UNIVERSIDAD EUROPEA
MIGUEL DE CERVANTES

ESCUELA POLITÉCNICA SUPERIOR

TITULACIÓN:
MÁSTER UNIVERSITARIO EN GESTIÓN Y ANÁLISIS DE GRANDES VOLÚMENES DE DATOS: BIG DATA

TRABAJO FIN DE MÁSTER

Title of the project

AUTOR

Daniel García Teba

TUTOR

Miguel Angel Gomez Lopez

VALLADOLID, septiembre de 2020

Índice de contenidos

Índice de figuras

Índice de tablas

Capítulo 1

Objetivos del trabajo

Dados unos datos experimentales provenientes de una nariz electrónica, **insertar aquí enlace UCI data** se desea hacer un estudio de la viabilidad de clasificar correctamente de qué gas se trata, e intentar dar una estimación de la concentración de dicho gas.

Los datos provienen de cada medición de un gas con 16 sensores. De esta forma, cada medición genera 16 series temporales, de las cuales de cada una se han extraido 8 *features*. Esto hace un total de 128 componentes para cada medición. Para cada medición es conocido el gas que se ha ensayado y su nivel de concentración.

Con este tipo de información, se va a estudiar los resultados que pueden ofrecer las redes neuronales para resolver este problema.

En el caso que nos ocupa, los sensores derivan a lo largo del tiempo, lo cual implica que para el mismo gas, con la misma concentración, las series temporales obtenidas en un mes dado podrían ser diferentes a las realizadas con las mismas condiciones meses después.

En este trabajo se tomará como primera aproximación dividir la tarea de clasificación y la tarea de estimación.

Para la tarea de clasificación, se probarán las siguientes configuraciones, de más simple a más compleja:

- ▷ Perceptron
- ▷ **Long short-term memory Neural net**

Una vez conseguida la tarea de clasificación, se usará otra red neuronal para determinar, si es posible, la concentración del gas.

En este trabajo se utilizará Keras y TensorFlow para el diseño de las redes neuronales, en el entorno Python.

Capítulo 2

Ánalisis de la situación

De los estudios realizados en *Chemical gas sensor drift compensation using classifier ensembles* **??** se deriva que los sensores utilizados derivan a lo largo del tiempo.

Esto presenta un reto para entrenar un algoritmo de clasificación. En este trabajo primero comprobaremos que existe esta deriva de los sensores, para posteriormente aplicar algoritmos que sean capaces de hacer buenas predicciones a partir de los datos disponibles.

Capítulo 3

Obtención, procesado y almacenamiento de los datos

Los datos provienen del artículo Chemical gas sensor drift compensation using classifier ensembles (?, ?),

donde el objetivo era tratar de detectar el drift (la deriva) de los sensores a lo largo de los meses, y poder calibrarlos utilizando el minimo numero de experimentos posibles. Es decir, de la forma más rápida y eficiente posible. (ver (?, ?))

Los datos estan disponibles para su descarga desde UCI data repository. en 10 archivos formato *.dat*.

Cada lote cuenta con una estructura de 129 columnas, donde la primera nos informa del gas y la concentración, y el resto es la información obtenida del sensor.

El primer paso que se va a realizar es, dada las 128 componentes X, averiguar a qué tipo de gas pertenece la medición.

Una vez la red es capaz de inferir correctamente de qué gas se trata, alimentaremos otra red cuya función sea averiguar la concentración del mismo.

3.0.1 Procesado

La lectura de los archivos *.dat* se ha realizado utilizando el código adjunto en Apendices **??**

3.0.2 Exploratory Data Analysis

Los datos se nos presentan en 10 lotes, correspondientes a experimentos a lo largo de tres años, donde se ensayaron 6 diferentes gases a diferentes concentraciones.

Batch ID	Month IDs
Batch 1	Months 1 and 2
Batch 2	Months 3, 4, 8, 9 and 10
Batch 3	Months 11, 12, and 13
Batch 4	Months 14 and 15
Batch 5	Month 16
Batch 6	Months 17, 18, 19, and 20
Batch 7	Month 21
Batch 8	Months 22 and 23
Batch 9	Months 24 and 30
Batch 10	Month 36

Tabla 3.1: Distribución de los lotes a lo largo del tiempo.

