

NOMBRE	MATERIA	CI	NUMERO DE PREGUNTA
Joel Modesto Anara Michua	Inteligencia Artificial inf-354	10911944	4

4. Con el uso de librerías realiza en Python los mismos preprocesamiento del punto 3.

## CODIGO

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import Normalizer
from sklearn.preprocessing import KBinsDiscretizer
import numpy as np
print("-----ARCHIVO-----")
archivo=pd.read_csv("DatasetEx.csv")

#PREGUNTA 4
print("-----LABELENCODER-----")
archivo["class"]=archivo["class"].astype(str)
#clase=archivo["class"]
#preprocesamiento = LabelEncoder()
#archivo["class"] = preprocesamiento.fit_transform(clase)
#print(archivo["class"])
#print("----ONEHOTCODE-----")
from sklearn.preprocessing import OneHotEncoder

preprocesamiento = OneHotEncoder(sparse_output=False)

resultado5 = preprocesamiento.fit(archivo[["class"]])

print(resultado5.categories_)

archivo["class"] = preprocesamiento.transform(archivo[["class"]])

print(archivo["class"])

def Normalizar(data, norm_type='l2'):
    print("-----NORMALIZACION-----")
    preprocesamiento = Normalizer(norm=norm_type)
    normalized_data = preprocesamiento.fit_transform(data.values.reshape(-1, 1)) #
    Convertir a 2D
    return normalized_data

anio = archivo["year"]
duracion = archivo["duration_ms"]
danceabilidad = archivo["danceability"]
loudness = archivo["loudness"]
```

```

speechiness = archivo["speechiness"]
acousticness = archivo["acousticness"]
instrumentalness = archivo["instrumentalness"]
valence = archivo["valence"]
tempo = archivo["tempo"]
time_signature = archivo["time_signature"]

archivo["year"] = Normalizar(anio, '12')
archivo["duration_ms"] = Normalizar(duracion, '12')
archivo["danceability"] = Normalizar(danceabilidad, '12')
archivo["loudness"] = Normalizar(loudness, '12')
archivo["speechiness"] = Normalizar(speechiness, '12')
archivo["acousticness"] = Normalizar(acousticness, '12')
archivo["instrumentalness"] = Normalizar(instrumentalness, '12')
archivo["valence"] = Normalizar(valence, '12')
archivo["tempo"] = Normalizar(tempo, '12')
archivo["time_signature"] = Normalizar(time_signature, '12')

print("----discretizacion----")

def discretizar(vector, n_bins=5):
    print("-----DISCRETIZACION-----")
    vector = np.array(vector).reshape(-1, 1)
    discretizer = KBinsDiscretizer(n_bins=n_bins, encode='ordinal',
strategy='uniform')
    discretized_vector = discretizer.fit_transform(vector)
    return discretized_vector.flatten()

archivo["year"] = discretizar(anio, n_bins=10)
archivo["duration_ms"] = discretizar(duracion, n_bins=10)
archivo["danceability"] = discretizar(danceabilidad, n_bins=10)
archivo["loudness"] = discretizar(loudness, n_bins=10)
archivo["speechiness"] = discretizar(speechiness, n_bins=10)
archivo["acousticness"] = discretizar(acousticness, n_bins=10)
archivo["instrumentalness"] = discretizar(instrumentalness, n_bins=10)
archivo["valence"] = discretizar(valence, n_bins=10)
archivo["tempo"] = discretizar(tempo, n_bins=10)
archivo["time_signature"] = discretizar(time_signature, n_bins=10)
print(archivo)
for col in ["year", "duration_ms", "danceability", "loudness", "speechiness",
            "acousticness", "instrumentalness", "valence", "tempo",
            "time_signature"]:
    print(f"--- Discretización de {col} ---")
    print(archivo[col])

```

CORRIDA

-----DISCRETIZACION-----											
	year	duration_ms	danceability	energy	key	...	instrumentalness	valence	tempo	time_signature	
0	4.0	1.0	4.0	0.826	4	...	0.0	5.0	4.0	7.0	
1	8.0	1.0	4.0	0.430	7	...	0.0	0.0	2.0	7.0	
2	7.0	1.0	5.0	0.429	5	...	0.0	1.0	5.0	7.0	
3	7.0	1.0	5.0	0.490	4	...	0.0	4.0	1.0	7.0	
4	8.0	1.0	8.0	0.720	6	...	0.0	8.0	5.0	7.0	
...	...	...	...	...	...	...	...	...	...	...	
1095	8.0	1.0	3.0	0.851	1	...	6.0	6.0	5.0	7.0	
1096	8.0	3.0	1.0	0.953	10	...	9.0	0.0	4.0	7.0	
1097	7.0	1.0	0.0	0.239	3	...	0.0	0.0	1.0	7.0	
1098	8.0	0.0	7.0	0.721	9	...	2.0	3.0	1.0	7.0	
1099	4.0	1.0	8.0	0.307	10	...	0.0	7.0	1.0	7.0	