

## ▼ Imports and Functions

### ▼ Imports

```
!pip install pca

[?] Collecting pca
  Downloading pca-2.0.3-py3-none-any.whl (34 kB)
Collecting datazets (from pca)
  Downloading datazets-0.1.6-py3-none-any.whl (10 kB)
Requirement already satisfied: statsmodels in /usr/local/lib/python3.10/dist-packages (from pca) (0.13.5)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from pca) (3.7.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from pca) (1.22.4)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (from pca) (1.2.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from pca) (1.10.1)
Collecting colourmap>=1.1.14 (from pca)
  Downloading colourmap-1.1.14-py3-none-any.whl (9.2 kB)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from pca) (1.5.3)
Collecting scatterd>=1.3.1 (from pca)
  Downloading scatterd-1.3.5-py3-none-any.whl (11 kB)
Collecting adjusttext (from pca)
  Downloading adjustText-0.8-py3-none-any.whl (9.1 kB)
Requirement already satisfied: seaborn in /usr/local/lib/python3.10/dist-packages (from scatterd>=1.3.1->pca) (0.12.2)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from scatterd>=1.3.1->pca) (2.27.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pca) (1.1.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pca) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pca) (4.40.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pca) (1.4.4)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pca) (23.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pca) (8.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pca) (3.1.0)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pca) (2.8)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas->pca) (2022.7.1)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn->pca) (1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn->pca) (3)
Requirement already satisfied: patsy>=0.5.2 in /usr/local/lib/python3.10/dist-packages (from statsmodels->pca) (0.5.3)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from patsy>=0.5.2->statsmodels->pca) (1.16.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->scatterd>=1)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->scatterd>=1.3)
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.10/dist-packages (from requests->scatterd>=1.3)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->scatterd>=1.3.1->pca)
Installing collected packages: datazets, colourmap, adjusttext, scatterd, pca
Successfully installed adjusttext-0.8 colourmap-1.1.14 datazets-0.1.6 pca-2.0.3 scatterd-1.3.5
```

```
# importing packages
import pandas as pd
import glob
import numpy as np
import scipy as sp
from scipy.stats import kurtosis, skew
# Importing matplotlib to plot the graphs.
import matplotlib.pyplot as plt
from datetime import datetime, date, time, timezone
from natsort import index_natsorted
from sklearn.decomposition import PCA as sk1PCA
from scipy.fft import fft, fftfreq, rfft, rfftfreq
from pca import pca as betterPCA

import seaborn as sns
```

### ▼ Old Functions

```

def rms(array):
    return np.sqrt(np.mean(array**2))

def computeRMS(df, filename):
    # For the df given, index = filename, column = channel
    index = [filename]
    columns = range(0,df.shape[1])

    result = pd.DataFrame(columns = columns,
                          index = index)

    for i in range(0,df.shape[1]):
        data = rms(df.iloc[:,i])
        result.loc[filename,i] = data

    return result

def computeFilesRMS(path, amount = -1):
    # First get the files
    folder_path = path
    file_list = glob.glob(folder_path + "/")

    filenames = []
    for filepath in file_list:
        filenames.append(filepath.split("/")[-1])

    print("Progress: Done loading files")

    # Do the task for the first file
    df = pd.DataFrame(pd.read_table(file_list[0], header = None))
    res = computeRMS(df, filename = filenames[0])

    # And then for the rest
    if (amount < 0 or amount > len(file_list)):
        amount = len(file_list)
    print("Progress: {} / {}".format(1,amount))

    for i in range(1, amount):
        temp_df = computeRMS(
            pd.DataFrame(pd.read_table(file_list[i], header = None)),
            filename = filenames[i],
        )
        res = res.append(temp_df)
        print("Progress: {} / {}".format(i+1,amount))

    return res

```

## ▼ New Functions

```

def computeRMSFFT(data, filename):
    # For the df given, index = filename, column = channel
    mean = rms(data.iloc[:])

    normalized_tone = np.int16((data / data.max()) * 32767)

    SAMPLE_RATE = len(data)
    DURATION = 1
    N = SAMPLE_RATE * DURATION

    yf = rfft(normalized_tone)
    xf = rfftfreq(N, 1 / SAMPLE_RATE)

    return mean, xf, yf

def computeFilesRMSFFT(path, amount = -1, channel=1):
    channel = -1
    # First get the files
    folder_path = path
    file_list = glob.glob(folder_path + "/")

```

```

filenames = []
for filepath in file_list:
    filenames.append(filepath.split("/")[-1])

print("Progress: Done loading files")

# Do the task for the first file
df = pd.DataFrame(pd.read_table(file_list[0], header = None))
df = df.iloc[:, channel]

rms, xf, yf = computeRMSFFT(df, filename = filenames[0])

res1 = pd.DataFrame(columns = ["RMS"], data=[rms])
res2 = pd.DataFrame(columns = xf, data = [yf])

# And then for the rest
if (amount <0 or amount > len(file_list)):
    amount = len(file_list)
print("Progress: {}/{}.".format(1,amount))
for i in range(1, amount):
    temp_df = pd.DataFrame(pd.read_table(file_list[i], header = None))
    temp_df = temp_df.iloc[:, channel]
    rms, xf, yf = computeRMSFFT(
        temp_df,
        filename = filenames[i],
    )
    temp_df2 = pd.DataFrame(columns = ["RMS"], data=[rms])

    res1 = res1.append(temp_df2, ignore_index = True)
    res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)

    print("Progress: {}/{}.".format(i+1,amount))

# Put everything together
print(res1)
print(res2)
print("Progress: Wrapping up the results...")
res = pd.concat([res1,res2],axis=1)

return res

```

```

def parseFilenameToDate(Filename):
    new_str = Filename.replace(".", "/")
    dt = datetime.strptime(new_str, "%Y/%m/%d/%H/%M/%S")
    return dt

```

## ▼ Better Functions

```

def speed_rms_from_acceleration(data, fs):
    """
    Data is an array like, fs the sampling frequency

    Data is assumed that is measured in g
    """
    # First center the data using the mean to cancel out gravity
    mean_values = data.mean()
    data = data - mean_values

    # Perform the operations
    G = 9.80665
    accel = G * data * 1000

    velocity = sp.integrate.cumulative_trapezoid(accel, dx = 1/fs, initial = 0)

    return rms(velocity)

```

```

def computeRMSFFTbetter(data, filename):
    # For the df given, index = filename, column = channel
    SAMPLE_RATE = len(data)

```

```

mean = speed_rms_from_acceleration(data.iloc[:, SAMPLE_RATE])

normalized_tone = np.int16((data / data.max()) * 32767)

DURATION = 1
N = SAMPLE_RATE * DURATION

yf = rfft(normalized_tone)
xf = rfftfreq(N, 1 / SAMPLE_RATE)

yf = np.abs(yf)

columns = ["Filename", "RMS"]
mean2 = pd.DataFrame(columns = columns, data = [[filename, mean]])

return mean2, xf, yf

def computeFilesRMSFFTbetter(path, amount = -1, channel=1):
    channel = channel - 1
    # First get the files
    folder_path = path
    file_list = glob.glob(folder_path + "/*")

    filenames = []
    for filepath in file_list:
        filenames.append(filepath.split("/")[-1])

    print("Progress: Done loading files")

    # Do the task for the first file
    df = pd.DataFrame(pd.read_table(file_list[0], header = None))
    df = df.iloc[:, channel]

    rms, xf, yf = computeRMSFFTbetter(df, filename = filenames[0])

    print(rms)

    res1 = rms
    res2 = pd.DataFrame(columns = xf, data = [yf])

    # And then for the rest
    if (amount < 0 or amount > len(file_list)):
        amount = len(file_list)
    print("Progress: {} / {}".format(1,amount))
    for i in range(1, amount):
        temp_df = pd.DataFrame(pd.read_table(file_list[i], header = None))
        temp_df = temp_df.iloc[:, channel]
        rms, xf, yf = computeRMSFFTbetter(
            temp_df,
            filename = filenames[i],
        )
        temp_df2 = rms

        res1 = res1.append(temp_df2, ignore_index = True)
        res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)

    print("Progress: {} / {}".format(i+1,amount))

    # Put everything together
    print(res1)
    print(res2)
    print("Progress: Wrapping up the results...")
    res = pd.concat([res1,res2],axis=1)

    return res

```

## ▼ ISO criteria (RMS Peak)

```

def fromRMSToMedium(value):
    if (value < 0.10):
        return "Good"
    if (value < 0.25):

```

```

        return "Satisfactory"
    if (value < 0.62):
        return "Unsatisfactory (alert)"
    else:
        return "Unacceptable (danger)"

def fromRMStoSmall(value):
    if (value < 0.06):
        return "Good"
    if (value < 0.16):
        return "Satisfactory"
    if (value < 0.40):
        return "Unsatisfactory (alert)"
    else:
        return "Unacceptable (danger)"

def fromRMStoExtraSmall(value):
    if (value < 0.02):
        return "Under 0.02"
    if (value < 0.04):
        return "Under 0.04"
    if (value < 0.06):
        return "Under 0.06"
    if (value < 0.07):
        return "Under 0.07"
    if (value < 0.08):
        return "Under 0.08"
    if (value < 0.09):
        return "Under 0.09"
    else:
        return "Others"

def fromRMStoSmallMedium(value):
    if (value < 0.06):
        return "Very Good (0.06)"
    if (value < 0.10):
        return "Good"
    if (value < 0.16):
        return "Satisfactory"
    if (value < 0.25):
        return "Barely satisfactory"
    if (value < 0.40):
        return "Unsatisfactory (alert)"
    if (value < 0.62):
        return "Very Unsatisfactory (alert)"
    else:
        return "Unacceptable (danger)"

def tokenizeRMS(column):
    """Transforms a column of a df into a tokenized column, according to ISO-10816"""
    # Extra detail for Small Machines
    #result = column.apply(fromRMStoExtraSmall)

    # For Small Machine Class I
    #result = column.apply(fromRMStoSmall)

    # For Medium Machine Class II
    #result = column.apply(fromRMStoMedium)

    # Mix between Small and Medium (More Information)
    result = column.apply(fromRMStoSmallMedium)

    return result

