Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay

Detección del riesgo de abandono de llamadas en un callcenter

Reconocimiento de Patrones

Instituto de Ingeniería Eléctrica

Juan Pablo Chavat
Diciembre de 2016

Índice

1. Introducción

En la actualidad, los centros masivos de atención telefónica (*callcenters* en inglés) son muy populares. Estos centros permiten a variadas organizaciones prestar diversos servicios tales como soporte a usuarios, atención de denuncias, ejecución tramites, entre otros. En estas organizaciones trabajan una gran cantidad de personas cumpliendo la función de operadores telefónicos (también llamados agentes), que se encargan de atender las llamadas que entran al sistema o realizar las llamadas salientes. En los casos en los que se atienden llamadas entrantes, existen franjas horarias en las cuales la cantidad de agentes disponibles para atender las llamadas es significativamente menor a la tasa de arribo de estas, creándose una congestión de llamadas. El escenario de congestión provoca que muchos clientes desistan en su espera y finalicen la llamada antes de ser atendida por un agente. En franjas horarias pico, un conjunto de llamadas son dejadas en espera y tratadas según las políticas de cada empresa, una práctica usual es encolarlas a una cola de espera y reproducir un audio informativo.

Este trabajo realiza un análisis de los estados por los que transitan las llamadas a lo largo de su vida en un *callcenter* y propone técnicas de clasificación de patrones con el fin de generar alertas ante llamadas que presenten riesgo de ser abandonadas. De esta forma, ante una alerta de riesgo de abandono, se podrían tomar medidas específicas sobre la llamada con el fin de que el cliente no la abandone.

2. Especificación del problema

Se dispone de la información de un conjunto de llamadas que arriban a un departamento de un *callcenter* que registra aproximadamente 2800 llamadas mensuales, y presenta una tasa de abandono entre 1 y 2 llamadas cada 10 recibidas. Cada llamada se encuentra representada por trazas, las cuales representan un estado por el que la llamada transitó y, además, contiene valores de las características que la llamada presenta en ese momento. El conjunto de datos abarca la actividad del departamento por un período de seis meses.

3. Objetivo

El objetivo de este trabajo es explorar un abanico de soluciones que permitan someter una llamada, aún en progreso, a un clasificador que de como resultado o bien un valor indicativo del riesgo de abandono de la llamada o directamente le asigne la etiqueta correspondiente. Con este fin, se proponen dos tipos de soluciones:

- Tipo I: valor indicativo del riesgo de abandono a partir del tiempo restante antes de ser abandonada
- Tipo II: clasificación de la llamada en las clases `ABANDON` o `COMPLETE`

4. Preprocesamiento

El preprocesamiento consta de dos etapas. En la primer etapa se procesan las trazas y se convierten en llamadas. En la segunda etapa, se procesan las llamadas y a partir de estas se generan dos tipos de patrones, uno para cada tipo de solución.

4.1. Conversión de trazas a llamadas

El conjunto de trazas se encuentran repartidas en seis archivos con formato CSV, cada uno corresponde a un mes. Se implementa un *script* en lenguaje *Python* para procesar. Al ejecutar este *script*, se genera una colección de estructuras que contienen la información completa de cada llamada.

A continuación se puede observar el conjunto de trazas de una llamada finalizada por el cliente, luego de haber sido atendida correctamente:

```
139,"ENTERQUEUE","95xxxxxx","2016-05-01 09:28:56",IDAXWQEDWE
139,"RINGING","2016-05-01 09:28:56",IDAXWQEDWE
139,"CONNECT",17,"2016-05-01 09:29:14",IDAXWQEDWE
139,"COMPLETECALLER",17,236,"2016-05-01 09:33:09",IDAXWQEDWE
```

Como resultado de esta etapa, se obtiene una colección de 17250 llamadas de las cuales 2257 ($\tilde{1}3.1\,\%$) son abandonadas y 14993 ($\tilde{8}6.9\,\%$) son completadas.

