Reto Salesforce
Predictive Modelling

Juan José Vidal – Another Machine Teacher

VNIVERSITAT DE VALÈNCIA

"Machine learning is a core, transformative way by which we're rethinking everything we're doing.“

Google CEO - Sundar Pichai

La preparación

Cómo se han manipulado los datos y se han seleccionado variables

El código

Utilizamos el software estadístico **R**, con el que manipularemos los datos y entrenaremos los modelos.

Hemos utilizado los paquetes *data.table*, *dplyr*, *foreach*, *ggplot2*, *xgboost*, *rpart*, *e1071* y *ranger*.

El código se divide en cinco partes:

- Manipulación de los datos
- Selección de variables significativas
- Modelización con eXtreme Gradient Boosted Models (XGB)
- Ajuste de hiperparámetros
- Entrenamiento de otros modelos

En el repositorio de GitHub se podrán encontrar tanto los scripts como las tablas intermedias generadas así como los resultados obtenidos.

VNIVERSITAT DE VALÈNCIA

UNIVERSITYHACK 2018 DATATHON

Manipulación de datos

Se consideran dos subtablas disjuntas, una con las variables numéricas y otra con las categóricas. El conjunto numérico se utilizará para el análisis de componentes principales (PCA) y el factorial para crear un diccionario para transformar estos factores a numéricos en el entrenamiento del XGBoost.

Observamos que la variable respuesta toma la forma de una Gamma, con una media de 16,421.41.

Separamos la table resultante en conjunto de entrenamiento *train* (90%) y conjunto de validación test (10%) para comparar los modelos que entrenaremos.

VNIVERSITAT DE VALÈNCIA

UNIVERSITYHACK 2018 DATATHON

Manipulación de datos

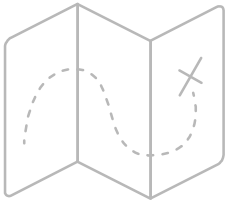
Sólo aparecen valores vacíos en la variable Socio_Demo_01, que para los XGBoosters los dejaremos como están y para los otros modelos los imputaremos con la moda de tal factor.

Para añadir variables, realizamos a partir de las variables numéricas un análisis de componentes principales. Se tiene en cuenta que la aplicación al negocio de variables sacadas de un PCA no suelen ser interpretables, por lo que el trabajo se ha realizado con y sin ellas. Igualmente, las hemos utilizado por la reducción del error que se produce al añadirlas.

VNIVERSITAT DE VALÈNCIA

UNIVERSITYHACK 2018 DATATHON

Selección de variables



Escogemos distintos subconjuntos aleatorios del train y realizamos un entrenamiento de eXtreme Gradient Boosted Models para cada uno.

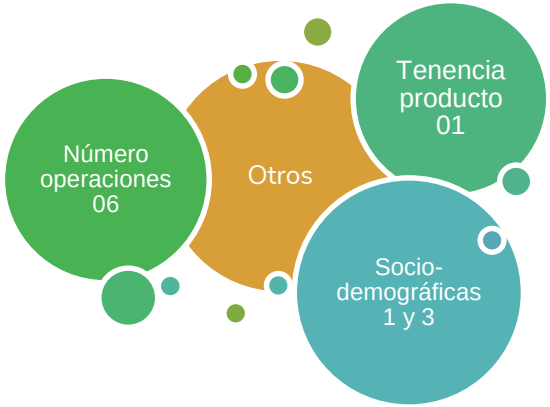
Seleccionamos las variables que más importancia tengan para explicar la respuesta, con un requerimiento mínimo de 0.5% de ganancia. Nos quedamos con las que aparezcan al menos en la mitad de los modelos.

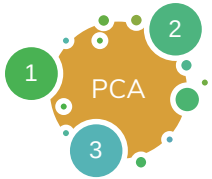
VNIVERSITAT DE VALÈNCIA

UNIVERSITYHACK 2018 DATATHON

Selección de variables

Resultados con el PCA





VNIVERSITAT DE VALÈNCIA

UNIVERSITYHACK 2018 DATATHON

El entrenamiento

Qué modelos se han entrenado y qué resultados hemos obtenido

Los modelos

Modelo Lineal

Intenta reducir el error a partir de ajustar los coeficientes que formen una combinación lineal sobre los datos para predecir la respuesta.

Modelo Lineal Generalizado

Expansión flexible del modelo lineal, ahora con transformación de la variable respuesta y con función de distribución de ésta.

Árbol de Decisión

Secuencia de condiciones que termina en un resultado que se aplica al subconjunto descrito en éstas.

Random Forest

Múltiples árboles de decisión sobre subconjuntos aleatorios de los datos que predicen a partir de una media ponderada de los resultados.

Support Vector Machine

Buscamos un hiperplano de los datos que intente clasificar la respuesta de la mejor manera posible.

eXtreme Gradient Boosting

Se crea una sucesión de árboles de decisión donde cada uno aprende de los anteriores y perfeccionan la predicción.

