Centro de Investigación en Matemáticas (CIMAT). Unidad Monterrey

# Tópicos selectos de Cómputo. Proyecto Final: Clases desbalanceadas y selección de características.

Victor Manuel Gómez Espinosa

9 de diciembre de 2020

Para este proyecto se utilizó un conjunto de datos de una campaña de marketing de un banco y se intenta predecir si los clientes contratarán un producto, por lo cual se tienen dos clases (*yes, no*), donde la clase de interés es desbalanceada con una tasa de desbalance de 7.55, esto quiere decir que aproximadamente por cada 7.5 observaciones de la clase *no*, tenemos 1 de la clase *yes* (Figura 1).

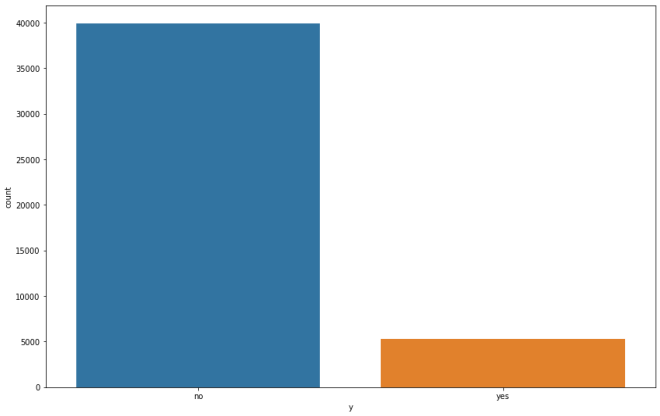


Figura 1: Clase de interés (*yes*) desbalanceada. Tasa de desbalance: 7.55.

Sobre este conjunto de datos hay trabajos previos (Moro et al. 2011), donde el objetivo de predecir la clase de interés es conocer cuáles son las características más importantes para la clasificación y de esta forma enfocar los recursos humanos y económicos en estas áreas para incrementar el éxito de futuras campañas.

El conjunto original cuenta con 45211 observaciones y 16 variables relacionadas con los usuarios, como edad, trabajo, estado marital, educación, datos de campañas previas y relacionadas a su cuenta bancaria (crédito, prestamos, etc.).

Dentro de estas variables en trabajos previos se determinaron 5 variables más importantes para la clasificación, entre ellas la duración de la llamada y el mes del contacto (marzo), resultado previo de las campañas entre otras, sin embargo, no se menciona para nada como se manejó el problema de clases desbalanceadas lo cual podría conducir a sobre estimar la importancia de alguna de las variables o no considerar algunas otras.

Para resolver el problema anterior se realizó lo siguiente, primero se revisó el conjunto de datos y se obtuvieron las variables dummies para las variables categóricas, removiendo una variable resultante de cada una de estas, principalmente las que no nos dan una interpretación adecuada (por ejemplo, las que tienen la clase unknow), posterior a esto se dividió el conjunto en uno de entrenamiento y uno de prueba (80% y 20% respectivamente).

Posteriormente se eligió un modelo de ML para utilizarlo como baseline y las métricas a utilizar para este problema.

Para el primer caso, dados los objetivos de este problema, se prefirieron los modelos de ML basados en arboles dado que tienen la ventaja que permiten de forma muy fácil conocer cuáles fueron las características más importantes para el clasificador, por lo que se consideró el modelo XGBoost ya que es muy rápido y generalmente entrega buenos resultados, esto permite poder hacer rápidamente varias pruebas (a diferencia de máquinas de soporte vectorial por ejemplo), adicionalmente tiene la ventaja que permite trabajar muy fácil con valores faltantes o con clases desbalanceadas variando la tasa de desbalance como hiper parámetro y otros modelos no poseen estas ventajas.

Para el segundo caso se eligió el AUC como el criterio principal, debido que la categoría de interés (yes) esta desbalanceada. Inicialmente se probó con el Recall, pero a pesar de maximizar este para la categoría de interés se pierde balance con la categoría que no es de interés (además de Precisión) y se consideró que es importante tener un equilibrio entre estas para que el clasificador aprenda a distinguir

correctamente cuando pertenece o no a la clase de interés, ya que el objetivo de este problema es determinar las características más importantes para la clasificación.

En este caso como las clases no son balanceadas, no se puede utilizar la exactitud (o Accuracy) como métrica para evaluar los modelos, sin embargo, en estos casos podemos utilizar alternativamente el Balanced Accuracy que es el promedio del Recall de las clases.

