Selección de atributos y support vector machines adaptado al problema de fuga de clientes

Álvaro Flores [*]
Sebastián Maldonado[**]
Richard Weber[*]

Resumen

Uno de los grandes desafíos de la Minería de Datos aplicada al Análisis de Negocios es la selección de atributos para un modelo de clasificación. La mayoría de las técnicas de selección de atributos se basan en criterios de validación estadística, perdiendo en muchos casos el objetivo del negocio en sí mismo, lo que no necesariamente lleva a modelos que optimicen las metas definidas. Para generar el modelo y la selección de atributos se utiliza un enfoque basado en utilidades utilizando el modelo de Support Vector Machines, donde las métricas basadas en utilidades simulan la realización de una campaña de retención de clientes considerando beneficios y costos (Maximum Profit Criterion (MPC) y Expected Maximum Profit Criterion (EMPC)) o bien sólo costos, como es el caso de H-measure*. El enfoque presentado en este trabajo consiste en un método de selección de atributos empotrado en la construcción del modelo clasificador, que apunta a la eliminación secuencial de atributos removiendo los que tienen menor relevancia de acuerdo a estas métricas. Utilizando un caso dentro del área de Telecomunicaciones, los resultados indican que estos métodos de selección de atributos y evaluación de modelos son más estables y obtienen mejores resultados tanto en términos de métricas usuales de evaluación de modelos predictivos, como en métricas de desempeño basadas en utilidades orientadas al negocio.*

Palabras Clave: *Fuga de Clientes, Selección de atributos, Support Vector Machines.*

[*]Departamento de Ingeniería Industrial, Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile, Santiago, Chile.

[**]Facultad de Ingeniería y Ciencias Aplicadas, Universidad de los Andes, Santiago, Chile.

1. Introducción

La clasificación es una tarea relevante en muchas aplicaciones orientadas a mejorar las utilidades de una empresa, tales como *Credit Scoring* o Predicción de Fuga de Clientes [2]. Además, se ha demostrado que el desempeño de un clasificador puede ser mejorado enfocándose en los atributos más relevantes usados para la construcción de éste. La selección de atributos tiene importantes ventajas:

1. Una representación usando menos atributos realza el poder predictivo de los modelos de clasificación disminuyendo su complejidad, reduciendo de esta manera el riesgo de *Overfitting (sobreajuste)*, causado por la *Maldición de la Dimensionalidad* [20].

2. Dicha selección permite una mejor interpretación del clasificador, lo que es particularmente importante en *Business Analytics*, puesto que muchos profesionales consideran que las técnicas de *Machine Learning* son *cajas negras* y se rehúsan a emplear estos métodos debido a su complejidad [2].

Tratar de predecir qué clientes van a dejar una compañía, fugarse o simplemente *churn*[1], como se conoce en la literatura, es una de las tareas más importantes de las empresas de servicio, principalmente la banca y telecomunicaciones. La importancia de la predicción de fuga de cliente se incrementa debido a la creciente cantidad de clientes dispuestos a cambiar sus proveedores, junto a la fuerte competencia por captar a clientes nuevos. Es por esto que surge la necesidad de crear y desarrollar modelos capaces de identificar clientes actuales con tendencia a dejar la compañía en un periodo dado de tiempo.

El *Churn* puede ser observado de dos maneras diferentes: **voluntario**, en donde el cliente decide terminar el contrato, o bien **involuntario**, donde la compañía en cuestión decide terminar el contrato con el cliente [3]. En este trabajo se estudia el *churn* como un fenómeno voluntario.

Uno de los objetivos relevantes al realizar modelos de predicción de fuga, es establecer estrategias orientadas a la retención del cliente. Si la compañía es capaz de identificar los posibles *churners*, el siguiente paso es desarrollar campañas comerciales y estrategias de retención enfocadas en este grupo en particular, potenciando de esta manera la lealtad del cliente y obteniendo otros beneficios, como por ejemplo:

[1] en este trabajo se hace referencia indistintamente a fuga o *churn*

- Un incremento en la proporción de clientes fieles, los cuales generan 1,7 veces más ingreso que los otros clientes [12].
- Un impacto directo en la rentabilidad: un 5 % de incremento en la tasa de retención de clientes, puede llevar a un 18 % de reducción en costos operacionales [12].
- Una disminución del gasto en retención innecesario, enfocando los recursos en clientes en riesgo de fuga y no en la base de clientes completa; reduciendo así los costos operacionales y de marketing [25].

Atendiendo a estos hechos, la tasa de fuga es puesta explícitamente en la fórmula para el *Customer Lifetime Value* (en adelante **CLV**), que toma la siguiente forma considerando periodos anuales [3]:

$$CLV = \sum_{t=1}^{\infty} \frac{m(1-c)^{t-1}}{(1+r)^{t-1}} = m\frac{(1+r)}{(r+c)} \tag{1}$$

en donde c es la tasa anual de fuga y m es el retorno esperado medio por cliente. El Parámetro r es la tasa de descuento anual. Existen dos maneras clásicas de determinar este último valor. La primera es obtener directamente el *Weighted Average Cost of Capital (WACC)* de la compañía. La segunda, es usar la tasa de descuento del sector industrial en particular. Se puede entender el **CLV** como el valor presente neto del beneficio por cliente, luego un descenso en la tasa de fuga de clientes impacta de manera directa en las utilidades de la empresa.

El fenómeno de la fuga de clientes puede ser modelado con técnicas que dependen del tiempo [3], o bien como predicciones sobre el siguiente periodo. En el primer caso este tipo de modelos no asume que la fuga ocurrirá explícitamente en un período de tiempo, proponiendo probabilidades de fuga hasta un número fijo de períodos desde el origen de los datos, pudiendo incluso variar con el tiempo [3]. En el segundo caso, encontramos formas de enfocarnos a predecir si un cliente decide o no fugarse en el período siguiente, donde los enfoques más tradicionales son regresión logística [5, 17, 22], modelos estadísticos no paramétricos como *K-nearest neighbors (KNN)* [8], árboles de decisión [31], y redes neuronales [16]. Una revisión sobre la modelación de fuga de clientes, puede ser encontrada en Verbeke et al. [29]. En este trabajo se utilizan clasificadores basados en SVM, prediciendo la fuga de los clientes en el siguiente período de tiempo.

Para poder evaluar el desempeño de un clasificador, es necesario usar alguna medida de rendimiento que permita conocer el grado de asertividad que

tiene dicho clasificador, esto a su vez permite comparar entre clasificadores. En la literatura se proponen muchas métricas de rendimiento para evaluar el desempeño de un clasificador. Una revisión más exhaustiva del tema puede ser encontrada en [7].

