HOMEWORK 3

Devansh Goenka
908 335 1354

Instructions: Although this is a programming homework, you only need to hand in a pdf answer file. There is no need to submit the latex source or any code. You can choose any programming language, as long as you implement the algorithm from scratch.
Use this latex file as a template to develop your homework. Submit your homework on time as a single pdf file to Canvas. 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.

   Since we are predicting the salary here, which is a real number, this is is a regression problem. The data set has 500 firms, so $n = 500$ and we are using 3 features [profit, number of employees and industry], so $p = 3$.

   (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.

   Since we are predicting whether the product will be a success/failure, this is a binary classification problem. Moreover, $n = 20$ as we use the data from 20 similar products, and $p = 13$ [price charged, marketing budget, competition price + 10 more 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.

   Since the percentage change is a real number, this is again a regression problem. The weekly data for 2012 consists of 52 weeks, and so $n = 52$. $p = 3$ as we are using three features [% change in US market, % change in British market and % change in 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, $X_1 = X_2 = X_3 = 0$.
The Euclidean distance between each observation and the test point $X_1 = X_2 = X_3 = 0$ is as follows:

X_1	X_2	X_3	$Distance$
0	3	0	3
2	0	0	2
0	1	3	3.162
0	1	2	2.236
-1	0	1	1.414
1	1	1	1.732

(b) (2 pts) What is our prediction with $K = 1$? Why?
With $K = 1$, we only look at the closest neighbor to the test point from the given set of observation points. In this case, the closest point to the test point $X_1 = X_2 = X_3 = 0$ is the point $X_1 = -1, X_2 = 0, X_3 = 1$.
Since the label of the closest neighbor is Green, the prediction for our test point will also be Green.

(c) (2 pts) What is our prediction with $K = 3$? Why?
With $K = 3$, we now look at the 3-closest neighbors of the given test point. In this case, the three closest neighbors are:
$X_1 = -1, X_2 = 0, X_3 = 1$
$X_1 = 1, X_2 = 1, X_3 = 1$
$X_1 = 2, X_2 = 0, X_3 = 0$

Now, since among these 3 points, the majority prediction is Red, the prediction for our test point 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?
We have 3 cases to consider here. Case 1: $x < 0.05$: In this case, the range for x will be $[0, x + 0.05]$
Case 2: $x > 0.95$: In this case, the range for x will be $[x - 0.05, 1]$
Case 3: $x \in [0.05, 0.95]$: In this case, the range for x will be $[x - 0.05, x + 0.05]$

Since x is uniform, we can think of x as a continuous variable and integrate over the given ranges to find the fraction of the observations that we use.
For Case 1: We have $\int_0^{0.05}(x + 0.05)dx = 0.00375$
For Case 2: We have $\int_{0.95}^{1}(1 - (x - 0.05))dx = 0.00375$
For Case 3: We have $\int_{0.05}^{0.95} 0.1dx = 0.09$

Therefore, combining these, we get the fraction of input space as 0.0975, or 9.75% of the available observations.

(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?

Since $X1$ and $X2$ are individually uniformly distributed, we can assume that they are independent. For each individual variable, we have 9.75% of the available range as we derived earlier. Thus, for $(X1, X2)$, we have:
$9.75\% * 9.75\% = 0.95\%$ of the combined range of observations.

(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?

Similar to the above approach, we can say that when $p = 100$, the percentage of available range of observations becomes:
9.75^{100}, the limits of which are very close to 0.

(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.

From the above answers, we can clearly see that the number of available observations near a given test observation diminishes exponentially with respect to the number of features p. Thus, it is true that KNNs suffer when the dimensionality of the feature space increases.

