

Implementation the Entropy Calculation of any File in Python

Functionality

It calculates chunks' entropy of a required file.

Requirements

```
argon2-cffi==20.1.0
async-generator==1.10
attrs==20.3.0
backcall==0.2.0
bleach==3.2.3
cffi==1.14.4
colorama==0.4.4
cyclер==0.10.0
decorator==4.4.2
defusedxml==0.6.0
entrypoints==0.3
ipykernel==5.4.3
ipython==7.19.0
ipython-genutils==0.2.0
ipywidgets==7.6.3
jedi==0.18.0
Jinja2==2.11.2
jsonschema==3.2.0
jupyter==1.0.0
jupyter-client==6.1.11
jupyter-console==6.2.0
jupyter-core==4.7.0
jupyterlab-pygments==0.1.2
jupyterlab-widgets==1.0.0
kiwisolver==1.3.1
MarkupSafe==1.1.1
matplotlib==3.3.4
mistune==0.8.4
mpmath==1.1.0
nbclient==0.5.1
nbconvert==6.0.7
nbformat==5.1.2
nest-asyncio==1.5.1
nose==1.3.7
notebook==6.2.0
numpy==1.19.5
packaging==20.9
pandas==1.2.1
pandocfilters==1.4.3
parso==0.8.1
pickleshare==0.7.5
Pillow==8.1.0
prometheus-client==0.9.0
prompt-toolkit==3.0.14
pycparser==2.20
Pygments==2.7.4
pyparsing==2.4.7
pyrsistent==0.17.3
python-dateutil==2.8.1
pytz==2020.5
pywin32==300
pywinpty==0.5.7
```

```
pyzmq==22.0.2
qtconsole==5.0.2
QtPy==1.9.0
scipy==1.6.0
Send2Trash==1.5.0
six==1.15.0
sympy==1.7.1
terminado==0.9.2
testpath==0.4.4
tornado==6.1
traitlets==5.0.5
wcwidth==0.2.5
webencodings==0.5.1
widgetsnbextension==3.5.1
```

Imported Modules

- **import collections**

It is used to store collections of data, for instance, list, dict, set, tuple.

- **import io**

It is used for dealing with various types of I/O. Three main types of I/O: text I/O, binary I/O, and raw I/O exist.

- **import sys**

It is used to provide information about the methods of the Python interpreter.

- **import matplotlib.pyplot as plt**

It is used to create static visualizations and interactive one. matplotlib.pyplot creates plots area in figures, labels, etc like the ones in MATLAB.

- **from scipy.stats import entropy**

It is used to calculate the entropy of distribution for given probability values. 'scipy.stats.entropy(pk, qk=None, base=None, axis=0)' is used to the following equation: - $S = -\sum(pk * \log(pk), axis=axis)$.

```
Parameters
pk: sequence
Defines the (discrete) distribution. pk[i] is the (possibly unnormalized) probability of event i.

qk: sequence, optional
Sequence against which the relative entropy is computed. Should be in the same format as pk.

base: float, optional
The logarithmic base to use, defaults to e (natural logarithm).

axis: int, optional
The axis along which the entropy is calculated. Default is 0.

Returns S : float
The calculated entropy.
```

Functions

ID	Function Name	Parameters	Type	Data Structue	Returned Values	Type	Data Type/Structue
1	calc_shannon_entropy	data	Hex values	List	entropy_value	Float	Number
2	diplay_entropy	thefile, chunk_size	String, Integer	String, Number	Plot, chunk_inranges, chunk_ent_values	Plot Figure, String, Float	Figure, List, List

1. **calc_shannon_entropy(data: List of Hex values):**

This function calculates the entropy for each chunk and return its float entropy value.

2. **diplay_entropy(thefile: String name of the required file, chunk_size: Integer value of each chunk in Bytes, the default value is 700):**

This function reads the required file (it reads all kind of file types) as binary , then read each value index and value ('0xXX') in hex. It creates list of hex values with desired length as it is entered in chunk size, then calculates the entropy to each chunk and stores the results in new list. After that, it creates plot with entropy range for 'y' axis and index range for 'x' axis. Each point in the plot indicates to ('Start and End point of Chunk in Hex', Entropy Value of this chunk).

3. **main:**

It uses like that '-f filename -c chunksize' | String, Integer | String, Number | 1 | Plot Figure | Figure This function reads the arguments (Inputs) from the command line and calls the diplay_entropy function with those arguments. It shows help usage instruction also.

User Guide:

We run this code as the following in command line:

```
pip install -r requirements.txt
python -f filename -c chunksize_in_Bytes:default:700
```

Example

```
python -f filename.png -c 5000
```

- For help:

```
python -f help
python -f --help
python -f -help
python -f -h
python -f --h
```

- You can also run the code without specifying chunk size, the default chunk size = 700 Bytes.

Example

```
python -f filename.png
```

- You can setup the chunk size:

Example

```
python -f filename.png -c 5000
```