HOMEWORK 3

BATCHU S S S S HARISH BABU
sbatchu2(NetID) - 9084603043

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.

https://github.com/bharishb/ECE760_spring2022/tree/main/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.

   Its a Regression problem since we are predicting CEO salary which is a continuous variable y $\in R$. That is infinite possibilities. Classification works for labelling into fixed number of outputs which is not the case here. $y = \theta 1x1 + \theta 2x2 + \theta 3x3$ where features x1, x2, x3 = profit,number of employees, industry. Number of data points n = 500, Number of features 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 problem since the output $y \in success, failure$, fixed discrete outputs. Since we are trying are predict the output into two discrete categories. Number of data points n = 20. Number of features p = 13, price, marketing budget, competition price, 10 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.

   It is Regression problem since we are predicting percent change in US dollar which is a continuous variable $y \in R$. Number of data points n = 52(since 52 weeks in an year), Number of features p = 3, % change in US market, % change in British market, % 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, $X_1 = X_2 = X_3 = 0$.

X_1	X_2	X_3	Y	$EuclideanDistance$
0	3	0	Red	3
2	0	0	Red	2
0	1	3	Red	$\sqrt{10}$ = 3.162278
0	1	2	Green	$\sqrt{5}$ = 2.236068
-1	0	1	Green	$\sqrt{2}$ = 1.414214
1	1	1	Red	$\sqrt{3}$ = 1.732051

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

With K=1 , we need to look for the label of nearest neighbour in terms of Euclidean distance. As per the above table, X1=-1, X2=0, X3=1 is the nearest neighbour. So, the prediction is Y(-1,0,1) = Green

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

With K=3, we need to look for 3 nearest neighbours. We will find the first three data points in a sorted list of the data points w.r.t Euclidean distance. From the above table, (X1, X2, X3) = (-1,0,1), (1,1,1) , (2,0,0) are the 3 nearest neighbours. Now we will look at the label w.r.t plurality class of Y. That is two of them has RED and one has Green. So, our prediction is the plurality class which 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?

For X $\in [0, 1]$, and since the window of observations is 0.1, That is +/- 0.05 of the test point x. The window is not exactly 0.1 at the ends of the interval. That is $X <= 0.05, X >= 0.95$. For finding out the average of the window fraction, we need to integrate this over test point x.
$\int_0^1 f(x)\,dx = \int_0^{0.05} f(x)\,dx + \int_{0.05}^{0.95} f(x)\,dx + \int_{0.95}^1 f(x)\,dx$, where f(x) is fraction of observations as a function of test point x. Implies,
$\int_0^1 f(x)\,dx = \int_0^{0.05}(x + 0.05)\,dx + \int_{0.05}^{0.95} 0.1\,dx + \int_{0.95}^1 0.05 + (1 - x)\,dx$
$= 0.05^2/2 + 0.05^2 + 0.1 * 0.9 + 0.05^2 + 0.05^2/2 = 0.0975$

(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?

Same as above problem, we need to use integrals to solve this. Instead of one integral, we need to use double integral over x1, x2. Since x1, x2 window fractions are independent of each other. we can seperate the double integral to product of individual integrals
$\int_0^1 \int_0^1 f(x1, x2)\,dx1dx2 = \int_0^1 f(x1)\,dx1 * \int_0^1 f(x2)\,dx2$ where f(x) is fraction of observations as a function of test point x, Implies
= (0.0975) * (0.0975) = 0.00950625.
If they are to be dependent features, we need to know the joint probability distribution to evaluate the fraction.

(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?

Extending the above problem, to 100 features. Assuming all these features are independent of each other. Fraction of available observations = (0.0975^{100}) = 7.9517E-102

(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.

Using the generalization we can say that 0.0975^p is the fraction of available distances. As p tends to infinity, this fraction tends to zero. That means as p increase, there are very few training observations if distance is defined as above.