Otra medida de utilidad es el F1 score, el cual resume la información del Recall y la Precisión en un solo número para la clase de interés, esta medida ayuda a mitigar el efecto de tasas altas, por ejemplo en el caso de que se tenga un Recall muy alto de la clase de interés significará que hay una Precisión muy baja (recordando que la Precisión nos ayuda a cuantificar que tan efectivo es el modelo en detectar la clase de interés, mientras que el Recall a cuantificar que tan bien fueron clasificados los verdaderos de la categoría de interés, por lo cual un balance entre ellos es importante).

Una vez que se seleccionó lo anterior, se seleccionaron los métodos para manejo de clases desbalanceadas, estos son:

- Métodos a nivel de algoritmo: XGB y SVM con pesos.
- Métodos a nivel de datos: Over-sampling (SMOTE), Under-sampling (ENN), métodos híbridos (SMOTE+ENN).
- Métodos basados en preprocesamiento de datos y ensamble: RusBoost con AdaBoost como base.

Una vez seleccionado esto, se pasó a ajustar los modelos mediante búsqueda aleatoria con validación cruzada 5-fold y métrica ROC-AUC para encontrar los mejores modelos y obtener métricas adicionales sobre el conjunto de prueba (AUC, Recall, F1, matriz de confusión).

Observe la Figura 2 y note que para este problema los métodos de manejo de clases desbalanceadas según el criterio del AUC, tienen resultados muy similares excepto 2, el basado en Ensamble (AdaBoost-RUSBoost) y el de a nivel de algoritmo (SVM con pesos).

Para poder diferenciar cual podría ser el mejor, se utilizaron las demás métricas, principalmente el Balanced Accuracy, observe la Figura 3 y note que los 3 mejores en base a este criterio (y con resultados muy similares) son el basado a nivel de datos

hibrido (SMOTE+ENN), a nivel de algoritmo (XGB con pesos) y el basado en preprocesamiento y ensamble (RusBoost).

Dentro de estos 3, se podría decir que el mejor, basado en el Recall y F1, además de rapidez, facilidad de uso y para obtener características, es el método a nivel de algoritmo, para el caso particular del modelo XGBoost con pesos.

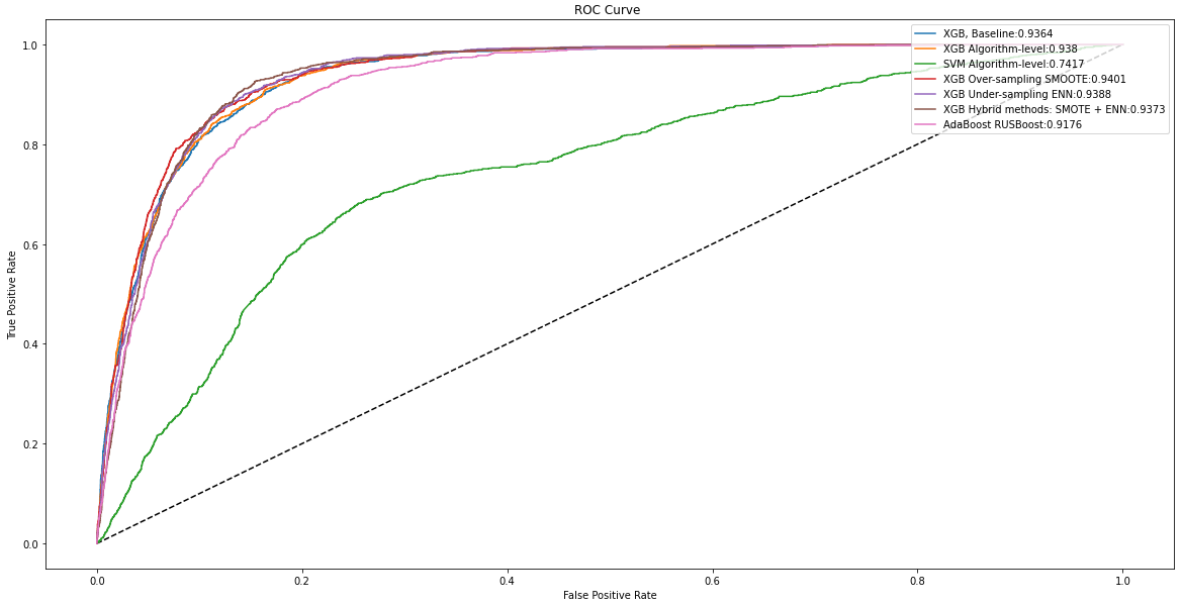


Figura 2: Resultados ROC Curve, AUC.

