Homework 1, Machine Learning, Fall 2022

***IMPORTANT* Homework Submission Instructions**

1. All homeworks must be submitted in one PDF file to Gradescope.
2. Please make sure to select the corresponding HW pages on Gradescope for each question
3. For all coding components, complete the solutions with a Jupyter notebook/Google Colab, and export the notebook (including both code and outputs) into a PDF file. Concatenate the theory solutions PDF file with the coding solutions PDF file into one PDF file which you will submit.
4. Failure to adhere to the above submission format may result in penalties.

All homework assignments must be your independent work product, no collaboration is allowed. You may check your assignment with others or the internet after you have done it, but you must actually do it by yourself. **Please copy the following statements at the top of your assignment file**:

Agreement 1) This assignment represents my own work. I did not work on this assignment with others. All coding was done by myself.

Agreement 2) I understand that if I struggle with this assignment that I will reevaluate whether this is the correct class for me to take. I understand that the homework only gets harder.

1 Concepts of Learning (Ed Tam)

The goal of this question is to get you familiarized with the conceptual taxonomy of different types of machine learning tasks and to get you thinking about how ML could be applied in real life.
Some common types of machine learning tasks/problems are listed below:

- classification
- regression
- ranking
- clustering
- conditional probability estimation
- density estimation
- pattern mining

You are given a list of situations below. Assign one machine learning task from the list above to each situation below.

The situations are deliberately designed to simulate real-life applications and hence are open-ended. For each situation, there will be more than one answer that could be appropriate, depending on your interpretation of the data/task. **You only have to give ONE reasonable answer for each situation to get full credit.** Please limit any justification/explanation to at most two sentences.

1. You love betting on horse races. One popular betting mechanism is that if you guess the 1st, 2nd and 3rd place horses in a race correctly, you will win a grand prize. You have features about each horse, and would like to train a ML algorithm to create bets that would maximize the chances of winning a grand prize.

2. You are an astronomer studying galaxies. You have used the James Webb space telescope to collect 10000 images of different galaxies. Galaxies can be of 4 different types: spiral, elliptical, peculiar and irregular. You hand labelled 100 of these images yourself using your expert knowledge, and you would like an ML algorithm to distinguish what types of galaxies the remaining images correspond to.

3. You are a labor economist interested in having an ML algorithm being able to predict individuals' post-college incomes. You have a dataset on 1000 people, with data on each person's characteristics (e.g., educational level, what state they are from) and their starting salary in their first post-college job.

4. You are the owner of a restaurant that is famous for your vegetable soup. You are trying to determine how many pounds of vegetables to buy for next week. If you buy too much, the leftover vegetables go to waste. If you buy too little, you will run out of vegetables prematurely. You have a database that contains data about all past weeks (whether there's a holiday coming up, what the weather is likely to be, and how many customers you had that week, etc.).

5. You are a product analyst at Forever 21. You discover that different products sell at drastically different rates depending on the current fashion trends and the current season. You would like to use the available sales data to uncover some of the main current trends and correlations between different clothing products.

6. You are a UX researcher at a social network company. You are wondering whether the introduction of a new option will affect how much time users spend watching videos on the platform. In particular, you would like to learn about how the introduction of this new option will change the mean, kurtosis, and variance of the video-watch-time distribution. You start collecting data after the new option is implemented.

7. You are a salesperson that needs to pursue a set of customers. You have features about each of them, and would like to pursue them according to how likely it is that you will make a sale.

8. You are a mortgage specialist at Fannie Mae. You are trying to decide whether to extend a mortgage to a company for them to buy an office. You have a model that determines the likelihood of the company defaulting. You have some newly collected data which shows that the company has had a steady stream of cash flow for the past 5 years and has never defaulted on any loans before. In light of this new information, you would like the model to update its estimate of default.

9. You are a radiologist working on applying ML research in assisting medical diagnosis. In particular, you would like to develop an ML algorithm that can take in MRI scans of individual patients and output how likely it is that the patient is suffering from a malignant tumor.

10. You are a biologist studying the genetic lineage of different species. You have genomic data (a genome is the collection of all genes/genetic materials present in an organism) on 100 different species, and you would like group these species according to their "genetic distance" from one another.

2 Information Theory (Ed Tam)

To understand how decision trees work, it helps to internalize some fundamentals of information theory. In particular, the notion of "information gain" that is used in decision trees (see Decision Tree Lecture Notes for reference) is closely related to the notion of Kullback-Leibler divergence (KL-Divergence), entropy and mutual information in information theory. In the subquestions below, we will explore some of these connections.

