MAT 3373: HOMEWORK 1

Homework Policy Basics

(1) You can find homework deadlines in the course schedule, available on Brightspace.
(2) You can find detailed homework grading policies in the first week's lecture notes, available on Brightspace.
(3) You can find overall grading policies in the syllabus, available on Brightspace.
(4) Homework should be written in R Markdown format.[1] You should always submit both the raw workbook (in .Rmd format) *and* a compiled version where I can see the charts (in .html or .pdf format). **Marks will be taken off if the homework is not compiled correctly.** I suggest using the *set.seed()* command near the top of each homework set, and only re-compiling chunks as you need to. This will reduce the chance of bugs popping up in previously-completed parts of the homework.
(5) Homework solutions should appear in the same order as the questions in the document. I will dock 3 percent on assignments with out-of-order solutions, and I will generally *not* post grades of individual questions.
(6) It is possible to submit several versions of the homework. I will always grade the last one submitted before the deadline. With this in mind, **I strongly encourage you to submit drafts of your homework well before the deadline.** In particular, you might as well submit a draft every time you finish a question.

[1] See first week's lecture notes for comments on Python people.

Introduction and Homework Focus

This homework is concerned with material from Chapters 1-4 of the textbook, with one introductory question from Section 5.2. It focuses on getting practice with programming, and on the methods of regression, logistic regression, and k-nearest neighbours. It uses only the basic statistical tests seen in Chapters 1-4. In particular, the homework doesn't use the main tools for statistical analysis of a collection of very different models, which are introduced in Chapter 5. As such, many of the questions are either straightforward applications of the methods (1,2,7,10) or ask you to identify and resolve an *enormous* problem that appears when you try a straightforward application (3,4,5,8,9).

Finding and fixing these sorts of enormous problems is of course important - all of the mistakes highlighted in this homework are fairly easy to make, even for professionals. Still, this might give the impression that machine learning is mostly about identifying these sorts of gross violations of modelling assumptions. In the following homework set, we'll have access to more of the machine learner's standard toolkit, and we'll do more examples where a "normal" workflow gives sensible results that can be improved with small tweaks and good statistical practice.

1. Introduction to Data Exploration

Do Question 8 from Chapter 2 of the textbook.

2. Empirical Study: KNN for MNIST Data

Open the (small subsample of the) MNIST testing and training datasets:

```
MNIST_train = read.csv("mnist_train.csv")
MNIST_test = read.csv("mnist_test.csv")
```

Fit the knn classifier using the training data for $k \in \{3, 4, 5, 6\}$. Select the value of k that minimizes the test error, and calculate the confusion matrix.

Comment on the results. If you saw a collection of new datapoints from the same collection, would you expect the error to be larger, smaller, or about the same as the observed test error?

3. Short Conceptual Questions

(1) Do Question 3 from Chapter 3 of the textbook.

(2) Imagine that you're going to collect data, and have committed to doing a one-dimensional linear regression analysis with intercept parameter β_0 known to be 0 and variance parameter σ known to be 1 (so you are just trying to learn the slope term β_1). Furthermore, you are absolutely certain that the linear regression model is true.[2]

Before collecting the response variable, you need to choose the predictors. You have the following options:

$$X^{(1)} = (-12, -9, -6, -3, 0, 3, 6, 9, 12)$$
$$X^{(2)} = (-1, -0.9, -0.8, \ldots, 0.8, 0.9, 1)$$
$$X^{(3)} = (-28, -2, 76, 412).$$

Which of those should you choose? Why?

(3) People often use statistical models to do optimization, as follows. You have some function f, and would like to find the largest value $\max_x f(x)$. You try the following procedure:
   (a) Collect data $(X_1, Y_1), \ldots, (X_n, Y_n)$.
   (b) Based on this data, estimate a model $\hat{f}$.
   (c) Use the maximum of the predicted values $\text{argmax}_x \hat{f}(x)$ as a prediction for the true location of the maximum value $\text{argmax}_x f(x)$.

[2] Of course this is unrealistic - but please take it seriously for this question.

This is a reasonable thing to try: you get an estimate $\hat{f}$ for an entire function, and you can try to use that as a surrogate for the real function when doing optimization (or any other task). For *some* models, this approach works rather well. Explain why it will almost never work well when the model is a linear regression model on all of $\mathbb{R}$.

4. Lying with Linear Regression

It is often possible to get very misleading results by deliberately choosing a bad model with a certain structure. In this question, we'll practice doing this in a simple setting.[3]

Throughout the question, we'll assume that every X-value is paired with the observed Y-value $Y = \sin(X)$; there is no measurement error. We will then fit this data to the usual linear regression model with *unknown* β_0, β_1, σ.

(1) Find a collection of $X_1, \ldots, X_n$ of predictors so that the 99-percent confidence interval for the slope β_1 is contained in the interval $(-\infty, -0.95]$. Display the collection of points, fit the model in R, and give the output of the *summary* function applied to the fit.