On a side note, I dont think this is the way to put it. Since we need to consider training instances near to a test point on a whole considering all features. Lets say, being in 10% of each of the feature dont mean the observations as a whole are in 10% near to the test point. We need to define a nearby metrix considering all the features. That is absolute sum of all distances of individual feature distances should be in 10% of sum of ranges considered for all 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 on your answer.

For P =1 , hypercube is just a line with midpoint as testpoint. So, length = 0.1

For P =2, hypercube is a square with centre as testpoint, Area of square = s^2 = 0.1 , length = $\sqrt{0.1} = 0.316228$

For P=100, $s^{100} = 0.1$ length = $0.1^{\frac{1}{100}}$ = 0.9772

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 TP : True Positive, TN : True Negative, FP : False Positive, FN : False Negative
Accuracy is defined w.r.t both TP, TN = $\frac{TP+TN}{TP+TN+FP+FN} = \frac{8+974}{8+974+16+2} = \frac{982}{1000}$ = 0.982

(b) (2 pts) Precision Precision is defined w.r.t positives = $\frac{TP}{TP+FP} = \frac{8}{8+16} = \frac{8}{24}$ = 0.333

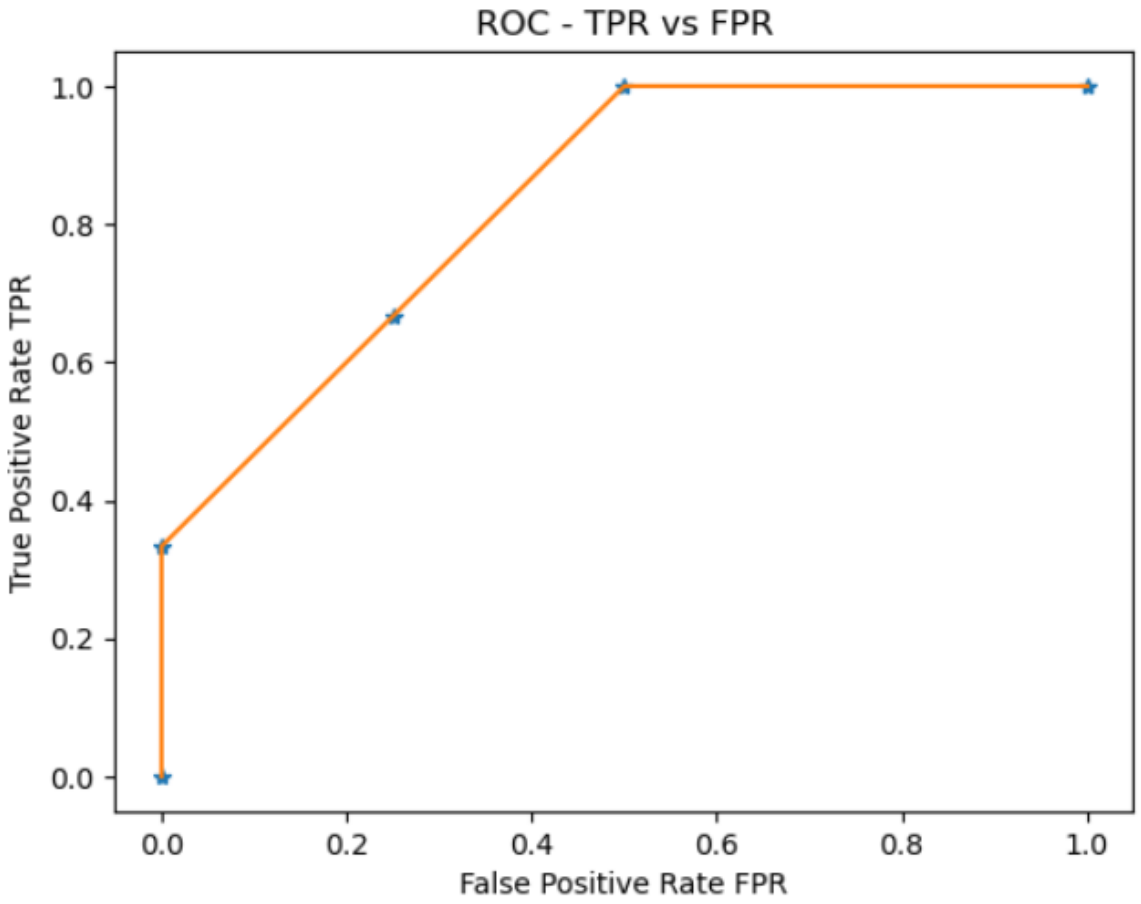
(c) (2 pts) Recall Recall = $\frac{TP}{TP+FN} = \frac{8}{8+2} = \frac{8}{10}$ = 0.8