Cuando se usan modelos de predicción de fuga mensual, las tasa de fuga usualmente se mantiene bajo el 5 % [28], lo que nos lleva naturalmente al problema de desbalance de clases que también será tratado en los modelos propuestos en este artículo.

La estructura de este trabajo es la siguiente: La Sección 2 presenta la derivación del método de clasificación Support Vector Machines. Técnicas recientes de selección de atributos para Support Vector Machines se presentan en la Sección 3. La Sección 4 describe la metodología para selección de atributos propuesta en este trabajo. La Sección 5 presenta los principales resultados. Finalmente, la Sección 6 muestra las conclusiones del trabajo.

2. Support Vector Machines

Support Vector Machine (SVM) [27] es un modelo de clasificación basado en minimizar el error cuadrático de la clasificación, construyendo un hiperplano que separa los datos de la forma más precisa posible. SVM es considerado uno de los modelos más precisos y robustos posibles dentro de los algoritmos de clasificación binaria [32], siendo ampliamente utilizado por su versatilidad y efectividad a la hora de clasificar. Esto último se logra principalmente porque incluye la minimización del error estructural al clasificar nuevos objetos, que está directamente relacionado con uno de los principales objetivos de un clasificador, que es tener la habilidad de *generalizar* de forma correcta.

El método SVM define un hiperplano en $\mathbb{R}^M$, donde M es la cantidad de atributos, que separa lo mejor posible una clase de la otra. El objetivo es maximizar la distancia entre el hiperplano óptimo y los hiperplanos canónicos (que representan los *bordes* de cada clase). Para lograr esto se minimiza la norma Euclidiana de $\mathbf{w}$, que coresponde a los coeficientes que definen el hiperplano, dando origen al siguiente problema de minimización como formulación primal:

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}\left\|\mathbf{w}\right\|^2 + C\sum_{i=1}^{N}\xi_i \\ \text{s.a.} \quad & y_i \cdot (\mathbf{w}^\top \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \qquad i = 1, \ldots, N, \\ & \xi_i \geq 0, \qquad i = 1, \ldots, N, \end{aligned} \tag{2}$$

donde $\mathbf{x_i} \in \mathbb{R}^{\mathbf{M}}$, $y_i \in \{-1, 1\}$, y ξ_i $(i = 1, \ldots, N)$ son variables de holgura que tienen por objetivo relajar las restricciones, permitiendo que ocurran errores, pero penalizándolos en la función objetivo. Esta penalización se controla con un parámetro C.

La formulación previa puede ser extendida a clasificadores no lineales, usando el *kernel trick*: Los datos de entrenamiento son transformados en un espacio de mayor dimensionalidad $\mathscr{H}$, a través de una función $\phi : \mathbf{x} \to \phi(\mathbf{x}) \in \mathscr{H}$ [23]. Una función de Kernel $K(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x})^\top \cdot \phi(\mathbf{y})$ define un producto interno en el espacio de $\mathscr{H}$, lo que nos lleva a la siguiente formulación (luego de calcular el dual):

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,s=1}^{N} \alpha_i \alpha_s y_i y_s K(\mathbf{x}_i, \mathbf{x}_s) \\ \text{s.a.} \quad & \sum_{i=1}^{N} \alpha_i y_i = 0, \\ & 0 \leq \alpha_i \leq C, \qquad i = 1, \ldots, N. \end{aligned} \tag{3}$$

3. Selección de Atributos para SVM

En esta sección se hará una revisión del estado del arte en lo que a selección de variables se refiere, y se explican los tres enfoques principales [13], así como los métodos que serán usados en este trabajo.

3.1. Métodos de Filtro (*Filter Methods*)

La aplicación de este enfoque, ocurre antes de aplicar cualquier algoritmo de clasificación, y usa propiedades estadísticas de los atributos, con el objetivo de dejar fuera los que aportan menos *información* (de acuerdo a alguna métrica) al modelo. Ejemplos clásicos de la literatura, son **el estadístico de** χ^2, que mide la dependencia entre la distribución de cada atributo y las etiquetas de las observaciones [26], la **Ganancia de Información** que usa la entropía para medir la relevancia de un atributo [26], y finalmente el **Fisher Score** (F), que estima la relevancia de cada atributo calculando la diferencia (absoluta) entre las medias del valor de la variable en ambas clases, normalizando según la suma de las varianzas intra-clases:

$$F(j) = \left| \frac{\mu_j^+ - \mu_j^-}{(\sigma_j^+)^2 + (\sigma_j^-)^2} \right| \tag{4}$$

en donde μ_j^+ (μ_j^-) es la media del j-ésimo atributo en la clase positiva (negativa) y σ_j^+ (σ_j^-) es la correspondiente desviación estándar.

Usando este indicador, es posible observar qué atributos difieren *más* entre clases. El **Método de Fisher** para selección de atributos consiste en obtener el indicador señalado en la ecuación (4) para todos los atributos y elegir la cantidad de atributos deseada que tengan mejor *Fisher Score*.

3.2. Métodos de Envoltura (*Wrapper Methods*)

Estos métodos buscan entre los posibles subconjuntos de atributos, evaluando su potencial predictivo. Esto es altamente demandante en términos computacionales, puesto que la cantidad de subconjuntos a revisar tiene un tamaño exponencial en la cantidad de atributos. Las estrategias más populares para llevar a cabo esta tarea son *Sequential Forward Selection (SFS)* y *Sequential Backward Elimination (SBE)*. SFS empieza con un conjunto vacío de atributos, y luego intenta agregar variables de manera secuencial, donde en cada paso se agrega la más relevante (de acuerdo a algún método de clasificación en particular) del conjunto de atributos pendientes por agregar. SBE, por su lado, empieza con el espacio completo de variables, y calcula la significancia estadística de cada una, eliminando en cada iteración la menos relevante.

3.3. Métodos Empotrados (*Embedded Methods*)

Estos métodos realizan la selección de atributos de manera simultánea a la construcción del clasificador y son específicos para cada técnica de clasificación. Por ende incluyen la interacción entre los atributos y el clasificador en el proceso de modelación. Los métodos *embedded* son computacionalmente menos intensivos que las estrategias *wrapper* [13].

Una técnica muy popular y relevante para el desarrollo de este trabajo es *Recursive Feature Elimination* (RFE-SVM [14]). El objetivo de este método es encontrar un subconjunto de tamaño r entre n variables (con $r < n$), eliminando aquellos atributos cuya extracción contribuye a alcanzar el mayor margen de separación entre clases. Esto puede ser logrado utilizando una estrategia SBE, eliminando de manera secuencial atributos basándose en las componentes del vector de pesos en SVM $\mathbf{w}$. El caso lineal toma la siguiente forma:

Algorithm 1 *Recursive Feature Elimination, SVM* - Caso Lineal

1. **repetir**
2. $\mathbf{w} \leftarrow$ Entrenamiento SVM (formulación primal).
3. Eliminar el atributo p con el valor más pequeño de $|w_p|$.
4. **hasta** reducir la cantidad de atributos a r.