```

## ▼ RMS Criteria (mm/s RMS)

```

def new_fromRMStoMedium(value):
    if (value < 1.80):
        return "Good"
    if (value < 4.50):
        return "Satisfactory"

```

```

if (value < 11.20):
    return "Unsatisfactory (alert)"
else:
    return "Unacceptable (danger)"

def new_tokenizeRMS(column):
    """
    Transforms a column of a df into a tokenized column
    according to ISO-10816, RMS mm/s
    """
    # For Medium Machine Class II
    result = column.apply(new_fromRMSToMedium)

    return result

```

## ▼ Testing the Functions

This can be left out of execution

```

res = tokenizeRMS(Y)

-----
NameError                                 Traceback (most recent call last)
<ipython-input-17-1270d97751cc> in <cell line: 1>()
----> 1 res = tokenizeRMS(Y)

NameError: name 'Y' is not defined

```

BUSCAR EN STACK OVERFLOW

res

```

test = pd.DataFrame(columns = range(0,5), data=[range(0,5)])
test

```

No se están guardando bien los valores, algo pasa con los indices y las columnas

```

res = computeFilesRMSFFT('/content/drive/MyDrive/test1', amount= 10, channel = 1)

res

pd.DataFrame(columns = ["1","2"], data = [{"filename": "mean"}])

result = computeFilesRMSFFTbetter('/content/drive/MyDrive/test1', amount=10, channel = 5)

result

```

## ▼ Compute Dataset

With the new function, calculate the dataset with all the info and save it in fftRMS-testX-CHY.csv

Only meant to be executed once

```
result = computeFilesRMSFFT('/content/drive/MyDrive/test1', channel = 5)
```

```

Progress: Done loading files
Progress: 1/2156
Progress: 2/2156
Progress: 3/2156
Progress: 4/2156
Progress: 5/2156
Progress: 6/2156
Progress: 7/2156
Progress: 8/2156
Progress: 9/2156
Progress: 10/2156
Progress: 11/2156

```

```
Progress: 12/2156
Progress: 13/2156
Progress: 14/2156
Progress: 15/2156
Progress: 16/2156
Progress: 17/2156
Progress: 18/2156
Progress: 19/2156
Progress: 20/2156
Progress: 21/2156
Progress: 22/2156
Progress: 23/2156
Progress: 24/2156
Progress: 25/2156
Progress: 26/2156
Progress: 27/2156
Progress: 28/2156
Progress: 29/2156
Progress: 30/2156
Progress: 31/2156
Progress: 32/2156
Progress: 33/2156
Progress: 34/2156
Progress: 35/2156
Progress: 36/2156
Progress: 37/2156
Progress: 38/2156
Progress: 39/2156
Progress: 40/2156
Progress: 41/2156
Progress: 42/2156
Progress: 43/2156
Progress: 44/2156
Progress: 45/2156
Progress: 46/2156
Progress: 47/2156
Progress: 48/2156
Progress: 49/2156
Progress: 50/2156
Progress: 51/2156
Progress: 52/2156
Progress: 53/2156
Progress: 54/2156
Progress: 55/2156
Progress: 56/2156
Progress: 57/2156
```

result

	RMS	0.0	1.0	2.0	3.0
0	0.138225	-1.320164e+08+0.000000e+00j	2.267193e+05+3.492255e+05j	-5.372242e+05+8.345162e+05j	4.627124e+05+4.350854e+05
1	0.139235	-2.463488e+08+0.000000e+00j	5.260366e+05+3.054365e+06j	-5.871066e+04+1.171528e+06j	-6.997404e+05 6.503697e+05
2	0.140671	-2.475478e+08+0.000000e+00j	4.187243e+05+1.654904e+06j	1.075200e+06+2.012796e+06j	8.881755e+05-4.017281e+04
3	0.138306	-2.511538e+08+0.000000e+00j	-8.346629e+05+8.734368e+05j	-3.673626e+06+2.909342e+06j	-2.012035e+06 7.558679e+05
4	0.138224	-1.532273e+08+0.000000e+00j	-2.306992e+05+6.869863e+05j	-4.675913e+05+1.268478e+06j	1.099824e+06+4.811586e+05
...	...	...	...	...	...
2151	0.136956	-1.809671e+08+0.000000e+00j	-5.378152e+04+2.226614e+05j	-7.280349e+05+1.565110e+06j	1.299636e+06-2.667688e+05
2152	0.115042	-2.123740e+08+0.000000e+00j	-1.858666e+06+1.608982e+06j	-9.116272e+05+1.437034e+05j	2.407033e+05+2.129536e+06
2153	0.115306	-1.784532e+08+0.000000e+00j	-1.879803e+06+1.285439e+06j	5.727774e+05+7.880351e+03j	4.729190e+05+4.768111e+05
2154	0.115713	-1.824240e+08+0.000000e+00j	-1.360917e+06+2.418075e+06j	-1.444286e+05+1.092288e+06j	-5.362432e+05 5.141278e+04
2155	0.136412	-1.734211e+08+0.000000e+00j	-5.165542e+05+7.720937e+05j	-8.237782e+05+1.955373e+06j	-9.006567e+04 1.762546e+05

2156 rows × 10242 columns

```
result.to_csv("/content/drive/MyDrive/fftRMS-probandotest1.csv")
```

Same, but with the better function





```

Progress: 8/2156
<ipython-input-55-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas
    res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
<ipython-input-55-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas
    res1 = res1.append(temp_df2, ignore_index = True)
Progress: 9/2156
<ipython-input-55-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas
    res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
Progress: 10/2156
<ipython-input-55-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas
    res1 = res1.append(temp_df2, ignore_index = True)
<ipython-input-55-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas
    res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
Progress: 11/2156
<ipython-input-55-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas
    res1 = res1.append(temp_df2, ignore_index = True)
<ipython-input-55-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas

```

result

	Filename	RMS	0.0	1.0	2.0	3.0	4.0	5.0
0	2003.11.16.06.38.46	0.162778	112382607.0	5.251097e+05	1.007576e+06	6.417581e+05	3.878949e+05	3.000966e+05
1	2003.11.16.07.28.46	0.160860	116223318.0	1.096605e+06	3.027029e+05	5.355070e+05	4.298853e+05	2.845411e+05
2	2003.11.16.06.28.46	0.162380	170263650.0	2.201489e+06	7.231786e+05	1.833845e+06	1.118220e+06	1.390141e+06
3	2003.11.16.08.18.46	0.160941	125443278.0	3.040756e+05	7.183433e+05	7.288441e+05	6.183690e+05	7.198218e+05
4	2003.11.16.07.38.46	0.161260	99094220.0	9.598522e+05	7.007767e+05	4.588762e+05	4.939984e+05	3.038675e+05
...	...	...	...	...	...	...	...	...
2151	2003.10.29.14.39.46	0.159728	191054085.0	1.885194e+06	3.858770e+06	1.807041e+06	1.335664e+06	1.794642e+06
2152	2003.10.23.10.04.13	0.131624	184262107.0	1.896252e+06	6.821482e+05	3.015916e+06	1.705190e+06	1.156464e+06
2153	2003.10.23.08.44.13	0.131232	127782936.0	1.054726e+06	1.134781e+06	5.808984e+05	3.467499e+05	9.052551e+05
2154	2003.10.23.09.24.13	0.132458	148794651.0	1.503662e+06	1.027029e+06	5.241795e+05	3.545158e+05	5.017679e+05
2155	2003.10.29.14.49.46	0.158270	136161777.0	4.328283e+05	1.517314e+06	8.308056e+05	8.628585e+05	6.313178e+05

2156 rows × 10243 columns

```
result = computeFilesRMSFFTbetter('/content/drive/MyDrive/test1', channel = 5, amount=10)
```

```

Progress: Done loading files
      Filename      RMS
0  2003.11.16.06.38.46  2.452215
Progress: 1/10
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas
    res1 = res1.append(temp_df2, ignore_index = True)
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas
    res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
Progress: 2/10
Progress: 3/10
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas
    res1 = res1.append(temp_df2, ignore_index = True)
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas
    res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas
    res1 = res1.append(temp_df2, ignore_index = True)
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas
    res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas
    res1 = res1.append(temp_df2, ignore_index = True)
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas
    res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
Progress: 4/10
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas
    res1 = res1.append(temp_df2, ignore_index = True)
Progress: 5/10
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas
    res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas
    res1 = res1.append(temp_df2, ignore_index = True)
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas
    res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
Progress: 6/10
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas
    res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas
    res1 = res1.append(temp_df2, ignore_index = True)

