HOMEWORK 3

Yichen Lin
908 531 9599
Link: https://github.com/whocansavemyhair/CS-760.git

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.
   Salary is a continuous value, so this is a regression problem.
   We record profit, number of employees and industry to predict the CEO salary. Therefore, number of features $p = 4$.
   We use data of 500 firms, so number of data points $n = 500$.

   (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.
   We want to know whether the product will be successful, so this is a classification problem.
   We recorded price, marketing budget, competition price and ten other variables, so the number of features is $p = 3 + 10 = 13$.
   We collect data from 20 similar products, so the number of data points is $n = 20$.

   (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.
   We want to predict the percentage of change in the US dollar in relation to the stock market, so this problem is a regression problem.
   We use data from the US market, the British market and the German market, so the number of features is $p = 3$.
   we collect weekly data for all of 2012, so the number of data points is $n = 52 \times 5 = 260$

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, $X_1 = X_2 = X_3 = 0$.

ID	X_1	X_2	X_3	Y	dis
1	0	3	0	Red	3
2	2	0	0	Red	2
3	0	1	3	Red	$\sqrt{10}$
4	0	1	2	Green	$\sqrt{5}$
5	-1	0	1	Green	$\sqrt{2}$
6	1	1	1	Red	$\sqrt{3}$

(b) (2 pts) What is our prediction with $K = 1$? Why?

The label of the closest point is Green, so the prediction will be Green.

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

The labels of the three nearest points are Green, Red, Red. So the prediction will be 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?

For one evenly distributed feature only, the fraction is 10%.

(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?

For two evenly distributed features, the fraction is $10\%^2 = 1\%$.

(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?

For one hundred evenly distributed features, the fraction is $10\%^{100}$.

(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.

For one specific data point, only 10% of data from each dimension can be used to make prediction. And data points are evenly distributed in each dimension, where we can make an assumption that these data points are i.i.d.. So the the formula is as follows:
suppose test observation is $k5$,
so $P(X_1, X_2, ...X_n in [k - 0.1, k + 0.1])$
$= P(X_1 in [k - 0.1, k + 0.1])P(X_2 in [k - 0.1, k + 0.1])...P(P(X_n in [k - 0.1, k + 0.1]))$
$= 10\%^p$

So as p increase, the number of available data points is decreasing rapidly.

(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 $p = 1$, length $s = V_1 = 0.1$.
For $p = 2$, length $s = V_2 = \sqrt[2]{0.1}$.
For $p = 100$, length $s = V_{100} = \sqrt[100]{0.1}$

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 $Accuracy = \frac{8+974}{8+2+16+974} = 0.982$

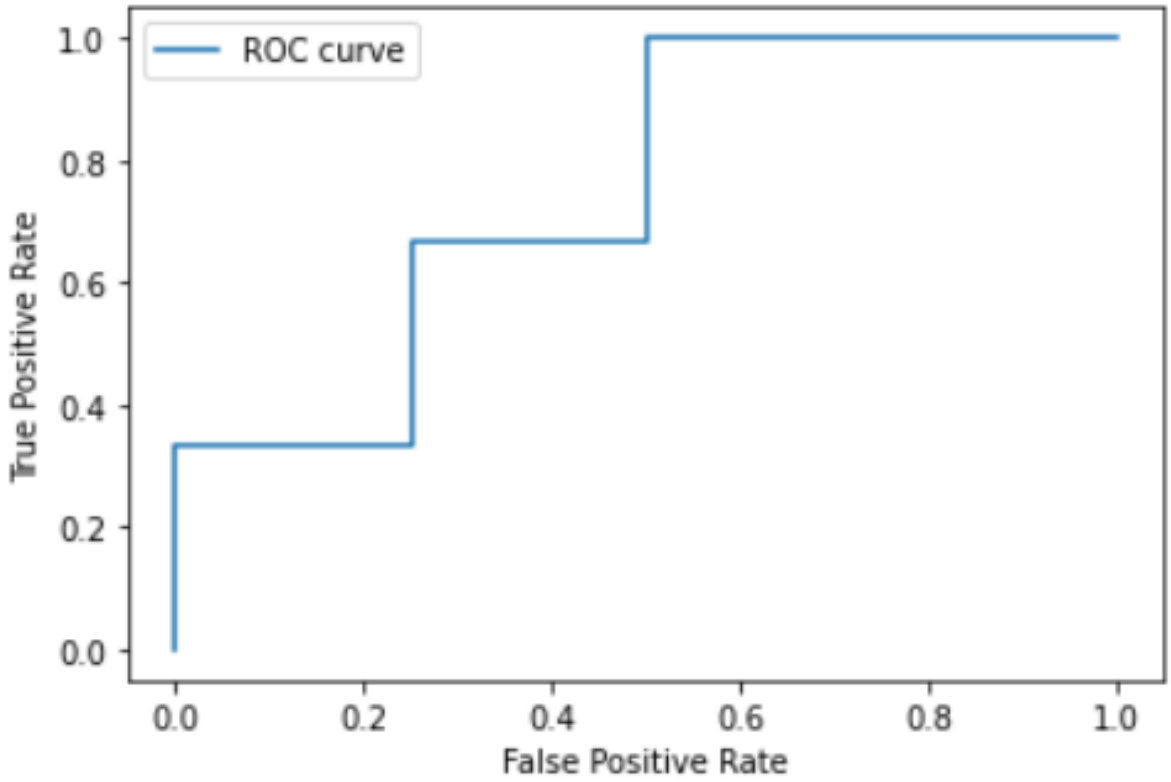
(b) (2 pts) Precision $Precision = \frac{8}{8+16} \approx 0.333$

