Compiling and running the code:

Unzip the folder. cd into the directory PA1 and run:

```
python SpamLord.py data_dev/dev/ data_dev/devGOLD
```

Results and Analysis:

SpamLord.py contains two regular expressions patterns. One is for email and the other for phone numbers.

Email Id Regular Expression:

```
my_first_pat = "([\w.-]+) *(@|WHERE| at |&#x40;) *([\w;.-]+) *(.|DOM|dot|) *(?i)(edu|-e-d-u)|(obfuscate\('(\w+\.edu)','(\w+)'\))"
```

The first group ([\w.-]+) matches the words of the email before @ or at or WHERE or &#x40; *([\w;.-]+) matches the domain name of the email id like stanford. or cs.stanford. (.|DOM|dot|) matches . or dot or DOM. (?i)(edu|-e-d-u) matches edu or -e-d-u and is case insensitive. (obfuscate\('(\w+\.edu)','(\w+)'\)) matches the email which has the format obfuscate('cse.tamu.edu','huangrh').

Phone Number Regular Expression:

```
my_second_pat = '\(?(\d{3})\)?[\s-]?(\d{3})[\s-](\d{4})\D+'
```

The first part matches if it starts with a parenthesis. The d{3} checks if it has 3 numbers. \) checks if it has a closing parenthesis. \s- checks if there is a space or -. d{3} again matches with 3 numbers and d{4} matches with 4 numbers in the end. \D checks if the expression doesn't have any more digits.

With the above regular expressions, I am getting a tp of 58, fp of 0 and fn of 1.

Limitations:

A False Negative occurs for the case hager at cs dot jhu dot edu. This case is not handled in the email regular expression.