Cabe notar que el caso RFE-SVM lineal puede extenderse al caso no-lineal (es decir, usando funciones de Kernel), notando que la distancia de los hiperplanos canónicos que separan ambas clases (el **margen**) es inversamente proporcional a la norma del vector de pesos $\mathbf{w}$ y este último valor puede ser escrito en términos de las variables duales del modelo generado por SVM, tomando la siguiente forma funcional:

$$W^2(\alpha) = \sum_{i,s=1}^{N} \alpha_i \alpha_s y_i y_s K(\mathbf{x}_i, \mathbf{x}_s) \tag{5}$$

El atributo removido en cada iteración, es aquel cuya eliminación minimice la variación de $W^2(\boldsymbol{\alpha})$. El algoritmo queda descrito como sigue:

Algorithm 2 Recursive Feature Elimination, SVM - Caso no Lineal

1. **repetir**
2. $\mathbf{w} \leftarrow$ Entrenamiento SVM (formulación dual).
3. Eliminar el atributo p con menor valor de $|W^2(\boldsymbol{\alpha}) - W^2_{(-p)}(\boldsymbol{\alpha})|$.
4. **hasta** reducir la cantidad de atributos a r.

4. Metodología propuesta

A continuación se detalla la metodología propuesta para la selección de atributos maximizando el beneficio asociado. En primera instancia, se describen las métricas utilizadas para evaluar el desempeño de una campaña de retención de clientes a partir de la solución entregada por un modelo de clasificación, para luego detallar el algoritmo propuesto.

4.1. Fuga de Clientes y Medidas de desempeño

La fuga de clientes puede ser modelada como una clasificación binaria donde un cliente pertenece a una de dos clases: clientes leales (*non-churners*) o clientes fugados (*churners*). Dado un objeto **x** (observación, evento, cliente, etc.) un clasificador $\mathcal{C}$ producirá una puntuación s, donde por convención una puntuación más alta implica que tiene mayor tendencia a ser etiquetado con (+1), es decir, *churn* en nuestro caso. Se fija un valor de **umbral** t para proveer una clasificación binaria de la base completa basada en sus puntajes. De esta manera todas las instancias que tengan una puntuación s menor que t son clasificados como *no-churners* (-1) y los clientes con s mayor o igual a t son clasificados como *churners* (+1).

Se consideran las siguientes definiciones (notación presentada en [28]):

- **Probabilidades a Priori**: π_{-1} y π_1 son las probabilidades a priori de que una observación posea la etiqueta -1 o 1, respectivamente. Notemos que $\pi_{-1} + \pi_1 = 1$, es decir, son las únicas posibilidades para una observación.

- **Distribuciones de Probabilidades**: Dado una puntuación s, las funciones de densidad de probabilidad para los *non-churners* y los *churners* son respectivamente $f_{-1}(s)$ y $f_1(s)$, y las funciones de densidad acumulativa son denotadas por $F_{-1}(s)$ y $F_1(s)$.

- **Términos de Costo-Beneficio**: Se define b_{-1} (b_1) como el beneficio obtenido de clasificar correctamente a un *non-churner (churner)*, y c_{-1} (c_1) al costo de clasificar incorrectamente un *non-churner (churner)*. Así mismo se define $\theta = (b_1 + c_1)/(b_{-1} + c_{-1})$ como el **Ratio Costo Beneficio** para simplificar notación. Tanto el beneficio esperado de realizar la campaña, como el umbral óptimo dependerán de este ratio de costos y beneficios.

Con la ayuda de la notación recién presentada, es posible construir la siguiente matriz, conocida como la **Matriz de Confusión**:

		Clasificado como	
		Clase -1	Clase 1
Pertenece a	Clase -1	**True Negative (TN)** $[c(-1\|-1) = b_{-1}]$	**False Positive (FP)** $[c(1\|-1) = c_{-1}]$
	Clase 1	**False Negative (FN)** $[c(-1\|1) = c_1]$	**True Positive (TP)** $[c(1\|1) = b_1]$

Tabla 1: Matriz de confusión para un problema de clasificación binaria

Una medida usada frecuentemente en la literatura para evaluar el desempeño de un clasificador es el AUC, que corresponde al área bajo la curva de ROC. La curva de ROC (*Receiver Operating Characteristic*, es una representación del desempeño del clasificador en la medida que cambia el valor umbral t, que marca el límite para determinar si una observación pertenece a la clase positiva o negativa dado el score obtenido por el clasificador para esa observación. Dicho de otro modo, la curva de ROC corresponde a un gráfico que muestra la **Sensibilidad** vs $1 -$ **Especificidad**, es decir $F_{-1}(t)$ como función de $F_1(t)$, en donde:

$$\text{Sensibilidad} = F_{-1}(t),$$
$$\text{Especificidad} = 1 - F_1(t),$$
$$\text{AUC} = \int F_{-1}(s) f_1(s) \mathrm{d}s.$$

La sensibilidad indica la tasa de todos los positivos que el modelo es capaz de reconocer como positivos, mientras la especificidad es la tasa de todos los negativos que el modelo es capaz de reconocer como negativos.

En términos simples, el AUC de un método de clasificación es la probabilidad de que una observación positiva elegida de forma aleatoria sea puntuada más alta que una clasificación negativa escogida al azar [10]. Lo que implica que a mayor AUC, mejor capacidad predictiva.

Al realizar una campaña de retención, una fracción η de los clientes más propensos a la fuga es contactada, incurriendo en un costo de f por persona, para ofrecerles un incentivo con un costo monetario d para la empresa. Entre este conjunto de clientes, habrá una fracción que no tiene intenciones de fugarse. Se asume entonces que éstos aceptan el incentivo y no se fugan.

Para aquellos que sí tienen intenciones de fugarse, se estima que una fracción γ acepta la oferta, lo que resulta en una ganancia directa en CLV (Ecuación (1)), mientras que una fracción $1-\gamma$ efectivamente deja la compañía a pesar del incentivo. Los clientes no contactados por la campaña de retención mantendrán su decisión, ya sea fugarse o permanecer en la empresa según corresponda. Este proceso se puede ver gráficamente en la Figura 1.

