HOMEWORK 3

Matt Myers
908-464-4252
GitHub

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.

   This is a prime example of a regression problem. There is 500 data points ($n = 500$) with three features each ($p = 3$). These 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 the price charged for the product, marketing budget, competition price, and ten other variables. These add up to be 13 total features

   This is a classification problem. The 20 similar products make up our 20 data points ($n = 20$). There are thirteen features as described there is success or failure, price change, marketing budget, competition price, and then ten other variables ($p = 13$).

   (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. The number of data points is equal to the number of weeks in 2012 which was 52 ($n = 52$). There are three features that go with this, the % change in the US market, the % change in the British market, and the % change in the German Market ($p = 3$).

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, $X_1 = X_2 = X_3 = 0$.

Note that Euclidean distance is calculated with the formula $d = \sqrt{X_1^2 + X_2^2 + X_3^2}$. Because $X_1 = X_2 = X_3 = 0$, the starting Euclidean distance $d = (0, 0, 0)$.

$d_1 = \sqrt{0^2 + 3^2 + 0^2} = \sqrt{9} = 3$ $d_2 = \sqrt{2^2 + 0^2 + 0^2} = \sqrt{4} = 2$
$d_3 = \sqrt{0^2 + 1^2 + 3^2} = \sqrt{10} \approx 3.162$ $d_4 = \sqrt{0^2 + 1^2 + 2^2} = \sqrt{5} \approx 2.236$
$d_5 = \sqrt{-1^2 + 0^2 + 1^2} = \sqrt{2} \approx 1.414$ $d_6 = \sqrt{1^2 + 1^2 + 1^2} = \sqrt{3} \approx 1.732$

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?

For $K = 1$ the best prediction is $Y =$ Green. This is because the smallest distance is in observation 5, where $d = \sqrt{2} \approx 1.414$. This means that the nearest neighbor is the fifth observation which has label $Y =$ Green.

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

For $K = 3$ we take the three nearest neighbors which leads us to the prediction $Y =$ Red. The three nearest neighbors are observations 5, 6, and 2 in that order. The most common label amongst those three results is $Y =$ Red. That is how the prediction is found.

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 a given test observation with value X, the range within 10% would have to [X - 0.05, X + 0.05]. Since X is uniformly distributed on [0, 1] all intervals are equally likely meaning on average 10% of the observations will be used to make predictions.

(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 the range of X_1 that is within 10% of 0.6 is [0.55, 0.65], and the range of X_2 that is within 10% of 0.35 is [0.3, 0.4]. Since X_1 and X_2 are independent and uniformly distributed on [0, 1], the fraction of observations that fall within both ranges is equal to $\frac{0.1 \cdot 0.1}{1} = 0.01$. This means that on average we use 1% of the available observations 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 similar logic as the previous question, The range of each feature that is within 10% of the test observation is [0.45, 0.55]. Since each feature is independent and uniformly distributed on [0, 1], the fraction of observations that fall within this range for each feature is equal to the product of all their ranges. This means that on average we will use $0.1^p = 0.1^{100} = 1 \cdot 10^{-100}$ of the available observation to make the prediction.

(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.

In parts (a)-(c) we demonstrate that as p increases, the fraction of observations that are "near" the test observation decreases. This means that there are very few training observations that are helpful for creating predictions for any test observation. This causes KNN to not work very well as p becomes very large.

(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 a hypercube with p-dimensions centered around the test observation we can create a fairly simple equation for calculating the length of each side of the hypercube. This equation would be $l = 0.1^{\frac{1}{p}}$. This could be seen with part (c), therefore for the calculations for each p, we have the following:

$l_{p=1} = 0.1^1 = 0.1$
$l_{p=2} = 0.1^{\frac{1}{2}} = \sqrt{0.1} = 0.316$
$l_{p=100} = 0.1^{\frac{1}{100}} = 0.9772$

These results show that to stay within a 10% threshold of the observations used in the prediction, we have to keep incorporating points further and further away.