```

```

    res1 = res1.append(temp_df2, ignore_index = True)
Progress: 7/10
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas
    res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas
    res1 = res1.append(temp_df2, ignore_index = True)
Progress: 8/10
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas
    res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas
    res1 = res1.append(temp_df2, ignore_index = True)
Progress: 9/10
Progress: 10/10
      Filename      RMS
0  2003.11.16.06.38.46  2.452215
1  2003.11.16.07.28.46  5.139736
2  2003.11.16.06.28.46  4.141117
3  2003.11.16.08.18.46  2.781673
4  2003.11.16.07.38.46  4.906813
5  2003.11.16.08.28.46  5.609575
6  2003.11.16.08.08.46  2.020371
7  2003.11.16.07.18.46  3.662010
8  2003.11.16.10.38.46  2.432886

```

result

	Filename	RMS	0.0	1.0	2.0	3.0	4.0	5.0	6.0
0	2003.11.16.06.38.46	2.452215	112382607.0	5.251097e+05	1.007576e+06	6.417581e+05	3.878949e+05	3.000966e+05	7.715366e+05
1	2003.11.16.07.28.46	5.139736	116223318.0	1.096605e+06	3.027029e+05	5.355070e+05	4.298853e+05	2.845411e+05	2.401439e+05
2	2003.11.16.06.28.46	4.141117	170263650.0	2.201489e+06	7.231786e+05	1.833845e+06	1.118220e+06	1.390141e+06	4.214828e+05
3	2003.11.16.08.18.46	2.781673	125443278.0	3.040756e+05	7.183433e+05	7.288441e+05	6.183690e+05	7.198218e+05	1.341246e+05
4	2003.11.16.07.38.46	4.906813	99094220.0	9.598522e+05	7.007767e+05	4.588762e+05	4.939984e+05	3.038675e+05	5.315657e+05
5	2003.11.16.08.28.46	5.609575	118779777.0	1.796949e+06	2.775060e+05	4.627790e+05	3.621752e+05	1.738021e+06	7.300260e+05
6	2003.11.16.08.08.46	2.020371	138071291.0	3.036110e+05	1.197513e+06	7.505787e+05	1.546466e+05	1.133583e+06	2.537985e+05
7	2003.11.16.07.18.46	3.662010	176844211.0	7.734716e+05	1.194564e+06	1.336130e+06	9.135430e+05	1.696695e+06	3.995375e+05
8	2003.11.16.09.08.46	3.585637	123243986.0	2.361027e+05	9.952148e+05	1.126314e+06	1.330815e+06	6.092705e+05	9.323621e+05
9	2003.11.16.10.38.46	2.432886	97241434.0	2.816409e+05	9.146980e+05	1.070925e+06	8.237442e+05	1.992226e+05	4.651680e+05

10 rows × 10243 columns

```

result = computeFilesRMSFFTbetter('/content/drive/MyDrive/test1', channel = 5, amount=10)
result

```

```

Progress: Done loading files
      Filename          RMS
0  2003.11.16.06.38.46  645.076078
Progress: 1/10
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res1 = res1.append(temp_df2, ignore_index = True)
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
Progress: 2/10
Progress: 3/10
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res1 = res1.append(temp_df2, ignore_index = True)
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res1 = res1.append(temp_df2, ignore_index = True)
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res1 = res1.append(temp_df2, ignore_index = True)
Progress: 4/10
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res1 = res1.append(temp_df2, ignore_index = True)
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
Progress: 5/10
Progress: 6/10
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res1 = res1.append(temp_df2, ignore_index = True)
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res1 = res1.append(temp_df2, ignore_index = True)
Progress: 7/10
Progress: 8/10
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res1 = res1.append(temp_df2, ignore_index = True)
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res1 = res1.append(temp_df2, ignore_index = True)
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
Progress: 9/10
Progress: 10/10
      Filename          RMS
0  2003.11.16.06.38.46  645.076078
1  2003.11.16.07.28.46  655.712869
2  2003.11.16.06.28.46  655.095704
3  2003.11.16.08.18.46  649.823013
4  2003.11.16.07.38.46  647.501314
5  2003.11.16.08.28.46  657.322479
6  2003.11.16.08.08.46  649.022553
7  2003.11.16.07.18.46  645.056566
8  2003.11.16.09.08.46  645.649241
9  2003.11.16.10.38.46  645.696442
      0.0       1.0       2.0       3.0       4.0      \
0  112382607.0  5.251097e+05  1.007576e+06  6.417581e+05  3.878949e+05
1  116223318.0  1.096605e+06  3.027029e+05  5.355070e+05  4.298853e+05
2  170263650.0  2.201489e+06  7.231786e+05  1.833845e+06  1.118220e+06
3  125443278.0  3.040756e+05  7.183433e+05  7.288441e+05  6.183690e+05
4  99094220.0   9.598522e+05  7.007767e+05  4.588762e+05  4.939984e+05
5  118779777.0  1.796949e+06  2.775060e+05  4.627790e+05  3.621752e+05
6  138071291.0  3.036110e+05  1.197513e+06  7.505787e+05  1.546466e+05
7  176844211.0  7.734716e+05  1.194564e+06  1.336130e+06  9.135430e+05
8  123243980.0  2.361027e+05  9.952148e+05  1.126314e+06  1.330815e+06
9  97241434.0   2.816409e+05  9.146980e+05  1.070925e+06  8.237442e+05

      5.0       6.0       7.0       8.0       9.0      \
0  3.000966e+05  7.715366e+05  146574.051666  3.481314e+05  834529.197877
1  2.845411e+05  2.401439e+05  851738.877201  4.449902e+05  647811.259441
2  1.390141e+06  4.214828e+05  195077.086086  1.572459e+06  675969.324429
3  7.198218e+05  1.341246e+06  797807.268300  2.766471e+05  310361.722120
4  3.038675e+05  5.315657e+05  762735.172306  5.015622e+05  592880.687185
5  1.738021e+06  7.300260e+05  524023.409937  7.937528e+05  288244.115971
6  1.133583e+06  2.537985e+05  554059.620066  6.717955e+05  415272.865760
7  1.696695e+06  3.995375e+05  589848.436717  3.420603e+06  493236.455167
8  6.092705e+05  9.323621e+05  324304.678170  9.694686e+05  246176.535093
9  1.992226e+05  4.651680e+05  232894.675609  3.894273e+05  469616.539124

```

		۰.۱۵۷۲۷۴۱	۰.۷۷۲۷۰۱	۰.۳۷۷۲۷۰۱	۰.۰۶۷۲۷۰۱	۰.۴۴۷۲۷۰۱	\
۰	...	2.147986e+05	263485.634138	412473.137408	3.824612e+05		
۱	...	3.493010e+05	200481.937177	174001.028441	3.971850e+05		
۲	...	7.423671e+05	859327.890393	826167.640523	7.829263e+05		
۳	...	8.411353e+05	554435.785461	745246.433183	1.322200e+05		
۴	...	4.221061e+05	134414.794783	244430.153471	8.207775e+05		
۵	...	1.750191e+05	559814.748083	74763.061337	3.942516e+05		
۶	...	2.003443e+05	510011.029981	442312.590214	9.663688e+05		
۷	...	1.523036e+06	852440.376296	810418.573088	1.427358e+06		
۸	...	5.762015e+05	394049.215533	885433.068054	6.444331e+05		
۹	...	4.298477e+05	279624.763983	307287.756569	3.535598e+05		

		۱۰۲۳۵.۰	۱۰۲۳۶.۰	۱۰۲۳۷.۰	۱۰۲۳۸.۰	۱۰۲۳۹.۰	\
۰	4.440119e+05	7.007581e+05	452243.316939	1.961439e+05	271148.593111		
۱	2.935413e+05	6.240676e+05	307914.226043	3.284071e+05	392658.513855		
۲	3.922242e+05	1.006343e+06	875086.756451	1.076903e+06	451007.033859		
۳	5.057205e+05	4.909244e+05	353459.978945	8.675352e+05	445328.663944		
۴	3.069563e+05	2.668356e+05	226756.327364	6.746988e+05	463383.271847		
۵	7.558672e+05	7.683846e+05	120856.779717	1.084952e+06	524581.246538		
۶	2.486788e+05	1.140700e+06	162046.623640	2.528359e+05	401036.875015		
۷	1.457648e+06	1.231249e+06	818776.486095	1.164744e+06	496008.685367		
۸	2.518661e+05	7.367486e+05	660476.937514	5.555124e+05	522687.846217		
۹	4.546060e+05	4.830596e+05	275249.608593	1.790385e+05	161480.708942		