Los gases que se estudiaron son los siguientes:

1. Ethanol
2. Ethylene
3. Ammonia
4. Acetaldehyde
5. Acetone
6. Toluene

Los lotes contienen una cantidad de muestras desigual, ni los 6 gases de estudio están presentes en todos los lotes.

La tabla **??** muestra el numero de ensayos para cada gas.

En la Tabla**??** podemos ver que el número de muestras en cada lote es desigual. En los lotes 3, 4 y 5 el gas 6 no está presente. A la hora de crear un dataset de entrenamiento, convendria generar un lote donde haya un numero equitativo de muestras de todos los gases, y si las mediciones no son distantes en el tiempo podremos ver el efecto de la deriva si el algorimo entrenado con los primeros lotes falla para cada vez más conforme nos alejamos en el tiempo.

	Batch ID	1	2	3	4	5	6
1	90	98	83	30	70	74	
2	164	334	100	109	532	5	
3	365	490	216	240	275	0	
4	64	43	12	30	12	0	
5	28	40	20	46	63	0	
6	514	574	110	29	606	467	
7	649	662	360	744	630	568	
8	30	30	40	33	143	18	
9	61	55	100	75	78	101	
10	600	600	600	600	600	600	

La Figura 3.1 muestra la cantidad de gases ensayados por lote, mientras que la Figura3.2 muestra el numero de mediciones totales sobre cada gas.

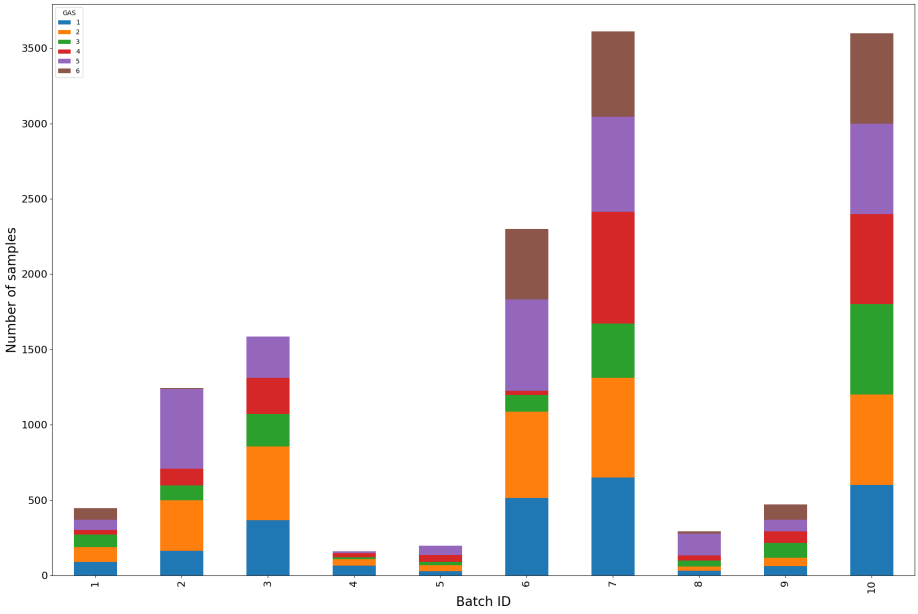

Figura 3.1: Número de muestras de gas por Batch

Si observamos los rangos de concentración para cada gas, han sido también diferentes.