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
The accuracy has to do with how many were predicted that were correct this can be found by the total correct divided by the total. In this case that would be $\frac{8+974}{1000} = \frac{982}{1000} = 0.982 = 98.2\%$.

(b) (2 pts) Precision
The precision is based on how precisely it predicted the spam out of the emails. In this example it

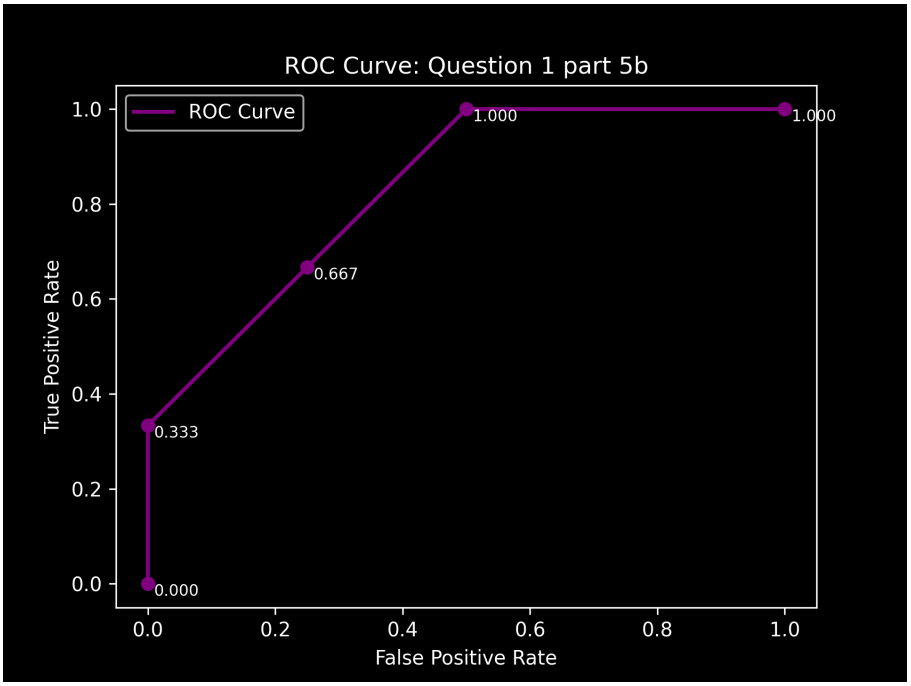
found 8 spam that were actually spam but improperly flagged 16 emails as spam that weren't. This leads to the precision being $\frac{8}{8+16} = \frac{8}{24} = \frac{1}{3} = 0.3\bar{3} = 33.3\bar{3}\%$.

(c) (2 pts) Recall
Recall has to do with the actual spam that was predicted to be spam versus the spam that was not found to be spam. For this example there was ten total spam emails. Eight were correctly identified and two snuck through. This leads are recall to be $\frac{8}{8+2} = \frac{8}{10} = \frac{4}{5} = 0.8 = 80\%$.

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.

Looking at the chart I believe that it would depend on what the user values more. The only way to get zero false positives would be to also get zero true positives at which point the spam is not even being filtered. I believe a good sweet spot would either be 0.45 or 0.2 depending on how aggressive you would want. If you value having no spam then at 0.45 false positive rate 100% accurate filtering is achieved. However for the sake of not missing possible important emails I feel that I would go with the 0.2 false positive rate. Only getting 1 out of every 5 false positives while getting nearly 50% of all

the spam emails. I think this question does boil down to preference however.