(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 $p = 1$, the hypercube is 1-dimensional and the length of its side is equal to the set of 10% training examples in its vicinity. Therefore, $l_{p=1} = 0.1$.
For $p = 2$, we use the euclidean distance in the 2D space. Therefore, $l_{p=2} = \sqrt{0.1}$.
Now, we can generalize this for an n-dimensional space as follows:
$l_{p=n} = 0.1^{1/n}$. Therefore, for $p = 100$, we have $l_{p=100} = 0.1^{1/100}$.
This also goes to show how the prediction space decreases exponentially as the dimensionality of the feature space increases, and aligns with the drawback we identitified in the previous answer.

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 = $\frac{TP + TN}{TP + FP + TN + FN} = \frac{8 + 974}{8 + 974 + 2 + 16} = 0.982$

(b) (2 pts) Precision

Precision = $\frac{TP}{TP + FP} = \frac{8}{8 + 16} = 0.333$

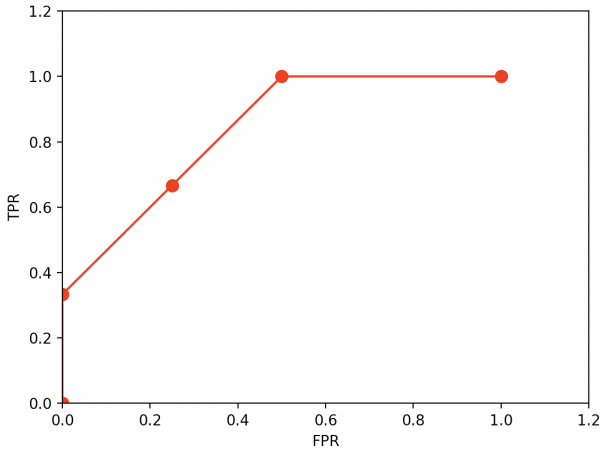
(c) (2 pts) Recall

Recall = $\frac{TP}{TP + FN} = \frac{8}{8 + 2} = 0.8$

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.
Here is the ROC curve for 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.
Based on the above ROC curve, it would be prudent to choose the threshold around the 55% confidence positive mark. This is because at that point, we have considerable trade off between the TPR and the FPR. If we go any higher, our TPR reduces drastically, and if we go lower, the FPR starts to increase.

6. (8 pts) In this problem, we will walk through a single step of the gradient descent algorithm for logistic regression. Assume two dimension input. Recap:

$$f(x; w, b) = \sigma(w \cdot x + b)$$

$$\text{Loss} L(\hat{y}, y) = -y \log \sigma(w \cdot x + b) + (1 - y) \log(1 - \sigma(w \cdot x + b))$$

$$\text{The single update step } \theta^{t+1} = \theta^t - \eta \nabla_\theta L(f(x; \theta), y), \text{ where } \theta = [w_1, w_2, b]^T$$

Now given

$$\text{Initial parameters} : w_1 = w_2 = b = 0, (\Rightarrow \theta^0 = [0, 0, 0]))$$

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

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

(a) (4 pts) Compute the first gradient $\nabla_\theta L(f(x; \theta), y)$.
We know that:

$$\nabla_\theta L(f(x; \theta), y) = \begin{bmatrix} \frac{\partial L(f(x;\theta),y)}{\partial w_1} \\ \frac{\partial L(f(x;\theta),y)}{\partial w_2} \\ \frac{\partial L(f(x;\theta),y)}{\partial b} \end{bmatrix}.$$

And, $\frac{\partial L(f(x;\theta),y)}{\partial w_j} = [\sigma(w \cdot x + b)] * x_j$

Therefore,

$$\nabla_\theta L(f(x;\theta),y) = \begin{bmatrix} [\sigma(0) - 1] * 3 \\ [\sigma(0) - 1] * 2 \\ [\sigma(0) - 1] \end{bmatrix} = \begin{bmatrix} -1.5 \\ -1.0 \\ -0.5 \end{bmatrix}$$

(b) (4 pts) Compute the updated parameter vector θ^1 from the single update step.

