

HOMEWORK 3

>>NAME HERE<<

>>ID HERE<<

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.
Regression Problem,
 n : 500,
 p : 3 - profit, number of employees, 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.
Classification Problem,
 n : 20,
 p : 13 - price charged for the product, marketing budget, competition price and ten 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.
Regression Problem,
 n : 52,
 p : 3 - % change in the US market, the % change in the British market, and the % change in the 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, $X_1 = X_2 = X_3 = 0$.

Observation	Label	Euclidean Distance
1	Red	3
2	Red	2
3	Red	$\sqrt{10}$
4	Green	$\sqrt{5}$
5	Green	$\sqrt{2}$
6	Red	$\sqrt{3}$

- (b) (2 pts) What is our prediction with $K = 1$? Why?
 Green, because the closest 1 ($K=1$) observation is observation 5 with distance $\sqrt{2}$ and label Green, therefore the prediction is Green.
- (c) (2 pts) What is our prediction with $K = 3$? Why?
 Red, because the closest 3 ($K=3$) observations are observations 5, 6 and 2 with distances $\sqrt{2}$, $\sqrt{3}$, 2 and labels Green, Red, Red respectively. Since the count of Red labels in the closest 3 observations is greater than the count of Green labels ($2 > 1$), therefore the prediction 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?
 When $0.05 \leq x \leq 0.95$,
 the interval of observations is $[x - 0.05, x + 0.05]$, which is 10% of the observation set.
 When $x < 0.05$,
 the interval of observations is $[0, x + 0.05]$, which is $(100x + 5)\%$ of the observation set.
 When $x > 0.95$,
 the interval of observations is $[x - 0.05, 1.0]$, which is $(105 - 100x)\%$ of the observation set.
 Then the average fraction of the available observations we will use to make the prediction is given by:
 $\int_{0.05}^{0.95} 10x \, dx + \int_0^{0.05} (100x + 5) \, dx + \int_{0.95}^1 (105 - 100x) \, dx = 9 + 0.375 + 0.375 = 9.75\%$.
 Therefore, on average we will use 9.75% of the set of observations to make the prediction.
- (b) (2pts) Now suppose that we have a set of observations, each with measurements on $p = 2$ features, X_1 and X_2 . We assume that predict a test observation's response using only observations that (X_1, X_2) are uniformly distributed on $[0, 1] \times [0, 1]$. We wish to are within 10% of the range of X_1 and within 10% of the range of X_2 closest to that test observation. For instance, in order to predict the response for a test observation with $X_1 = 0.6$ and $X_2 = 0.35$, we will use observations in the range $[0.55, 0.65]$ for X_1 and in the range $[0.3, 0.4]$ for X_2 . On average, what fraction of the available observations will we use to make the prediction?
 Assuming X_1 and X_2 are independent, then the fraction of available observations will be:
 $9.75\% * 9.75\% = 0.950625\%$
- (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?
 $(9.75\%)^{100} = 7.951729 * 10^{-100}\%$

- (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.

In KNN, we find the nearest K nearest neighbors to make the prediction. From parts (a)–(c), we see that the fraction of the available observations we can use to make the prediction is $(9.75\%)^p$. As $p \rightarrow \infty$, the available observations can be given by:

$$\lim_{p \rightarrow \infty} (9.75\%)^p = 0.$$

Hence, there are fewer training observations “near” any given test observation as p becomes larger.

- (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 $\lim_{p \rightarrow \infty}$.

A p -dimensional hypercube centered around the test observation that contains on average 10% of the training observations is equivalent to our hypercube having a volume of 0.1 . Since the side length l of

a p -dimensional hypercube is given by $l = V^{\frac{1}{p}}$ for volume V , then:

For $p = 1$, $l = 0.1$, For $p = 2$, $l = 0.1^{1/2}$, For $p = 100$, $l = 0.1^{1/100}$. As $\lim_{p \rightarrow \infty}$, the length of each side goes to 1.