La estructura que representa la información de cada llamada, y que son obtenidas al finalizar esta etapa, tienen el siguiente formato:

```
{
    #Entradas a Cola: 1,
    Día semana: 1,
    #Agentes: 3,
    Localidad: 96,
    Hora: 10,
    Minuto: 14,
    # Ringing: 1,
    Mes: 7,
    Tiempo espera: 30,
    Motivo de fin: COMPLETECALLER,
    estados: [
        ...
```

```
        ["RINGING", "2016-05-01 09:28:56", ...],
        ...
    ]
}
```

4.2. Generación de patrones a partir de llamadas

El hecho de que la clasificación de una llamada tiene sentido que ocurra antes de que esta finalice obliga a que las llamadas que sean sometidas al clasificador no sean directamente las estructuras obtenidas en la etapa anterior sino que sean patrones que representen llamadas en curso. Con el fin de poder entrenar los clasificadores con patrones que representen el escenario de uso, se establecen criterios para la generación de patrones a partir de la colección de llamadas obtenidas en la etapa anterior.

El proceso de generación de patrones se basa en tomar fragmentos de cada llamada, comenzando siempre desde el inicio, cada vez más largos (temporalmente), determinados por un cierto parámetro (en la implementación se le llamó `STEP_TIME`). Con cada fragmento de llamada se generan dos patrones, cada uno de estos será utilizado como patrón de entrenamiento de las dos soluciones planteadas.

Para los patrones a ser utilizados en la solución Tipo I, se agrega una característica que determina el tiempo que resta para que la llamada representada por el patrón sea abandonada. En caso de que la llamada en cuestión realmente sea abandonada, el tiempo está dado por el lapso entre el instante de fin del fragmento y el instante en el que la llamada es efectivamente abandonada, en otro caso (en el que la llamada es completada) se asigna un valor de tiempo suficientemente grande como para no crear alarma de abandono (en este trabajo se utilizó el tiempo en segundos correspondiente a cinco minutos).

Para los patrones a ser utilizados en la solución Tipo I, se agrega una característica de clase. La clase `ABANDON` es asignada al patrón si en una ventana de tiempo (definida por el valor del parámetro `WINDOWS_TIME`) que comienza en el instante de fin del fragmento y se extiende hacia adelante (en este trabajo se utilizó una ventana de 60 segundos), ocurre el evento de fin de llamada y ésta es efectivamente abandonada. En otro caso, se asigna al patrón la clase `COMPLETE`.

La Figura 1 representa una llamada completa y los eventos que la componen. Además, se representan los tres saltos de tiempo que definen tres fragmentos, luego convertidos en tres patrones para el tipo de solución Tipo II. Al primer patrón generado (correspondiente al primer fragmento) se le asigna la clase `COMPLETE` ya que la ventana de tiempo (Ventana 1) no contiene el estado de fin (Estado fin) de la llamada.

