HOMEWORK 3

William Powell
9083433244

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 for this assignment can be found at https://github.com/wgraysonp/CS760/tree/main/HW_3.

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 since we are interested in predicting CEO salary which is best understood as a real-valued or continuous variable. Here $n = 500$, and $p = 3$ where the features are 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 (binary) classification problem. The variable representing whether or not a new product is a success that we are trying to predict is discrete. Here $n = 20$ and $p = 13$.

   (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 since the percentage change in the US dollar is a continuous variable. In this situation $n = 52$ (there are 52 weeks in a year) and $p = 3$.

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, $X_1 = X_2 = X_3 = 0$.
   The distances, ordered as in the table from top to bottom, are 3, 2, $\sqrt{10}$, $\sqrt{5}$, $\sqrt{2}$, and $\sqrt{3}$.

(b) (2 pts) What is our prediction with $K = 1$? Why?
The prediction is Green. The observation with smallest distance to the point $(0, 0, 0)$ is $(-1, 0, 1)$ which has corresponding response $Y =$ Green

(c) (2 pts) What is our prediction with $K = 3$? Why?
The closest three observations to $(0, 0, 0)$ in order are $(-1, 0, 1)$, $(1, 1, 1)$, and $(2, 0, 0)$. Among these, the most common response is Red. So our prediction when $K = 3$ is Red.

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 will use 10% of the observations as we will only consider observations in an interval of length 0.1 which is 10% of the entire range of X: $[0, 1]$.

(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?
Here we will only use $1/100$ of the total observations. We will consider only those observations lying in a square of area $0.1 \times 0.1 = 0.01$ centered at (X_1, X_2). This is only $1/100$ of the full range $[0, 1]^2$.

(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?
In this case we will only consider observations lying in a set with area $(0.1)^{100}$ which is only a fraction 10^{-100} of the full feature space. In general, the fraction we will use is 10^{-p}.

(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.
The previous arguments show that if X is a random vector of dimension p we are using for our predictions, then the probability that our training instances are close to X exhibits geometric decay with p. Specifically for our example, if Y is an i.i.d copy of X then $P(\|X - Y\| \leq \beta) = O(\beta^p)$ for $\beta \in (0, 1)$ and dimension p. Therefore, when p is large there are likely very few observations "near" a given test example X.

(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 what happens to the length of the sides as $\lim_{p\to\infty}$.
Suppose our hypercube has side length ℓ. Then we need $\ell^p = 0.1$ to ensure we use 10% of the training observations on average. Therefore, we need $\ell = (0.1)^{1/p}$. For $p = 1, 2$, and 100 this is to require $\ell = \frac{1}{10}, \frac{1}{\sqrt{10}}$, and $\frac{1}{10^{1/100}}$ respectively. In the limit we have $\lim_{p\to\infty} \ell = \lim_{p\to\infty}(0.1)^{1/p} = 1$.

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 Accuracy is the total correct responses divided by the total number of tests. Here that is accuracy $= \frac{8+974}{8+2+16+974} = 0.982$.

(b) (2 pts) Precision Precision is the total number of correctly predicted positives divided by the number of all predicted positives. Precision in this examples is $\frac{8}{8+16} = 0.34$

(c) (2 pts) Recall Recall is the total number of correctly predicted positives divided y the total number of actual positives. Here that is $\frac{8}{10} = 0.8$

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.

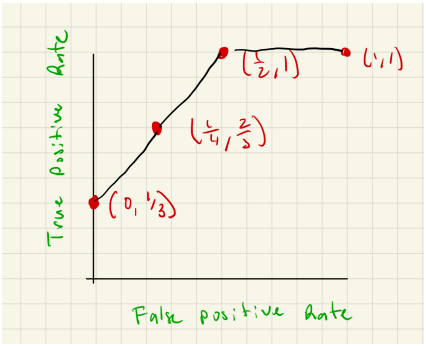

Figure 1: An ROC curve for the data given in problem 5

(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.

According to the table, we can take the threshold value to be 0.3. At this level the true positive rate is 1.

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

$$\hat{y} = f(x, \theta)$$

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

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

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

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

$$\begin{aligned}\nabla_\theta L(f(x;\theta),y) &= -[y\nabla_\theta \log \hat{y} + (1-y)\nabla_\theta \log(1-\hat{y})] \\ &= -x[\frac{y}{\hat{y}}\hat{y}(1-\hat{y}) - \frac{1-y}{1-\hat{y}}\hat{y}(1-\hat{y})] \\ &= -x\left[y(1-\hat{y}) - \hat{y}(1-y)\right] \\ &= x[\hat{y}-y]\end{aligned}$$

(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.

