COM S 474/574 Spring 2020
Introduction to Machine Learning

Homework 2

1 Directions:

- **Due: Thursday February 27, 2020 at 10pm.** Late submissions will be accepted for 24 hours after that time, with a 15% penalty.
- Upload the homework to Canvas as a pdf file. Answers to problems 1-4 can be handwritten, but writing must be neat and the scan should be high-quality image. Other responses should be typed or computer generated.
- Any non-administrative questions must be asked in office hours or (if a brief response is sufficient) Piazza.

2 Problems

In this homework, we will focus on using nearest neighbor methods and logistic regression to classify (predict a discrete-valued feature y, such as $y \in \{0, 1\}$).

Problem 1. [5 points] Suppose you are predicting feature y using feature x with logistic regression, and x is measured in kilometers. After fitting, you get coefficients $\beta_0 = 1.24$ and $\beta_1 = -3.74$. Thus, your model is

$$\text{Prob}(y = 1|x) = \frac{e^{1.24-3.74x}}{1 + e^{1.24-3.74x}}.$$

Suppose our friend Sammie has an innate fear of the metric system, starts with the same data set, converts the x values to miles, does not change y values, and then fits. What will Sammie's β_0 and β_1 be?

Problem 2. [15 points] Book problem Chapter 4, #4 "When the number of features ..."

Problem 3. [10 points] Book problem Chapter 4, #6 "Suppose we collect data ..."

Problem 4. [5 points] Book problem Chapter 4, #8 "Suppose that we take a data set ..."

Note: For the following problem, instead of reporting training/testing loss, for simplicity you will be asked to report accuracy. Accuracy is the percentage of samples that were correctly labeled as $\widehat{y} = 0$ or $\widehat{y} = 1$.

Problem 5. [65 points]

A. Download the data sets `HW2train.csv` and `HW2test.csv` from Canvas. In both files, the first column is a binary-valued feature y. The second column is a continuous-valued feature x. Make a scatter-plot of the data-set HW2train with y values on the vertical axis, x values on the horizontal axis.

B. Fit a logistic model to predict y. Use the whole data set HW2train. If you are using Python, you can use `https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html`.

- The argument 'penalty' is whether we want to penalize the coefficients. For this assignment, we will use 'penalty'=none.
- Set the argument 'fit_intercept'=True to add a β_0 (provided that we do not include a column of ones when we call the .fit() function).

(1) Report the β_0 and β_1 values you obtain.

(2) Report the accuracy for HW2train (you can do this with the .score() function).

(3) Also, make a copy of the scatter-plot of the data-set HW2train plot. Add the function $\text{Prob}(y = 1|x)$ on the plot.

- To plot the $\text{Prob}(y = 1|x)$ function, you can first generate uniformly spaced values along the horizontal axis, such as with `https://docs.scipy.org/doc/numpy/reference/generated/numpy.linspace.html` 1000 values evenly spaced between 0 and 100 should be enough for a good picture.
- then determine $\text{Prob}(y = 1|x)$ for each of those evenly spaced points. One way to obtain this is the .predict_proba() function `https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html#sklearn.linear_model.LogisticRegression.predict_proba` using those 1000 evenly spaced values as inputs; that function will return an array with $\text{Prob}(y = 0|x)$ for the first column and $\text{Prob}(y = 1|x)$ for the second; use the second column. Alternatively, you can use the β_0 and β_1 coefficients to calculate $\text{Prob}(y = 1|x)$ by yourself.

The scatter plot markers of the HW2train data should be plotted on top of the function $\text{Prob}(y = 1|x)$ (i.e. in the foreground), so it is not painted over by the $\text{Prob}(y = 1|x)$ function. Title this plot 'HW2train Scatter Plot and $\text{Prob}(y = 1|x)$'.

(4) Next make another scatter plot titled 'HW2test Scatter Plot and $\text{Prob}(y = 1|x)$' just like the previous plot except where you plot the HW2test data instead of HW2train data. Include the same $\text{Prob}(y = 1|x)$ function as in the previous plot.

(5) Report the total accuracy for the HW2test data.

C. Now we will try k-nearest neighbors. Our predictions will be based on the data set HW2train and odd-values of k, using majority vote. Since we only have a single

(one-dimensional) feature, x, we will measure the distance between sample i and a new sample using the absolute difference, $|x(i) - x(\text{new})|$.
You can implement $k$nn manually or by using built in functions. For Python, you can use `https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClass html#sklearn.neighbors.KNeighborsClassifier` Some usage notes

- the argument 'n_neighbors' is k, so set 'n_neighbors'=1 when you just want to use the nearest neighbor.
- set the argument 'weights'='uniform' (for this assignment we will just use uniform weights, but we encourage you to explore what happens with 'weights'='distance' which uses a built-in distance weighting or one you make yourself)
- the argument 'algorithm' effects how many distances are computed to find the nearest neighbors for a new sample. Use 'auto' for this assignment.

(1) For each value of $k \in \{1, 3, 9\}$

a. Fit the $k$nn classifier using the HW2train data set. Report the training accuracy (if using Python, you can use the .score()); briefly mention how you calculated it, such as if you used .score() or some other way.

b. Make a plot of the classifier's prediction $\widehat{y}(x)$ function. This should be a step-function (piece-wise constant), though your plot of the function can have steeply slanted lines instead of perfectly vertical jumps. Also plot HW2train data in the foreground (as a scatter plot). Use the title '1nn Classifier with Training data' for $k = 1$ and similar titles for other k.

c. Report the total accuracy for HW2test data set.

d. Make another plot, also with the classifier's prediction $\widehat{y}(x)$ function, but show the HW2test data instead. Use the title '1nn Classifier with Testing data' for $k = 1$ and similar titles for other k.

(2) Make a plot with the title 'Training accuracy as a function of k' where the horizontal axis is the parameter k with the odd-numbered values $\{1, 3, 5, \ldots, 13, 15\}$. The vertical axis should be the training accuracy for the HW2Train data set using the $k$nn classifier fit on the HW2Train data.

(3) Make a plot with the title 'Testing accuracy as a function of k' where the horizontal axis is the parameter k with the odd-numbered values $\{1, 3, 5, \ldots, 13, 15\}$. The vertical axis should be the training accuracy for the HW2Test data set using the $k$nn classifier fit on the HW2Train data.

D. In about 4-6 sentences, comment on the performance of the different nearest neighbor classifiers for the different k values you used, including whether you see any evidence of over-fitting or under-fitting, and how they compare to the logistic regression classifier, and any other note-worthy aspects.