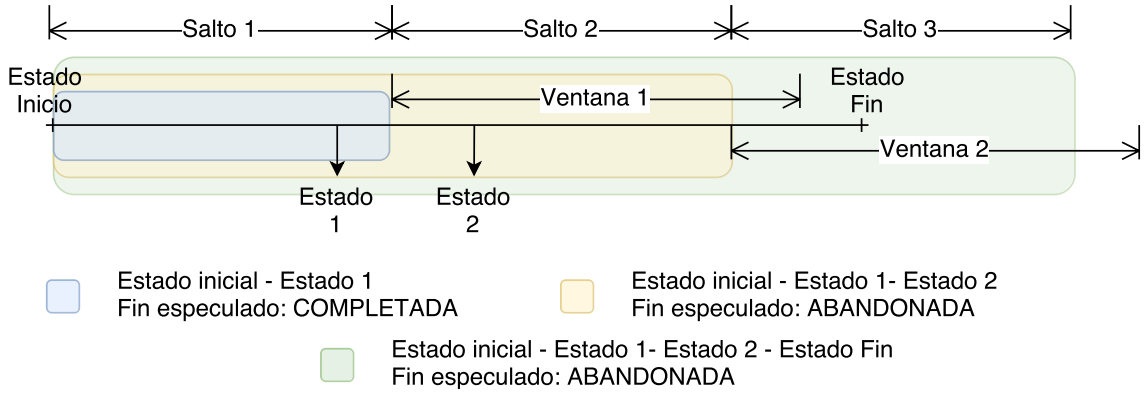

Figura 1: Llamada fragmentada que resulta en 3 patrones diferentes. Cada rectángulo de color corresponde al tiempo que abarca cada patrón generado.

A continuación se pueden observar dos patrones, el primero definido a partir de una llamada en curso que será completada y el segundo a partir de una llamada en curso que será abandonada. Es preciso aclarar que de las características Restante para abandonoz "Motivo fin.estará presente solo una, dependiendo del tipo de solución a la que se estén aplicando.

```
{
    #Entradas a Cola: 1,
    Día semana: 1,
    #Agentes: 3,
    Localidad: 96,
    Hora: 10,
    Minuto: 14,
    #Ringing: 1,
    Mes: 7,
    Tiempo espera: 30,
    Restante para abandono: 300,
    Motivo de fin: COMPLETE
}

{
    #Entradas a Cola: 1,
    Día semana: 1,
    #Agentes: 1,
    Localidad: 2216,
    Hora: 1,
    Minuto: 7,
    #Ringing: 0,
    Mes: 5,
    Tiempo espera: 120,
```

```
    Restante para abandono: 13,
    Motivo de fin: ABANDON
}
```

Se debe tener en cuenta que la forma de generación de patrones que se emplea en este trabajo resulta en un desbalanceo de clases, en el caso del tipo de solución Tipo II. La etapa de selección de características y/o el método de clasificación deberán tener en cuenta esto.

Como resultado de esta etapa se obtienen 223470 patrones donde, teniendo en cuenta los correspondientes al tipo de solución Tipo I, 8553 ($\tilde{3}.83\,\%$) son patrones con tiempo restante para abandono menor a 300 y el resto igual a 300; teniendo en cuenta los correspondientes al tipo de solución Tipo II, 5932 ($\tilde{2}.65\,\%$) son de clase `ABANDON` y el restante `COMPLETE`.

5. Características

El conjunto de características de un patrón está compuesto por 8 datos numéricos descritos a continuación:

- Cantidad entradas a cola: refiere a la cantidad de veces que una misma llamada es redirigida a al departamento de agentes. En caso de existir transferencias de la llamada entre agentes de un mismo departamento, este número se ve incrementado.
- Día de la semana: es un número entero en el rango de uno a 7 que corresponde a los días de la semana comenzando en el Lunes.
- Cantidad de agentes: número entero que indica la cantidad de agentes telefónicos vinculados a la cola de llamadas (i.e. disponibles para atender llamadas).
- Localidad: se refiere a la característica del número de teléfono que origina la llamada (e.g. los teléfonos móviles figuran con localidad 99, 98, etc; los teléfonos del barrio Crodón con 403, etc).
- Hora: corresponde a la hora de 0 a 23 en la que ocurrió la llamada.
- Minuto: corresponde al minuto de 0 a 59 en la que ocurrió la llamada.
- Cantidad de estados *ringing*: indica la cantidad de veces que una llamada entrante le fue ofrecida a diferentes agentes. Al ingresar una llamada al sistema, ésta es ofrecida a algún agente disponible, pasado un tiempo límite, si no fue atendida por el agente se vuelve a ofrecer a otro, así hasta que algún agente la atiende.
- Mes: número del mes en el que se generó la llamada, en el rango de 1 a 12.

- Tiempo de espera: es el tiempo, en segundos, que transcurre desde que la llamada entra al departamento hasta que es atendida por un agente telefónico.

A partir del conjunto de patrones obtenidos en el preprocesamiento, se aplican técnicas de selección de características. Estas técnicas se aplican con el fin de eliminar aquellas características que resultan poco relevantes al problema y como consecuencia generan ruido o "distracciones". Otra ventaja que presentan es que se evita el procesamiento de aquellos datos que no aportan, mejorando el rendimiento computacional. Además, un conjunto de datos más estable puede redundar en una mejora en el clasificador, disminuyendo el riesgo de *overfitting* y facilitando la visualización y la comprensión de los datos.

De las características obtenidas, lo intuitivo es pensar que cuanto más espera un cliente para ser atendido, más riesgo de abandono tiene la llamada, por lo tanto la característica *tiempo de espera* sería lógico que tuviera una relevancia alta. Además del tiempo de espera, otra característica que parecería importante es la cantidad de agentes que están vinculados al departamento en ese momento ($\#Agentes$) ya que sin agentes telefónicos, no se atienden las llamadas. Si bien la intuición nos lleva a pensar de esta forma, lo mejor es aplicar técnicas de selección y extracción, que de forma objetiva indiquen qué atributos conservar, cuáles descartar o cómo combinarlos.

En el caso del tipo de solución Tipo I, se aplica el método Principal component analysis (PCA) y para el caso del tipo de solución Tipo II, se aplica Ganancia de información y PCA.

6. Entrenamiento de clasificadores

Por un lado se entrenan clasificadores de dos clases y por otro modelos de regresión. En ambos casos se aplicarán técnicas de selección y extracción de características, previas al entrenamiento.

Para cada uno de los dos tipos de patrones con los que se cuenta, se aplican diferentes métodos de clasificación y regresión. Para la solución Tipo I, se aplican los métodos Regresión lineal, Support vector regression (SVR) y Árboles de decisión; mientras que para la Tipo II, se aplican los métodos Support vector machine (SVM) y Árboles de decisión.

Puesto que el tiempo de clasificación debe ser pequeño (en relación al tiempo de una llamada), se deja afuera el método k-NN, ya que para clasificar un patrón debería calcular la distancia a todos los patrones de entrenamiento y redundaría en un costo computacional más alto.

A continuación se detallan las configuraciones a ser ejecutadas para cada uno de los tipos de solución planteada:

Solución Tipo I

- Todas las características + SVR
- Todas las características + Regresión lineal
- Todas las características + Árbol de decisión
- PCA + SVR
- PCA + Regresión lineal
- PCA + Árbol de decisión

Solución Tipo II

- Todas las características + SVM (Polynomial Kernel)
- Todas las características + Árbol de decisión
- Todas las características + Matriz de costos + SVM (Polynomial Kernel)
- Todas las características + Matriz de costos + Árbol de decisión
- Ganancia de información + SVM (Polynomial Kernel)
- Ganancia de información + Árbol de decisión
- PCA + SVM (Polynomial Kernel)
- PCA + Árbol de decisión

Para el entrenamiento se utilizaron dos conjuntos de datos diferentes: clases balanceadas y tasas reales. El conjunto de patrones de clases balanceadas contiene tantos patrones que representan abandono como aquellos que representan completadas. El conjunto de patrones de tasas reales, contiene un porcentaje de patrones que representan las llamadas abandonadas, igual al porcentaje de llamadas abandonadas en la realidad (13.1 %). Por una cuestión de capacidad de procesamiento (ligado directamente al hardware con el que se cuenta), los conjuntos de datos son reducidos a 3600 patrones, siempre respetando las condiciones establecidas para cada uno (balanceo y tasa).

7. Análisis experimental

Para el entrenamiento se utilizó la herramienta Weka en modalidad *Explorer* y *Experimenter*. Para el método SVM se utilizó la implementación SMO, para el método SVR la implementación SMOreg, para el método de árbol de decisión en el clasificador se utilizó C4.5 implementado como J48 mientras que para el de regresión se utilizó REPTree, y para el método de regresión lineal se utilizó la implementación LinearRegression. El hardware utilizado fue una notebook Sony Vaio SVP-132 que cuenta con un procesador Intel-i7 4500U, 8 Gb de memoria RAM, disco de almacenamiento del tipo SSD y sistema operativo Ubuntu 16.04 de 64bits.