		۱۰۲۴۰.۰
۰	643049.۰	
۱	504696.۰	
۲	700550.۰	
۳	206144.۰	
۴	110182.۰	
۵	42859.۰	
۶	172145.۰	

```
result = computeFilesRMSFFTbetter('/content/drive/MyDrive/3rd_test', channel = 3, amount=10)
result
```

```

Progress: Done loading files
      Filename      RMS
0  2004.04.10.20.51.57  6.87695
Progress: 1/10
Progress: 2/10
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
    res1 = res1.append(temp_df2, ignore_index = True)
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
    res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
    res1 = res1.append(temp_df2, ignore_index = True)
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
    res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
    res1 = res1.append(temp_df2, ignore_index = True)
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
    res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
Progress: 3/10
Progress: 4/10
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
    res1 = res1.append(temp_df2, ignore_index = True)
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
    res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
Progress: 5/10
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
    res1 = res1.append(temp_df2, ignore_index = True)
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
    res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
    res1 = res1.append(temp_df2, ignore_index = True)
Progress: 6/10
Progress: 7/10
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
    res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
    res1 = res1.append(temp_df2, ignore_index = True)
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
    res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
    res1 = res1.append(temp_df2, ignore_index = True)
Progress: 8/10
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
    res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
    res1 = res1.append(temp_df2, ignore_index = True)
Progress: 9/10
Progress: 10/10
      Filename      RMS
0  2004.04.10.20.51.57  6.876950
1  2004.04.10.19.01.57  9.048869
2  2004.04.10.22.01.57  13.578375
3  2004.04.10.21.51.57  11.873315
4  2004.04.10.23.31.57  11.934865
5  2004.04.10.20.31.57  13.706378
6  2004.04.10.20.41.57  11.273408
7  2004.04.10.23.01.57  11.828150
8  2004.04.10.21.31.57  12.415120
9  2004.04.10.20.21.57  9.798522
      0.0      1.0      2.0      3.0      4.0      \
0  2585548.0  320630.631555  6.104331e+05  1.106187e+06  3.601262e+05
1  3791810.0  530556.251499  5.350751e+05  4.388777e+05  9.217964e+05
2  4364429.0  764308.513841  7.056213e+05  1.091912e+06  7.180075e+05
3  4603726.0  237330.488443  1.029207e+06  4.611465e+05  1.034588e+06
4  3196963.0  434394.173347  6.775690e+05  5.283433e+05  8.615319e+05
5  4566870.0  332237.916503  5.547874e+05  8.388096e+05  4.349426e+05
6  4183334.0  551724.934388  1.820933e+06  7.292423e+05  5.583775e+05
7  3897631.0  902878.895639  4.319061e+05  1.969255e+06  6.042036e+05
8  5057580.0  998725.967995  1.094233e+06  8.262256e+05  4.418027e+05
9  3412566.0  819421.321700  8.596838e+05  1.087779e+06  9.593036e+05
      5.0      6.0      7.0      8.0      9.0      ...      \
0  4.478630e+05  1.015797e+06  7.075479e+04  5.215831e+05  8.657247e+05  ...
1  1.278354e+06  5.452965e+05  7.485325e+05  8.994027e+05  3.806340e+05  ...
2  1.252435e+06  6.392894e+05  1.769737e+06  1.212216e+06  7.073338e+05  ...
3  6.858914e+05  6.827317e+05  8.369891e+05  4.179313e+05  6.727194e+05  ...
4  5.809486e+05  2.205649e+05  3.609792e+05  5.626613e+05  1.038944e+06  ...
5  5.922124e+05  6.551139e+05  1.193108e+06  4.040229e+05  4.659772e+05  ...
6  1.159532e+06  3.355016e+05  8.267079e+05  1.867494e+05  4.934797e+05  ...
7  8.376575e+05  1.074734e+06  6.606502e+05  8.380635e+05  2.184956e+05  ...
8  8.819547e+05  9.178601e+05  7.421775e+05  5.097198e+05  9.994744e+05  ...
9  1.014108e+06  3.132024e+05  4.858284e+05  1.217160e+06  4.023880e+05  ...
      10231.0     10232.0     10233.0     10234.0     10235.0      \
0  317534.815654  256569.473038  262252.192832  7.150794e+05  3.087552e+05

```

```
result = computeFilesRMSFFTbetter('/content/drive/MyDrive/3rd_test', channel = 3, amount=10)
result
```

```

Progress: Done loading files
      Filename      RMS
0  2004.04.10.20.51.57  0.974655
Progress: 1/10
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res1 = res1.append(temp_df2, ignore_index = True)
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res1 = res1.append(temp_df2, ignore_index = True)
Progress: 2/10
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res1 = res1.append(temp_df2, ignore_index = True)
Progress: 3/10
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res1 = res1.append(temp_df2, ignore_index = True)
Progress: 4/10
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res1 = res1.append(temp_df2, ignore_index = True)
Progress: 5/10
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res1 = res1.append(temp_df2, ignore_index = True)
Progress: 6/10
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res1 = res1.append(temp_df2, ignore_index = True)
Progress: 7/10
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res1 = res1.append(temp_df2, ignore_index = True)
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
Progress: 8/10
Progress: 9/10
<ipython-input-26-2e81a3400b3e>:41: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res1 = res1.append(temp_df2, ignore_index = True)
<ipython-input-26-2e81a3400b3e>:42: FutureWarning: The frame.append method is deprecated and will be removed from pandas in
  res2 = res2.append(pd.DataFrame(columns = xf, data = [yf]), ignore_index = True)
Progress: 10/10
      Filename      RMS
0  2004.04.10.20.51.57  0.974655
1  2004.04.10.19.01.57  1.013293

```

## ► First try:

In this try, the ISO RMS criteria was used incorrectly, meaning that the colours of the graphs are not right (although surprisingly similar to the actual real results)

The graphs contained in this section should be used with this fact in mind, but they were left here to show the development process of the project.

```
[ ] ↴ 84 celdas ocultas
A  2106063  B  121204  173317  C  77560001AE  E  28243301AE  O  51E21001AE
```

## ▼ Second Try

The mistake made previously over the RMS criteria was corrected, so these graphs are the ones that are seen in the documentation.

Choose the file below:

- Test 1 CH5
- Test 2 CH1
- Test 3 CH3

Example: "test3-CH3"

```
o 0.01754/ET+03  9.1/0001E+03  /421//3E+03  3.07/170E+03  3.334/44E+03  ...
```

```
test_to_check = "test3-CH3"
.....
```

```
A.45274T A.5274AT A.5C74AT A.7C74AT A.9C74AT
```

## ▼ Load Dataset

```
A.45274T A.5274AT A.5C74AT A.7C74AT A.9C74AT \
```

```
datafile = f"/content/drive/MyDrive/fftRMS-{test_to_check}-fixed.csv"
data = pd.read_csv(datafile)

Y = data.loc[:, "RMS"]
X = data.iloc[:, 3:]

Y

0      1.579103
1      0.827692
2      0.940373
3      1.173243
4      0.864917
...
979     5.762988
980     5.626528
981     5.646649
982    12.529171
983     4.206188
Name: RMS, Length: 984, dtype: float64
```

```
 2004.04.10.21.51.57 0.841561 1603726.0 237230.488443 1.029207e+06 4.611465e+05 1.034588e+06 6.858011e+05 6.827317e+05
```

```
data = data.iloc[:, 1:]
```

```
reordered_data = data.sort_values(
    by="Filename",
    key=lambda x: np.argsort(index_natsorted(data["Filename"])))
)
```

```
8 2004.04.10.21.31.57 1.329275 5057580.0 998725.967995 1.094233e+06 8.262256e+05 4.418027e+05 8.819547e+05 9.178601e+05
```

```
reordered_data.loc[:, "Filename"] = reordered_data.loc[:, "Filename"].apply(parseFilenameToDate)
```

```
<ipython-input-137-256ec5d2b8df>:1: DeprecationWarning: In a future version, `df.iloc[:, i] = newvals` will attempt to set
reordered_data.loc[:, "Filename"] = reordered_data.loc[:, "Filename"].apply(parseFilenameToDate)
```

```
reordered_data
```

	<b>Filename</b>	<b>RMS</b>	<b>0.0</b>	<b>1.0</b>	<b>2.0</b>	<b>3.0</b>	<b>4.0</b>	<b>5.0</b>	<b>6.0</b>
7	2004-02-12 10:32:39	1.298048	15069683.0	2.035246e+05	5.937756e+05	6.605724e+05	6.630525e+05	5.365002e+05	7.593722e+05
0	2004-02-12 10:42:39	1.579103	4635277.0	7.149984e+05	3.434028e+05	6.732487e+05	1.515483e+05	7.289445e+05	4.137006e+05
17	2004-02-12 10:52:39	1.384534	3313657.0	1.588991e+05	5.166064e+05	2.239593e+05	2.081808e+05	3.846052e+05	5.875088e+05

reordered\_data

	<b>Filename</b>	<b>RMS</b>	<b>0.0</b>	<b>1.0</b>	<b>2.0</b>	<b>3.0</b>	<b>4.0</b>	<b>5.0</b>	<b>6.0</b>
7	2004-02-12 10:32:39	1.298048	15069683.0	2.035246e+05	5.937756e+05	6.605724e+05	6.630525e+05	5.365002e+05	7.593722e+05
0	2004-02-12 10:42:39	1.579103	4635277.0	7.149984e+05	3.434028e+05	6.732487e+05	1.515483e+05	7.289445e+05	4.137006e+05
17	2004-02-12 10:52:39	1.384534	3313657.0	1.588991e+05	5.166064e+05	2.239593e+05	2.081808e+05	3.846052e+05	5.875088e+05
12	2004-02-12 11:02:39	1.386667	2512340.0	3.501494e+05	4.422848e+05	3.065208e+05	1.042964e+05	3.287442e+05	2.126002e+05
15	2004-02-12 11:12:39	1.448256	4125430.0	4.170901e+05	4.267451e+05	2.355288e+05	7.229696e+05	1.385244e+05	1.053861e+06
...	...	...	...	...	...	...	...	...	...
975	2004-02-19 05:42:39	30.429151	744391.0	1.086981e+06	1.400724e+05	3.627966e+05	3.387068e+05	6.644871e+05	4.744903e+04
967	2004-02-19 05:52:39	5.642043	727141.0	6.736387e+05	5.378731e+05	4.123522e+05	5.768794e+05	5.585799e+05	4.175475e+05
969	2004-02-19 06:02:39	10.660907	130174.0	6.806790e+05	2.047971e+05	3.620101e+05	6.036861e+05	2.372859e+05	3.809290e+05
972	2004-02-19 06:12:39	0.486068	249199420.0	1.312705e+07	3.866019e+06	2.729268e+06	2.248963e+06	1.679738e+06	1.916148e+06
976	2004-02-19 06:22:39	0.454893	388764029.0	2.742618e+07	1.133304e+07	1.206723e+07	5.595371e+06	6.641057e+06	4.992995e+06

