Homework 3

$RishideepReddyRallabandi$
9084949099

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.
Github link: https://github.com/Rishideep08/CS760-HW3

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.
   It is regression because we are predicting the CEO salary which is a continuous variable, n = 500, p = 3.

   (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.
   It is classification because we are predicting a discreate variable, n = 20, 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.
   It is regression becuase we are predicting % change in the US dollar which is the continous variable, n=52, 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, $X_1 = X_2 = X_3 = 0$.
   $3, 2, \sqrt{10} \approx 3.162, \sqrt{5} \approx 2.236, \sqrt{2} \approx 1.414, \sqrt{3} \approx 1.732$

   (b) (2 pts) What is our prediction with $K = 1$? Why?
   Prediction : Green. The 5th observation is closest of all and its prediction is Green.

(c) (2 pts) What is our prediction with $K = 3$? Why?
Prediction: Red. The closest 3 observations are 5th,6th,2nd observations and the majority among them 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?
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?
0.01

(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?
$(0.1)^{100}$

(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.
If we denote the fraction of available observations needed for prediction as $f(p)$, where p is the number of features, then the generic equation is given by:

$$f(p) = (0.1)^p$$

This is a decreasing function, indicating that as the number of features p increases, the fraction of available training observations needed for prediction decreases. In the limit as p approaches infinity, $f(p)$ tends towards zero. This suggests that with a large number of features, very few training observations are used for prediction.

(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 $\lim_{p\to\infty}$.
Let's consider a p-dimensional hypercube with side length a. The volume of this hypercube is given by a^p. Given that all features X are uniformly distributed on the interval $[0, 1]$, the total volume is 1. If 10% of the data is used for training observations, this implies $a^p = 0.1$. Solving for a, we get $a = (0.1)^{1/p}$.
For different values of p:

$$\begin{aligned}
&\text{For } p = 1, \quad a = 0.1\\
&\text{For } p = 2, \quad a = (0.1)^{1/2} \approx 0.3162\\
&\text{For } p = 100, \quad a = (0.1)^{1/100} \approx 0.9772
\end{aligned}$$

As p approaches infinity, a approaches 1. This suggests that the hypercube centered on the test observation, which on average contains 10% of the training observations, needs to be nearly the same size as the hypercube with all observations.

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 $\frac{8+974}{8+2+16+974} \approx 0.982$

(b) (2 pts) Precision $\frac{8}{8+16} \approx 0.33$

(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.

Table 1: Threshold, TPR, and FPR Values

Threshold	**True Positive Rate (TPR)**	**False Positive Rate (FPR)**
0.0	1	1
0.35	$\frac{5}{6}$	0.5
0.6	0.5	0.25
0.82	$\frac{1}{3}$	0
0.97	0	0

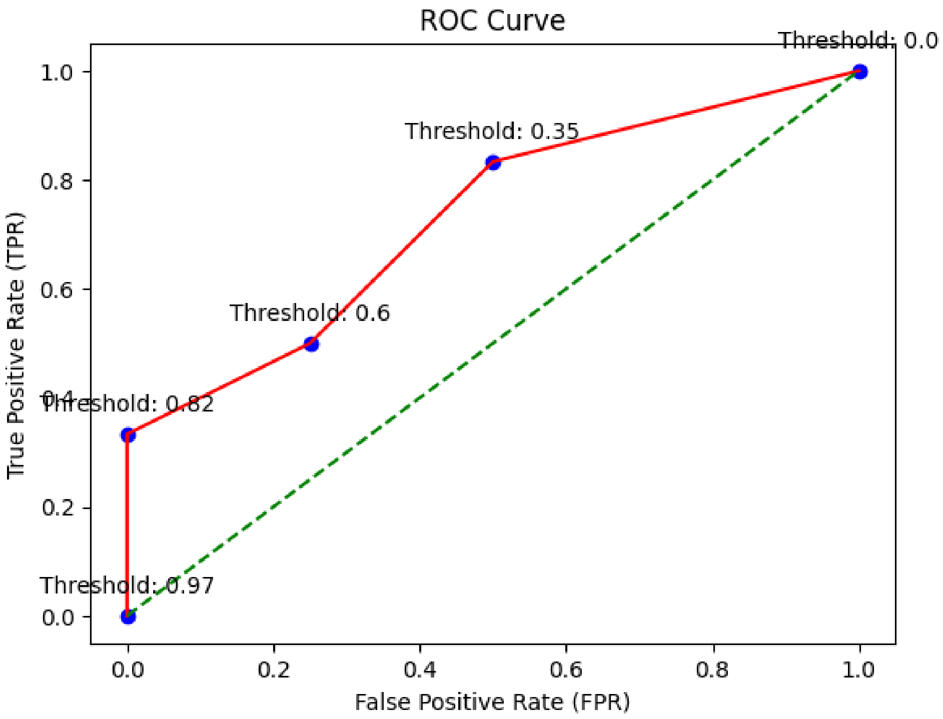


Figure 1: ROCcurve

(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.