(2) Find a collection of $X_1, \ldots, X_n$ of predictors so that the 99-percent confidence interval for the slope β_1 is contained in the interval $[0.95, \infty)$. Display the collection of points, fit the model in R, and give the output of the *summary* function applied to the fit.

(3) Would it be possible to do part (2) of this question if I replaced the interval $[0.95, \infty)$ by the interval $[100, \infty)$? Why or why not?

Note: A complete proof is not required, but will be considered for bonus marks.

5. Simulation Study: Post-Selection Inference

Generate predictors "*Pred*" and response variables "*Resp*" according to the following R code:

```
m= 80 # You can experiment with your own large value.
n=100 # You can experiment with your own large value.
Pred = matrix(rnorm(m*n,0,1), nrow = n, ncol = m)
Resp = rnorm(n,0,1)
```

We interpret this as n datapoints, with the j'th datapoint looking like $(X_1^{(j)}, \ldots, X_m^{(j)}, Y^{(j)})$.

(1) Do the following two steps:

[3] In case it wasn't clear: you shouldn't actually do this! However, seeing very misleading results can help you learn to diagnose problems with statistical analyses.

(a) For each $i \in \{1, 2, \ldots, m\}$, compute the correlation of each $(X_i^{(1)}, \ldots, X_i^{(n)})$ with $(Y^{(1)}, \ldots, Y^{(n)})$.
Denote by i_1, i_2 the indices with the *largest* correlation in absolute value.

(b) Fit the linear model

$$Y = \beta_0 + \beta_1 X_{i_1} + \beta_2 X_{i_2} + \epsilon$$

based on the response variable Y and the two chosen predictors. Comment on the fit and the confidence intervals for these quantities.

(2) Split the data into two equal-sized parts - a training dataset and a testing data set.

Repeat the above question, but use *only* the training dataset in part (a) and *only* the testing dataset in part (b). Comment on the difference between the results you see.

6. Theory: Consistency of K-NN Classification

We will prove that the k-nearest neighbour classification algorithm eventually gets the right answer, at least in a simple setting.

Define the function $f : [0, 1] \mapsto \{0, 1\}$ by the piecewise-constant formula

$$f(x) = 0, \qquad x < 0.5$$
$$f(x) = 1, \qquad x \geq 0.5.$$

Consider data $X_1, X_2, \ldots \overset{i.i.d.}{\sim} Unif[0, 1]$ and let $Y_i = f(Y_i)$. For integer $k \in \{3, 5, 7, \ldots\}$ and $n \in \{k+1, k+2, \ldots\}$, let $\hat{f}_{k,n}$ be the k-nearest neighbour classifier associated with the dataset $\{(X_1, Y_1), \ldots, (X_n, Y_n)\}$.

Prove that, for any fixed $x \in [0, 0.5) \cup (0.5, 1]$ and any fixed $k \in \{3, 5, 7, \ldots\}$, we have

$$\lim_{n \to \infty} P[\hat{f}_{k,n}(x) = f(x)] = 1.$$

Hint 1: You don't have to *directly* estimate the probability of the event $A_n = \{\hat{f}_{k,n}(x) = f(x)\}$ in the question. It is enough to estimate the probability of some event B_n *contained in* A_n.

Hint 2: If you're stuck proving this for *all* x, start with a concrete number such as $x = 0.75$. Can you write down a formula for the probability of the event "at least k of the points $X_1, \ldots, X_n$ are in the interval $(0.5, x)$"? When that event happens, what can you say about $\hat{f}_{n,k}(x)$?

Hint 3: You may find the following fact useful: for *any* constants $\alpha, \beta > 0$,

$$\lim_{n\to\infty} n^{\alpha} e^{-\beta n} = 0.$$

You may remember this as a phrase like "exponentials beat polynomials" from a plotting section in calculus class.

7. Empirical Study: Bayes Rate of Classifiers

Open the dataset *weight-height.csv*. This dataset was downloaded from Kaggle.com. Split this dataset (randomly) into a training set and a testing dataset, each with 2500 men and 2500 women.

(1) For both genders separately, calculate the sample average weight, sample average height, and the sample variance for the weight and height based on this training dataset. **Note:** you should have eight numbers at this point - for each of two genders, and each of two measurements, you should both the mean and variance.

(2) For the remainder of the question, assume that the male data is *exactly* bivariate Gaussian with means and variances as calculated in the previous part of the question (and 0 covariance). Similarly, assume that the male data is *exactly* bivariate Gaussian with means and variances as calculated in the previous part of the question (and 0 covariance). [4]

Based on this modelling assumption and using only the observed height and weight, predict the gender of the 5000 elements of the testing set according to the Bayes classifier (see page 38 of the textbook for a definition).

Comment on the quality of this classifier. Can you think of anywhere that the model can be improved, or does it look about as good as you could hope for (given the data)?

Hint: You may wish to make an illustrative plot along the following lines: let the x-axis and y-axis be the observed weight and height, and colour each point in one of 4 colours corresponding to true/predicted gender. You need not make exactly this plot (and indeed you will probably need to make small adjustments to end up with a useable picture).

