**Project 1: Machine learning-based SMS Spam Filtering**

**Project Statements**

**- Objective**

For this project, you are asked to implement a detection program supporting Short Message Service (SMS) spam filtering. The main concern is to design/generate features to differentiate SMS spam messages from legitimate ones, and run machine learning techniques (i.e., supervised learning) to classify SMS spam messages. Unlike email spam filtering, SMS spam filtering poses its own intrinsic problem because the length of text messages is relatively small (up to 160 characters or less). To come up with this project successfully, you must devise robust and efficient detection features to solve this problem.

.

**- Dataset**

You will explore real SMS Spam Collection Data Set, corpus of mobile SMS labeled Spam/Legitimate messages from the UCI Machine Learning Repository website (<https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection>). The SMS Spam collection contains a total of 5574 SMS messages, which is composed of 747 spam and 4827 legitimate messages. Each line has one SMS message including two instances: the first instance indicates label if the SMS message is spam or legitimate (ham), and the second one is the content of the message (i.e., raw text). Table 1 shows some samples of SMS Spam/Legitimate messages in the given dataset.

Table 1: SMS Spam/legitimate message samples

| **Label** | **Text** |
| --- | --- |
| spam | "For the most sparkling shopping breaks from 45 per person; call 0121 2025050 or visit www.shortbreaks.org.uk" |
| spam | December only! Had your mobile 11mths+? You are entitled to update to the latest colour camera mobile for Free! Call The Mobile Update Co FREE on 08002986906 |
| ham | Yup next stop. |
| ham | No. I dont want to hear anything |

Note that there is an input file you can use: SMS\_Spam\_data (466KB) and you must add the citation (SMS Spam Collection v.1 “http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/”) in your report.

**Project Requirements**

**- Feature Extraction**

Write a java (or python) program for the detection features from raw text and generate a feature set file. You will be programming **FOUR** detection features and are responsible for justifying the implementation strategy for each of the features in the report. If you are using any references in terms of the implementation, you should cite and explain it concisely. For project 1, the detection features are:

| **Detection feature** | **Description** |
| --- | --- |
| Number of Characters typed  in Message | In Table 1 above, it appears that SMS spam messages include relatively large number of characters than legitimate messages. From the SMS spammers' (e.g., marketers) perspective, they are likely to use more characters available as long as it doesn't exceed the limit of SMS to send the sufficient information to customers for illicit profits. |
| Number of Currency Symbols | To take the bait by mobile users, SMS spammers might emphasize the prize (or cash) using the currency symbol (e.g., £) in the SMS message. This is typically different from legitimate messages. Here is an example, “*Please call our customer service representative on 0800 169 6031 between 10am-9pm as you have WON a guaranteed £1000 cash or £5000 prize!*” |
| Number of Numeric String | One of the intrinsic factors from the SMS spams is a CONTACT number or PROMOTION code. Since the phone number is sensitive, it is not likely to be in the legitimate messages frequently. (Example: “*PRIVATE! Your 2003 Account Statement for shows 800 un-redeemed S. I. M. points. Call 08719899230 Identifier Code: 41685 Expires 07/11/04*”) |
| The frequency of most popular term/word |  |

**- Binary Classification**

From WEKA, run different binary classifiers to identify SMS Spam messages using the feature set you devised. You should run the following **FIVE** classification algorithms with Cross Validation (by default, K =10) and report the experimental results you analyzed. Specifically, you must report the best classifier with i) Accuracy, ii) True Positive rate (TPR), and iii) False Positive rate (FPR).

* Decision Tree (J48)
* Multinomial Naive Bayes
* K-Nearest Neighbors
* SVM (LibSVM)
* RandomForest

**Programming Requirements**

We highly recommend you to use **Java (or Python)** and **WEKA**  for this assignment. You can also use other langagues but you need to clearly write the instrucitons to configure the compling and running environment to test your code. You might also be required to give a demo of your code. You need to write the programs to extract detection features and apply machine learning techniques using WEKA. Here is official Weka documentation (e.g., Weka Wiki, FAQ, Tutorials) link (<http://www.cs.waikato.ac.nz/ml/weka/documentation.html>). Additionally, your program should compile and run on the Virtual Cyber Security Lab (<https://engineering-computer-science.wright.edu/computer-science-and-engineering/advising-and-resources/virtual-cyber-security-lab-vcsl>). If the program only runs in your own machine, you should show the demo to TA in person.

Your program should generate the output in the format of Attribute-Relation File Format (ARFF), which can be directly analyzed by the Weka.

**Submission Requirements**

The assignment must be done **originally**.Please submit **TWO** files:

i) A single tar (or zip) file that include all files below:

* **Source codes**: your java (or python) codes and a shell script files (if applicable) that you use for testing. Your code also needs to be well-documented, with any major constructs (i.e., functions) clearly commented.
* **README:** overall high-level documentation of instructions to run your program.(1 page)
* **Report:** well-documentedsummary that includes the annotation/justification of detection features and the experimental results you analyzed. (at least 5 pages, the more formal the better.)
* **Dataset**: a feature set data in the format of ARFF to be tested

ii) a MS word (or Acrobat PDF, Plain Text) file, including ONLY source codes for originality checking.

**Late & Plagiarism Policy**

Please check the late/plagiarism policy on the course syllabus.