HOMEWORK 3

Anshuman Senapati
908 590 6916

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.

Link to GitHub repo: https://github.com/Anshu1245/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.
   Solution:
   As the target variable ”CEO Salary” is numeric, continuous and real-valued and hence, this is a regression problem. Here, $n = 500$, $p = 3$.

   (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.
   Solution:
   The target variable ”success” or ”failure” is binary and categorical and hence, this is a classification problem. Here, $n = 20$, $p = 13$.

   (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.
   Solution:
   The target variable ”% change in USD” is numeric, continuous and real-valued and hence, this is a regression problem. Here, $n = 52$, $p = 3$.

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, $X_1 = X_2 = X_3 = 0$.

Solution:

X_1	X_2	X_3	Y	Euclidean distance from $(0, 0, 0)$
0	3	0	Red	3
2	0	0	Red	2
0	1	3	Red	$\sqrt{10} = 3.16$
0	1	2	Green	$\sqrt{5} = 2.24$
-1	0	1	Green	$\sqrt{2} = 1.41$
1	1	1	Red	$\sqrt{3} = 1.73$

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

Solution:
For $K = 1$, the prediction is the label of the nearest data point to $(0, 0, 0)$, which is $(-1, 0, 1)$ according to Euclidean distance. Hence, the prediction is "Green".

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

Solution:
For $K = 3$, the prediction is the most popular label among the 3 nearest data points to $(0, 0, 0)$. These are $(-1, 0, 1)$, $(1, 1, 1)$ and $(2, 0, 0)$ with labels "Green", "Red" and "Red" respectively. Hence, 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?

Solution:
Let the test observation be $X = x_{test}$
The range where we are drawing our observations for comparison = $[x_{test} - 0.05, x_{test} + 0.05]$

When $0.05 \leq x_{test} \leq 0.95$, the fraction of observations in that range is
$= P(x_{test} - 0.05 \leq X \leq x_{test} + 0.05)$
$= \int_{x_{test}-0.05}^{x_{test}+0.05} 1 * dx = 0.1 = f_1$

When $x_{test} < 0.05$, the fraction of observations in that range is
$= P(0 \leq X \leq x_{test} + 0.05)$
$= \int_{0}^{x_{test}+0.05} 1 * dx = x_{test} + 0.05 = f_2$

When $x_{test} > 0.95$, the fraction of observations in that range is
$= P(x_{test} - 0.05 \leq X \leq 1)$
$= \int_{x_{test}-0.05}^{1} 1 * dx = 1.05 - x_{test} = f_3$

To get the fraction of available observations used on an average, we have to find the expectation of these fractions over the entire range of values x_{test} can take,
$\mathbb{E}$[fraction of observations used] $= \int_0^1 f * dx_{test}$
$= \int_0^{0.05} f_2 * dx_{test} + \int_{0.05}^{0.95} f_1 * dx_{test} + \int_{0.95}^{1} f_3 * dx_{test}$
$= \int_0^{0.05} (x_{test} + 0.05) * dx_{test} + \int_{0.05}^{0.95} 0.1 * dx_{test} + \int_{0.95}^{1} (1.05 - x_{test}) * dx_{test} = 0.0975$

(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:
Let $x_{test} = (x1_{test}, x2_{test})$. Similar to part (a), we will have the fraction of observations used as piece-wise functions of $x1_{test}$ and $x2_{test}$. Let the fractions for dimension $X1$ be $f_1 = \{f_{11}, f_{12}, f_{13}\}$ corresponding to the intervals $0.05 \leq X1 \leq 0.95, X1 < 0.05$ and $X1 > 0.95$ respectively. Similarly, for dimension $X2$ let them be $f_2 = \{f_{21}, f_{22}, f_{23}\}$. Now, x_{test} can be any combination of intervals in $f_1 \times f_2$. Expectation over these fractions is,

$\mathbb{E}[f_1 f_2] = \mathbb{E}[f_1] * \mathbb{E}[f_2]$, assuming $X1$ and $X2$ are independent of each other
$\mathbb{E}$[fraction of observations used] $= 0.0975 * 0.0975 = 0.0095$

(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:
Using the results derived above, $\mathbb{E}$[fraction of observations used] $= 0.0975^{100}$

(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.