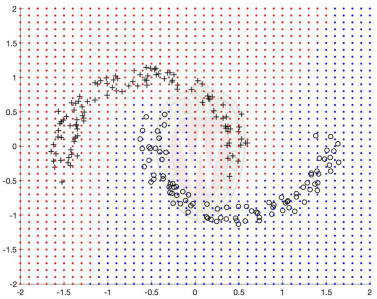
$\theta^1 = \theta^0 - \eta \nabla_\theta L(f(x;\theta),y)$

$$\theta^1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} - 0.1 \begin{bmatrix} -1.5 \\ -1.0 \\ -0.5 \end{bmatrix} = \begin{bmatrix} 0.15 \\ 0.10 \\ 0.05 \end{bmatrix}$$

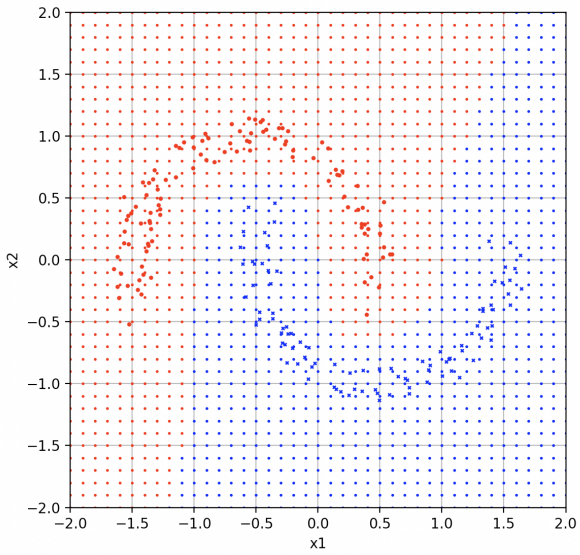
2 Programming (50 pts)

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

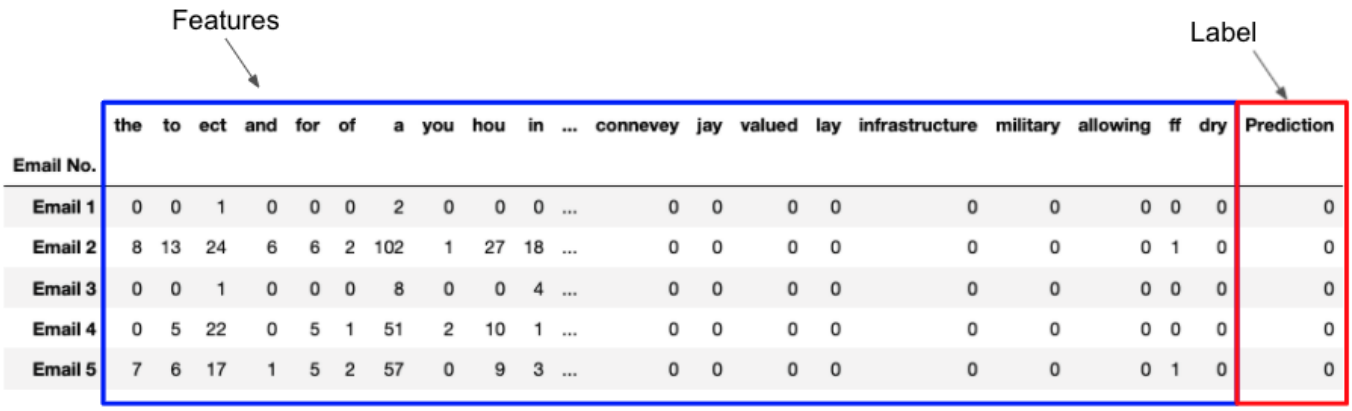
   The expected figure looks like this.



The plot for the test points is as follows:
(Note, the red points indicate prediction label 1 and the blue points indicate prediction label 0)



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



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. (10 pts) Implement 1NN, Run 5-fold cross validation. Report accuracy, precision, and recall in each fold.