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.

ROC curve is drawn between True Posivite Rate and False Positive Rate. Data points for the plot is collected at all transition points from TP to FP in the observation table.
plot points = [(TPR, FPR)] = [(0/6, 0/4), (2/6, 0/4), (4/6, 1/4), (6/6, 2/4), (6/6, 4/4)]



(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.

As seen from the ROC curve, TPR = 2/6 at FPR = 0/4 =0. That means for you to not get any FPR, you can the threshold as 0.85 which is confidence positive for TPR = 2/6. In real life, its ok to have false negatives(that is which are not TP). In this case, spams can still come. But its not ok to get important emails go into spam filter. That is FP!=0. Therefore I will choose threshold as 0.85

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

$\nabla_\theta L(f(x;\theta), y) = (-1) * ((\nabla_\theta y) * \log \hat{y} + y * (\nabla_\theta \log \hat{y}))$
$= 0 - (\frac{y}{\hat{y}}) * (\nabla_\theta \hat{y}) - (\frac{1-y}{1-\hat{y}}) * (\nabla(1-\hat{y})) - 0$
$\nabla_\theta \hat{y} = \nabla_\theta(\sigma(\theta^\top x)) = (\nabla_\theta(\frac{1}{1+\exp -\theta^\top x})) = (\frac{-1}{(1+\exp -\theta^\top x)^2}) * (\exp -\theta^\top x) * (-x)) = \frac{x \exp -\theta^\top x}{(1+\exp -\theta^\top x)^2}$
$= x * (\sigma(\theta^\top x)) * (1 - \sigma(\theta^\top x))$
$\nabla_\theta L(f(x;\theta), y) = -(\frac{y}{\hat{y}}) * (\nabla_\theta \hat{y}) - (\frac{1-y}{1-\hat{y}}) * (\nabla(1-\hat{y}))$
$= -(\frac{y}{\sigma(\theta^\top x)}) * x * (\sigma(\theta^\top x)) * (1 - \sigma(\theta^\top x)) - (\frac{1-y}{1-\sigma(\theta^\top x)}) * x * (\sigma(\theta^\top x)) * (1 - \sigma(\theta^\top x)) * (-1)$
$= -y * x * (1 - \sigma(\theta^\top x)) + (1-y) * x * (\sigma(\theta^\top x))$
$= -xy + xy * \sigma(\theta^\top x) + x * \sigma(\theta^\top x) - xy * \sigma(\theta^\top x) = -xy + x * \sigma(\theta^\top x)$
$= x * (\sigma(\theta^\top x) - y) = x * (f(x;\theta) - y)$