984 rows × 10243 columns

```
plt.figure(figsize=(17,6))
plt.plot(reordered_data.Filename, reordered_data["RMS"].values, label = "RMS")
# plt.plot(reordered_data.Filename, reordered_data["1.0"].values, label = "Amplitude 1")
plt.legend()
plt.show()
```



## ▼ Preprocessing

```

from sklearn import preprocessing
X_train = np.array(X)
min_max_scaler = preprocessing.MinMaxScaler()
X_train_minmax = min_max_scaler.fit_transform(X_train)
X_train_minmax

array([[0.0116709 , 0.02483629, 0.0288747 , ..., 0.4893543 , 0.13345338,
       0.07593244],
       [0.00612779, 0.00697081, 0.03045932, ..., 0.15451717, 0.05672485,
       0.05555797],
       [0.00540885, 0.0114299 , 0.04469492, ..., 0.22592355, 0.05269414,
       0.21555764],
       ...,
       [0.00154993, 0.01736089, 0.0152368 , ..., 0.23612505, 0.11789608,
       0.06579102],
       [0.00411912, 0.00567691, 0.0338778 , ..., 0.15616296, 0.15976158,
       0.27798053],
       [0.00394843, 0.00258203, 0.02090133, ..., 0.18827207, 0.11797558,
       0.1131235 ]])

X = X_train_minmax

X

array([[0.0116709 , 0.02483629, 0.0288747 , ..., 0.4893543 , 0.13345338,
       0.07593244],
       [0.00612779, 0.00697081, 0.03045932, ..., 0.15451717, 0.05672485,
       0.05555797],
       [0.00540885, 0.0114299 , 0.04469492, ..., 0.22592355, 0.05269414,
       0.21555764],
       ...,
       [0.00154993, 0.01736089, 0.0152368 , ..., 0.23612505, 0.11789608,
       0.06579102],
       [0.00411912, 0.00567691, 0.0338778 , ..., 0.15616296, 0.15976158,
       0.27798053],
       [0.00394843, 0.00258203, 0.02090133, ..., 0.18827207, 0.11797558,
       0.1131235 ]])

tokenized = new_tokenizeRMS(Y)

tokenized

```

0	Good
1	Good
2	Good
3	Good
4	Good
...	
979	Unsatisfactory (alert)
980	Unsatisfactory (alert)
981	Unsatisfactory (alert)
982	Unacceptable (danger)
983	Satisfactory

Name: RMS, Length: 984, dtype: object

## ▼ Palette for graphs

```

seaborn_palette = sns.color_palette(n_colors=8)
seaborn_palette

```



```

custom_palette = {
    "Good": seaborn_palette[0],
    "Satisfactory" : seaborn_palette[2],
    "Unsatisfactory (alert)" : seaborn_palette[1],
    "Unacceptable (danger)" : seaborn_palette[3]
}

```

## ▼ Using PCA:

### ▼ With sklearn

```

pca = sklPCA(n_components=4)
pca.fit(X)

PCs = pca.components_ # Principal Components
variances = pca.explained_variance_ratio_ # Variances explained by each PC
projection = pca.transform(X) # Get the projection

print("PCs:\n", PCs)
print("Variances explained by each PC:\n", variances)
print("Projection of the data onto the PCs:\n", projection)

PCs:
[[ 0.00183782  0.00197348  0.00287966 ...  0.00662441  0.00509022
  0.0052742 ]
 [ 0.00908458  0.00863611  0.00795448 ...  0.01142757  0.01294989
  0.00863141]
 [ 0.00194313  0.001916     0.00121716 ... -0.00133229  0.00130789
  0.0002831 ]
 [ 0.01391681  0.01286841  0.01081847 ...  0.00509459  0.00937426
 -0.00524206]]
Variances explained by each PC:
[0.10176727 0.01587644 0.00857836 0.00660087]
Projection of the data onto the PCs:
[[ 5.7957058   1.30154086  0.68747849  0.42654868]
 [ 0.79362712 -1.18044128  0.46116297  1.23550663]
 [-1.94387399 -1.50629636  0.05652432  1.48169046]
 ...
 [-6.51695185  0.46056265 -0.60365648  2.71673599]
 [-1.88731775  3.17041406 -0.12333468  4.06682843]
 [ 0.72012426  7.46589127  0.93766811  1.91559261]]

```

Loadings, the components, in absolute value, can show their contribution to the variance and thus showing which are the most important ones:

```

# Get the loadings of the original variables
loadings = pca.components_

# Calculate the absolute values of the loadings
abs_loadings = np.abs(loadings)

# Sum the absolute values of the loadings across the components
total_abs_loadings = np.sum(abs_loadings, axis=0)

# Sort the amplitudes by their total absolute loadings in descending order
sorted_amplitudes = np.argsort(total_abs_loadings)[::-1]

# Print the top n amplitudes
n = 10
print("Top", n, "amplitudes:")
for i in range(n):
    print("Amplitude", sorted_amplitudes[i], "with total absolute loading of", total_abs_loadings[sorted_amplitudes[i]])
# Las amplitudes estan en base 0

    Top 10 amplitudes:
    Amplitude 1888 with total absolute loading of 0.1571409365677623
    Amplitude 4712 with total absolute loading of 0.15004462542920954
    Amplitude 1881 with total absolute loading of 0.14742624794515036
    Amplitude 1895 with total absolute loading of 0.14412833243609768
    Amplitude 1082 with total absolute loading of 0.13872360217003507
    Amplitude 1009 with total absolute loading of 0.136719412525025
    Amplitude 4713 with total absolute loading of 0.1359979384907175
    Amplitude 3297 with total absolute loading of 0.13506544456188652

```

```
Amplitude 4726 with total absolute loading of 0.1331595187070295
Amplitude 3068 with total absolute loading of 0.13309895804876537
```

```
pd.DataFrame(sorted_amplitudes)
```

	θ
0	1888
1	4712
2	1881
3	1895
4	1082
...	...
10236	3780
10237	4975
10238	4267
10239	3320
10240	4716

```
10241 rows × 1 columns
```

```
pd.DataFrame(total_abs_loadings[sorted_amplitudes])
```

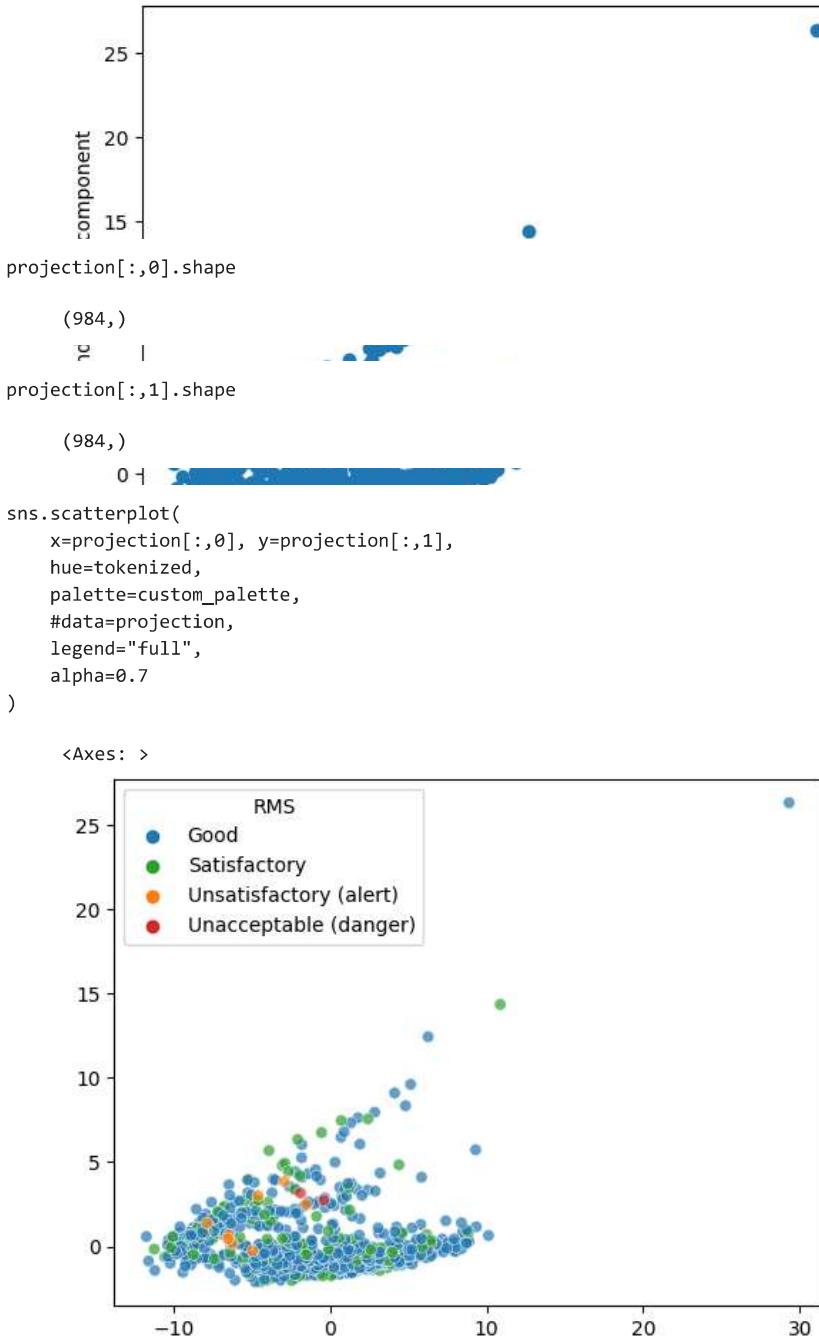
	θ
0	0.157141
1	0.150045
2	0.147426
3	0.144128
4	0.138724
...	...
10236	0.005366
10237	0.005225
10238	0.005212
10239	0.005037
10240	0.003612

```
10241 rows × 1 columns
```

```
np.max(sorted_amplitudes)
```

```
10240
```

```
plt.scatter(projection[:, 0], projection[:, 1], )
plt.xlabel('First principal component')
plt.ylabel('Second principal component')
plt.show()
```



