UNIDAD TEMÁTICA 7: Ajuste, evaluación y sintonía de modelos

Trabajo de Aplicación 2

ESTUDIO DE CASO – MARKETING BASADO EN AFINIDAD

Un banco crea un nuevo producto financiero, un tipo de cuenta corriente con ciertos costos y tasas de interés, diferentes de otros productos preexistentes. Pasado un tiempo desde el lanzamiento del nuevo producto, una cierta cantidad de clientes han abierto cuentas del nuevo tipo, pero hay muchos otros que aún no lo han hecho.

El departamento de marketing del banco quiere promover las ventas del nuevo producto mediante una campaña de mail directo a los clientes que aún no han optado por el mismo. Sin embargo, con el objetivo de no desperdiciar esfuerzos en clientes que no es probable que compren, desea dirigirse solamente al 20% de clientes que tengan la mayor ***afinidad*** por el nuevo producto.

Entonces, ¿cómo podemos determinar si un cliente tiene una gran afinidad por nuestro nuevo producto?

Asumiremos que los clientes que ya han comprado el producto (los compradores) son representativos de aquellos que tienen gran afinidad hacia el mismo. Entonces buscamos clientes que todavía no hayan comprado (los no compradores) pero que sean similares a los compradores en otros aspectos. Nuestra esperanza es que, cuanto más similares sean, mayor será su afinidad.

Nuestro desafío principal es entonces identificar las propiedades de los clientes que nos puedan ayudar a encontrar la similitudes, y que se encuentren disponibles en los datos del banco.

Asumiendo que tenemos buenos datos, podemos utilizar un método de minería de datos estándar para tratar de diferenciar entre compradores y no compradores. Afortunadamente la mayoría de algoritmos pueden generar un *ranking* de clientes, en el cual los que tienen ranking más alto serán clientes predichos como compradores, con mayor nivel de confianza o probabilidad que aquéllos que tengan menor ranking.

Deseamos entonces desarrollar varios modelos de minería, cada uno de ellos capaz de desarrollar un ranking de no compradores en el que los que tengan mayor ranking sean aquéllos para los que el modelo ofrezca más confianza de que deberían, de hecho, ser compradores (si sólo lo supieran!). Veremos también cómo decidir qué modelo es más útil.

Podremos entonces satisfacer al departamento de marketing, proveyéndolo con el 20% superior de los no compradores de nuestro ranking final.

Paso 1 – comprensión del negocio

El banco ofrece cuatro tipos de cuentas corriente, CC01 CC04, siendo este último el nuevo tipo que se desea promover.

Básicamente, cada tipo de cuenta viene con ciertos costos e intereses mensuales fijos para créditos y débitos, pero algunos clientes pueden tener tasas especiales o estar exentos del pago de los costos mensuales debido a su status VIP u otras particularidades.

Un cliente puede tener cualquier cantidad de cuentas corriente (incluso cero) y cualquier combinación de tipos

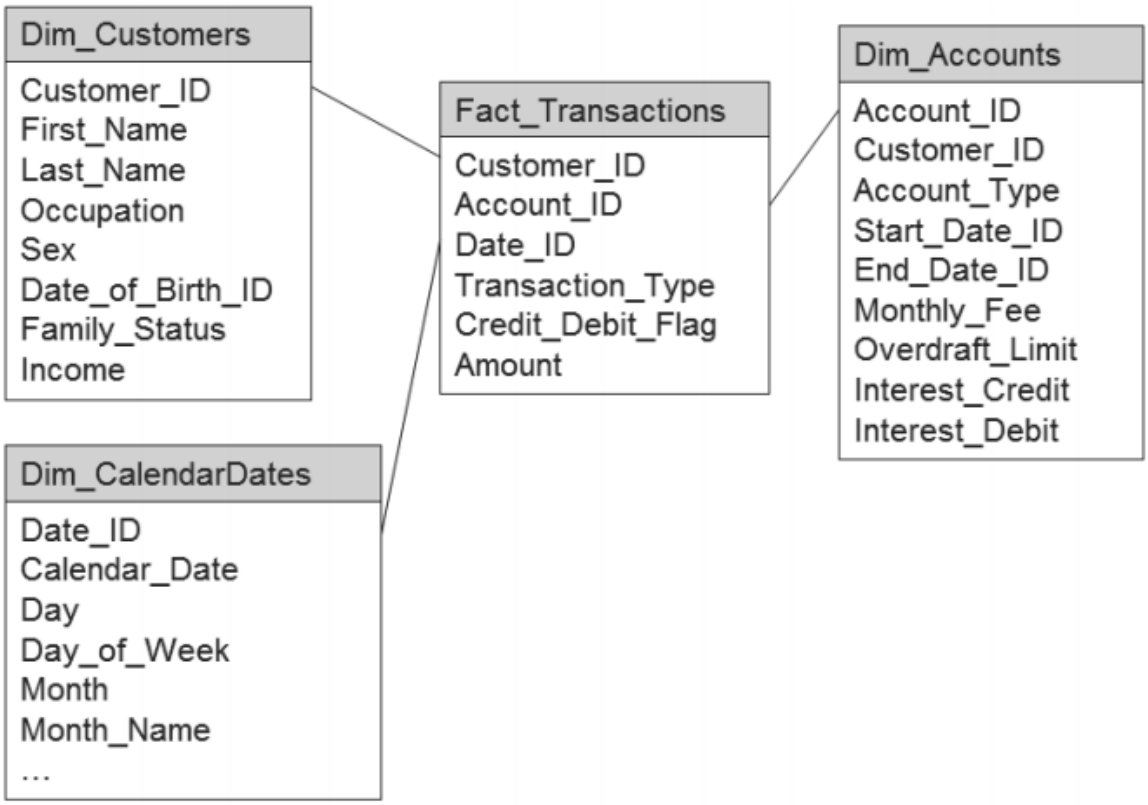
Las cuentas tienen fechas de apertura y cierre. Cuando se cierra una cuenta, su balance es cero y no puede aceptar más transacciones de dinero. Una cuenta abierta cuyo cierre aún no ha pasado, se denomina activa, y un cliente que tiene al menos una cuenta activa es llamado activo.