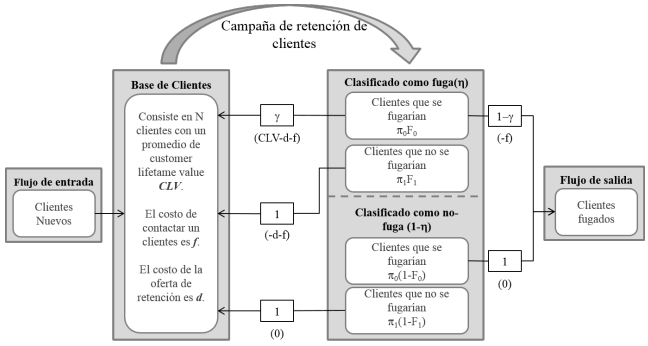

Figura 1: Esquema de evaluación de desempeño de una campaña de retención de clientes. (Fuente: [30])

El proceso descrito en la Figura 1, se sintetiza en la siguiente ecuación para el *Profit* [22]:

$$\text{Profit } = N\eta\left[(\gamma CLV + d(1-\gamma))\,\pi_{-1}\lambda - d - f\right] - A, \tag{6}$$

donde η es la fracción de clientes contactados, CLV es el *Customer Lifetime Value* (ver ecuación (1)), d es el costo del incentivo, f es el costo de contactar al cliente, y A son los costos administrativos fijos para la campaña. El coeficiente de *lift* λ es la fracción de clientes que se fugarían en la fracción η de clientes dividido por la tasa base de fuga para todos los clientes (a saber, π_{-1} [28]). Finalmente, γ se interpreta como la probabilidad de que un cliente que tenía pensado fugarse acepte el incentivo y no se vaya de la empresa. Para todo efecto, CLV, A, f y d son positivos, y $CLV > d$ para que esto tenga sentido en primera instancia. Es importante notar que claramente η depende de la selección del valor umbral t, lo que permite a la empresa tener el control sobre qué fracción de clientes contactar para ofrecer el incentivo de retención. Al ajustar la matriz de confusión vista en la Tabla 1, se obtiene lo siguiente:

		Clasificado como	
		Clase -1	Clase 1
Pertenece a	Clase -1	$\pi_{-1}F_{-1}(t)\,N$ $[c(-1\|-1)=b_{-1}]$	$\pi_{-1}(1-F_{-1}(t))\,N$ $[c(1\|-1)=c_{-1}]$
	Clase 1	$\pi_1F_1(t)\,N$ $[c(-1\|1)=c_1]$	$\pi_1(1-F_1(t))\,N$ $[c(1\|1)=b_1]$

Tabla 2: Matriz de confusión a nivel agregado.

Se estudiará el *Profit* promedio en vez del *Profit* total. Además se puede descartar A puesto que es un costo fijo que no depende del clasificador. Dicho esto, se define el **Profit medio generado por un clasificador para fuga de clientes** como la siguiente expresión:

$$P_{\mathcal{C}}(t;\gamma,CLV,\delta,\phi)=CLV\,(\gamma(1-\delta)-\phi)\,\pi_{-1}F_{-1}(t)-CLV(\delta+\phi)\cdot\pi_1F_1(t). \tag{7}$$

donde $\delta=\frac{d}{CLV}$ y $\phi=fCLV$. Es posible notar que $b_{-1}=CLV\,(\gamma(1-\delta)-\phi)$, y $c_1=CLV(\delta+\phi)$. También es posible deducir las siguientes relaciones, identificando términos y viendo la estructura de la ecuación (7):

- $\eta(t)=\pi_{-1}F_{-1}(t)+\pi_1F_1(t)$
- $\lambda(t)=\frac{F_{-1}(t)}{\pi_{-1}F_{-1}(t)+\pi_1F_1(t)}$.

De lo anterior y teniendo en consideración un conjunto de entrenamiento $\mathcal{T}$ con un conjunto de atributos $\mathcal{F}$, se estudian tres métricas diferentes, descritas a continuación:

H- Measure

En Hand [15] se propone esta métrica como una alternativa al AUC. La diferencia principal entre este indicador y las MP que serán descritas posteriormente, es que la medida H sólo se enfoca en costos. El foco de esta medida no es maximizar beneficios, si no minimizar los costos esperados. La pérdida media de clasificación Q se define como:

$$Q_{\mathcal{C}}(t;c,b)=b\cdot\left[c\pi_{-1}\left(1-F_{-1}(t)\right)+(1-c)\pi_1F_1(t)\right], \tag{8}$$

donde $c=c_0/(c_{-1}+c_1)$ y $b=c_{-1}+c_1$. Calcular el valor de la pérdida mínima esperada requiere hacer supuestos sobre la densidad de probabilidad tanto de

b como de c. Si se asume independencia de estos valores; y definiendo $w(b, c)$ como la distribución conjunta de b y c, y $u(c)$ y $v(b)$ como las densidades de probabilidad marginal de c y b respectivamente; se obtiene la siguiente relación: $w(b, c) = u(c)v(b)$. Usando esto último, la pérdida mínima esperada L toma la siguiente forma:

$$\mathrm{L} = \mathrm{E}[b] \int_0^1 Q_{\mathcal{C}}(T(c); b, c) \cdot u(c)\mathrm{d}c. \tag{9}$$

Se asume que c sigue una distribución Beta con parámetros α y β, cuya forma funcional es:

$$u_{\alpha,\beta}(x) = \begin{cases} \frac{x^{\alpha-1} \cdot (1-x)^{\beta-1}}{B(1,\alpha,\beta)} & \text{si } x \in [0, 1], \\ 0 & \text{en caso contrario,} \end{cases} \tag{10}$$

donde α, $\beta \in \mathbb{R}$ y α, $\beta > 1$, y además:

$$B(x, \alpha, \beta) = \int_0^x t^{\alpha-1} \cdot (1-t)^{\beta-1}\mathrm{d}t. \tag{11}$$

Una vez definido esto, para llegar a una métrica final (*H-measure*), se normaliza lo obtenido (para obtener una métrica acotada) entre cero y uno:

$$\mathrm{H} = 1 - \frac{\int_0^1 Q_{\mathcal{C}}(T(c); b, c) \cdot u(c)\mathrm{d}c}{\pi_0 \int_0^{\pi_1} c \cdot u(c)\mathrm{d}c + \pi_1 \int_{\pi_1}^1 (1-c) \cdot u(c)\mathrm{d}c}. \tag{12}$$

Acá $u(c)$ es una abreviación de notación de $u_{\alpha,\beta}(c)$. El denominador corresponde a la pérdida ocasionada por el peor clasificador posible, que corresponde a una función de predicción aleatoria.