Esta información es necesario tenerla en cuenta a la hora de entrenar nuestro

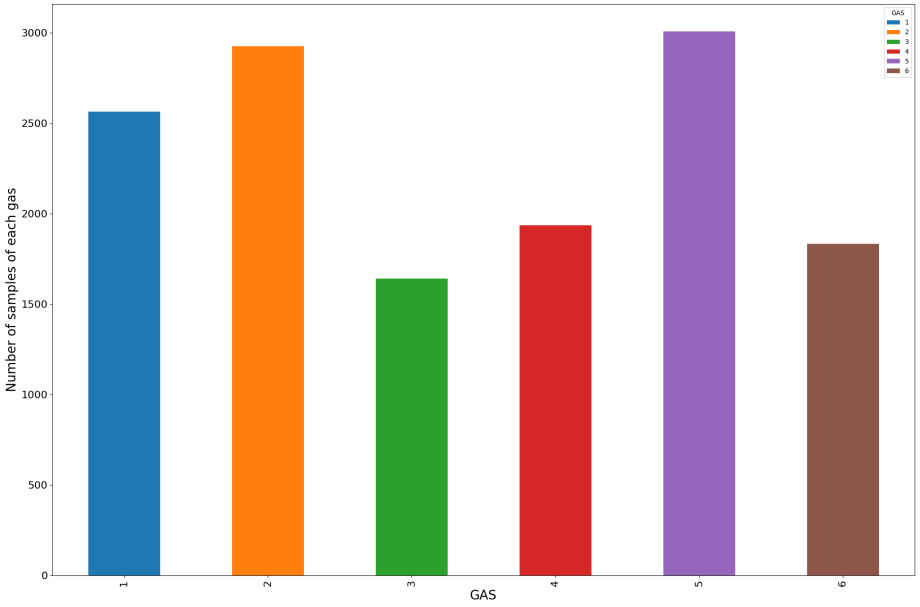

Figura 3.2: Número de muestras de cada gas en total

modelo, ya que si el rango de variación de los datos es dispar, será recomendable normalizar.

	GAS	Minimo	Máximo	Media	StdDesv
1	2.5	600	114.95	86.64	
2	2.5	300	116.1	79.89	
3	2.5	1000	323.55	272.02	
4	2.5	300	126.32	76.71	
5	10	1000	228.57	217.38	
6	1	230	47.66	32.58	

Capítulo 4

Diseño e implementación de los modelos o ténicas necesarias

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Capítulo 5

Análisis de los resultados obtenidos con los modelos

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Capítulo 6

Conclusiones y planes de mejora

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Capítulo 7

Conclusiones

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet

vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Apéndice A

Código utilizado

A.1 Código Exploratory Data Analysis

```
import pandas as pd
import matplotlib.pyplot as plt

from python.LoadUciData import LoadDatFolder
from python.StandardFigure import save_figure
pd.set_option('display.max_columns', 10)


def plot_count_per_batch_and_gas(df_gas):
    props = df_gas.groupby("Batch ID")['GAS'].value_counts(normalize=
    False).unstack()
    ax = props.plot(kind='bar', stacked='True', figsize=(24, 16))
    ax.set_ylabel('Number of samples')
    save_figure(plt.gcf(), 'Step0_Count_Batch_Gas')


def plot_sample_count_per_gas(df_gas):
    props = df_gas.groupby('GAS')['GAS'].value_counts(normalize=False
    ).unstack()
    ax = props.plot(kind='bar', stacked='True', figsize=(24, 16))
    ax.set_ylabel('Number of samples of each gas')
    save_figure(plt.gcf(), 'Step0_Count_Gas')


if __name__ == '__main__':

    # Load .dat files
    folder = r'data_uci/driftdataset'
    df_gas = LoadDatFolder(folder).df

    ## Tables
```

```
    # Show samples per Batch and Gas
    gas_batch_group = pd.crosstab(df_gas['Batch ID'], df_gas['GAS']).
    sort_index()
    print('\n', gas_batch_group.to_markdown())

    # Show concentration range statistics per GAS
    pivot = pd.pivot_table(df_gas,
                           index=['GAS'],
                           values='CONCENTRATION',
                           aggfunc=['min', 'max', 'mean', 'std', '
    count'])
    pivot.round(2)
    print('\n', pivot.round(2).to_markdown())

    ## Plots
    # Show samples count per GAS
    plot_count_per_batch_and_gas(df_gas)
    plot_sample_count_per_gas(df_gas)
```

