HOMEWORK 3

>>JIAHUI ZHANG<<
>>908 449 6323<<

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.

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.

   Solution: Regression, since the value to be predicted (CEO salary) is continuous. $n = 500, p = 3$ for 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.

   Solution: Classification, since the value to be predicted (failure or success) is binary. $n = 20, p = 13$ for the 13 variables recorded (the response variable is excluded).

   (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.

   Solution: Regression, since the value to be predicted (% change in the US dollar) is continuous. $n = 52$ since we have 52 weeks for a year, and $p = 3$ for % change in the US market, British market and German market.

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, $X_1 = X_2 = X_3 = 0$.

   Solution:

X_1	X_2	X_3	Y	Euclidean distance to test point
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?

Solution: Green, because the 5th training data point is the closest to the test point, and it's green.

(c) (2 pts) What is our prediction with $K = 3$? Why?

Solution: Red, because the top-3 nearest points to test point are the 5th, the 6th and the 2nd points, and among them two are red and one is green. So the majority red is predicted for the test point.

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?

Solution: denote $P(x)$ as the fraction of available observations used to make the prediction for a sample x. Then

$$\begin{aligned} P(x) &= \int_{x-0.05}^{x+0.05} 1dt \\ &= \begin{cases} x + 0.05, & x < 0.05 \\ 0.1, & 0.05 \le x \le 0.95 \\ 1.05 - x, & x > 0.95 \end{cases} \end{aligned} \tag{1}$$

Then the average fraction is

$$\begin{aligned} \mathbb{E}(P(X)) &= \int_0^1 P(x)1dx \\ &=0.00375 + 0.09 + 0.00375 \\ &=0.0975 \end{aligned} \tag{2}$$

(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?

Solution: Since the two dimensions are independent, the average fraction is the square of the one-dimensional fraction, that is $0.0975^2 = 0.00950625$.

(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?

Solution: Similarly, now it becomes $0.0975^{100} = 7.95 \times 10^{-102}$.

(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.

Solution: If we define the concept of being “near” as that the value of each dimension should be within a certain range, then as the dimensionality goes large, the condition is harder to be obtained, since the final probability that data points being near will be the multiple of being near at each dimension. When p goes large, the space will become sparse and it’s harder to find a point that satisfies the “near” condition at all dimensions.

(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.

Solution: For $p = 1$, this is just a one-dimensional range with length of 10% of the total length, which is 0.1 (half length $r = 0.05$). For $p = 2$, we have $r = \sqrt{0.1} = 0.32$. For $p = 100$: $r = 0.1^{1/100} = 0.98$. This result means that for high-dimensional case, for the hypercube to reach a certain size, we need very very wide ranges on each dimension. Therefore, KNN for large dimensional case is not effective, or even not applicable.

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 Solution:

$$ACC = \frac{TP + TN}{TOTAL} = \frac{8 + 974}{1000} = 98.2\% \tag{3}$$

(b) (2 pts) Precision Solution:

$$PRE = \frac{TP}{TP + FP} = \frac{8}{8 + 16} = 33.3\% \tag{4}$$

(c) (2 pts) Recall Solution:

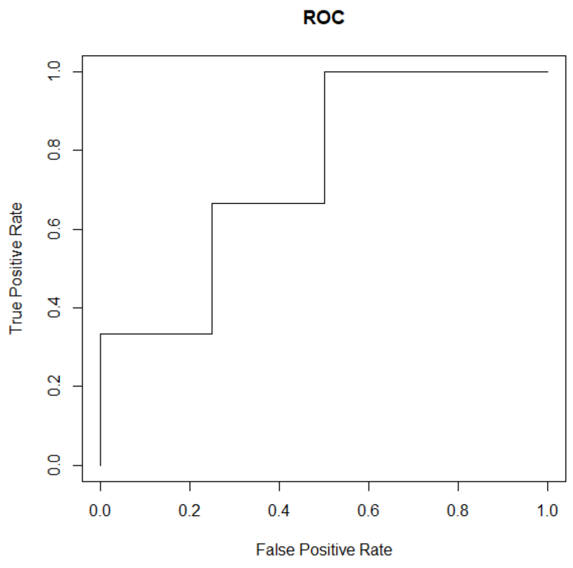
$$REC = \frac{TP}{TP + FN} = \frac{8}{8 + 2} = 80.0\% \tag{5}$$

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.

Solution:



(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.

