HOMEWORK 3

GURUPRASAD VISWANATHAN RAMESH
9082378762

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/Guruprasad68/CS760-Machine-Learrning-Spring2023
Code can be found in the ipynb folder of the repo

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).
The type of our output space decides whether it is a classification or regression problem. In the case of classification, prediction is discrete, and for regression, the output is continuous.

   (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 salary is a continuous value, this problem is regression. And $n = 500$, $p = 3$

   (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.

   This is a classification problem as the output is discrete. It can either be a success or a failure. 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.

   Since % change is a continuous variable, our problem is regression. Here, n is the number of weeks in a year, i.e., 52, and p equals 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, $X_1 = X_2 = X_3 = 0$.

The euclidean distance of two points (x_1, y_1, z_1) and (x_2, y_2, z_2) is given by:

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

Using this formula, we can compute the distance of $(0, 0, 0)$ from each of the six points in the dataset. The distances are tabulated in the table below:

X_1	X_2	X_3	Y	Distance to (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	$\sqrt{5} = 2.23$
-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?

If $K = 1$, then we take the closest point to $(0, 0, 0)$ and use the label of that point as the prediction of $(0, 0, 0)$. In this case, the closest point is $(-1, 0, 1)$ with a distance of 1.414. Thus, the prediction is Green.

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

If $K = 3$, then we'll consider the three closest points to $(0, 0, 0)$ and use the majority label of the three points as the prediction for $(0, 0, 0)$. In this case, the three closest points to $(0, 0, 0)$ are $(-1, 0, 1)$, $(1, 1, 1)$, $(2, 0, 0)$ with distances 1.414, 1.73 and 2 respectively. The majority label of these three is Red, and thus, our prediction of $(0, 0, 0)$ 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?

We know that: $P(a \leq x \leq b) = \int_a^b f(x)dx$, where $f(x)$ is the probability density function of our distribution. In the case of uniform distribution, the pdf is $\frac{1}{b-a}$ and, in our case, $f(x) = 1$. Therefore, the integral is equal to $b - a$. Since, we are considering points lying within 10% of x, $a = x - 0.05$ and $b = x + 0.05$. Thus, $P(x - 0.05 \leq x \leq x + 0.05) = 0.1$. Therefore, on average we can say we will be using 10% of the available observations to make our prediction.

(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?

Using the same logic as (a), we can say fraction of available observations within 10% of $X1$ is 10%, and 10% of $X2$ is 10%. However, the region common to $X1$ and $X2$ is what we're interested in. The probability of a point lying in this region is 0.01 or 1% of available points in the data on average will be needed for making 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?

Extrapolating the logic of (a) and (b), we get 0.1^{100} as the probability of a point lying in the common region of the 10% range of the 100 features. That is, only $(0.1)^{98}\%$ of available observations will be used to make a prediction. This is a very, very small number and would require a huge amount of observations in order to find a single point in the 10% region.

(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.

Observations in (a)-(c) show empirical evidence for the curse of dimensionality. The probability of finding an observation for a given point reduces as the number of features increases, from what we notice in (a)-(c). The empirical evidence is done using a uniform distribution, and thus a more formal proof that is applicable across all distributions would be a better way to say KNN doesn't do well for high-dimensional data. Also, we are restricting ourselves to 10% range of each feature instead of K-closest observations for a given point. This assumption is also crucial in our evidence

(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.

From (a) to (c), we can say that probability of the region we are choosing to find observations for a given point is equal to a^p, where a is the side of the hypercube and p is the number of features. Given hypercube volume is 0.1 or 10%. Thus, we can say:
When p=1, a=0.1
When p=2, $a = 0.1^{1/2}$
When p=100, $a = 0.1^{1/100}$

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=$\frac{TP+TN}{TP+FN+TN+FP} = \frac{8+974}{1000}$=0.982

(b) (2 pts) Precision Precision=$\frac{TP}{TP+FP} = \frac{8}{24}$=0.33

(c) (2 pts) Recall Recall=$\frac{TP}{TP+FN} = \frac{8}{10}$=0.80

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.