## ▼ With PCA library

[Install and Info](#)

```

model = betterPCA()
pca_res = model.fit_transform(X)

pca_res
[pca] >Column labels are auto-completed.
[pca] >Row labels are auto-completed.
[pca] >The PCA reduction is performed to capture [95.0%] explained variance using the [10241] columns of the input data.
[pca] >Fit using PCA.
[pca] >Compute loadings and PCs.
[pca] >Compute explained variance.
[pca] >Number of components is [4486] that covers the [95.00%] explained variance.
[pca] >The PCA reduction is performed on the [10241] columns of the input dataframe.
[pca] >Fit using PCA.
[pca] >Compute loadings and PCs.
[pca] >Outlier detection using Hotelling T2 test with alpha=[0.05] and n_components=[4486]
[pca] >Multiple test correction applied for Hotelling T2 test: [fdr_bh]
[pca] >Outlier detection using SPE/DmodX with n_std=[3]
{'loadings':           1         2         3         4         5         6      \n
 PC1   -0.000431 -0.004372 -0.008991 -0.003980 -0.002402 -0.004633 -0.004561

```

```

PC2    -0.000092 -0.001553 -0.006591 -0.002157 -0.000435 -0.001846 -0.002549
PC3    -0.001161 -0.000006  0.000090  0.001517  0.000181  0.000780  0.002129
PC4    -0.000582  0.002262  0.001907  0.000395  0.001628  0.001500  0.003400
PC5    -0.001272 -0.004276 -0.009122 -0.002687 -0.002524 -0.004443  0.000375
...
...
PC4482  0.000450 -0.015576 -0.017698 -0.007955 -0.001223  0.004644  0.003514
PC4483 -0.002867 -0.000713  0.004017 -0.010342 -0.003117 -0.003857  0.010395
PC4484 -0.003046  0.001240 -0.004862 -0.012548  0.005970  0.006093  0.006788
PC4485  0.002422  0.000717 -0.000687 -0.007259 -0.003889 -0.011219  0.005110
PC4486  0.000253  0.010630 -0.007706 -0.014723 -0.006297  0.013017 -0.002174

          8         9        10     ...      10232      10233      10234  \
PC1    -0.006147 -0.009071 -0.008098 ... -0.010927 -0.010279 -0.011696
PC2    -0.004270 -0.005108 -0.005345 ...  0.010596  0.010193  0.009636
PC3     0.000154 -0.001382 -0.000508 ... -0.001960 -0.002190 -0.004893
PC4     0.002599  0.002495  0.003991 ... -0.009394 -0.010873 -0.007268
PC5    -0.004324  0.000506 -0.004831 ...  0.001287 -0.000968  0.000227
...
...
PC4482 -0.003568 -0.011368  0.001229 ...  0.001237  0.003771  0.001524
PC4483  0.010715  0.006430 -0.002678 ...  0.006251 -0.018960  0.016905
PC4484  0.001989 -0.014730 -0.003656 ... -0.018631 -0.003435 -0.021527
PC4485 -0.011372  0.011111  0.017787 ... -0.018647 -0.012601 -0.005429
PC4486 -0.005503  0.010124  0.012399 ...  0.020092 -0.022638 -0.011063

      10235      10236      10237      10238      10239      10240      10241
PC1    -0.011129 -0.011843 -0.009173 -0.010894 -0.011411 -0.012337 -0.008232
PC2     0.009491  0.013638  0.009119  0.011595  0.011509  0.010511  0.006839
PC3    -0.001519 -0.006251  0.000706 -0.004478 -0.004635 -0.002364 -0.001081
PC4    -0.017839 -0.001254 -0.009254 -0.010561 -0.008231 -0.008226 -0.005788
PC5    -0.001320 -0.000083  0.003148  0.004269 -0.003165 -0.000171  0.006255
...
...
PC4482  0.007263  0.008396 -0.005834 -0.014889 -0.010517 -0.001520  0.006672
PC4483  0.002237 -0.002932 -0.005508 -0.000401  0.009828 -0.002978  0.001275
PC4484  0.019273  0.012884 -0.019091  0.001305 -0.003679  0.000597  0.003006
PC4485 -0.012280  0.003351 -0.002319  0.000191  0.008974  0.001705 -0.022048
PC4486  0.001597 -0.003219  0.013904 -0.010497 -0.011066 -0.003968  0.017385

```

```

[4486 rows x 10241 columns],
'PC':           PC1      PC2      PC3      PC4      PC5      PC6      PC7  \
0     1.820220  1.172266 -0.773599 -0.158146  0.031630  0.092280 -0.038258
1    -2.200881  1.207383 -0.820077 -0.195616  0.334458  0.250416 -0.035178
2    -0.470641  0.936739 -0.590898 -0.295546  0.296927 -0.201278 -0.442139
3    -1.030663  1.232253 -0.564540 -0.424315  0.255667 -0.011472  0.066776

```

```
pca_res["topfeat"].to_csv(f"/content/drive/MyDrive/loadings-PCA-{test_to_check}-fixed.csv")
```

```
pca_res["topfeat"]
```

	PC	feature	loading	type
0	PC1	31	-0.031007	best
1	PC2	1010	-0.152509	best
2	PC3	1018	-0.127041	best
3	PC4	1011	0.208769	best
4	PC5	1003	0.160704	best
...	...	...	...	...
11382	PC2499	10233	0.042450	weak
11383	PC1853	10234	-0.034429	weak
11384	PC720	10235	-0.036912	weak
11385	PC4098	10237	-0.037777	weak
11386	PC2126	10238	-0.039211	weak

```
11387 rows x 4 columns
```

```
pca_res["explained_var"]
```

```
array([0.03872506, 0.04476259, 0.04777566, ..., 0.99999432, 1.        ,
       1.        ])
```

```
pca_res["variance_ratio"]
```

```
array([3.87250613e-02, 6.03753230e-03, 3.01306180e-03, ...,  
5.80450492e-06, 5.68246412e-06, 0.00000000e+00])
```

```
#fig, ax = model.scatter(PC=[0,1])
```

## ▼ Use TSNE

```
from sklearn.manifold import TSNE
```

### ▼ Over everything (It will take its time)

```
tsne = TSNE(n_components = 2)  
tsne_res = tsne.fit_transform(X)
```

```
tsne_res
```

```
array([[-6.438874 ,  7.454953 ],  
[ 1.4325515 ,  2.964208 ],  
[-0.07009482,  1.6300343 ],  
...,  
[ 7.7537665 ,  1.130982 ],  
[ 6.9994955 ,  1.9711753 ],  
[ 7.104163 ,  2.1064413 ]], dtype=float32)
```

```
tsne_df = pd.DataFrame(tsne_res, columns=['tsne-2d-one', 'tsne-2d-two'])
```

```
tsne_df
```

	tsne-2d-one	tsne-2d-two
0	-6.438874	7.454953
1	1.432552	2.964208
2	-0.070095	1.630034
3	0.580156	4.500815
4	-3.166657	4.623641
...	...	...
979	7.095268	0.416556
980	7.542512	0.767212
981	7.753767	1.130982
982	6.999496	1.971175
983	7.104163	2.106441

