Santander Customer Transaction Prediction

Coder House

Integrantes:
Breitman, Zarina Madelaine
Marinella, Santiago
Ramos, Mateo
Navarro Quantín, Denise

Fecha de presentación: 27/11/2022

Data Science
Comisión 29780

Profesor: Miguel Magaña Fuentes

Tabla de contenidos

1. Introducción

El banco Santander busca comprender mejor a sus clientes, de forma tal que pueda ofrecerles los productos que les permitan alcanzar sus metas financieras. Es por ello que con este dataset se busca predecir cuáles son los clientes que van a realizar una determinada transacción en el futuro, independientemente del monto de dicha transacción.

Se trata de un problema de clasificación, donde se busca predecir la variable target que es dicotómica: dicha variable será “0” si el cliente no realizó la operación y “1” si efectivamente la realizó. Para poder hacer la predicción, se llevará adelante un proyecto donde se utilizan modelos de Machine Learning.

Los datos con los que trabajamos provienen de un dataset del banco Santander y fueron tomados del siguiente link de Kaggle:

https://www.kaggle.com/datasets/lakshmi25npathi/santander-customer-transaction-prediction-dataset

2. EDA (Exploratory Data Analysis)

El dataset fue extraído de Kaggle y cuenta con 200,000 registros y 201 variables. Una de ellas es la variable “Target” que, como mencionamos anteriormente, indica si el cliente ha realizado o no la transacción según adopte el valor “1” o “0” respectivamente. Las 200 variables restantes son anónimas y se encuentran encriptadas en el dataset original. Cabe destacar que, aunque no tengamos interpretabilidad de las variables, de igual modo se podrá confeccionar un algoritmo que permita predecir la variable target.

Realizando el EDA descubrimos que el dataset se encuentra preprocesado, esto se puede observar en el hecho de que no posee valores nulos, al mismo tiempo de que la distribución de las variables tiende a ser normal en todos los casos (salvo en la “Target”). A su vez, todas las variables son independientes entre sí, por lo que no hemos encontrado ninguna variable redundante para eliminar del dataset. Asimismo, no es posible descartar ninguna de las variables de forma objetiva puesto que no se tiene referencia de a qué dato corresponde cada una de ellas. Por otro lado, la presencia de outliers no es significativa; con lo cual tampoco se han podido descartar variables por esta causa.

Ante el anonimato de las variables y la inexistencia de correlación entre las mismas, no es posible realizar un análisis univariado, bivariado o multivariado que vaya más allá de lo que comentamos en el párrafo anterior y que nos sea de utilidad.

Por último, en esta etapa hemos reducido el uso de memoria creando un dataset más liviano que hará más rápido el trabajo de los modelos, disminuyendo así los costos computacionales.

3. Modelos

3.1 Generalidades

Utilizaremos como modelo base a la Regresión Logística, que se trata de una técnica de regresión aplicada a un problema de clasificación. Luego compararemos los resultados de la Regresión Logística con los que se obtengan con los otros modelos, de forma tal que podamos evaluar cuál es que resulta más conveniente. Para ello precisaremos observar una mejora significativa, teniendo en cuenta tanto las métricas como el tiempo de ejecución.

El dataset con el que trabajamos es muy asimétrico ya que la variable target tiene una distribución donde el 90% de los registros son "0" y tan solo el 10% de los registros son "1". Es por ello que utilizaremos Stratified K Fold con 10 folds para mantener en cada partición de datos la misma proporción de la variable target que en el dataset original. A su vez, emplearemos un loop que nos permitirá entrenar a cada fold y obtener sus métricas. Finalmente, buscaremos la media geométrica de las métricas de los 10 folds, de forma tal que obtengamos las métricas finales.

En lo que respecta a las métricas utilizadas, para medir la relación entre aciertos y errores optamos por utilizar Precision, F1 y Recall; siendo esta última la métrica principal. Con ella se puede observar la cantidad que se predice como verdaderos positivos sobre el total de los positivos. La hemos elegido como métrica principal ya que por la investigación que se realizó de la data se muestre que el 90% de los clientes no realizará la transacción entonces es importante evaluar con mayor peso ese 10 % restante.

Por otro lado, la métrica Precision mide la cantidad de verdaderos positivos sobre el total de los que se predicen como positivos (algunos en realidad son positivos y otros son negativos) y la métrica F1 mide que es una combinación entre Precisión y Recall.

Por otra parte, se utilizó la matriz de confusión para visualizar de forma rápida los resultados obtenidos por cada uno de los modelos planteados y poder apreciar de forma general la performance de los modelos.

Cabe aclarar que para este proyecto no empleamos la métrica accuracy ya que, como mencionamos anteriormente, el dataset no se encuentra balanceado.