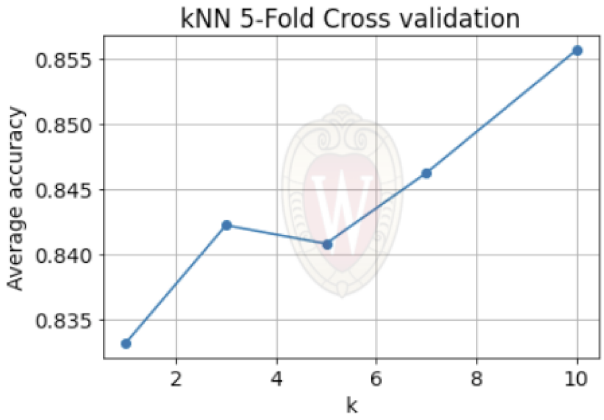
   After implementing 1NN, here are the results:
   Fold 1:
   Accuracy = 0.788
   Precision = 0.627
   Recall = 0.669
   Fold 2:
   Accuracy = 0.827
   Precision = 0.666
   Recall = 0.803
   Fold 3:
   Accuracy = 0.786
   Precision = 0.607
   Recall = 0.751
   Fold 4:
   Accuracy = 0.767
   Precision = 0.571
   Recall = 0.767
   Fold 5:
   Accuracy = 0.737
   Precision = 0.526
   Recall = 0.677

3. (10 pts) Implement logistic regression, Run 5-fold cross validation. Report accuracy, precision, and recall in each fold.

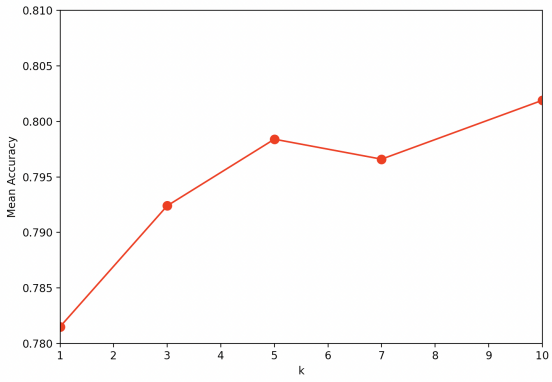
   After implementing logistic regression, here are the results:
   Fold 1:

Accuracy = 0.916
Precision = 0.871
Recall = 0.835
Fold 2:
Accuracy = 0.940
Precision = 0.880
Recall = 0.915
Fold 3:
Accuracy = 0.908
Precision = 0.816
Recall = 0.885
Fold 4:
Accuracy = 0.932
Precision = 0.848
Recall = 0.933
Fold 5:
Accuracy = 0.935
Precision = 0.869
Recall = 0.906

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.



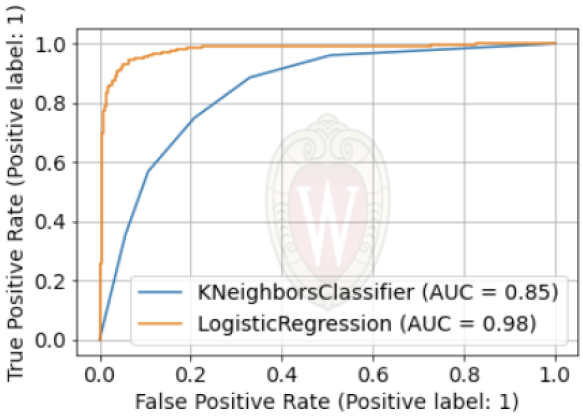
After varying k as mentioned in the problem and plotting mean accuracy across all folds, here is the plot:



Here are the average accuracy for each case:
k=1, accuracy = 0.7815
k=3, accuracy = 0.7924

k=5, accuracy = 0.7984
k=7, accuracy = 0.7966
k=10, accuracy = 0.8019

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.



Here is the ROC curve derived from a single train/test split: