Memoria Proyecto Claim Insurance Prediction

Aniana González & Montse Figueiro

27 de octubre de 2016

MEMORIA PROYECTO "CLAIM INSURANCE PREDICTION"

INTRODUCCIÓN

Una empresa aseguradora necesita predecir en función de los vehículos siniestrados en los años 2005, 2006 y 2007 que vehículos tendrán siniestro con o sin daño corporal en los años 2008 - 2009 y la indemnización que conllevaría ese daño.

Objetivo 1)

Clasificar que vehiculos tendrán siniestro con daño corporal en función de sus características.

Objetivo 2)

Predecir la indemnización por daños corporales en función de las caracteristicas del vehiculo asegurado.

Información aportada por la empresa aseguradora

El fichero contiene los siguientes campos:

- Cada observación contiene la información anual sobre el seguro de un vehículo.

- La variable "Claim_Amount" ha sido ajustada para tener en cuenta los efectos de las características no correspondientes al vehículo, pero pueden tener interacciones interesantes con las variables del vehículo.
- "Calendar_Year" es el año en el que el vehículo fué asegurado.
- "Household_ID" es la identificación del hogar, en un hogar puede haber más de un vehículo asegurado.
- "Vehicle" es el número que identifica al vehículo, pero el mismo vehículo no tiene porque tener el mismo número en los diferentes años.
- Tenemos para identifiar el vehículo Model_Year, Blind_Make (manufacturer), Blind_Model, Blind_Submodel.
- El resto de columnas contienen características del vehículo así como, otras características asociadas a la poliza.
- Las variables numéricas han sido normalizadas, tienen media 0 y desviación standar 1.
- Tenemos dos datasets:
  - Training de 2005-2007 para construir el modelo
  - Test de 2008-2009 sobre el que se tendrían que realizar las predicciones. (no lo vamos a utilizar, no tenemos los importes para poder validar)

DESCRIPCIÓN DE LOS DATOS DE ENTRADA

Fuente de datos

Allstate Claim Prediction Challenge

Code Book Variables

Dictionary

METODOLOGIA

Hemos utilizado las metodologias R.

Lectura y visualización de datos

A) Leemos el fichero train_set.csv, con la función fread, debido al tamaño del fichero.

B) Tenemos un problema de memoria, el cual queda solventado con la función: memory.limit.

C) Debido al uso de la función "fread" realizamos el cambio a factor de algunas de las variables.

D) Visualizamos los datos generales de la póliza:

- Comprobación de observaciones duplicadas. Vemos que no tenemos ninguna, aunque el mismo vehículo puede estar asegurado en años distintos.
- El porcentaje de observaciones que tienen daño corportal (0,725%) y las que no tienen (99,274%).
- El número de observaciones desglosadas por año, así como el desglose de las mismas dependiendo de si tienen daño corporal o no. Vemos que en el año 2005 teníamos un porcentaje de observaciones con daño corporal de 0,7488%, en el año 2006 de 0,7124% y en el año 2007 de 0,7167%.
- El número de casas que tienen asegurados 1 o más coches, en total son 4.309.042 casas diferentes.
- Cuantos vehículos únicos tenemos, los cuales pueden estar asegurados en diferentes años. El total de vehículos es 7.366.649.
- El número de modelos diferentes, el mismo modelo puede tener caracteristicas diferentes según la casa, ya que puede variar el color, la potencia, etc. El número de modelos diferentes es de 13.315.
- El número de observaciones que contienen algún NA (en el fichero original son "?"), en total son 9.457.989, y el número de bservaciones con todos los datos completos son 3.726.301.
- Creamos una tabla de resumen a través de la libreria VIM y un gráfico del patrón de los missing Values por Variable. Comprobamos que tenemos un total de 23.438.318 de campos con NA's.
- Vemos la correlación de todas las variables númericas en función del Claim_Amount y lo representamos gráficamente.

Problemas que presenta el Proyecto

```
- Número elevado de Missing Values
- Multicolinealidad entre variables
- Unbalanced Data (desequilibrio de datos). El 99,274% de las observaciones tienen Claim_Amount = 0. Pos
  - Under-Sampling: Eliminamos observaciones = 0, solo vale para ahorrar tiempo, perdemos información.
  - Over-Sampling: Implica hacer copias de la Clase mínima causando overfitting.
```

Limpieza de datos

A) Uno de nuestros objetivos es intentar redimensionar el dataset intentando reducir el número de observaciones.

B) Vemos la relación entre las variables categóricas y output binomial (target).

Utilizamos las observaciones que no tienen Missing Values para seleccionar las variables categóricas que tienen relación con el output. Las variables númericas no tienen Missing
Values y no necesitamos seleccionarlas en éste momento para reducir el dataset y realizar la limpieza de datos.

Comprobamos en que variables rechazamos la hipotesis nula (p-value$<$ 0.05), a través de la
función chisq.test. Vemos que las únicas variables que pueden ser independientes del output y para las cuales no rechazamos la hipotesis nula son: Cat10, Cat11 y Cat12.