Solution:
It can be clearly seen from the above results that as $p \to \infty$, $\mathbb{E}$[fraction of observations used] $\to 0$. And hence, as p becomes large, there are fewer and fewer training examples near the test observation on average, which is a drawback of using KNN for data with a large number of features.

(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 $\lim_{p\to\infty}$.

Solution:
Let the side be d. For a hypercube to contain 10% of the observations, it's volume should be 0.1.
if number of features = p,
$\implies 0.1 = d^p$
$\implies d = 0.1^{1/p}$

If $p = 1, d = 0.1$
If $p = 2, d = 0.1^{1/2} = 0.32$
If $p = 100, d = 0.1^{1/100} = 0.98$
Hence, as $lim_{p\to\infty} d = 1 \implies$ As the number of dimensions increases, the side of the hypercube required around the test observation to contain 10% of the observations (the same volume of 0.1) tends towards 1 (the complete range of any feature). That is, to get the same fraction of observations we have to look at larger and larger range of each feature around the test observation, as p increases. This corresponds with the earlier result that if we look at the same range each feature, we end up with smaller and smaller fraction of observations as p increases. This is due to the observations getting more spaced out as the number of dimensions increases.

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:
$Accuracy = \frac{TruePos+TrueNeg}{TruPos+FalsePos+TruNeg+FalseNeg} = \frac{8+974}{8+2+16+974} = 0.982$

(b) (2 pts) Precision
Solution:
$Precision = \frac{TruePos}{TruePos+FalsePos} = \frac{8}{8+16} = 0.33$

(c) (2 pts) Recall Solution:
$Recall = \frac{TruePos}{TruePos+FalseNeg} = \frac{8}{8+2} = 0.8$

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.
Solution:

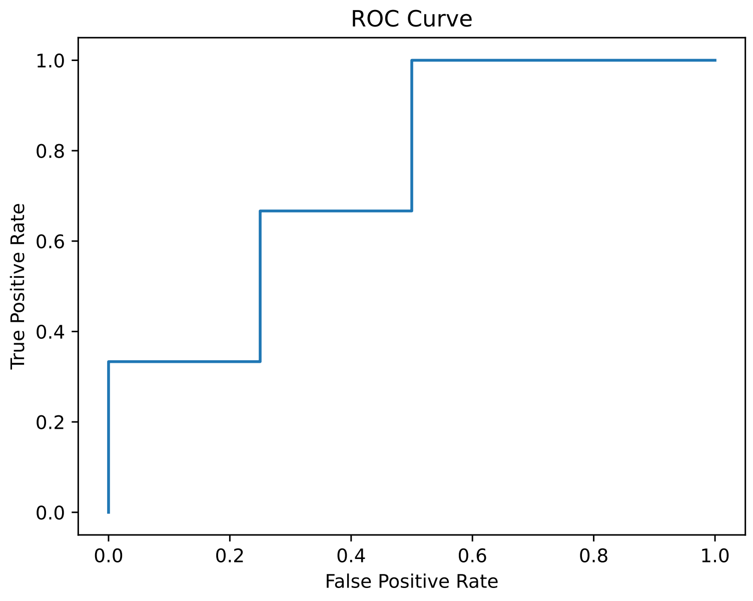

Figure 1: ROC Curve based on the above table

(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.
Solution:
We will ideally like to choose the threshold that gives us the point on the ROC curve closest to the

top left corner (corresponding to True Positive Rate = 1 and False Positive Rate = 0). This is the point where TPR = 0.66 and FPR = 0.25 and gives us the range of thresholds to be in the interval $(0.45, 0.55)$.

