scoring óptimo para una institución financiera ecuatoriana. En el desarrolo de esta aplicación se utilizará el software de código abierto *R*, disponible en http://cran.r-project.org. El código utilizado en el desarrollo de esta tesis se presenta en el Apéndice (6.9).

4.1. Descripción del problema

Desarrollar un modelo de credit scoring que sirva como soporte del proceso de decisión si otorgar o rechazar una solicitud de crédito de consumo utilizando características propias del solicitante.

4.2. Descripción de los datos

Para el desarrollo del modelo de credit scoring se dispone de 6.497 observaciones de 12 características correspondientes a individuos que han sido beneficiarios de crédito en la institución financiera en el año 2002.

Los individuos considerados son ecuatorianos que han recibido un crédito en la institución involucrada que hasta el corte no tienen cuotas pendientes. La solicitud de crédito fue clasificada de carácter personal.

En el Cuadro (4.1) se detalla las 12 características consideradas en el desarrollo del modelo de credit scoring.

Etiqueta	Significado
tele	Determina si el individuo posee línea telefónica. Toma el valor de 1 si no posee línea telefónica y 2 si es que posee.
sexo	Determina el género del individuo. Toma el valor de 1 si el individuo es Mujer y 2 si es hombre.
esci	Representa el estado civil del individuo. Toma los valores de 1, 2 y 3 si el individuo es casado, soltero o tiene otro estatus.
tido	Representa el tipo de domicilio del individuo. Toma los valores de 1, 2 y 3 cuando el lugar donde vive es arrendado, de familiares o propia, respectivamente.
prov	Representa la provincia de orígen del individuo. Toma los valores de 1 a 6 cuando es original de Cotopaxi, Guayas, Pichincha, del Oriente, de otras provincias de la Costa y de otras provincias de la Sierra, respectivamente.
ries	Indica si el individuo ha sido considerado de riesgo aceptable (good risk) o si ha sido considerado de riesgo no aceptable (bad risk). Toma el valor de 1 para cuando es B y 2 cuando es G.
edad	Representa la edad del individuo al momento del desembolso del crédito.
nhij	Representa la cantidad de hijos que dependen del individuo. Toma los valores 0, 1, 2 y 3 cuando no tiene hijos, tiene un hijo, tiene dos hijos y tiene tres o más hijos.
tres	Representa el tiempo de residencia en el domicilio actual.
mdes	Representa el monto desembolsado.
ting	Representa el total de ingresos de la familia que lidera el individuo.
tgas	Representa el total de gastos de la familia que lidera el individuo.

Cuadro 4.1: Características de los individuos consideradas en el desarrollo del modelo de credit scoring

4.3. Conjuntos de prueba y de entrenamiento

Como se detalló en la Sección (2.1), es necesario dividir el conjunto de 6.497 datos en dos conjunto, de entrenamiento y de prueba.

Debido a que se dispone de información correspondiente solo al año 2000 no fue posible utilizar una característica temporal que permita definir los conjuntos de entrenamiento y prueba. Por tanto, se decidió extraer una muestra aleatoria de tamaño 4.332, que corresponde a las dos terceras partes del total de datos, como el conjunto de entrenamiento, y los 2.165 restantes como el conjunto de prueba. La proporción de malos clientes en la muestras de entrenamiento y de prueba son de 12,3 % y de 11,4 %, respectivamente.

Hubiese sido ideal tener datos correspondientes al año 2003 en el conjunto de prueba, puesto que esto hubiese permitido crear el modelo de credit scoring con datos solo del 2000 que incorporen todos los comportamientos que se dan en un año y

pronosticar el comportamiento de los individuos del año 2003.

Durante el desarrollo de la aplicación, se observo que algunas técnicas de clasificación no tenían inconveniente con la muestra de entrenamiento con proporciones reales de buenos y malos pagadores; para otras técnicas, esto fue un inconveniente en el proceso de discriminación puesto que los casos de buenos pagadores eran al menos siete veces los casos malos pagadores. Así, con las técnicas que presentaban inconvenientes, se decidió trabajar con una muestra de entrenamiento balanceada, una que tenga la misma proporción de buenos y malos pagadores, con el propósito de reconocer las características de los malos pagadores. La muestra balanceada se logró luego de tomar una muestra aleatoria de 529 casos de un total de 4.310 buenos pagadores, junto con los 529 malos pagadores; en definitiva, la muestra balanceada tiene un total de 1.058 casos. También, se decidió mantener el mismo conjunto de prueba para todas las técnicas investigadas, para que los modelos resultantes sean comparables de acuerdo al error de clasificación obtenido.

4.4. Preparación de los datos

Debido a que el conjunto original de datos proviene de una base de datos histórica, es necesario iniciar un proceso de preparación de los datos para poder proceder a construir un modelo de credit scoring óptimo.

El primer paso de la preparación de los datos que se realizó fue tomar una desición respecto a los registros del mismo individuo en diferentes créditos. Considerando que el modelo de credit scoring debe reflejar en lo posible la actual capacidad de pago de un crédito, se decidió considerar solamente la información del último crédito y descartar la información de otros créditos de un mismo individuo.

El conjunto de datos resultante del proceso descrito en el párrafo anterior presenta valores perdidos en las características consideradas, por tanto se realizó un proceso de imputación de los datos. Como se consideró que el proceso de imputación es fundamental en el desarrollo del modelo de credit scoring, se decidió imputar solamente el conjunto de entrenamiento y no el de prueba. La definición de los conjuntos de prueba y entrenamiento se detalla en la Sección (4.3).

El tamaño original del conjunto de entrenamiento fue de 4.332 registros, que finalmente quedó en 4.310 registros, puesto que se decidió excluir del estudios a los casos que tenían el valor de 0 en las variables del ingreso y gasto, por considerar que estas variables son fundamentales en el proceso de otorgamiento de créditos.

También, se decidió que aquellos ingresos de 0 correspondientes a gastos positivos debían ser considerados como nulos en el proceso de imputación, para que este proceso ponga valores adecuados de acuerdo a la interacción global de las variables. Así mismo, se procedió con los valores de tiempo de residencia superiores a la edad del solicitante.

Aquellos valores de las variables considerados extremos, más de 1.5 veces el rango intercuartílico, también se les consideró como nulos para que el proceso de imputación ponga valores adecuados de acuerdo a la correlación global de las variables. Los valores extremos de las variables edad, tres, mdes, ting, tgas son 66, 40, 4.300, 520 y 535 respectivamente.

En el proceso de imputación de los datos se utilizó el paquete *mix* del software *R*. Este paquete realiza las imputaciones tanto de variables cualitativas y cuantitativas utilizando las técnicas EM y aumentación de datos, suponiendo el modelo de ubicación general. Este modelo supone que las variables categóricas y continuas siguen un modelo logístico y normal multivariado, respectivamente. Así, es conveniente transformar las variables continuas sesgadas antes del proceso de imputación.

Las figuras (4.1) y (4.2) muestran las distribuciones de los datos crudos e imputados, respectivamente. Estas dos figuras dan una justificación por lo menos gráfica de que el conjunto de datos imputados puede reemplazar al conjunto de datos crudos en el desarrollo del modelo de credit scoring.

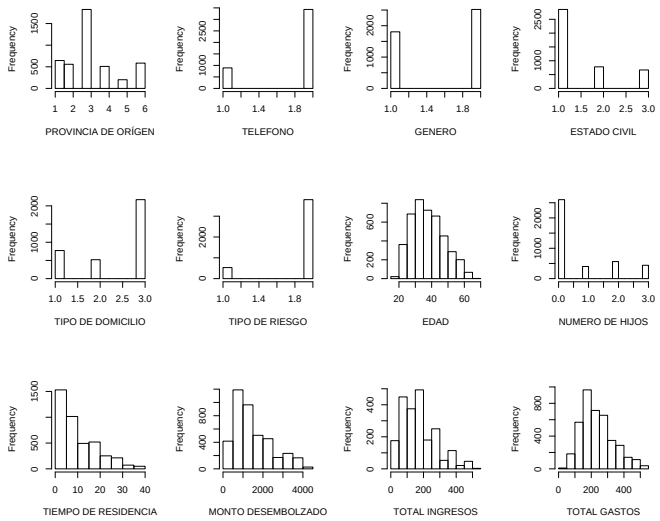


Figura 4.1: Distribuciones de los datos crudos

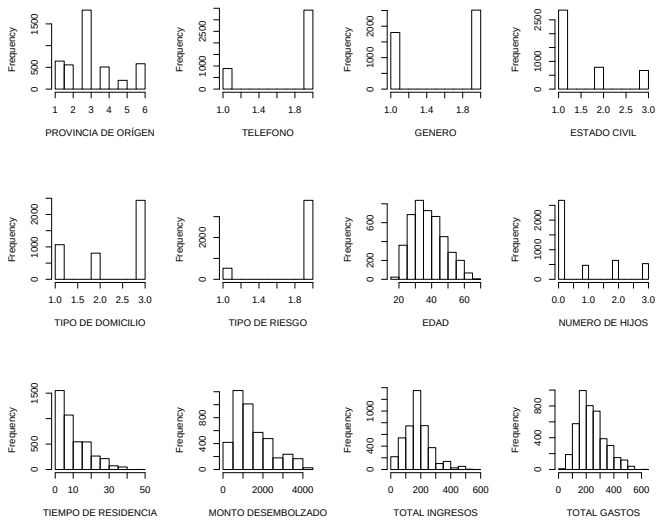


Figura 4.2: Distribuciones de los datos imputados

Para tener un sustento técnico que decida sobre la idoneidad del conjunto de datos imputados se realizó pruebas Mann-Whitney con las variables continuas edad, tres, ting y tgas. Los valores p de estas pruebas son 0.1446, 0.3678, 0.3763 y 0.9455, respectivamente; indicando que se ha realizado una imputación adecuada.

Es importante mencionar en este momento que las variables continuas serán estandarizadas cuando sean consideradas en el ajuste de los modelos, básicamente debido a la diferencia de escalas.

Finalmente, se realizó una depuración del conjunto de prueba debido a que se planificó de antemano no imputar este conjunto. Los registros con valores perdidos no serán considerados para evaluar el desempeño de los modelos elaborados; esto nos limita a que el tamaño del conjunto de prueba se reduzca a 1.559 individuos.

4.5. Definición de riesgo aceptable

Antes de comenzar con la construcción de modelos de credit scoring es de vital importancia establecer el significado que tiene para la institución financiera el término riesgo aceptable.

En el ambiente financiero, típicamente, un individuo se dice que es de riesgo aceptable cuando ha fallado a lo sumo en reembolzar dos pagos consecutivos. En términos generales, un individuo es considerado “no riesgoso” cuando no ha sido

severamente moroso y ha mostrado ser rentable.

La definición anterior solamente es un ejemplo, y varias definiciones son posibles. Esta definición debe ser realizada por los representantes de la institución financiera y comunicada al desarrollador del modelo de credit scoring. Se debe tener en cuenta que diferentes definiciones pueden producir diferentes modelos de credit scoring. Sin embargo, esto no significa que los resultados serán diferentes. En realidad, diferentes definiciones pueden generar diferentes modelos pero las aplicaciones aceptadas y rechazadas son similares.

Para esta aplicación en particular, se considera, unilateralmente, que un cliente es "riesgoso" cuando ha caído en mora por más de ocho días en un crédito en particular y "no moroso" caso contrario. Para los primeros se utiliza la etiqueta B y para los segundos la etiqueta G. Se considera esta definición puesto que en algunas tercerizadoras de cobranza ecuatorianas un cliente pasa a la gestión de cobranza luego de ocho días de mora.

Con la definición mencionada, se procedió a comparar gráficamente a los clientes riesgosos y no riesgosos para tener una idea inicial de sus características. La Figura (4.3) muestra la comparación gráfica de los dos conjuntos en la variables categóricas

En la Figura (4.3) se observa que las variables género, número de hijos, posesión de línea telefónica tiene un comportamiento similar en los dos grupos así como en la población. Se puede observar también que los solteros, los que su vivienda es arrendada, los residentes en Pichincha tienden a ser más riesgosos.

Para tener una idea sobre la dependencia de la variable riesgo y las variables categóricas se realizó una prueba Ji-cuadrado con cada una de ellas. Los valores p que se obtuvieron con las variables prov, tele, sexo, esci, tido y nhij son 0.001, 0.584, 0.939, 0.050, 0.000, 0.689, respectivamente. Según este estadístico las variables provincia, estado civil y tipo de domicilio y la variable riesgo muestran una dependencia estadística.

La Figura (4.4) muestra la comparación gráfica en las variables continuas y la Figura (4.5) muestra la comparación de los dos grupos mediante los valores descriptivos mínimo, máximo y valor medio. Se puede observar que los solicitantes riesgosos tienen un promedio de edad de 36 años y gastan en promedio 225 dólares y los no riesgosos son en promedio de de 38 años y gastan 247 dólares.

Se puede observar en la Figura (4.4) que la edad promedio de los clientes no riesgosos es un poco superior a la edad de los riesgosos. Al parecer, individualmente, el número de hijos, el tiempo de residencia, el monto desembolsado, los ingresos y gastos

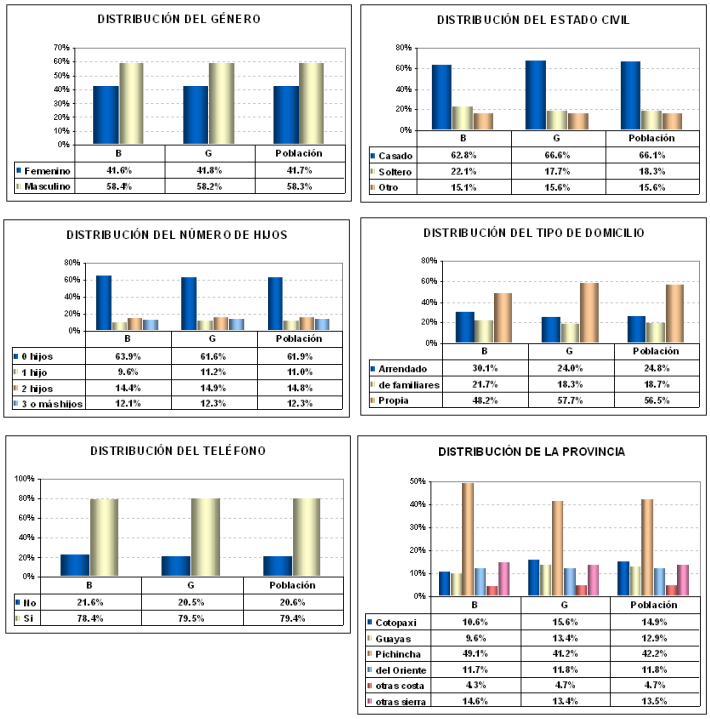


Figura 4.3: Distribución de las variables categóricas

totales tienen un comportamiento similar en los dos grupos en consideración, B y G. Esto sugiere que individualmente las variables no son los suficientemente capaces de discriminar entre los dos grupos, siendo necesario recurrir a combinaciones de estas para lograr el objetivo de discriminación.

4.6. Análisis discriminante

De acuerdo con lo expuesto en la Sección (3.1), el análisis discriminante utiliza variables continuas y categóricas, siempre y cuando las variables categóricas se conviertan en continuas creando variables dicotómicas auxiliares.

Es importante notar que si en el análisis discriminante se incluyen variables categóricas, los supuestos de normalidad no se dan y no es de utilidad realizar pruebas de estos supuestos, y mas aún, la selección de variables expuesta en la Sección (3.1) dejaría de tener sustento.

En el proceso de búsqueda del mejor modelo discriminante lineal para los datos disponibles se utilizaron combinaciones de las variables continuas y categóricas explicativas. Los errores resultantes en el conjunto de prueba y las combinaciones

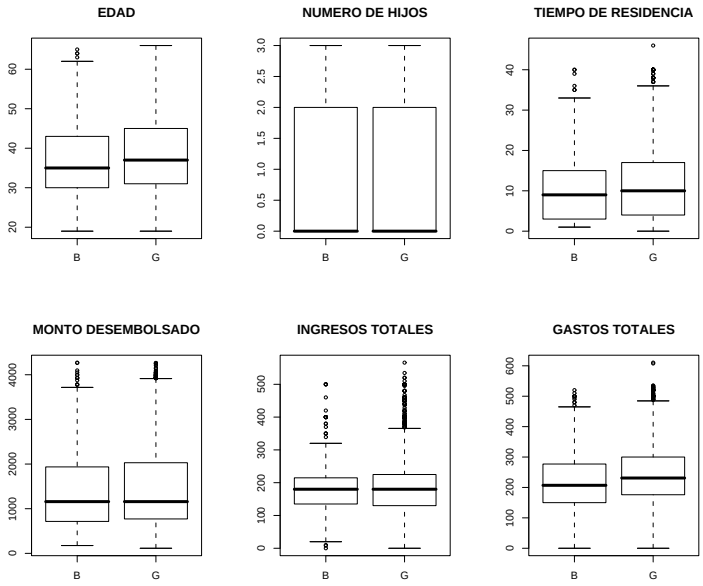


Figura 4.4: Comparación gráfica de las características de los clientes G y B.

	Riesgo						Población		
	B			G					
	Mínimo	Media	Máximo	Mínimo	Media	Máximo	Mínimo	Media	Máximo
Edad	19	36.8	65	19	38.4	66	19	38.2	66
Tiempo de residencia	1	10.7	40	0	11.1	46	0	11.1	46
Total de ingresos	0	179.2	500	0	187.1	566.1	0	186.2	566.1
Total de gastos	0	224.6	520	0	246.6	610.4	0	243.9	610.4
Monto solicitado	175.1	1448.9	4271.3	114.2	1486.0	4264.0	114.2	1481.4	4271.3

Figura 4.5: Descriptivos de las variables continuas

tratadas se muestran en el Cuadro (4.2).

Analizando el Cuadro (4.2) podemos observar que solamente las variables continuas logran generar modelos lineales de clasificación superiores a una clasificación al azar y que la inclusión de variables categóricas por medio de variables dicótomas no fueron de utilidad en esta ocasión. De los modelos de clasificación que incluyen solamente variables continuas, podemos destacar aquel que utiliza las variables ting y tgas, con el menor porcentaje de error en la muestra de prueba, 41,8 %.

Hay que notar, que con el mejor modelo obtenido hasta el momento, el que incluye las variables ting y tgas, se ha logrado reducir el error en aproximadamente un 8 % respecto de una clasificación al azar. Esta disminución puede parecer no tan significativa a simple vista, pero cuando se trata de un gran volumen de solicitudes, pequeños cambios pueden significar un gran resultado.

Variables incluidas en los modelos de clasificación											
Variables categóricas						Variables continuas					
prov	tele	sexo	esciv	tido	nhij	edad	tres	mdes	ting	tgas	error
						x	x	x	x	x	48,2 %
						x			x	x	48,0 %
									x	x	41,8 %
						x	x	x			48,7 %
							x	x			44,4 %
							x	x	x	x	44,2 %
						x	x				47,8 %
							x		x	x	43,0 %
x	x	x	x	x	x	x	x	x	x	x	-
x	x	x	x	x	x				x	x	-
	x		x	x			x		x	x	-
x			x	x		x					-

Cuadro 4.2: Errores en la muestra de prueba logrados con modelos discriminantes lineales basados en la muestra balanceada. - significa que el error obtenido no es mejor que el del clasificador al azar.

Para buscar el mejor modelo discriminante cuadrático se utilizó los datos disponibles en el conjunto de prueba y combinaciones de variables continuas y categóricas. Los errores en el conjunto de prueba resultantes de los modelos probados se muestran en el Cuadro (4.3).

Variables incluidas en los modelos de clasificación											
Variables categóricas						Variables continuas					
prov	tele	sexo	esciv	tido	nhij	edad	tres	mdes	ting	tgas	error
						x	x	x	x	x	-
									x	x	-
						x			x	x	-
						x	x	x			-
x	x	x	x	x	x	x	x	x	x	x	42,8 %
x	x	x	x	x			x		x	x	43,1 %
x	x	x	x	x					x	x	43,8 %
x			x	x		x			x	x	43,0 %
x			x	x		x					44,7 %

Cuadro 4.3: Errores en la muestra de prueba logrados con modelos discriminantes cuadráticos basados en la muestra balanceada. - significa que el error obtenido no es mejor que el del clasificador al azar.

El Cuadro (4.3) revela, en este caso específico, que la inclusión de variables categóricas a través de variables ficticias permite que los modelos discriminantes se desempeñen mejor que los modelos con solo variables continuas. Los mejores errores logrados oscilan alrededor del valor 43 %. El modelo discriminante cuadrático que debería ser elegido, sería aquel con el menor número de variables. Dicho modelo incluye a las variables prov, esciv, tido, edad, ting y tgas con un error en el conjunto de de

prueba de 43,0 %.

A pesar que los modelos discriminantes lineal y cuadrático son diseñados para variables continuas, se verificó que la inclusión de variables categóricas mediante variables dicotómicas auxiliares pueden incrementar el desempeño de estos modelos, claro está, violando los supuestos que los sustentan.

4.7. Análisis de Fisher

Los errores obtenidos utilizando las funciones discriminantes de Fisher se muestran en el Cuadro (4.4). Es evidente que las variables continuas no lograron superar la discriminación realizada al azar, mientras que la inclusión de solamente variables categóricas por medio de variables ficticias lograron mejorar la discriminación. El modelo de clasificación de Fisher con las variables categóricas sexo, esciv, tido y nhij tiene el menor error de clasificación, 40,7 %.

Variables incluidas en los modelos de clasificación											
Variables categóricas						Variables continuas					
prov	tele	sexo	esciv	tido	nhij	edad	tres	mdes	ting	tgas	error
						combinación de variables continuas					-
x	x	x	x	x	x	x	x	x	x	x	45,0 %
x	x	x	x	x	x						42,9 %
	x	x	x	x	x						41,9 %
		x	**x**	**x**	**x**						40,7 %
			x	x	x						43,4 %

Cuadro 4.4: Errores en la muestra de prueba logrados con modelos de funciones discriminantes de Fisher basados en la muestra balanceada. - significa que el error obtenido no es mejor que el del clasificador al azar.

La librería *MASS* de *R* posee la función *lda* que permite realizar el análisis por medio de las funciones discriminantes canónicas.

4.8. Regresión Logística

Los errores obtenidos en la muestra de prueba con los modelos de regresión y las variables utilizadas se muestran en el Cuadro(4.5). Como se puede observar, la combinación de solamente variables cuantitativas no producen resultados mejores que una clasificación al azar. Es notable que la inclusión de variables categóricas en los modelos de regresión logística mejoran los errores. El modelo de regresión logística con

el menor error de clasificación en la muestra de prueba es aquel que solamente involucra a la variables tido y ting, con un error de 35,5 % de error.

Variables incluidas en los modelos de clasificación											
Variables categóricas						Variables continuas					
prov	tele	sexo	esciv	tido	nhij	edad	tres	mdes	ting	tgas	error
						combinación de variables continuas					-
x	x	x	x	x	x	x	x	x	x	x	45,1 %
x	x	x	x	x	x	x	x	x			43,9 %
				x	x			x	x	x	41,8 %
				x				x	x	x	40,1 %
				x					x	x	39,9 %
				x					**x**		35,3 %

Cuadro 4.5: Errores en la muestra de prueba logrados con modelos de regresión logística basados en la muestra balanceada. - significa que el error obtenido no es mejor que el del clasificador al azar.

La librería *VGAM* de *R* posee la función *vglm* que permite ajustar un modelo de regresión logística múltiple utilizando el último factor como referencia. Para esto se especifica que el parámetro familia de la función sea multinomial.

4.9. Árboles de Clasificación

Se observó que este método de clasificación es altamente dependiente de la selección de la muestra, es decir, del tipo de información que comprende el conjunto de entrenamiento.

Los modelos de árboles de clasificación estimados con el conjunto de entrenamiento que mantiene la proporción real de los clientes buenos y malos no fueron capaces de reconocer a los malos clientes puesto que se perdió las características de estos. En cambio, los modelos estimados con el conjunto de entrenamiento en el que la proporción de buenos y malos clientes es la misma fueron de mejor la clasificación lograda por el azar significativamente.

La libería *tree* de *R* posee la función *tree* que permite obtener árboles de clasificación mediante particionamiento recursivo. La medida de impureza que se puede utilizar con esta función es el índice de Gini o deviance .

La elección de los parámetros que permiten a un árbol crecer es crucial. Para que un árbol crezca se debe especificar el tamaño de los nodos candidatos a dividir, el tamaño que deberían tener los nodos hijos, la mínima medida de impureza que se desea, y por su puesto, la medida de impureza misma. La función *tree* permite la selección

de las medidas de impureza *gini* o *deviance* que es equivalente a la entropía detallada en la Sección (3.4). Los modelos basados en el índice de *deviance* mostraron un mejor desempeño que los basados en el índice de gini. La menor medida de impureza que se utilizó para criar a los árboles estudiados fue del 0 % y se especificó mediante el parámetro *mindev* igual a cero. Se consideró conveniente que el tamaño que debería tener un nodo candidato a división debía de ser del 5 % de los datos considerados, y se especificó mediante el parámetro *minsize* igual a 50. El menor tamaño que un nodo hijo debería se consideró que debía de ser la mitad de un nodo candidato adividir, es decir, de 2,5 %, y se especifico mediante el parámetro *mincut* igual a 25.

Con los parámetros mencionados en el párrafo anterior se formó un árbol con 34 nodos terminales, 5 de los cuales realizaban el mismo pronóstico. Para tener un árbol con menos nodos terminales y con el mismo desempeño se utilizó la función *snip.tree* que permite excluir los nodos especificados del árbol. También se utilizó la función *prune.tree* para podar árboles

En árbol que mejor desempeño mostró tiene 27 nodos terminales y un error de clasificación en la muestra de prueba del 35,5 % y fue resultado de utilizar la función *snip.tree* solamente. Para efecto de exposición se muestra el mejor árbol creado en la Figura (4.6), pero se debe recordar que los modelos de clasificación son modelos predictivos y no explicativos, es decir, las reglas que se obtienen del árbol, mostradas en el siguiente párrafo, no son descriptivas de los buenos y malos pagadores.

Las reglas obtenidas por el árbol de clasificación con el mejor desempeño y los datos particulares de cada nodo terminal se muestran a continuación tal y como se obtienen en el software utilizado,

```
node), split, n, deviance, yval, (yprob)
      * denotes terminal node

   1) root 1058 1467.00 B ( 0.5000 0.5000 )
     2) ting < 237.306 857 1185.00 B ( 0.5298 0.4702 )
       4) tgas < 234.097 512  697.90 B ( 0.5762 0.4238 )
         8) prov: Pichincha,Otras Costa 245  317.60 B ( 0.6490 0.3510 )
          16) edad < 43.5 177  216.10 B ( 0.7006 0.2994 )
            32) ting < 148.903 66   62.59 B ( 0.8182 0.1818 ) *
            33) ting > 148.903 111  146.20 B ( 0.6306 0.3694 )
              66) mdes < 1136.33 60   83.18 B ( 0.5000 0.5000 )
               132) esci: Casado 34   46.07 B ( 0.5882 0.4118 ) *
               133) esci: Soltero,Otro 26   34.65 G ( 0.3846 0.6154 ) *
              67) mdes > 1136.33 51   53.18 B ( 0.7843 0.2157 ) *
          17) edad > 43.5 68   94.21 B ( 0.5147 0.4853 )
            34) sexo: Mujer 36   48.11 B ( 0.6111 0.3889 ) *
            35) sexo: Hombre 32   43.23 G ( 0.4063 0.5938 ) *
         9) prov: Cotopaxi,Guayas,Oriente,Otras Sierra 267  370.00 B ( 0.5094 0.4906 )
          18) ting < 207.853 239  330.80 G ( 0.4770 0.5230 )
            36) tgas < 129 59   79.73 B ( 0.5932 0.4068 )
              72) edad < 33.5 32   44.24 G ( 0.4688 0.5313 ) *
              73) edad > 33.5 27   30.90 B ( 0.7407 0.2593 ) *
            37) tgas > 129 180  246.80 G ( 0.4389 0.5611 )
```

```
          74) tido: de Familiar 27   35.59 B ( 0.6296 0.3704 ) *
          75) tido: Arrendada,Propia 153  206.60 G ( 0.4052 0.5948 )
           150) mdes < 657.067 33   38.67 G ( 0.2727 0.7273 ) *
           151) mdes > 657.067 120  164.70 G ( 0.4417 0.5583 )
             302) mdes < 1741.21 94  130.30 G ( 0.4894 0.5106 )
               604) mdes < 1202.52 68   92.79 G ( 0.4265 0.5735 )
                1208) mdes < 945.075 43   59.40 B ( 0.5349 0.4651 ) *
                1209) mdes > 945.075 25   27.55 G ( 0.2400 0.7600 ) *
               605) mdes > 1202.52 26   33.54 B ( 0.6538 0.3462 ) *
             303) mdes > 1741.21 26   30.29 G ( 0.2692 0.7308 ) *
      19) ting > 207.853 28   29.10 B ( 0.7857 0.2143 ) *
    5) tgas > 234.097 345  476.20 G ( 0.4609 0.5391 )
     10) ting < 129.672 82   99.14 G ( 0.2927 0.7073 ) *
     11) ting > 129.672 263  364.40 B ( 0.5133 0.4867 )
       22) prov: Cotopaxi,Pichincha,Oriente,Otras Costa 174  237.30 B ( 0.5747 0.4253 )
         44) tres < 12.5 122  162.60 B ( 0.6148 0.3852 )
           88) prov: Pichincha,Oriente 92  116.20 B ( 0.6739 0.3261 ) *
           89) prov: Cotopaxi,Otras Costa 30   41.05 G ( 0.4333 0.5667 ) *
         45) tres > 12.5 52   72.01 G ( 0.4808 0.5192 )
           90) tgas < 293.5 27   37.10 B ( 0.5556 0.4444 ) *
           91) tgas > 293.5 25   33.65 G ( 0.4000 0.6000 ) *
       23) prov: Guayas,Otras Sierra 89  119.30 G ( 0.3933 0.6067 )
         46) edad < 38.5 46   63.42 B ( 0.5435 0.4565 ) *
         47) edad > 38.5 43   46.64 G ( 0.2326 0.7674 ) *
  3) ting > 237.306 201  265.60 G ( 0.3731 0.6269 )
    6) tgas < 382.25 165  224.30 G ( 0.4182 0.5818 )
     12) sexo: Mujer 75   94.03 G ( 0.3200 0.6800 ) *
     13) sexo: Hombre 90  124.80 B ( 0.5000 0.5000 )
       26) tgas < 232.454 38   48.82 G ( 0.3421 0.6579 ) *
       27) tgas > 232.454 52   69.29 B ( 0.6154 0.3846 )
         54) mdes < 1706.94 27   37.39 G ( 0.4815 0.5185 ) *
         55) mdes > 1706.94 25   27.55 B ( 0.7600 0.2400 ) *
    7) tgas > 382.25 36   32.44 G ( 0.1667 0.8333 ) *
```

Observemos como en los nodos terminales 7, 10, 12, 47, 150, 303, 1209, que en definitiva son particiones del espacio definido por las variables utilizadas, al menos cerca de la tercera parte de los individuos que caen es estas regiones o nodos son etiquetados como de buen riesgo.

4.10. Redes neuronales

Se utilizó las variables continuas y categóricas en los modelos de redes neuronales. Los modelos con variables continuas mostraron mejor desempeño que los modelos que incluían los dos tipos de variables. Los modelos que resultaron óptimos en cuanto a error de clasificación se construyeron con la muestra balanceada, puesto que con la muestra normal no se podía reconocer a los malos clientes.

La librería *nnet* de *R* posee la función *nnet* que permite obtener una red neuronal de una sola capa oculta. En esta aplicación se utilizó un tamaño variado de nodos en la capa oculta y la red que se consideró óptima tiene cuatro nodos en la capa oculta.

En la función *nnet* es posible trabajar con el paraámetro de calibración o decay para básicamente no encontrar el vector de pesos óptimo global sino un

Figura 4.6: Árbol de clasificación con el mejor desempeño en la muestra de prueba.

óptimo local para evitar un sobreajuste de la red a los datos de entrenamiento. En los modelos considerados no se exigió que todas las neuronas estén conectadas, esto se lo especificó con el parámetro skip=T.

El modelo de red neuronal que se consideró óptimo tiene cuatro nodos en la capa oculta y provocó un error del 38,7 % en la muestra de prueba. Los pesos de esta red neuronal se muestra a continuación tal y como el programa utilizado lo muestra,

```
a 5-4-2 network with 44 weights
options were - skip-layer connections  softmax modelling
    b->h1    i1->h1    i2->h1    i3->h1    i4->h1    i5->h1
-12512.47   5503.73 -16601.61  23770.13  -6856.33 -12099.00
    b->h2    i1->h2    i2->h2    i3->h2    i4->h2    i5->h2
-14712.13  -3372.60  13942.53 -20520.07  35894.37 -32945.47
    b->h3    i1->h3    i2->h3    i3->h3    i4->h3    i5->h3
 15532.02  -1023.35   6425.35 -15503.21   5615.77  11230.25
    b->h4    i1->h4    i2->h4    i3->h4    i4->h4    i5->h4
-10602.57 -11692.02  27698.35 -15754.63  25335.42 -44983.59
    b->o1    h1->o1    h2->o1    h3->o1    h4->o1    i1->o1    i2->o1    i3->o1
    -1.28      0.68     -0.51      0.56      1.22      0.19     -0.11     -0.01
   i4->o1    i5->o1
     0.31     -0.32
    b->o2    h1->o2    h2->o2    h3->o2    h4->o2    i1->o2    i2->o2    i3->o2
     0.33     -0.56      0.84     -0.79     -0.27      0.29     -0.09      0.02
   i4->o2    i5->o2
     0.28     -0.15
```

4.11. Support vector machines

Los modelos support vector machines estimados en el conjunto de entrenamiento que mantiene las proporciones entre buenos y malos clientes no fueron capaces de reconocer a los malos clientes. En cambio, los modelos estimados a partir del conjunto de entrenamiento que tiene la misma proporción de individuos mostraron mejor desempeño que un clasificador al azar.

La librería *e1071* de *R* posee la función *svm* que permite ajustar modelos support vector machines. El núcleo utilizado en la presente aplicación es kernel radial, que por defecto dicha función lo toma en cuenta.

