NLP Programming Assignment #1 SpamLord.
Report:

Submitted by – Vaibhav Rawat
UIN – 626008171

Steps to compile and run the code with Python 2.7:

1. Open the terminal
2. Go to the folder NLPAssign1.
3. Enter the following command:
   python SpamLord.py data_dev/dev/ data_dev/devGOLD

Results and Analysis:

I have used three patterns for emails extraction and one pattern for phone number extraction –

1. Pattern for phone -

```
phn_pattern = '(?:\D|^)(?:\(|\{)?(\d{3})(?:\)|\})?(?:\s|-| )*(\d{3})(?:\s|-
| )+(\d{4})(?:\b|[^0-9])'
```

covers all the following scenarios for phone numbers :

Phone: (979) 862-2908
Tel (+1): 979-862-2908
TEL +1 979 862 2908

2. Pattern for emails –

2.1 Short emails (with single dot) :

```
email_pattern = '([\w-]+|[\w-]+\.[\w-
]+)(?:(?:[\(\{\s]+(?:at|where)[\)\}\s]+)|(?:[\(\{\s]*(?:@|&#x40;)[\)\}\s]*))([\
w-]+)(?:(?:[\(\{\s]+(?:do?t|DOM)[\)\}\s]+)|(?:[\(\{\s]*(?:\.|;)[\)\}\s]*))(?:-
?e-?d-?u|com)'

covers scenarios of the following types:

ashishg@stanford.edu
ashishg at stanford dot edu
```

2.2 Long emails (with double dot):

```
email_pattern_long = "([\w-]+|[\w-]+\.[\w-
]+)(?:(?:[\(\{\s]+(?:at|where)[\)\}\s]+)|(?:[\(\{\s]*(?:@|&#x40;)[\)\}\s]*))([\
```

```
w-]+)(?:(?:[\(\{\s]+(?:do?t|DOM)[\)\}\s]+)|(?:[\(\{\s]*(?:\.|;)[\)\}\s]*))([\w-
]+)(?:(?:[\(\{\s]+(?:do?t|DOM])[\)\}\s]+)|(?:[\(\{\s]*(?:\.|;)[\)\}\s]*))(?:-
?e-?d-?u|com)"
```

covers scenarios of the following types:

huangrh@cse.tamu.edu
cheriton@cs.stanford.edu
huangrh(at)cse.tamu.edu
huangrh at cse dot tamu dot edu

2.3 Obfuscated emails:

```
obfs_email_pattern =
"obfuscate\s*(?:\()\s*(?:')(.+)(?:')\s*(?:\,)\s*(?:')(.+)(?:')\s*[\)]"
```

covers obfuscated emails of the following format:

<script type="text/javascript">obfuscate('cse.tamu.edu','huangrh')</script>

Following results are obtained with my code

True Positives (59):
set([('ashishg', 'e', 'ashishg@stanford.edu'),
('ashishg', 'e', 'rozm@stanford.edu'),
('ashishg', 'p', '650-723-1614'),
('ashishg', 'p', '650-723-4173'),
('ashishg', 'p', '650-814-1478'),
('balaji', 'e', 'balaji@stanford.edu'),
('bgirod', 'p', '650-723-4539'),
('bgirod', 'p', '650-724-3648'),
('bgirod', 'p', '650-724-6354'),
('cheriton', 'e', 'cheriton@cs.stanford.edu'),
('cheriton', 'e', 'uma@cs.stanford.edu'),
('cheriton', 'p', '650-723-1131'),
('cheriton', 'p', '650-725-3726'),
('dabo', 'e', 'dabo@cs.stanford.edu'),
('dabo', 'p', '650-725-3897'),
('dabo', 'p', '650-725-4671'),
('dlwh', 'e', 'dlwh@stanford.edu'),
('engler', 'e', 'engler@lcs.mit.edu'),
('engler', 'e', 'engler@stanford.edu'),
('eroberts', 'e', 'eroberts@cs.stanford.edu'),

```
    ('eroberts', 'p', '650-723-3642'),
    ('eroberts', 'p', '650-723-6092'),
    ('fedkiw', 'e', 'fedkiw@cs.stanford.edu'),
    ('hager', 'e', 'hager@cs.jhu.edu'),
    ('hager', 'p', '410-516-5521'),
    ('hager', 'p', '410-516-5553'),
    ('hager', 'p', '410-516-8000'),
    ('hanrahan', 'e', 'hanrahan@cs.stanford.edu'),
    ('hanrahan', 'p', '650-723-0033'),
    ('hanrahan', 'p', '650-723-8530'),
    ('horowitz', 'p', '650-725-3707'),
    ('horowitz', 'p', '650-725-6949'),
    ('jks', 'e', 'jks@robotics.stanford.edu'),
    ('jurafsky', 'e', 'jurafsky@stanford.edu'),
    ('jurafsky', 'p', '650-723-5666'),
    ('kosecka', 'e', 'kosecka@cs.gmu.edu'),
    ('kosecka', 'p', '703-993-1710'),
    ('kosecka', 'p', '703-993-1876'),
    ('kunle', 'e', 'darlene@csl.stanford.edu'),
    ('kunle', 'e', 'kunle@ogun.stanford.edu'),
    ('kunle', 'p', '650-723-1430'),
    ('kunle', 'p', '650-725-3713'),
    ('kunle', 'p', '650-725-6949'),
    ('lam', 'e', 'lam@cs.stanford.edu'),
    ('lam', 'p', '650-725-3714'),
    ('lam', 'p', '650-725-6949'),
    ('latombe', 'e', 'asandra@cs.stanford.edu'),
    ('latombe', 'e', 'latombe@cs.stanford.edu'),
    ('latombe', 'e', 'liliana@cs.stanford.edu'),
    ('latombe', 'p', '650-721-6625'),
    ('latombe', 'p', '650-723-0350'),
    ('latombe', 'p', '650-723-4137'),
    ('latombe', 'p', '650-725-1449'),
    ('levoy', 'e', 'ada@graphics.stanford.edu'),
    ('levoy', 'e', 'melissa@graphics.stanford.edu'),
    ('levoy', 'p', '650-723-0033'),
    ('levoy', 'p', '650-724-6865'),
    ('levoy', 'p', '650-725-3724'),
    ('levoy', 'p', '650-725-4089')])
False Positives (0):
set([])
False Negatives (0):
set([])
```

Summary for the results: tp=59, fp=0, fn=0

Results are pretty good with true positives, tp = 59. None of the emails and phone numbers are incorrectly classified (fp = 0, fn = 0).

Any known bugs, problems, or limitations of your program

Program is able to correctly handle all the possible scenarios for emails and phone numbers, with training data set. Additionally, I have covered up more scenarios from the web which seemed plausible.