Solution: I may choose 0.85 as the threshold, since it gives a zero false positive rate on the current data. That's because in real-world situations, failing to detect a spam results in far less bad consequences than mistakenly classifying one important email into spam. I may want to reduce the false positive rate as much as possible.

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 $\nabla_\theta L(f(x;\theta), y)$.

Solution:

$$\begin{aligned} L(f(x;\theta), y) &= -\left[y \log \sigma(\theta^T x) + (1-y)\log(1-\sigma(\theta^T x))\right] \\ \nabla_\theta L(f(x;\theta), y) &= \frac{dL(f(x;\theta), y)}{df(x;\theta)} \frac{df(x;\theta)}{d\theta} \\ &= -\left[\frac{y}{f(x;\theta)} - \frac{1-y}{1-f(x;\theta)}\right] \sigma(\theta^T x)(1-\sigma(\theta^T x))x \\ &= \left[\sigma(\theta^T x) - y\right] x \end{aligned} \quad (6)$$

(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.

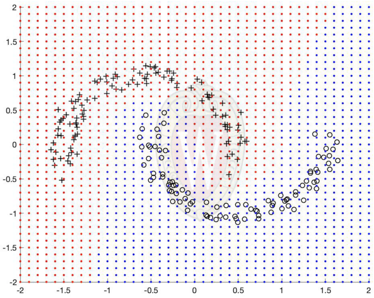
Solution:

$$\begin{aligned} \theta^1 =& \theta^0 - \eta \nabla L(f(x;\theta), y) \\ =& [0, 0, 0]^T + 0.1 \times (1 - \sigma(0))[1, 3, 2]^T \\ =& 0.1 \times 0.5 \times [1, 3, 2]^T \\ =& [0.05, 0.15, 0.10]^T \end{aligned} \quad (7)$$

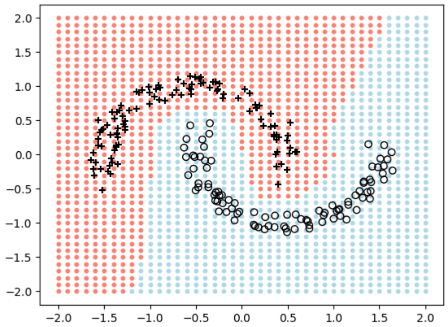
2 Programming (50 pts)

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.

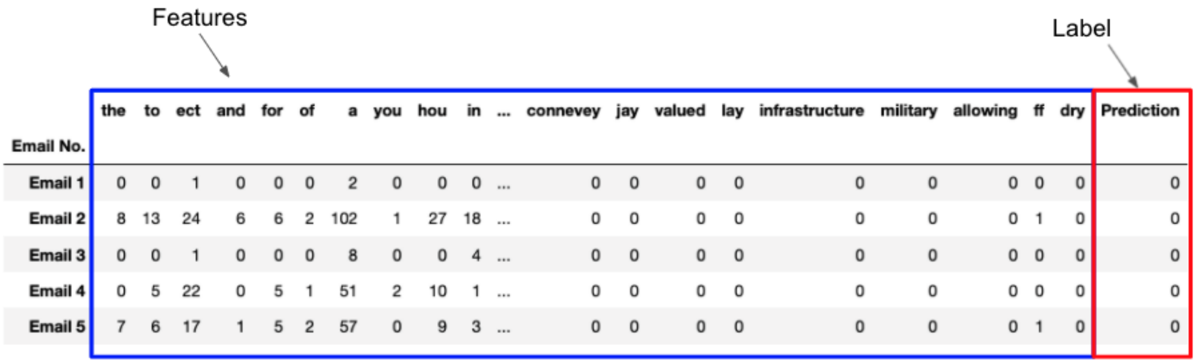
   The expected figure looks like this.



Solution: Here is my figure:



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.

   Solution:

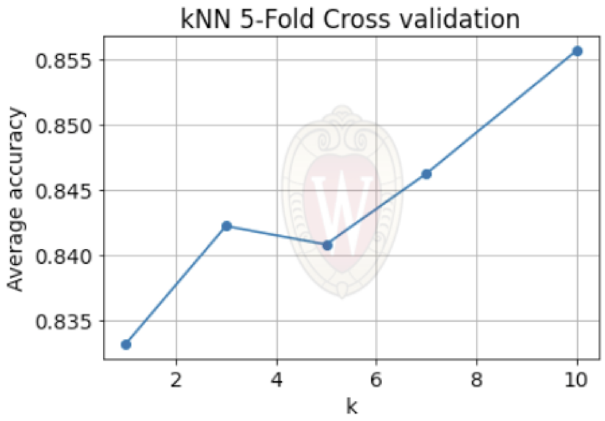
Fold	Accuracy	Precision	Recall
Test 1-1000	83.50%	65.62%	88.42%
Test 1001-2000	86.80%	69.65%	92.78%
Test 2001-3000	87.00%	72.38%	87.68%
Test 3001-4000	85.40%	71.14%	84.69%
Test 4001-5000	81.90%	65.43%	86.60%