El modelo support vector machines con variables solamente continuas produjo un error del 44,7 % en la muestra de prueba. Se decidió por tanto incluir las variables categóricas mediante variables dicótomas, esto ayudó a mejorar el desempeño del modelo. El mejor modelo support vector machines co variables continuas y categóricas produjo un 39,5 % de error en la muestra de prueba. El número de vectores soporte en este modelo es de 986.

Capítulo 5

Conclusiones y recomendaciones

Se pudo constatar en el desarrollo de este trabajo que la piedra angular de los modelos de credit scoring es la noción que una persona o institución tiene sobre el significado de lo que es un cliente riesgoso. Por tal motivo es de vital importancia en el desarrollo de modelos de credit scoring definir el significado de cliente riesgoso. Por tal motivo es indispensable que el individuo o institución que requiere un sistema de credit scoring comunique al desarrollador de modelos el significado que para ellos tiene las palabras cliente riesgoso, para que luego en conjunto se pueda establecer una defínción de cliente riesgoso que no sea ambigua y que permita desarrollar este tipo de modelos.

Otro tema de vital importancia en el desarrollo de modelos de credit scoring es la selección del conjunto de entrenamiento. En el desarrollo de este trabajo se cosntató que el desempeño (errores de clasificación) de los modelos de clasificación considerados son altamente dependientes de este conjunto. Por tanto se puede afirmar que para la estimación de modelos de credit scoring no necesariamente se debe utilizar un solo conjunto de entrenamiento, puesto que a la final lo que se está buscando es una regla de desición que permita que el proceso de aprobación de créditos sea transparente, rápido y con el menor riesgo de no pago. Se debe enfatizar que el conjunto de prueba debe ser único para que los errores sean comparables y permitan discernir el modelo de mejor desempeño con el conjunto de datos disponibles.

El desarrollo de este trabajo permite afirmar que el conocer adecuadamente los modelos de clasificación da ventajas al momento de la aplicación puesto que se puede interactuar de mejor manera con los programas diseñados para tareas de clasificación y se puede interpretar convenientemente los resultados arrojados por dichos programas.

En referencia a la aplicación, se puede mencionar que los esstadísticos y

gráficos obtenidos para los clientes buenos y malos no mostraban diferencias significativas, lo que habría echo pensar en la dificultad de encontrar una regla de desición que permita discriminarlos. A pesar de eso se puede ver que el modelo de regresión logística obtenido con las variables ting y tido produce un error del 35,3 % en el conjunto de prueba, es decir, clasifica adecuadamente aproximadamente 65 de 100 aplicaciones.

Para ahondar en las cualidades del mejor modelo de regresión logística, en la Figura (5.1) se muestra la curva ROC correspondiente. Se puede apreciar que para un amplio rango de valores de corte esta curva se diferencia notablemente de un clasificador al azar. Es notorio que se localiza en el sector medio entre un clasificador perfecto y uno al zar.

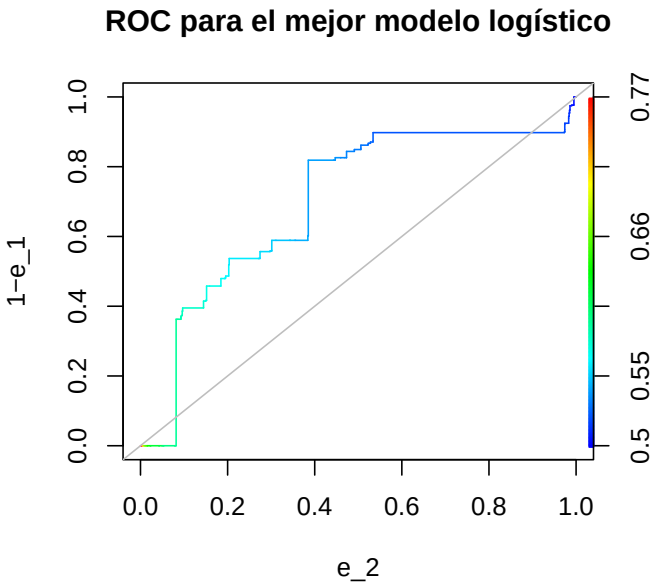


Figura 5.1: Curva ROC para el mejor modelo de regresión logística.

Para mejorar el desempeño de los modelos de clasificación específicamente en esta aplicación se debe primero, buscar información confiable de los individuos que han sido sujetos a crédito en la institución y, segundo, obtener una mayor cantidad de variables continuas y categóricas que complemente a las doce disponibles en esta aplicación con la esperanza que ellas permitan realizar una discriminación más acertada.

Las herramientas de clasificación examinadas en este estudio no son todas las existentes, y se recomienda ahondar sobre el tema. Como sugerencia personal, creo que sería valioso explorar la técnica de análisis discriminante flexible debido al echo

que aparece en recientes referencias.

Finalmente, me permito sugerir, para complementar el tema de credit scoring, el tema de behavioral scoring que básicamente analiza el comportamiento de los clientes de una institución en lo referente a hábitos de consumo y pago de deudas.

En el presente, el énfasis es cambiar el objetivo de tratar de minimizar la posibilidad de que un cliente sea moroso en un producto en particular por el objetivo de buscar cómo la firma puede maximizar las ganancias que se pueden obtener de ése cliente. Más aún, la idea original de estimar el riesgo de mora ha sido aumentado por scorecards que estiman respuestas (¿cuán probable es que un consumidor responda a un direct mailing de un nuevo producto?), uso (¿cuàn probable es que un consumidor utilice un producto?), retención (¿cuán probable es que un consumidor continue utilizando un producto luego de que el periodo de oferta introductoria terminó?), attrition (¿se cambiará el cliente a otro prestamista?), administración de deudas (si el consumidor comienza a fallar al préstamo , cuán probable es que varios acercamientos prevengan caer en mora), y scoring de fraude (¿cuán probable es que la aplicación sea fraudulenta?).

Capítulo 6

Apéndice

6.1. Reject Inference - Estudio de los Rechazados

Cómo se puede utilizar la información parcial disponible de los rechazados para mejorar el sistema de scoring?

La forma más básica de lidiar con el sesgo de la muestra es construir una muestra en la cual ninguno fue rechazado. Minoristas y compañías de pedidos por correo tradicionalmente hacen esto. Consideran a cada uno que aplica en un cierto periodo con la intención de utilizar esa muestra para construir la próxima generación de scorecards. La cultura de las organizaciones financieras no acepta esta solución. "Hay algunos malos allá afuera y nosotros simplemente no podemos considerarlos" es un argumento, sin embargo, por supuesto, la gran pérdida involucrada en alguien que va mal en un préstamo personal, comparado con no pagar por dos libros ordenados, puede tener algo que ver. Sin embargo, este método tiene considerable mérito y puede ser modificado para direccionar la inquietud que costará demasiado en pérdidas. Tradicionalmente los bancos toman la probabilidad de mora como su criterio puesto que asumen que las pérdidas de los morosos no tienen grandes variaciones. En este caso, se puede disminuir estas pérdidas no tomando a cada uno pero tomando una proporción $p(x)$ de aquellos cuya probabilidad de mora es x. Uno deja que esta proporción varíe con x; es muy pequeña cuando x es casi 1, y tiende a 1 cuando x es muy pequeña. Permitiendo la posibilidad de seleccionar a cada uno, reponderando permitiría reconstruir una muestra sin rechazados sin tener que contraer las pérdidas totales tal como las muestras traen. Sin embargo este método no ha tenido mucha acogida entre los otorgadores de crédito, algunos tratan de obtener información del desempeño de los rechazados de otros otorgadores de crédito que dieron a estos clientes crédito.

Si uno tiene rechazados en la muestra a partir de la cual se construirá el sistema, entonces hay cinco formas que han sido sugeridas para lidiar los rechazados.

6.1.1. Definirlos como malos

El método más ligero es asignar el status de malo a todos los rechazados sobre la base de que debe haber habido mala información sobre ellos para que hayan sido rechazados anteriormente. El sistema de scoring entonces es construido utilizando esta clasificación total. Los problemas con este método son obvios. Una vez que algunos grupos de clientes potenciales han sido dados una clasificación de malos, no importa cuán erradamente, ellos nunca tendrán la oportunidad de refutar esta suposición. Es un método errado desde el punto de vista estadístico y ético.

6.1.2. Extrapolación

Se pueden presentar dos situaciones diferentes dependiendo de la relación entre las características X_{old} del sistema, que fue utilizado para tomar la decisión de aceptar o rechazar, y las características disponibles para construir el nuevo scorecard X_{new}. Si X_{old} es un subconjunto de X_{new}, es decir, las nuevas características incluyen a todas aquellas que fueron utilizadas en la clasificación original, entonces para alguna combinación de atributos (aquellos donde X_{old} rechazaron al aplicante), no conoceremos nada sobre la división bueno-malo puesto que todos fueron rechazados. Para las otras combinaciones, donde X_{old} aceptó a los aplicantes, deberíamos tener información sobre la proporción de buenos y malos pero solo entre los aceptados. Sin embargo, X_{old}, aceptaría todos los aplicantes con esa combinación de características. Entonces se tiene que extrapolar, es decir, ajustar un modelo para todas la probabilidades de ser bueno para las combinaciones de atributos que fueron aceptadas y extender este modelo a las combinaciones que fueron previamente rechazadas. Este método funciona mejor en métodos que estiman $q(G|x)$ directamente, como la regresión logística, en lugar de los que estiman $p(x|G)$ y $p(x|B)$, como el análisis discriminante. Esto es porque cuando estimamos $p(x|G)$, la fracción muestral está variando con x así que sesgará cuando se trata de estimar los parámetros de $p(x|G)$. Sin embargo, para $q(x|G)$, la fracción de la población subyacente con ese valor de x que es muestreado es 0 o 1, así que no hay sesgo en la estimación de los parámetros del modelo. Lo que pasa es que el modelo da una probabilidad de ser bueno a cada miembro de la población que fue rechazada y los sistemas de scoring son construidos contada la población con los rechazados asignados

este valor. Esto no funciona para métodos como los vecinos más cercanos, pero puede funcionar para la regresión logística si se cree en la forma de este modelo.

6.1.3. Aumentación

Si X_{old} no es un subconjunto de X_{new}, habría variables desconocidas o razones de las decisiones originales de rechazo, entonces la situación sería más complicada. Primero uno construye un modelo de buenos-malos utilizando solo la población aceptada para estimar $p(G|x, A)$, la probabilidad de ser bueno si fue aceptado y con valores x en las características. Luego se construyeun modelo aceptado-rechazado utilizando técnicas similares para obtener $p(A|x) = p(A|s(x)) = p(A|s)$, donde s es la marca de aceptación rechazo. El modelo original supone que $p(G|s, R) = p(G|s, A)$, la probabilidad de ser bueno es la misma entre los aceptados y los rechazados al mismo nivel de la marca s, donde

Esto es como repesar la distribución de la población muestal tal que el porcentaje con un score s pase de $p(A, s)$ a $p(s)$. Se construye un nuevo scorecard bueno-malo con toda la muestra incluyendo los rechazados. Los rechazados con score s de aceptado-rechazado tienen la probabilidad $p(G|s, A)$ se ser buenos.

6.1.4. Mezcla de distribuciones

Si se está haciendo suposiciones, un método alternativo es decir que la población es una mezcla de dos distribuciones, una para buenos y otra para malos, y que la forma de estas distribuciones es conocida. Por ejemplo, si $p(x)$ es la proporción de aplicantes con característica x, uno dice que

$$p(x) = p(x|G)p_G + p(x|B)p_B$$

y uno puede estimar los parámetros de $p(x|G)$ y $p(x|B)$ utilizando los aceptados y, utilizando el algoritmo EM, incluso los rechazados.

6.1.5. Tres grupos

Un método es clasificar la muestra en tres grupos: buenos, malos y rechazados. El problema es que queremos utilizar el sistema de clasificación para dividir a los futuros aplicantes en solo dos clases, los buenos, que serán aceptados, y los malos, que serán rechazados. Lo que uno hace con los que son clasificados como rechazados no está claro.

Si uno los rechaza, este método reduce a clasificar a todos los rechazados como malos. En análisis discriminante lineal, cuando se clasifica en tres grupos, se asume que los tres grupos tienen matriz de covarianza común. Por tanto esta es una forma de utilizar la información de los rechazados para mejorar la estimación de la matriz de covarianza.

Para resumir, parece que reject inference es válida solo si se puede hacer suposiciones confiables sobre las poblaciones de aceptados y rechazados. Puede funcionar en la práctica puesto que estas suposiciones pueden algunas veces ser razonables o al menos se mueven en la dirección correcta.

6.2. Comparación estadística de modelos de decisión

Dos modelos de decisión son usualmente comparados estadísticamente mediante sus porcentajes de error en la muestra de prueba.

Si hay n objetos en la muestra de prueba, el porcentaje de error estimado es

$$\widehat{P}_{\text{err}} = \frac{\text{Número de clasificaciones incorrectas}}{n}.$$

Sea la variable aleatoria Y definida como

$$Y_i = \begin{cases} 1 & \text{si el i-ésimo objeto es clasificado incorrectamente} \\ 0 & \text{si el i-ésimo objeto es clasificado correctamente} \end{cases},$$

entonces, la media y la varianza de Y_i son

$$\begin{aligned} E[Y_i] &= P_{\text{err}} \\ V[Y_i] &= E[Y_i]^2 - (E[Y_i])^2 = P_{\text{err}}(1 - P_{\text{err}}). \end{aligned}$$

Como $\widehat{P}_{\text{err}} = \sum_{i=1}^{n} Y_i/n$ y por el teorema del límite central, $\widehat{P}_{\text{err}}$ está distribuida normalmente con media P_{err} y varianza $P_{\text{err}}(1 - P_{\text{err}})/n$.

Suponga que dos modelos de decisión notados por A y B son comparados en la misma muestra de prueba. Entonces,

$$Z = \frac{(\widehat{P}_{\text{Aerr}} - \widehat{P}_{\text{Berr}}) - \mu_{\widehat{P}_{\text{Aerr}} - \widehat{P}_{\text{Berr}}}}{\left(V[\widehat{P}_{\text{Aerr}} - \widehat{P}_{\text{Berr}}]\right)^{1/2}},$$

sigue una distribución normal estandar. Como

$$\begin{aligned} \widehat{P}_{\text{Aerr}} &= \tfrac{1}{n} \textstyle\sum_{i=1}^{n} Y_i^A \\ \widehat{P}_{\text{Berr}} &= \tfrac{1}{n} \textstyle\sum_{i=1}^{n} Y_i^B \end{aligned},$$

se obtienen de la misma muestra de prueba, son dependientes y es necesario realizar una comparación pareada. En este caso la varianza del estimador de la diferencia es

$$V[\widehat{P}_{\text{Aerr}} - \widehat{P}_{\text{Berr}}] = \frac{1}{n} V[Y_i^A - Y_i^B],$$

donde

$$V[Y_i^A - Y_i^B] = \frac{1}{n}\sum_{i=1}^{n}\left[(Y_i^A - Y_i^B) - (\widehat{P}_{\text{Aerr}} - \widehat{P}_{\text{Berr}})\right]^2.$$

Por tanto, el valor Z es

$$Z = \frac{(\widehat{P}_{\text{Aerr}} - \widehat{P}_{\text{Berr}}) - \mu_{\widehat{P}_{\text{Aerr}} - \widehat{P}_{\text{Berr}}}}{\frac{1}{n}\left(\sum_{i=1}^{n}\left[(Y_i^A - Y_i^B) - (\widehat{P}_{\text{Aerr}} - \widehat{P}_{\text{Berr}})\right]^2\right)^{1/2}}.$$

Como la hipótesis nula en esta sección es

$$H_0 = \mu_{\widehat{P}_{\text{Aerr}} - \widehat{P}_{\text{Berr}}} = 0,$$

es decir, no hay diferencia en el desempeño de los modelos de decisión A y B, el estadístico de prueba conveniente es

$$Z = \frac{(\widehat{P}_{\text{Aerr}} - \widehat{P}_{\text{Berr}})}{\frac{1}{n}\left(\sum_{i=1}^{n}\left[(Y_i^A - Y_i^B) - (\widehat{P}_{\text{Aerr}} - \widehat{P}_{\text{Berr}})\right]^2\right)^{1/2}}.$$

Podemos observar que cuando el valor de n es grande, se obtiene un valor grande de Z, causando que pequeñas diferencias se consideran significantes.

6.3. Prueba de normalidad multivariada

Existen varias pruebas cuando deseamos averiguar si un conjunto de observaciones $x_1, \ldots, x_n$, $x_i \in \mathbb{R}$, provienen de una distribución normal univariada. Seber [11] menciona que una de las pruebas más potentes se debe a Shapiro y Wilk que se conoce como la prueba W. El estadístico asociado es

$$W = \frac{\left(\sum_{i=1}^{n} a_i^{(n)} x_{(i)}\right)^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2}, \tag{6.1}$$

donde $x_{(1)} \leq \cdots \leq x_{(n)}$ son las observaciones ordenadas y los coeficientes $a_i^{(n)}$ son pesos que dependen del tamaño de la muestra. Cualquier desviación de la normalidad es detectada por un valor pequeño de W.

