HOMEWORK 3

RYAN YEE
9074025223

Instructions: Use this latex file as a template to develop your homework. Submit your homework on time as a single pdf file to Canvas. Late submissions may not be accepted. Please wrap your code and upload to a public GitHub repo, then attach the link below the instructions so that we can access it. You can choose any programming language (i.e. python, R, or MATLAB). Please check Piazza for updates about the homework.
Code used for this homework can be found at https://github.com/ryanyee3/CS760_Machine_Learning/tree/master/homework3.

1 Questions (50 pts)

1. (9 pts) Explain whether each scenario is a classification or regression problem. And, provide the number of data points (n) and the number of features (p).

   (a) (3 pts) We collect a set of data on the top 500 firms in the US. For each firm we record profit, number of employees, industry and the CEO salary. We are interested in predicting CEO salary with given factors.

   This is a regression problem. We are interested in predicting salary which is a continuous variable, not categorical. $n = 500$ since we have data from 500 companies. $p = 3$ corresponding to record profit, number of employees, and industry.

   (b) (3 pts) We are considering launching a new product and wish to know whether it will be a success or a failure. We collect data on 20 similar products that were previously launched. For each product we have recorded whether it was a success or failure, price charged for the product, marketing budget, competition price, and ten other variables.

   This is a classification problem. We are interested in predicting if a product will be successful which has a binary outcome. Thus, we would like to classify our product as successful or unsuccessful. $n = 20$ since we have 20 products we can use as comparisons. $p = 13$ corresponding to product price, marketing budget, competitor price, and 10 other variables.

   (c) (3 pts) We are interesting in predicting the % change in the US dollar in relation to the weekly changes in the world stock markets. Hence we collect weekly data for all of 2012. For each week we record the % change in the dollar, the % change in the US market, the % change in the British market, and the % change in the German market.

   This is a regression problem. We are interested in predicting the performance of the U.S. dollar which has a continuous outcome space. $n = 52$ since there are 52 weeks in a year and we have weekly data. $p = 3$ since we have performance data from three stock markets from which to compare (US, UK, and Germany).

2. (6 pts) The table below provides a training data set containing six observations, three predictors, and one qualitative response variable.

X_1	X_2	X_3	Y
0	3	0	Red
2	0	0	Red
0	1	3	Red
0	1	2	Green
-1	0	1	Green
1	1	1	Red

Suppose we wish to use this data set to make a prediction for Y when $X_1 = X_2 = X_3 = 0$ using K-nearest neighbors.

(a) (2 pts) Compute the Euclidean distance between each observation and the test point, $X_1 = X_2 = X_3 = 0$.
For this example, the Euclidean distance is $\sqrt{X_1^2 + X_2^2 + X_3^2}$. Let d be the Euclidean distance from $(0, 0, 0)$.

X_1	X_2	X_3	Y	d
0	3	0	Red	3
2	0	0	Red	2
0	1	3	Red	$\sqrt{10}$
0	1	2	Green	$\sqrt{5}$
-1	0	1	Green	$\sqrt{2}$
1	1	1	Red	$\sqrt{3}$

(b) (2 pts) What is our prediction with $K = 1$? Why?
Our prediction is Green because the closest point to $(0, 0, 0)$ is the 5th observation, which is green ($d = \sqrt{2}$).

(c) (2 pts) What is our prediction with $K = 3$? Why?
Now, our prediction is red because 2/3 of the closest points are red and only one is green. The distances to the three closest points are $d = \sqrt{2}, \sqrt{3}, 2$.

3. (12 pts) When the number of features p is large, there tends to be a deterioration in the performance of KNN and other local approaches that perform prediction using only observations that are near the test observation for which a prediction must be made. This phenomenon is known as the curse of dimensionality, and it ties into the fact that non-parametric approaches often perform poorly when p is large.

(a) (2pts) Suppose that we have a set of observations, each with measurements on $p = 1$ feature, X. We assume that X is uniformly (evenly) distributed on [0, 1]. Associated with each observation is a response value. Suppose that we wish to predict a test observation's response using only observations that are within 10% of the range of X closest to that test observation. For instance, in order to predict the response for a test observation with $X = 0.6$, we will use observations in the range [0.55, 0.65]. On average, what fraction of the available observations will we use to make the prediction?
On average, we would use $\frac{1}{10}$ of the observations to make a prediction. The PDF of a uniform distribution is a constant line, so integrating over $X\%$ of the distribution will return $\frac{X}{100}$.