B) Dividimos el subset en Train y Test (75% y 25% respectivamente)

C) Reducimos el fichero Train:

-Creamos un train_reducido, aggregamos los coches con importe 0 que tengan las mismas características independientemente de a que casa pertenezcan.(Ésto nos reduce el fichero de más de 9 millones a 600K observaciones)

-A este train_reducido le añadimos las observaciones que tienen importe por daño corporal.

-Observamos que las variables Cat1, Cat2, Cat3, Cat4, Cat5, Cat6, Cat7, Cat8, Cat9 se da la circunstancia de que no varían para el mismo Blind_Submodel, son iguales independientemente de la póliza, del asegurado y de la household_ID, con lo que summarizamos esta información para reducir el caso de Missing Values que tenemos en el dataset.

```
Una vez sumarizado aplicamos la función MICE para imputar las variables en las que tenemos missing valu
```

```
Grabamos el resultado como "traindf.csv".
```

D) Limpiamos el dataset Test, ya que también contiene NA's. Lo hacemos igual que el train cruzando los datos que nos ha aportado MICE.

   Grabamos el fichero obtenido como "testdf".

E) Sustituimos los Missing Values del fichero Train original sin reducir con los valores más frecuentes imputados con la función MICE.

   Grabamos el resultado como "train_completodf".

F) Finalmente tenemos que equilibrar los datos, esto lo hacemos sobre el dataset traindf, el cual tiene los datos agregados y no tiene missing values, para ello utilizamos la función downSample de la libreria caret.

Grabamos el resultado como "train_downSample".

ORDEN DE EJECUCIÓN DE LOS FICHEROS ADJUNTOS

En el fichero ClaimCarInsurance.rmd se realizaron todos los cálculos previos al desglose entre los distintos ficheros para cada modelo. Una vez finalizado el proyecto se ha eliminado el contenido del presente .rmd y se deja indicado el orden de ejecución de los diferentes archivos.

1) Memoria.pdf
2) ReadingData.rmd
3) DataCleaning.rmd
4) Correlaciones_Graficos_Datos.md
5) GLM_Clasificacion.rmd

6) LM_Predicción.rmd
7) SVM_Regression.rmd
8) RandomForestRegression.rmd
9) GBM_Regression.rmd
10) TreeRegression.rmd
11) RandomForestClassification.rmd
12) Comparacion_Modelos.md
13) ResultadosFinalesPredicciones.md

RESUMEN RESULTADO: COMPARATIVA DE MODELOS APLICADOS A CLASIFICACIÓN Y PREDICCIÓN

MODELO DE CLASIFICACIÓN - Machine learning

MODELO LINEAL GENERALIZADO (GLM)

Los modelos lineales se basan en los siguientes supuestos:

1. Los errores se distribuyen normalmente.
2. La varianza es constante.
3. La variable dependiente se relaciona linealmente con la(s) variable(s) independiente(s).

Los GLM son una extensión de los modelos lineales que permiten utilizar distribuciones no normales de los errores (binomiales, Poisson, gamma, etc) y varianzas no constantes.

A) Creamos un modelo GLM, (model_train_clas) para predecir si va a existir daño corporal o no. Utilizando el dataset (traindf), agregado pero sin balancear.

B) creamos un modelo GLM (model_down_clas) para predecir si va a existir daño corporal o no. Utilizando el dataset (train_downSample) con los datos balanceados y agregados.

Tabla con el resultado de la ejecución de los dos modelos aplicados al fichero test

	AIC	R^2	Accuracy	Precisión	Recall
model_train_clas	335.472	0.2720185	0.6608617	0.3029006	0.006511096
model_down_clas	139.578	0.299275	0.1950311	0.7569576	0.006766421

Con esta tabla podemos comprobar que el modelo que mejor se ajusta está entrenado con el train equilibrado, basándonos en su precisión del 75%. Su matriz de confusión es la siguiente:

	no	si
no	624705	5790
si	2647039	18033

Como podemos ver aunque el porcentaje de precisión es alto, la clasificación no es buena puesto que el número de Falsos Positivos es muy elevado, con lo que podemos confirmar que con el modelo de Lineal Generalizado las variables independientes no aportan información significativa para clasificar.

MODELO DE CLASIFICACIÓN CON RANDOM FOREST

Hemos realizado la clasificación tanto con el paquete randomForest como el paquete Caret con k-folder. Los resultados para los dos ficheros, el desequilibrado y el equilibrado mediante método Under-Sampling son los siguientes:

MODELO	Accuracy	Precisión	Recall
RF Train	**0.730748**	0.2409016	0.00655891
RF Train Caret	0.6445112	0.3096168	0.006345132
RF Train down	0.194633	0.7939806	0.007089187
RF Train down Caret	0.1789495	**0.8170256**	0.007153431

En el caso de los ficheros desequilibrados con alto porcentaje de observaciones con clase 0, el Accuracy suele ser elevado puesto que tiende a clasificar en la clase mayoritaria, no está clasificando todos en 0 porque no estamos utilizando el fichero Train Completo, con el fichero completo ningún modelo clasifica observaciones en la clase 1.

Al equilibrar el fichero tiende a clasificar el mismo número de observaciones con clase 0 y con clase 1. Esto se suele solucionar calibrando las probabilidades.

Calibrado de probabilidades para el modelo con mayor precisión (clasificamos como 1 cuando la probabilidad es mayor que 0.6, 0.7 o 0.95 respectivamente):

MODELO	Accuracy	Precisión	Recall
RF Train down Caret 60	0.2439089	0.7481426	0.0071190
RF Train down Caret 70	0.3178382	0.6653654	0.0070261
RF Train down Caret 95	0.179567	0.81627	0.00715226

Con las variables aportadas el modelo no clasifica bien, el número de Falsos Positivos y Falsos Negativos es muy elevado. En el caso de seleccionar como clase 1 aquellos que tienen una probabilidad mayor del 95% nos da 2.699.415 de observaciones como clase 1 erroneamente (Falsos Positivos).

MODELOS DE REGRESIÓN

Pasos realizados:

- Modelo sobre fichero Train Completo
- Modelo sobre dichero Train Agregado (Agregadas las observaciones con importe 0)
- Modelo Equilibrado (Under-Sampling)
- Cálculo R-Squared del modelo
- Predicción Claim_Amount fichero Test
- Cálculo RSME de las predicciones del fichero Test

MODELO DE REGRESIÓN LINEAL

Resultados validados con el fichero Test con 3.295.567 observaciones (El entrenamiento se ha hecho por triplicado: train entero,train agregado y train reducido con downsample)

Fichero Train	R-squared	RMSE
model_train_complet(Sin log)	3.31e-05	**39.34202**
model_train_completo (Con log)	0.0005613	39.36571

Fichero Train	R-squared	RMSE
model_train_SinBalan_Sinlog	0.03828	79.15849
model_train_SinBalan_log	0.1641	39.34774
model_downSample	0.05042	162.3858
model_downSample_log	0.274	39.38087

Cuando creamos el modelo a partir del fichero train completo vemos que R-squared o el coeficiente de determinación no llega a explicar ni el 1% de la variable "Claim_Amount". En nuestro fichero "Correlaciones_Graficos_Datos.Rmd" se puede observar como las correlaciones entre las variables independientes y la variable dependiente son casi nulas.

Nuestro coeficiente de determinación mejora cuando aplicamos la log-transformación a la variable dependiente. Llegando a obtener un R-Squared de 27,4% cuando aplicamos una técnica de Under-Sampling para igualar la proporción de las observaciones con importe 0 y con importe positivo.

Siendo el objetivo final la mejor predicción para las observaciones del fichero test, seleccionaríamos la opción con menor Error Cuadrático Medio (RMSE), en éste caso nos los ha dado el modelo a partir del train completo sin transformación logarítmica ni técnicas para redimensionar el dataset.

MODELOS DE REGRESIÓN NO LINEALES

Todos los modelos de predicción no lineales se han realizado a partir de una muestra aleatoria de 5000 observaciones del fichero Train Agregado y Train Down Sample, lo hemos realizado de ésta manera por los tiempos computacionales y para poder ver de una manera rápida si existe relación entre el tipo de logarítmo que se aplique y el resultado final en cuanto a Error Cuadrático Medio.

MODELO	LOG S/OUTPUT	EQUILIBRADO	RMSE	R^2
SVM	NO	NO	40.90	0.028
SVM	**SI**	NO	**39.3599**	**0.1694**
SVM	NO	SI	61.23	0.042
SVM	SI	SI	39.38	0.2959
RPART	NO	NO	112.36	0.013
RPART	**SI**	NO	**39.35003**	**0.0831**
RPART	NO	SI	140.93	0.040
RPART	SI	SI	39.38	0.1609
GBM	NO	NO	48.13	0.030
GBM	**SI**	NO	**39.34993**	**0.1794**
GBM	NO	SI	122.68	0.0555
GBM	SI	SI	39.35612	0.3732
RF	NO	NO	55.47	0.02
RF	**SI**	NO	**39.36**	**0.1501**
RF	NO	SI	130.74	0.0373
RF	SI	SI	39.38	0.3524

Lo primero que llama la atención es que R-Squared no tiene una relación directa con RMSE, en nuestro caso vamos a seleccionar el mejor RMSE puesto que nos aporta más información sobre las desviaciones que hemos tenido entre lo que hemos predicho y lo real.

Vemos que los modelos que mejores resultados nos dan no tienen porque estar equilibrados pero si que nos disminuye el RMSE aplicar una transformación logarítmica a nuestra variable dependiente.

RESULTADOS FICHERO TEST - VISUALIZACIÓN

Para cada Modelo aplicado seleccionamos según el criterio indicado el modelo que mejor predicción nos aporta. Éste resultado lo añadimos a una nueva columna dentro del fichero "testdf.csv". (Ejemplo: "predGLM")

La visualización de las predicciones para las observaciones del fichero Test se encuentran en el fichero "ResultadosFinalesPrediccionesTest.rmd".