Para probar si un conjunto de observaciones $x_1, \ldots, x_n$, $x_i \in \mathbb{R}^p$, provienen de una distribución normal multivariada, Svantesson [12], menciona el estadístico H de Royston, que es una extensión multivariada del estadístico W de Shapiro y Wilk.

Sea W_j el estadístico W de la j-ésima variable, entonces defina

$$R_j = \left\{ \Phi^{-1} \left[\frac{1}{2} \Phi \left\{ - \left((1 - W_j)^{\lambda} - \mu \right) / \sigma \right\} \right] \right\}^2,$$

donde λ, μ y σ se obtienen utilizando expansiones polinomiales [12] y $\Phi(\cdot)$ es la función de distribución de la normal cero-uno.

Si las observaciones vienen de una distribución normal multivariada, es estadístico asociado a la prueba H

$$H = \frac{\xi \sum_{j=1}^{p} R_j}{p} \tag{6.2}$$

sigue aproximadamente una distribución $\mathbb{X}^2_{\widehat{\xi}}$ donde $\widehat{\xi} = p/[1 + (p-1)\overline{c}]$, donde $\overline{c}$ es un estimador de la correlación media entre los R_j.

6.4. Estimadores de máxima verosimilitud para poblaciones normales

Si $X_1, \ldots, X_n$ son i.i.d. como la distribución $N_d(\mu, \Sigma)$ y $n - 1 \geq d$, la función de verosimilitud L se define como la distribución conjunta de los X_i expresada como función de μ y Σ, esto es,

$$L(\mu, \Sigma) = (2\pi)^{-\frac{nd}{2}} |\Sigma|^{-\frac{n}{2}} \exp \left[-\frac{1}{2} \sum_{i=1}^{n} (X_i - \mu)' \Sigma^{-1} (X_i - \mu) \right]. \tag{6.3}$$

Seber ([11]) prueba que la media muestral $\overline{X}$ y la matriz de dispersión muestral $\frac{1}{n}Q = \frac{1}{n} \sum_{i=1}^{n} (X_i - \overline{X})'(X_i - \overline{X})$ son los estimadores de máxima verosimilitud de los parámetros μ y Σ, respectivamente. La función de verosimilitud en estos estimadores toma el valor

$$L(\widehat{\mu}, \widehat{\Sigma}) = (2\pi)^{-\frac{nd}{2}} |\widehat{\Sigma}|^{-\frac{n}{2}} e^{-\frac{nd}{2}}, \tag{6.4}$$

donde $\widehat{\mu}$ y $\widehat{\Sigma}$ son los estimadores de máxima verosimilitud.

6.5. Prueba de hipótesis: razón de verosimilitud generalizada

En la prueba de razón de verosimilitud generalizada se supone que se tiene n observaciones, $X_1, \ldots, X_n$, con densidad conjunta $f(X_1, \ldots, X_n|\theta)$. La hipótesis nula H_0 especifica que $\theta \in w_0$, donde w_0 es un subconjunto del conjunto de todos los valores posibles del vector θ, y la hipótesis alternativa H_1 supone que $\theta \in w_1$, donde w_1 es disjunto de w_0. Sea $\Omega = w_0 \cup w_1$. Entonces, el estadístico de la razón de verosimilitud generalizada es

$$\Lambda^* = \frac{\max\limits_{\theta \in w_0} L(\theta)}{\max\limits_{\theta \in w_1} L(\theta)},$$

donde $L(\theta)$ es la verosimilitud de las observaciones $X_1, \ldots, X_n$ que depende del vector θ. Valores pequeños de Λ^* favorecen a la hipótesis alternativa H_1.

Por motivos técnicos, es preferible utilizar el estadístico

$$\Lambda = \frac{\max\limits_{\theta \in w_0} L(\theta)}{\max\limits_{\theta \in \Omega} L(\theta)}, \tag{6.5}$$

en lugar de Λ^*. Note que $\Lambda = \text{mín}(\Lambda^*, 1)$, así valores pequeños de Λ^* corresponden a valores pequeños de Λ. La región de rechazo para esta prueba consiste de valores pequeños de Λ; por ejemplo, todos los $\Lambda \leq \lambda_0$.

Para que la razón de verosimilitud generalizada tenga un valor p, λ_0 debe elegirse tal que $P(\Lambda \leq \lambda_0) = p$ si H_0 es verdadera. Si la distribución muestral de Λ dado H_0 es conocida se puede determinar λ_0. Generalmente, la distribución muetral no es simple, entonces el siguiente teorema proporciona una aproximación de la distribución nula ([8]).

Teorema 6.5.1 *Bajo condiciones de suavidad de las densidades de probabilidad involucradas, la distribución nula de* $-2\log\Lambda$ *tiende a una distribución Ji cuadrada con* $dim(\Omega) - dim(w_0)$ *grados de libertad, cuando el tamaño de la muestra tiende al infinito.*

6.6. Comparación de dos poblaciones normales

Sean $v_1 \ldots, v_{n_1}$ una muestra aleatoria de la distribución $N_d(\mu_1, \Sigma_1)$ y $w_1 \ldots, w_{n_2}$ una muestra de la distribución $N_d(\mu_2, \Sigma_2)$, independiente de la anterior. Para probar la hipótesis nula $H_0 : \Sigma_1 = \Sigma_2 = \Sigma$ recurrimos a la prueba de razón de verosimilitud generalizada detallada en la Sección (6.5).

La función de verosimilitud de las muestras $v_1 \ldots, v_{n_1}$ y $w_1 \ldots, w_{n_2}$ es

$$L_{12}(\mu_1, \mu_2, \Sigma_1, \Sigma_2) = L_1(\mu_1, \Sigma_1) L_2(\mu_2, \Sigma_2).$$

donde $L_i(\mu_i, \Sigma_i)$ están dadas por la Ecuación (6.3). Maximizar L_{12} es equivalente a maximizar simultáneamente cada L_i; así, L_{12} se maximiza cuando $\mu_1 = \widehat{\mu}_1 = \overline{v}$, $\mu_2 = \widehat{\mu}_2 = \overline{w}$, $\Sigma_1 = \widehat{\Sigma}_1 = \frac{Q_1}{n} = \frac{1}{n}\sum_{i=1}^{n}(v_i - \overline{v})'(v_i - \overline{v})$ y $\Sigma_2 = \widehat{\Sigma}_2 = \frac{Q_2}{n} = \frac{1}{n}\sum_{i=1}^{n}(w_i - \overline{w})'(w_i - \overline{w})$ y la verosimilitud máxima, utilizando la Ecuación (6.4), es

$$\begin{aligned} L_{12}(\widehat{\mu}_1, \widehat{\mu}_2, \widehat{\Sigma}_1, \widehat{\Sigma}_2) &= L_1(\widehat{\mu}_1, \widehat{\Sigma}_1) L_2(\widehat{\mu}_2, \widehat{\Sigma}_2) \\ &= (2\pi)^{-nd/2} |\widehat{\Sigma}_1|^{-n_1/2} |\widehat{\Sigma}_2|^{-n_2/2} e^{-nd/2}, \end{aligned} \tag{6.6}$$

donde $n = n_1 + n_2$.

Haciendo $\Sigma_1 = \Sigma_2 = \Sigma$, Seber ([11]) probó que $L_{12}(\mu_1, \mu_2, \Sigma, \Sigma)$ es máxima cuando $\mu_1 = \widehat{\mu}_1 = \overline{v}$, $\mu_2 = \widehat{\mu}_2 = \overline{w}$ y $\Sigma = \widehat{\Sigma} = \frac{Q}{n} = \frac{Q_1 + Q_2}{n}$ y que la verosimilitud máxima es

$$L_{12}(\widehat{\mu}_1, \widehat{\mu}_2, \widehat{\Sigma}, \widehat{\Sigma}) = (2\pi)^{-nd/2} |\widehat{\Sigma}|^{-n/2} e^{-nd/2}. \tag{6.7}$$

Con las Ecuaciones (6.6) y (6.7) formamos el estadístico de razón de verosimilitud de acuerdo a la Ecuación (6.5.1)

$$\begin{aligned} \Lambda &= \frac{L_{12}(\widehat{\mu}_1, \widehat{\mu}_2, \widehat{\Sigma}, \widehat{\Sigma})}{L_{12}(\widehat{\mu}_1, \widehat{\mu}_2, \widehat{\Sigma}_1, \widehat{\Sigma}_2)} \\ &= \frac{|\widehat{\Sigma}|^{-n/2}}{|\widehat{\Sigma}_1|^{n_1/2} |\widehat{\Sigma}_2|^{n_2/2}} \\ &= c_{12} \frac{|Q_1|^{n_1/2} |Q_2|^{n_2/2}}{|Q_1 + Q_2|^{(n_1+n_2)/2}}. \end{aligned} \tag{6.8}$$

donde $c_{12} = \frac{n^{nd/2}}{n_1^{n_1 d/2} n_2^{n_2 d/2}}$.

El Teorema (6.5.1) asegura que para muestras grandes $-2\log\Lambda$ sigue una distribución $\mathbb{X}^2_v$ donde $v = \frac{1}{2}d(d+1)$, cuando H_0 es verdadera.

6.7. Comparación de I poblaciones normales

Para probar la hipótesis nula $H_0 : \Sigma_1 = \Sigma_2 = \ldots = \Sigma_I$ recurrimos nuevamente a la razón de verosimilitud generalizada. Si las muestras aleatorias Y_{ij} tomadas de las I poblaciones son i.i.d. como $N_d(\mu_i, \Sigma_i)$ tenemos que, de manera similar a la Ecuación

(6.8),

$$\begin{aligned}\Lambda &= \frac{|\widehat{\Sigma}|^{-n/2}}{|\widehat{\Sigma}_1|^{-n_1/2}|\widehat{\Sigma}_2|^{-n_2/2}\ldots|\widehat{\Sigma}_I|^{-n_I/2}} \\ &= \left(\prod_{i=1}^{I}|Q_i|^{n_i/2}n^{dn/2}\right)/\left|\sum_{i=1}^{I}Q_i\right|^{n/2}\prod_{I=1}^{I}n_i^{dn_i/2},\end{aligned} \tag{6.9}$$

donde $n_i\widehat{\Sigma}_i = Q_i = \sum_{j=1}^{n_i}(Y_{ij}-\overline{Y}_{i.})(Y_{ij}-\overline{Y}_{i.})'$ y $n = \sum_i n_i$.

Cuando H_0 es verdadera, por el Teorema (6.5.1), $-2\log\Lambda$ está distribuido como $\mathbb{X}^2_{v_1}$, donde $v_1 = \frac{1}{2}d(d+1)(I-1)$. Sin embargo, se puede obtener una mejor aproximación $\mathbb{X}^2$ haciendo que los grados de libertad de Q_i sean $f_i = n_i - 1$, entonces la prueba se convierte en insesgada y el estadístico resultante es

$$M = \left(\prod_{i=1}^{I}|Q_i|^{f_i/2}f^{df/2}\right)/\left|\sum_{i=1}^{I}Q_i\right|^{f/2}\prod_{I=1}^{I}f_i^{df_i/2}, \tag{6.10}$$

donde $f = \sum_i f_i = n - I$. Seber ([11]) menciona que para la mayoría de los casos prácticos es adecuado utilizar el echo de que $-2(1-c_1)\log M$ es aproximadamente $\mathbb{X}^2_{v_1}$, donde $v_1 = \frac{1}{2}d(d+1)(I-1)$ y $c_1 = \frac{2d^2+3d-1}{6(d+1)(I-1)}\left\{\sum_i f_i^{-1} - f^{-1}\right\}$.

6.8. Probabilidades de clasificación incorrecta

Para encontrar las probabilidades de asignación errónea debemos calcular

$$P(j|i) = P\left[\left[X^t - \frac{1}{2}(\mu_i+\mu_j)^t\right]\Sigma^{-1}(\mu_i-\mu_j) < \ln(\pi_j/\pi_i)|X\in G_i\right].$$

Como $X|X\in G_i \sim N_d(\mu_i,\Sigma)$, entonces $\left[X^t - \frac{1}{2}(\mu_i+\mu_j)^t\right]\Sigma^{-1}(\mu_i-\mu_j)$ también sigue una distribución normal multivariada de vector de medias

$$\begin{aligned}E\left[\left[X^t - \tfrac{1}{2}(\mu_i+\mu_j)^t\right]\Sigma^{-1}(\mu_i-\mu_j)\right] &= \left[\left[\mu_i^t - \tfrac{1}{2}(\mu_i+\mu_j)^t\right]\Sigma^{-1}(\mu_i-\mu_j)\right] \\ &= \tfrac{1}{2}(\mu_i-\mu_j)^t\Sigma^{-1}(\mu_i-\mu_j) \\ &= \tfrac{1}{2}\Delta^2,\end{aligned}$$

donde Δ^2 es la distancia de Mahalanobis al cuadrado entre μ_i y μ_j, y de matriz de varianza-covarianza

$$\begin{aligned}V\left[\left[X^t - \tfrac{1}{2}(\mu_i+\mu_j)^t\right]\Sigma^{-1}(\mu_i-\mu_j)\right] &= V\left[X^t\Sigma^{-1}(\mu_i-\mu_j) - \tfrac{1}{2}(\mu_i+\mu_j)^t\Sigma^{-1}(\mu_i-\mu_j)\right] \\ &= V\left[X^t\Sigma^{-1}(\mu_i-\mu_j)\right] \\ &= (\mu_i-\mu_j)^t\Sigma^{-1}\Sigma\Sigma^{-1}(\mu_i-\mu_j) \\ &= (\mu_i-\mu_j)^t\Sigma^{-1}(\mu_i-\mu_j) \\ &= \Delta^2,\end{aligned}$$

entonces,

$$\begin{aligned} P(j|i) &= P\left[Z < \frac{\ln(\pi_j/\pi_i) - \Delta^2/2}{\Delta} | X \in G_i\right] \\ &= \Phi\left(\frac{\ln(\pi_j/\pi_i) - \Delta^2/2}{\Delta}\right) \end{aligned}$$

donde $Z \sim N_1(0,1)$ con función de distribución Φ.

