SISTEMA DE PREDICCIÓN DE FRAUDE EN TARJETAS DE CRÉDITO

DANIEL CIRUELO SANZ

Introducción

La detección del fraude en tarjetas de crédito de manera automática ha sido un gran reto en los últimos tiempos. Este proyecto en concreto trata de predecir de forma efectiva el default en el próximo mes en tarjetas de crédito.

La idea original del proyecto era realizar, para un proyecto de mi empresa, un sistema de credit scoring en diferentes niveles según definición de usuario. La dificultad del cliente para darnos datos reales ha hecho que lo haga al final por mi cuenta.

Los datos no eran nada fácil de conseguir con lo que he utilizado un dataSet que encontré en el repositorio de GitHub https://github.com/AlexPnt/Default-Credit-Card-Prediction/tree/master/dataset.

FASES DEL PROYECTO Y ORGANIZACIÓN

El proyecto está dividido en las siguientes fases:

1. Preproceso de los datos.

2. Refinamiento y entrenamiento de los algoritmos

3. Visualización de resultados.

Está compuesto por diferentes carpetas:

1. Data: donde residirán los archivos de datos de origen y de resultados;

2. Función: con un script de Python, que contiene las funciones que van a utilizar todos los notebook donde se entrenarán los algoritmos

3. Notebook: la carpeta donde residen los notebook con los algoritmos,

4. Visualization: que alberga el dashboard de resultados.

1 . Preproceso de los datos:

El preproceso de los datos lo hago en le notebook de Python preproveso.ipynb.

Los datos se leen del archivo ‘data/default of credit card clients.csv’.

Una vez leídos exploramos los datos y los dominios de las variables para identificar cuales son categóricas, cuales discretas y cuales continuas. Se limpian de nulos, he determinado que se eliminan filas o columnas con al menos la mitad de datos nulos. Tras la limpieza, la gente suele limpiar los datos de outliers, yo he decidido dejarlos porque tendremos que tratar con ellos si el sistema entra en producción.

Los algoritmos de machine learning suelen funcionar mejor con datos numéricos que categóricos. Por eso se sustituimos las variables sex, marriage, education.

Para ver la distribución de las variables discretas las mostramos en un gráfico de barras y las continuas en un boxplot.

2. Entrenamiento de los algoritmos.

En este proyecto hemos trabajado con 3 algoritmos: Random Forest, SVM y Gradient Boosting, en notebook homónimos.

Las fases que realizamos son:

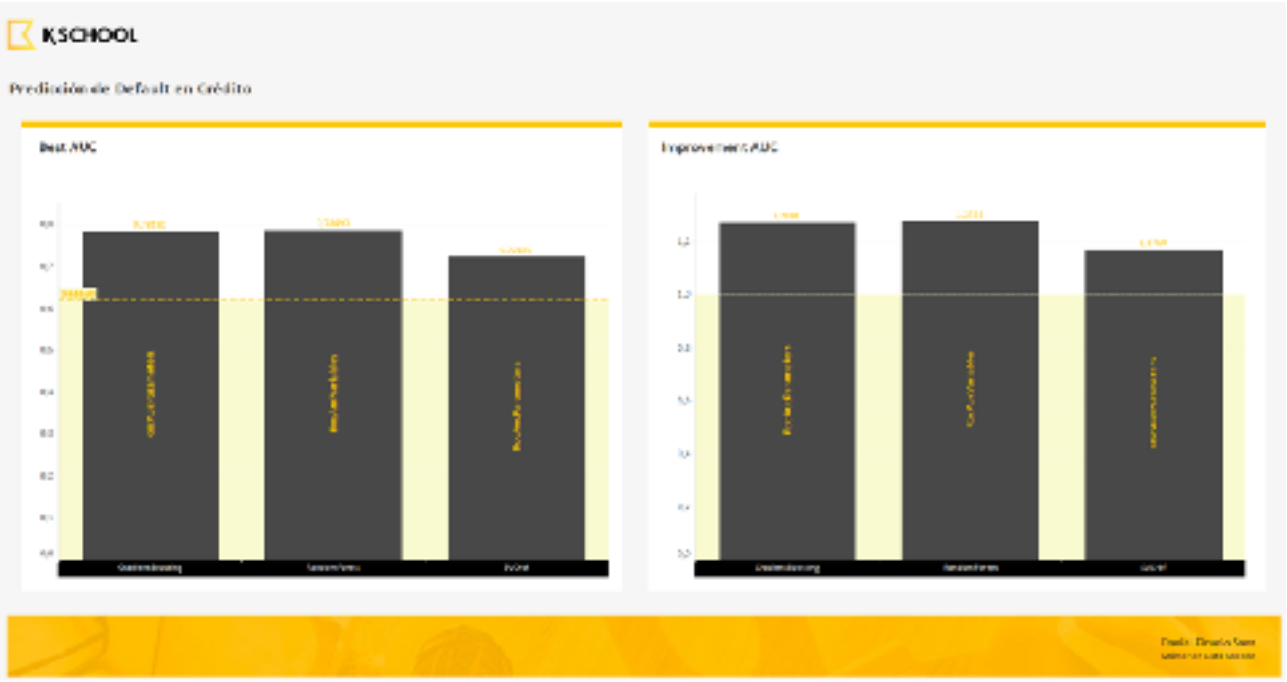
1. Entrenamiento en bruto.
2. Prueba del algoritmos con datos normalizados.
3. Cross validation entre 2 y 20 particiones para buscar la más óptima,
4. Búsqueda de los parámetros óptimos para el algoritmo.
5. Selección de variables.

La evaluación se realiza por el Área bajo la curva ron o AUC. Por qué hemos elegido esta medida, porque corresponde a la razón entre la tasa de True Positives y los False Positive de la clasificación. Los True Positive son los default bien clasificados, cuantos más default bien clasificados menos lo son de forma errónea, menos fraude. Los False Positive son los registros que no eran default pero se han clasificado como tal, son tarjetas que se han dejado de dar erróneamente, menos beneficio. Para haber sido totalmente precisos deberíamos haber ponderado la tasa de True Negative con mayor peso que la tasa de False Positive. Por qué, porque es mucho más costoso conceder una tarjeta y que sea default que contratar una nueva tarjeta. Al carecer de los datos de tipo de interés de cada tarjeta, historial de pagos, etc, no me he atrevido.

3 . Visualización de los resultados.

La visualización de los datos es muy sencilla es una comparativa entre la situación anterior del sistema antes de sus automatización y los resultados obtenidos en cada uno de los algoritmos.

Para ello hemos realizado un Dashboard en Tableau.



Es un dashboard que tiene dos gráficas. En una compara el mejor AUC de cada uno de los algoritmos con el AUC del sistema sin automatización, que es de 0.61. En la otra muestra la mejora porcentual del AUC de cada algoritmo contra el del sistema sin automatizar.

Posea una acción de filtrado. Si pinchamos sobre la barra de un algoritmo nos muestra una gráfica con las medidas de AUC de cada algoritmos en cada fase de la puesta a punto de cada algoritmo.