Se registran 28 configuraciones diferentes, cada configuración es ejecutada 10 veces en la modalidad Cross-Validation con 10-fold, lo que totaliza 100 resultados diferentes por configuración.

Para el tipo de solución Tipo I, nos interesa conocer el error medio y su desviación estándar; mientras que para el tipo de solución Tipo II consideramos importante la tasa de clasificaciones correctas y la tasa de verdaderos positivos en la clasificación correspondiente a `ABANDON`, pudiendo descuidar, con mesura, la tasa de falsos positivos (es más conveniente clasificar `ABANDON` cuando debió ser `COMPLETE`, que lo contrario).

7.1. Solución Tipo I

En las Figuras 2 y 3, se presentan dos tablas con los errores medios obtenidos al procesar los conjuntos de test en los modelos de regresión, junto a la desviación estándar (por cada conjunto de 10 ejecuciones y en el total).

Ejecución	PCA + LinearReg	PCA + SMOReg	PCA + REPTree	LinearRegre	SMOReg	REPTree
1 (10)	62.56 (3.17)	58.28 (3.87)	40.63 (4.83)	45.42 (2.99)	29.05 (4.14)	30.53 (2.47)
2 (10)	62.55 (3.16)	58.21 (3.73)	40.28 (4.62)	45.43 (3.35)	29.05 (4.87)	30.17 (2.82)
3 (10)	62.51 (3.11)	58.22 (3.66)	40.89 (2.71)	45.38 (2.84)	29.03 (3.37)	30.33 (2.93)
4 (10)	62.45 (4.03)	58.23 (4.49)	40.56 (4.53)	45.41 (3.08)	29.04 (4.71)	30.45 (3.81)
5 (10)	62.51 (3.27)	58.18 (3.97)	40.96 (3.64)	45.40 (2.92)	29.05 (4.00)	30.69 (3.54)
6 (10)	62.47 (3.74)	58.19 (3.75)	41.77 (3.89)	45.38 (2.75)	29.03 (2.98)	30.69 (3.07)
7 (10)	60.64 (8.33)	55.19 (12.14)	41.16 (4.73)	45.50 (5.28)	29.23 (7.48)	30.35 (5.33)
8 (10)	62.45 (3.56)	58.20 (3.77)	40.16 (4.99)	45.38 (2.45)	29.07 (3.73)	30.81 (3.39)
9 (10)	62.56 (3.33)	58.26 (4.24)	41.34 (3.16)	45.49 (2.88)	29.03 (4.81)	30.66 (2.80)
10 (10)	62.57 (2.56)	58.26 (2.76)	40.52 (3.72)	45.42 (2.50)	29.04 (3.43)	30.02 (3.31)
Media (SD)	**62.33 (3.98)**	**57.92 (5.12)**	**40.83 (3.98)**	**45.42 (3.05)**	**29.06 (4.31)**	**30.47 (3.28)**

Figura 2: Tabla de resultados de aplicar los métodos al conjunto de patrones con clases balanceadas.