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: I used Gradient Descent with learning rate $\eta = 0.1$. The stopping criteria is $||\theta^{(t)} - \theta^{(t-1)}|| \leq 0.1$. For prediction, I classify $\hat{y} \geq 0.5$ as positive, and $\hat{y} < 0.5$ as negative.

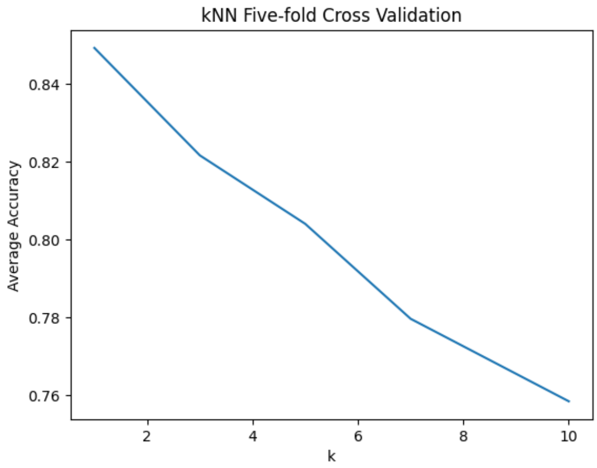
Fold	Accuracy	Precision	Recall
Test 1-1000	92.60%	82.26%	94.39%
Test 1001-2000	92.30%	78.90%	98.56%
Test 2001-3000	90.20%	77.03%	93.31%
Test 3001-4000	91.10%	80.60%	91.84%
Test 4001-5000	89.90%	76.76%	96.08%

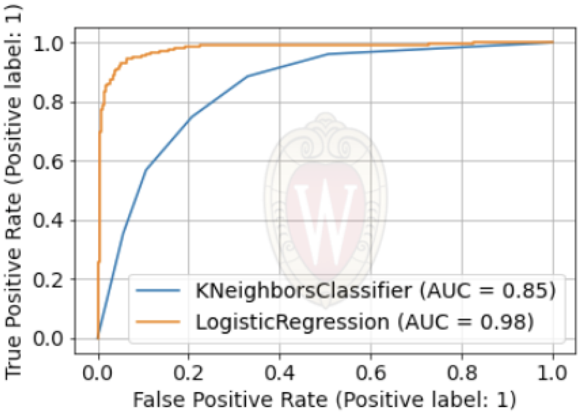
4. (10 pts) Run 5-fold cross validation with kNN varying k (k=1, 3, 5, 7, 10). Plot the average accuracy versus k, and list the average accuracy of each case.
   Expected figure looks like this.



   Solution:

K	Average Accuracy
1	84.92%
3	82.16%
5	80.40%
7	77.96%
10	75.84%

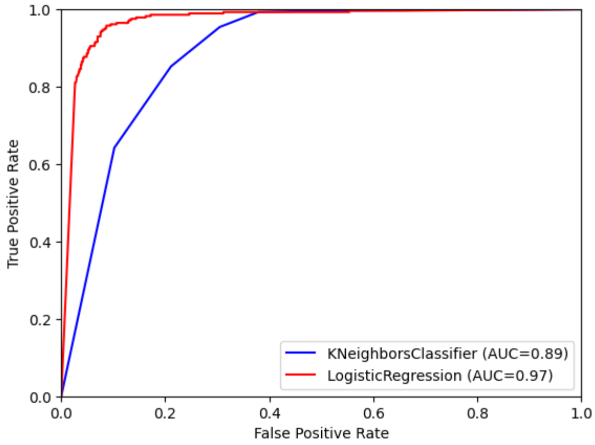




5. (10 pts) Use a single training/test setting. Train kNN (k=5) and logistic regression on the training set, and draw ROC curves based on the test set.
Expected figure looks like this.

Note that the logistic regression results may differ.

Solution: I used the first training/test splitting.


True Positive Rate
False Positive Rate
KNeighborsClassifier (AUC=0.89)
LogisticRegression (AUC=0.97)