Sin embargo, las métricas no son el único factor importante, ya que a la hora de elegir un modelo es fundamental tener en consideración el tiempo que tarda en ejecutarse. Es por ello que para cada modelo también tendremos en cuenta esta variable, que será de vital importancia a la hora de seleccionar el mejor modelo.

3.2 Proceso de trabajo

Para realizar el Proyecto hemos dividido el trabajo en una fase individual y otra grupal. El objetivo de la fase de trabajo individual fue explorar los distintos modelos, de forma tal que pudiésemos ver cuáles eran los que tenían mejor rendimiento. De este modo, se puede observar en el repositorio que cada miembro del equipo tiene una notebook distinta en esta etapa. Finalizada esa fase, hemos continuado el proyecto en una notebook común, donde solamente se han volcado los modelos más relevantes para el Proyecto.

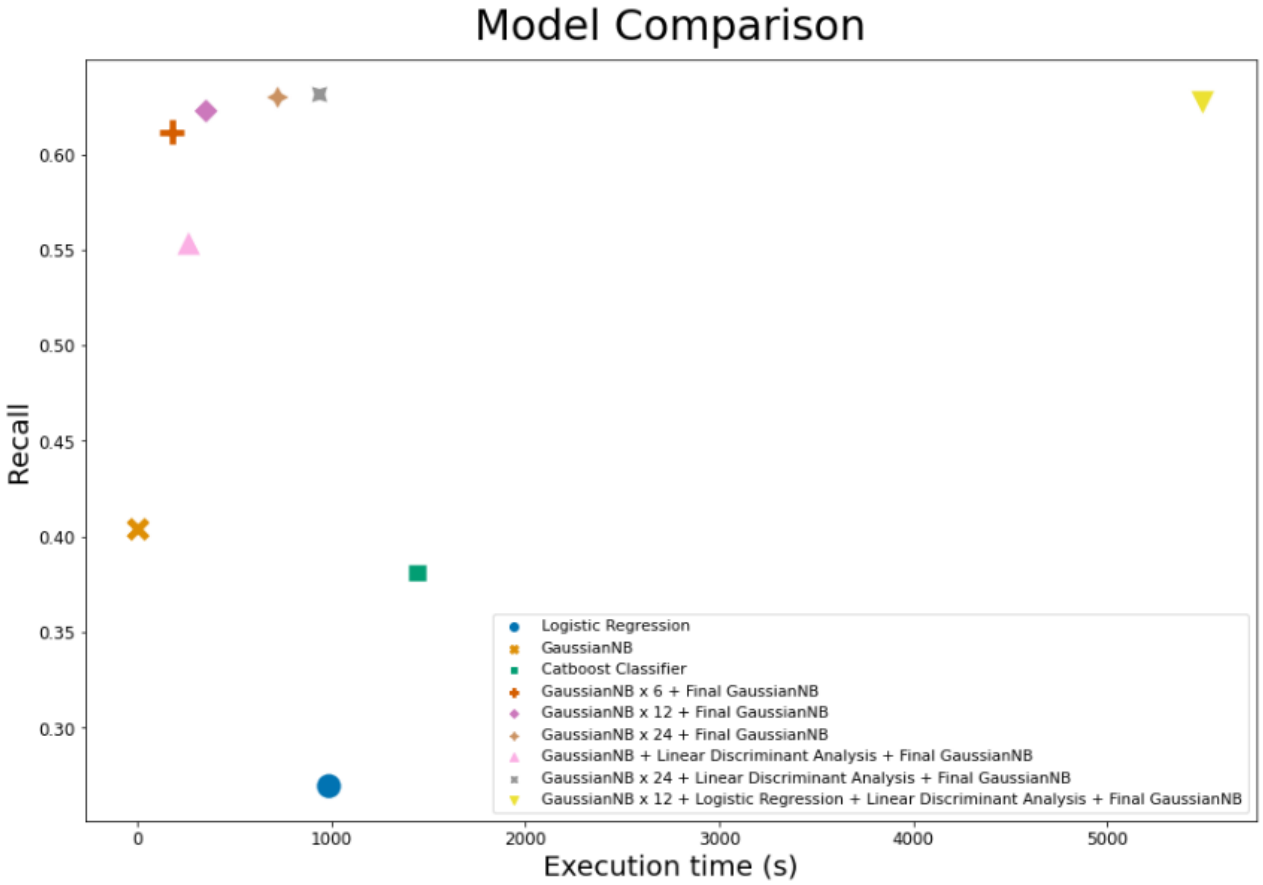
En lo que respecta a la fase de trabajo individual, el primer paso ha sido analizar distintos modelos de clasificación con los hiperparámetros que vienen por default, de forma tal que pudimos ver cuáles son los que brindan mejores métricas, siempre teniendo en cuenta que a su vez mantuviesen un equilibrio con el costo del modelo.

