**CSE 5334 Data Mining  
Fall 2013 Project-1**

**Project Document**

**Requirements:**

You need to submit your project document and source code

* The project document describes how you tackle the problem, and how you evaluate your solution.
  + - No limitation on the number of pages, or the format of the document.
    - The document should clearly describe how you design, implement, and evaluate your classifier.
* Source code must be submitted in a .zip file
  + - You have to implement the whole project by yourself. No software package allowed, except standard libraries (such as C/C++ library), stemming and stop-words removal tools, if necessary
    - You can use any language, though I recommend C++, Java, Python
    - Your source code must pass compilation. Any non executable submission is not acceptable.
* Compilation and execution
  + - Compile and test your program on omega.uta.edu before you submit
    - Please strictly follow following compilation instruction, your submission may be rejected if compilation fails:
      * First, go to the folder holds your source code;
      * For C++, compile your program using: g++ \*.cpp
      * For Java, use find. -type f -name "\*.java" -print | xargs javac
    - Please strictly follow following execution instruction, your submission may be rejected if execution fails:
      * For C++, run your program using: ./main train\_file test\_file
      * For Java, use: java Main train\_file test\_file
      * For Python, use: python main.py train\_file test\_file
    - For other languages, you may have to demonstrate to the TA how to compile and run your program.
    - The output of the program should be a file named result.txt. The file should contain a header and have the following format:

Id, Tags

1,"c++ javaScript"

2,"php python mysql"

3,"django"

Vatsal Kesarwani

1000869718

**Solution:**

The given problem in this project for the tag prediction of the questions from Stack Exchange sites (eg: www.stackoverflow.com) when only questions text and their title is given is tackled by using the following:

* Term frequency
* Tf-idf weighting
* Vector space model
* Naive bayes classifier

Firstly the term frequency is calculated corresponding to each question title with each tag and each question body with each tag. Then tf-idf weights are calculated corresponding to each tag along with each title and question body using the following formula:

The tf-idf weights are then used to calculate the cosine similarity between the title vector and the body vector. The ratio of the importance of question body with question title is taken equal to the cosine similarity between them.

**For eg:** if between a title vector and body vector, angle is zero, cosine of angle will be 1. That is when angle is zero both title and body are completely similar and thus ratio of body to title equals 1.

This is used to merge the weights of title and body for a particular tag making them just 1 weight per question for each tag. Then the naive bayes classifier is applied on the weights between question and tag by calculating various required identifiers to get the probabilistic classification of each question’s belonging with respect to each tag which acts as separate classes for classification.

The tags with maximum probabilities for a question are predicted as potential tags for the given question. Also prior to tackling text mining the data reading and cleaning is performed. The reading of csv file is performed using Super CSV library. The library used for the purpose of cleaning is lucene 4.5. Using lucene the stop words are removed and the stemming is performed on raw data before breaking it in tokens to start text mining. The stop words removed are the standard ones defined in the lucene library.

**References:**

* http://lucene.apache.org/core/4\_5\_1/index.html
* http://supercsv.sourceforge.net/
* http://docs.oracle.com/javase/7/docs/api/
* www.stackoverflow.com for various topics including lucene, super csv.