Predicción de fuga de clientes y selección de atributos

Camilo Escobar, Diego Garrido
Departamento de Ingeniería industrial

***Resumen*—Diversas técnicas de *Machine Learning* han demostrado buenos resultados en la predicción de fuga de clientes, careciendo sin embargo, de capacidad explicativa, cuestión muy importante en *Business Analytics*. En el presente trabajo se utilizó la técnica *Recursive Feature Elimination* (RFE) en *Support Vector Machines* (SVM) sobre la *database* de una compañia de telecomunicaciones para el caso de un kernel lineal y uno gaussiano. En ambos se logró reducir considerablemente la cantidad de *features* utilizadas (10 - 35 %) teniéndose una mínima reducción en la capacidad de predicción (0.5 - 3 % de pérdida en *test accuracy* general), demostrando así que RFE-SVM es muy útil en la reducción de características con mínima pérdida de precisión.**

***Index Terms*—Support Vector Machine, Recursive Feature Elimination, Clasificación, Regresión Logística**

I. Descripción del Problema

La intención del trabajo aquí presentado es desarrollar metodologías que **no solo logren predecir**, sino que también permitan **conocer cuales de las características utilizadas tienen mayor impacto** en la predicción con el fin de que las empresas puedan generar programas focalizados para evitar la fuga de clientes.

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 $churns$, 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. Un incremento en la proporción de clientes fieles, los cuales generan 1,7 veces más ingreso que los otros clientes [1].

2. 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 [1].

3. 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* [2].

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. En este trabajo se estudia el *churn* como un fenómeno voluntario.

Respecto a la selección de atributos se ha demostrado que el desempeño de un clasificador puede ser mejorado enfocándose en las características más relevantes usadas para la construcción de este. 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*, causado por la *"Maldición de la dimensionalidad"* [3].

2. Dicha selección permite una mejor interpretación del clasificador, porque al identificar los atributos más relevantes de la fuga de los clientes se pueden detectar fallas en el servicio ofrecido, 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 [4].

II. Métodos existentes

Los métodos más comunes usados en clasificación son redes neuronales, support vector machines, regresión logística, entre otros, teniendo diversas ventajas y desventajas.

Para el caso de redes neuronales se tiene:

Ventajas:
- Muy útil para la predicción en sí, al tener -en general- una **gran precisión**.

Desventajas:
- **Muy baja capacidad explicativa**, lo que se vuelve un problema importantísimo en el *Business Analytics* al no poder tomar acción de los resultados obtenidos.
- La heterogeneidad en el comportamiento de los clientes en general es alta, y el hecho de que las redes neuronales sean **susceptibles al *overfitting*** es un problema importante.
- Tienen **tiempos largos de entrenamiento** y en general en las telecomunicaciones existe gran cantidad de datos,

siendo por lo tanto imposibles de analizar en un tiempo razonable.

Por otro lado para el caso de support vector machines se tiene:

Ventajas:

- Tienen **tiempos bajos de entrenamiento** v/s redes neuronales, pudiendo obtenerse resultados bastantes buenos en tiempos más razonables.
- **Precisión media a alta**, pudiendo entregar en muchos casos resultados muy buenos.
- **Robusto frente a outliers**, lo que es muy útil por la heterogeneidad de los datos.

Desventajas:

- **Baja capacidad explicativa**, lo que es -tal como se ha dicho- un gran problema en el *Business Analytics* al no poder tomarse acción a partir de los modelos.

Y por último para regresión logística:

Ventajas:

- **Tiempo de entrenamiento bajo**, siendo muy útil esta técnica como una cota inferior de los resultados que se obtendrán con las otras.
- **Alta capacidad explicativa**, debido a los pesos que se obtienen para las variables, especificando cómo afecta cada una en la clase a predecir.

Desventajas:

- En general se tiene **precisión media a alta**, siendo en la mayoría de los casos menor a SVM o redes neuronales.
- **Débil frente a outliers** por la manera en que se configura la técnica, lo que podría generar modelos débiles en cierto tipo de clientes.

