```
Help on module S3DataUtils:

NAME
    S3DataUtils - Utils Functions involving usage of DataFrame

FUNCTIONS
    create_FunctionFrame(fs: int, Ns: int, Ss: int) -> pandas.core.frame.DataFrame
        Takes Sampling Frequency and returns a DataFrame
        with function vectors of frequencies

    predict_fs(fs: int, Ns: int, Ss: int, reg: sklearn.linear_model.base.LinearRegression) -> numpy.ndarray
        Returns predicted signal of given frequency
        Ss is sample rate
        Fs is natural frequency
        Ns is number of samples

    train_S3(FuncFrame: pandas.core.frame.DataFrame, sig: numpy.ndarray) -> sklearn.linear_model.base.LinearRegression
        Function That trains FuncFrame on input signal

        Returns:
            LinearRegression

FILE
    /mnt/4427FDEE206BF5AE/Documents/codes/S3_Smart_Sampling_Synthesiser/S3DataUtils.py


Help on module S3GuiMain:

NAME
    S3GuiMain - # -*- coding: utf-8 -*-

CLASSES
    builtins.object
        Ui_MainWindow

    class Ui_MainWindow(builtins.object)
     |  Methods defined here:
     |
     |  retranslateUi(self, MainWindow)
     |
     |  setupUi(self, MainWindow)
     |
     |  ----------------------------------------------------------------------
     |  Data descriptors defined here:
     |
     |  __dict__
     |      dictionary for instance variables (if defined)
     |
     |  __weakref__
     |      list of weak references to the object (if defined)

FILE
    /mnt/4427FDEE206BF5AE/Documents/codes/S3_Smart_Sampling_Synthesiser/S3GuiMain.py


Help on module S3Synth:

NAME
    S3Synth - Synthesiser Class of S3

CLASSES
    builtins.object
        S3Synth

    class S3Synth(builtins.object)
     |  S3Synth(wavecoef_: numpy.ndarray, transpo=1, mul=1)
     |
     |  Main Synth Class that manages backend of Synthesiser
     |
     |  Methods defined here:
```

```
     |
     |  __init__(self, wavecoef_: numpy.ndarray, transpo=1, mul=1)
     |      Initialize self.  See help(type(self)) for accurate signature.
     |
     |  out(self)
     |      Sends the synth's signal to the audio output and return the object itself.
     |
     |  sig(self)
     |      Returns the synth's signal for future processing.
     |
     |  ----------------------------------------------------------------------
     |  Data descriptors defined here:
     |
     |  __dict__
     |      dictionary for instance variables (if defined)
     |
     |  __weakref__
     |      list of weak references to the object (if defined)

FUNCTIONS
    random(...) method of random.Random instance
        random() -> x in the interval [0, 1).

FILE
    /mnt/4427FDEE206BF5AE/Documents/codes/S3_Smart_Sampling_Synthesiser/S3Synth.py


Help on module S3SignalUtils:

NAME
    S3SignalUtils - Utils function related to signals for S3

FUNCTIONS
    cos(fs: float, Ns: int, Ss: int) -> numpy.ndarray
        Returns a Cosine wave of Sample rate Ss with Ns number of samples and Sample Frequency Fs

    filt_bp(sig: numpy.ndarray, Ss: int, Cfs0: int, Cfs1: None, order=5) -> numpy.ndarray
        return a filtered signal; band pass filter

    filt_hp(sig: numpy.ndarray, Ss: int, Cfs: int, Cfs1: None, order=5) -> numpy.ndarray
        return a filtered signal; high pass filter

    filt_lp(sig: numpy.ndarray, Ss: int, Cfs: int, Cfs1: None, order=5) -> numpy.ndarray
        return a filtered signal; low pass filter

    sawtooth(fs: float, Ns: int, Ss: int) -> numpy.ndarray
        Returns a Sawtooth wave of Sample rate Ss with Ns number of samples and Sample Frequency Fs

    sigin(wavname: str) -> Tuple[int, numpy.ndarray]
        Functions that reads wave file and return sample rate and signal as np.array

    sin(fs: float, Ns: int, Ss: int) -> numpy.ndarray
        Returns a Sine wave of Sample rate Ss with Ns number of samples and Sample Frequency Fs

    triangle(fs: float, Ns: int, Ss: int) -> numpy.ndarray
        Returns a Triangle wave of Sample rate Ss with Ns number of samples and Sample Frequency Fs

DATA
    Tuple = typing.Tuple

FILE
    /mnt/4427FDEE206BF5AE/Documents/codes/S3_Smart_Sampling_Synthesiser/S3SignalUtils.py


Help on module S3Utils:

NAME
    S3Utils - Utils Functions for S3 Synthesiser App

FUNCTIONS
```

```
    create_env(sig: numpy.ndarray, Fs: float, Ss: int, Ns: int) -> numpy.ndarray
        return envelope of signal

    create_partial_envelope(sig: numpy.ndarray, Fs: float, Ss: int) -> numpy.ndarray
        Creates a partial envelope using min and max of in one cycle.

    find_Ns(Freq: float, Ss: int) -> int
        Finds the Ns for Training Phase

    find_maxsig(sig: numpy.ndarray, Ns: int) -> numpy.ndarray
        returns part of signal where its in constant sustain

    freq_calc(sig: numpy.ndarray, Ss: int) -> float
        Calculates the average frequency of the input signal (of a recorded note)

    freq_from_HPS(sig, fs)
        Estimate frequency using harmonic product spectrum (HPS)

    freq_from_autocorr(sig, fs)
        Estimate frequency using autocorrelation

    freq_from_crossings(sig, fs)
        Estimate frequency by counting zero crossings

    freq_from_fft(sig, fs)
        Estimate frequency from peak of FFT

    get_note(freq: float) -> Tuple[float, str]
        Returns the Note (and its Natural Frequency)
        corresponding to input frequency

    make_natural_env(env: numpy.ndarray, Ns: int) -> numpy.ndarray
        Returns an envelope in natural time for the
        signal by upsampling and uniforming partial envelope

    make_octaves() -> numpy.ndarray
        Creates Octaves with their corresponding frequency

    time(...)
        time() -> floating point number

        Return the current time in seconds since the Epoch.
        Fractions of a second may be present if the system clock provides them.

DATA
    Tuple = typing.Tuple
    log =

FILE
    /mnt/4427FDEE206BF5AE/Documents/codes/S3_Smart_Sampling_Synthesiser/S3Utils.py


Help on module S3SynthMain:

NAME
    S3SynthMain

CLASSES
    builtins.object
        S3App

    class S3App(builtins.object)
     |  Class to manage  interface of S3 Synthesiser
     |
     |  Methods defined here:
     |
     |  __init__(self)
     |      Initialize self.  See help(type(self)) for accurate signature.
     |
     |  load_file(self, file_path: str)
     |      Loads a Sample into the synthesiser
     |  
     |  load_trainedsynth(self)
     |      Loads all properties of S3 trains S3 and initialises S3Synth
     |  
     |  ----------------------------------------------------------------------
     |  Data descriptors defined here:
     |  
     |  __dict__
     |      dictionary for instance variables (if defined)
     |  
     |  __weakref__
     |      list of weak references to the object (if defined)

FUNCTIONS
    main()
        Driver code

FILE
    /mnt/4427FDEE206BF5AE/Documents/codes/S3_Smart_Sampling_Synthesiser/S3SynthMain.py
```