Ejecución	PCA + LinearReg	PCA + SMOReg	PCA + REPTree	LinearRegre	SMOReg	REPTree
1 (10)	41.19 (4.29)	31.28 (5.64)	33.89 (4.80)	41.12 (4.24)	31.31 (5.68)	30.71 (3.77)
2 (10)	41.20 (2.68)	31.30 (5.23)	33.89 (2.81)	41.09 (2.69)	31.33 (5.19)	30.65 (2.95)
3 (10)	41.21 (3.49)	31.29 (4.92)	34.14 (4.92)	41.12 (3.44)	31.34 (4.91)	31.72 (3.30)
4 (10)	41.22 (3.71)	31.32 (5.70)	33.71 (3.06)	41.15 (3.70)	31.33 (5.66)	31.43 (4.49)
5 (10)	41.12 (2.58)	31.39 (3.59)	33.89 (2.03)	41.09 (2.60)	31.32 (3.55)	30.60 (2.71)
6 (10)	41.22 (2.43)	31.33 (2.37)	33.76 (3.18)	41.12 (2.48)	31.33 (2.35)	31.06 (2.94)
7 (10)	41.18 (3.83)	31.34 (6.08)	34.66 (5.34)	41.10 (3.80)	31.38 (6.06)	31.31 (4.39)
8 (10)	41.24 (2.58)	31.38 (5.93)	34.01 (2.64)	41.12 (2.55)	31.35 (5.96)	31.13 (3.31)
9 (10)	41.27 (2.51)	31.33 (4.98)	34.15 (2.96)	41.12 (2.48)	31.31 (4.96)	30.78 (3.90)
10 (10)	41.26 (2.59)	31.35 (3.89)	33.69 (2.50)	41.10 (2.61)	31.35 (3.95)	30.79 (2.81)
Media (SD)	**41.21 (2.99)**	**31.33 (4.73)**	**33.98 (3.44)**	**41.11 (2.98)**	**31.33 (4.73)**	**31.02 (3.37)**

Figura 3: Tabla de resultados de aplicar los métodos al conjunto de patrones con tasa real de abandonadas y completadas.

Tomando el total de los modelos analizados, podemos destacar el buen desempeño de SVR y árbol de decisión, ambos entrenados con el conjunto de clases balanceadas.

7.2. Solución Tipo II

Las Figuras 4 y 5 presentan las tasas de patrones correctamente clasificados en la solución de Tipo II. En este tipo de solución, no se debe tener únicamente en cuenta este valor, sino que también es importante la tasa de patrones `ABANDON` correctamente clasificados. Para esto último, usaremos las matrices de confusión, que nos ayudan a tener mejor idea del comportamiento del clasificador.

Ejecución	InfoGain + SMO	PCA + SMO	InfoGain + J48	PCA + J48	CostSensitive +	CostSensitive +	SMO	J48
1 (10)	93.64(1.29)	92.75(1.46)	94.17(1.24)	91.81(1.43)	93.64(1.29)	92.83(1.64)	93.64(1.29)	94.17(1.24)
2 (10)	93.64(1.33)	92.75(1.19)	94.14(1.56)	92.08(1.47)	93.64(1.33)	92.81(1.51)	93.64(1.33)	94.14(1.56)
3 (10)	93.64(1.63)	92.69(1.29)	94.17(1.45)	91.89(1.84)	93.64(1.63)	93.11(1.69)	93.64(1.63)	94.17(1.45)
4 (10)	93.64(1.33)	92.83(1.33)	94.11(1.21)	91.92(1.41)	93.64(1.33)	93.00(1.55)	93.64(1.33)	94.11(1.21)
5 (10)	93.64(1.52)	92.75(1.56)	94.11(1.46)	91.69(1.44)	93.64(1.52)	92.92(1.32)	93.64(1.52)	94.11(1.46)
6 (10)	93.64(1.29)	92.72(1.26)	94.17(1.39)	91.97(1.23)	93.64(1.29)	93.11(1.60)	93.64(1.29)	94.17(1.39)
7 (10)	93.64(1.25)	92.81(1.57)	94.17(1.23)	91.67(1.49)	93.64(1.25)	92.42(1.51)	93.64(1.25)	94.17(1.23)
8 (10)	93.64(1.14)	92.78(1.08)	94.08(1.08)	91.72(1.43)	93.64(1.14)	93.00(1.17)	93.64(1.14)	94.08(1.08)
9 (10)	93.64(1.48)	92.78(1.37)	94.17(1.40)	91.53(1.70)	93.64(1.48)	92.83(1.04)	93.64(1.48)	94.17(1.40)
10 (10)	93.64(1.22)	92.69(1.19)	94.14(1.17)	91.56(1.36)	93.64(1.22)	92.94(0.97)	93.64(1.22)	94.14(1.17)
Media (SD)	**93.64(1.29)**	**92.76(1.28)**	**94.14(1.26)**	**91.78(1.43)**	**93.64(1.29)**	**92.90(1.37)**	**93.64(1.29)**	**94.14(1.26)**

Figura 4: Tasa de clasificación correcta al aplicar los métodos al conjunto de patrones con clases balanceadas.

