

Sistema de alerta temprana de abandono de comercios filiales de un negocio

PROBLEMA

En la actualidad, obtener un cliente nuevo es más caro que retener a los que ya se tienen. No contar con una manera *data driven* de determinar aquellos clientes en riesgo de abandono podría resultar en campañas de retención no óptimas económicamente.

OBJETIVO GENERAL

Crear una aplicación web que permita a los tomadores de decisiones identificar las filiales con riesgo de abandono para facilitar la ejecución de acciones preventivas.



PROPUESTA

Se propuso una solución en la cuál, mediante machine learning, se obtiene una lista de aquellos negocios en riesgo de abandono. De acuerdo a la revisión literaria, se tenían 3 opciones de algoritmos principales:

Árboles de decisión

- La ventaja principal de este algoritmo es que es posible visualizar la manera en que el modelo toma las decisiones. Para el negocio podría ser de gran ayuda ver qué factores ayudan a determinar el abandono de un negocio.

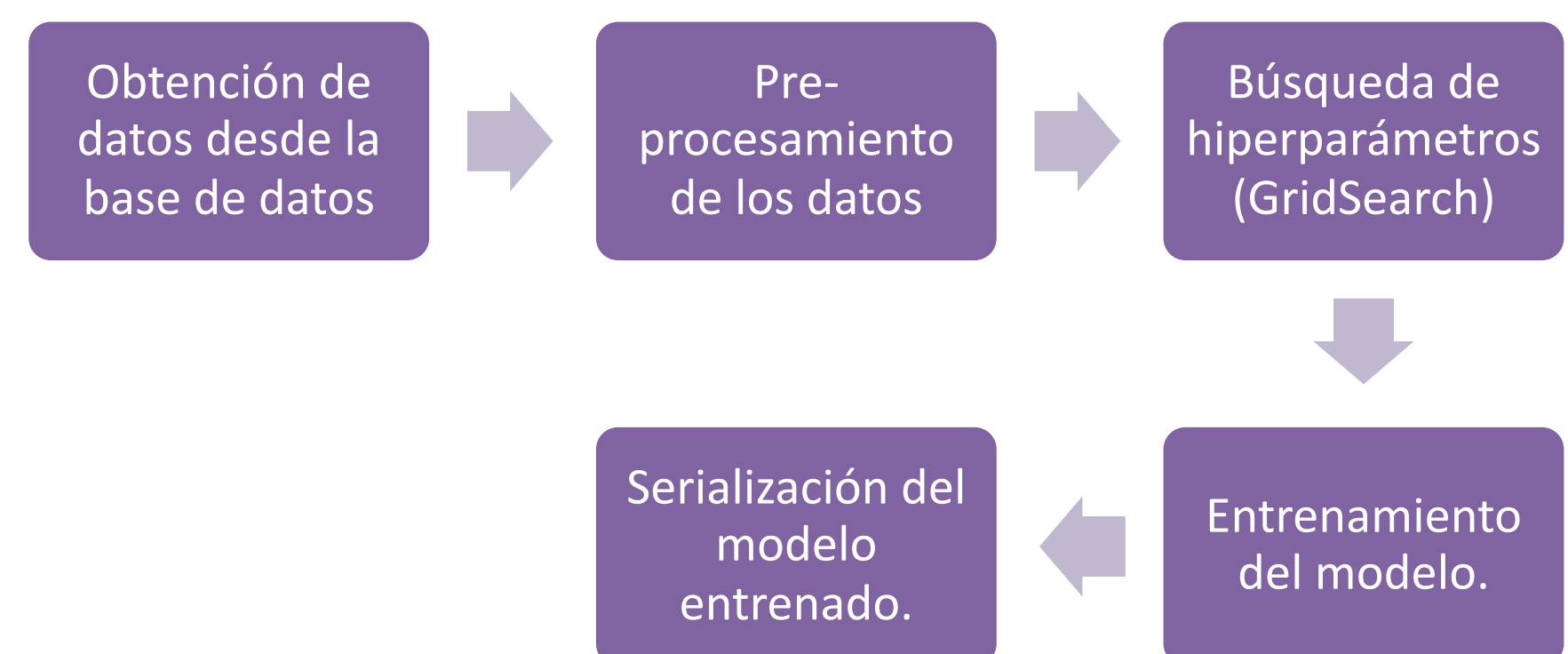
Gradient Boosting

- Este algoritmo sirve para crear modelos predictivos fuertes de a partir de modelos predictivos débiles. Una gran fortaleza de este algoritmo es su resistencia al overfitting.

Redes LSTM

- Estas redes neuronales son conocidas el campo del Deep Learning por la capacidad de "recordar" estados previos para luego utilizarlos y decidir cuál será el siguiente. Dicha capacidad podría ser útil para predecir si un negocio abandona.

Para evitar el degradado del modelo con el tiempo, se resolvió que este se reentrene mensualmente. Para lo cuál fue necesario encontrar una serie de hiperparámetros mediante los cuales el algoritmo GridSearch entrenaría el modelo de la mejor manera. El proceso mensual de reentrenamiento es el siguiente:



Este proceso necesita de un Backend, para el cuál se utilizó Django (Python) por sus múltiples herramientas de data science. Además, para mostrar los resultados se realizó un Frontend con React.js por la rapidez de desarrollo y gran disponibilidad de librerías.

RESULTADOS

Los resultados obtenidos para los modelos son los siguientes:

Árboles de decisión

Dataset	Accuracy	Precision	Recall
Train	91.63%	81.42%	100%
Test	92.74%	85.24%	98.96%

Gradient Boosting

Dataset	Accuracy	Precision	Recall
Train	91.63%	81.45%	99.97%
Test	92.74%	85.28%	98.93%

LSTM

Dataset	Accuracy	Precision	Recall
Train	69.30%	69%	100%
Test	65.52%	65.51%	99.99%

Por estos resultados, y el hecho de que GradientBoosting sea el modelo más resistente a overfitting, dicho algoritmo fue elegido. Los hiperparámetros elegidos para en reentrenamiento mensual son los siguientes:

Hiperparámetro	Valores
learning_rate	0.01, 0.05, 0.1, 0.2
min_samples_split	0.01, 0.05, 0.1, 0.17
min_samples_leaf	0.3, 0.37, 0.45
max_features	"sqrt", 2, 3, 5, 7, None
subsample	0.8, 0.95, 1
n_estimators	100, 250, 500, 1000

Estos hiperparámetros fueron encontrados mediante múltiples pruebas, en las que se eliminaban todos aquellos valores que se determinaban que causaban overfitting.

CONCLUSIONES

- El mejor modelo encontrado para el sistema desarrollado es GradientBoosting. Esto por dos razones principales: el buen desempeño en las 3 métricas evaluadas y su resistencia al overfitting
- El sistema desarrollado, además de proporcionar la lista de clientes en riesgo de abandono, permite visualizar datos históricos de los mismos. Esto con el propósito de que los usuarios tengan una idea general del estado de cada comercio.
- El sistema permite comparar todos aquellos comercios que están en riesgo de abandono con aquellos que son productivos. De esta forma, es posible obtener un contexto general y entender las diferencias entre estos dos tipos de comercios.
- Una vez confirmado por los usuarios que un comercio efectivamente ha abandonado, el sistema permite asignar la etiqueta de abandono. Esta etiqueta ayudará a reentrenar el modelo mes a mes.