(b) (2pts) Now suppose that we have a set of observations, each with measurements on $p = 2$ features, $X1$ and $X2$. We assume that predict a test observation's response using only observations that $(X1, X2)$ are uniformly distributed on [0, 1] × [0, 1]. We wish to are within 10% of the range of $X1$ and within 10% of the range of $X2$ closest to that test observation. For instance, in order to predict the response for a test observation with $X1 = 0.6$ and $X2 = 0.35$, we will use observations in the range [0.55, 0.65] for $X1$ and in the range [0.3, 0.4] for $X2$. On average, what fraction of the available observations will we use to make the prediction?
Since the observations are distributed uniformly over the unit square, the proportion of observations used is equivalent to the ratio of the area of the prediction "box" and the total area. Therefore, on average, $\frac{.01}{1} = \frac{1}{100}$ of the available observations would be used to make a prediction.

(c) (2pts) Now suppose that we have a set of observations on $p = 100$ features. Again the observations are uniformly distributed on each feature, and again each feature ranges in value from 0 to 1. We wish to predict a test observation's response using observations within the 10% of each feature's range that is closest to that test observation. What fraction of the available observations will we use to make the prediction?
Using the same logic as the two previous questions, we get $\frac{.1^{100}}{1} = 10^{-100}$.

(d) (3pts) Using your answers to parts (a)–(c), argue that a drawback of KNN when p is large is that there are very few training observations "near" any given test observation.
A drawback to KNN when p is large is that there are very few training observations near any given test observation. As we can see from parts (a)-(c), the number of observations that fall within a hypercube with a side length proportional to 10% of the input space decreases exponentially.

(e) (3pts) Now suppose that we wish to make a prediction for a test observation by creating a p-dimensional hypercube centered around the test observation that contains, on average, 10% of the training observations. For p =1, 2, and 100, what is the length of each side of the hypercube? Comment on your answer.

For $p = 1$ the hypercube would have a slide length of .1 per the first question. Denote this $\ell_1 = .1$. Then $\ell_2 = .1^{1/2} \approx 0.3162$ and $\ell_{100} = .1^{1/100} \approx .9772$. In general, $\ell_p = .1^{1/p}$. What these cases show is that to retain a 10% threshold of observations used in our prediction, we need to rely on points that are farther away from our test point as p gets large.

4. (6 pts) Supoose you trained a classifier for a spam detection system. The prediction result on the test set is summarized in the following table.

		Predicted class	
		Spam	not Spam
Actual class	Spam	8	2
	not Spam	16	974

Calculate

(a) (2 pts) Accuracy $\frac{8+974}{1000} = .982$

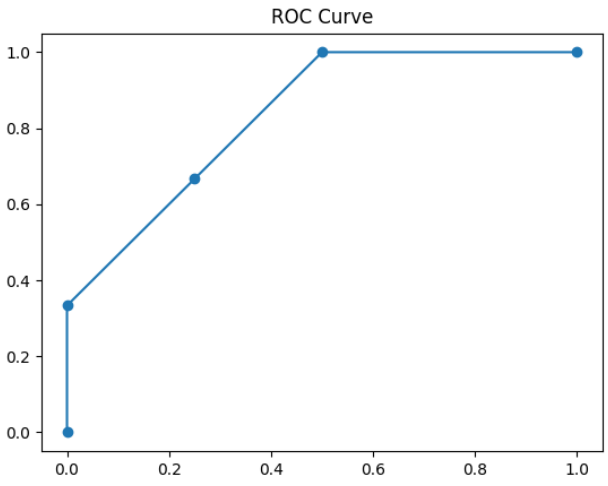
(b) (2 pts) Precision $\frac{8}{8+16} = .333$

(c) (2 pts) Recall $\frac{8}{8+2} = .8$

5. (9pts) Again, suppose you trained a classifier for a spam filter. The prediction result on the test set is summarized in the following table. Here, "+" represents spam, and "-" means not spam.

Confidence positive	Correct class
0.95	+
0.85	+
0.8	-
0.7	+
0.55	+
0.45	-
0.4	+
0.3	+
0.2	-
0.1	-

(a) (6pts) Draw a ROC curve based on the above table.