Ejecución	InfoGain + SMO	PCA + SMO	InfoGain + J48	PCA + J48	CostSensitive +	CostSensitive + SMO	J48
1 (10)	89.89(0.95)	89.89(0.95)	90.64(1.76)	89.89(1.28)	89.89(0.95)	89.58(1.05)	89.89(0.95)	90.64(1.80)
2 (10)	89.89(1.61)	89.89(1.61)	90.83(1.65)	89.72(1.92)	89.89(1.61)	89.75(1.73)	89.89(1.61)	90.89(1.57)
3 (10)	89.89(1.55)	89.89(1.55)	90.61(0.64)	89.83(1.85)	89.89(1.55)	89.58(1.75)	89.89(1.55)	90.69(0.60)
4 (10)	89.89(1.72)	89.89(1.72)	90.58(1.19)	89.97(1.70)	89.89(1.72)	89.67(1.44)	89.89(1.72)	90.64(1.19)
5 (10)	89.89(1.14)	89.89(1.14)	91.33(1.45)	89.42(1.26)	89.89(1.14)	89.97(1.58)	89.89(1.14)	91.33(1.42)
6 (10)	89.89(1.52)	89.89(1.52)	90.69(0.77)	90.33(1.36)	89.89(1.52)	89.92(1.42)	89.89(1.52)	90.64(0.71)
7 (10)	89.89(1.42)	89.89(1.42)	90.31(1.87)	89.56(1.41)	89.89(1.42)	89.53(1.32)	89.89(1.42)	90.33(1.83)
8 (10)	89.89(1.71)	89.89(1.71)	91.42(1.33)	89.67(1.57)	89.89(1.71)	89.78(1.61)	89.89(1.71)	91.39(1.30)
9 (10)	89.89(2.07)	89.89(2.07)	91.25(1.18)	89.86(1.83)	89.89(2.07)	89.75(2.11)	89.89(2.07)	91.25(1.17)
10 (10)	89.89(1.54)	89.33(1.49)	91.17(1.11)	89.72(1.57)	89.89(1.54)	89.56(1.51)	89.89(1.54)	91.11(1.16)
Media (SD)	**89.89(1.48)**	**89.83(1.48)**	**90.88(1.34)**	**89.80(1.54)**	**89.89(1.48)**	**89.71(1.51)**	**89.89(1.48)**	**90.89(1.31)**

Figura 5: Tasa de clasificación correcta al aplicar los métodos al conjunto de patrones con tasa real de llamadas abandonadas y completadas.

Matrices de confusión con el conjunto de patrones con clases balanceadas:

Matriz de confusión InfoGain + SMO:

```
   a    b   <-- classified as
1757   43 |    a = ABANDON
 186 1614 |    b = COMPLETE
```

Matriz de confusión PCA + SMO:

```
   a    b   <-- classified as
1746   54 |    a = ABANDON
 207 1593 |    b = COMPLETE
```

Matriz de confusión InfoGain + J48:

```
   a    b   <-- classified as
1776   24 |    a = ABANDON
 186 1614 |    b = COMPLETE
```

Matriz de confusión PCA + J48

```
   a    b   <-- classified as
1730   70 |    a = ABANDON
 225 1575 |    b = COMPLETE
```

Matriz de confusión CostSensitive + SMO:

```
   a    b   <-- classified as
1757   43 |    a = ABANDON
 186 1614 |    b = COMPLETE
```

Matriz de confusión CostSensitive + J48:

```
   a    b   <-- classified as
1783   17 |    a = ABANDON
 241 1559 |    b = COMPLETE
```

