Homework 1

Deadline: Thursday, Sept. 26, at 11:59pm.

Submission: You need to submit two files through MarkUs[1]:

- Your answers to Questions 1 and 3, and outputs requested for Question 2, as a PDF file titled `hw1_writeup.pdf`. You can produce the file however you like (e.g. LaTeX, Microsoft Word, scanner), as long as it is readable.
- Your code for Question 2, as the Python file `hw1_code.py`. This should contain the functions `load_data` and `select_model`.

Late Submission: 10% of the marks will be deducted for each day late, up to a maximum of 3 days. After that, no submissions will be accepted.

Computing: To install Python and required libraries, see the instructions on the course web page.

Collaboration: Homeworks are individual work. See the course web page for detailed policies.

1. [**3pts**] **Nearest Neighbours and the Curse of Dimensionality.** In this question, you will verify the claim from lecture that "most" points in a high-dimensional space are far away from each other, and also approximately the same distance. There is a very neat proof of this fact which uses the properties of expectation and variance. If it's been a long time since you've studied these, you may wish to review Chapter 6 of *Mathematics for Machine Learning*[2], or the Metacademy resources[3].

   (a) [**2pts**] First, consider two independent univariate random variables X and Y sampled uniformly from the unit interval $[0, 1]$. Determine the expectation and variance of the random variable Z, defined as the squared distance $Z = (X - Y)^2$. You are allowed to evaluate integrals numerically (e.g. using `scipy.integrate.quad` or `scipy.integrate.dblquad`), but you should explain what integral(s) you are evaluating, and why.

   (b) [**1pt**] Now suppose we sample two points independently from a unit cube in d dimensions. Observe that each coordinate is sampled independently from $[0, 1]$, i.e. we can view this as sampling random variables $X_1, \ldots, X_d, Y_1, \ldots, Y_d$ independently from $[0, 1]$. The squared Euclidean distance can be written as $R = Z_1 + \cdots + Z_d$, where $Z_i = (X_i - Y_i)^2$. Using the properties of expectation and variance, determine $\mathbb{E}[R]$ and $\mathrm{Var}[R]$. You may give your answer in terms of the dimension d, and $\mathbb{E}[Z]$ and $\mathrm{Var}[Z]$ (the answers from part (a)).

   (c) [**for your own benefit, not to be handed in**] Based on your answer to part (b), compare the mean and standard deviation of R to the maximum possible squared Euclidean distance (i.e. the distance between opposite corners of the cube). Why does this support the claim that in high dimensions, "most points are far away, and approximately the same distance"?

[1] `https://markus.teach.cs.toronto.edu/csc2515-2019-09`
[2] `https://mml-book.github.io/`
[3] `https://metacademy.org/graphs/concepts/expectation_and_variance`

2. **[3pts] Decision Trees.** In this question, you will use the `scikit-learn` decision tree classifier to classify real vs. fake news headlines. The aim of this question is for you to read the `scikit-learn` API and get a sense for the sort of functionality it provides. You may use any `scikit-learn` functionality, even if you feel like it trivializes the question. (This is not meant to be a hard question.)

   We will use a dataset of 1298 "fake news" headlines (which mostly include headlines of articles classified as biased, etc.) and 1968 "real" news headlines, where the "fake news" headlines are from `https://www.kaggle.com/mrisdal/fake-news/data` and "real news" headlines are from `https://www.kaggle.com/therohk/million-headlines`. The data were cleaned by removing words from fake news titles that are not a part of the headline, removing special characters from the headlines, and restricting real news headlines to those after October 2016 containing the word "trump". For your interest, the cleaning script is available as `clean_script.py` on the course web page, but you do not need to run it. The cleaned-up data are available as `clean_real.txt` and `clean_fake.txt` on the course web page.

   Each headline appears as a single line in the data file. Words in the headline are separated by spaces, so just use `str.split()` in Python to split the headlines into words.