De los 3 métodos en general existe un *trade-off* marcado entre capacidad explicativa y capacidad de precisión. En vista de que redes neuronales tiene una capacidad explicativa mínima es que se elimina de las posibles metodologías a usar, ya que el principal objetivo planteado es llegar a una predicción "buena" con un nivel explicativo considerable que permita a la compañía utilizar algún método para evitar la fuga de clientes.

III. Metodología propuesta

Dado que el problema de fuga de clientes presenta usualmente un alto desbalance de clases se introdujo una técnica de *resampling* para mitigar posibles problemas.

En este trabajo se propone una técnica de *oversampling* que tiene como objetivo inducir un balance entre las clases del conjunto de entrenamiento generando ejemplos artificiales a partir de la clase minoritaria, lo que se conoce como **SMOTE** [5].

Synthetic Minority Oversampling Technique: SMOTE

1. El algoritmo selecciona una muestra del conjunto minoritario.
2. Se calcula la diferencia entre el vector de característica (muestra) en consideración y su vecino más cercano.
3. Se multiplica la diferencia del punto anterior por un número aleatorio entre 0 y 1, y se añade al vector de características considerado.
4. Se repite el algoritmo hasta que las clases queden balanceadas.

Los ejemplos sintéticos hacen que el clasificador cree regiones de decisión más grandes y menos específicas, en lugar de regiones más pequeñas y más específicas, como es causado típicamente por el sobre-muestreo con replicación. Produciendo así que el clasificador aprenda regiones más generales para la clase minoritaria en lugar de ser subordinada por la clase mayoritaria.

Respecto a la selección de atributos la metodología propuesta consiste en eliminación recursiva de características (**RFE** [6-7] en inglés). El objetivo de este método es encontrar subconjuntos de tamaño r entre m características (con $r < m$) eliminando aquellos atributos cuya extracción contribuye a alcanzar el mayor margen de separación entre clases.

Formulación dual del problema original de SVM:

$$max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \qquad (1)$$

$$W^2(\alpha) = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

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

Algoritmo *Recursive Feature Elimination, SVM*

1. Se inicializa con m características.
2. Se resuelve (1).
3. Se elimina el atributo p con menor valor de $|W^2(\alpha) - W^2_{-p}(\alpha)|$.
4. Se vuelve a 2 con $m - 1$ características. Hasta que ya no quedan características por eliminar obteniéndose un ranking.

IV. Resultados

Como base de datos se utilizó *UCI-Telecom* [8] consistente de 3333 observaciones con 20 variables, siendo una correspondiente al vector y_i que toma dos valores: *churn* o *no churn*.
De las variables se decidió eliminar 3 ya que no contribuían mayormente a los resultados. Se tiene finalmente una base con 3333 observaciones con 17 *features* y 1 variable correspondiente a la clase a predecir.

El conjunto total de datos contiene un 85 % para la clase *no churn* y 15 % para la clase *churn*. El entrenamiento fue realizado dividiendo la base de datos en un 80 % para datos de entrenamiento y un 20 % para datos de prueba.

IV-A. Caso desbalanceado

Como un primer acercamiento se realizó regresión logística (Regl) y SVM para kernel lineal (KL) y gaussiano (KG). Para los últimos dos casos se usó $C = 1; \gamma = 0{,}01$ según corresponda.

Tabla I: Matrices de confusión sin balance de clases

		RegL		KL		KG	
		NC	C	NC	C	NC	C
True	NC	0.982	0.018	**1.0**	0.0	0.974	0.026
	C	0.808	0.192	1.0	0.0	0.394	**0.606**
		Predicted					
Test Accuracy		84 %		86 %		**92 %**	

Al comparar la regresión logística con el kernel lineal, se tiene que aunque el segundo tiene mejor *test accuracy* general, su predicción de la clase *churn* es nula. Siendo por lo tanto preferible la primera técnica, ya que además tiene un poder explicativo mucho mayor al tener los pesos de cada *feature*.

Por otro lado, al observar el caso gaussiano se obtienen mejores resultados: un mayor *test accuracy* general; consierablemente mejor predicción de la clase *churn* (40 % superior v/s regresión logística), siendo por lo tanto esta la mejor técnica de las 3.