(c) (2 pts) Recall $Recall = \frac{8}{8+2} = 0.8$

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.



(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.

I will choose 0.85 as the threshold parameter. The reason the cost of is classify an email to be spam by mistake is very high, which may cause great loss to users. So we should try our best to prevent that from happening.

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

$\nabla_\theta L(\hat{y}, y) = -\frac{y}{\hat{y}} \nabla_\theta \hat{y} + \frac{1-y}{1-\hat{y}} \nabla_\theta \hat{y}$

$\nabla_\theta \hat{y} = \nabla_\theta \sigma(\theta^\top x) = \sigma(\theta^\top x)(1 - \sigma(\theta^\top x))x$

$\nabla_\theta L(f(x;\theta), y) = \left[-y + \sigma(\theta^\top x)y + \sigma(\theta^\top x) - y\sigma(\theta^\top x)\right] x = (\sigma(\theta^\top x) - y)x$

(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.

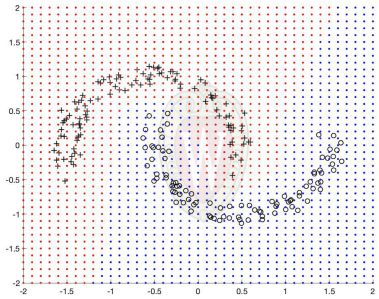
From (a), we have $\nabla_\theta L(f(x;\theta), y) = (\hat{y} - y)x$

so $\theta^{t+1} = \theta^t - \eta \nabla_\theta L(f(x;\theta), y) = [0,0,0] - 0.1([0.5, 0.5, 0.5] - [1,1,1]) * [1,3,2] = [0.05, 0.15, 0.1]$

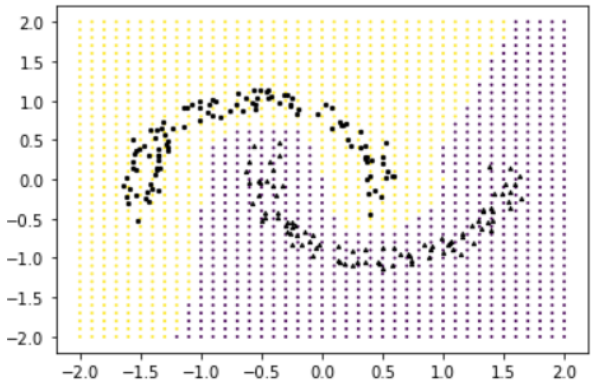
2 Programming (50 pts)

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.



The result is as follows:



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.

