

## Aprendizaje Automático

### Entrega 2: 17/06 Descripción del Dataset y Origen

- En esta segunda entrega, debe tener acceso al dataset que utilizará en su proyecto de [Aprendizaje Automático](#).
- Proporcione una descripción completa del dataset, incluyendo la cantidad de instancias, características (columnas), tipos de datos, y cualquier información relevante.
- Informe sobre el origen del dataset, es decir, de dónde provienen los datos. Esto puede incluir la fuente, la fecha de adquisición y cualquier proceso de recopilación o preprocesamiento que haya realizado.

Los data sets a los que se tiene acceso son tres, uno por tipo de vehículo, en formato json. "response.json" Son los reportes generados por el camión. "Camioneta furgon Iveco Daily.json" como lo indica el nombre es de los reportes generados por el vehículo furgón. Y el último reporte es de un vehículo pequeño conocido como "xampi" Linde H25.json

Reporte	Tipo de vehículo
response.json	Camion General ATEGO 916
CamionetafurgonIvecoDaily.json	Camioneta furgon
LindeH25.json	Xampi

El conjunto de datos JSON se compone de tres objetos principales:

#### 1. Objeto **gral**:

Este objeto proporciona información general sobre el viaje:

- **number**: Identificador único del viaje (se presume un número).
- **from**: Ubicación de inicio del viaje (se presume una cadena de texto).
- **to**: Ubicación final del viaje (se presume una cadena de texto).
- **type**: Tipo de viaje (cadena de texto, valor fijo "can").
- **dvcType**: Tipo de dispositivo (cadena de texto, valor fijo "BCE").

- **\_v**: Versión del formato de datos (cadena de texto, valor fijo "next").
- **hash**: Valor hash único para el viaje (cadena de texto).
- **kmRec**: Distancia total recorrida en kilómetros (número).
- **engOnSec**: Tiempo total de motor encendido en segundos (número).
- **motionTime**: Tiempo total de movimiento en segundos (número).
- **maxSpeed**: Objeto que contiene la velocidad máxima alcanzada:
  - **speed**: Valor de la velocidad máxima (número).
  - **date**: Fecha y hora en que se alcanzó la velocidad máxima (número, posiblemente como timestamp).
- **canKmRec**: Distancia total registrada por el CAN bus en kilómetros (número).
- **canEngOnSec**: Tiempo total de motor encendido registrado por CAN bus en segundos (número).
- **canConsL**: Consumo total de combustible registrado por CAN bus en litros (número).
- **canL100Km**: Rendimiento de combustible promedio registrado por CAN bus en litros por cada 100 kilómetros (número).
- 

## 2. Array **trips**:

Este array contiene información detallada de cada segmento dentro del viaje (ya sean viajes propiamente dichos, paradas o segmentos sin GPS):

- Cada elemento del array es un objeto que describe un segmento específico.
- **type**: Tipo de segmento (cadena de texto):
  - **"no-gps"**: Segmento sin señal GPS.
  - **"parked"**: Segmento en el que el vehículo estuvo estacionado.
  - **"trip"**: Segmento de viaje con información detallada.
- Los elementos pueden tener campos adicionales dependiendo del tipo de segmento:
  - **"no-gps"** (sin cambios respecto a la descripción anterior): \*
- **start**: Fecha y hora de inicio del segmento sin GPS (número, posiblemente timestamp). \* **end**: Fecha y hora de fin del segmento sin GPS (número, posiblemente timestamp). \*
- **from** (opcional): Objeto con la ubicación inicial del segmento sin GPS (misma estructura que en **gral**). \*
- **to** (opcional): Objeto con la ubicación final del segmento sin GPS (misma estructura que en **gral**). \*
- **events** (opcional): Array de objetos que describen eventos ocurridos durante el segmento sin GPS. \*

- **evDesc**: Descripción del evento (cadena de texto). \* **date**: Fecha y hora del evento (número, posiblemente timestamp).
- **"parked"** (sin cambios respecto a la descripción anterior): \*
- **lat**: Latitud de la ubicación donde se estacionó el vehículo (número). \*
- **lng**: Longitud de la ubicación donde se estacionó el vehículo (número). \*
- **start**: Fecha y hora de inicio del estacionamiento (número, posiblemente timestamp). \*
- **end**: Fecha y hora de fin del estacionamiento (número, posiblemente timestamp). \*
- **dleTime**: Tiempo en segundos que el vehículo estuvo en ralentí durante el estacionamiento (número). \*
- **events** (opcional): Array de objetos que describen eventos ocurridos durante el estacionamiento, similar a **"no-gps"**. \*
- **dir** (opcional): Dirección cardinal hacia la que apuntaba el vehículo mientras estaba estacionado (cadena de texto). \*
- **status** (opcional): Estado del vehículo durante el estacionamiento (cadena de texto, significado específico desconocido).
- **"trip"** (cambios resaltados): \*
- **start**: Fecha y hora de inicio del segmento de viaje (número, posiblemente timestamp). \*
- **end**: Fecha y hora de fin del segmento de viaje (número, posiblemente timestamp). \*
- **kmRec**: Distancia total recorrida en el segmento en kilómetros (número). \*
- **engOnSec**: Tiempo total de motor encendido en el segmento en segundos (número). \*
- **motionTime**: Tiempo total de movimiento en el segmento en segundos (número). \*
- **maxSpeed** (opcional): Objeto con la velocidad máxima alcanzada en el segmento, con la misma estructura que en **gral**. \*
- **canKmRec**: Distancia total registrada por el CAN bus en el segmento en kilómetros (número). \*
- **canEngOnSec**: Tiempo total de motor encendido registrado por CAN bus en el segmento en segundos (número). \*
- **canConsL**: Consumo total de combustible registrado por CAN bus en el segmento en litros (número). \*
- **canL100Km**: Rendimiento de combustible promedio registrado por CAN bus en litros por cada 100 kilómetros (número). \*
- **from** (opcional): Objeto con la ubicación inicial del segmento de viaje (misma estructura que **to** en **gral**). \*