4. (6 pts) Suppose 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 $\frac{8 + 974}{8 + 16 + 2 + 974} = 0.982$

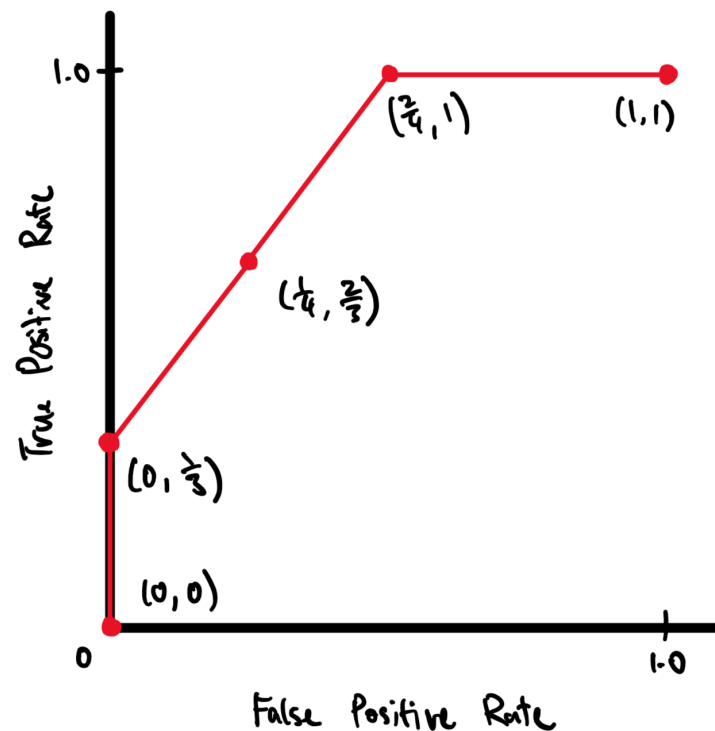
(b) (2 pts) Precision $\frac{8}{8 + 16} = \frac{1}{3}$

(c) (2 pts) 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.

Since we do not want to miss any important emails, we have to ensure that all mail that is classified as spam are actually spam. We can take the threshold of confidence positive ≥ 0.85 to ensure all mails are spam. Based on the ROC curve, we can see that the threshold of confidence positive ≥ 0.85 gives us the greatest true positive rate while maintaining a false positive rate of 0.

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

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

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

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

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

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

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

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

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

Learning rate $\eta = 0.1$

data example : $x = [1, 3, 2], y = 1$

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

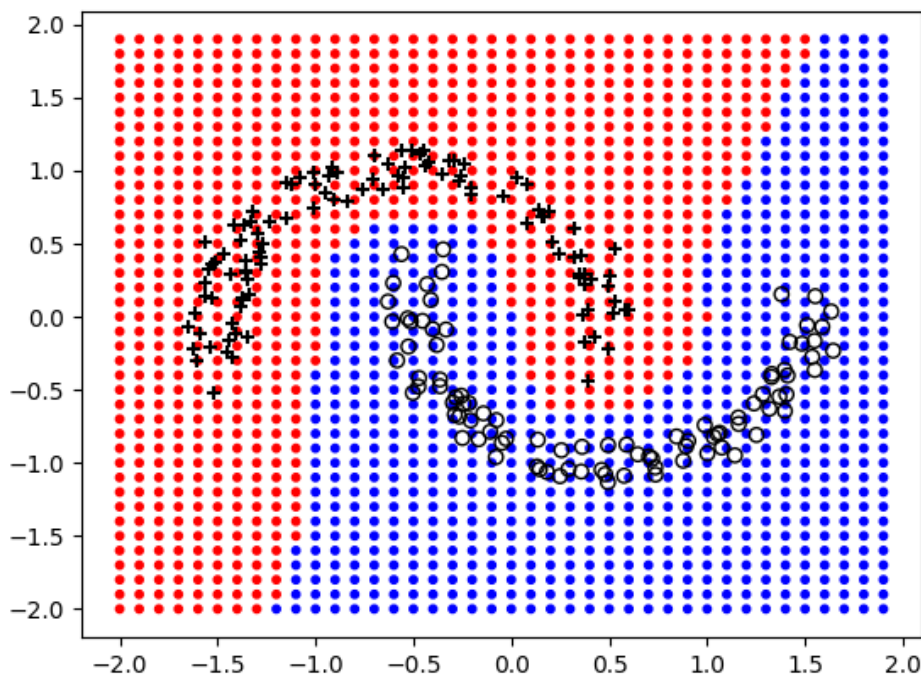
$$\begin{aligned}
 \theta^{t+1} &= \theta^t - \eta \nabla_{\theta} L(f(x; \theta), y) \\
 &= \theta^t - \eta x (\sigma(\theta^T x) - y) \\
 &= [0, 0, 0] - 0.1 [1, 3, 2] (\sigma([0, 0, 0]^T [1, 3, 2]) - 1) \\
 &= [0, 0, 0] - 0.1 [1, 3, 2] (-0.5, -0.5, -0.5) \\
 &= [0, 0, 0] - 0.1 [-0.5, -1.5, -1.0] \\
 &= [0, 0, 0] - [-0.05, -0.15, -0.10] \\
 &= [0.05, 0.15, 0.10]
 \end{aligned}$$