Consider a discrete random variable X with distribution P on a set $\mathcal{O}$ of all possible outcomes that X can take. We use $P(a)$ to denote the probability $\text{Prob}(X = a)$ for any $a \in \mathcal{O}$.

Suppose we are given another discrete random variable Y with distribution Q on the same set $\mathcal{O}$ of outcomes. We use $Q(b)$ to denote the probability $\text{Prob}(Y = b)$ for any $b \in \mathcal{O}$.

Recall from class that the entropy of a single random variable X is defined as

$$H(X) = \sum_{a \in \mathcal{O}} -P(a) \log P(a).$$

We can also define the entropy of pair of random variables jointly. Let X, Y be jointly distributed according to the distribution J. Let $J(a, b)$ denote the probability $\text{Prob}(X = a, Y = b)$ for $a, b \in \mathcal{O}$. The joint entropy of X, Y is defined as

$$H(X, Y) = \sum_{(a,b) \in (\mathcal{O}, \mathcal{O})} -J(a, b) \log J(a, b).$$

We can also define a notion of entropy via conditioning. The conditional entropy of X conditioning on Y is defined as

$$H(X|Y) = \sum_{(a,b) \in (\mathcal{O}, \mathcal{O})} -J(a, b) \log \frac{J(a, b)}{Q(b)}.$$

For simplicity, throughout the question, we can assume that $P(a) > 0$, $Q(a) > 0$ and $J(a, b) > 0$ for any outcomes $a, b \in \mathcal{O}$.

Given P and Q, we can define the KL Divergence between them as:

$$KL(P, Q) = \sum_{a \in \mathcal{O}} P(a) \log \frac{P(a)}{Q(a)}.$$

In other words, it is a function that takes in two distributions as input, and returns a real number as output.

You can assume that all logarithms in the question are natural.

2.1

Show that the KL divergence between any discrete distributions P and Q on $\mathcal{O}$ is always a non-negative quantity.

(Hint 1: Jensen's inequality says that $\mathbb{E}(f(X)) \geq f(\mathbb{E}(X))$ for any convex function f, where $\mathbb{E}$ denotes the expectation.

(Hint 2: Note that $-\log$ is a convex function.)

2.2

Show that the KL Divergence between P and Q is equal to 0 if P and Q are the same distribution. (Note: this implication does not generally hold in the other direction)

2.3

Our results in 2.1 and 2.2 suggest that one intuitive way to think about the KL-Divergence is as a sort of "distance" between probability distributions. However, while this intuition is useful, it is well known that the KL-Divergence is not, strictly speaking, a distance. One reason that it is not a distance is that it is not always symmetric, i.e. $KL(P, Q)$ is not necessarily equal to $KL(Q, P)$ in general. A simple example can be shown using Bernoulli distributions. Come up with an example of two Bernoulli distributions and show that the KL between them is asymmetric.

2.4

Consider the joint distribution J of X and Y, and also consider the product distribution (which we denote as PQ) of X and Y, which is defined as $PQ(a, b) := P(a)Q(b)$. Show that $KL(J, PQ) = H(X) - H(X|Y)$.

2.5

Given X and Y, we can define a quantity called the mutual information between X and Y, denoted $I(X, Y)$. One way to define this is as $I(X, Y) := H(Y) - H(Y|X)$. Show that $I(X, Y) = KL(J, PQ)$.

2.6

Our work above has shown that $KL(J, PQ) = H(Y) - H(Y|X) = H(X) - H(X|Y) \geq 0$. Interpreting this in a decision tree context, we can see that splitting on an attribute decreases the entropy at a branch. This is a nice result, but you may wonder how entropy is related to our main goal of learning. It turns out that there is a close connection between entropy and misclassification error in decision trees.

For simplicity, consider a binary classification problem. You are inspecting the training samples at a node. Let p denoted the proportion of samples that are in class "0" at the node, and $1 - p$ denote the proportion of samples that are in class "1". You can treat p as a continuous variable that can range from 0 to 1.

The misclassification error at the node will be

$$\text{Error}(p) = \min(p, 1 - p).$$

The entropy at the node will be

$$H(p) = -p \log p - (1-p) \log(1-p).$$

For simplicity (and with no loss of generality), suppose you know that $p < 0.5$. Show that a smaller entropy H corresponds to a smaller misclassification error under this $p < 0.5$ setting.

