Inteligencia Artificial y Aprendizaje Automático
Actividad Semana 3:
Regresión Logística y Problemas de Clasificación

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

Nombre: _____Matrículas: _____

Esta Tarea se deberá resolver de manera individual e incluye los temas que estarás estudiando en la semana 3 del curso. Deberás generar un archivo de Jupyter-Notebook con los análisis y comentarios que se te piden en cada uno de los ejercicios.

La rotación de personal es uno de los problemas que afecta actualmente a muchas empresas y organizaciones, grandes o pequeñas y de cualquier tipo de negocio. En esta actividad usaremos una base de datos generada por IBM para estudiar y tratar de enfrentar dicho problema mediante un modelo de regresión logística para un problema de clasificación. Deberás descargar el archivo de la siguiente liga de Kaggle, la cual consta de 1470 registros y 35 columnas:

https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset

PARTE 1: Análisis descriptivo y preprocesamiento de los datos:

1) Incluye una breve introducción sobre lo que se entiende por el problema de rotación de personal en las organizaciones (*employee attrition problem*).

2) Carga la base de datos y utiliza el método "describe" para un DataFrame de Pandas con el argumento include= "all". Usa además la traspuesta para desplegar todos los datos explícitamente.

3) ¿Cuál es la diferencia entre utilizar o no el argumento include="all" en el ejercicio anterior?

4) Con base a la información desplegada por la instrucción anterior, hemos decidido eliminar los siguientes factores de nuestro problema: *Over18*, *EmployeeCount*, *StandardHours* y *EmployeeNumber*. Elimina dichas columnas del DataFrame y explica cuál es la justificación que nos permite cancelar cada una de estas:
    a) *Over18*:
    b) *EmployeeCount*:
    c) *StandardHours*:
    d) *EmployeeNumber*:

De la documentación de esta base de datos se nos proporciona la siguiente información. En las variables categóricas el entero dentro de los paréntesis indica el total de niveles de dicha variable:

i. **variables numéricas (14):**
   NumCompaniesWorked,
   TrainingTimesLastYear,
   Age,
   DailyRate,
   DistanceFromHome,
   HourlyRate,
   MonthlyIncome,
   MonthlyRate,
   PercentSalaryHike,
   TotalWorkingYears,
   YearsAtCompany,
   YearsInCurrentRole,
   YearsSinceLastPromotion,
   YearsWithCurrManager

ii. **Variables ordinales (9):**
   Education(5),
   EnvironmentSatisfaction(4),
   JobInvolvement(4),
   JobLevel(5),
   JobSatisfaction(4),
   PerformanceRating(2),
   RelationshipSatisfaction(4),
   StockOptionLevel(4),
   WorkLifeBalance(4)

iii. **Variables binarias (3):**
   Attrition (variable de salida),
   Gender,
   OverTime

iv. **Variables nominales (5):**
   BusinessTravel(3),
   Department(3),
   EducationField(6),
   JobRole(9),
   MaritalStatus(3)

5) Realiza una partición de los datos en Entrenamiento, Validación y Prueba, del 70%, 15% y 15%, respectivamente. Llama a dichos conjuntos Xtrain, Xval, Xtest, ytrain, yval, ytest, para los datos de entrada y de salida, respectivamente. Asegúrate que dicha partición conserve la estratificación de las clases de la variable "Attrition". Despliega además la dimensión obtenida de los tres conjuntos: Entrenamiento, Validación y Prueba.

6) Aplica la transformación LabelEncoder() de sklearn a la variable de salida "Attrition", en los conjuntos de entrenamiento, validación y prueba, evitando el filtrado de información. Encuentra además la proporción de cada clase en el conjunto de entrenamiento. ¿Podemos decir que tenemos un problema de clasificación desbalanceado?

7) Incluye un análisis descriptivo y/o gráfico de las variables del conjunto de entrenamiento.

E incluye tus conclusiones en relación con la información importante que hayas encontrado.

8) Con base al análisis previo ahora deberás aplicar las transformaciones a las variables de entrada utilizando las clases Pipeline y ColumTransformer de Scikit-learn, de acuerdo a como se te indica a continuación:

   a) Incluir las transformaciones que consideres adecuadas para manejar datos perdidos en todas las variables de entrada (NOTA: independientemente de que los datos de entrenamiento analizados tengan o no valores perdidos, siempre debes estar preparado por si surge el caso en algunos de los conjuntos de validación o de prueba).
   b) Incluye las transformaciones que consideres adecuadas en las variables numéricas, ya sea para ajustar el rango de las variables y/o para ajustar la forma de su distribución. Justifica las transformaciones que decidas aplicar.
   c) Aplica la transformación OrdinalEncoder() de Scikit-learn a las variables categóricas ordinales de entrada.
   d) Aplica la transformación que consideres adecuada para las variables binarias de entrada. Justifica tu respuesta.
   e) Aplica la transformación OneHotEncoder() de Scikit-learn a las variables categóricas nominales de entrada.

PARTE 2: Entrenamiento de los modelos

9) Antes de continuar y como vamos a estar utilizando Validación Cruzada (Cross-Validation) durante el entrenamiento, reagrupa los conjuntos de entrenamiento y validación en un solo DataFrame. A este nuevo DataFrame llamarlo Xtv.

10) Con la métrica de exactitud ("accuracy"), busca los mejores hiperparámetros de los modelos de Regresión Logística (RL), RL con regularización L1 (o Lasso), RL con regularización L2 (o Ridge), RL con regularización L1 y L2 (o elastic-net) y k-vecinos-más-cercanos (kNN). Verifica que los modelos no queden sobre entrenados o subentrenados.
    NOTA: Recuerda que decimos que un modelo de clasificación no está sobreentrenado, si la diferencia en porcentaje del desempeño de un modelo entre el conjunto de entrenamiento y el de validación no es mayor al 3%.

11) Selecciona el modelo que consideres te da el mejor desempeño y realiza ahora una búsqueda de malla (gridSearch), para hacer una búsqueda más fina de los mejores hiperparámetros.

12) Con los mejores valores de hiperparámetros del mejor modelo encontrado:
   a) Obtener el valor de la exactitud (accuracy) y la matriz de confusión.
   b) Realiza un análisis de importancia de factores, indicando cuales son los factores que ayudan a explicar mejor la variabilidad de la variable de salida.
   c) Obtener un reporte del modelo obtenido con el método classification_report() de scikit-learn.

13) Obtener el desempeño final del modelo (accuracy) con el conjunto de prueba (test).
14) Incluye tus conclusiones finales de la actividad.