2 Programming (50 pts)

<https://github.com/poonlucas/CS760/tree/main/HW3>

- (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 $-2, -1.9, -1.8, \dots, 1.9, 2$, 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.

My Figure:



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

- 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	0.6544943820224719	0.8175438596491228
2	0.853	0.6857142857142857	0.8664259927797834
3	0.862	0.7212121212121212	0.8380281690140845
4	0.851	0.7164179104477612	0.8163265306122449
5	0.775	0.6057441253263708	0.7581699346405228

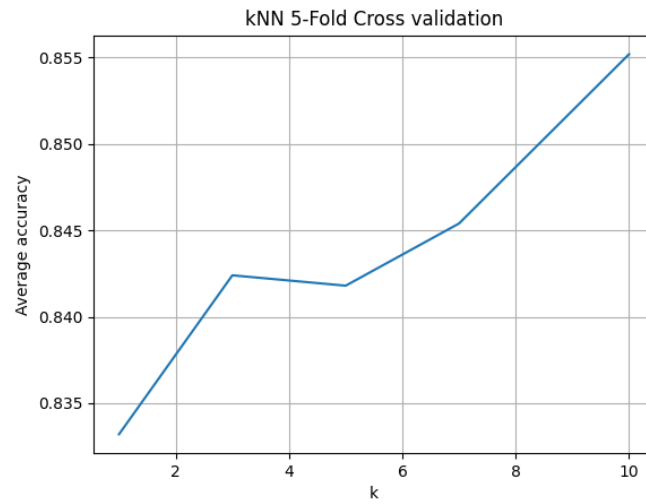
3. (12 pts) Implement logistic regression (from scratch). Use gradient descent (refer to question 6 from part 1) to find the optimal parameters. You may need to tune your learning rate to find a good optimum. Run 5-fold cross validation. Report accuracy, precision, and recall in each fold.

Hyperparameters: $\eta = 0.001$, Episodes = 100

Fold	Accuracy	Precision	Recall
1	0.957	0.9653846153846154	0.8807017543859649
2	0.959	0.9436090225563910	0.9061371841155235
3	0.936	0.9330708661417323	0.8345070422535211
4	0.930	0.9057971014492754	0.8503401360544217
5	0.918	0.8435582822085890	0.8986928104575164

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.

My figure: (Slightly different from the expected figure, there is no drop for my figure. I believe this is due to tie breaking when two training data points are equidistant to the test point)



k	Average Accuracy
1	0.8332
3	0.8424
5	0.8418
7	0.8454
10	0.8552

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.

My figure:

