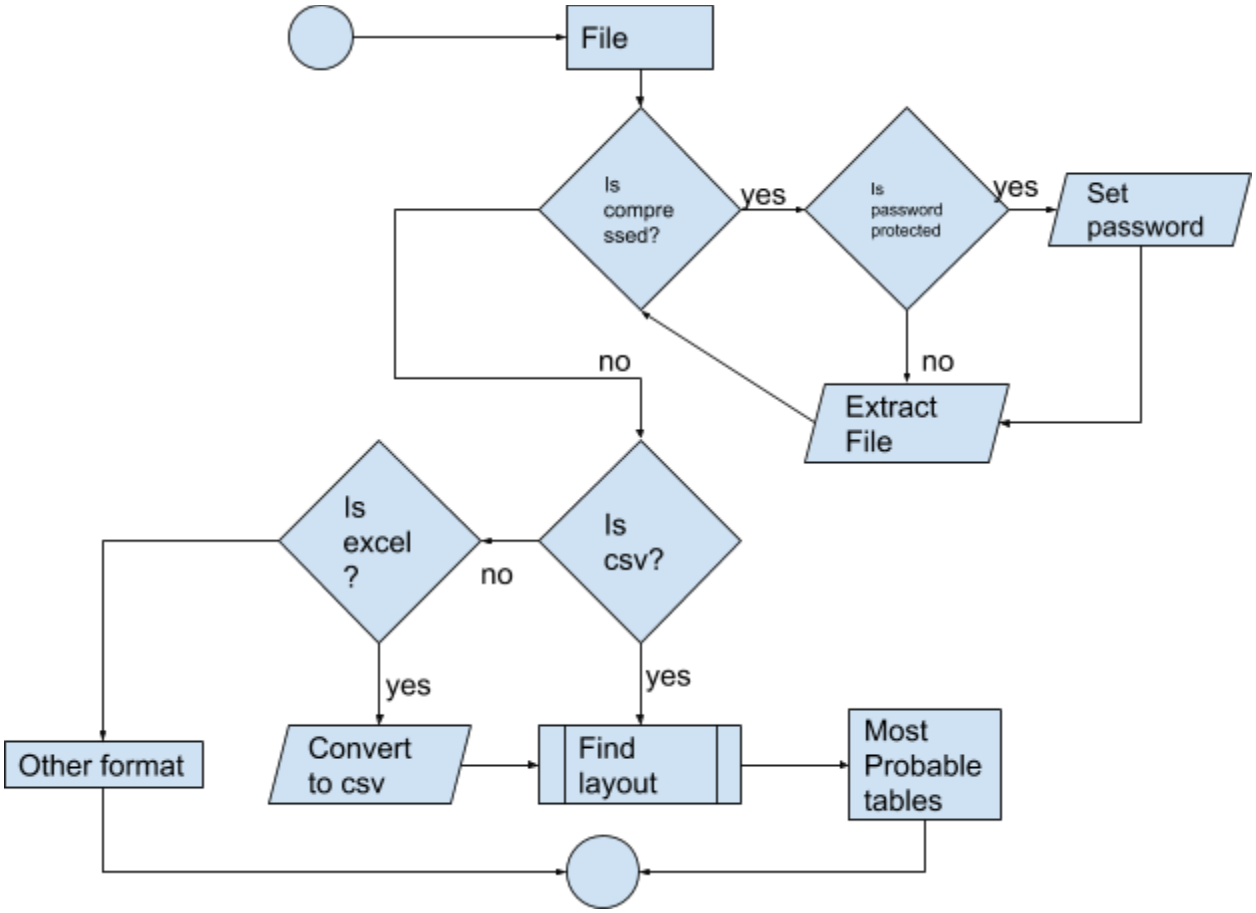
Layout Detector



Features

1. Determine the file type & handle accordingly
2. Currently it handles compressed files (zip, rar files-can be added as necessary), text files(.txt, .csv, etc) & excel files
3. If the file is in excel, convert it to a delimited csv, (default delimiter is |, can be changed per requirements)
4. Determine the layout of the file, i.e. in which table the given files belong to!

Extracting files

Different tools were tried to extract the files. Some tools & their limitations.

Tools	Pros	Cons
zipfile	Default python library	Only support zipfile
Rar file	Default python library	Only support rar files.
patoolib	Supports many formats	No support for password
WinRAR.exe	Supports many format, has password support	Not a python module. Can be called using subprocesses.

Major Problem with python tools

The tools in python that are used to deal with compressed files do not support all type of encryption (This same problem was faced while I was dealing with the compressed files in python. During my experience AES 256 encryption could not be handled). The issue was addressed as a bug in python 3.2. The link is available here. However, this problem is dealt by using WinRAR. WinRAR is available for both windows & ubuntu! We can call sub-process for WinRAR adn deal with

Converting excel to csv

There are different ways to convert excel to csv. Some ways to do this are :

1. Using pandas
2. Using csv & openpyxl

We will discuss both methods, the first method is relatively easy but requires external library.

Using pandas

Using pandas we can deal with multi sheet excel files:

```
import pandas as pd

def to_csv(pathname):
    '''
    takes in a xlsx file with one or more sheets and produces corresponding
csv and returns their paths.
    '''
    excel = pd.ExcelFile(pathname)
    print(excel.sheet_names)
    table_path = []
    for i in excel.sheet_names:
        print(pathname+"_" +
              i+".csv")
        table_path.append(pathname+"_" +
                          i+".csv")
        df = pd.read_excel(excel, i)
        df.to_csv(pathname+"_" +
                  i+".csv", sep="|", index=False)
    return table_path


'''
excel_talbes = to_csv(
                  "G:\Siddhi\Office      Personal\Content      Based\Content
Based.rarextracted\movies.xlsx")
for i in excel_tables:
```

```
    print(i)
'''
```

Using csv & openpyxl

```
import xlrd
import csv
def to_csv(file_path):
    with xlrd.open_workbook(file_path) as wb:
        # or wb.sheet_by_name('name_of_the_sheet_here')
        sh = wb.sheet_by_index(0)
        # open('a_file.csv', 'w', newline="") for
        with open(file_path+'.csv', 'w', newline="") as f:
            c = csv.writer(f)
            for r in range(sh.nrows):
                c.writerow(sh.row_values(r))
```

Identifying file format

We cannot always depend on the the extension of the file to know the file type. We must look deeper. For our purpose we can look at the MIME type of the file to identify its type. We can use a library called filemagic. This library does not give the information about the extension of the file.

```
>>> with magic.Magic() as m:
...     m.id_filename('setup.py')
...
'Python script, ASCII text executable'
```

Limitations

1. How to pass the more than one passwords if there are files that are encrypted with more than password.
2. How to pass password if there are compressed files inside a compressed file.
3. Cannot determine the actual length of the fields

Finding Layout