Cada transacción monetaria, de cada cuenta, es clasificada automáticamente mediante un sistema de análisis de texto interno, basado en un formulario opcional de texto libre que puede ser llenado por el iniciador de la transacción.

Existen varias categorías, como ser “retiro de caja”, “sueldo”, “prima de seguro”, etc., incluyendo una categoría “desconocido”.

Los datos personales como estado familiar o fecha de nacimiento son conocidos, para la mayoría de los clientes, pero no siempre están actualizados.

Los clientes pueden comprar muchos otros productos del banco, incluyendo cuentas de ahorros, tarjetas de crédito, préstamos o seguros (si bien es muy valiosa, esta información no está disponible en nuestro dataset).



Paso 1 – preparación de los datos

- Crea un dataset con 10000 ejemplos utilizando el operador "**Generate Direct Mailing Data**"
  seleccionando una semilla aleatoria local para asegurar la repetibilidad del ejercicio.
- Convierte el atributo "label" a binomial. Esto te permitirá seleccionar métricas de rendimiento específicas para clasificación binomial.
- Divide el dataset con Split Data en dos conjuntos, 80% para entrenamiento y 20% para testeo.
- Conecta la salida de 80% a un operador "Split Validation" y configura éste con una proporción de 0.7/0.3, y muestreo aleatorio.

Paso 2 – Operador de modelado y parámetros

Inserta un operador "Naive Bayes" dentro del proceso "**Split Validation**", seguido del "**Apply Model**" y un operador "**Performance (Binomial Classification)**". Configura las siguientes opciones en este último:

- accuracy,
- false positive,
- false negative,
- true positive,
- true negative,
- sensitivity,

- specificity, y
- AUC

Paso 3 Evaluación

Agrega otro operador **Apply Model** fuera del Split Validation, conéctale el modelo a su puerto "mod", y el dataset de test (20%) a su puerto "unl".

Agrega un operador "**Create Lift Chart**" con las siguientes opciones:

- target class = response,
- binning type =frequency, y
- number of bins = 10.

Paso 4 Ejecución e interpretación

Al ejecutar el modelo se generará la matriz de confusión y la curva **ROC** para la muestra de validación (30% del original 80%), mientras que generaremos una "**lift curve**" para la muestra de test (20%). Podríamos agregar otro "**Performance (Binomial Classification)** para el dataset de test.

Registra los valores de TP, TN, FP, FN

Observa los resultados de los parámetros seleccionados y verifica los cálculos:

Term	Definition
Sensitivity	TP/(TP+FN)
Specificity	TN/(TN+FP)
Precision	TP/(TP+FP)
Recall	TP/(TP+FN)
Accuracy	(TP+TN)/(TP+TN+FP+FN)

Observa que RapidMiner hace una distinción entre las dos clases cuando calcula la **precisión** y **recall**.

Para calcular el **recall** para "no response", la clase positiva tomada es "no response". ¿Cuál es el TP correspondiente? Y el FN? Cuánto el **recall** entonces para "no response"? Ver que esto contrasta con el valor que nosotros calculamos, pues asumimos que "response" era la clase "positiva".

El **recall** de la clase es una métrica muy importante, a tener particularmente en cuenta cuando tratamos con datos muy desbalanceados. Se considera que los datos están muy desbalanceados si la proporción de las clases está sesgada.

Al entrenar un modelo con datos desbalanceados, los valores de **recall** de las clases resultantes también tienden a quedar sesgados. Por ejemplo, en un dataset en el que hubiera

sólo 2% de “responses” el modelo resultante puede tener un muy alto valor de **recall** para “no responses” pero un valor muy pequeño de **recall** de clase para “responses”.

Este sesgo no se aprecia en la exactitud general del modelo, pero luego utilizar este modelo sobre datos no vistos puede resultar en severos errores de clasificación.

La solución para este problema puede ser o bien balancear los datos de entrenamiento para tener una proporción más o menos similar de las clases, o insertar penalidades o costos sobre las clasificaciones erróneas utilizando un operador “**Metacost**”. Analiza el funcionamiento de este operador y evalúalo.

El valor **AUC** (“Area Under Curve”) se muestra junto con la curva **ROC**.

Valores de AUC cercanos a 1 son indicativos de un buen modelo.

Mientras la predicción es correcta para los ejemplos, la curva da un paso hacia arriba (TP incrementado). Si la predicción es errónea, la curva da un paso hacia la derecha (FP incrementado).

Nota que RapidMiner puede mostrar dos curvas más **AUC**: “**optimistic**” y “**pessimistic**”. Investiga y documenta qué hacen estas opciones.

Analiza la curva y documenta los hallazgos. ¿Cómo se puede interpretar en función de los datos y del modelo?

EJERCICIO DOMICILIARIO: ANALIZAR LA SALIDA DE LIFT CHART Y DOCUMENTARLA