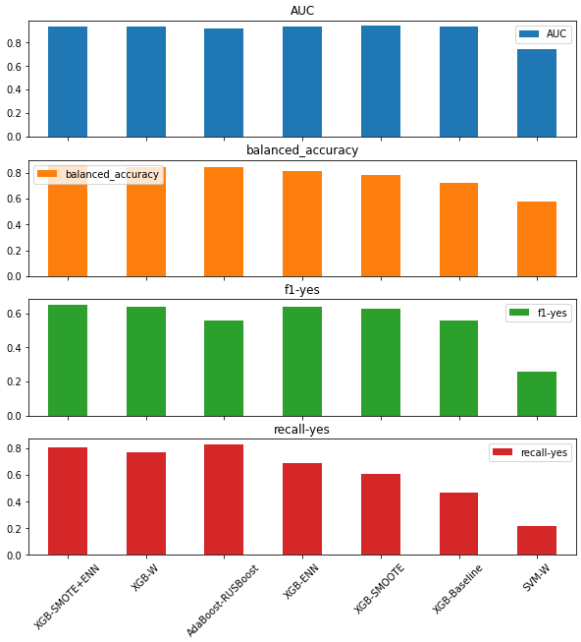


Figura 3: Resultados, métricas: AUC, F1-yes, Recall-yes.

Por otra parte, el método hibrido mostró mejor desempeño que por separado, esto tiene sentido si recordamos que el método SMOTE genera muestras artificiales mediante interpolación con los vecinos más cercanos para balancear el conjunto de datos, sin embargo, tiene la desventaja que puede inducir ruido, pero combinado con el método ENN (que justamente es para manejo de ruido), sirve para filtrar estas estancias con ruido y dejar el data set mejor balanceado.

En el caso de RusBoost que utiliza el método a nivel de datos RUS para seleccionar aleatoriamente muestras sin remplazo para balancear el conjunto de datos, y posteriormente utilizar un modelo de ensamble, en este caso AdaBoost y nuevamente vemos muy buenos resultados, aunque se pierde Precisión a diferencia del modelo XGBoost con pesos (Figura 4).

