# CS494 - IR - SAMPLE EXAM 2

**Name:**

**University NetID:**

This test consists of 5 questions. The number of points for each question is shown below.

- Read all questions carefully before starting to answer them.
- Write all your answers in the space provided in the exam paper.
- The order of the questions is arbitrary, so the difficulty may vary from question to question. Do not get stuck by insisting on doing them in order.
- Show your work. Correct answers without justification will not receive full credit. However, also be concise. Excessively verbose answers may be penalized.
- Clearly state any simplifying assumptions you may make when answering a question.
- **Be sure to write your name on the test paper.**

| Question | 1 | 2 | 3 | 4 | 5 | total |
|---|---|---|---|---|---|---|
| Points | 20 | 20 | 20 | 20 | 20 | 100 |
| Your Points | | | | | | |

**Exercise 1 - 20 points. (Naïve Bayes)**

You are given a collection of 1,000 documents, which are classified into one of the two classes: *politics* and *technology*. The vocabulary of words in the collection is as follows:

$$V = \{Android, camera, pictures, Obama, elections, Foxnews\}$$

Assume there are 400 documents from *politics* and 600 documents from *technology*. Assume further that the frequency counts of words in each class are as follows:

$$\textit{politics} : Android(10), camera(5), pictures(5), Obama(700), elections(423), Foxnews(365)$$

$$\textit{technology} : Android(700), camera(668), pictures(400), Obama(40), elections(30), Foxnews(71)$$

Train a Multinomial Naïve Bayes model and predict the class for the test document below:

*I took lots of pictures of Obama with my Android.*

Ignore any words that are not in the vocabulary. Do add-1 smoothing.

**Exercise 2 - 20 points. (Web Crawling)**

Consider the following web graph:

Page A points to pages B, D, and E.
Page B points to pages C and E.
Page C points to pages F and G.
Page D points to page G.
Page G points to page E.

Show the order in which the pages are indexed when starting at page A and using a breadth-first spider (with duplicate page detection). Assume links on a page are examined in the orders given above.

**Exercise 3 - 20 points. (HITS)**

Consider the web graph from Exercise 2, shown below for convenience.

Page A points to pages B, D, and E.
Page B points to pages C and E.
Page C points to pages F and G.
Page D points to page G.
Page G points to page E.

Run the HITS (Hubs and Authorities) algorithm on this graph of web pages. Simulate the algorithm for two iterations.

**Exercise 4 - 20 points. (Page Rank)**

Consider the following pages and the set of web pages that they link to:

```
Page A points to pages C, D.
Page B points to page C.
Page C points to pages D, B.
Page D points to page B.
```

Consider running the PageRank algorithm on this graph of pages. Assume $\epsilon = 0.15$. Simulate the algorithm for two iterations. Show the page rank scores for each page for each iteration. Order the elements in the vectors in the sequence: A, B, C, D.

Remember:

$$R(A) = \frac{\epsilon}{n} + (1 - \epsilon) \sum_{(B,A) \in G} \frac{R(B)}{out(B)}$$

**Exercise 5 - 20 points. (The kNN Algorithm)**

Given a database about whether a user should go skiing or not, we want to learn a classifier, which will be used to advise the user with respect to skiing activities. The decision of going skiing depends on the attributes *snow*, *weather*, *season*, and *current physical condition* of the user, as shown in the table below:

| | snow | weather | season | physical condition | go skiing |
|---|---|---|---|---|---|
| 1 | sticky | foggy | low | rested | no |
| 2 | fresh | sunny | low | injured | no |
| 3 | fresh | sunny | low | rested | yes |
| 4 | fresh | sunny | high | rested | yes |
| 5 | fresh | sunny | mid | rested | yes |
| 6 | frosted | windy | high | tired | no |
| 7 | sticky | sunny | low | rested | yes |
| 8 | frosted | foggy | mid | rested | no |
| 9 | fresh | windy | low | rested | yes |
| 10 | fresh | windy | low | rested | yes |
| 11 | fresh | foggy | low | rested | yes |
| 12 | fresh | foggy | low | rested | yes |
| 13 | sticky | sunny | mid | rested | yes |
| 14 | frosted | foggy | low | injured | no |

Use the 3-NN algorithm to make a recommendation to the user for a scenario where:

```
snow=sticky, weather=windy, season=high, physical condition=tired
```

Use Hamming distance (i.e., number of attributes where two instances differ) to calculate the distance between instances.