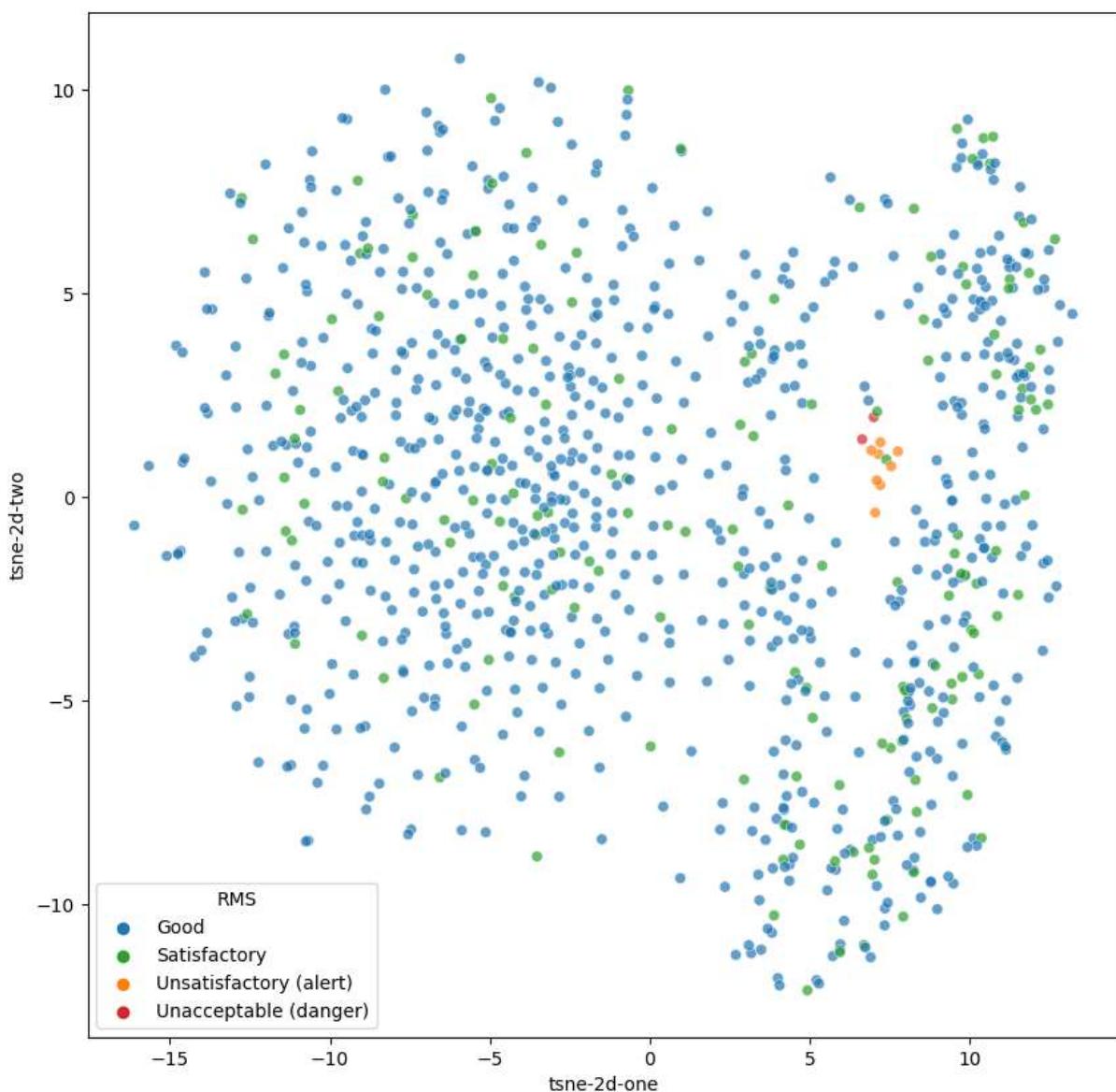
```
984 rows × 2 columns
```

```
tokenized.value_counts()
```

```
Good                      811  
Satisfactory                163  
Unsatisfactory (alert)      8  
Unacceptable (danger)       2  
Name: RMS, dtype: int64
```

```
plt.figure(figsize=(10,10))  
sns.scatterplot(  
    x="tsne-2d-one", y="tsne-2d-two",  
    hue=tokenized,  
    palette=custom_palette,  
    data=tsne_df,  
    legend="full",  
    alpha=0.7  
)
```

<Axes: xlabel='tsne-2d-one', ylabel='tsne-2d-two'>



▼ Over the previous PCA transformation

projection

```
array([[ 5.7957058 ,  1.30154086,  0.68747849,  0.42654868],
       [ 0.79362712, -1.18044128,  0.46116297,  1.23550663],
       [-1.94387399, -1.50629636,  0.05652432,  1.48169046],
       ...,
       [-6.51695185,  0.46056265, -0.60365648,  2.71673599],
       [-1.88731775,  3.17041406, -0.12333468,  4.06682843],
       [ 0.72012426,  7.46589127,  0.93766811,  1.91559261]])
```

```
tsnePCA = TSNE(n_components = 2)
tsnePCA_res = tsnePCA.fit_transform(projection)
```

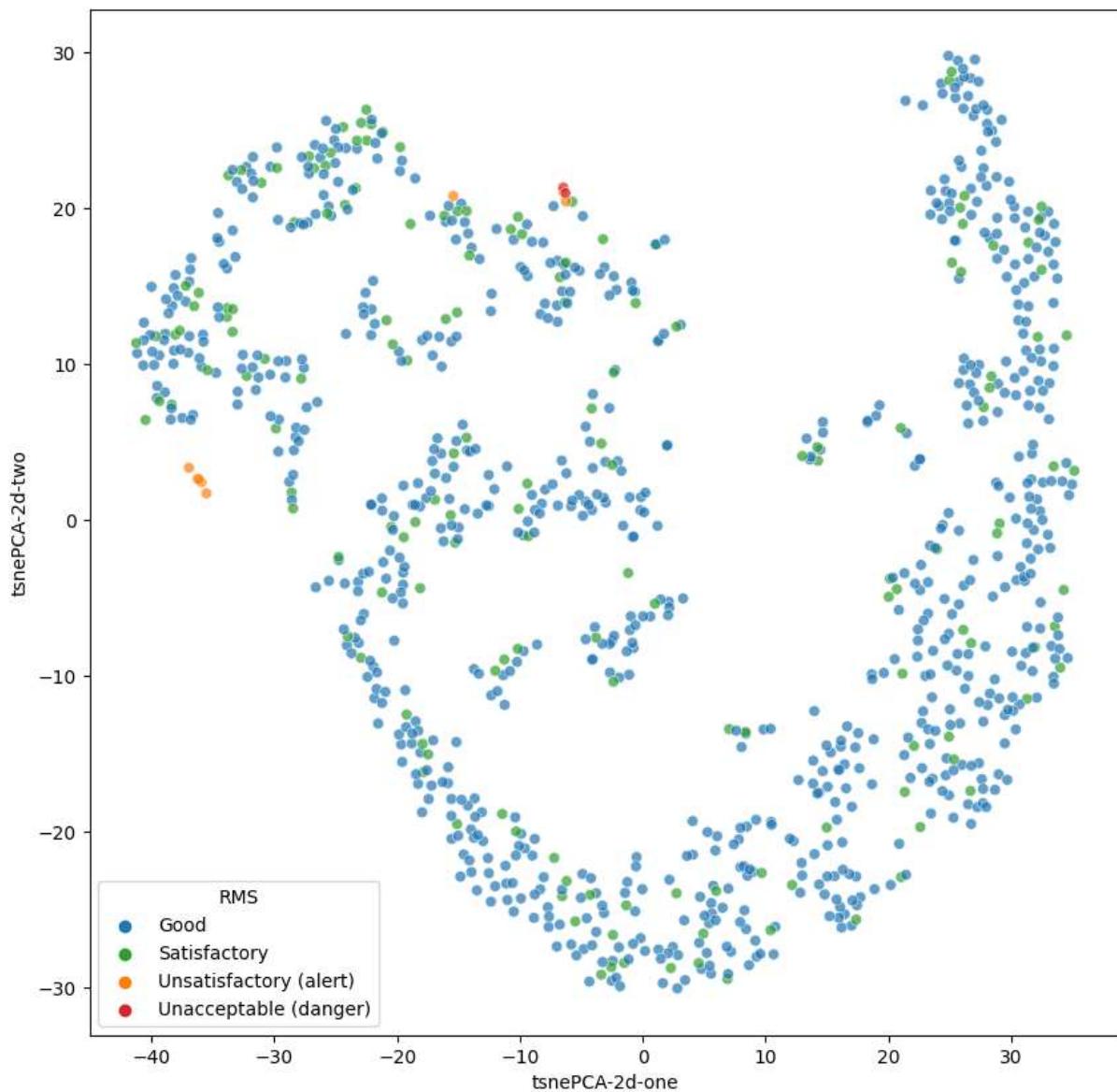
```
tsnePCA_df = pd.DataFrame(tsnePCA_res, columns=['tsnePCA-2d-one', 'tsnePCA-2d-two'])
```

```
plt.figure(figsize=(10,10))
sns.scatterplot(
    x="tsnePCA-2d-one", y="tsnePCA-2d-two",
    hue=tokenized,
    palette=custom_palette,
    data=tsnePCA_df,
    legend="full",
```

```

alpha=0.7
)
<Axes: xlabel='tsnePCA-2d-one', ylabel='tsnePCA-2d-two'>

```



## ▼ Select K Best

```

from sklearn.feature_selection import SelectKBest, chi2

X
array([[ 0.00736069,  0.04810775,  0.19452403, ...,  0.56675017,  0.44233868,
       0.09953225],
       [ 0.01079597,  0.0809075 ,  0.17020873, ...,  0.3260267 ,  0.25828431,
       0.10762901],
       [ 0.01242672,  0.11743003,  0.22523782, ...,  0.08527754,  0.43346584,
       0.25253103],
       ...,
       [ 0.01376968,  0.0713753 ,  0.30501622, ...,  0.04644405,  0.14899647,
       0.39228452],
       [ 0.01646488,  0.25255761,  0.18152931, ...,  0.33813674,  0.15901263,
       0.45059829],
       [ 0.01325791,  0.12042465,  0.30746902, ...,  0.64401745,  0.14364645,
       0.71451115]])

