Mentoría Churn Prediction - Aprendizaje Supervisado

Input:

Archivo input editado del TP Nº2, luego de la “curación” de datos.

Entregable:

- Se puede ir desarrollando cada punto en la misma notebook donde se escriba el código.
- Se debe subir el entregable a un repositorio GitHub o enviar el link a un Google Colab.
- Tener en cuenta que si bien pueden realizar diversos análisis y visualizaciones, se debe dejar en el entregable sólo aquello que sea relevante.
- Luego de cada análisis es importante poder obtener una conclusión de lo observado.

1. **Preparación de los datos para aplicar modelos de clasificación**

a. Generar a partir del dataset, los conjuntos de train, test y validation.
    i. Analizar las proporciones consideradas para cada conjunto con respecto al total del dataset. Verificar el total de datos de cada conjunto.
    ii. Para cada conjunto (train, test, val), ¿cuántos datos de cada clase target hay (churn y no churn)?, se debe buscar mantener la proporción del dataset total (el desbalanceo que era casi de 3 a 1).

2. **Creación de un modelo baseline**

a. Entrenar un modelo “baseline”, es decir lo más simple posible, para con ello tener un punto de partida con el cual comparar modelos más complejos. (Hint: se puede usar por ejemplo la clase “DummyClassifier” de scikit-learn)

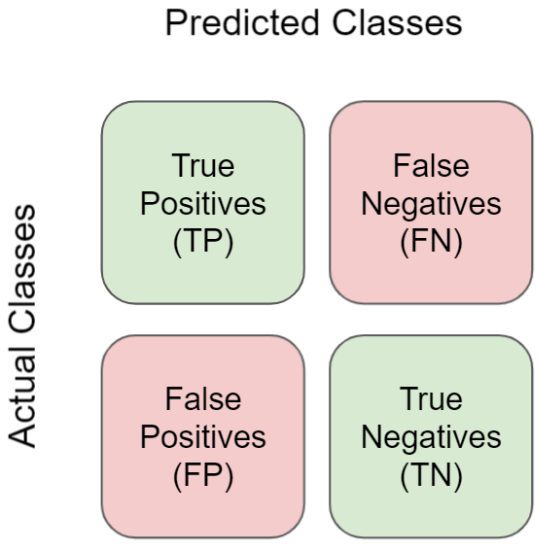
Documentación:
https://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyClassifier.html

Hint: fijar la semilla aleatoria para hacer repetible el experimento.

b. Evaluar sobre el conjunto de entrenamiento, validación y test reportando:
    a. Accuracy
    b. Precision
    c. Recall
    d. F1
    e. Matriz de confusión

c. Pensar cuál métrica es conveniente optimizar en éste problema de clasificación donde se predice churn. ¿Sería el Accuracy una buena métrica?

Reminder:



Measure	Fórmula
Accuracy	$\frac{TP+TN}{TP+FP+FN+TN}$
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$
F1 Score	$\frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$

3. **Predicción de modelos lineales**

a. Entrenar modelos lineales de clasificación para predecir la variable objetivo. Para ello, deberán utilizar “LogisticRegression” de scikit-learn.
Y elegir al menos uno de los siguientes otros modelosr:
   i. La clase SGDClassifier de scikit-learn.
   ii. La clase LinearSVC de scikit-learn.

   Documentación:

   https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
   https://scikit-learn.org/stable/modules/sgd.html
   https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html
   https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

   Hint: fijar la semilla aleatoria para hacer repetible el experimento.

b. Evaluar sobre el conjunto de entrenamiento, validación y test reportando:
   iii. Accuracy
   iv. Precision
   v. Recall
   vi. F1
   vii. Matriz de confusión

c. Elabore conclusiones en base a la métrica a optimizar y compare con el modelo baseline.

4. **Predicción de modelos basados en árboles de decisión**

a. Para ello, deberán elegir al menos dos de los siguientes modelos:
   i. La clase DecisionTreeClassifier de scikit-learn.
   ii. La clase RandomForestClassifier de scikit-learn.
   iii. La clase GradientBoostingClassifier de scikit-learn.

Documentación:

https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

Hint: fijar la semilla aleatoria para hacer repetible el experimento.

b. Evaluar sobre el conjunto de entrenamiento, validación y test reportando:
    i. Accuracy
    ii. Precision
    iii. Recall
    iv. F1
    v. Matriz de confusión

c. Elabore conclusiones en base a la métrica a optimizar, comparando los modelos elegidos y contrastando con el modelo lineal elegido.

5. Ajuste por hiperparámetros

a. Para los dos “mejores modelos” obtenidos en los puntos anteriores, seleccionar valores para los hiperparámetros principales de dichos modelos (ajustar con por lo menos 3 parámetros). Utilizar grid-search y k-fold cross-validation.
b. Mencionar el mejor modelo obtenido de la Optimización de Hiperparámetros y con cuáles parámetros se obtuvo ese resultado.
c. Con el mejor modelo obtenido realizar las predicciones sobre test y val.

d. Reportar las métricas del mejor modelo, incluyendo las matrices de confusión. Comparar el mejor modelo obtenido, con el modelo con parámetros por defecto y con el modelo baseline. Elabore conclusiones al respecto pensando en la resolución de nuestro problema de clasificación.

Deadline de entrega: 29/07/2022.