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

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

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

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

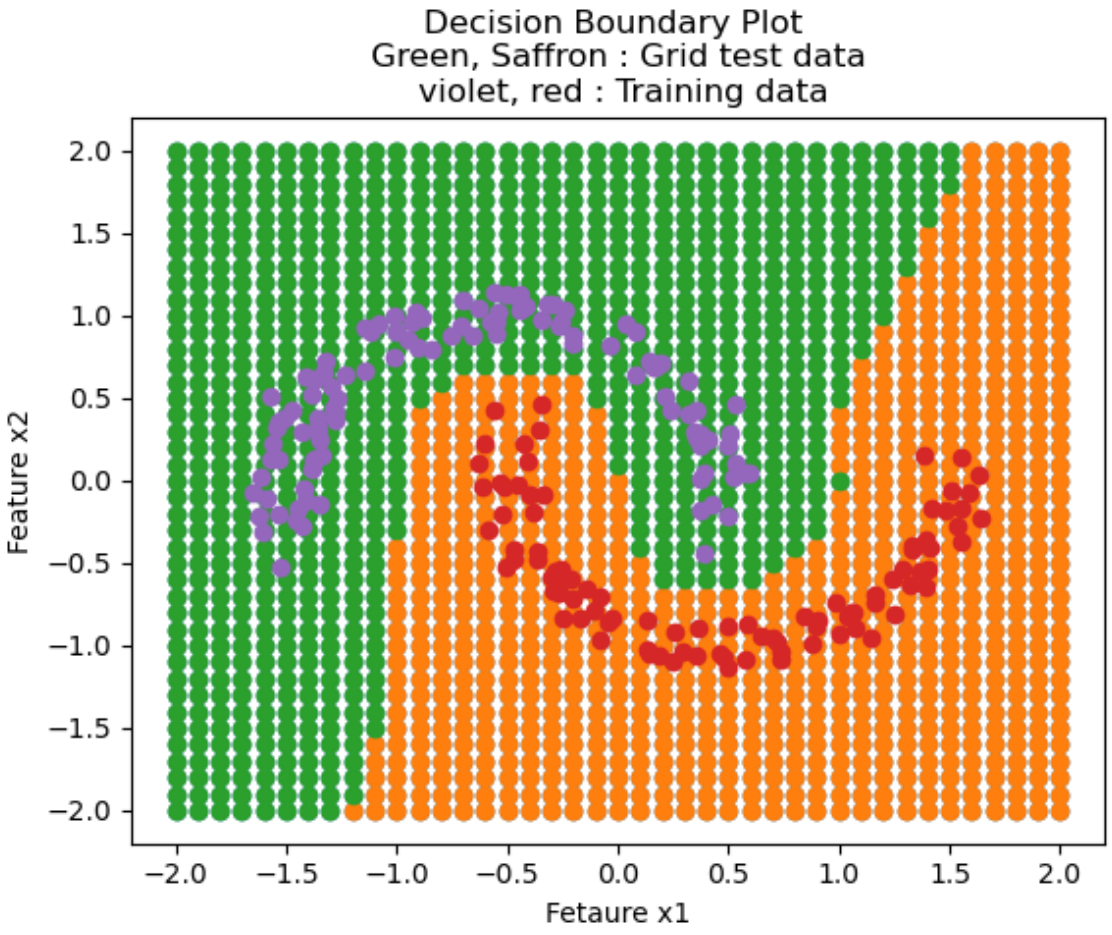
Compute the updated parameter vector θ^1 from the single update step.

For $\theta^0 = [0,0,0], \theta^\top x = 0$
$implies f(x;\theta) = \sigma(\theta^\top x) = (\frac{1}{1+\exp -\theta^\top x}) = \frac{1}{1+1} = 0.5$
$From above problem, gradient, \nabla_\theta L(f(x;\theta), y) = x * (f(x;\theta) - y) = x * (0.5 - 1) = -0.5 * x$
$\theta^{t+1} = \theta^t - \eta \nabla_\theta L(f(x;\theta), y) \theta^1 = \theta^0 - \eta \nabla_\theta L(f(x;\theta), y) = [0,0,0] - 0.1 * (-0.5 * x) = [0,0,0] + 0.05 * [1,3,2] = [0.05, 0.15, 0.1]$

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 expected figure looks like this.



Features

Label

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

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.