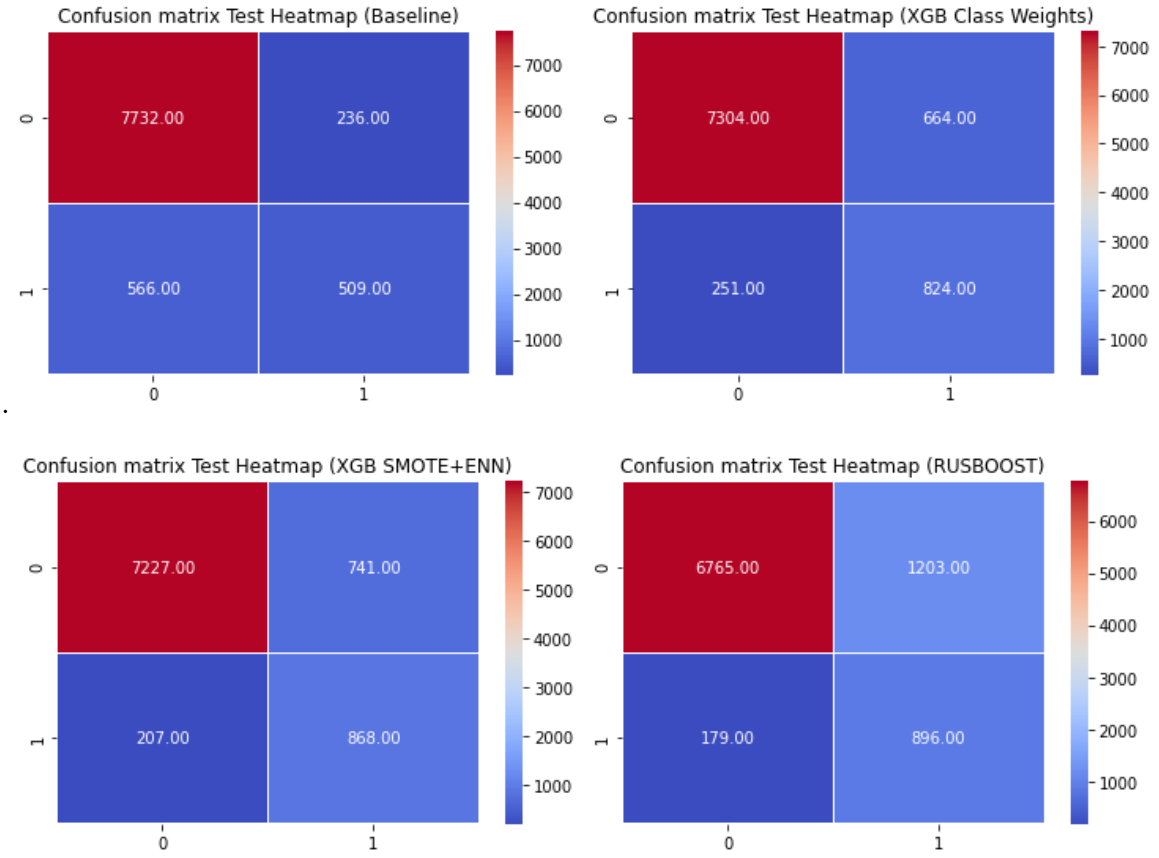


Figura 4: Matrices de confusión para el Baseline y los 3 mejores métodos.

Selección de características:

Finalmente, para el mejor modelo (XGBoost con pesos), se determinaron sus características más importantes, bajo el criterio del *Gain,* el cual nos da una puntuación que podemos utilizar para cuantificar la calidad de la estructura de un árbol (similar al modelo CART con el índice Gini).

$$Gain = \frac{1}{2}\left[\left(\frac{G_L^2}{H_L+\lambda}+\frac{G_R^2}{H_R+\lambda}\right)-\left(\frac{\left(G_L+G_R\right)^2}{\left(H_L+H_R\right)+\lambda}\right)\right]-\gamma$$

Donde $G$ representa el gradiente (suma de residuales), $H$ el hessiano (puede ser el número de residuales), y $\lambda, \gamma$ son parámetros de regularización.

Dentro de los corchetes de la expresión anterior, la parte izquierda representa las hojas (izquierda y derecha) y la parte derecha la raíz. La expresión para cada hoja

se podrían ver como similaridades por lo cual se busca que el corte maximice estas similaridades para cada hoja (Figura 5), que a su vez maximizaría el *Gain,* por el contrario, residuales muy diferentes lo minimiza y mediante este criterio se busca la mejor estructura del árbol.

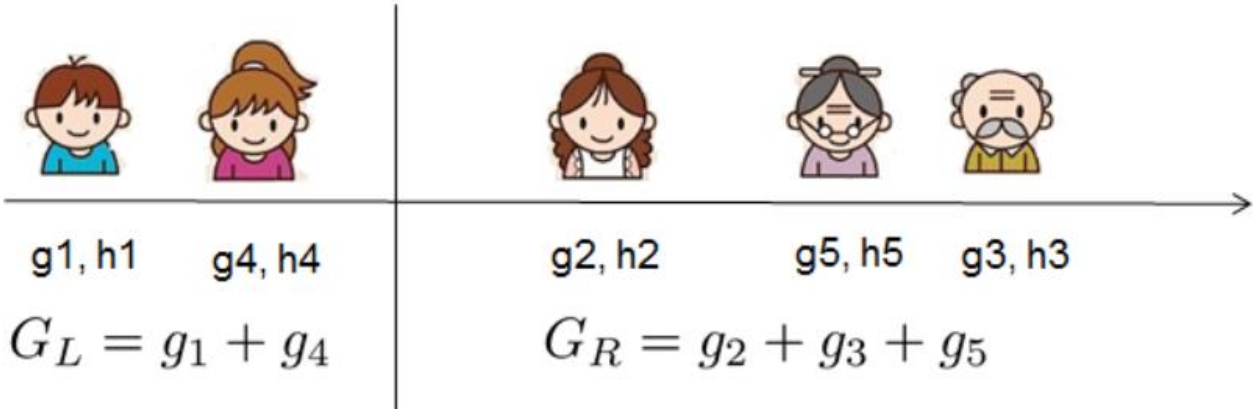


Figura 5: Ejemplo de corte que maximiza el *Gain* (Chen and Guestrin 2016)

Observe la Figura 6 y note que algunas de estas concuerdan con los resultados de trabajos previos, es decir, la duración de la llamada y el mes de contacto en marzo, sin embargo, la más importante fue el resultado exitoso en campañas previas además de otras como préstamo de vivienda y si el tipo de contacto fue por celular.