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

Solution:
$\frac{\partial}{\partial\theta}\{\sigma(\theta^\top x)\} = \frac{\partial\sigma(\theta^\top x)}{\partial(\theta^\top x)} \cdot \frac{\partial(\theta^\top x)}{\partial\theta}$
$= \frac{\partial}{\partial(\theta^\top x)}\{\frac{1}{1+e^{-\theta^\top x}}\} \cdot \frac{\partial(\theta^\top x)}{\partial\theta}$
$= \frac{0-(-e^{-\theta^\top x})}{(1+e^{-\theta^\top x})^2} \cdot x = \sigma(\theta^\top x) \cdot \frac{1+e^{-\theta^\top x}-1}{1+e^{-\theta^\top x}} \cdot x$
$= \sigma(\theta^\top x) \cdot (1 - \sigma(\theta^\top x)) \cdot x$

$\nabla_\theta L(f(x; \theta), y) = \frac{\partial L}{\partial\theta} = \frac{\partial L}{\partial\hat{y}} \cdot \frac{\partial\hat{y}}{\partial\theta} = -[\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}] \cdot \frac{\partial\hat{y}}{\partial\theta}$
$= -[\frac{y}{\sigma(\theta^\top x)} - \frac{1-y}{1-\sigma(\theta^\top x)}] \cdot \frac{\partial\sigma(\theta^\top x)}{\partial\theta}$
$= -[\frac{y}{\sigma(\theta^\top x)} - \frac{1-y}{1-\sigma(\theta^\top x)}] \cdot \sigma(\theta^\top x) \cdot (1 - \sigma(\theta^\top x)) \cdot x$
$= -[y \cdot (1 - \sigma(\theta^\top x)) - (1 - y) \cdot \sigma(\theta^\top x)] \cdot x$
$= -[y - \sigma(\theta^\top x)] \cdot x$

(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.

Solution:
$\hat{y} = \sigma(\theta_0^\top x) = \sigma((0, 0, 0) \cdot (1, 3, 2)) = \sigma(0) = 0.5$
$\nabla_\theta L(\hat{y}, y)|_{x,\theta_0} = -[1 - 0.5] \cdot (1, 3, 2) = (-0.5, -1.5, -1)$
$\theta_1 = \theta_0 - \eta\nabla_\theta L = (0, 0, 0) - 0.1 \cdot (-0.5, -1.5, -1) = (0.05, 0.15, 0.1)$

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.

Solution:

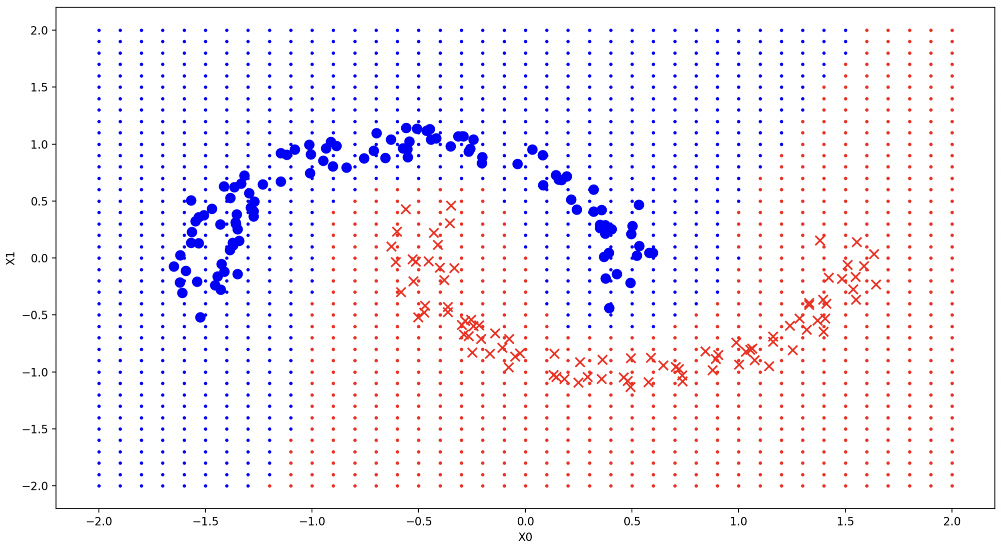

Figure 2: Scatter plot showing training data (big dots and crosses) from D2z.txt and the prediction by 1-KNN over a mesh. Blue corresponds to Class = 1, Red corresponds to Class = 0

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.
   Solution:

