Domain Extractor

1. Problem Statement

Write a program in Python to count unique domain addresses from an input file and output the count for each unique domain.

1. Input

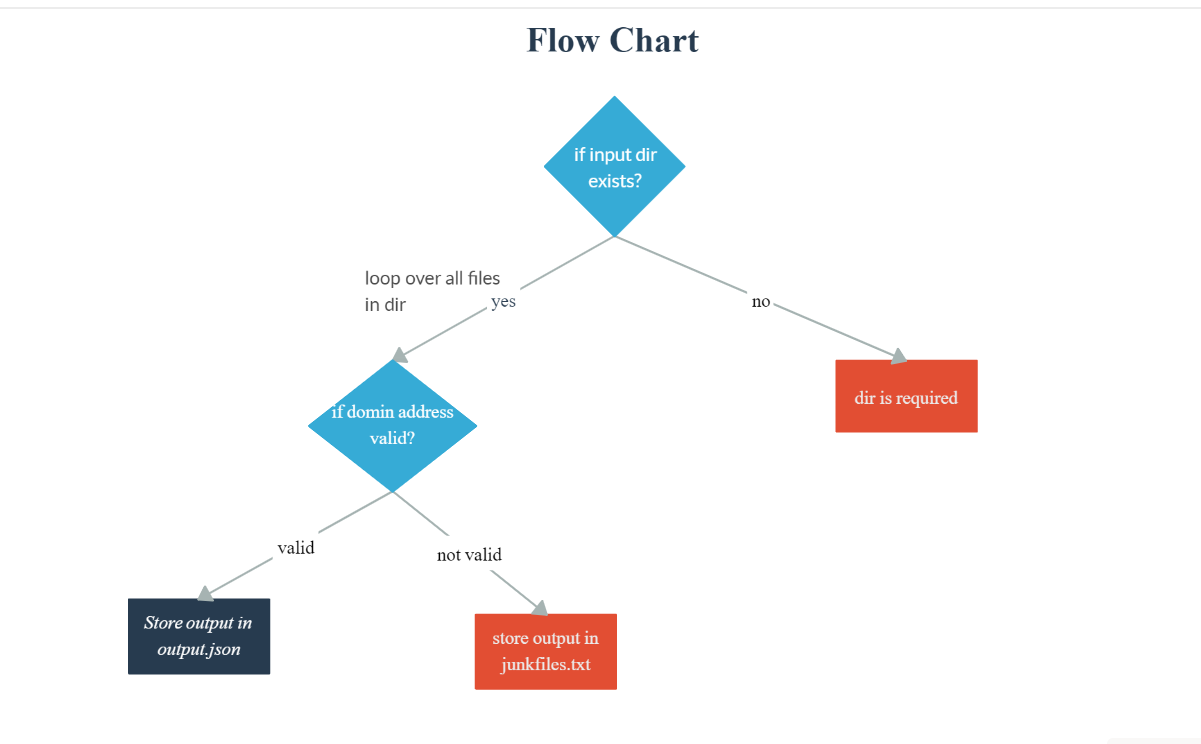
Input files are the text files that contains the email addresses

1. Expected output

Counting unique domain addresses from input file and display count of unique domain names.

1. Assumptions
2. Assuming that unique addresses are required from one file.
3. There should be no special characters except hyphen in domain name
4. Approach

Below is the high level approach :-



The program is tested for python 2.7.13 and python 3.7 version

Below steps are followed for the program

1. **Input** : The program is capable handling as many files present in the folder addressFiles
2. **Output**: The output shows the count of unique domain addresses along with the count of each unique domain address.

At the same time, a folder ‘outputFiles’ is created to put the outputs in 2 files ‘junkmail.tx’ which can be used for further investigation and ‘output.json’

The format of ‘output.json’ looks like below

*{*

*"easylist.txt": {*

*"afterlivesbaseline": 2,*

*"airbrushedliterally": 2,*

*"americanizingguerrero": 1,*

*"amortizedladdered": 3,*

*"angoradiego": 2,*

*.*

*.*

*.*

1. **Program Execution**: The program structure is like below

rackspace/

├── addressFiles

│   ├── easylist.txt

│   ├── empty.txt

│   ├── hardestlist.txt

│   ├── hardlist.txt

│   └── mediumlist.txt

├── extractor.py

├── \_\_init\_\_.py

├── MANIFEST.in

├── outputFiles

│   ├── junkmail.txt

│   └── output.json

├── README.txt

├── requirements.txt

└── setup.py

In the main folder, executor.py is the main file to run the program which will in turn deliver the results.

1. Deploying the application

The application is tested on python2.7 and 3.7 on windows and linux chromium distribution.

To deploy the application there is MANIFEST.in and setup.py file in place to deploy the application in various platforms. The file requirements.txt contains all the dependencies required for the program to run.

*python setup.py sdist*

The above command is capable of deploying the application in versions described in setup.py and MANIFEST.in

1. Test Cases

Included in TestCases.xlsx file

1. Further Considerations

1. **Coding and Encoding**: The input files ae in different encodings. Correctly detecting the encoding for as many files at times can be challenging. Below code can help find the encodings of the files.

*import chardet*

*for i in onlyfiles:*

*rawdata = open(i, 'rb').read()*

*result11 = chardet.detect(rawdata)*

*print(result11['encoding'])*

output:

*Windows-1254*

*utf-8*

*ISO-8859-1*

*utf-8*

2. **Logging:** We can create a logger to create the logging of the application