The choice of a threshold in classification models, such as those used for distinguishing spam emails, is contingent on the relative costs associated with False Positives (FP) and False Negatives (FN). In the context of email filtering, prioritizing the reduction of FP is often crucial, as misclassifying important non-spam emails as spam could have significant consequences. Therefore, a lower False Positive Rate (FPR) is desirable to minimize the likelihood of flagging legitimate emails.

However, it's essential to strike a balance, as an overly stringent threshold that results in an extremely low True Positive Rate (TPR) could lead to a high number of False Negatives (FN). In this scenario, many spam emails might go undetected, posing a risk to the effectiveness of the spam filter. Therefore, the choice of a threshold involves a trade-off, and in this case, opting for a threshold of 0.6 strikes a balance by maintaining a low FPR to avoid unnecessary filtering of non-spam emails while achieving a satisfactory TPR for effective spam detection.

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

$$\frac{\partial L}{\partial \theta_i} = -\left[y \frac{1}{\hat{y}} \frac{\partial \hat{y}}{\partial \theta_i} + (1-y) \frac{1}{1-\hat{y}} \frac{\partial (1-\hat{y})}{\partial \theta_i} \right]$$

$$= -\left[y \frac{1}{\hat{y}} \frac{\partial \sigma(\theta^\top x)}{\partial \theta_i} - (1-y) \frac{1}{1-\hat{y}} \frac{\partial \sigma(\theta^\top x)}{\partial \theta_i} \right]$$

As we know,

$$\frac{\partial \sigma(z)}{\partial z} = \sigma(z)(1-\sigma(z))$$

$$\frac{\partial L}{\partial \theta_i} = -\left[y\frac{1}{\hat{y}}\sigma(\theta^\top x)(1-\sigma(\theta^\top x))\frac{\partial(\theta^\top x)}{\partial \theta_i} - (1-y)\frac{1}{1-\hat{y}}\sigma(\theta^\top x)(1-\sigma(\theta^\top x))\frac{\partial(\theta^\top x)}{\partial \theta_i}\right]$$

$$= -\left[y(1-\hat{y})x_i - (1-y)\hat{y}x_i\right]$$

$$\frac{\partial L}{\partial \theta_i} = (\hat{y}-y)x_i$$

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

(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.

$$\hat{y} = \sigma(0*1 + 0*3 + 0*2) = \sigma(0) = 0.5$$

$$\theta_0 = \begin{bmatrix}0\\0\\0\end{bmatrix}$$

$$\nabla_\theta L(f(x;\theta), y) = \begin{bmatrix}(0.5-1)*1\\(0.5-1)*3\\(0.5-1)*2\end{bmatrix} = \begin{bmatrix}-0.5\\-1.5\\-1\end{bmatrix}$$

Substituting all these values in the update step:

$$\theta_1 = \theta_0 - \eta\nabla_\theta L(f(x;\theta), y)$$

$$\theta_1 = \begin{bmatrix}0\\0\\0\end{bmatrix} - 0.1\begin{bmatrix}-0.5\\-1.5\\-1\end{bmatrix}$$