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

$$\begin{aligned}
\nabla_\theta L(f(x;\theta), y) &= \nabla_\theta[-y\log(\sigma(\theta^T x)) - (1-y)\log(1-\sigma(\theta^T x))] \\
&= \frac{-y}{\sigma(\theta^T x)} \cdot \sigma'(\theta^T x) \cdot x - \frac{1-y}{1-\sigma(\theta^T x)} \cdot \sigma'(\theta^T x) \cdot x \\
&= -y \cdot (1-\sigma(\theta^T x)) \cdot x - 1 + y \cdot \sigma(\theta^T x) \cdot x \\
&= x(\sigma(\theta^T x) - y)
\end{aligned}$$

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

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

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

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

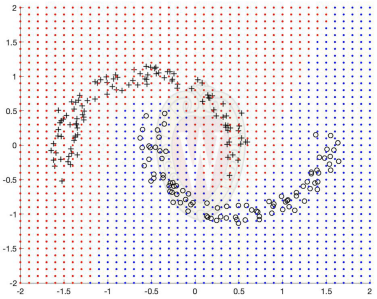
Compute the updated parameter vector θ^1 from the single update step.

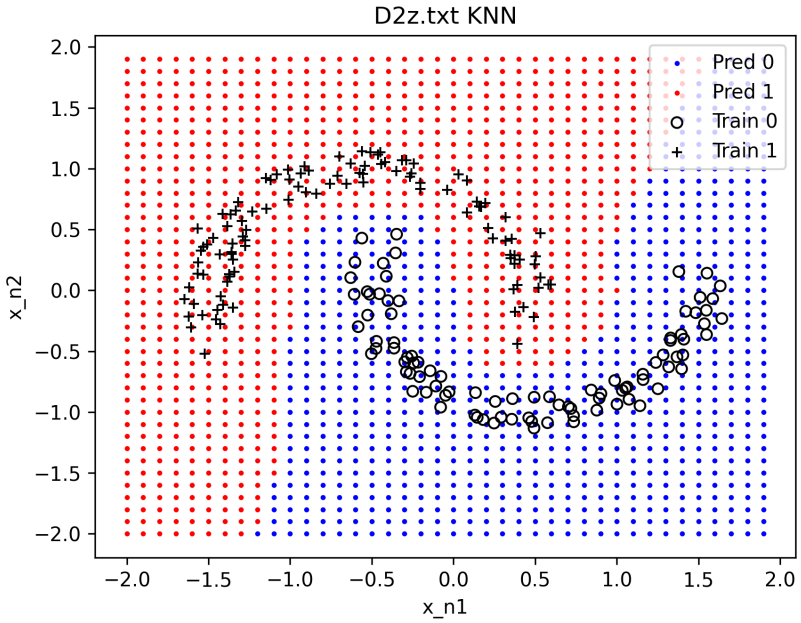
$$\begin{aligned}
\theta^1 &= \theta^0 - \eta \cdot x(\sigma((\theta^0)^T \cdot x) - y) \\
&= \theta^0 - 0.1 \cdot [1,3,2](\sigma([0,0,0]^T[1,3,2]) - 1) \\
&= \theta^0 - 0.1 \cdot [1,3,2](\sigma(0) - 1) \\
&= \theta^0 - 0.1 \cdot [1,3,2](\frac{1}{2}) - 1) \\
&= \theta^0 - 0.1 \cdot [1,3,2](-\frac{1}{2}) \\
&= [0,0,0] + \frac{1}{20} \cdot [1,3,2] \\
&= [\frac{1}{20}, \frac{3}{20}, \frac{1}{10}] \\
&= [0.05, 0.15, 0.10]
\end{aligned}$$

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.

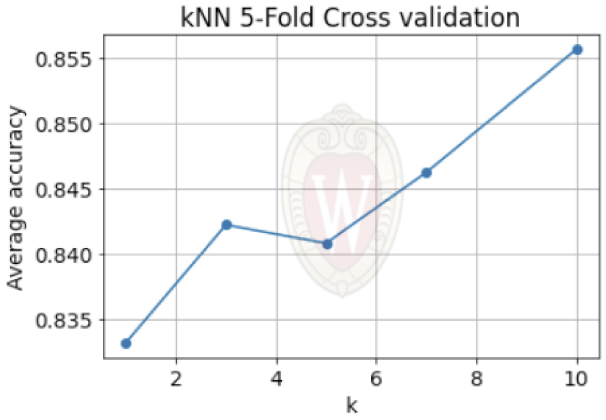
Fold	Accuracy	Precision	Recall
1	0.825	0.65449	0.81754
2	0.853	0.68571	0.86643
3	0.862	0.72121	0.83803
4	0.851	0.71642	0.81633
5	0.775	0.60574	0.75817

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 Learning rate = 0.1, and the number of iterations = 1000.