IV-B. Caso balanceado

Se realizó un balanceo de clases con el método SMOTE[1] en el conjunto de entrenamiento con el fin de mejorar los resultados anteriores.

Igual que en el caso anterior se realiza primero regresión logística en conjunto con SVM lineal y gaussiano con los mismos parámetros ($C = 1; \gamma = 0{,}01$).

[1]Usando librería Imblearn

Tabla II: Matrices de confusión con balance de clases

		RegL		KL		KG	
		NC	C	NC	C	NC	C
True	NC	0.754	0.246	0.752	0.248	**0.868**	0.132
	C	0.289	0.711	0.266	0.744	0.172	**0.828**
		Predicted					
Test Accuracy		75 %		75 %		**86 %**	

De los resultados se puede ver que el método de balanceo es bastante útil ya que aunque en general el *test accuracy* es menor, la predicción de la clase *churn* es mucho mayor (20 % aproximadamente), lo que es **muy importante** al ser la clase de mayor interés.

SVM-RFE con kernel lineal

Figura 1: *Test accuracy* por número de *features* usadas.

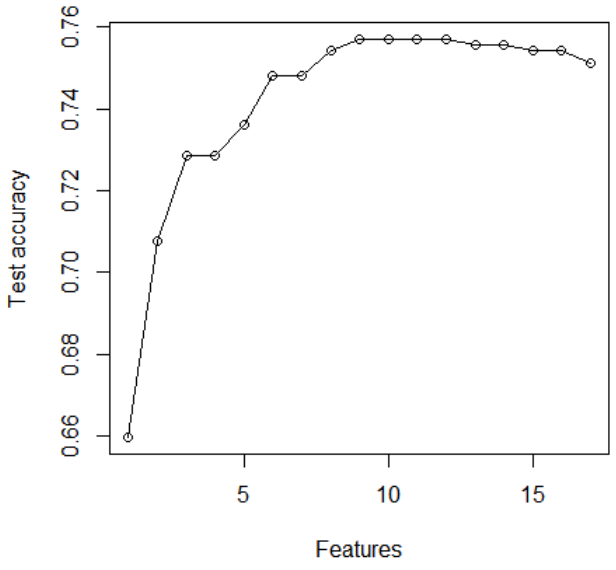
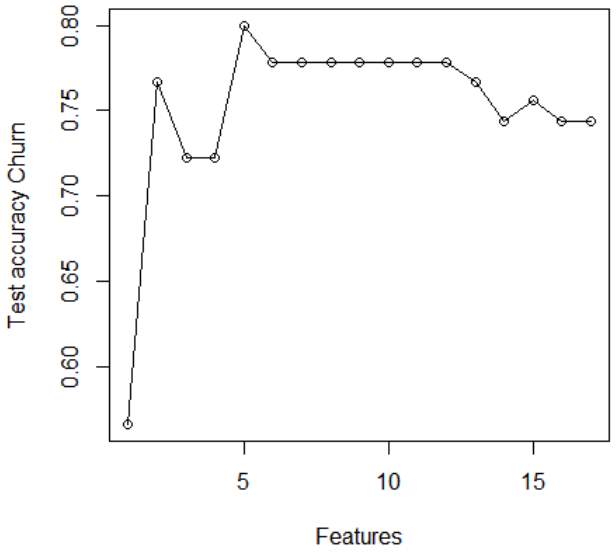


Figura 2: Test accuracy en clase *churn* por número de *features* usadas.



De la figura 1 se puede notar que el mayor *test accuracy* se obtiene con 9 *features*, siendo de 75 %, lo que sugiere que usando todas las variables existe un sobreajuste de más o menos un 1 %.

Por otro lado de la figura 2 se puede ver que el *test accuracy* sobre la clase *churn* es mayor con 14 *features* obteniéndose un 75 %. Es interesante notar también que a veces al aumentar la cantidad de atributos usados existe una menor predicción, lo que sugiere no solo efectos de *overfitting*, sino que de interacción entre variables.

SVM-RFE con kernel gaussiano

Figura 3: *Test accuracy* por número de *features* usadas.