- **to** (opcional): Objeto con la ubicación final del segmento de viaje (misma estructura que **to** en **gral**). \*
- **rows**: Array de objetos que contienen datos detallados a lo largo del segmento de viaje: \* **date**: Fecha y hora del punto de registro (número, posiblemente timestamp). \*
- **lat**: Latitud del punto de registro (número). \*
- **lng**: Longitud del punto de registro (número). \*
- **alt**: Altitud del punto de registro (número). \*
- **speed**: Velocidad del vehículo en el punto de registro (número). \*
- **heading**: Dirección de desplazamiento del vehículo en el punto de registro (número). \*
- **km**: Distancia total recorrida desde el inicio del viaje hasta el punto de

### Array **details**:

Este array puede contener información puntual registrada en algún momento del viaje, pero no necesariamente asociada a un segmento específico.

Cada elemento del array es un objeto con los siguientes campos:

- **date**: Fecha y hora del registro (número, posiblemente timestamp).
- **lat**: Latitud del punto de registro (número).
- **lng**: Longitud del punto de registro (número).
- **speed**: Velocidad del vehículo en el punto de registro (número).
- **heading**: Dirección de desplazamiento del vehículo en el punto de registro (número).
- **km**: Distancia total recorrida desde el inicio del viaje hasta el punto de registro en kilómetros (número).
- **bat**: Nivel de batería del vehículo en el punto de registro (número).
- **engOn**: Indica si el motor estaba encendido en el punto de registro (booleano).
- **evDesc** (opcional): Descripción de un evento ocurrido en el punto de registro (cadena de texto).
- **wheel\_speed** (opcional): Velocidad de las ruedas del vehículo en el punto de registro (número).
- **total\_fuel** (opcional): Consumo total de combustible desde el inicio del viaje hasta el punto de registro (número).
- Los campos **can\_fls**, **can\_taho**, **can\_engine\_hrs**, **can\_mileage**, **eng\_temp**, y **dir** pueden estar presentes de forma opcional con significados específicos desconocidos (número)

El origen de los datos es de la empresa Transdat generados por los reportes de los vehículos. Los datos fueron facilitados por un colaborador de la empresa. Y son datos recolectados con información de un periodo de una semana.

Luego de haber extraído ['details'] del objeto 'gral' del reporte 'response.json' (camion). Se puede observar que existen un total de 12630 registros, 16 Columnas, y los tipos de datos son bool(1), float64(12), int64(1), object(2)

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12631 entries, 0 to 12630
Data columns (total 16 columns):
#   Column                Non-Null Count  Dtype
---  -
0   date                  12631 non-null  int64
1   lat                   12627 non-null  float64
2   lng                   12627 non-null  float64
3   speed                 12627 non-null  float64
4   heading               12627 non-null  float64
5   bat                   12631 non-null  float64
6   km                    12631 non-null  float64
7   engOn                 12631 non-null  bool
8   evDesc                12631 non-null  object
9   wheel_speed           11961 non-null  float64
10  total_fuel            11961 non-null  float64
11  can_fls               11961 non-null  float64
12  can_taho              11961 non-null  float64
13  can_mileage           11961 non-null  float64
14  eng_temp              11961 non-null  float64
15  dir                   12631 non-null  object
dtypes: bool(1), float64(12), int64(1), object(2)
```

En el reporte 'Linde H25.json' (Xampi) también se accede al array 'details'. Pudiendo observarse 15 columnas y 25244 registros. Los tipos de datos son bool(1), float(10), int (2), object(2)

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25244 entries, 0 to 25243
Data columns (total 15 columns):
#   Column                Non-Null Count  Dtype
---  -
0   date                  25244 non-null  int64
1   bat                   25244 non-null  float64
2   km                    25244 non-null  int64
3   engOn                 25244 non-null  bool
4   evDesc                25244 non-null  object
5   dir                  25244 non-null  object
6   lat                   22890 non-null  float64
7   lng                   22890 non-null  float64
8   speed                 22890 non-null  float64
9   heading               22890 non-null  float64
10  wheel_speed           22396 non-null  float64
11  can_fl                22396 non-null  float64
12  can_taho              22396 non-null  float64
13  can_engine_hrs        22396 non-null  float64
14  eng_temp              22396 non-null  float64
dtypes: bool(1), float64(10), int64(2), object(2)

```

Y en cuanto al ultimo reporte de la furgoneta, se accedió a ‘details’ y ‘trips’

Trips tiene en total 25 columnas y 62 registros, por lo que no sería posible trabajar con este dataset, debido a que la mayoría de las columnas están incompletas y tienen datos faltantes.