Maximum Profit

Si se asume que todos los parámetros de la ecuación (7) son conocidos, para un clasificador $\mathcal{C}$ es posible generar una medida determinista. Considerando el valor máximo de $P_{\mathcal{C}}$ sobre todos los umbrales t, se obtiene la siguiente medida de rendimiento [28]:

$$\mathrm{MPC} = \max_t P_{\mathcal{C}}(t; \gamma, CLV, \delta, \phi). \tag{13}$$

De esta manera es posible obtener la fracción de clientes $\bar{\eta}_{\mathrm{mpc}}$ que debe ser contactada para maximizar el beneficio generado por la campaña de retención:

$$\bar{\eta}_{\text{mpc}} = \pi_{-1}F_{-1}(T) + \pi_1 F_1(T), \tag{14}$$

en donde

$$T(\gamma) = \arg\max_t P_{\mathcal{C}}(t; \gamma, CLV, \delta, \phi). \tag{15}$$

Expected Maximum Profit

Para este caso particular, se modela γ, que corresponde a la probabilidad de que un cliente que se iba de la empresa acepte el incentivo y se quede, como una variable aleatoria distribuida como una función Beta, lo que conduce a la siguiente expresión:

$$\text{EMPC} = \int_\gamma P_{\mathcal{C}}(T(\gamma); \gamma, CLV, \delta, \phi) \cdot h(\gamma) \text{d}\gamma, \tag{16}$$

donde $T(\gamma)$ es el corte óptimo según la ecuación (15) y $h(\gamma)$ la densidad de probabilidad para γ. Los parámetros α y β, relacionados a la distribución Beta de γ fueron obtenidos de un trabajo previo de fuga de clientes, que puede ser revisado en detalle en [30]. De manera análoga al **MPC**, el porcentaje o fracción de clientes contactados en la campaña de retención sugerida por esta métrica es:

$$\bar{\eta}_{\text{empc}} = \int_\gamma \left[\pi_{-1}F_{-1}\left(T(\gamma)\right) + \pi_1 F_1(T\left(\gamma\right))\right] \cdot h(\gamma) \text{d}\gamma. \tag{17}$$

4.2. Algoritmo basal

Se construyen nuevos métodos de selección de atributos en base al algoritmo HOSVM [19]. El concepto fundamental que está detrás de éstos es eliminar aquellos atributos cuya extracción tenga menor impacto en métrica considerada, tomando en cuenta los costos y beneficios que un clasificador puede ocasionar (dependiendo de la medida de desempeño). Para lograr esto, se busca encontrar el mejor desempeño predictivo en un conjunto desconocido del conjunto de entrenamiento, utilizando las métricas MPC, EMPC, y H-Measure. Los métodos de selección de atributos propuestos toman los siguientes nombres: SVM$_{MPC}$, SVM$_{EMPC}$, y SVM$_H$; de acuerdo a qué métrica de desempeño es utilizada.

Dado que el problema de fuga de clientes presenta usualmente un alto desbalance de clases [2], primero se redefine el algoritmo de *Holdout SVM* para incorporar un paso donde se realizan técnicas de *resampling* para mitigar el efecto del desbalance. En este trabajo se proponen dos estrategias: *Undersampling* aleatorio, y una combinación de *Undersampling* aleatorio y *SMOTE Oversampling* para aumentar el tamaño de la clase minoritaria.

Undersampling consiste en eliminar de manera aleatoria observaciones de la clase mayoritaria, para equilibrar la distribución de clases. Análogamente al *Undersampling*, el *Oversampling* tiene como objetivo inducir un balance entre las clases del conjunto de entrenamiento generando ejemplos artificiales a partir de la clase minoritaria. Una manera práctica de realizar esta técnica consiste en crear observaciones sintéticas interpolando los ejemplos de un subconjunto de observaciones de dicha clase, lo que se conoce como ***SMOTE Oversampling*** [7].

El propósito final del algoritmo es encontrar un subconjunto $\mathcal{K}$ ($\mathcal{K} \subseteq \mathcal{F}$) de atributos, de tal manera que el desempeño predictivo del clasificador sea maximizado. Se considera un conjunto de entrenamiento $\mathcal{T}$, el cual es particionado en un subconjunto de entrenamiento $\mathcal{TR}$ y otro de validación $\mathcal{V}$. En $\mathcal{TR}$ se aplican las técnicas de *resampling* mencionadas, dando origen a un nuevo conjunto $\mathcal{TR}'$, en donde se construye el clasificador. La función de contribución de cada atributo se construye en $\mathcal{V}$ en base a las métricas MPC, EMPC, y H-Measure. Esquemáticamente, lo recién expuesto se presenta en el siguiente algoritmo:

El clasificador entrenado sobre el conjunto $\mathcal{TR}'$ en el paso 5 del algoritmo recién descrito corresponde al par ordenado $\Lambda = (\boldsymbol{\alpha}, b)$, y esta información se usa para calcular una función de pérdida en el conjunto de validación, a saber $\text{LOSS}^{(-j)}(\boldsymbol{\Lambda}, \mathcal{TR}', \mathcal{V})$. Se propone calcular las medidas MPC, EMPC, y H-Measure utilizando el subconjunto $\mathcal{V}$ cuando el atributo j es eliminado. El atributo cuya eliminación lleve a obtener el beneficio mayor (o menor costo en el caso de H-Measure) debe ser eliminado de la base. Para adaptar estas métricas, la versión propuesta del algoritmo solo difiere de las versiones originales de MPC, EMPC y H-Measure en el cálculo de los *scores* para cada observación, mientras que ni los costos y beneficios de una solución dada ni la definición de γ se ven afectados. Se define $s_k^{(-j)}$ como el *score* de la observación $k \in \mathcal{V}$ cuando el atributo j es eliminado, y toma la siguiente forma:

[2]Incluso cuando en muchos sectores de la industria de servicios se encuentran tasas de fuga anuales en torno al 20 % y 50 % [22], cuando se usan modelos de predicción de fuga mensual, las tasa de fuga usualmente se mantiene bajo el 5 % [28]

Algorithm 3 Algoritmo de *Holdout* para *Backward Feature Elimination* usando SMOTE