6.9. Código R utilizado

```
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
##XXXXXXXXXXXXXXX IMPUTACION XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
##en la variable edad se utilizó los códigos 1,2,3 y 4 en lugar de 0,1,2 y 3
datose=read.table("../entrenamiento_depurado.txt",header=T,sep="\t")
attach(datose)
library(mix)
datose_m=as.matrix(cbind(prov,tele,sexo,esci,tido,ries,nhij,edad,tres,mdes,ting,tgas))
##datose_m=as.matrix(cbind(prov,tele,sexo,esci,tido,ries,nhij,edad,log(1+tres),log(mdes),log(1+ting),tgas))
datose.pr=prelim.mix(datose_m,8)
datose.em=em.mix(datose.pr)
rngseed(1234567)
datose.im=imp.mix(datose.pr,datose.em,datose_m)
datose.im[,9]=exp(datose.im[,9])-1
datose.im[,10]=exp(datose.im[,10])-1
datose.im[,11]=exp(datose.im[,11])-1
##prov
summary(datose_m[,1])
summary(datose.im[,1])
##tele
summary(datose.im[,2])
summary(datose_m[,2])
##sexo
summary(datose.im[,3])
summary(datose_m[,3])
##esci
summary(datose.im[,4])
summary(datose_m[,4])
##tido
summary(datose.im[,5])
summary(datose_m[,5])
##ries
summary(datose.im[,6])
summary(datose_m[,6])
##nhij
summary(datose.im[,7])
summary(datose_m[,7])
##edad
summary(datose.im[,8])
summary(datose_m[,8])
##salio, luego d imputar un 1 (1434) y un 13 (2338) q le pongo 19
datose.im[1434,8]=19
datose.im[2338,8]=19
##tres
datose.im[datose.im[,9]<0,9]=0
datose.im[,9]=floor(datose.im[,9])
summary(datose.im[,9])
summary(datose_m[,9])
##mdes
datose.im[datose.im[,10]<0,10]=0
summary(datose.im[,10])
summary(datose_m[,10])
##abria q sacar los 0s desembolsados (2 datos) (casos 4 y 33)
##ting
datose.im[datose.im[,11]<0,11]=0
summary(datose.im[,11])
summary(datose_m[,11])
```

```
##abria q sacar los 0s d ingreso (12 casos)
##tgas
summary(datose.im[,12])
summary(datose_m[,12])
datosei=as.data.frame(datose.im)
names(datosei)=c("prov","tele","sexo","esci","tido","ries","nhij","edad","tres","mdes","ting","tgas")
## Histogramas de los datos imputados
detach(datose)
attach(datosei)
datosei=datosei[!(mdes==0),]
datosei=datosei[!(ting==0),]
datosei=datosei[1:4310,]
detach(datosei)
write.table(datosei,"../entrenamientoimpu.txt",sep="\t",row.names=F)
##recordar cambiar las etiquetas de la variable edad
##wilcox.test necesita que los dos conjuntos de datos sean independientes
##hace la prueba si las medias son iguales
##si p es pequeño, las poblaciones tienen medias diferentes;
##sino, no hay una razón para concluir que las dos medias difieren
rngseed(1835537)
s1=sample(1:4310,1000)
s2=sample(1:4310,1000)
wilcox.test(datosei$edad[s1],datose$edad[s2])
##p_value=0.1446 y W=516771.5
rngseed(7234597)
s1=sample(1:4310,1000)
s2=sample(1:4310,1000)
wilcox.test(datosei$tres[s1],datose$tres[s2])
##p_value=0.3678 y W=491251
rngseed(3184193)
s1=sample(1:4310,1000)
s2=sample(1:4310,1000)
wilcox.test(datosei$ting[s1],datose$ting[s2])
##p_value=0.3763 y W=244481
rngseed(1984123)
s1=sample(1:4310,1000)
s2=sample(1:4310,1000)
wilcox.test(datosei$tgas[s1],datose$tgas[s2])
##p_value=0.9455 y W=464163.5
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX COMPARACION CRUDOS IMPUTADOS XXXXXXXXXXXXXXXXXXXXXXXXXXXX
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
dc=read.table("../entrenamiento_depurado_con_etiquetas_adecuadas.txt",sep="\t",header=T)
di=read.table("../entrenamientoimpu.txt",sep="\t",header=T)
attach(dc)
par(mfrow=c(3,4))
hist(prov,main="",xlab="PROVINCIA DE ORÍGEN")
hist(tele,main="",xlab="TELEFONO")
hist(sexo,main="",xlab="GENERO")
hist(esci,main="",xlab="ESTADO CIVIL")
hist(tido,main="",xlab="TIPO DE DOMICILIO")
hist(ries,main="",xlab="TIPO DE RIESGO")
hist(edad,main="",xlab="EDAD")
hist(nhij,main="",xlab="NUMERO DE HIJOS")
hist(tres,main="",xlab="TIEMPO DE RESIDENCIA")
hist(mdes,main="",xlab="MONTO DESEMBOLZADO")
hist(ting,main="",xlab="TOTAL INGRESOS")
hist(tgas,main="",xlab="TOTAL GASTOS")
detach(dc)
attach(di)
par(mfrow=c(3,4))
hist(prov,main="",xlab="PROVINCIA DE ORÍGEN")
hist(tele,main="",xlab="TELEFONO")
hist(sexo,main="",xlab="GENERO")
hist(esci,main="",xlab="ESTADO CIVIL")
hist(tido,main="",xlab="TIPO DE DOMICILIO")
hist(ries,main="",xlab="TIPO DE RIESGO")
hist(edad,main="",xlab="EDAD")
hist(nhij,main="",xlab="NUMERO DE HIJOS")
hist(tres,main="",xlab="TIEMPO DE RESIDENCIA")
hist(mdes,main="",xlab="MONTO DESEMBOLZADO")
```

```
hist(ting,main="",xlab="TOTAL INGRESOS")
hist(tgas,main="",xlab="TOTAL GASTOS")
detach(di)
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX CARACTERIZACION XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
de=read.table("../entrenamientoimpu.txt",sep="\t",header=T)
## Hacer factor con fix() y arreglar dentro
attach(de)
par(mfrow=c(2,3))
boxplot(edad,main="EDAD")
boxplot(nhij,main="NUMERO DE HIJOS")
boxplot(tres,main="TIEMPO DE RESIDENCIA")
boxplot(mdes,main="MONTO DESEMBOLSADO")
boxplot(ting,main="INGRESOS TOTALES")
boxplot(tgas,main="GASTOS TOTALES")
par(mfrow=c(2,3))
hist(edad,main="EDAD")
hist(tres,main="TIEMPO DE RESIDENCIA")
hist(mdes,main="MONTO DESEMBOLSADO")
hist(ting,main="INGRESOS TOTALES")
hist(tgas,main="GASTOS TOTALES")
riesgo=factor(ries,labels=c("B","G"))
par(mfrow=c(2,3))
boxplot(edad~riesgo,main="EDAD")
boxplot(nhij~riesgo,main="NUMERO DE HIJOS")
boxplot(tres~riesgo,main="TIEMPO DE RESIDENCIA")
boxplot(mdes~riesgo,main="MONTO DESEMBOLSADO")
boxplot(ting~riesgo,main="INGRESOS TOTALES")
boxplot(tgas~riesgo,main="GASTOS TOTALES")
##table hace prueba de independencia usando xi-cuadrado
summary(table(ries,prov))
##p-value = 0.001107
summary(table(ries,tele))
##p-value = 0.5849 ->rechaza independencia
summary(table(ries,sexo))
##p-value = 0.9396 -> acepta independiencia
summary(table(ries,esci))
##p-value = 0.05048 -> casi acepta independiencia
summary(table(ries,tido))
##p-value = 0.0001889 ->rechaza independencia
summary(table(ries,nhij))
##p-value = 0.6895 -> acepta independiencia
dettach(de)

datosi$conj=factor(datosi$conj,labels=c("entre","prue"))
datosi$riesgo=factor(datosi$riesgo,labels=c("br","gr"))
datosi$tel=factor(datosi$tel,labels=c("no","si"))
datosi$sex=factor(datosi$sex,labels=c("f","m"))
datosi$e_ci=factor(datosi$e_ci,labels=c("cas","div","sep","sol","uli","viu"))
datosi$d_cre=factor(datosi$d_cre,labels=c("captra","matpri","mave","otros"))
datosi$t_dom=factor(datosi$t_dom,labels=c("arren","dfam","pro"))

##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
## Código análisis discriminate lineal solo variables continuasXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

de=read.table("../entrenamientoimpu.txt",sep="\t",header=T)
de5=read.table("../entrenamientoimpu5050.txt",sep="\t",header=T)
dp=read.table("../prueba1.txt",sep="\t",header=T)
library(mvnormtest)
attach(de)
par(mfrow=c(3,2))
hist(edad)
hist(tres)
hist(mdes)
hist(ting)
hist(tgas)
par(mfrow=c(3,2))
hist(edad)
hist(log(1+tres))
```

```
hist(log(mdes))
hist(ting)
hist(tgas)
x=cbind(edad,tres,mdes,ting,tgas)
##x=cbind(ting,tgas)
detach(de)
attach(dp)
y=cbind(edad,tres,mdes,ting,tgas)
##y=cbind(ting,tgas)
detach(dp)
x=as.data.frame(scale(x))
y=as.data.frame(scale(y))
mshapiro.test(t(x))
xb=x[1:529,]
xg=x[530:4310,]
ug=mean(xg)
ub=mean(xb)
sg=cov(xg)
sb=cov(xb)
s=(3780/4308)*sg+(528/4308)*sb
si=solve(s)
ldfm<-function(x,u,s,p){
d=as.matrix(x)%*%s%*%u-0.5*t(u)%*%s%*%u+(log(p))
}
d=matrix(nrow=1559,ncol=2)
c=0
for(j in 1:1559){
d[j,1]=ldfm(y[j,],ug,si,0.877)
d[j,2]=ldfm(y[j,],ub,si,0.123)
c[j]=(sort(d[j,],decreasing=T,index.return=T)$ix[1])
}
##matrix de mezcla
confusion=function(true, predict){
jt=table(true,predict)
jn=dimnames(jt)[[2]]
jt1=jt[jn,]
structure(jt,error=(1-sum(diag(jt1))/length(true)))
}
confusion(dp$ries,c)
    predict
true    1
   1  178
   2 1381
##88.5% DE ERROR
##no puede reconocer a la categoria 2 (Good)

##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
##XXXXXXXXXXXXXXXXX 50:50 XXXXXXXXXXX
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
attach(de5)
##x=cbind(edad,tres,mdes,ting,tgas)
x=cbind(tres,ting,tgas)
detach(de5)
attach(dp)
##y=cbind(edad,tres,mdes,ting,tgas)
y=cbind(tres,ting,tgas)
detach(dp)
x=as.data.frame(scale(x))
y=as.data.frame(scale(y))
mshapiro.test(t(x))
xb=x[1:529,]
xg=x[530:1058,]
ug=mean(xg)
ub=mean(xb)
sg=cov(xg)
sb=cov(xb)
s=(528/1056)*sg+(528/1056)*sb
si=solve(s)
ldfm<-function(x,u,s,p){
d=as.matrix(x)%*%s%*%u-0.5*t(u)%*%s%*%u+(log(p))
}
```

```
d=matrix(nrow=1559,ncol=2)
c=0
for(j in 1:1559){
d[j,1]=ldfm(y[j,],ug,si,0.5)
d[j,2]=ldfm(y[j,],ub,si,0.5)
c[j]=(sort(d[j,],decreasing=T,index.return=T)$ix[1])
}
##matrix de mezcla
confusion=function(true, predict){
jt=table(true,predict)
jn=dimnames(jt)[[2]]
jt1=jt[jn,]
structure(jt,error=(1-sum(diag(jt1))/length(true)))
}
confusion(dp$ries,c)

    predict
true   1   2
   1  57 121
   2 630 751
##48.2% DE ERROR

attach(de5)
plot(ting[1:529],tgas[1:529],col="red")
points(ting[530:1058],tgas[530:1058],col="blue")

plot(ting[1:178],tgas[1:178],col="red")
points(ting[179:1559],tgas[179:1559],col="blue")

##balancear la muestra ayudó a la discriminación
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
## Código análisis discriminate lineal variables continuas y categóricasXXXXXXXXXXXXXXXXXXXX
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
de=read.table("../entrenamientoimpu.txt",sep="\t",header=T)
de5=read.table("../entrenamientoimpu5050.txt",sep="\t",header=T)
dp=read.table("../prueba1.txt",sep="\t",header=T)
attach(de)
tele1=0
for(k in 1:4310){
if (tele[k]==1) tele1[k]=1 else tele1[k]=0}
sexo1=0
for(k in 1:4310){
if (sexo[k]==1) sexo1[k]=1 else sexo1[k]=0}
esci2=0
esci3=0
for(k in 1:4310){
if (esci[k]==2) esci2[k]=1 else esci2[k]=0
if (esci[k]==3) esci3[k]=1 else esci3[k]=0}
prov2=0
prov3=0
prov4=0
prov5=0
prov6=0
for(k in 1:4310){
if (prov[k]==2) prov2[k]=1 else prov2[k]=0
if (prov[k]==3) prov3[k]=1 else prov3[k]=0
if (prov[k]==4) prov4[k]=1 else prov4[k]=0
if (prov[k]==5) prov5[k]=1 else prov5[k]=0
if (prov[k]==6) prov6[k]=1 else prov6[k]=0}
tido2=0
tido3=0
for(k in 1:4310){
if (tido[k]==2) tido2[k]=1 else tido2[k]=0
if (tido[k]==3) tido3[k]=1 else tido3[k]=0}
nhij1=0
nhij2=0
nhij3=0
for(k in 1:4310){
if (nhij[k]==1) nhij1[k]=1 else nhij1[k]=0
if (nhij[k]==2) nhij2[k]=1 else nhij2[k]=0
if (nhij[k]==3) nhij3[k]=1 else nhij3[k]=0}
```

```
detach(de)
attach(dp)
tele1y=0
for(k in 1:1559){
if (tele[k]==1) tele1y[k]=1 else tele1y[k]=0}
sexo1y=0
for(k in 1:1559){
if (sexo[k]==1) sexo1y[k]=1 else sexo1y[k]=0}
esci2y=0
esci3y=0
for(k in 1:1559){
if (esci[k]==2) esci2y[k]=1 else esci2y[k]=0
if (esci[k]==3) esci3y[k]=1 else esci3y[k]=0}
prov2y=0
prov3y=0
prov4y=0
prov5y=0
prov6y=0
for(k in 1:1559){
if (prov[k]==2) prov2y[k]=1 else prov2y[k]=0
if (prov[k]==3) prov3y[k]=1 else prov3y[k]=0
if (prov[k]==4) prov4y[k]=1 else prov4y[k]=0
if (prov[k]==5) prov5y[k]=1 else prov5y[k]=0
if (prov[k]==6) prov6y[k]=1 else prov6y[k]=0}
tido2y=0
tido3y=0
for(k in 1:1559){
if (tido[k]==2) tido2y[k]=1 else tido2y[k]=0
if (tido[k]==3) tido3y[k]=1 else tido3y[k]=0}
nhij1y=0
nhij2y=0
nhij3y=0
for(k in 1:1559){
if (nhij[k]==1) nhij1y[k]=1 else nhij1y[k]=0
if (nhij[k]==2) nhij2y[k]=1 else nhij2y[k]=0
if (nhij[k]==3) nhij3y[k]=1 else nhij3y[k]=0}
detach(dp)

attach(de)
xc=cbind(edad,tres,mdes,ting,tgas)
xc=scale(xc)
x=as.data.frame(cbind(xc,esci2,esci3,prov2,prov3,prov4,prov5,prov6,tele1,sexo1,tido2,tido3,nhij1,nhij2,nhij3))
detach(de)
attach(dp)
yc=cbind(edad,tres,mdes,ting,tgas)
yc=scale(yc)
y=as.data.frame(cbind(yc,esci2y,esci3y,prov2y,prov3y,prov4y,prov5y,prov6y,tele1y,sexo1y,tido2y,tido3y,nhij1y,nhij2y,nhij3y))
detach(dp)
xb=x[1:529,]
xg=x[530:4310,]
ug=mean(xg)
ub=mean(xb)
sg=cov(xg)
sb=cov(xb)
s=(3780/4308)*sg+(528/4308)*sb
si=solve(s)
d=matrix(nrow=1559,ncol=2)
c=0
ldfm<-function(x,u,s,p){
d=as.matrix(x)%*%s%*%u-0.5*t(u)%*%s%*%u+(log(p))
}
for(j in 1:1559){
d[j,1]=ldfm(y[j,],ug,si,0.877)
d[j,2]=ldfm(y[j,],ub,si,0.123)
c[j]=(sort(d[j,],decreasing=T,index.return=T)$ix[1])
}
##matrix de mezcla
confusion=function(true, predict){
jt=table(true,predict)
jn=dimnames(jt)[[2]]
jt1=jt[jn,]
```

```
structure(jt,error=(1-sum(diag(jt1))/length(true)))
}
confusion(dp$ries,c)
    predict
true    1
   1  178
   2 1381
##88.5% de error

##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
##XXXXXXXXXXXXXXXXX 50:50 XXXXXXXXXXXX
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
attach(de5)
tele1=0
for(k in 1:1058){
if (tele[k]==1) tele1[k]=1 else tele1[k]=0}
sexo1=0
for(k in 1:1058){
if (sexo[k]==1) sexo1[k]=1 else sexo1[k]=0}
esci2=0
esci3=0
for(k in 1:1058){
if (esci[k]==2) esci2[k]=1 else esci2[k]=0
if (esci[k]==3) esci3[k]=1 else esci3[k]=0}
prov2=0
prov3=0
prov4=0
prov5=0
prov6=0
for(k in 1:1058){
if (prov[k]==2) prov2[k]=1 else prov2[k]=0
if (prov[k]==3) prov3[k]=1 else prov3[k]=0
if (prov[k]==4) prov4[k]=1 else prov4[k]=0
if (prov[k]==5) prov5[k]=1 else prov5[k]=0
if (prov[k]==6) prov6[k]=1 else prov6[k]=0}
tido2=0
tido3=0
for(k in 1:1058){
if (tido[k]==2) tido2[k]=1 else tido2[k]=0
if (tido[k]==3) tido3[k]=1 else tido3[k]=0}
nhij1=0
nhij2=0
nhij3=0
for(k in 1:1058){
if (nhij[k]==1) nhij1[k]=1 else nhij1[k]=0
if (nhij[k]==2) nhij2[k]=1 else nhij2[k]=0
if (nhij[k]==3) nhij3[k]=1 else nhij3[k]=0}
detach(de5)
attach(de5)
##xc=cbind(edad,tres,mdes,ting,tgas)
xc=cbind(tgas)
xc=scale(xc)
##x=as.data.frame(cbind(xc,esci2,esci3,prov2,prov3,prov4,prov5,prov6,tele1,sexo1,tido2,tido3,nhij1,nhij2,nhij3))
x=as.data.frame(cbind(xc,prov3,tido3))
detach(de5)
attach(dp)
##yc=cbind(edad,tres,mdes,ting,tgas)
yc=cbind(tgas)
yc=scale(yc)
##y=as.data.frame(cbind(yc,esci2y,esci3y,prov2y,prov3y,prov4y,prov5y,prov6y,tele1y,sexo1y,tido2y,tido3y,nhij1y,nhij2y,nhij3y))
y=as.data.frame(cbind(yc,prov3y,tido3y))
detach(dp)
xb=x[1:529,]
xg=x[530:1058,]
ug=mean(xg)
ub=mean(xb)
sg=cov(xg)
sb=cov(xb)
s=(528/1056)*sg+(528/1056)*sb
si=solve(s)
d=matrix(nrow=1559,ncol=2)
```

```
c=0
ldfm<-function(x,u,s,p){
d=as.matrix(x)%*%s%*%u-0.5*t(u)%*%s%*%u+(log(p))
}
for(j in 1:1559){
d[j,1]=ldfm(y[j,],ug,si,0.5)
d[j,2]=ldfm(y[j,],ub,si,0.5)
c[j]=(sort(d[j,],decreasing=T,index.return=T)$ix[1])
}
##matrix de mezcla
confusion=function(true, predict){
jt=table(true,predict)
jn=dimnames(jt)[[2]]
jt1=jt[jn,]
structure(jt,error=(1-sum(diag(jt1))/length(true)))
}
confusion(dp$ries,c)
    predict
true   1   2
   1  66 112
   2 745 636
##55.0% de error
##no mejoró el error respecto al de solo las continuas
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
## Código análisis discriminate cuadrático solo variables continuasXXXXXXXXXXXXXXXXXXXXXXXXX
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

de=read.table("../entrenamientoimpu.txt",sep="\t",header=T)
de5=read.table("../entrenamientoimpu5050.txt",sep="\t",header=T)
dp=read.table("../prueba1.txt",sep="\t",header=T)
attach(de)
x=as.data.frame(cbind(edad,tres,mdes,ting,tgas))
detach(de)
attach(dp)
y=as.data.frame(cbind(edad,tres,mdes,ting,tgas))
detach(dp)

x=as.data.frame(scale(x))
y=as.data.frame(scale(y))
xb=x[1:529,]
xg=x[530:4310,]
ug=mean(xg)
ub=mean(xb)
sg=solve(cov(xg))
sb=solve(cov(xb))
d=matrix(nrow=1559,ncol=2)
c=0
ldfm<-function(x,u,s,p){
d=(as.matrix(x)-u)%*%s%*%t(as.matrix(x)-u)-0.5*log(det(s))+(log(p))
}
for(j in 1:1559){
d[j,1]=ldfm(y[j,],ug,sg,0.877)
d[j,2]=ldfm(y[j,],ub,sb,0.123)
c[j]=(sort(d[j,],decreasing=T,index.return=T)$ix[1])
}
confusion=function(true, predict){
jt=table(true,predict)
jn=dimnames(jt)[[2]]
jt1=jt[jn,]
structure(jt,error=(1-sum(diag(jt1))/length(true)))
}
confusion(dp$ries,c)
    predict
true    1    2
   1  168   10
   2 1292   89
##83.5%
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
##XXXXXXXXXXXXXXXXX 50:50 XXXXXXXXXXXX
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
attach(de5)
```

```
##x=as.data.frame(cbind(edad,tres,mdes,ting,tgas))
x=as.data.frame(cbind(edad,tres,mdes))
detach(de5)
attach(dp)
##y=as.data.frame(cbind(edad,tres,mdes,ting,tgas))
y=as.data.frame(cbind(edad,tres,mdes))
detach(dp)
x=as.data.frame(scale(x))
y=as.data.frame(scale(y))
##library(mvnormtest)
##mshapiro.test(t(x))
##
xb=x[1:529,]
xg=x[530:1058,]