Listing A.1: Realiza el analisi exploratorio de los datos disponibles

A.2 Código Red neuronal secuencial

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.python.keras.callbacks import TensorBoard
from time import time

class SeqModel:
    """ Sequential Neutal Net."""
    def __init__(self):
        pass

    def _gen_model_seq(self):
        model = keras.Sequential([
            keras.layers.Flatten(input_shape=(self.x_size, 1)),
            keras.layers.Dense(64, activation='relu'),
            keras.layers.Dense(10)
            ])
        return model

    def _gen_and_complile_model(self):
        model = self._gen_model_seq()
        model.summary()
        model.compile(optimizer='adam',
                      loss=tf.keras.losses.
    SparseCategoricalCrossentropy(from_logits=True),
                      metrics=['accuracy'])
```

```
        return model

    def model_train(self, X_train, y_train):
        # TensorFlow and tf.keras
        self.x_size = X_train.shape[1]
        model = self.gen_and_complile_model()
        tb = TensorBoard(log_dir="logs/{}".format(time()))
        model.fit(X_train, y_train, epochs=30, callbacks=[tb])
        self.model = model

    def model_evaluate(self, X_test, y_test):
        test_loss, test_acc = self.model.evaluate(X_test, y_test,
   verbose=2)
        print('\nTest accuracy:', test_acc)
        return test_loss, test_acc
```

Listing A.2: Red neuronal secuencial

```
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow import keras
from time import time
from tensorflow.python.keras.callbacks import TensorBoard
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import StandardScaler
from sklearn import metrics
import matplotlib.pyplot as plt

from python.LoadUciData import load_data
from python.StandardFigure import save_figure
from python.SeqModel import SeqModel


def gas_clasification_with_seq():
    # Load data
    df_gas = load_data()

    ## GAS CLASIFICATION
    ## First, we will NOT use concentration data
    gas_X = df_gas.drop(columns=['Batch ID', 'GAS', 'CONCENTRATION'])
   .to_numpy()
    gas_y = df_gas['GAS'].to_numpy()

    X_train, X_test, y_train, y_test = train_test_split(gas_X, gas_y,
                                                        test_size
   =0.33,
```

```
                                                        random_state=42)
    seq = SeqModel()
    model = seq.model_train(X_train, y_train)
    test_loss0, test_acc0 = seq.model_evaluate(model, X_test, y_test)

    # Check the changes if we include concentration info
    gas_X = df_gas.drop(columns=['Batch ID', 'GAS']).to_numpy()
    gas_y = df_gas['GAS'].to_numpy()

    X_train, X_test, y_train, y_test = train_test_split(gas_X, gas_y,
                                                        test_size=0.33,
    random_state=42)
    seq = SeqModel()
    model = seq.model_train(X_train, y_train)
    test_loss1, test_acc1 = seq.model_evaluate(model, X_test, y_test)


def evolution_drift_with_seq(n_batch_training):
    ## Check the drift importance. Use the first 3 batch to train, and check the
    ## clasifications results with the others.
    df_gas = load_data()

    df_gas_train = df_gas[df_gas['Batch ID'].isin(range(1, n_batch_training + 1))]
    gas_train_X = df_gas_train.drop(columns=['Batch ID', 'GAS']).to_numpy()
    gas_train_y = df_gas_train['GAS'].to_numpy()
    seq = SeqModel()
    model = seq.model_train(gas_train_X, gas_train_y)

    model_dict = {}
    for batch in range(n_batch_training + 1, 11):
        df_gas_test  = df_gas[df_gas['Batch ID'] == batch]
        gas_test_X   = df_gas_test.drop(columns=['Batch ID', 'GAS']).to_numpy()
        gas_test_y   = df_gas_test['GAS'].to_numpy()

        print(f'Batch {batch}')

        loss, acc  = seq.model_evaluate(model, gas_test_X, gas_test_y)
        model_dict[batch] = {'acc': acc, 'loss': loss}