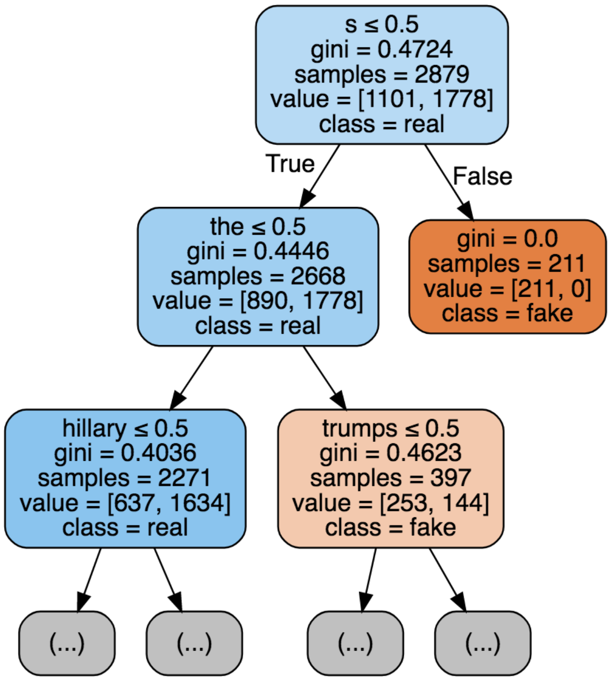
   You will build a decision tree to classify real vs. fake news headlines. Instead of coding the decision trees yourself, you will do what we normally do in practice — use an existing implementation. You should use the `DecisionTreeClassifier` included in `sklearn`. Note that figuring out how to use this implementation is a part of the assignment.

   All code should be included in the file `hw1_code.py` which you submit through MarkUs.

   (a) **[1pt]** Write a function `load_data` which loads the data, preprocesses it using a vectorizer (`http://scikit-learn.org/stable/modules/classes.html#module-sklearn.feature_extraction.text`), and splits the entire dataset randomly into 70% training, 15% validation, and 15% test examples.

   (b) **[1pt]** Write a function `select_model` which trains the decision tree classifier using at least 5 different values of `max_depth`, as well as two different split criteria (information gain and Gini coefficient), evaluates the performance of each one on the validation set, and prints the resulting accuracies of each model. Include the output of this function in your solution PDF (`hw1_writeup.pdf`).

   (c) **[1pt]** Now let's stick with the hyperparameters which achieved the highest validation accuracy. Extract and visualize the first two layers of the tree. Your visualization may look something like what is shown below, but it does not have to be an image: it is perfectly fine to display text. It may be hand-drawn. Include your visualization in your solution PDF (`hw1_writeup.pdf`).



This question is taken from a project by Lisa Zhang and Michael Guerzhoy.

3. [**4pts**] **Information Theory.** The goal of this question is to help you become more familiar with the basic equalities and inequalities of information theory. They appear in many contexts in machine learning and elsewhere, so having some experience with them is quite helpful. We review some concepts from information theory, and ask you a few questions.

   Recall the definition of the entropy of a discrete random variable X with probability mass function p: $H(X) = \sum_x p(x) \log_2 \left(\frac{1}{p(x)}\right)$. Here the summation is over all possible values of $x \in \mathcal{X}$, which (for simplicity) we assume is finite. For example, $\mathcal{X}$ might be $\{1, 2, \dots, N\}$.

   (a) [**1pt**] Prove that the entropy $H(X)$ is non-negative.

   An important concept in information theory is the relative entropy or the KL-divergence of two distributions p and q. It is defined as

$$\mathsf{KL}(p||q) = \sum_x p(x) \log_2 \frac{p(x)}{q(x)}.$$

   The KL-divergence is one of the most commonly used measure of difference (or divergence) between two distributions, and it regularly appears in information theory, machine learning, and statistics. For this question, you may assume $p(x) > 0$ and $q(x) > 0$ for all x.