VNIVERSITAT DE VALÈNCIA

UNIVERSITYHACK 2018 DATATHON

Resultados

Se han calculado los Errores Medios Absolutos (MAE) y la Raíz de los Errores Cuadráticos Medios (RMSE) de los distintos modelos entrenados. Se ha realizado Cross-Validation para ajustar los hiperparámetros. El sobreajuste del XGBoost ha sido controlado mediante la monitorización del error en el test, obligando al modelo a parar de entrenar cuando éste dejase de disminuir.

	Modelo Lineal	Modelo Lineal Generalizado*	Árbol de Decisión	Random Forest	SVM	XGBoost**
MAE	5911.06	No Converge	6420.97	4584.76	No Converge	4334.75
RMSE	31376	No Converge	31951.09	29763.76	No Converge	30043.2

* Link Gamma

** Objetivo: Regresión Gamma / Métrica de evaluación: MAE

$$MAE = \frac{\sum_{i=1}^{n} |Y'_i - Y_i|}{n}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y'_i - Y_i)^2}$$

VNIVERSITAT DE VALÈNCIA

UNIVERSITYHACK 2018 DATATHON

Las conclusiones

Qué se ha decidido a partir de los resultados

Conclusiones

La recámara

El hecho de que el **GLM** y la **SVM** no hayan convergido no es de extrañar. Estos modelos requieren de un preprocesamiento de los datos muy amplio y costoso.

Otro punto en contra es la necesidad de imputación de valores vacíos, que siempre poducirá una pérdida de información.

Si que cabe mencionar la alta interpretabilidad de los GLM frente a la ausencia de ésta en las SVM.

La simplicidad

Pese a la simpleza que caracteriza tanto a los **modelos lineales** como a los **árboles de decisión**, éstos no han sido capaces de encontrar una segmentación que produzca unos resultados comparables con los de los modelos más complejos.

Aún así, su rápida ejecución y su fácil interpretabilidad hacen que cualquier Data Scientist los tenga en cuenta para un modelo rápido e interpretable para posibles jefes que no tengan conocimientos de Data Science.

La precisión

El **RandomForest** ha sido ejecutado con el paquete Ranger, preparado para bases de datos de mayor tamaño. Con este algoritmo, ha competido dignamente contra el **XGBoost**, conocido por la comunidad como uno de los mejores, sino el mejor, modelo de Machine Learning para regresión y clasificación.

Pese a que los resultados no son del todo concluyentes, se ha decidido dar la victoria al XGBoost por la posibilidad de manipular datos missing, capacidad que Ranger no posee si no se imputan previamente.

VNIVERSITAT DE VALÈNCIA

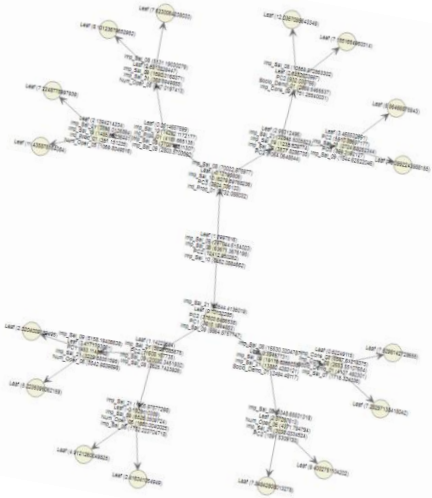
UNIVERSITYHACK 2018 DATATHON

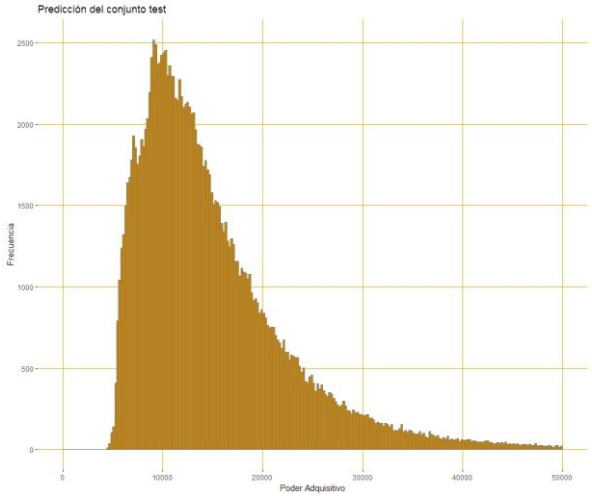
Conclusiones

El modelo escogido

Con la función *xgb.plot.multi.trees* se intenta capturar la complejidad de los árboles generados que usualmente se ven como una caja negra. Esta función nos permite seleccionar qué variables queremos mostrar como un único árbol de decisión con el objetivo de interpretar qué ocurre en las predicciones según el valor de cada variable. +Info

Se pueden encontrar en el repositorio de GitHub los gráficos con mayor calidad.





La distribución de las predicciones

Observamos como al predecir el poder adquisitivo de los clientes del conjunto test, conjunto del que no conocemos su valor, la distribución que toma es la de una Gamma, igual que lo hacían los valores conocidos del conjunto train. Este resultado es un buen indicador de que se ha controlado correctamente la sobreparametización del modelo.

VNIVERSITAT DE VALÈNCIA

UNIVERSITYHACK 2018 DATATHON

Muchas gracias 

Another Machine Teacher

- Juan José Vidal Llana
- xenxovidal@gmail.com