   The result is as follows:

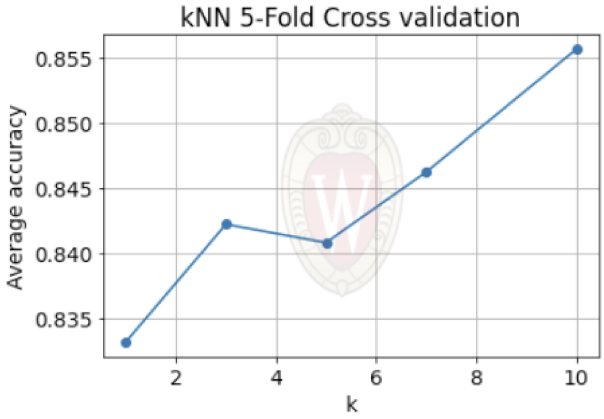
```
Fold 1.0
Accuracy 0.825
Precision 0.6544943820224719
Recall 0.8175438596491228
Fold 2.0
Accuracy 0.853
Precision 0.6857142857142857
Recall 0.8664259927797834
Fold 3.0
Accuracy 0.862
Precision 0.7212121212121212
Recall 0.8380281690140845
Fold 4.0
Accuracy 0.851
Precision 0.7164179104477612
Recall 0.8163265306122449
Fold 5.0
Accuracy 0.775
Precision 0.6057441253263708
Recall 0.7581699346405228
```

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.

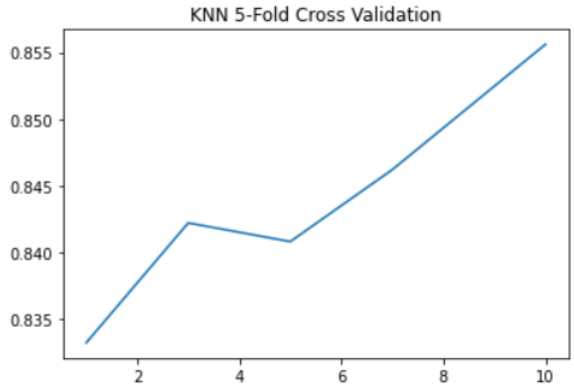
   The result is as follows:

```
Fold 1.0
Accuracy 0.878
Precision 0.7127937336814621
Recall 0.9578947368421052
Fold 2.0
Accuracy 0.901
Precision 0.7680722891566265
Recall 0.9205776173285198
Fold 3.0
Accuracy 0.771
Precision 0.5538160469667319
Recall 0.9964788732394366
Fold 4.0
Accuracy 0.877
Precision 0.7449856733524355
Recall 0.8843537414965986
Fold 5.0
Accuracy 0.821
Precision 0.6395604395604395
Recall 0.9509803921568627
```

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.

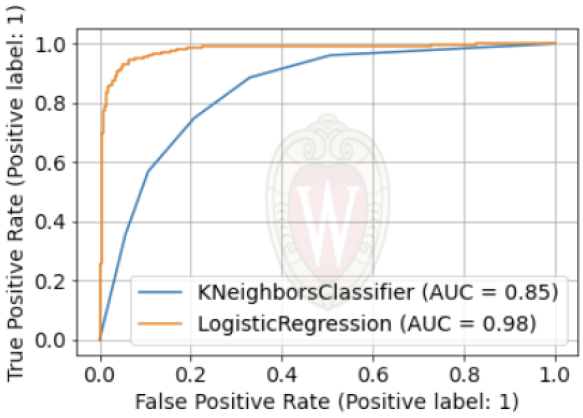


The result is as follows:

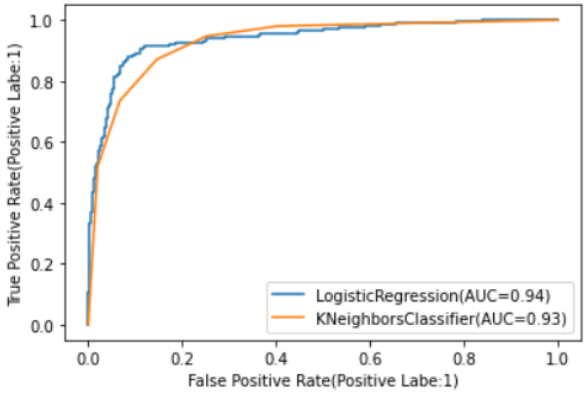



```
1
0.8332
3
0.8422000000000001
5
0.8408
7
0.8462
10
0.8556000000000001
```

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.



The result is as follows:


True Positive Rate(Positive Labe:1)
False Positive Rate(Positive Labe:1)
LogisticRegression(AUC=0.94)
KNeighborsClassifier(AUC=0.93)
0.0
0.2
0.4
0.6
0.8
1.0