   If two distributions are close to each other, their KL divergence is small. If they are exactly the same, their KL divergence is zero. KL divergence is not a true distance metric (since it isn't symmetric and doesn't satisfy the triangle inequality), but we often use it as a measure of dissimilarity between two probability distributions.

   (b) [**2pt**] Prove that $\mathsf{KL}(p||q)$ is non-negative. *Hint: you may want to use Jensen's Inequality, which is described in the Appendix.*

(c) **[1pt]** The Information Gain or Mutual Information between X and Y is $I(Y;X) = H(Y) - H(Y|X)$. Show that

$$I(Y;X) = \mathsf{KL}(p(x,y)||p(x)p(y)),$$

where $p(x) = \sum_y p(x,y)$ is the marginal distribution of X.

Appendix: Convexity and Jensen's Inequality. Here, we give some background on convexity which you may find useful for some of the questions in this assignment. You may assume anything given here.

Convexity is an important concept in mathematics with many uses in machine learning. We briefly define convex set and function and some of their properties here. Using these properties are useful in solving some of the questions in the rest of this homework. If you are interested to know more about convexity, refer to Boyd and Vandenberghe, Convex Optimization, 2004.

A set C is *convex* if the line segment between any two points in C lies within C, i.e., if for any $x_1, x_2 \in C$ and for any $0 \leq \lambda \leq 1$, we have

$$\lambda x_1 + (1-\lambda)x_2 \in C.$$

For example, a cube or sphere in $\mathbb{R}^d$ are convex sets, but a cross (a shape like X) is not.

A function $f : \mathbb{R}^d \to \mathbb{R}$ is *convex* if its domain is a convex set and if for all x_1, x_2 in its domain, and for any $0 \leq \lambda \leq 1$, we have

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2).$$

This inequality means that the line segment between $(x_1, f(x_1))$ and $(x_2, f(x_2))$ lies above the graph of f. A convex function looks like $\smile$. We say that f is *concave* if $-f$ is convex. A concave function looks like $\frown$.

Some examples of convex and concave functions are (you do not need to use most of them in your homework, but knowing them is useful):

- Powers: x^p is convex on the set of positive real numbers when $p \geq 1$ or $p \leq 0$. It is concave for $0 \leq p \leq 1$.
- Exponential: e^{ax} is convex on $\mathbb{R}$, for any $a \in \mathbb{R}$.
- Logarithm: $\log(x)$ is concave on the set of positive real numbers.
- Norms: Every norm on $\mathbb{R}^d$ is convex.
- Max function: $f(x) = \max\{x_1, x_2, \ldots, x_d\}$ is convex on $\mathbb{R}^d$.
- Log-sum-exp: The function $f(x) = \log(e^{x_1} + \ldots + e^{x_d})$ is convex on $\mathbb{R}^d$.

An important property of convex and concave functions, which you may need to use in your homework, is *Jensen's inequality.* Jensen's inequality states that if $\phi(x)$ is a convex function of x, we have

$$\phi(\mathbb{E}\left[X\right]) \leq \mathbb{E}\left[\phi(X)\right].$$

In words, if we apply a convex function to the expectation of a random variable, it is less than or equal to the expected value of that convex function when its argument is the random variable. If the function is concave, the direction of the inequality is reversed.

Jensen's inequality has a physical interpretation: Consider a set $\mathcal{X} = \{x_1, \ldots, x_N\}$ of points on $\mathbb{R}$. Corresponding to each point, we have a probability $p(x_i)$. If we interpret the probability as mass, and we put an object with mass $p(x_i)$ at location $(x_i, \phi(x_i))$, then the centre of gravity of these objects, which is in $\mathbb{R}^2$, is located at the point $(\mathbb{E}\left[X\right], \mathbb{E}\left[\phi(X)\right])$. If ϕ is convex $\smile$, the centre of gravity lies above the curve $x \mapsto \phi(x)$, and vice versa for a concave function $\frown$.