(b) (3pts) (Real-world open question) Suppose you want to choose a threshold parameter so that mails with confidence positives above the threshold can be classified as spam. Which value will you choose? Justify your answer based on the ROC curve.

Based on the thresholds used by the ROC curve, I would want to choose either .8, .45, or .2 as my threshold value. I am personally paranoid that I am going to miss an email and would rather have more spam come into my inbox as opposed to missing a potentially important email, so I would chose .8 as the threshold for myself. With .8 as my threshold, my false positive rate would be very low while still filtering our some spam emails.

6. (8 pts) In this problem, we will walk through a single step of the gradient descent algorithm for logistic regression. As a reminder,

$$f(x;\theta) = \sigma(\theta^\top x)$$

$$\text{Cross entropy loss } L(\hat{y}, y) = -[y \log \hat{y} + (1-y)\log(1-\hat{y})]$$

$$\text{The single update step } \theta^{t+1} = \theta^t - \eta \nabla_\theta L(f(x;\theta), y)$$

(a) (4 pts) Compute the first gradient $\nabla_\theta L(f(x;\theta), y)$.

$$\begin{aligned}
\nabla_\theta L(f(x;\theta), y) &= \nabla_\theta[-y\log(\sigma(\theta^\top x))) - (1-y)\log(1-\sigma(\theta^\top x))] \\
&= \frac{-y}{\sigma(\theta^\top x)} \cdot \sigma'(\theta^\top x) \cdot x - \frac{1-y}{1-\sigma(\theta^\top x)} \cdot -\sigma'(\theta^\top x) \cdot x \\
&= \frac{-y}{\sigma(\theta^\top x)} \cdot \sigma(\theta^\top x)(1-\sigma(\theta^\top x)) \cdot x + \frac{1-y}{1-\sigma(\theta^\top x)} \cdot \sigma(\theta^\top x)(1-\sigma(\theta^\top x)) \cdot x \\
&= -xy(1-\sigma(\theta^\top x)) + x(1-y)\sigma(\theta^\top x) \\
&= x(-y + y\sigma(\theta^\top x) + \sigma(\theta^\top x) - y\sigma(\theta^\top x)) \\
&= x(\sigma(\theta^\top x) - y)
\end{aligned}$$

(b) (4 pts) Now assume a two dimensional input. After including a bias parameter for the first dimension, we will have $\theta \in \mathbb{R}^3$.

$$\text{Initial parameters : } \theta^0 = [0,0,0]$$

$$\text{Learning rate } \eta = 0.1$$

$$\text{data example : } x = [1,3,2], y = 1$$

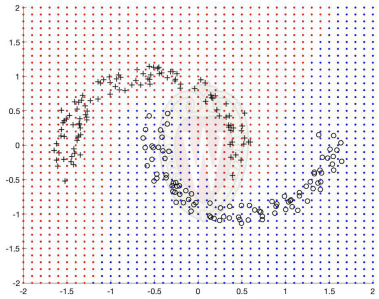
Compute the updated parameter vector θ^1 from the single update step.

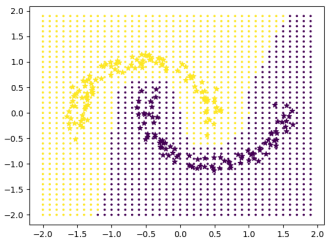
$$\begin{aligned}
\theta^1 &= \theta^0 - 0.1 \cdot [1,3,2](\sigma([0,0,0]^\top [1,3,2]) - 1) \\
&= \theta^0 - 0.1 \cdot [1,3,2](\sigma(0) - 1) \\
&= \theta^0 - 0.1 \cdot [1,3,2](\frac{1}{2} - 1) \\
&= [.05, .15, .1]
\end{aligned}$$

2 Programming (50 pts)

1. (10 pts) Use the whole D2z.txt as training set. Use Euclidean distance (i.e. $A = I$). Visualize the predictions of 1NN on a 2D grid $[-2 : 0.1 : 2]^2$. That is, you should produce test points whose first feature goes over $-2, -1.9, -1.8, \ldots, 1.9, 2$, so does the second feature independent of the first feature. You should overlay the training set in the plot, just make sure we can tell which points are training, which are grid.

   The expected figure looks like this.