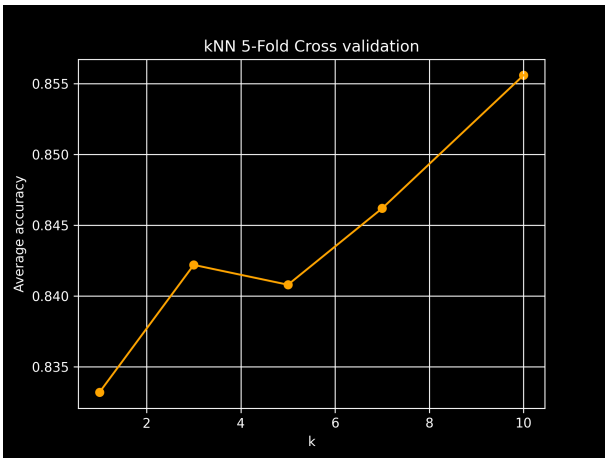
Fold	Accuracy	Precision	Recall
1	0.909	0.89113	0.77544
2	0.854	0.90184	0.53069
3	0.889	0.82156	0.77817
4	0.843	0.66348	0.94558
5	0.849	0.78598	0.69608

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.

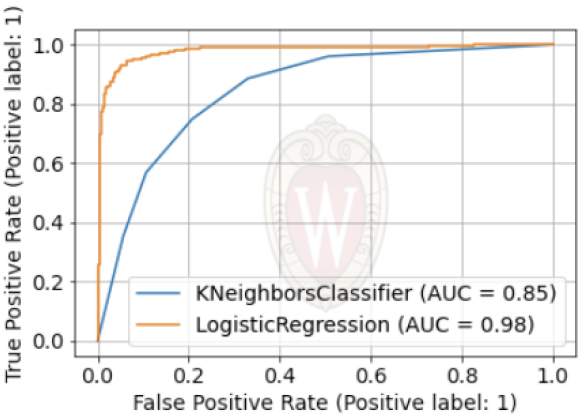


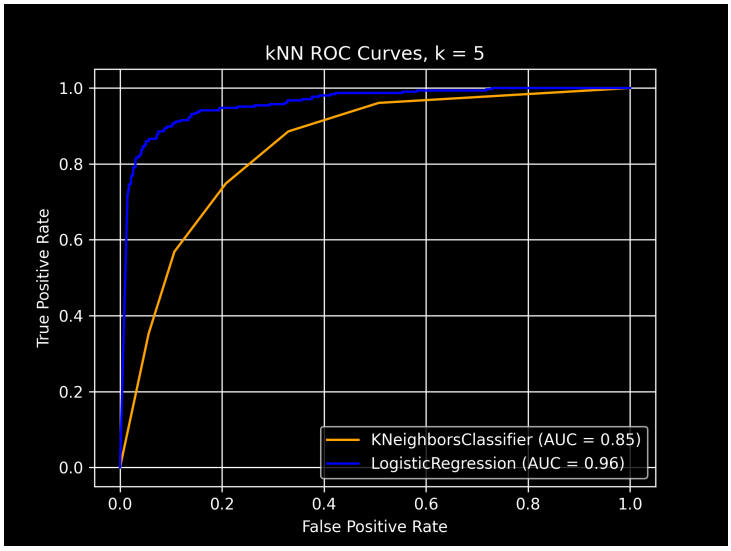
k	Average Accuracy
1	0.8332
3	0.8422
5	0.8408
7	0.8462
10	0.8556

The above chart houses the values for the figure below.



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.





Learning rate = 0.1, Number of iterations = 10,000

https://github.com/myersmt/Hw002-Comp760