Input: El conjunto original de atributos $\mathcal{F}$

Output: Un vector ordenado de atributos $\mathcal{F}^\dagger$

1. $\mathcal{F}^\dagger \leftarrow \emptyset$
2. **repetir**
3. $(\mathcal{TR}, \mathcal{V}) \leftarrow$ Holdout usando $\mathcal{T}$
4. $\mathcal{TR}' \leftarrow$ Resampling($\mathcal{TR}$)
5. $\mathbf{\Lambda} \leftarrow$ Entrenamiento SVM usando $\mathcal{TR}'$
6. $\mathcal{I} \leftarrow \text{argmin}_{\mathcal{I}} \sum_{j \in \mathcal{I}} \text{LOSS}^{(-j)}(\mathbf{\Lambda}, \mathcal{TR}', \mathcal{V}),\ \mathcal{I} \subset \mathcal{F}$
7. $\mathcal{F} \leftarrow \mathcal{F} \setminus \mathcal{I}$
8. $\mathcal{F}^\dagger \leftarrow (\mathcal{F}^\dagger, \mathcal{I})$
9. **hasta** $\mathcal{F} = \emptyset$

$$s_k^{(-j)}(\mathbf{\Lambda}, \mathcal{TR}', \mathcal{V}) = \sum_{i \in \mathcal{TR}'} \alpha_i y_i K(\mathbf{x}_i^{(-j)}, \mathbf{x}_k^{v(-j)}) + b, \tag{18}$$

donde $\mathbf{x}_i^{(-j)}$ corresponde a una observación del conjunto de entrenamiento originado por el *resampling* cuando el atributo j es eliminado y $\mathbf{x}_k^{v(-j)}$ a un objeto de validación k con la variable j eliminada. Análogo a RFE-SVM, el vector α se asume que es igual a la solución obtenida en el paso 5 del Algoritmo 3 para reducir la complejidad computacional.

Luego de lo anterior, se proponen las siguientes métricas de desempeño basadas en utilidades para el paso 6 del algoritmo 3, donde la única diferencia respecto a la definición original de las métricas es la inclusión de la fórmula para $s^{(-j)}$:

- **H measure**:

$$\text{LOSS}^{(-j)}(\mathbf{\Lambda}, \mathcal{TR}', \mathcal{V}) = \text{H}(s^{(-j)}) \tag{19}$$

- **Maximum Profit (MPC)**:

$$\text{LOSS}^{(-j)}(\mathbf{\Lambda}, \mathcal{TR}', \mathcal{V}) = \text{MPC}(s^{(-j)}) \tag{20}$$

- **Expected Maximum Profit (EMPC)**:

$$\text{LOSS}^{(-j)}(\mathbf{\Lambda}, \mathcal{TR}', \mathcal{V}) = \text{EMPC}(s^{(-j)}) \tag{21}$$

Utilizando estas funciones en el paso 6 del algoritmo 3, se generan tres variantes del enfoque propuesto: HOSVM_H, HOSVM_{MPC}, y HOSVM_{EMPC}; cuyos desempeños serán evaluados junto a las otras metodologías de selección de atributos en la siguiente sección.

Finalmente, en el paso 6 el algoritmo determina un conjunto de atributos a ser eliminado ($\mathcal{I}$). Si bien es posible eliminar un sólo elemento en cada iteración, esto es ineficiente dado que en general existe un número considerable de atributos irrelevantes. Por otro lado, remover muchos atributos a la vez aumenta el riesgo de eliminar atributos relevantes [14].

5. Resultados

En esta sección se analizan los resultados de tres problemas de predicción de fuga de clientes mediante diversas técnicas de selección de atributos. Primero se describen las bases de datos utilizadas y el diseño experimental, y luego se presentan los resultados obtenidos.

5.1. Descripción de las Bases de Datos y Diseño Experimental

Las tres bases de datos utilizadas de fuga de clientes se describen a continuación:

- **UCI-Telecom**: Esta base de datos de clientes fue extraída desde el repositorio UCI [1] y contiene la información de 5,000 clientes de una compañía de telecomunicaciones, descritos por 20 atributos.
- **Operator 1**: Esta base de clientes de telecomunicaciones fue originalmente estudiada por [21], y contiene 47,761 clientes descritos por 47 variables. Esta base de datos fue además utilizada en Verbeke et al. [28] bajo el nombre de Operator 1 (O1).
- **Cell2Cell**: Esta base de datos fue propuesta en [9] como caso de estudio, contiene información de 20,406 clientes descritos por 73 variables. Esta base de datos fue además utilizada en Verbeke et al. [28] bajo el nombre de D2.

Para los diferentes enfoques de SVM se usó LIBSVM [6] para Matlab. La tabla 3 resume la información relevante de cada base de datos:

Base de Datos	#variables	#obs(min.,may.)	tasa de churn
UCI-Telecom	20	(707;4,293)	16.5 %
Operator 1	47	(1,761;46,000)	3.8 %
Cell2Cell	73	(406;20,000)	2.0 %

Tabla 3: Número de variables, número de observaciones de cada clase y tasa de *churn* para cada una de las 3 bases.

El diseño experimental consiste en la metodología KDD [11], que es una técnica con éxito comprobado en *business analytics*, tanto en predicción de fuga como en *credit scoring* [4].

- **Recopilación y consolidación de datos:** El primer paso consiste en identificar las fuentes relevantes de datos y consolidarlas en un repositorio único creado para la construcción de los modelos de clasificación. Este paso fue directo, puesto que todas las bases de datos ya estaban consolidadas.
- **Pre-procesamiento de datos:** El siguiente paso, es la eliminación de observaciones con valores faltantes, y transformación de datos. Estos pasos también fueron directos, puesto que las bases ya estaban pre-procesadas para el análisis.
- **Minado de datos:** Se siguió el siguiente procedimiento para realizar el ranking de atributos y la selección de hiper-parámetros del modelo: Para cada una de las bases, los conjuntos de entrenamiento y testeo fueron generados mediante Validación Cruzada en 10 particiones, que es una técnica común en la predicción de fuga de clientes [28], luego se realiza tanto el ranking de atributos como la clasificación de las observaciones en el conjunto de entrenamiento. El desempeño de la clasificación final es calculado promediando los resultados de la clasificación de los diferentes conjuntos de testeo, utilizando tanto métricas usuales como indicadores de rendimiento basados en utilidades. La selección del modelo fue realizada a través de una búsqueda en grilla y evaluando los siguientes valores para los parámetros C y σ (solo para Kernel RBF): $C \in \{2^{-7}, ..., 2^{7}\}$ y $\sigma \in \{2^{-7}, ..., 2^{7}\}$.
- **Evaluación de los resultados:** El desempeño de todos los métodos fue estudiado para diferentes valores de los hiper-parámetros y comparados con diferentes indicadores, para ver cuál es el impacto en el clasificador final.