    # figures
    df_results = pd.DataFrame.from_dict(model_dict).T
```

```
71
72      fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12,8))
73      fig.suptitle(f'Training with first {n_batch_training} batches')
74      ax1 = df_results.plot(kind='bar', y='acc', ax=ax1)
75      ax1.set_ylim([0,1])
76      ax2 = df_results.plot(kind='bar', y='loss', ax=ax2)
77      save_figure(fig, f'Step1_NBATCH_{n_batch_training}_acc_loss')
78
79
80  if __name__ == '__main__':
81
82      # Check the results for the sequential Neural Net
83      gas_clasification_with_seq()
84
85      # Check drift relevance  with different training sets.
86      for i in range(1, 10):
87          evolution_drift_with_seq(i)
```

Listing A.3: Red neuronal secuencial para clasificar gases

```
1   from sklearn.model_selection import train_test_split
2
3   from python.LoadUciData import load_data
4
5   if __name__ == '__main__':
6
7       # Load data
8       df_gas = load_data()
9
10      df_train_reg = df_gas[df_gas['GAS'] == 1]
11
12      gas_X = df_train_reg.drop(columns=['Batch ID', 'GAS', '
        CONCENTRATION']).to_numpy()
13      gas_y = df_train_reg['CONCENTRATION'].to_numpy()
14
15      X_train, X_test, y_train, y_test = train_test_split(gas_X, gas_y,
16                                                          test_size
        =0.33,
17                                                          random_state
        =42)
```

Listing A.4: Red neuronal secuencial para estimar la concentración

A.3 Código Utilities

```
1   import os
2   import pandas as pd
3   import re
4   import pickle
```

```
from python.FileUtils import get_list_of_files_with_extension


class LoadDatFile:
    """
    This class aims to load a .dat files from UCI
    https://archive.ics.uci.edu/ml//datasets/Gas+Sensor+Array+Drift+Dataset
    , and returns a pandas.dataframe object

    :arg .dat file
    :return df
    """

    def __init__(self, file):
        self.file = file

    @property
    def batch_number(self):
        base = os.path.basename(self.file)
        name, ext = os.path.splitext(base)
        num = re.findall(r'\d+', name)[0]
        # num = num.zfill(2)
        return int(num)

    @property
    def df(self):
        df = pd.read_table(self.file, engine='python', sep='\s+\d+:', header=None)
        df['Batch ID'] = self.batch_number
        return df

class GasDataFrame:
    """ Process the .dat file to get all the information contained:
    - Gas, concentration and measures."""

    def __init__(self, file):
        self.file = file

    @property
    def df(self):
        df_raw = LoadDatFile(self.file).df
        return self._add_gas_info(df_raw)

    @staticmethod
    def _add_gas_info(df):
        df[['GAS', 'CONCENTRATION']] = df.iloc[:, 0].str.split(";", expand=True, )
        df.drop(df.columns[0], axis=1, inplace=True)
```

```
        df['GAS'] = df['GAS'].astype('int')
        df['CONCENTRATION'] = df['CONCENTRATION'].astype('float')
        return df

class LoadDatFolder:
    """
    This class aims to load all .dat files contained in a folder,
    gives each file a GasDataframe format and concats all in a pandas
    .dataframe object with

    :inputs: folder with many .dat files
    :return df
    """
    def __init__(self, folder):
        self.folder = folder

    @property
    def df(self):
        files = get_list_of_files_with_extension(self.folder, 'dat')
        df_full = pd.DataFrame()
        for f in files:
            dftemp = GasDataFrame(f).df
            df_full = df_full.append(dftemp)
        return df_full


def load_data():
    folder = r'data_uci/driftdataset'
    df_gas = LoadDatFolder(folder).df
    return df_gas

if __name__ == '__main__':
    file_data = r'data_uci/driftdataset/batch1.dat'
    lf = LoadDatFile(file_data)
    my_dataframe = lf.df

    gdf = GasDataFrame(file_data)
    my_dataframe_gas = gdf.df

    folder = r'data_uci/driftdataset/'
    ldf = LoadDatFolder(folder)
    my_dataframe_full = ldf.df
```

Listing A.5: Clases para cargar datos en memoria

Referencias

Finazzi, I. F. (2020). Latex template for master's thesis of UEMC. *UEMC*, *1*(1), 1.