Midterm for CSC411/2515, Machine Learning and Data Mining Fall 2018, Version A Friday, October 19, 6:10-7pm

Name: _____

Student number: _____

This is a closed-book test. It is marked out of 15 marks. Please answer ALL of the questions. Here is some advice:

- The questions are NOT arranged in order of difficulty, so you should attempt every question.
- Questions that ask you to "briefly explain" something only require short (1-3 sentence) explanations. Don't write a full page of text. We're just looking for the main idea.
- None of the questions require long derivations. If you find yourself plugging through lots of equations, consider giving less detail or moving on to the next question.
- Many questions have more than one right answer.

.

Q1: _____ / 1
Q2: _____ / 2
Q3: _____ / 1
Q4: _____ / 1
Q5: _____ / 2
Q6: _____ / 2
Q7: _____ / 2
Q8: _____ / 2
Q9: _____ / 2

Final mark: _____ / 15

Have you taken CSC321 at UofT? (This question is used for calibration purposes.)

1. **[1pt]** Give one reason why an algorithm implemented in terms of matrix and vector operations can be faster than the same algorithm implemented in terms of `for`-loops.

2. **[2pts]** Briefly explain two advantages of decision trees over K-nearest-neighbors.

3. **[1pt]** TRUE or FALSE: AdaBoost will eventually choose a weak classifier that achieves a weighted error rate of 0. Briefly justify your answer.

4. **[1pt]** In class, we considered using the squared Euclidean norm of the weights, $\|\mathbf{w}\|^2$, as a regularizer. Suppose we instead used the Euclidean norm $\|\mathbf{w}\|$ as a regularizer (i.e. without squaring it). Briefly explain one way in which this would lead to different behavior.

5. **[2pts]** Recall that hyperparameters are often tuned using a validation set.

   (a) **[1pt]** Give an example of a hyperparameter which it is OK to tune on the training set (rather than a validation set). Briefly explain your answer.

   (b) **[1pt]** Give an example of a hyperparameter which should be tuned on a validation set, rather than the training set. Briefly explain your answer.

6. **[2pts]** In this question, you will write NumPy code to implement a 1-nearest-neighbour classifier. Assume you are given an $N \times D$ data matrix `X`, where each row corresponds to one of the input vectors, and an integer array `y` with the corresponding labels. (You may assume the labels are integers from 1 to K.) You are given a query vector `x_query`. Your job is to return the predicted class (as an integer). Do not use a `for`-loop.

   If you don't remember the API for a NumPy operation, then for partial credit, explain what you are trying to do.

7. **[2pts]** Consider the classification problem with the following dataset:

x_1	x_2	x_3	t
0	0	0	1
0	1	0	0
0	1	1	1
1	1	1	0

Your job is to find a linear classifier with weights w_1, w_2, w_3, and b which correctly classifies all of these training examples. None of the examples should lie on the decision boundary.

(a) **[1pt]** Give the set of linear inequalities the weights and bias must satisfy.

(b) **[1pt]** Give a setting of the weights and bias that correctly classifies all the training examples. You don't need to show your work, but it might help you get partial credit.

8. **[2pts]** Recall that in bagging, we compute an average of the predictions $y_{\text{avg}} = \frac{1}{m}\sum_{i=1}^{m} y_i$, where y_i denotes the prediction made by the model trained on the ith bootstrapped dataset. Recall that these predictions are not fully independent, i.e. they are correlated because their training sets come from the same underlying dataset. Suppose $\text{Var}[y_i] = \sigma^2$ and the correlation between y_i and y_j is ρ for $i \neq j$. Give the formula for the variance $\text{Var}[y_{\text{avg}}]$. (If you don't remember the formula from lecture, you should be able to figure it out from the properties of covariance. In either case, you should justify where the formula comes from.)

9. **[2pts]** Consider a regression problem where the input is a scalar x. Suppose we know that the dataset is generated by the following process. First, the target t is chosen from $\{0, 1\}$ with equal probability. If $t = 0$, then x is sampled from a uniform distribution over the interval $[1, 2]$. If $t = 1$, then x is sampled from a uniform distribution over the interval $[0, 2]$. Give a function $f(x)$, defined for $x \in [0, 2]$, such that $y_* = f(x)$ is the Bayes optimal predictor for t given x. (Note that even though t is binary valued, this is a regression problem, with squared error loss.)

(Scratch work or continued answers)