CLASSIFICATION [20%]

a) [7%] The decision function for a linear support vector machine is given as $f(\mathbf{x}) = sign(\mathbf{w}^T\mathbf{x} + b)$. Explain what a support vector is, and describe the relation between the weight vector $\mathbf{w}$ and the support vectors.
b) [7%] You have a binary classification problem that you have discovered is not linearly separable. Given that you are to use a support vector machine to distinguish between the two classes, outline two different approaches for dealing with the fact that the problem is not linearly separable.
c) [6%] Describe how *k*-fold cross-validation works and what its purpose is.

EVALUATION [20%]

a) [4%] Define, precisely, the two metrics precision and recall. Give examples of situations where you'd clearly want to prioritize one over the other.
b) [4%] Consider the two metrics precision and recall. For each of these metrics, provide at least one good example of a search-related feature (related to either query- or content-processing) that is designed to increase that metric. Explain why this would increase the metric, referring back to the metric's definition.
c) [4%] Give a precise definition of *F*-score.
d) [4%] Explain what mean average precision (MAP) is and how it is computed.
e) [4%] Explain what normalized discounted cumulative gain (NDCG) is and how it is computed. Describe a situation where the NDCG score is artificially high without giving much/any value back to the user.

STRINGS [20%]

a) [8%] Consider the string *hakkebakke*. Show how to construct the suffix array for this string, and explain how you can use the suffix array to efficiently locate all occurrences of the substring *ak*.
b) [12%] Given a set of n strings $\{S_1, S_2, \ldots, S_n\}$, you want to determine their longest common substring. For example, with $n = 3$ and {*abababca*, *aababc*, *aaababca*} the longest common substring would be *ababc*. Outline how you could use a suffix array to determine the longest common substring for such a set of n strings.

HEAPS' LAW [20%]

a) [5%] Heaps' law is given as $M = kT^b$. Explain what M, k, T and b are.
b) [15%] Looking at a collection of web pages, you find that there are 3×10^3 different terms among the first 10^4 tokens and 3×10^4 different terms among the first 10^6 tokens. Assume a search engine that indexes a total of 2×10^{10} pages from this collection, each page containing 2×10^2 tokens on average. What is the size of the vocabulary of the indexed collection as predicted by Heaps' law?

POTPOURRI [20%]

a) [10%] Explain the idea behind the Rocchio algorithm for relevance feedback.
b) [10%] Explain how a Bloom filter works. What are some of its most important properties?