ug=mean(xg)
ub=mean(xb)
sg=solve(cov(xg))
sb=solve(cov(xb))
d=matrix(nrow=1559,ncol=2)
c=0
ldfm<-function(x,u,s,p){
d=(as.matrix(x)-u)%*%s%*%t(as.matrix(x)-u)-0.5*log(det(s))+(log(p))
}
for(j in 1:1559){
d[j,1]=ldfm(y[j,],ug,sg,0.5)
d[j,2]=ldfm(y[j,],ub,sb,0.5)
c[j]=(sort(d[j,],decreasing=T,index.return=T)$ix[1])
}
confusion=function(true, predict){
jt=table(true,predict)
jn=dimnames(jt)[[2]]
jt1=jt[jn,]
structure(jt,error=(1-sum(diag(jt1))/length(true)))
}
confusion(dp$ries,c)
    predict
true   1   2
   1 121  57
   2 784 597
##53.9% de error
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
## Código análisis discriminate cuadrático variables continuas y categóricasXXXXXXXXXXXXXXXX
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

de=read.table("../entrenamientoimpu.txt",sep="\t",header=T)
de5=read.table("../entrenamientoimpu5050.txt",sep="\t",header=T)
dp=read.table("../prueba1.txt",sep="\t",header=T)
attach(de)
tele1=0
for(k in 1:4310){
if (tele[k]==1) tele1[k]=1 else tele1[k]=0}
sexo1=0
for(k in 1:4310){
if (sexo[k]==1) sexo1[k]=1 else sexo1[k]=0}
esci2=0
esci3=0
for(k in 1:4310){
if (esci[k]==2) esci2[k]=1 else esci2[k]=0
if (esci[k]==3) esci3[k]=1 else esci3[k]=0}
prov2=0
prov3=0
prov4=0
prov5=0
prov6=0
for(k in 1:4310){
if (prov[k]==2) prov2[k]=1 else prov2[k]=0
if (prov[k]==3) prov3[k]=1 else prov3[k]=0
if (prov[k]==4) prov4[k]=1 else prov4[k]=0
if (prov[k]==5) prov5[k]=1 else prov5[k]=0
if (prov[k]==6) prov6[k]=1 else prov6[k]=0}
tido2=0
```

```
tido3=0
for(k in 1:4310){
if (tido[k]==2) tido2[k]=1 else tido2[k]=0
if (tido[k]==3) tido3[k]=1 else tido3[k]=0}
nhij1=0
nhij2=0
nhij3=0
for(k in 1:4310){
if (nhij[k]==1) nhij1[k]=1 else nhij1[k]=0
if (nhij[k]==2) nhij2[k]=1 else nhij2[k]=0
if (nhij[k]==3) nhij3[k]=1 else nhij3[k]=0}
detach(de)
attach(dp)
tele1y=0
for(k in 1:1559){
if (tele[k]==1) tele1y[k]=1 else tele1y[k]=0}
sexo1y=0
for(k in 1:1559){
if (sexo[k]==1) sexo1y[k]=1 else sexo1y[k]=0}
esci2y=0
esci3y=0
for(k in 1:1559){
if (esci[k]==2) esci2y[k]=1 else esci2y[k]=0
if (esci[k]==3) esci3y[k]=1 else esci3y[k]=0}
prov2y=0
prov3y=0
prov4y=0
prov5y=0
prov6y=0
for(k in 1:1559){
if (prov[k]==2) prov2y[k]=1 else prov2y[k]=0
if (prov[k]==3) prov3y[k]=1 else prov3y[k]=0
if (prov[k]==4) prov4y[k]=1 else prov4y[k]=0
if (prov[k]==5) prov5y[k]=1 else prov5y[k]=0
if (prov[k]==6) prov6y[k]=1 else prov6y[k]=0}
tido2y=0
tido3y=0
for(k in 1:1559){
if (tido[k]==2) tido2y[k]=1 else tido2y[k]=0
if (tido[k]==3) tido3y[k]=1 else tido3y[k]=0}
nhij1y=0
nhij2y=0
nhij3y=0
for(k in 1:1559){
if (nhij[k]==1) nhij1y[k]=1 else nhij1y[k]=0
if (nhij[k]==2) nhij2y[k]=1 else nhij2y[k]=0
if (nhij[k]==3) nhij3y[k]=1 else nhij3y[k]=0}
detach(dp)
attach(de)
xc=cbind(edad,tres,mdes,ting,tgas)
xc=scale(xc)
x=as.data.frame(cbind(xc,esci2,esci3,prov2,prov3,prov4,prov5,prov6,tele1,sexo1,tido2,tido3,nhij1,nhij2,nhij3))
detach(de)
attach(dp)
yc=cbind(edad,tres,mdes,ting,tgas)
yc=scale(yc)
y=as.data.frame(cbind(yc,esci2y,esci3y,prov2y,prov3y,prov4y,prov5y,prov6y,tele1y,sexo1y,tido2y,tido3y,nhij1y,nhij2y,nhij3y))
detach(dp)
xb=x[1:529,]
xg=x[530:4310,]
ug=mean(xg)
ub=mean(xb)
sg=solve(cov(xg))
sb=solve(cov(xb))
d=matrix(nrow=1559,ncol=2)
c=0
ldfm<-function(x,u,s,p){
d=(as.matrix(x)-u)%*%s%*%t(as.matrix(x)-u)-0.5*log(det(s))+(log(p))
}
for(j in 1:1559){
d[j,1]=ldfm(y[j,],ug,sg,0.877)
```

```
d[j,2]=ldfm(y[j,],ub,sb,0.123)
c[j]=(sort(d[j,],decreasing=T,index.return=T)$ix[1])
}
confusion=function(true, predict){
jt=table(true,predict)
jn=dimnames(jt)[[2]]
jt1=jt[jn,]
structure(jt,error=(1-sum(diag(jt1))/length(true)))
}
confusion(dp$ries,c)
    predict
true   1   2
   1 147  31
   2 956 425
##63.3% con todas.

##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
##XXXXXXXXXXXXXXXXX 50:50 XXXXXXXXXXXX
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
attach(de5)
tele1=0
for(k in 1:1058){
if (tele[k]==1) tele1[k]=1 else tele1[k]=0}
sexo1=0
for(k in 1:1058){
if (sexo[k]==1) sexo1[k]=1 else sexo1[k]=0}
esci2=0
esci3=0
for(k in 1:1058){
if (esci[k]==2) esci2[k]=1 else esci2[k]=0
if (esci[k]==3) esci3[k]=1 else esci3[k]=0}
prov2=0
prov3=0
prov4=0
prov5=0
prov6=0
for(k in 1:1058){
if (prov[k]==2) prov2[k]=1 else prov2[k]=0
if (prov[k]==3) prov3[k]=1 else prov3[k]=0
if (prov[k]==4) prov4[k]=1 else prov4[k]=0
if (prov[k]==5) prov5[k]=1 else prov5[k]=0
if (prov[k]==6) prov6[k]=1 else prov6[k]=0}
tido2=0
tido3=0
for(k in 1:1058){
if (tido[k]==2) tido2[k]=1 else tido2[k]=0
if (tido[k]==3) tido3[k]=1 else tido3[k]=0}
nhij1=0
nhij2=0
nhij3=0
for(k in 1:1058){
if (nhij[k]==1) nhij1[k]=1 else nhij1[k]=0
if (nhij[k]==2) nhij2[k]=1 else nhij2[k]=0
if (nhij[k]==3) nhij3[k]=1 else nhij3[k]=0}
detach(de5)
attach(de5)
##xc=cbind(edad,tres,mdes,ting,tgas)
xc=cbind(ting,tgas)
xc=scale(xc)
##x=as.data.frame(cbind(xc,esci2,esci3,prov2,prov3,prov4,prov5,prov6,tele1,sexo1,tido2,tido3,nhij1,nhij2,nhij3))
x=as.data.frame(cbind(xc,esci2,esci3,prov2,prov3,prov4,prov5,prov6,tido2,tido3))
detach(de5)
attach(dp)
##yc=cbind(edad,tres,mdes,ting,tgas)
yc=cbind(ting,tgas)
yc=scale(yc)
##y=as.data.frame(cbind(yc,esci2y,esci3y,prov2y,prov3y,prov4y,prov5y,prov6y,tele1y,sexo1y,tido2y,tido3y,nhij1y,nhij2y,nhij3y))
y=as.data.frame(cbind(yc,esci2y,esci3y,prov2y,prov3y,prov4y,prov5y,prov6y,tido2y,tido3y))
detach(dp)
xb=x[1:529,]
xg=x[530:1058,]
```

```
ug=mean(xg)
ub=mean(xb)
sg=solve(cov(xg))
sb=solve(cov(xb))
d=matrix(nrow=1559,ncol=2)
c=0
ldfm<-function(x,u,s,p){
d=(as.matrix(x)-u)%*%s%*%t(as.matrix(x)-u)-0.5*log(det(s))+(log(p))
}
for(j in 1:1559){
d[j,1]=ldfm(y[j,],ug,sg,0.5)
d[j,2]=ldfm(y[j,],ub,sb,0.5)
c[j]=(sort(d[j,],decreasing=T,index.return=T)$ix[1])
}
confusion=function(true, predict){
jt=table(true,predict)
jn=dimnames(jt)[[2]]
jt1=jt[jn,]
structure(jt,error=(1-sum(diag(jt1))/length(true)))
}
confusion(dp$ries,c)
    predict
true   1   2
   1 111  67
   2 601 780
##42.8% de error

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXX FUNCIONES DISCRIMINANTES CANÓNICAS CON VARIABLES CONTINUAS XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

library(MASS)
de=read.table("../entrenamientoimpu.txt",sep="\t",header=T)
de5=read.table("../entrenamientoimpu5050.txt",sep="\t",header=T)
dp=read.table("../prueba1.txt",sep="\t",header=T)
attach(de)
x=cbind(edad,tres,mdes,ting,tgas)
x=scale(x)
detach(de)
attach(dp)
y=cbind(edad,tres,mdes,ting,tgas)
y=scale(y)
detach(dp)


f1=lda(x,de$ries)
##f1$svd^2/sum(f1$svd^2)
confusion=function(true, predict){
jt=table(true,predict)
jn=dimnames(jt)[[2]]
jt1=jt[jn,]
structure(jt,error=(1-sum(diag(jt1))/length(true)))
}
confusion(dp$ries,predict(f1,y)$class)
confusion(dp$ries,predict(f1,y,dimen=2)$class)

##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
##XXXXXXXXXXXXXXXXXX 50:50 XXXXXXXXXXXX
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
attach(de5)
##x=cbind(edad,tres,mdes,ting,tgas)
x=cbind(edad,tres)
x=scale(x)
detach(de5)
attach(dp)
##y=cbind(edad,tres,mdes,ting,tgas)
y=cbind(edad,tres)
y=scale(y)
detach(dp)
```

```
f1=lda(x,de5$ries)
##f1$svd^2/sum(f1$svd^2)
confusion=function(true, predict){
jt=table(true,predict)
jn=dimnames(jt)[[2]]
jt1=jt[jn,]
structure(jt,error=(1-sum(diag(jt1))/length(true)))
}
confusion(dp$ries,predict(f1,y)$class)
confusion(dp$ries,predict(f1,y,dimen=2)$class)

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXX FUNCIONES DISCRIMINANTES CANÓNICAS CON VARIABLES CONTINUAS Y CATEGÓRICAS XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
de=read.table("../entrenamientoimpu.txt",sep="\t",header=T)
attach(de)
tele1=0
for(k in 1:4310){
if (tele[k]==1) tele1[k]=1 else tele1[k]=0}
sexo1=0
for(k in 1:4310){
if (sexo[k]==1) sexo1[k]=1 else sexo1[k]=0}
esci2=0
esci3=0
for(k in 1:4310){
if (esci[k]==2) esci2[k]=1 else esci2[k]=0
if (esci[k]==3) esci3[k]=1 else esci3[k]=0}
prov2=0
prov3=0
prov4=0
prov5=0
prov6=0
for(k in 1:4310){
if (prov[k]==2) prov2[k]=1 else prov2[k]=0
if (prov[k]==3) prov3[k]=1 else prov3[k]=0
if (prov[k]==4) prov4[k]=1 else prov4[k]=0
if (prov[k]==5) prov5[k]=1 else prov5[k]=0
if (prov[k]==6) prov6[k]=1 else prov6[k]=0}
tido2=0
tido3=0
for(k in 1:4310){
if (tido[k]==2) tido2[k]=1 else tido2[k]=0
if (tido[k]==3) tido3[k]=1 else tido3[k]=0}
nhij1=0
nhij2=0
nhij3=0
for(k in 1:4310){
if (nhij[k]==1) nhij1[k]=1 else nhij1[k]=0
if (nhij[k]==2) nhij2[k]=1 else nhij2[k]=0
if (nhij[k]==3) nhij3[k]=1 else nhij3[k]=0}
detach(de)
dp=read.table("../prueba1.txt",sep="\t",header=T)
attach(dp)
tele1y=0
for(k in 1:1559){
if (tele[k]==1) tele1y[k]=1 else tele1y[k]=0}
sexo1y=0
for(k in 1:1559){
if (sexo[k]==1) sexo1y[k]=1 else sexo1y[k]=0}
esci2y=0
esci3y=0
for(k in 1:1559){
if (esci[k]==2) esci2y[k]=1 else esci2y[k]=0
if (esci[k]==3) esci3y[k]=1 else esci3y[k]=0}
prov2y=0
prov3y=0
prov4y=0
prov5y=0
prov6y=0
for(k in 1:1559){
if (prov[k]==2) prov2y[k]=1 else prov2y[k]=0
```

```
if (prov[k]==3) prov3y[k]=1 else prov3y[k]=0
if (prov[k]==4) prov4y[k]=1 else prov4y[k]=0
if (prov[k]==5) prov5y[k]=1 else prov5y[k]=0
if (prov[k]==6) prov6y[k]=1 else prov6y[k]=0}
tido2y=0
tido3y=0
for(k in 1:1559){
if (tido[k]==2) tido2y[k]=1 else tido2y[k]=0
if (tido[k]==3) tido3y[k]=1 else tido3y[k]=0}
nhij1y=0
nhij2y=0
nhij3y=0
for(k in 1:1559){
if (nhij[k]==1) nhij1y[k]=1 else nhij1y[k]=0
if (nhij[k]==2) nhij2y[k]=1 else nhij2y[k]=0
if (nhij[k]==3) nhij3y[k]=1 else nhij3y[k]=0}
detach(dp)


attach(de)
xc=cbind(edad,tres,mdes,ting,tgas)
xc=scale(xc)
x=as.data.frame(cbind(xc,esci2,esci3,prov2,prov3,prov4,prov5,prov6,tele1,sexo1,tido2,tido3,nhij1,nhij2,nhij3))
detach(de)
attach(dp)
yc=cbind(edad,tres,mdes,ting,tgas)
yc=scale(yc)
y=as.data.frame(cbind(yc,esci2y,esci3y,prov2y,prov3y,prov4y,prov5y,prov6y,tele1y,sexo1y,tido2y,tido3y,nhij1y,nhij2y,nhij3y))
detach(dp)



f1=lda(x,de$ries)
##f1$svd^2/sum(f1$svd^2)
confusion=function(true, predict){
jt=table(true,predict)
jn=dimnames(jt)[[2]]
jt1=jt[jn,]
structure(jt,error=(1-sum(diag(jt1))/length(true)))
}
confusion(dp$ries,predict(f1,y)$class)
confusion(dp$ries,predict(f1,y,dimen=2)$class)

##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
##XXXXXXXXXXXXXXXXX 50:50 XXXXXXXXXXXX
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
de5=read.table("../entrenamientoimpu5050.txt",sep="\t",header=T)
attach(de5)
tele1=0
for(k in 1:1058){
if (tele[k]==1) tele1[k]=1 else tele1[k]=0}
sexo1=0
for(k in 1:1058){
if (sexo[k]==1) sexo1[k]=1 else sexo1[k]=0}
esci2=0
esci3=0
for(k in 1:1058){
if (esci[k]==2) esci2[k]=1 else esci2[k]=0
if (esci[k]==3) esci3[k]=1 else esci3[k]=0}
prov2=0
prov3=0
prov4=0
prov5=0
prov6=0
for(k in 1:1058){
if (prov[k]==2) prov2[k]=1 else prov2[k]=0
if (prov[k]==3) prov3[k]=1 else prov3[k]=0
if (prov[k]==4) prov4[k]=1 else prov4[k]=0
if (prov[k]==5) prov5[k]=1 else prov5[k]=0
if (prov[k]==6) prov6[k]=1 else prov6[k]=0}
tido2=0
```

```
tido3=0
for(k in 1:1058){
if (tido[k]==2) tido2[k]=1 else tido2[k]=0
if (tido[k]==3) tido3[k]=1 else tido3[k]=0}
nhij1=0
nhij2=0
nhij3=0
for(k in 1:1058){
if (nhij[k]==1) nhij1[k]=1 else nhij1[k]=0
if (nhij[k]==2) nhij2[k]=1 else nhij2[k]=0
if (nhij[k]==3) nhij3[k]=1 else nhij3[k]=0}
detach(de5)

attach(de5)
##xc=cbind(edad,tres,mdes,ting,tgas)
xc=cbind(tgas)
xc=scale(xc)
##x=as.data.frame(cbind(xc,esci2,esci3,prov2,prov3,prov4,prov5,prov6,tele1,sexo1,tido2,tido3,nhij1,nhij2,nhij3))
x=as.data.frame(cbind(xc,prov3,tido3))
detach(de5)
attach(dp)
##yc=cbind(edad,tres,mdes,ting,tgas)
yc=cbind(tgas)
yc=scale(yc)
##y=as.data.frame(cbind(yc,esci2y,esci3y,prov2y,prov3y,prov4y,prov5y,prov6y,tele1y,sexo1y,tido2y,tido3y,nhij1y,nhij2y,nhij3y))
y=as.data.frame(cbind(yc,prov3y,tido3y))
detach(dp)

f1=lda(x,de5$ries)
##f1$svd^2/sum(f1$svd^2)
confusion=function(true, predict){
jt=table(true,predict)
jn=dimnames(jt)[[2]]
jt1=jt[jn,]
structure(jt,error=(1-sum(diag(jt1))/length(true)))
}
confusion(dp$ries,predict(f1,y)$class)
confusion(dp$ries,predict(f1,y,dimen=2)$class)


##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
##XXXXXXXX REGRESIÓN LOGÍSTICA VARIABLES CONTINUAS XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
library(VGAM)
de=read.table("../entrenamientoimpu.txt",sep="\t",header=T)
de5=read.table("../entrenamientoimpu5050.txt",sep="\t",header=T)
dp=read.table("../prueba1.txt",sep="\t",header=T)
attach(de)
x=cbind(edad,tres,mdes,ting,tgas)
x=as.data.frame(scale(x))
x=cbind(ries,x)
detach(de)
attach(dp)
y=cbind(edad,tres,mdes,ting,tgas)
y=as.data.frame(scale(y))
detach(dp)
modelo = vglm(ries ~ ., multinomial, x)
yt=predict(modelo,y)
e=exp(yt)
q1=e/(1+e)
q2=1/(1+e)
q=cbind(q1,q2)
c=0
for(j in 1:1559){
if (q[j,1]>q[j,2]) c[j]=1 else c[j]=2
}
confusion=function(true, predict){
jt=table(true,predict)
jn=dimnames(jt)[[2]]
jt1=jt[jn,]
structure(jt,error=(1-sum(diag(jt1))/length(true)))
```

```
}
confusion(dp$ries,c)

##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
##XXXXXXXXXXXXXXXXX 50:50 XXXXXXXXXXXX
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

attach(de5)
##x=cbind(edad,tres,mdes,ting,tgas)
x=cbind(edad,tres)
x=as.data.frame(scale(x))
x=cbind(ries,x)
detach(de5)
attach(dp)
##y=cbind(edad,tres,mdes,ting,tgas)
y=cbind(edad,tres)
y=as.data.frame(scale(y))
detach(dp)
modelo = vglm(ries ~ ., multinomial, x)
yt=predict(modelo,y)
e=exp(yt)
q1=e/(1+e)
q2=1/(1+e)
q=cbind(q1,q2)
c=0
for(j in 1:1559){
if (q[j,1]>q[j,2]) c[j]=1 else c[j]=2
}
confusion=function(true, predict){
jt=table(true,predict)
jn=dimnames(jt)[[2]]
jt1=jt[jn,]
structure(jt,error=(1-sum(diag(jt1))/length(true)))
}
confusion(dp$ries,c)

##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
##XXXXXXXX REGRESION LOGÍSTICA VARIABLES CONTINUAS Y CATEGORICAS XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
library(VGAM)
de=read.table("../entrenamientoimpu.txt",sep="\t",header=T)
attach(de)
tele1=0
for(k in 1:4310){
if (tele[k]==1) tele1[k]=1 else tele1[k]=0}
sexo1=0
for(k in 1:4310){
if (sexo[k]==1) sexo1[k]=1 else sexo1[k]=0}
esci2=0
esci3=0
for(k in 1:4310){
if (esci[k]==2) esci2[k]=1 else esci2[k]=0
if (esci[k]==3) esci3[k]=1 else esci3[k]=0}
prov2=0
prov3=0
prov4=0
prov5=0
prov6=0
for(k in 1:4310){
if (prov[k]==2) prov2[k]=1 else prov2[k]=0
if (prov[k]==3) prov3[k]=1 else prov3[k]=0
if (prov[k]==4) prov4[k]=1 else prov4[k]=0
if (prov[k]==5) prov5[k]=1 else prov5[k]=0
if (prov[k]==6) prov6[k]=1 else prov6[k]=0}
tido2=0
tido3=0
for(k in 1:4310){
if (tido[k]==2) tido2[k]=1 else tido2[k]=0
if (tido[k]==3) tido3[k]=1 else tido3[k]=0}
nhij1=0
nhij2=0
```

```
nhij3=0
for(k in 1:4310){
if (nhij[k]==1) nhij1[k]=1 else nhij1[k]=0
if (nhij[k]==2) nhij2[k]=1 else nhij2[k]=0
if (nhij[k]==3) nhij3[k]=1 else nhij3[k]=0}
detach(de)
dp=read.table("../prueba1.txt",sep="\t",header=T)
attach(dp)
tele1y=0
for(k in 1:1559){
if (tele[k]==1) tele1y[k]=1 else tele1y[k]=0}
sexo1y=0
for(k in 1:1559){
if (sexo[k]==1) sexo1y[k]=1 else sexo1y[k]=0}
esci2y=0
esci3y=0
for(k in 1:1559){
if (esci[k]==2) esci2y[k]=1 else esci2y[k]=0
if (esci[k]==3) esci3y[k]=1 else esci3y[k]=0}
prov2y=0
prov3y=0
prov4y=0
prov5y=0
prov6y=0
for(k in 1:1559){
if (prov[k]==2) prov2y[k]=1 else prov2y[k]=0
if (prov[k]==3) prov3y[k]=1 else prov3y[k]=0
if (prov[k]==4) prov4y[k]=1 else prov4y[k]=0
if (prov[k]==5) prov5y[k]=1 else prov5y[k]=0
if (prov[k]==6) prov6y[k]=1 else prov6y[k]=0}
tido2y=0
tido3y=0
for(k in 1:1559){
if (tido[k]==2) tido2y[k]=1 else tido2y[k]=0
if (tido[k]==3) tido3y[k]=1 else tido3y[k]=0}
nhij1y=0
nhij2y=0
nhij3y=0
for(k in 1:1559){
if (nhij[k]==1) nhij1y[k]=1 else nhij1y[k]=0
if (nhij[k]==2) nhij2y[k]=1 else nhij2y[k]=0
if (nhij[k]==3) nhij3y[k]=1 else nhij3y[k]=0}
detach(dp)

attach(de)
xc=cbind(edad,tres,mdes,ting,tgas)
xc=scale(xc)
x=as.data.frame(cbind(ries,xc,esci2,esci3,prov2,prov3,prov4,prov5,prov6,tele1,sexo1,tido2,tido3,nhij1,nhij2,nhij3))
detach(de)
attach(dp)
yc=cbind(edad,tres,mdes,ting,tgas)
yc=scale(yc)
y=as.data.frame(cbind(yc,esci2y,esci3y,prov2y,prov3y,prov4y,prov5y,prov6y,tele1y,sexo1y,tido2y,tido3y,nhij1y,nhij2y,nhij3y))
names(y)=names(x)[-1]
detach(dp)

modelo = vglm(ries ~ ., multinomial, x)