5.2. Presentación de Resultados

En esta sección se presenta un resumen de los resultados para facilitar la evaluación de la mejor instancia de cada uno de los distintos enfoques. En las tablas 4, 5, y 6 se resume el desempeño promedio entre diferentes conjuntos de atributos para cada método en las bases de datos Telecom1, Cell2Cell, y Operator1 respectivamente. Para este efecto, consideramos las siguientes métricas: AUC, EMPC, MPC y H-measure. Las métricas EMPC y MPC están medidas en Euros por cliente. El mejor desempeño

entre todos los métodos se destaca en negrita. Adicionalmente, se destaca con un asterisco cuando el desempeño es significativamente más bajo que el mejor método a un 10 % de nivel de significancia estadística, con doble asterisco a un 5 %, y con 3 asteriscos a un 1 %. Un test t es usado para hacer comparaciones entre las medias de pares de enfoques y el mejor método para cada base de datos. Los resultados son mostrados para la mejor estrategia de *resampling*, que corresponde a *undersampling* aleatorio en cada base de datos.

	Fisher	RFE	HOSVM$_{EMPC}$	HOSVM$_{H}$	HOSVM$_{MPC}$
AUC	62.4**	63.5*	64.5	64.4	**64.6**
EMPC	2.21**	2.45	2.58	2.55**	**2.61**
MPC	2.06**	2.36	2.51	2.49**	**2.55**
H	0.064**	0.089	0.092	0.085	**0.094**

Tabla 4: Desempeño medio para todos los métodos e indicadores, para la base Telecom1

	Fisher	RFE	HOSVM$_{EMPC}$	HOSVM$_{H}$	HOSVM$_{MPC}$
AUC	64.81***	94.09	94.09	94.13	**94.13**
EMPC	0.224***	**0.879**	0.860	0.860	0.859
MPC	0.223***	**0.876**	0.859	0.860	0.859
H	0.097***	**0.462**	0.429	0.381	0.385

Tabla 5: Desempeño medio para todos los métodos e indicadores, para la base Cell2Cell

	Fisher	RFE	HOSVM$_{EMPC}$	HOSVM$_{H}$	HOSVM$_{MPC}$
AUC	49.65**	54.35	54.49	**55.18**	54.47
EMPC	0.006	0.006	**0.008**	0.007	0.007
MPC	0.005	0.006	0.007	**0.007**	0.006
H	0.001***	0.001	**0.002**	0.001	0.002

Tabla 6: Desempeño medio para todos los métodos e indicadores, para la base Operator1

De la Tabla 4, para la base Telecom1, observamos que el método propuesto usando la métrica MPC para eliminación de atributos y construcción del clasificador tiene mejor desempeño para todos los indicadores considerados. El método superó a la selección vía *Fisher Score* con un nivel de significancia del 5 % en todos los indicadores y al método RFE-SVM a un nivel de 10 % de significancia en AUC. Mientras que el

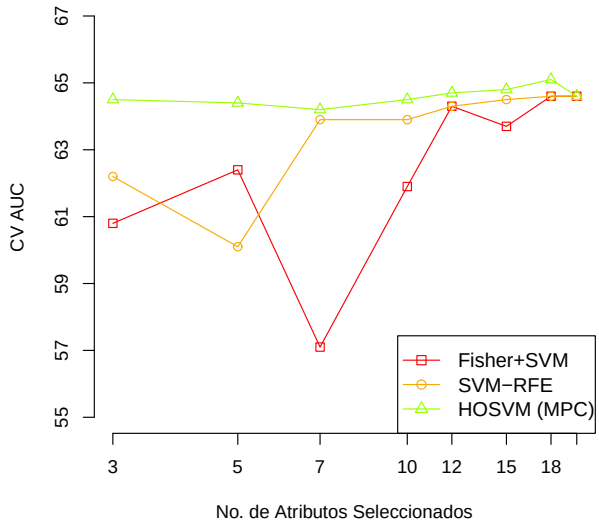

Figura 2: AUC versus el número de variables para diferentes enfoques de selección de atributos. Base Telecom1.

método HOSVM con EMPC nunca es significativamente peor que el mejor de los métodos para todas las métricas consideradas.

Para la Tabla 5, que corresponde a la base Cell2Cell, el método propuesto HOSVM con MPC tiene mejor AUC en general, mientras que RFE usando SVM tiene mejor desempeño para el resto de las métricas. Nuevamente *Fisher Score* fue superado por todos los métodos con un nivel de significancia del 1 % en todas las métricas, mientras que los otros métodos nunca se comportaron significativamente peor que el mejor método.

Adicionalmente, para la Tabla 6, que corresponde a la base Operator1, el método propuesto de HOSVM basado en EMPC tiene mejor desempeño para las métricas EMPC y H-measure, mientras que HOSVM basado en H-measure alcanza mejores resultados tanto en AUC como en MPC. Una vez más, *Fisher Score* es superado en AUC (significancia del 5 %) y en el caso de H-measure alcanza un 1 % de significancia, mientras que los otros métodos nunca están significativamente por debajo del mejor métodos para todas las métricas.

A modo ilustrativo, es posible analizar gráficamente el comportamiento de la selección de atributos para diferentes subconjuntos de variables. El gráfico 2 presenta el desempeño predictivo para un número creciente de atributos elegidos para la base Telecom1. Para cada subconjunto de atributos, la media de AUC para los métodos *Fisher Score*, RFE-SVM y el mejor método propuesto (HOSVM basado en MPC para Telecom) se presenta en el gráfico 2.

En la Figura 2 se puede observar que el método de selección de atributos propuesto

(HOSVM basado en MPC), alcanza el mejor desempeño AUC 0.648 con 18 atributos, y luego suavemente disminuye su desempeño. A diferencia de *Fisher Score* o bien RFE-SVM, que disminuyen abruptamente su desempeño en la medida que se van removiendo atributos.

6. Conclusiones

En este trabajo se propone un enfoque de eliminación de atributos recursivo, y empotrado en la construcción del modelo de clasificación usando SVM. El método propuesto estudia tres medidas de desempeño diferentes para la fuga de clientes: la *H-measure*, el *Maximum Profit Measure for Customer Churn (MPC)*, y el *Expected Maximum Profit Measure for Customer Churn (EMPC)*. Mientras que *H-measure* provee una estructura capaz de considerar de manera explícita los costos de clasificar de forma incorrecta como medida de la capacidad predictiva de un modelo, como se puede ver en [15], la medida MPC [28] y el EMPC [30] van un paso adicional e incorporan los potenciales beneficios de una campaña de retención realizada para evitar una eventual fuga de clientes, lo que genera una medida muy robusta y con una visión de negocios más recabada para evaluar el desempeño de un modelo de clasificación. La principal diferencia entre MPC y EMPC, es que este último considera la decisión de un cliente candidato a fuga como una variable aleatoria, y luego calcula el valor esperado del beneficio de una campaña de retención realizada orientada a los clientes que fueron señalados por el clasificador.