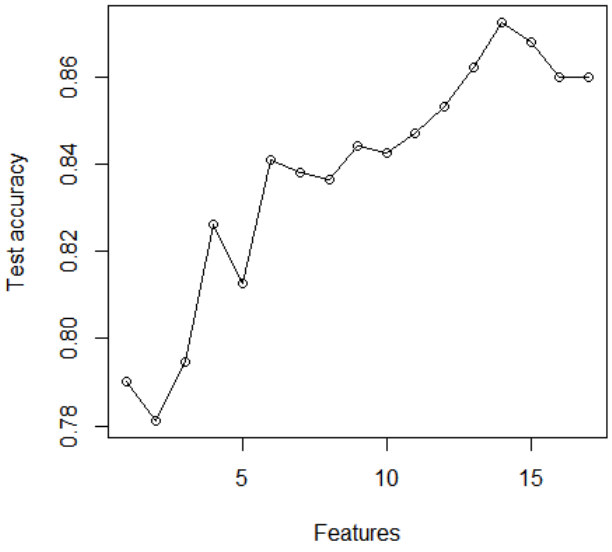
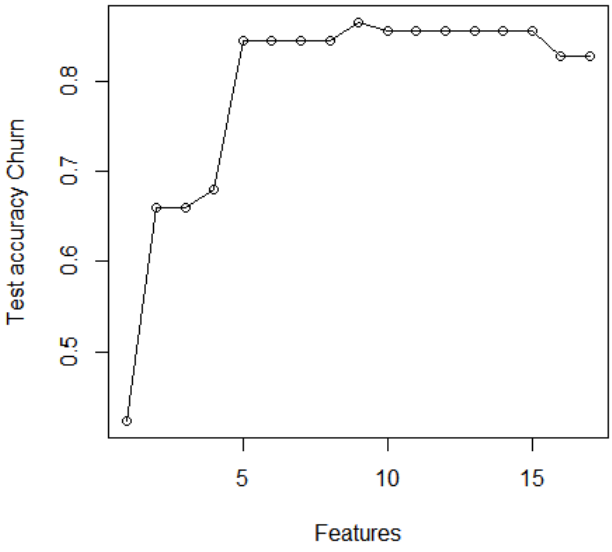


Figura 4: Test accuracy en clase *churn* por número de *features* usadas.



De la figura 3 se puede ver que el *test accuracy* obtiene su mejor resultado para 14 *features*, mientras que hasta 6 se obtiene una predicción considerablemente buena (2 % de pérdida respecto al mayor valor).

Por otro lado, al observar la figura 4, se puede ver que la predicción de la clase *churn*, para 9 *features* obtiene su mayor valor. Mientras que hasta 5 *features* no hay una gran pérdida de predicción (0.5 % menor respecto al mayor valor).

V. Conclusiones y trabajos futuros

En este trabajo se propone un enfoque de eliminación de atributos recursivo, y empotrado en la construcción del modelo de clasificación usando SVM con una técnica de balanceo de clases.
El enfoque presenta las siguientes ventajas:

- El balanceo de clases es muy útil para mejorar el % de clasificación de la clase minoritaria (20 % de mejora aproximadamente).
- El algoritmo RFE-SVM es muy útil para entender importancia de las *features* sin sacrificar mucha capacidad predictiva: para el caso de *churn* en kernel lineal hay reducción a 2 *features* con 3 % pérdida de predicción (80 a 77 %) y en kernel gaussiano hay reducción a 5 *features* con 0.5 % de pérdida (85.5 a 85 %)
- RFE-SVM en algunos casos logra mejorar levemente los resultados (1 - 2 % en general y 2 - 5 % en caso de *churn*)

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

- Extender algoritmo RFE-SVM con penalización asimétrica con el fin de predecir de mejor manera la clase importante (*churn*).
- Extender RFE-SVM a eliminar conjuntos de *features* más que a *features* por sí solas para entender de mejor manera la interacción entre variables.
- Utilizar métodos de validación cruzada con el fin de hacer más robusta la predicción.

REFERENCIAS

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

[2] 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.

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

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

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

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

[7] Maldonado Alarcón, S. (2007).Utilización de Support Vector Machines No Lineal y Selección de Atributos para Credit Scoring.

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