yt=predict(modelo,y)
e=exp(yt)
q1=e/(1+e)
q2=1/(1+e)
q=cbind(q1,q2)
c=0
for(j in 1:1559){
if (q[j,1]>q[j,2]) c[j]=1 else c[j]=2
}
confusion=function(true, predict){
jt=table(true,predict)
jn=dimnames(jt)[[2]]
jt1=jt[jn,]
structure(jt,error=(1-sum(diag(jt1))/length(true)))
```

```
}
confusion(dp$ries,c)

##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
##XXXXXXXXXXXXXXXX 50:50 XXXXXXXXXXX
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
de5=read.table("../entrenamientoimpu5050.txt",sep="\t",header=T)
attach(de5)
tele1=0
for(k in 1:1058){
if (tele[k]==1) tele1[k]=1 else tele1[k]=0}
sexo1=0
for(k in 1:1058){
if (sexo[k]==1) sexo1[k]=1 else sexo1[k]=0}
esci2=0
esci3=0
for(k in 1:1058){
if (esci[k]==2) esci2[k]=1 else esci2[k]=0
if (esci[k]==3) esci3[k]=1 else esci3[k]=0}
prov2=0
prov3=0
prov4=0
prov5=0
prov6=0
for(k in 1:1058){
if (prov[k]==2) prov2[k]=1 else prov2[k]=0
if (prov[k]==3) prov3[k]=1 else prov3[k]=0
if (prov[k]==4) prov4[k]=1 else prov4[k]=0
if (prov[k]==5) prov5[k]=1 else prov5[k]=0
if (prov[k]==6) prov6[k]=1 else prov6[k]=0}
tido2=0
tido3=0
for(k in 1:1058){
if (tido[k]==2) tido2[k]=1 else tido2[k]=0
if (tido[k]==3) tido3[k]=1 else tido3[k]=0}
nhij1=0
nhij2=0
nhij3=0
for(k in 1:1058){
if (nhij[k]==1) nhij1[k]=1 else nhij1[k]=0
if (nhij[k]==2) nhij2[k]=1 else nhij2[k]=0
if (nhij[k]==3) nhij3[k]=1 else nhij3[k]=0}
detach(de5)

attach(de5)
##xc=cbind(edad,tres,mdes,ting,tgas)
xc=cbind(ting)
xc=scale(xc)
##x=as.data.frame(cbind(ries,xc,esci2,esci3,prov2,prov3,prov4,prov5,prov6,tele1,sexo1,tido2,tido3,nhij1,nhij2,nhij3))
x=as.data.frame(cbind(ries,xc,tido2,tido3))
detach(de5)
attach(dp)
##yc=cbind(edad,tres,mdes,ting,tgas)
yc=cbind(ting)
yc=scale(yc)
##y=as.data.frame(cbind(yc,esci2y,esci3y,prov2y,prov3y,prov4y,prov5y,prov6y,tele1y,sexo1y,tido2y,tido3y,nhij1y,nhij2y,nhij3y))
y=as.data.frame(cbind(yc,tido2y,tido3y))
names(y)=names(x)[-1]
detach(dp)

modelo = vglm(ries ~ ., multinomial, x)
yt=predict(modelo,y)
e=exp(yt)
q1=e/(1+e)
q2=1/(1+e)
q=cbind(q1,q2)
c=0
for(j in 1:1559){
if (q[j,1]>q[j,2]) c[j]=1 else c[j]=2
}
confusion=function(true, predict){
```

```
jt=table(true,predict)
jn=dimnames(jt)[[2]]
jt1=jt[jn,]
structure(jt,error=(1-sum(diag(jt1))/length(true)))
}
confusion(dp$ries,c)

roc1=0
roc2=0
for(j in 1:1559){
if (q[j,1]>q[j,2]) roc1[j]=q[j,2] else roc1[j]=q[j,1]
if (q[j,1]>q[j,2]) roc2[j]=1 else roc2[j]=2
}
library(ROCR)
objpred=prediction(roc1,roc2)
perf=performance(objpred,"tpr","fpr")
plot(perf,colorize=TRUE,xlab=expression(1-e_1),ylab=expression(e_2))
abline(0,1)



p_left=rnorm(1000,3,1)
p_right=rnorm(1000,10,1)
p=c(p_left,p_right)
e=c(rep(1,1000),rep(2,1000))
pred=prediction(p,e)
perf=performance(pred,"tpr","fpr")
par(mfrow=c(1,2))
plot(perf,colorize=TRUE)
hist(p)
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
##XXXXXX ÁRBOLES DE CLASIFICACIÓN CON VARIABLES CONTINUAS Y CATEGÓRICAS XXXXXXXXXXXXXXXX
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

library(tree)
de=read.table("../entrenamientoimpu.txt",sep="\t",header=T)
de5=read.table("../entrenamientoimpu5050.txt",sep="\t",header=T)
dp=read.table("../prueba1.txt",sep="\t",header=T)
attach(de)
telet=factor(tele,labels=c("No","Si"))
sexot=factor(sexo,labels=c("Mujer","Hombre"))
escit=factor(esci,labels=c("Casado","Soltero","Otro"))
tidot=factor(tido,labels=c("Arrendada","de Familiar","Propia"))
provt=factor(prov,labels=c("Cotopaxi","Guayas","Pichincha","Oriente","Otras Costa","Otras Sierra"))
riest=factor(ries,labels=c("Malo","Bueno"))
detach(de)
x=de
x$tele=telet
x$sexo=sexot
x$esci=escit
x$tido=tidot
x$prov=provt
x$ries=riest
##minsize: umbral para el tamaño del nodo, así los nodos de tamaño minzise o más son candidatos a dividir,
##los nodos hijos deberían superar a mincut para que se permita la division
arbol=tree(ries~.,data=x,split="gini",mincut=50,minsize=100,mindev=1e-6)
plot(arbol)
text(arbol,cex=0.8)
attach(dp)
telet1=factor(tele,labels=c("No","Si"))
sexot1=factor(sexo,labels=c("Mujer","Hombre"))
escit1=factor(esci,labels=c("Casado","Soltero","Otro"))
tidot1=factor(tido,labels=c("Arrendada","de Familiar","Propia"))
provt1=factor(prov,labels=c("Cotopaxi","Guayas","Pichincha","Oriente","Otras Costa","Otras Sierra"))
riest1=factor(ries,labels=c("Malo","Bueno"))
detach(dp)
y=dp
y$tele=telet1
y$sexo=sexot1
y$esci=escit1
y$tido=tidot1
```

```
y$prov=provt1
y$ries=riest1
yt=predict(arbol,y[-6],type="class")
confusion=function(true, predict){
jt=table(true,predict)
jn=dimnames(jt)[[2]]
jt1=jt[jn,]
structure(jt,error=(1-sum(diag(jt1))/length(true)))
}
confusion(y$ries,yt)

##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
##XXXXXXXXXXXXXXXXX 50:50 XXXXXXXXXXXX
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX


##XXXXXXXXXXXXXXXXXXXXX
##XXX con deviance XXXXX
##XXXXXXXXXXXXXXXXXXXXX

attach(de5)
telet=factor(tele,labels=c("No","Si"))
sexot=factor(sexo,labels=c("Mujer","Hombre"))
escit=factor(esci,labels=c("Casado","Soltero","Otro"))
tidot=factor(tido,labels=c("Arrendada","de Familiar","Propia"))
provt=factor(prov,labels=c("Cotopaxi","Guayas","Pichincha","Oriente","Otras Costa","Otras Sierra"))
riest=factor(ries,labels=c("B","G"))
nhijt=factor(nhij,labels=c("0","1","2","3+"))
detach(de5)
x=de5
x$tele=telet
x$sexo=sexot
x$esci=escit
x$tido=tidot
x$prov=provt
x$ries=riest
x$nhij=nhijt
arbol=tree(ries~.,x,minsize=50,mincut=25,mindev=0) ##50 individuos representan el 5%, y quiero un arbol q se ajuste totalmente a los datos
plot(arbol)
text(arbol,cex=0.5)
##aqui podo a mi criterio
arbol.snip=snip.tree(arbol,nodes=c(10,12,32,67,88))
plot(arbol.snip)
text(arbol.snip,cex=0.6)
##pasa de un arbol con 34 nodos y un error de .3261 a un arbol de 27 nodos y error de .3261
##ahorra voy a podar arbol.snip con el método prune
plot(prune.tree(arbol.snip))
##parece que al disminuir a un arbol de 22 nodos la deviance no sube mucho
arbol.prune=prune.tree(arbol.snip,best=22)
plot(arbol.prune)
text(arbol.prune,cex=0.6)
##podamos en base a los datos del conjunto de prueba
attach(dp)
telet1=factor(tele,labels=c("No","Si"))
sexot1=factor(sexo,labels=c("Mujer","Hombre"))
escit1=factor(esci,labels=c("Casado","Soltero","Otro"))
tidot1=factor(tido,labels=c("Arrendada","de Familiar","Propia"))
provt1=factor(prov,labels=c("Cotopaxi","Guayas","Pichincha","Oriente","Otras Costa","Otras Sierra"))
riest1=factor(ries,labels=c("B","G"))
nhijt1=factor(nhij,labels=c("0","1","2","3+"))
detach(dp)
y=dp
y$tele=telet1
y$sexo=sexot1
y$esci=escit1
y$tido=tidot1
y$prov=provt1
y$ries=riest1
y$nhij=nhijt1
arbol.newdata=prune.tree(arbol.snip,best=22,newdata=y)
plot(arbol.newdata)
```

```
text(arbol.newdata,cex=0.6)
##podamos de acuerdo a validacion cruzada
##arbol.cv=cv.tree(arbol.prune,,prune.tree)
##plot(arbol.cv)
##no hay chance con cv xq dice q tine un solo nodo:raro noc q es
yt=predict(arbol.snip,y[-6],type="class")
confusion=function(true, predict){
jt=table(true,predict)
jn=dimnames(jt)[[2]]
jt1=jt[jn,]
structure(jt,error=(1-sum(diag(jt1))/length(true)))
}
confusion(y$ries,yt)

##XXXXXXXXXXXXXXXXXXXXXX
##XXX con gini XXXXXXXXX
##XXXXXXXXXXXXXXXXXXXXXX
attach(de5)
telet=factor(tele,labels=c("No","Si"))
sexot=factor(sexo,labels=c("Mujer","Hombre"))
escit=factor(esci,labels=c("Casado","Soltero","Otro"))
tidot=factor(tido,labels=c("Arrendada","de Familiar","Propia"))
provt=factor(prov,labels=c("Cotopaxi","Guayas","Pichincha","Oriente","Otras Costa","Otras Sierra"))
riest=factor(ries,labels=c("B","G"))
nhijt=factor(nhij,labels=c("0","1","2","3+"))
detach(de5)
x=de5
x$tele=telet
x$sexo=sexot
x$esci=escit
x$tido=tidot
x$prov=provt
x$ries=riest
x$nhij=nhijt
arbol=tree(ries~.,x,minsize=50,mincut=25,mindev=0,spli="gini") ##50 individuos representan el 5%, y quiero un arbol q se ajuste totalmente a los datos
plot(arbol)
text(arbol,cex=0.5)
##aqui podo a mi criterio
arbol.snip=snip.tree(arbol,nodes=c(11,14,30,62,8190,131070,131071))
plot(arbol.snip)
text(arbol.snip,cex=0.8)
##pasa de un arbol con 34 nodos y un error de .3261 a un arbol de 27 nodos y error de .3261
##ahorra voy a podar arbol.snip con el método prune
plot(prune.tree(arbol.snip))
##parece que al disminuir a un arbol de 22 nodos la deviance no sube mucho
arbol.prune=prune.tree(arbol.snip,best=20,method="misclass")
plot(arbol.prune)
text(arbol.prune,cex=0.6)
##podamos en base a los datos del conjunto de prueba
attach(dp)
telet1=factor(tele,labels=c("No","Si"))
sexot1=factor(sexo,labels=c("Mujer","Hombre"))
escit1=factor(esci,labels=c("Casado","Soltero","Otro"))
tidot1=factor(tido,labels=c("Arrendada","de Familiar","Propia"))
provt1=factor(prov,labels=c("Cotopaxi","Guayas","Pichincha","Oriente","Otras Costa","Otras Sierra"))
riest1=factor(ries,labels=c("B","G"))
nhijt1=factor(nhij,labels=c("0","1","2","3+"))
detach(dp)
y=dp
y$tele=telet1
y$sexo=sexot1
y$esci=escit1
y$tido=tidot1
y$prov=provt1
y$ries=riest1
y$nhij=nhijt1
arbol.newdata=prune.tree(arbol.snip,best=20,method="misclass",newdata=y)
plot(arbol.newdata)
text(arbol.newdata,cex=0.6)
##podamos de acuerdo a validacion cruzada
##arbol.cv=cv.tree(arbol.prune,,prune.tree)
```

```
##plot(arbol.cv)
##no hay chance con cv xq dice q tine un solo nodo:raro noc q es
yt=predict(arbol.newdata,y[-6],type="class")
confusion=function(true, predict){
jt=table(true,predict)
jn=dimnames(jt)[[2]]
jt1=jt[jn,]
structure(jt,error=(1-sum(diag(jt1))/length(true)))
}
confusion(y$ries,yt)



##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
##XXXXXXXXXXXX REDES NEURONALES VARIABLES CONTINUAS XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

library(nnet)
de=read.table("../entrenamientoimpu.txt",sep="\t",header=T)
de5=read.table("../entrenamientoimpu5050.txt",sep="\t",header=T)
dp=read.table("../prueba1.txt",sep="\t",header=T)
attach(de)
x=scale(cbind(edad,tres,mdes,ting,tgas))
riesf=factor(ries)
riesi=class.ind(riesf)
detach(de)
attach(dp)
y=scale(cbind(edad,tres,mdes,ting,tgas))
detach(dp)
set.seed(777)
neural=nnet(x,riesi,size=5,decay=1e-3,entropy=T,softmax=T,skip=T,maxit=500)
yp=predict(neural,y,type="class")
table(dp$ries,yp)

attach(de5)
x=scale(cbind(edad,tres,mdes,ting,tgas))
riesf=factor(ries)
riesi=class.ind(riesf)
detach(de5)
attach(dp)
y=scale(cbind(edad,tres,mdes,ting,tgas))
detach(dp)
set.seed(777)
neural=nnet(x,riesi,size=4,decay=0,entropy=T,softmax=T,skip=T,maxit=500)
yp=predict(neural,y,type="class")
confusion=function(true, predict){
jt=table(true,predict)
jn=dimnames(jt)[[2]]
jt1=jt[jn,]
structure(jt,error=(1-sum(diag(jt1))/length(true)))
}
confusion(dp$ries,yp)

##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
##XXX continuas y categoricas XXXXXXXXX
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

attach(de5)
tele1=0
for(k in 1:1058){
if (tele[k]==1) tele1[k]=1 else tele1[k]=0}
sexo1=0
for(k in 1:1058){
if (sexo[k]==1) sexo1[k]=1 else sexo1[k]=0}
esci2=0
esci3=0
for(k in 1:1058){
if (esci[k]==2) esci2[k]=1 else esci2[k]=0
if (esci[k]==3) esci3[k]=1 else esci3[k]=0}
prov2=0
prov3=0
```

```
prov4=0
prov5=0
prov6=0
for(k in 1:1058){
if (prov[k]==2) prov2[k]=1 else prov2[k]=0
if (prov[k]==3) prov3[k]=1 else prov3[k]=0
if (prov[k]==4) prov4[k]=1 else prov4[k]=0
if (prov[k]==5) prov5[k]=1 else prov5[k]=0
if (prov[k]==6) prov6[k]=1 else prov6[k]=0}
tido2=0
tido3=0
for(k in 1:1058){
if (tido[k]==2) tido2[k]=1 else tido2[k]=0
if (tido[k]==3) tido3[k]=1 else tido3[k]=0}
nhij1=0
nhij2=0
nhij3=0
for(k in 1:1058){
if (nhij[k]==1) nhij1[k]=1 else nhij1[k]=0
if (nhij[k]==2) nhij2[k]=1 else nhij2[k]=0
if (nhij[k]==3) nhij3[k]=1 else nhij3[k]=0}
x=scale(cbind(edad,tres,mdes,ting,tgas,esci2,esci3,prov2,prov3,prov4,prov5,prov6,tele1,sexo1,tido2,tido3,nhij1,nhij2,nhij3))
riesf=factor(ries)
riesi=class.ind(riesf)
detach(de5)
attach(dp)
tele1y=0
for(k in 1:1559){
if (tele[k]==1) tele1y[k]=1 else tele1y[k]=0}
sexo1y=0
for(k in 1:1559){
if (sexo[k]==1) sexo1y[k]=1 else sexo1y[k]=0}
esci2y=0
esci3y=0
for(k in 1:1559){
if (esci[k]==2) esci2y[k]=1 else esci2y[k]=0
if (esci[k]==3) esci3y[k]=1 else esci3y[k]=0}
prov2y=0
prov3y=0
prov4y=0
prov5y=0
prov6y=0
for(k in 1:1559){
if (prov[k]==2) prov2y[k]=1 else prov2y[k]=0
if (prov[k]==3) prov3y[k]=1 else prov3y[k]=0
if (prov[k]==4) prov4y[k]=1 else prov4y[k]=0
if (prov[k]==5) prov5y[k]=1 else prov5y[k]=0
if (prov[k]==6) prov6y[k]=1 else prov6y[k]=0}
tido2y=0
tido3y=0
for(k in 1:1559){
if (tido[k]==2) tido2y[k]=1 else tido2y[k]=0
if (tido[k]==3) tido3y[k]=1 else tido3y[k]=0}
nhij1y=0
nhij2y=0
nhij3y=0
for(k in 1:1559){
if (nhij[k]==1) nhij1y[k]=1 else nhij1y[k]=0
if (nhij[k]==2) nhij2y[k]=1 else nhij2y[k]=0
if (nhij[k]==3) nhij3y[k]=1 else nhij3y[k]=0}
y=scale(cbind(edad,tres,mdes,ting,tgas,esci2y,esci3y,prov2y,prov3y,prov4y,prov5y,prov6y,tele1y,sexo1y,tido2y,tido3y,nhij1y,nhij2y,nhij3y))
detach(dp)
set.seed(777)
neural=nnet(x,riesi,size=5,decay=0.001,entropy=T,softmax=T,skip=T,maxit=500)
yp=predict(neural,y,type="class")
confusion=function(true, predict){
jt=table(true,predict)
jn=dimnames(jt)[[2]]
jt1=jt[jn,]
structure(jt,error=(1-sum(diag(jt1))/length(true)))
}
```

```
confusion(dp$ries,yp)


##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
##XXXXX SUPPORT VECTOR MACHINES VARIABLES CONTINUAS XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
##XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
library(class)
library(e1071)
de=read.table("../entrenamientoimpu.txt",sep="\t",header=T)
de5=read.table("../entrenamientoimpu5050.txt",sep="\t",header=T)
dp=read.table("../prueba1.txt",sep="\t",header=T)
##el scalamiento lo hace el mismo programa
attach(de5)
riesf=factor(ries,labels=c("BR","GR"))
x=cbind(edad,tres,mdes,ting,tgas)
detach(de5)
attach(dp)
riesfy=factor(ries,labels=c("BRy","GRy"))
y=cbind(edad,tres,mdes,ting,tgas)
detach(dp)
machine=svm(x,riesf)
yp=predict(machine,y)
table(dp$ries,yp)


attach(de5)
tele1=0
for(k in 1:1058){
if (tele[k]==1) tele1[k]=1 else tele1[k]=0}
sexo1=0
for(k in 1:1058){
if (sexo[k]==1) sexo1[k]=1 else sexo1[k]=0}
esci2=0
esci3=0
for(k in 1:1058){
if (esci[k]==2) esci2[k]=1 else esci2[k]=0
if (esci[k]==3) esci3[k]=1 else esci3[k]=0}
prov2=0
prov3=0
prov4=0
prov5=0
prov6=0
for(k in 1:1058){
if (prov[k]==2) prov2[k]=1 else prov2[k]=0
if (prov[k]==3) prov3[k]=1 else prov3[k]=0
if (prov[k]==4) prov4[k]=1 else prov4[k]=0
if (prov[k]==5) prov5[k]=1 else prov5[k]=0
if (prov[k]==6) prov6[k]=1 else prov6[k]=0}
tido2=0
tido3=0
for(k in 1:1058){
if (tido[k]==2) tido2[k]=1 else tido2[k]=0
if (tido[k]==3) tido3[k]=1 else tido3[k]=0}
nhij1=0
nhij2=0
nhij3=0
for(k in 1:1058){
if (nhij[k]==1) nhij1[k]=1 else nhij1[k]=0
if (nhij[k]==2) nhij2[k]=1 else nhij2[k]=0
if (nhij[k]==3) nhij3[k]=1 else nhij3[k]=0}
riesf=factor(ries,labels=c("BR","GR"))
x=cbind(edad,tres,mdes,ting,tgas,esci2,esci3,prov2,prov3,prov4,prov5,prov6,tele1,sexo1,tido2,tido3,nhij1,nhij2,nhij3)
detach(de5)
attach(dp)
tele1y=0
for(k in 1:1559){
if (tele[k]==1) tele1y[k]=1 else tele1y[k]=0}
sexo1y=0
for(k in 1:1559){
if (sexo[k]==1) sexo1y[k]=1 else sexo1y[k]=0}
esci2y=0
```

```
esci3y=0
for(k in 1:1559){
if (esci[k]==2) esci2y[k]=1 else esci2y[k]=0
if (esci[k]==3) esci3y[k]=1 else esci3y[k]=0}
prov2y=0
prov3y=0
prov4y=0
prov5y=0
prov6y=0
for(k in 1:1559){
if (prov[k]==2) prov2y[k]=1 else prov2y[k]=0
if (prov[k]==3) prov3y[k]=1 else prov3y[k]=0
if (prov[k]==4) prov4y[k]=1 else prov4y[k]=0
if (prov[k]==5) prov5y[k]=1 else prov5y[k]=0
if (prov[k]==6) prov6y[k]=1 else prov6y[k]=0}
tido2y=0
tido3y=0
for(k in 1:1559){
if (tido[k]==2) tido2y[k]=1 else tido2y[k]=0
if (tido[k]==3) tido3y[k]=1 else tido3y[k]=0}
nhij1y=0
nhij2y=0
nhij3y=0
for(k in 1:1559){
if (nhij[k]==1) nhij1y[k]=1 else nhij1y[k]=0
if (nhij[k]==2) nhij2y[k]=1 else nhij2y[k]=0
if (nhij[k]==3) nhij3y[k]=1 else nhij3y[k]=0}
riesfy=factor(ries,labels=c("BRy","GRy"))
y=cbind(edad,tres,mdes,ting,tgas,esci2y,esci3y,prov2y,prov3y,prov4y,prov5y,prov6y,tele1y,sexo1y,tido2y,tido3y,nhij1y,nhij2y,nhij3y)
detach(dp)
machine=svm(x,riesf)
yp=predict(machine,y)
table(dp$ries,yp)
```