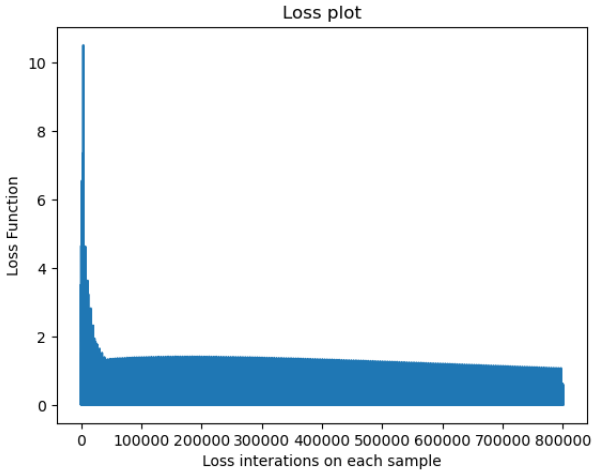
Implemented KNN from scratch.
$do\_5\_fold\_knn(1)$
1st Fold
$TP = 234$
$TN = 591$
$FP = 124$
$FN = 51$
$Recall = 0.8210526315789474$
$Precision = 0.6536312849162011$
$Accuracy = 0.825$
2nd Fold
$TP = 240$
$TN = 615$
$FP = 108$
$FN = 37$
$Recall = 0.8664259927797834$
$Precision = 0.6896551724137931$
$Accuracy = 0.855$
3rd Fold
$TP = 239$
$TN = 624$
$FP = 92$
$FN = 45$
$Recall = 0.8415492957746479$
$Precision = 0.7220543806646526$
$Accuracy = 0.863$

4th Fold
$TP = 241$
$TN = 613$
$FP = 93$
$FN = 53$
$Recall = 0.8197278911564626$
$Precision = 0.7215568862275449$
$Accuracy = 0.854$
5th Fold
$TP = 233$
$TN = 542$
$FP = 152$
$FN = 73$
$Recall = 0.761437908496732$
$Precision = 0.6051948051948052$
$Accuracy = 0.775$

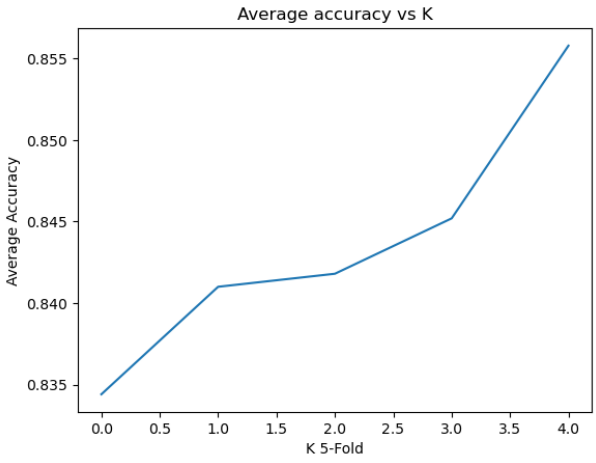
3. (12 pts) Implement logistic regression (from scratch). Use gradient descent (refer to question 6 from part 1) to find the optimal parameters. You may need to tune your learning rate to find a good optimum. Run 5-fold cross validation. Report accuracy, precision, and recall in each fold.

$learning\_rate = 0.005;$
$epochs = 200;$
$threshold = 0.5$
1st Fold
$TP = 279$
$TN = 663$
$FP = 52$
$FN = 6$
$Recall = 0.9789473684210527$
$Precision = 0.8429003021148036$
$Accuracy = 0.942$
2nd Fold
$TP = 273$
$TN = 669$
$FP = 54$
$FN = 4$
$Recall = 0.9855595667870036$
$Precision = 0.8348623853211009$
$Accuracy = 0.942$
3rd Fold
$TP = 275$
$TN = 681$
$FP = 35$
$FN = 9$
$Recall = 0.9683098591549296$
$Precision = 0.8870967741935484$
$Accuracy = 0.956$
4th Fold
$TP = 288$
$TN = 671$
$FP = 35$
$FN = 6$
$Recall = 0.9795918367346939$
$Precision = 0.891640866873065$
$Accuracy = 0.959$
5th Fold
$TP = 297$