[4] Of course, this is not really the true data-generating process. I'm making this assumption because it is only possible to calculate a Bayes rate if you know the *true* data-generating process. As you can probably guess, this means that you can almost never calculate the *true* Bayes rate. Nonetheless, it is useful to think about the Bayes rate to try to understand fundamental limits with any classification procedure.

(3) Compute the average Bayes error rate, *using the same modelling assumption as in the previous part of this question.*[5] Use a plot or other data-summarization technique to check if the Bayes error rate seems sensible, and explain/interpret the broad features of your plot.

8. Data Analysis: House Sizes and Collinearity

Open the file

`House_Data.csv`

using the command:

```
house = read.csv("House_Data.csv")
```

This dataset is a version of the dataset posted at https://www.kaggle.com/c/house-prices-advanced-regression-techniques/overview. The data itself is real; I just pruned various outliers to make the following data analysis slightly easier.

In the following, I denote by A_i, F_i and S_i the lot area, first floor area and second floor area of the i'th house. In the dataset, these can be accessed by writing:

```
house$LotArea[i], house$X1stFlrSF[i], house$X2ndFlrSF[i]
```

We will consider the following "full" regression model

$$A_i = \alpha + \beta_1 F_i + \beta_2 S_i + \epsilon_i \tag{8.1}$$

and the two "sub-"models

$$A_i = \alpha + \beta_1 F_i + \epsilon_i \tag{8.2}$$

and

$$A_i = \alpha + \beta_2 S_i + \epsilon_i. \tag{8.3}$$

Note: Equations (8.1), (8.2) and (8.3) are three different models - I'm not writing down one model where all three of these equations hold simultaneously! In reality none of these models are quite true, and when you fit them you will not get the same estimates for $\alpha, \beta_1, \beta_2, \epsilon_i$ in all three models.

(1) Plot F_i vs S_i. Do you notice anything about their relationship?
(2) Use the *lm* function to fit all three models and find p-values for the parameters α, β_1, β_2.

[5] Again, computing the Bayes error rate requires you to know (or assume you know) the true data-generating process.

(3) When you fit model (8.1), are the p-values for β_1, β_2 below 0.05? What about the p-values when you fit models (8.2) and (8.3)? Comment on any discrepancy, perhaps in light of the plot in the first part of the question.
(4) In the first three parts of the question, I told you to plot F_i, S_i together and *then* fit three models for A_i. In light of the plot from part **(1)** (and common-sense observations about the relationship between A_i, F_i, S_i), suggest a modelling procedure that would have produced better results.

9. Simulation Study: Bootstrap

There are several closely-related algorithms called "the boostrap." We first describe the bootstrap that will be studied in this question.

We have some parametric model $\{f_\theta\}_{\theta\in\Theta}$, data $X_1, \ldots, X_n \in \mathbb{R}$, and estimator $\hat{\theta} : \mathbb{R}^n \mapsto \Theta$ taking a dataset to a parameter value. A single *bootstrap replicate* is obtained by the following algorithm:

(1) Sample $X_1^*, \ldots, X_n^*$ uniformly and with replacement from the dataset $X_1, \ldots, X_n$.
(2) Return $\Theta^* = \hat{\theta}(X_1^*, \ldots, X_n^*)$.

Denote by $\theta_0 \in \Theta$ the "true" parameter, F the true distribution of $\hat{\theta}(X_1, \ldots, X_n)$, and F_{boot} the distribution of Θ^*.

(1) Consider the parametric model $f_\theta = N(\theta, 1)$ with the usual MLE $\hat{\theta} = \frac{1}{n}\sum_{i=1}^n X_i$. Fix $n = 100$ and $\theta = 0$. Do a simulation study and plot the densities of both F and F_{boot}. Do they look similar? [6]
(2) Consider the parametric model $f_\theta = \text{Unif}[0, \theta]$. Show that the MLE is $\hat{\theta} = \max(X_1, \ldots, X_n)$.
(3) Set $n = 100$ and $\theta = 1$. Do a simulation study to plot the densities of both F and F_{boot} for the setup in part (2).[7] Do they look similar? If not, comment on what would "go wrong" if you used F_{boot} as a surrogate for F.

Depending on how you plot initially, the densities might be "squished" and hard to see - rescale your x- and y-axes until you get a good luck,

[6] It is possible to compute F exactly, but computing F_{boot} is much harder. Rather than trying to do this, you should take many samples from F_{boot} using the bootstrap algorithm given at the start of the question, then plot the histogram or density plot of this sample.

[7] As in the previous part of the question, you can estimate F and F_{boot} by simulating from the appropriate densities and plotting the empirical distribution of the densities. However, it is not terribly difficult to compute both of these densities *exactly* using only tools from MAT2371/2377. If you can do these computations, you may find it easier to plot and compare them.

and comment in particular on the values in intervals that are very close to 1, such as $[0.998, 1]$.

10. Logistic Regression Practice

Do Parts (a,b,c,d,e) of Question 10 from Chapter 4 of the textbook. Comment on the quality of the result.