```

```

X_df = pd.DataFrame(X)
X_df

```

	0	1	2	3	4	5	6	7	8	9	...	10231	10232
0	0.007361	0.048108	0.194524	0.160220	0.033383	0.075734	0.167811	0.015068	0.186099	0.270557	...	0.197716	0.152981
1	0.010796	0.080907	0.170209	0.063117	0.086565	0.217150	0.089154	0.180845	0.323066	0.116277	...	0.245482	0.313938
2	0.012427	0.117430	0.225238	0.158143	0.067269	0.212737	0.104867	0.430619	0.436467	0.220182	...	0.463377	0.100093
3	0.013108	0.035093	0.329647	0.066358	0.097244	0.116266	0.112130	0.202480	0.148524	0.209173	...	0.186802	0.588303
4	0.009102	0.065883	0.216186	0.076136	0.080858	0.098396	0.034866	0.086054	0.200991	0.325648	...	0.091967	0.409107
...	...	...	...	...	...	...	...	...	...	...	...	...	...
6319	0.012319	0.085994	0.373771	0.029574	0.039884	0.174746	0.087096	0.142965	0.194748	0.116166	...	0.139316	0.069511
6320	0.014955	0.186089	0.213848	0.054475	0.033261	0.164773	0.141840	0.249907	0.171670	0.130435	...	0.271616	0.218637
6321	0.013770	0.071375	0.305016	0.256772	0.090904	0.105708	0.137344	0.209604	0.587604	0.048507	...	0.297582	0.175091
6322	0.016465	0.252558	0.181529	0.054543	0.033910	0.066582	0.136108	0.286618	0.075933	0.321739	...	0.033209	0.472089
6323	0.013258	0.120425	0.307469	0.290936	0.067240	0.242256	0.163352	0.170224	0.048370	0.181108	...	0.556035	0.434268

```
sel = SelectKBest(chi2,k=10)
X_kbest = sel.fit_transform(X_df, tokenized) # Y.astype('int'))
X_kbest = pd.DataFrame(X_kbest)
X_kbest.head()
```

	0	1	2	3	4	5	6	7	8	9			
0	0.048108	0.009379	0.011868	0.339001	0.014289	0.010078	0.013626	0.020256	0.018578	0.027677			
1	0.080907	0.024127	0.031373	0.417644	0.008426	0.076158	0.039331	0.091841	0.032351	0.018296			
2	0.117430	0.050449	0.038756	0.463613	0.029157	0.051227	0.039696	0.025276	0.032999	0.035957			
3	0.035093	0.024719	0.020686	0.439886	0.083334	0.081437	0.043778	0.110227	0.035212	0.036693			
4	0.065883	0.032426	0.019787	0.537206	0.048708	0.008388	0.050367	0.018093	0.041772	0.021540			

```
mask = sel.get_support()

selected_columns = X_df.columns[mask]

selected_columns

Int64Index([1, 711, 948, 1009, 4037, 4489, 4490, 4496, 4511, 4512], dtype='int64')

pd.DataFrame(sel.scores_)[mask].sort_values(0, ascending=False).apply(lambda x: round(x, 5))
```

	0
711	76.95660
1	52.73733
948	47.16216
4511	24.96895
4496	20.67175
4512	20.59872
4489	19.61724
4490	18.99177
1009	17.07909
4037	16.85579

```
pd.DataFrame(sel.scores_)[mask].sort_values(0, ascending=False)
```

	0
711	76.956601
1	52.737332
948	47.162159
4511	24.968948
4496	20.671750
4512	20.598724
4489	19.617237

▼ For Test3-CH3, choose the interesting points

tsnePCA\_df

	tsnePCA-2d-one	tsnePCA-2d-two
0	33.374310	29.723967
1	-34.470402	31.581984
2	-5.353730	36.885387
3	-14.315169	39.523678
4	62.257927	6.366529
...	...	...
6319	64.216072	-18.067184
6320	31.644821	-42.589016
6321	21.258493	-42.222939
6322	-48.072491	-49.547203
6323	-36.058273	-23.518888

6324 rows × 2 columns

tsnePCA\_df["RMS"] = Y

tsnePCA\_df["Filename"] = data["Filename"]

tokenized

0	Good
1	Good
2	Satisfactory
3	Good
4	Satisfactory
...	
6319	Good
6320	Good
6321	Good
6322	Satisfactory
6323	Satisfactory

Name: RMS, Length: 6324, dtype: object

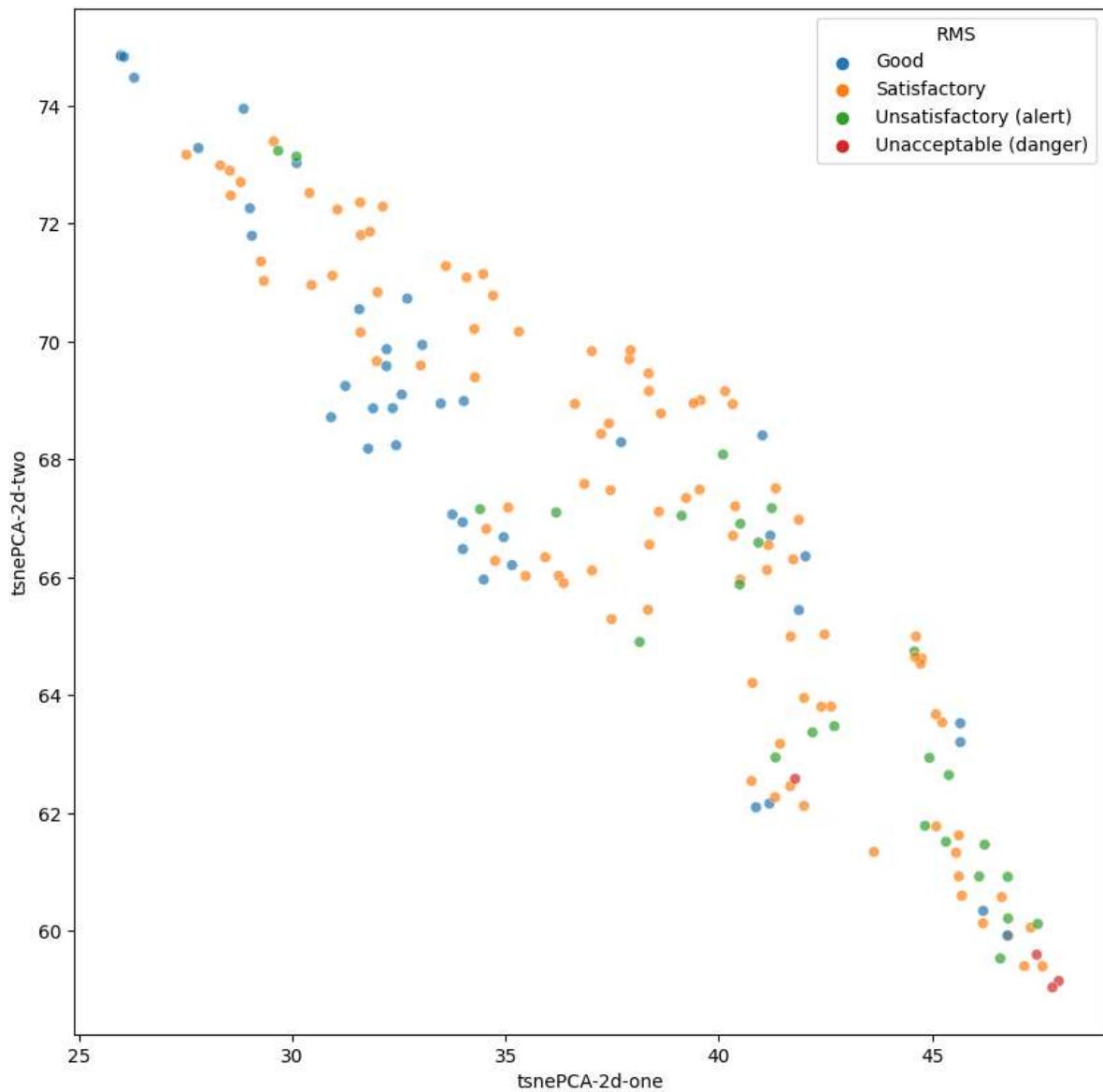
```
filtered = tsnePCA_df[
    (tsnePCA_df["tsnePCA-2d-one"] > 20)
    &
    (tsnePCA_df["tsnePCA-2d-two"] > 50)
]
```

```
plt.figure(figsize=(10,10))
sns.scatterplot(
    x="tsnePCA-2d-one", y="tsnePCA-2d-two",
    hue=tokenized,
    palette=sns.color_palette(),
```

```

data=filtered,
legend="full",
alpha=0.7
)
<ipython-input-110-67e0fb93ab64>:2: UserWarning: The palette list has more values (10) than needed (4), which may not be ir
sns.scatterplot(
<Axes: xlabel='tsnePCA-2d-one', ylabel='tsnePCA-2d-two'>

```



filtered

	tsnePCA-2d-one	tsnePCA-2d-two	RMS	Filename
813	31.244444	69.245735	0.970453	2004.04.17.00.02.55
825	34.486343	65.962936	1.188294	2004.04.16.23.52.55
826	30.908306	68.715034	1.697124	2004.04.16.23.42.55
840	32.211372	69.867264	1.534403	2004.04.17.01.12.55
841	27.794624	73.280060	1.199807	2004.04.17.02.42.55
...	...	...	...	...
995	47.969376	59.155186	98.199024	2004.04.18.02.22.55
996	36.847652	67.585159	2.334934	2004.04.18.00.12.55
997	46.599865	59.537941	5.665934	2004.04.18.00.22.55
998	47.822418	59.045563	89.106880	2004.04.18.02.32.55
999	44.834286	61.788311	8.125777	2004.04.18.01.32.55

162 rows × 4 columns

```
filtered.to_csv(f"/content/drive/MyDrive/nimbus-{test_to_check}-fixed.csv")
```