Matriz de confusión SMO:

```
   a    b   <-- classified as
1757   43 |    a = ABANDON
 186 1614 |    b = COMPLETE
```

Matriz de confusión J48:

```
   a    b   <-- classified as
1776   24 |    a = ABANDON
 186 1614 |    b = COMPLETE
```

Matrices de confusión con el conjunto de patrones con tasas reales:

Matriz de confusión InfoGain + SMO:

```
   a    b   <-- classified as
 461    7 |    a = ABANDON
 357 2775 |    b = COMPLETE
```

Matriz de confusión PCA + SMO:

```
   a    b   <-- classified as
 461    7 |    a = ABANDON
 357 2775 |    b = COMPLETE
```

Matriz de confusión InfoGain + J48:

```
   a    b   <-- classified as
 369   99 |    a = ABANDON
 238 2894 |    b = COMPLETE
```

Matriz de confusión PCA + J48

```
   a    b   <-- classified as
 424   44 |    a = ABANDON
 320 2812 |    b = COMPLETE
```

Matriz de confusión CostSensitive + SMO:

```
    a    b   <-- classified as
 461    7 |    a = ABANDON
 357 2775 |    b = COMPLETE
```

Matriz de confusión CostSensitive + J48:

```
  a    b   <-- classified as
450   18 |    a = ABANDON
357 2775 |    b = COMPLETE
```

Matriz de confusión SMO:

```
  a    b   <-- classified as
461    7 |    a = ABANDON
357 2775 |    b = COMPLETE
```

Matriz de confusión J48:

```
  a    b   <-- classified as
370   98 |    a = ABANDON
239 2893 |    b = COMPLETE
```

Se destacan como clasificadores con buen desempeño, CostSensitive + J48 y J48 en el conjunto de clases balanceadas mientras que en el conjunto de tasas reales SVN, CostSensitive + SVN, InfoGain + SVN, y PCA + SVN.

8. Conclusiones

Este trabajo concluye que es factible implementar un clasificador y un modelo de regresión que ayude a alertar sobre el riesgo de abandono de llamadas en un *callcenter*.

Mediante este informe, se da prueba de que es posible dar una solución práctica a la problematica que se abordó. Esta afirmación está basada en los buenos resultados que presenta, tanto la solución Tipo I como la Tipo II. En el caso de la solución Tipo I, los modelos de regresión, se llega a experimentar una media de error de 29 y una desviación estándar de 4, estos valores representan segundos y tomando en cuenta en el contexto que se aplican (llamadas telefónicas) representan tiempos totalmente manejables en un *callcenter*. En el caso del Tipo II, los clasificadores llegan a clasificar correctamente más del 98 % de los patrones que representan llamadas abandonadas y más del 88 % de las que representan completadas. Si bien la clasificación de patrones que representan llamadas completadas no es tan buena, en el problema planteado sería correcto intentar mejorar la clasificación de abandonadas aún cuando esto degrada en cierta forma la clasificación de las completadas.

Referencias

[1] C Bishop. Pattern recognition and machine learning (information science and statistics), 1st edn. 2006. corr. 2nd printing edn, 2007.

[2] Jason Brownlee. Design and run your first experiment in weka. http://machinelearningmastery.com/design-and-run-your-first-experiment-in-weka/, 2014. Último acceso 05-12-2016.

[3] Jason Brownlee. How to use regression machine learning algorithms in weka. http://machinelearningmastery.com/use-regression-machine-learning-algorithms-weka/, 2016. Último acceso 05-12-2016.

[4] Adrian Sampson. Statistical mistakes and how to avoid them. http://www.cs.cornell.edu/ãsampson/blog/statsmistakes.html, 2016. Último acceso 05-12-2016.

[5] Paul Paisitkriangkrai. Linear regression and support vector regression. http://cs.adelaide.edu.au/čhhshen/teaching/ML_SVR.pdf, 2012. The University of Adelaide. Último acceso 05-12-2016.