3 Classifiers and Metrics - Coding (Stark)

Age	likeRowing	Experience	Income	Y
20	1	0	20	0
18	1	1	33	0
11	0	1	21	1
31	0	0	18	1
19	1	1	7	1
21	1	0	10	0
44	1	0	23	1
15	1	1	16	0
16	0	1	15	1
17	1	0	6	0

You are given the dataset above with feature vector $\mathbf{x}$ including Age, likeRowing, Experience and Income, and the binary label Y. You are also given a linear classifier $g(\mathbf{x}) = \boldsymbol{\theta}^\top \mathbf{x} + \theta_0$ and a non-linear classifier $f(\mathbf{x}) = \tanh(\boldsymbol{\theta}^\top \mathbf{x} + \theta_0)$, where $\boldsymbol{\theta} = (0.05, -3, 2.1, 0.008)$, $\theta_0 = 0.3$, and "tanh" function $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$. (In this question, you are expected to write functions from scratch, but where packages including matplotlib and numpy are allowed.)

(3.1) First calculate the value of $g(\mathbf{x})$ for each data point. What choices of threshold would minimize misclassification error?

(3.2) Calculate the value $f(\mathbf{x})$ for each data point. What choice(s) of threshold would minimize misclassification error? Compute the confusion matrix, precision, recall, and F1 score for one such threshold.

(3.3) For classifier $f(\mathbf{x})$, plot the ROC curve. Please plot the ROC curve as a continuous, connected set of lines. Plot the points on the ROC curve that represent decision points with the minimum classification error.

4 Calculate Splits (Stark)

As we learned from class, decision trees can be learned from data. The following table shows a dataset that forms the logical rule $A \lor (B \land C)$.

A	B	C	$A \lor (B \land C)$
1	1	1	1
1	1	0	1
1	0	1	1
1	0	0	1
0	1	1	1
0	1	0	0
0	0	1	0
0	0	0	0

4.1 Decision trees can be used to represent logical rules. Please draw a decision tree to represent $A \vee (B \wedge C)$, and make sure that your decision tree has the smallest possible number of nodes.

4.2 If you had split according to Gini index, which feature would you choose to split first? (If you encounter multiple features with tied values of Gini index, follow alphabetical order. Please show your steps.)

4.3 If you had split according to information gain, is it the same as in 4.2? (If you encounter ties in information gain, follow alphabetical order. Please show your steps.)

5 Classification with KNN and Decision Trees (Stark)

5.1 Coding In this problem you will experiment with the "carseat" dataset. This dataset contains 10 features, and 1 target ("Sales"). It is a binary classification task.

5.2 Coding Please use two of the following decision tree packages: sklearn.tree.DecisionTreeClassifier, GOSDT, chefboost, pydl8.5. For each package, utilizing the F1 score function implemented in Q3, report their F1 scores. For each of the trees, please experiment with tuning at least 1 parameter with K-Fold cross validation. (See hints below.)

5.3 Coding Implement your own KNN algorithm from scratch. It should support at least two different distance metrics (such as Euclidean distance, cosine distance, Manhattan Distance). For the KNN classifier you implemented, tune the number-of-neighbours parameter and choice of distance metric, evaluate them with the F1 score and report the best result in your notebook.

5.4 For this dataset, which implementation performed the best?

Hints: Available decision tree implementations include: sklearn CART, GOSDT+guesses (Pypi: "pip install gosdt" for mac/linux, does not yet work on windows), pydl8.5, (other implementations are also acceptable). The most powerful tools are GOSDT and DL8.5 so we suggest trying one of these if you are able to to install it (Sklearn and GOSDT will allow you to calculate tree sparsity). You are encouraged to try out GridSearchCV or other cross-validation parameter search functions, but be sure to implement a version from scratch by yourself for this problem. We have also provided a template, feel free to use and change it. You will have to make adjustments to encoding methods depending on package requirements.

(5.5) What is the difference between K-fold cross-validation and leave-one-out-cross-validation (LOOCV)? What are some of the disadvantages of the LOOCV? Please list two of them.

(5.6) For the carseat dataset, compared to accuracy, do you think F1 is a good evaluation metric for the dataset and why?

