

Desafío - Predicción de renuncia de clientes

En este desafío validaremos nuestros conocimientos de redes neuronales. Para lograrlo, necesitarás aplicar un modelo de red neuronal feedforward con una capa oculta, aplicado a una base de datos de clientes de telecomunicaciones Iraní, con el objetivo de predecir si un cliente renunciará.

El conjunto de datos a cargar es: **Customer Churn.csv**.

Lee todo el documento antes de comenzar el desarrollo **individual**, para asegurarte de tener el máximo de puntaje y enfocar bien los esfuerzos.

Tiempo asociado: 3 horas cronológicas.

Descripción

Contamos con una base de datos de una compañía de telecomunicaciones, con la que la misma podrá tomar acciones dirigidas en la retención de clientes propensos a la fuga.

La base de datos cuenta con 12 atributos regresores más la etiqueta de clase, que son:

Call Failures: Número de fallos de llamadas

Complaints: Quejas (1) o ninguna queja (0)

Subscription Length: Meses totales de suscripción

Charge Amount: Importe del cargo, atributo ordinal (0: importe más bajo, 9: importe más alto)

Seconds of Use: Segundos totales de llamadas

Frequency of use: Número total de llamadas

Frequency of SMS: Número total de mensajes de texto

Distinct Called Numbers: Número total de llamadas distintas

Age Group: Grupo de edad, atributo ordinal (1: menor edad, 5: mayor edad)

Tariff Plan: Plan de tarifa, binario (1: pago por uso, 2: contractual)

Status: Estado, binario (1: activo, 2: inactivo)

Churn: Renuncia, binario (1: cancelación, 0: no cancelación) - **Etiqueta de clase**

Customer Value: El valor calculado del cliente

A partir de estos datos se te solicita lo siguiente:

1. Importa las librerías necesarias para aplicar preprocesamiento de datos, visualización y creación de un modelo de red neuronal feedforward con Keras y Tensorflow. Importa la base de datos **Customer Churn.csv** y realiza el preprocesamiento de los datos
2. Analiza la distribución de **"Subscription Length"** para cada clase (Churn), describiendo si hay alguna incidencia de esta variable en la renuncia o no. Realiza una estandarización de las variables regresoras
3. Muestra la frecuencia de cada clase y realiza un balanceo de clase usando SMOTE. Divide el conjunto de datos en entrenamiento y test, considerando para este último un 33%.
4. Entrene un modelo RandomForest con hiper parámetros por defecto, usando el conjunto de entrenamiento anterior y muestra la métrica F1-Score para ambas clases en el conjunto de test. Captura y muestra las siete características más importantes detectadas por el modelo RandomForest.
5. Usando las siete características encontradas anteriormente, filtra el conjunto de entrenamiento y entrena un modelo de red neuronal feedforward con sólo una capa oculta. Para esto debes definir una función de activación adecuada, tanto para la capa oculta como para la capa de salida, la cantidad de neuronas a usar, la tasa de aprendizaje (learning_rate), y la cantidad de épocas deben ser ajustadas manualmente. Como optimizador, utiliza el **Descenso del Gradiente Estocástico**. Los ajustes manuales de los hiper parámetros deben ser tales que la métrica AUC sea mayor al 90%. Muestra finalmente ROC y su AUC. **(3 Puntos)**

Requerimientos

1. Carga librerías y datos, y los prepara y analiza para construir modelos de redes neuronales. **(2 puntos)**
2. Identifica las características más importantes para un modelo de red neuronal, a través de un modelo Radom Forest. **(3 puntos)**
3. Implementa e interpreta las métricas de un modelo de red neuronal. **(5 puntos)**



¡Mucho éxito!

Consideraciones y recomendaciones

- Debes entregar tu trabajo en un archivo Jupyter, con tu código y las explicaciones correspondientes.