Iteration	**Test set**	**Train set**	**Accuracy**	**Precision**	**Recall**
1	email 1-1000	the rest	0.825	0.654	0.818
2	email 1000-2000	the rest	0.853	0.686	0.866
3	email 2000-3000	the rest	0.862	0.721	0.838
4	email 3000-4000	the rest	0.851	0.716	0.816
5	email 4000-5000	the rest	0.775	0.606	0.758

Table 1: Accuracy, Precision and Recall for each fold in 5-fold cross validation of 1NN on the Emails dataset

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:

Iteration	**Test set**	**Train set**	**Accuracy**	**Precision**	**Recall**
1	email 1-1000	the rest	0.918	0.901	0.8
2	email 1000-2000	the rest	0.927	0.878	0.856
3	email 2000-3000	the rest	0.895	0.924	0.687
4	email 3000-4000	the rest	0.891	0.890	0.718
5	email 4000-5000	the rest	0.884	0.860	0.742

Table 2: Accuracy, Precision and Recall for each fold in 5-fold cross validation of logistic regression on the Emails dataset

The number of gradient descent steps and the learning rate η were varied. As the $\#steps$ was increased from 500, 1000, 1500, 2000, there was an increase in the average accuracy with a decreasing marginal improvement. Similarly, as η was decreased from 0.01, 0.005, 0.001 the average accuracy increased. The best performance was obtained on $\eta = 0.001$ and $\#steps = 2000$.

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.

Solution:

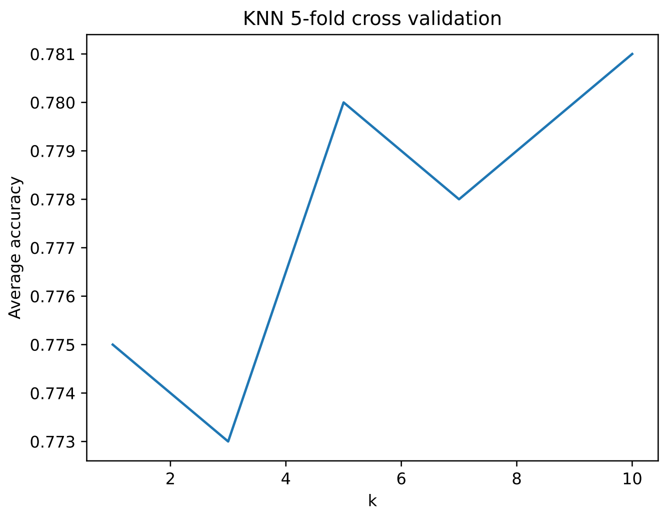

Figure 3: Average accuracy vs k plot for 5-fold cross validation with kNN

The list of average accuracies is as follows,

k	Avg Accuracy
1	0.775
3	0.773
5	0.78
7	0.778
10	0.781

Table 3: List of Avg Accuracies and k

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.

Solution:

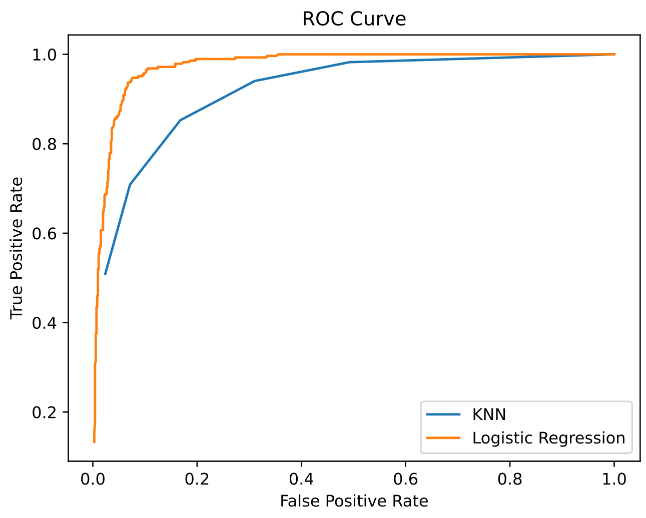

Figure 4: ROC Curve for 5NN and Logistic Regression. The first 1000 emails were taken as test set and the rest were used for training.