For plotting the ROC curve, we need to estimate the True Positive Rate (TPR) and False Positive Rate (FPR). They are defined by: TPR=$\frac{TP}{TP+FN}$ and FPR=$\frac{FP}{TN+FP}$. Using this and considering the points where the actual label changed from + to - as thresholds, we can arrive at Table 1. Using the TPR and FPR values, we can plot the ROC curve, refer to figure 1.

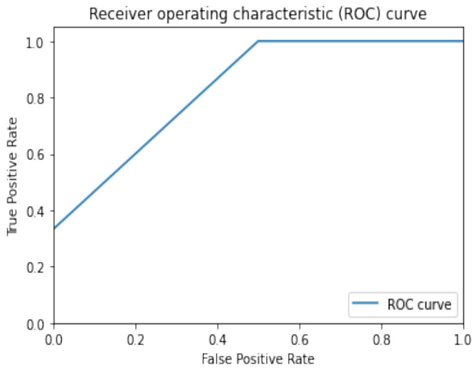

Figure 1: ROC Curve obtained from the TPR and FPR values in Table 1

Threshold	TPR	FPR
0.95		
0.85	2/6	0
0.8		
0.7		
0.55	4/6	0.25
0.45		
0.4		
0.3	1	0.5
0.2		
0.1	1	1

Table 1: Question 1-5(a): TPR and FPR for various thresholds

(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 application plays a huge role in deciding which threshold we choose for our model. For a spam classifier, it is important for the model to classify spam accurately but not completely at the cost of misclassifying a lot of proper mail. So, a fair point of threshold is 0.55 where TPR=4/6 and FPR=0.25. One could also go for threshold=0.3 if the main goal is to classify spam correctly, where TPR=1, and FPR=0.5.

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

To compute the gradient, we need to use the chain rule. If $\hat{y} = \sigma(\theta^\top x) = \sigma(z)$. Then,

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

Let us proceed by computing each of these terms separately.

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

$$\nabla_z \hat{y} = \sigma(z)(1-\sigma(z)) = \hat{y}(1-\hat{y})$$

$$\nabla_\theta z = x$$

.

Therefore, multiplying the individual terms we get,

$$\nabla_\theta L(f(x;\theta),y) = -x[y(1-\hat{y}) - \hat{y}(1-y)] = x(\hat{y}-y)$$

.

(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.

$\hat{y} = \sigma(\theta^{0\top}x) = \sigma([0,0,0][1,3,2]^\top) = \sigma(0) = 0.5.$
Substituting $\hat{y}$ and the gradient from (a), we have, $\theta^1 = \theta^0 - \eta\nabla_{\hat{y}}L(f(x;\theta),y) = [0,0,0] - 0.1 * [1,3,2](-0.5) = [0.05, 0.15, 0.1]$.

$$\theta^1 = [0.05, 0.15, 0.1]$$

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. The grid fused with the train data after 1NN can be seen in Figure 2.

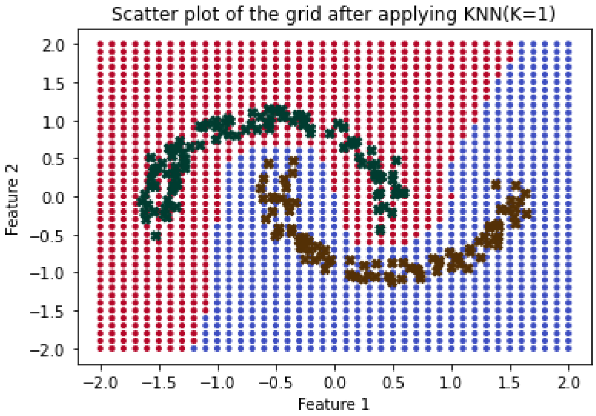

Figure 2

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.

