Homework 1

Due: 11.59pm on Friday, January 19

Submission instructions: Submit one write-up per group on gradescope.com. Group size should not be bigger than 3 people. You can work by yourself on the homework.

If you were not added to gradescope, you can register using the code: **MGE7Y8**

If you have not used gradescope before:

- This PDF guide explains how to create high-quality scans of your solutions http://gradescope-static-assets.s3-us-west-2.amazonaws.com/help/submitting_hw_guide.pdf
- This video guide also exlains how to upload the PDF of your solution https://www.youtube.com/watch?v=-wemznvGPfg

Please do not bring printouts of your solutions to the classroom.

Question 1

In this question, you will run a simulation study to explore the bias-variance trade-off in more depth. The exercise will walk you through prediction using linear regression versus k-NN. You will discover under what circumstances one outperforms the other (in terms of the test error). In the end you will reproduce something similar to what you see in figures 3.17 through 3.20 in Section 3.6 of *Introduction to Statistical Learning.*

1. You will start by exploring a scenario where the true relationship between x and y is linear. You will generate data from the linear model
$$y = f(x) + \epsilon$$
where $f(x) = 1.8x + 2$. To create a synthetic training set, make 100 independent draws of x from a standard normal distribution to form $\{x_1, \cdots, x_{100}\}$. Similarly draw random noise ϵ from standard normal to have $\{\epsilon_1, \cdots, \epsilon_{100}\}$. For each $i \in \{1, \cdots, 100\}$, generate y_i from the linear model where $y_i = 1.8x_i + 2 + \epsilon_i$. Now, repeat the above process and generate additional 10,000 observations that will form the test set. Note that we are constructing a huge test set in order to have an accurate out-of-sample MSE. You will use this simulated dataset to try out and compare different models.

2. Create a scatter plot of y vs x. In the same figure, draw the true relationship in black solid line.

3. Using ordinary linear regression, find a relationship between y and x of the form
$$y = b_0 + b_1 \times x + e$$
using the training data you simulated. On the same plot from the last question, draw a blue dashed line that is the least squares fit to the data.

4. You may find that the linear regression line provides a very good estimate of $f(X)$. Now, use k-NN to find the relationship between y and x. You should experiment with $k = 2, 3, \cdots, 15$ to see how model complexity affects prediction accuracy. On one plot, redraw the scatter plot and the true relationship, but this time overlay it with predicted fit using k-NN with $k = 2$. On a juxtaposed graph, do the same for $k = 12$.

5. Plot the test set mean squared error using k-NN against $\log(1/k)$ for $k = 2, 3, \cdots, 15$. On the same graph, draw a horizontal dashed line that represents the test set mean squared error using linear regression. Which model performs the best? Comment on the relative performance of linear regression and k-NN with different values of k.

6. Redo 1-5, but consider a different data generating process where the true relationship between x and y is near, but not perfectly linear. Precisely, simulate data from the following model

$$y_i = \tanh(1.1 \times x_i) + 2 + \epsilon_i.$$

How does your conclusion in 5 change?

7. Consider yet another data generating process where the true relationship is strongly non-linear. Simulate data from the following model

$$y_i = \sin(2x_i) + 2 + \epsilon_i.$$

What can you say about the relative performance of the two methods now?

8. You might suspect, from your previous results, that in real world when none of the relationship could be linear, it would always pay off using k-NN over linear regression. Examine this hypothesis in the situation with more than 1 variable. Stay with the same true relationship from the model in 7, add additional noise variables that are not predictors of y. In particular, consider the model

$$y_i = \sin(2 \times x_{i1}) + 2 + 0 \times x_{i2} + \cdots + 0 \times x_{ip} + \epsilon_i$$

where p ranges from 2 to 20. Additional, noisy $x_{i2}, \ldots, x_{ip}$ variables are drawn from independent standard normal distributions. Create a plot for every p taking values in $1, 2, 3, \cdots, 20$. Each plot should have the test set MSE against model complexity $\log(1/K)$ and a dashed horizontal line showing test set MSE for best linear fit. Briefly explain the result of the simulation.

OPTIONAL BONUS QUESTION: Suppose that instead of 100 training samples, you had 1,000 training samples. Would that change consluciones you made above? Think about how the range of values of k for which k-NN does bettter that linear regression would change. What does having a large traing set allow you to do?

Question 2

In this question, you will explore prices of used cars as a function of different input variables. You can download data from:
https://github.com/ChicagoBoothML/MLClassData/raw/master/UsedCars/UsedCars.csv

1. Take a look at the data-set and describe for what kind of business related problems you could use this data. That is, why would anyone care to collect this data?

2. Split data into two parts: training set consisting of 75% of observations and a test set consisting of 25% of observations.

3. Using ordinary linear regression, find a relationship between `price` and `mileage` of the form

$$\texttt{price} = b_0 + b_1 \times \texttt{mileage} + e$$

using the training data. Create a scatter plot of `price` vs `mileage`. Include the best linear regression fit onto the plot.

4. You might notice that the linear fit does not capture the true relationship well. Perhaps we can do better with a polynomial function. Recall polynomial regression from applied regression or business statistics class (see also Section 7.1 of Introduction to Statistical Learning) and fit polynomials of the form

$$\texttt{price} = b_0 + b_1 \times \texttt{mileage} + b_2 \times \texttt{mileage}^2 \ldots + b_d \times \texttt{mileage}^d + e.$$

Use cross-validation to pick the optimal degree of the polynomial. Plot the cross-validation estimate of the mean-squared error as a function of the degree of the polynomial. Also, report the optimal polynomial degree and add the best fitted polynomial to the scatter plot you created before. Remember to refit the polynomial on all of the training data once you choose the optimal degree.

5. Use k-NN and regression trees to find the relationship between `price` and `mileage`. Again, use cross-validation to find the optimal tuning parameters for these two procedures: k for k-NN and the number of leaves for decision trees. Create plots showing the cross-validation estimate of the mean-squared error as a function of tuning parameters. Create a scatter plot of `price` vs `mileage` that also shows the best polynomial regression, k-NN, and regression tree fit. Which fit would you use and why? Report the test error for the model you select.

6. Now try using `mileage` and `year` to predict `price` using k-NN and regression trees. Remember to rescale data when using k-NN. As before, use cross-validation to pick optimal tuning parameters and plot the cross-validation error curves. How does the optimal k change as compared to when you were using only `mileage`? How about the size of the optimal tree? Does your model perform better when including `mileage` as an additional variable?

7. Finally, run a regression tree using all the variables to predict `price`. Use cross-validation to select the optimal size of the regression tree. Report the mean-squared error on the test set for the optimal tree.

OPTIONAL BONUS QUESTION: We can use cross-validation to select relevant variables for predicting the price of a used car. Try finding out whether all variables are predictive using regression trees and cross-validation. Think about and describe how would you try to find a simpler model, that is, one that does not include all the variables.