Bibliografía

[1] Bridges S. and Disney R. Modelling consumer credit and default: The research agenda. *ExCEM, University of Nottingham*, 2001.

[2] Burges C. A tutorial on support vector machines for pattern recognition. *Kluwer Academic Publishers*.

[3] Altman E. Caouette J. and Narayanan P. *Managing Credit Risk: The next great financial challenge*. John Wiley and Sons, first edition, 1998.

[4] Tibshirani R. Hastie T. and Friedman J. . *The Elements of Statistical Learning*. Springer, first edition, 2001.

[5] Johnson D. *Métodos Multivariados Aplicados al Análisis de Datos*. Thomson, primera edition, 2000.

[6] Komorád K. On credit scoring estimation. Master's thesis, Humboldt University, Institute for Statistics and Econometrics, December 2002.

[7] Liu Y. The evaluation of classification models for credit scoring. Master's thesis, Georg August Universität, Institut für Wirtschaftsinformatik, 2002.

[8] Rice J. A. *Mathematical Statistics and Data Analysis*. Duxbury Press, second edition, 1995.

[9] Venables W. and Ripley B. *Modern Applied Statistics with S-Plus*. Springer-Verlag, first edition, 1994.

[10] Rosenberg E. and Gleit A. Quantitative methods in credit management: a survey. *Operations Research*, 42(4):589–613, 1994.

[11] Seber G. A. *Multivariate Observations*. John Wiley and Sons, first edition, 1984.

[12] Svantesson T. and Wallace J. Tests for assessing multivariate normality and the covariance structure of mimo data. *Brigham Young University.*

[13] Edelman D. Thomas L. and Crook J. *Credit Scoring and Its Applications.* Society for Industrial and Applied Mathematics, first edition, 2002.

[14] Welling M. Fisher linear discriminant analysis. *Department of Computer Science, University of Toronto*, 2003.

[15] Williams D. *Weighing the Odds, A course in probability and statistics.* Cambridge, first edition, 2001.