Spam filter Now, we will use 'emails.csv' as our dataset. The description is as follows.

Features → | Label →

Email No.	the	to	ect	and	for	of	a	you	hou	in	...	connevey	jay	valued	lay	infrastructure	military	allowing	ff	dry	Prediction
Email 1	0	0	1	0	0	0	2	0	0	0	...	0	0	0	0	0	0	0	0	0	0
Email 2	8	13	24	6	6	2	102	1	27	18	...	0	0	0	0	0	0	0	1	0	0
Email 3	0	0	1	0	0	0	8	0	0	4	...	0	0	0	0	0	0	0	0	0	0
Email 4	0	5	22	0	5	1	51	2	10	1	...	0	0	0	0	0	0	0	0	0	0
Email 5	7	6	17	1	5	2	57	0	9	3	...	0	0	0	0	0	0	0	1	0	0

- Task: spam detection
- The number of rows: 5000
- The number of features: 3000 (Word frequency in each email)
- The label (y) column name: 'Predictor'
- For a single training/test set split, use Email 1-4000 as the training set, Email 4001-5000 as the test set.
- For 5-fold cross validation, split dataset in the following way.
  - Fold 1, test set: Email 1-1000, training set: the rest (Email 1001-5000)
  - Fold 2, test set: Email 1000-2000, training set: the rest
  - Fold 3, test set: Email 2000-3000, training set: the rest
  - Fold 4, test set: Email 3000-4000, training set: the rest
  - Fold 5, test set: Email 4000-5000, training set: the rest

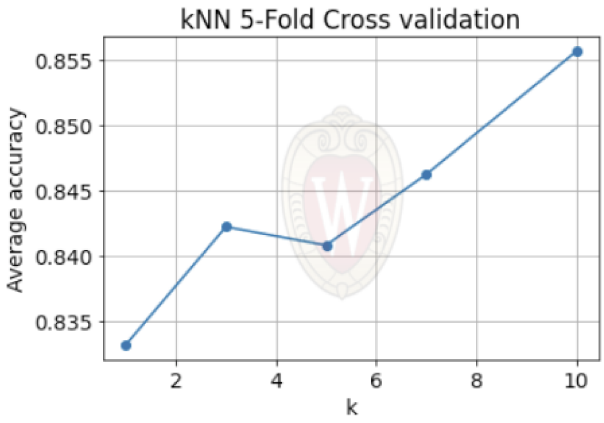
2. (8 pts) Implement 1NN, Run 5-fold cross validation. Report accuracy, precision, and recall in each fold.

Fold	Accuracy	Precision	Recall
1	0.825	0.6545	0.8175
2	0.853	0.6857	0.8664
3	0.862	0.7212	0.838
4	0.851	0.7164	0.8163
5	0.775	0.6057	0.7582

3. (12 pts) Implement logistic regression (from scratch). Use gradient descent (refer to question 6 from part 1) to find the optimal parameters. You may need to tune your learning rate to find a good optimum. Run 5-fold cross validation. Report accuracy, precision, and recall in each fold.

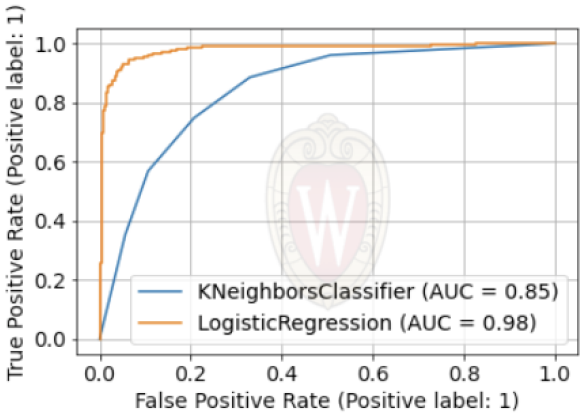
   Solution goes here.

4. (10 pts) Run 5-fold cross validation with kNN varying k (k=1, 3, 5, 7, 10). Plot the average accuracy versus k, and list the average accuracy of each case.
   Expected figure looks like this.



   Solution goes here.

5. (10 pts) Use a single training/test setting. Train kNN (k=5) and logistic regression on the training set, and draw ROC curves based on the test set.
   Expected figure looks like this. Note that the logistic regression results may differ.



   Solution goes here.