Luego hemos realizado una reducción de dimensionalidad utilizando PCA en algunos modelos. El PCA resulta de mucha utilidad en la presencia de datasets con gran cantidad de variables como este ya que permite disminuir el costo de ejecución de los modelos. Es por ello que resulta interesante observar de qué manera se comportan algunos modelos ante la reducción de dimensionalidad.

Posteriormente se ha utilizado la librería Optuna para hallar los hiperparámetros de los modelos que nos permiten obtener mejores resultados. Debido a que el dataset con el que trabajamos resulta muy grande, hemos trabajado en Optuna con reducción de dimensionalidad (utilizando PCA), de forma tal que pudimos disminuir los tiempos de ejecución de la optimización. En algunos modelos tales como Catboost Classifier y K Neighbors Classifier ha resultado viable extrapolar los hiperparámetros optimizados en menor cantidad de dimensiones al full dataset. Estos hiperparámetros fueron luego utilizados en la etapa de stacking de modelos para mejorar su rendimiento. Sin embargo, en la Regresión Logística no hemos podido utilizar en el full dataset los resultados de la optimización realizada con un dataset con menor dimensionalidad. Es por ello que con este último modelo se ha optimizado también el full dataset con Optuna, lamentablemente sin conseguir mejores resultados que con el modelo default.

Finalmente hemos realizado un stacking de modelos de forma tal que obtuvimos como resultado varios modelos de ensamble superiores a los modelos básicos.

La siguiente etapa es la del trabajo grupal. En la notebook común se encuentran los modelos básicos que resultaron más relevantes: Logistic Regression, Gaussian Naive Bayes y Catboost Classifier. A su vez, se incluyen los stacking de modelos con los que obtuvimos mejores resultados. Cabe aclarar que en ese punto no solo se buscó un buen Recall, sino que también que el tiempo de ejecución sea acorde. Posteriormente se graficó el resultado de los modelos, de forma tal que pudiésemos compararlos. Dicho gráfico es el que se encuentra a continuación.



4. Conclusión

El modelo base con el que se comenzó el proyecto es la Regresión Logística con un Recall de 0.269483 y un tiempo de ejecución de 988.74 segundos. A través del stacking de modelos hemos creado modelos de ensamble que alcanzan valores de recall superiores a nuestro modelo inicial.

El modelo de ensamble que mejor identificó si se harán o no transacciones, es GaussianNB x 24 + Linear Discriminant Analysis + Final Estimator GaussianNB que obtuvo un recall de 0.631196 con un tiempo de ejecución de 940.58 segundos. Es decir, un incremento de 134% de la métrica recall en comparación con el modelo base y una disminución del 5% del tiempo de ejecución. Teniendo en cuenta que el dataset cuenta con 200,000 registros, donde un 10% son personas que han realizado una transacción, con la Regresión Logística podíamos predecir 5,390 transacciones. En tanto que con el nuevo modelo, se pueden predecir 12,623 transacciones. Lo que significa que con el modelo de ensamble podemos predecir de forma eficiente 7,233 clientes más que realizarán una operación.

Sin embargo, GaussianNB x 6 + Final Estimator GaussianNB obtuvo un recall solamente 3% inferior al modelo de ensamble anterior, con 0.611569 y con un tiempo de ejecución de 178.87 segundos. Dicho de otro modo, 500% más veloz y en consecuencia con un menor costo computacional. Lo que representa una mejora de aciertos superior al 125% en comparación con el modelo base, alcanzando 6,842 más predicciones correctas para este set de datos.

En vista de esta situación dividimos nuestra recomendación en dos:

Si se desea realizar la mayor cantidad de detecciones, sin importar el costo computacional, utilizar el primer ensamble sería lo correcto. Sin embargo, observando el costo computacional, consideramos más conveniente emplear GaussianNB x 6 + Final Estimator GaussianNB como modelo final.

El análisis realizado con el set de datos obtenido, nos ha permitido crear un modelo de ensamble que prediga de manera correcta un poco más del 60% de las transacciones. Lo que representa una mejora de aciertos superior al 125% en comparación con el modelo base, alcanzando un total de 12,231 aciertos sobre las 20,000 transacciones en el set de datos.

5. Futuras líneas

Cabe destacar que, si bien hemos mejorado de forma significativa el resultado con respecto al modelo base, lo cierto es que este trabajo es un primer acercamiento al problema. Consideramos que para incrementar aún más el Recall a un tiempo de ejecución conveniente, sería necesario un mayor poder de procesamiento que nos permitiese realizar un análisis con una profundidad superior. Con mayor poder computacional se podría realizar optimizaciones más robustas y en mayor cantidad, al mismo tiempo que se podría explorar con modelos más complejos tales como las redes neuronales.