{desafío} latam_

Prueba 1: Análisis de Sentimientos de Twitter

A continuación se presenta un problema clásico en el análisis de texto: *Extraer el sentimiento asociado a un texto*.

Para esto, utilizaremos una base de datos provenientes de *CrowdFlower*.

Para descargar los datos puede ejecutar el siguiente código:

```
wget https://www.crowdflower.com/wp-content/uploads/2016/07/text_emotion.csv
```

El objetivo general de esta prueba es alcanzar el mejor desempeño posible para clasificar si un tweet es positivo o negativo.

Para medir el desempeño, se evaluará con un conjunto de datos del cuál no tendrán acceso. De esta manera evitaremos que los modelos aprendan información sobre el conjunto de validación.

- Crea una carpeta de trabajo y guarda todos los archivos correspondientes (notebook, archivos auxiliares y csv).
- Una vez terminada la prueba, comprime la carpeta y sube el `.zip` a la sección correspondiente.

Objetivos

Para alcanzar el objetivo general, su trabajo se puede desagregar en los siguientes puntos:

1. Generar un análisis exploratorio sobre los datos contenidos en el DataFrame, considerando palabras más comunes y distribución de las clases en el vector objetivo.

2. Preprocesamiento de Texto:

- Para trabajar adecuadamente con texto, debemos preprocesar y posteriormente representar cada oración como un conjunto de características.
- Para preprocesar los tweets, debemos transformarlos a lower case. Un problema recurrente en el análisis de texto es la alta ocurrencia de palabras comunes. Se recomienda eliminarlas

mediante la declaración de stopwords. Para generar la exclusión de stopwords, podemos utilizar la librería `nltk` (Natural Language ToolKit) y descargar los stopwords con la siguiente instrucción.

```
import nltk
nltk.download('stopwords')
```

- Puede refinar los atributos a capturar mediante el proceso de **lemantización** (la reducción de variadas palabras con un tronco léxico común; ejemplo: *Organización*, *Organiza*, y *Organizado* presentan `organi_` como tronco léxico en comúmn) o **Stemming** (la reducción de una palabra a una expresión generalizable). Cabe destacar que ésta última carece de análisis morfológico del lenguaje.
- Posterior a la refinación y preprocesamiento de las palabras, podemos representar cada oración en una matriz (o corpus) que permitirá reflejar la cantidad de ocurrencias de w_i palabra en un registro. Para ello, pueden hacer uso de las librerías de preprocesamiento `sklearn.feature_extraction.text.CountVectorizer` o `sklearn.feature_extraction.text.TfidfVectorizer`. DE esta manera, tendremos un conjunto de características es mediante la frecuencia de ocurrencia de una palabra o término en el texto.

3. Preparación del vector objetivo y las matrices de entrenamiento y validación:

- Nos interesa trabajar con dos tipos de emociones: positivas o negativas. Para ello deberá generar la recodificación de cada una de las clases en una de las dos emociones:

Original	Recodificación
`'worry'`	Negativa
`'happiness'`	Positiva
`'sadness'`	Negativa
`'love'`	Positiva
`'surprise'`	Positiva
`'fun'`	Positiva
`'relief'`	Positiva
`'hate'`	Negativa
`'empty'`	Negativa
`'enthusiasm'`	Positiva
`'boredom'`	Negativa
`'anger'`	Negativa

- Si el tweet está asignado como `neutral`, clasifíquelo aleatoriamente entre positivo o negativo.

4. Entrenamiento de modelos:
    - En base a los modelos vistos en clase, implemente por lo menos 5. Para cada uno de ellos justifique la elección de hiperparámetros. Si implementa búsqueda de grilla para cada uno de ellos, defina el rango de valores a tomar en cada hiperparámetro.
    - Reporte el desempeño de cada modelo en las muestras de entrenamiento y validación. Comente sobre la capacidad de generalización de cada uno de ellos haciendo uso de los conceptos vistos en el curso.

5. Seleccione los 2 mejores modelos, serialicelos y envíelos a evaluación. Recuerde que el modelo serializado debe ser posterior al `fit`, para poder ejecutar `predict` en los nuevos datos.

6. La evaluación del modelo será realizada en función a un conjunto de datos reservados al cual no tienen acceso.

Evaluación

La siguiente rúbrica detalla los elementos de evaluación:

- Notebook (**20 puntos**): El notebook debe ser un reporte con la estrategia analítica, explicando los siguientes puntos:
    - La definición de los requerimientos, la definición del vector objetivo, la definición de las métricas a utilizar.(**3 puntos**)
    - Un análisis exploratorio (univariado y gráfico). Como mínimo, debe analizar el comportamiento del vector objetivo antes del preprocesamiento y posterior al procesamiento.(**5 puntos**)
    - La estrategia de preprocesamiento/feature engineering.(**2 puntos**)
    - La elección de los algoritmos a implementar, así como sus hiperparámetros. Un reporte sobre qué modelos enviarán a competencia. (**10 puntos**)
- Modelos serializados:
    - Los modelos deben estar serializados con la siguiente nomenclatura: `nombre_grupo-modelo-1` y `nombre_grupo-modelo-2`.

La evaluación de los modelos serializados se realizará en función al desempeño predictivo del modelo en un conjunto de datos externos.

La primera instancia es evaluar los dos modelos enviados a competencia por el grupo, preservando el mejor modelo para la competencia con los otros grupos.

El segundo paso es rankear según desempeño entre grupos.

> El alumno debe obtener un mínimo de 16 puntos para aprobar