   The results of the implementation are as follows:
   Fold:0
   The precision score is:0.6544943820224719
   The recall score is:0.8175438596491228
   The accuracy score is:0.825
   Fold:1
   The precision score is:0.6857142857142857
   The recall score is:0.8664259927797834
   The accuracy score is:0.853
   Fold:2
   The precision score is:0.7212121212121212
   The recall score is:0.8380281690140845
   The accuracy score is:0.862
   Fold:3
   The precision score is:0.7164179104477612
   The recall score is:0.8163265306122449
   The accuracy score is:0.851
   Fold:4
   The precision score is:0.6057441253263708
   The recall score is:0.7581699346405228
   The accuracy score is:0.775

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.

   I ran logistic regression for 1000 epochs with LR=5e-3 on all the folds. Using loss as criterion, I used the best model obtained for prediction. This was done without early stopping as it hasn't been covered yet in the class. One observation I could make is the fluctuating nature of the losses irrespective of the learning rate and the model learning even after 800-900 epochs. The results of the different folds are as follows:
   Fold:0
   The precision score is:0.8598484848484849
   The recall score is:0.7964912280701755
   The accuracy score is:0.905
   Fold:1
   The precision score is:0.8530612244897959
   The recall score is:0.7545126353790613
   The accuracy score is:0.896
   Fold:2
   The precision score is:0.5204081632653061
   The recall score is:0.897887323943662
   The accuracy score is:0.736
   Fold:3
   The precision score is:0.9591836734693877
   The recall score is:0.1598639455782313
   The accuracy score is:0.751
   Fold:4
   The precision score is:0.5380333951762524
   The recall score is:0.9477124183006536
   The accuracy score is:0.735

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.

   The average accuracy for K=1 is: 0.8332.
   The average accuracy for K=3 is: 0.8422000000000001.
   The average accuracy for K=5 is: 0.8408.
   The average accuracy for K=7 is: 0.8462.
   The average accuracy for K=10 is: 0.8556000000000001.

   The accuracy plot is in Figure 3

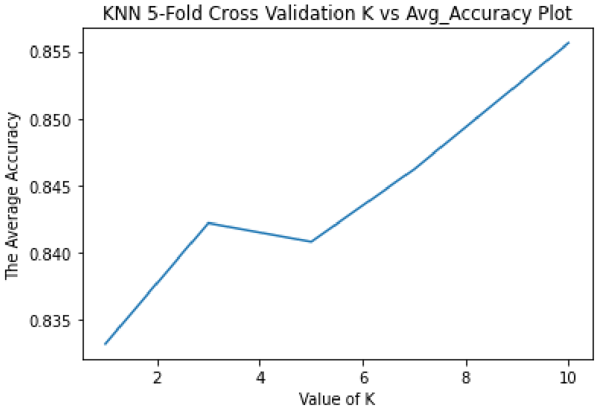

Figure 3

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.

   The ROC curve for KNN (k=5) and Logistic Regression can be found in Figure 4.

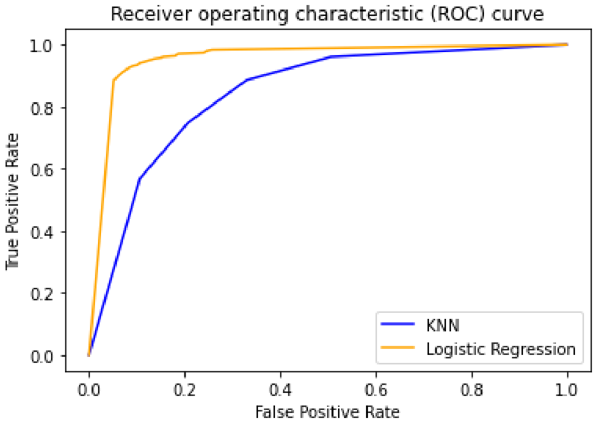

Figure 4