A diferencia de la literatura disponible en esta área, que se enfoca en seleccionar el mejor modelo entre varios métodos de clasificación, el objetivo de este trabajo es proporcionar un sustento teórico que permita la correcta selección de parámetros y atributos en la construcción del clasificador, basándonos en la herramienta *Support Vector Machines*. El enfoque presenta las siguientes ventajas:

- El método permite la incorporación explícita de los costos y beneficios obtenidos al clasificar en problemas de predicción de fuga de clientes, llevando a un proceso de selección de atributos especialmente diseñado para este problema en particular.
- El enfoque alcanza mejores resultados que otras técnicas de selección de atributos en problemas de predicción de fuga, considerando tanto métricas usuales de desempeño (como por ejemplo AUC), e indicadores basados en ganancias.
- La estrategia en sí misma es muy flexible y permite elegir diferentes funciones de Kernel para selección de atributos en entornos no lineales, y clasificación usando SVM. Incluso el enfoque permite ser extendido a otras herramientas de clasificación, no necesariamente SVM.

Existen muchas oportunidades de trabajo futuro, a modo de ejemplo se puede considerar las siguientes:

- El proceso de selección de atributos puede ser extendido a otras aplicaciones de *business analytics*, como por ejemplo *credit scoring* [4, 24]. El EMPC puede

adaptarse para incorporar los costos y los beneficios de aceptar o rechazar a los solicitantes de créditos, la regresión logística se puede establecer como el clasificador de referencia, puesto que es el método de clasificación más común para esta tarea debido a razones regulatorias [24].

- También es posible incorporar el costo de la adquisición de las variables en el modelo, enriqueciendo de esta manera el proceso de selección de atributos. Es posible ver este enfoque en [18], en donde el costo de los atributos se considera explícitamente en el modelo a través de variables binarias y una restricción presupuestaria explícita.

Agradecimientos: Este trabajo fue financiado por el Instituto Sistemas Complejos de Ingeniería (ICM: P-05-004-F, CONICYT: FB016) y Fondecyt (1140831).

Referencias

[1] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.

[2] B. Baesens. *Analytics in a Big Data World*. John Wiley and Sons, 2014.

[3] R.C. Blattberg, B.D. Kim, and S.A. Neslin. Database marketing: Analyzing and managing customers. 2008.

[4] C. Bravo, S. Maldonado, and R. Weber. Methodologies for granting and managing loans for micro-entrepreneurs: New developments and practical experiences. *European Journal of Operational Research*, 227(2):358–366, 2013.

[5] J. Burez and D. Van den Poel. Handling class imbalance in customer churn prediction. *Expert Systems with Applications*, 36(3):4626–4636, 2009.

[6] C.C. Chang and C.J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[7] N. Chawla. *Data mining for imbalanced datasets: An overview*. Springer, Berlin, 2010.

[8] P. Datta, B. Masand, D.R. Mani, and B. Li. Automated cellular modeling and prediction on a large scale. *Artificial Intelligence Review*, 14:485–502, 2000.

[9] Center for Customer Relationship Management Duke University, February 2014.

[10] T. Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27:861–874, 2006.

[11] U. Fayyad. Data mining and knowledge discovery- making sense out of data. *IEEE Expert-Intelligent Systems and Their Aplications*, 11,:20–25, 1996.

[12] J.H. Fleming and J. Asplund. *Human Sigma: Managing The Employee-Customer Encounter*. Gallup Press, New York, 2007.

[13] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh. *Feature extraction, foundations and applications*. Springer, Berlin, 2006.

[14] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002.

[15] D.J. Hand. Measuring classifier performance: a coherent alternative to the area under the roc curve. *Machine Learning*, 77(1):103–123, 2009.

[16] S.Y. Hung, D.C. Yen, and H.Y. Wang. Applying data mining to telecom churn management. *Expert Systems with Applications*, 31:515–524, 2006.

[17] A. Lemmens and C. Croux. Bagging and boosting classification trees to predict churn. *Journal of Marketing Research*, 43(2):276–286, 2006.

[18] S. Maldonado, J. Pérez, M. Labbé, and R. Weber. Feature selection for support vector machines via mixed integer linear programming. *Information Sciences*, 279:163–175, 2014.

[19] S. Maldonado and R. Weber. A wrapper method for feature selection using support vector machines. *Information Sciences*, 179:2208–2217, 2009.

[20] S. Maldonado, R. Weber, and J. Basak. Kernel-penalized SVM for feature selection. *Information Sciences*, 181(1):115–128, 2011.

[21] M. Mozer, R. Wolniewicz, D. Grimes, E. Johnson, and H. Kaushansky. Predicting subscriber dissatisfaction and improving retention in the wireless telecommunications industry. *IEEE Transactions on Neural Networks*, 11(3):690–696, 2000.

[22] S.A. Neslin, S. Gupta, W.A. Kamakura, J. Lu, and C.H. Mason. Defection detection: Measuring and understanding the predictive accuracy of customer churn models. *Journal of Marketing Research*, 43(2):204–211, 2006.

[23] B. Schlkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, USA., 2002.

[24] L.C. Thomas, J.N. Crook, and D.B. Edelman. *Credit Scoring and its Applications*. SIAM, 2002.

[25] D. Van den Poel and B. Larivière. Customer attrition analysis for financial services using proportional hazard models. *European Journal of Operational Research*, 157(1):196–217, 2004.

[26] J. Van Hulse, T.M. Khoshgoftaar, A. Napolitano, and R. Wald. Feature selection with high-dimensional imbalanced data. In *Proceedings of the IEEE International Conference on Data Mining Workshops*, pages 507–514, 2009.

[27] V. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, 1998.

[28] W. Verbeke, K. Dejaeger, D. Martens, J. Hur, and B. Baesens. New insights into churn prediction in the telecommunication sector: A profit driven data mining approach. *European Journal of Operational Research*, 218(1):211–229, 2012.

[29] W. Verbeke, D. Martens, C. Mues, and B. Baesens. Building comprehensible customer churn prediction models with advanced rule induction techniques. *Expert Systems with Applications*, 38:2354–2364, 2011.

[30] T. Verbraken, W. Verbeke, and B. Baesens. A novel profit maximizing metric for measuring classification performance of customer churn prediction models. *IEEE Transactions on Knowledge and Data Engineering*, 25(5):961 – 973, 2012.

[31] C.P. Wei and I.T. Chiu. Turning telecommunications call details to churn prediction: a data mining approach. *Expert Systems with Applications*, 23:103–112, 2002.

[32] X. Wu, V. Kumar, J. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. McLachlan, A. Ng, B. Liu, P. Yu, Z. Zhou, M. Steinbach, D. Hand, and D. Steinberg. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14:1–37, 2008.