$$\theta_1 = \begin{bmatrix}0.05\\0.15\\0.1\end{bmatrix}$$

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

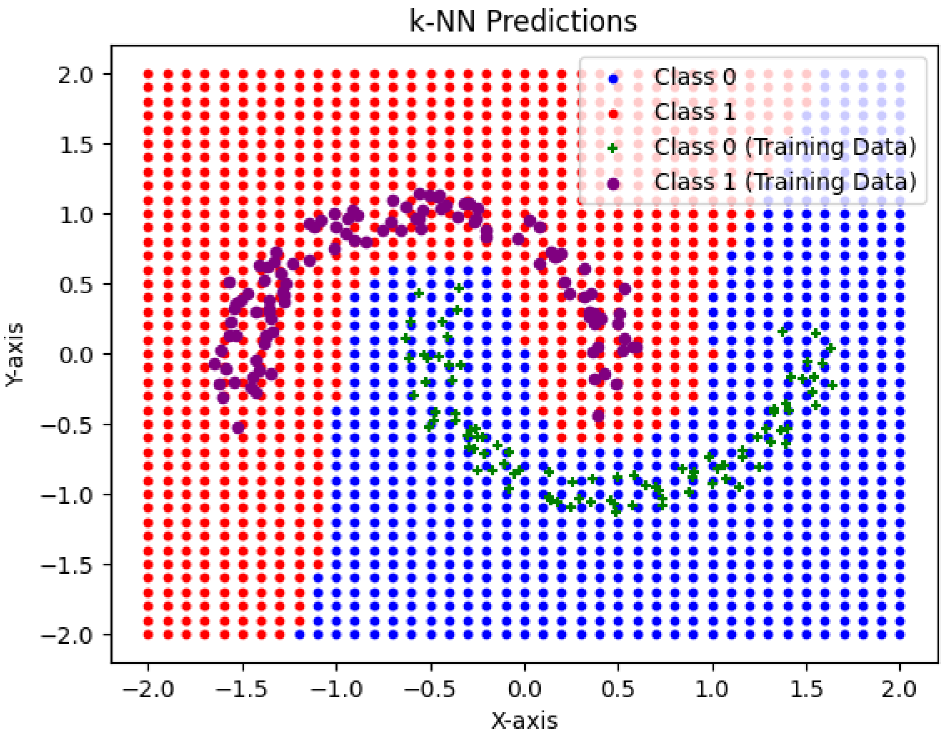


Figure 2: 1NN - D2z.txt

- 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.

Performance Metrics for Folds 1 to 5 - KNN

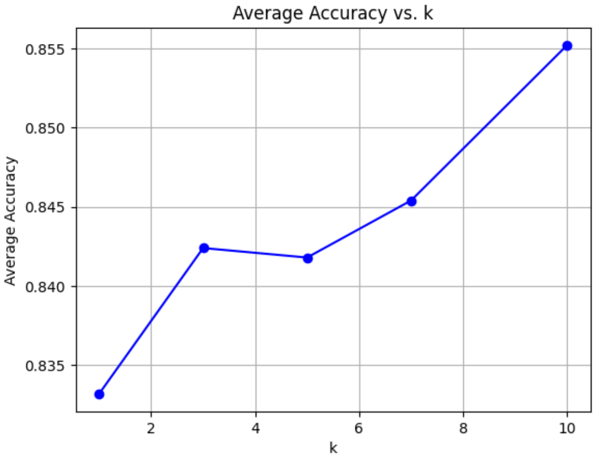
Fold	**Accuracy**	**Precision**	**Recall**
1	0.825	0.6545	0.8175
2	0.853	0.6857	0.8664
3	0.862	0.7212	0.8380
4	0.851	0.7164	0.8163
5	0.775	0.6057	0.7582

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.

Fold	Accuracy	Precision	Recall
1	0.909	0.83680556	0.84561404
2	0.896	0.82156134	0.79783394
3	0.89	0.82462687	0.77816901
4	0.844	0.66586538	0.94217687
5	0.849	0.78810409	0.69281046

Table 2: Performance Metrics for Each Fold - logistic

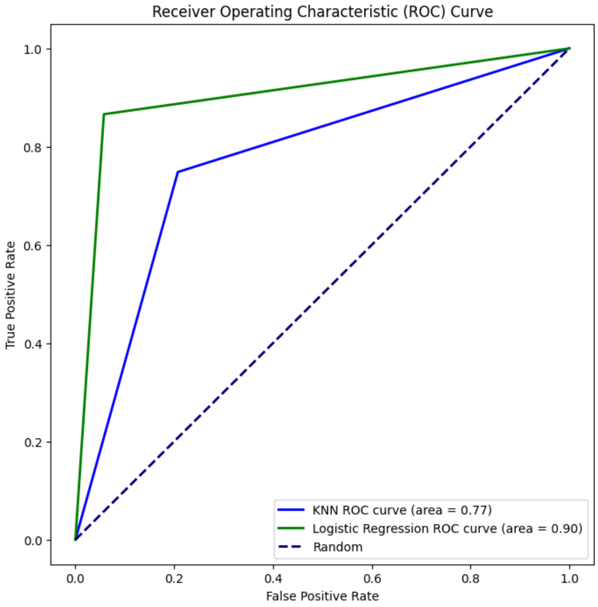
4. (10 pts) Run 5-fold cross validation with kNN varying k (k=1, 3, 5, 7, 10). Plot the average accuracy versus k, and list the average accuracy of each case.



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

Table 3: Average Accuracy for Different k Values

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.


Receiver Operating Characteristic (ROC) Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
KNN ROC curve (area = 0.77)
Logistic Regression ROC curve (area = 0.90)
Random