Modelo predictivo con variables numéricas y categóricas

Schneider Electric and NUWA - Hackathon 2022

EQUIPO 32:

- Andrés Jordán Gamito (andresjordangamito@gmail.com)
- Manuel Serrano Rodríguez (manuel.serranorod@gmail.com)

Desglose de Datos

1. Hemos importado los *.csv* y los *.json* como *dataframes* de *Pandas*. Una estructura de datos tabular que nos facilitará mucho la creación de nuestro modelo predictivo.
    - *Pandas* es una herramienta software de análisis y manipulación de datos en lenguaje python.
    - Mientras que *train1* y *test_x* pueden importarse directamente con la función *read_csv*, para *train2* es necesario especificar el delimitador “ ; ”, al encontrarse en un formato de *.csv* distinto.
    - Los *json* pueden importarse directamente con la función *read_json* cuyo argumento será el link proporcionado.
2. Los sets *train3*, *train4* y *train5* contienen 3 campos que no aparecen en *train1* ni *train2*, por lo que son directamente eliminados del *dataframe* con la función *drop*.
3. Se reordenan los campos de los 5 *dataframes* por orden alfabético según el nombre de cabecera.
4. Se concatenan los 5 *dataframes* de entrenamiento uno detrás de otro con la función *concat*.

Creación del modelo

1. Para nuestro predictor usaremos la librería scikit/sklearn de Python, una herramienta muy útil para el análisis predictivo de datos.
2. Establecemos el set de entrenamiento del predictor, donde X serán las variables a introducir e Y la variable a estimar.
    a. Es decir, X será todo el dataframe menos el campo “pollutant” mientras que Y será únicamente el campo “pollutant”.
3. Se definen transformadores para preparar las variables enteras, continuas y categóricas y así garantizar que se expone de la mejor manera posible la estructura del modelado a los algoritmos de aprendizaje.
    a. La aplicación de transformaciones de datos puede ser un reto cuando se tiene un conjunto de datos con tipos mixtos, como es el caso.
    b. La librería scikit-learn nos permite utilizar *ColumnTransformer* para aplicar las transformaciones a las diferentes columnas del *dataframe*.
4. Introducimos las variables del *dataset* en nuestro modelo y realizamos la predicción.

RESULTADOS

- Para comprobar la eficacia de nuestro predictor, hemos utilizado el set *train5* como set de testeo, ya que *test_x* no disponía de datos de contaminantes.
- Para la predicción obtenida, utilizando como sets de entrenamiento los otros 4 *train*, hemos obtenido un *f1-score* de 0,63.

```
In [160]: runfile('C:/Users/manue/Desktop/HACKATHON/predictor.py', wdir='C:/Users/manue/Desktop/HACKATHON')

In [161]: f1_score(df5['pollutant'],preds, average='macro')
Out[161]: 0.6335135576037635
```

- Posteriormente, utilizando los 5 *train* como sets de entrenamiento y *test_x* como set de validación, hemos exportado los resultados a *.csv* y *.json.*