6 Consistency and Curse of Dimensionality in K-Nearest Neighbors (Ed Tam)

A classical algorithm used in supervised learning (in both regression and classification) is the K-Nearest Neighbor (KNN) Algorithm.

In this question, we will explore some theoretical properties of the KNN algorithm under a toy setup. The goal is to have you gain intuitive understanding on why the KNN algorithm works, and under what situations it will fail to work well.

One way to evaluate whether an algorithm makes sense/work well is to look at what happens when it has access to an infinite amount of training data. Ideally, we would like the algorithm to have 0 prediction error in the limit as the number of training data n approaches infinity. This property is called consistency in statistics. In 6.1 and 6.2, we will guide you through a consistency proof.

Consider the following setting. Suppose we have d binary features. We can write our feature vector then as $\mathbf{x} \in \{0,1\}^d$. We can define distance between feature vectors that measures their "similarity" as:

$$\rho(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{d} I(\mathbf{x}_i \neq \mathbf{x}'_i).$$

ρ is called the Hamming distance.

Suppose we have training data $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$. Assume that the training features $x_1, x_2, \cdots, x_n$ are drawn I.I.D. from the uniform distribution over the d-dimensional hypercube $\{0,1\}^d$. Assume that the true relationship between the labels (y's) and the features (x's) is given by the formula $y = f(x)$, where f is an unknown but deterministic function that maps from $\{0,1\}^d \to [0,1]$.

Assume that f satisfies the following property

$$|f(\mathbf{x}) - f(\mathbf{x}')| \leq L \cdot \rho(\mathbf{x}, \mathbf{x}')$$

where L is a positive constant. Such a function is called a Lipschitz function.

6.1

Consider the simplified case where we run a KNN algorithm with $K = 1$. This means that when presented with an observed test point x_{test}, this algorithm will find the nearest neighbor of x_{test} (i.e., the closest training point) and use the label of $NN(x_{test})$ as the prediction for x_{test}. We will use $NN(x_{test})$ to denote the closest training point of x_{test}.

Show that as $n \to \infty$, $NN(x_{test})$ converges in probability to x_{test} (under the Hamming distance).

(Review: A sequence of random variable Z_n converges in probability to a target T under the Hamming distance ρ if for any $\epsilon > 0$, $\lim_{n\to\infty} P(\rho(Z_n, T) > \epsilon) = 0$. The target T can be a deterministic constant or a random variable.)

6.2

Based on the result in 6.1, argue that the absolute error loss

$$|f(NN(x_{test})) - f(x_{test})|$$

converges to 0 in probability as n goes to infinity.

6.3

We have shown in 6.1 and 6.2 that KNN (when $K = 1$) is a consistent algorithm under a regression setting with categorical features. (The argument is very similar for general K if you are interested).

KNN is a powerful algorithm, and it can be often be used in a latent representation setting in conjunction with other models like neural networks to tackle difficult, high-dimensional machine learning problems such as image classification. However, KNN is not directly useful for high dimensional problems in the original feature space. The reason is that the KNN algorithm suffers from the curse of dimensionality, i.e., when the dimension d of the features is big, the algorithm performs poorly. Below, we will do some quick math to gain intuition on why this is true.

We are given an observed test point on the d-dimensional hypercube $\mathbf{x}_{test} \in \{0,1\}^d$. One natural way to think about nearest neighbor algorithms is to consider any point within some Hamming distance r of $\mathbf{x}_{test}$ as a reasonable neighbor of $\mathbf{x}_{test}$. This can be formalized via the notion of a Hamming neighborhood around $\mathbf{x}_{test}$, denoted as $N_r(\mathbf{x}_{test}) := \{\mathbf{x} \in \{0,1\}^d : \rho(\mathbf{x}_{test}, \mathbf{x}) \leq r\}$.

You are given n points $x_1, x_2, \cdots, x_n$ drawn I.I.D. from the uniform distribution on the d-dimensional hypercube $\{0,1\}^d$. Let K' denote the expected number of points that will lie within the Hamming neighborhood $N_r(\mathbf{x}_{test})$. For simplicity, assume $r = 2$. Find an expression of K' in terms of d and n.

6.4

For fixed sample size n, what happens to K' as $d \to \infty$? Considering your solution to 6.3 as a fraction, how fast (polynomial or exponential) does the numerator grow as you let d go to infinity? How fast does the denominator grow?

6.5

Based on your answer to 6.4, give an intuitive explanation for why KNN does not work well in high dimensions in no more than three sentences.