$TN = 621$
$FP = 73$
$FN = 9$
$Recall = 0.9705882352941176$
$Precision = 0.8027027027027027$
$Accuracy = 0.918$



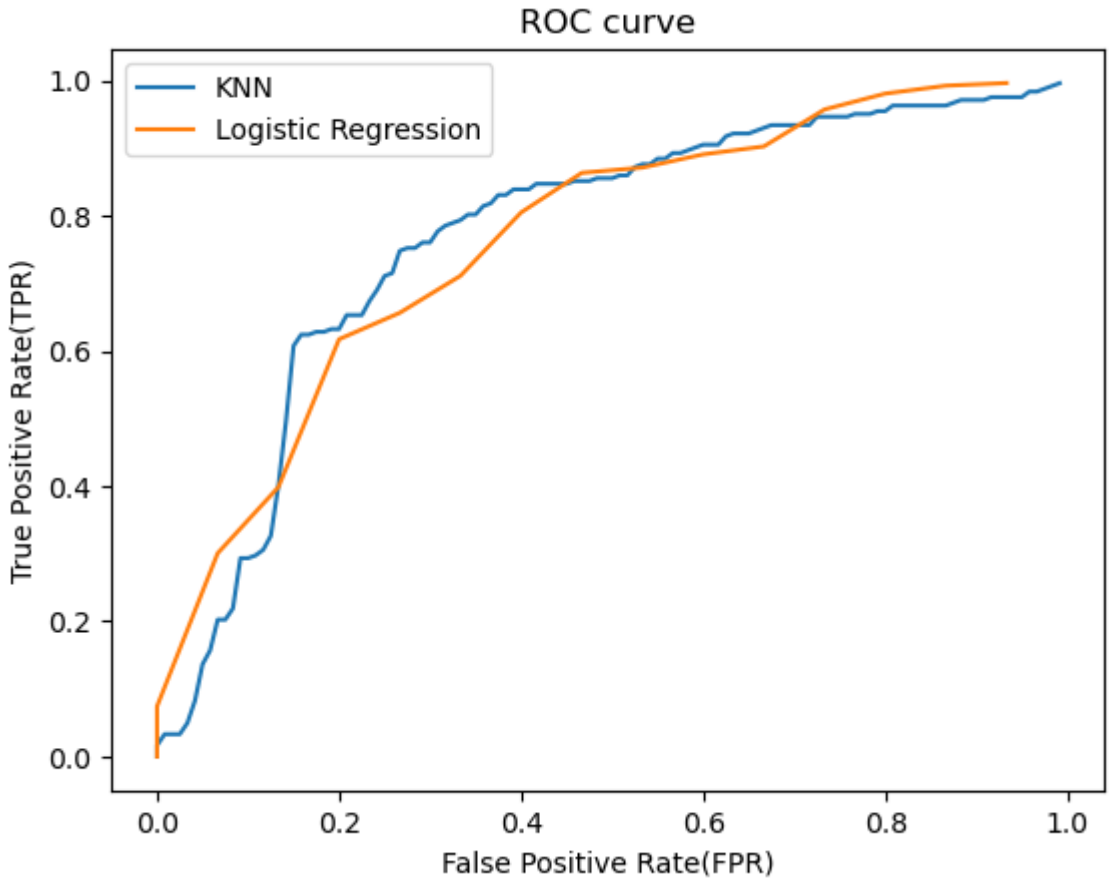
4. (10 pts) Run 5-fold cross validation with kNN varying k (k=1, 3, 5, 7, 10). Plot the average accuracy versus k, and list the average accuracy of each case.
Expected figure looks like this.



The above is computed by doing np.mean if all the 5 fold accuracies.

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.

Note that the logistic regression results may differ.


ROC curve
KNN
Logistic Regression
True Positive Rate(TPR)
False Positive Rate(FPR)
0.0
0.2
0.4
0.6
0.8
1.0