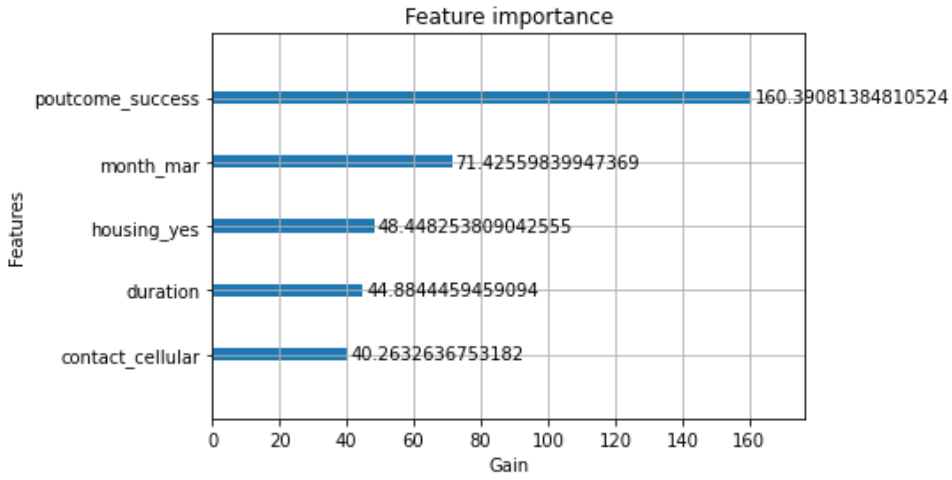


Figura 6: Características más importantes para el modelo XGBoost con pesos.

Conclusiones:

Cuando se trata de clases desbalanceadas las métricas de AUC, Balanced Accuracy y el F1 son de mucha utilidad a la hora de medir el desempeño de los modelos.

En los trabajos previos el mejor modelo resultó el de máquinas de soporte vectorial con kernel gaussiano, el cual fue el mismo tipo de modelo que se intentó ajustar con pesos para la clase de interés, sin embargo, los resultados obtenidos (AUC=0.579) fueron muy por debajo de los mostrados en trabajos previos (AUC=0.93).

Para este problema, el método a nivel de algoritmo fue el mejor, para el caso particular del modelo XGBoost con pesos, considerando el AUC, Balanced Accuracy, F1 y su velocidad, sin embargo, como ya se mencionó, no sucedió lo mismo para el caso de máquinas de soporte vectorial.

Los métodos híbridos, así como los métodos de preprocesamiento y ensamble también mostraron muy buenos resultados, y tomando ambos criterios el AUC y el F1, el método hibrido mostró ser mejor, debido a que la Precisión del método hibrido resultó mejor para este problema particular.

Adicional a lo anterior, también se le suma que estos métodos también hacen un poco más tardado el entrenamiento de los modelos, a diferencia de los que son a nivel de algoritmo lo cual en conjuntos con gran volumen de datos podría ser un gran problema.

Por último, las características más importantes obtenidas del mejor modelo brindan gran información para los objetivos originales de este problema, ya que de alguna forma permiten saber en qué segmento de usuarios concentrarse, cuando hacer la campaña y por qué medio contactar a los usuarios para maximizar las probabilidades de éxito y disminuir quizá tiempo, dinero y recursos humanos. Lo cual tiene sentido, ya que responde las preguntas: ¿a quién?, ¿cuándo?, ¿Cómo?

Lo anterior quiere decir que quizá se deberían concentrar principalmente en usuarios que en campañas previas aceptaron contratar un producto y que adicionalmente cuentan con un préstamo de vivienda (y tienen celular), además de que en el mes de marzo podría ser el mejor momento para realizar una campaña para este producto, contactarlos por medio del celular y tratar que la llamada dure lo más posible.

Referencias:

Chen, T., and Guestrin, C. (2016), "XGBoost: A scalable tree boosting system," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 13-17-Augu, 785–794. https://doi.org/10.1145/2939672.2939785.

Moro, S., Laureano, R. M. S., and Cortez, P. (2011), "Using data mining for bank direct marketing: An application of the CRISP-DM methodology," *ESM 2011 - 2011 European Simulation and Modelling Conference: Modelling and Simulation 2011*, 117–121.