Inteligencia Artificial y Aprendizaje Automático

Actividad Semanas 5 y 6: Riesgo Crediticio

Maestría en Inteligencia Artificial Aplicada — Tecnológico de Monterrey
Prof. Luis Eduardo Falcón Morales

Nombres: _____ Matrículas: _____

Esta Tarea deberá resolverse en equipos.

Esta actividad se complementa con el archivo "**MNA_IAyAA_semanas_5_y_6_Actividad_2025.ipynb**" que se encuentra en Canvas y donde deberán ir respondiendo los ejercicios. Al final deberán entregar la liga de GitHub donde se encuentra el archivo JupyterNotebook con las respuestas y el nombre de los dos miembros del equipo.

El asignar un crédito conlleva un riesgo para el prestamista en caso de que el deudor no pague al final la cantidad asignada, o bien, al equivocarnos en negarle el préstamo a alguien que sí era confiable. Durante décadas se ha tratado de resolver dicho problema desde muchas áreas del conocimiento y en particular las técnicas de Aprendizaje Automático (Machine Learning) han brindado y siguen proporcionando nuevas formas de enfrentar estos problemas.

Existen pocas bases de datos abiertas bien documentadas sobre este problema. Una de ellas son los datos de la página de la UCI llamada "South_German_Credit" y sobre la cual se ha hecho mucha investigación en torno a la asignación de créditos. En esta actividad trabajarás con los datos del archivo "**SouthGermanCredit.asc**", el cual se encuentra dentro del archivo **south+german+credit.zip** que puedes descargar de la liga : https://archive.ics.uci.edu/dataset/522/south+german+credit

En esta actividad nos proponemos tratar de obtener los mejores resultados que se reportan en la Tabla_12 del artículo de la IEEE y cuya liga de acceso está en Canvas.

En este artículo se comenta que los datos que utilizaron no son exactamente que los que se encuentran en la UCI. Los modificaron un poco por cuestiones de privacidad. Sin embargo, por la selección aleatoria se esperan obtener resultados análogos en la actividad de esta semana.

Como comentario sobre los datos que utilizaremos, estos datos llamados "South-Geraman-Credit" son una actualización mejorada de otros previos que se estuvieron usando durante décadas para investigación, pero como estaban en idioma alemán, no se habían percatado de varios errores que se habían generado al codificar las variables.

1. Descarga los datos, los cuales nos llevan a un arreglo de 1000 registros y 21 variables. Cambia los títulos de las columnas al nombre en inglés (originalmente están en alemán). La información la puedes encontrar en cualquiera de las ligas dadas arriba.
2. Aplica alguna transformación de manera que la clase negativa (mayoritaria) de los buenos clientes estén etiquetados con el valor de 0 y los malos clientes o clase positiva (minoritaria), estén etiquetados con el valor de 1.
3. El propósito de esta actividad es obtener un modelo con resultados que puedan ser lo más cercanos a los del artículo de la IEEE. Así que haremos una partición solamente en dos partes: el conjunto de Entrenamiento (Train) y el de Prueba (Test), con las mismas características que se siguieren en el artículo:

a. Para ello realiza una partición en Entrenamiento y Prueba con los mismos porcentajes que se utilizaron en el artículo. Llama a las variables Xtrain, ytrain, Xtest, ytest.

b. Con base al porcentaje de los niveles de la variable de salida ¿podemos decir que tenemos un problema de datos desbalanceado? ¿Por qué?

c. ¿Por qué se hizo el cambio de etiquetas 0 y 1 en la variable de salida?

4. Utiliza la información de la Tabla_3 del artículo para identificar y definir las variables de entrada en numéricas (quantitative), ordinales (ordinal) y las nominales (categorical, binary).

5. Utiliza las clases Pipeline() y ColumnTransformer() de Sklearn para definir y conjuntar las siguientes transformaciones (pueden ser diferentes a las aplicadas en el artículo de la IEEE, si consideras que te dan mejor resultado):

   a. A las variables numéricas aplica la misma transformación que mejor consideres.

      NOTA: En el artículo de la IEEE se comenta que se aplicó la transformación de normalización que se indica en el mismo artículo.

   b. A las variables nominales aplica la transformación que mejor consideres.

   c. A las variables ordinales aplica la transformación que mejor consideres.

6. Como vamos a utilizar validación cruzada, concatena los conjuntos de entrenamiento y validación en un nuevo conjunto llamado traintest, que tendrá el mismo número de columnas, pero con el total de renglones la suma de ambos.

7. En este ejercicio deberás buscar la mejor configuración con el mejor desempeño posible para cada modelo que se indica a continuación. Igualmente, en cada caso deberás determinar si ayuda el aplicar alguna técnica de submuestreo y/o sobremuestreo. Revisa el artículo de la IEEE para que revisar las técnicas que ellos aplicaron y que puedan darte una idea de cuáles probar; pero no necesariamente tienes que limitarte a ellas.

   NOTA: Puedes consultar la siguiente liga para las diferentes técnicas de submuestreo y/o subremuestreo: https://imbalanced-learn.org/stable/references/over_sampling.html
   NOTA: Recuerda que obtener el mejor desempeño significa que no esté sobreentrenado o subentrenado el modelo. En particular diremos que no está sobreentrenado un modelo si la diferencia entre Train y Test de cierta métrica es menor al 4% y mientras más pequeña esta diferencia, mejor. ¿Cuál métrica consideras que debiera usarse en este problema?

   a) Regresión logística

   b) K vecinos más cercanos

   c) Árbol de decisiones

   d) Bosque aleatorio

   e) XGBoost

   f) Red neuronal multicapa

   g) Máquina de vector soporte

8. En la Tabla_12 del artículo de la IEEE se muestran los mejores resultados que encontraron los autores de dicho artículo para el caso de los datos South-German. Observa que cada columna de dicha Tabla 12 nos está dando los desempeños de los modelos utilizados con las métricas indicadas, a saber: Accuracy, Precision, Recall, F1, ROC-AUC, G-mean. De todos los modelos que

aplicaste (Logística, kNN, DecisionTree, RandomForest, XGBoost, MLP, SVM) selecciona el que consideres fue la mejor configuración obtenida. Reporta tu mejor resultado a continuación:

Mejor modelo y configuración encontrada:

- Modelo y valores de hiperparámetros utilizados: RandomForest,
- Técnica de submuestreo y/o sobremuestreo utilizada (en caso de que se haya utilizado):SOMTE
- Reporte de las métricas obtenidas:

Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	ROC_AUC	G-mean
79.73	79.21	44.03	56.52	86.08	64.64

9. Incluye tus conclusiones finales de la actividad.

    - hiperparámetros utilizados: Bootstrap=True, criterion='entropy', max_depth=15,max_features='sqrt', min_samples_leaf=20, min_samples_split=10, n_estimators=100, random_state=1, n_jobs=-1