We can compute θ_1 as

$$\begin{aligned}\theta_1 &= \theta_0 - \eta\nabla_\theta L(f(x;\theta_0),y) && (1)\\ &= (0,0,0) - 0.1\cdot(1,2,3)\left[\sigma(\theta_0^T x) - 1\right] && (2)\\ &= (0,0,0) - 0.1\cdot(1,2,3)[0.5-1] && (3)\\ &= (0,0,0) + 0.1\cdot(0.5,1,1.5) && (4)\\ &= (0.05,0.1,0.15) && (5)\end{aligned}$$

where we used $\sigma(0) = 1/2$.

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.

   See figure 2 for a visualization of 1NN on the grid $[-2 : 0.1 : 2]^2$. The larger markers represent the training data.

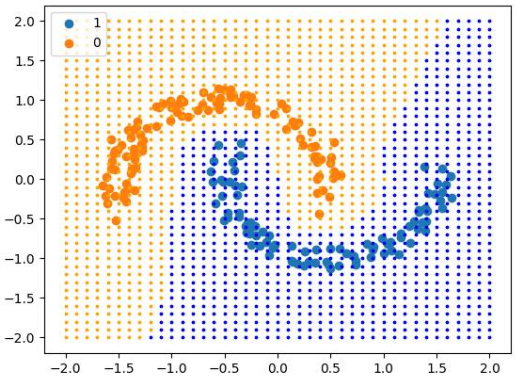

Figure 2: Visualization of 1NN for the data in D2z.txt

Spam filter Now, we will use 'emails.csv' as our dataset.

- 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.

   A table displaying the cross validation results of 1NN is below.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Accuracy	.825	.853	.862	.851	.775
Precision	.654	.686	.721	.716	.606
Recall	.816	.866	.838	.816	.758

Table 1: Results of Cross Validation for 1NN

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.

   A table displaying the cross validation results of logistic regression is below.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Accuracy	.919	.916	.896	.893	.885
Precision	.902	.907	.928	.891	.858
Recall	.803	.776	.686	.724	.748

Table 2: Results of Cross Validation for Logistic Regression

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.

   See figure 3 for a plot of the average accuracy as k increases.

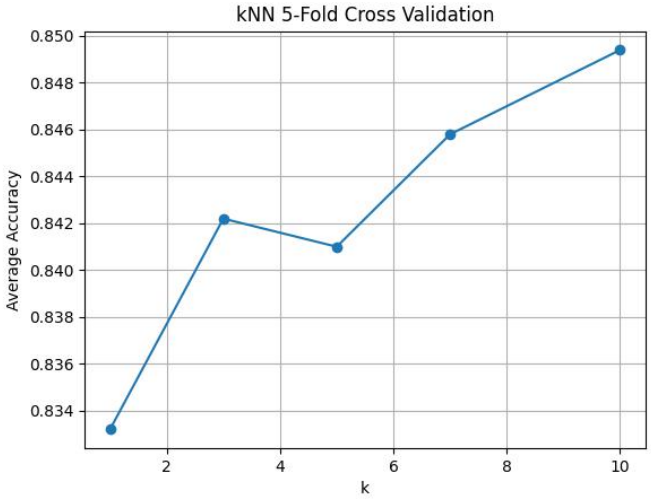

Figure 3: Average accuracy for kNN with various values of k

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.

   A comparison of ROC curves for K-nearest neighbors and logistic regression can be found in figure 4. For each method, we used fold 5 as the test set and the remaining emails as the training set.

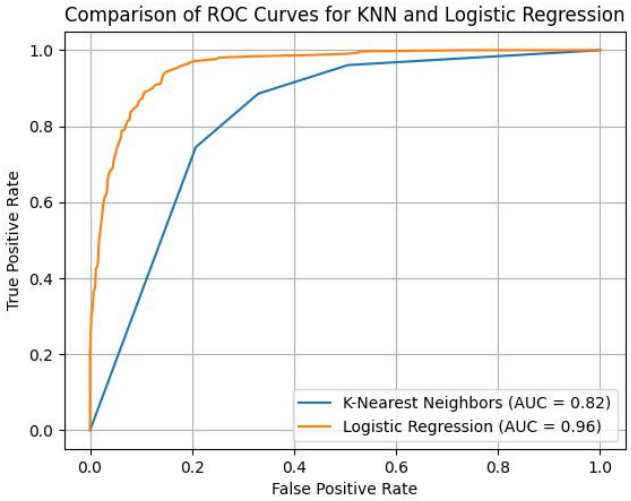

Figure 4: ROC curves for knn and logistic regression using fold 5 as the test set