HOMEWORK 3

>>VISHAL V NAIK<<
>>9084602235<<

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 : https://github.com/vvnaik2/ece760.git

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.

   This is a 'Regression' problem as we are trying to predict CEO salary by understanding the relationship of CEO salary to the firm's profit, number of employees and industry

   n = 500, which is the number of firms the data was collected on

   p = 3, profit, number of employees, and industry

   (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 binary 'Classification' problem as we are trying to ascertain whether a new product will succeed or failure based on the price of the product, marketing budget, competition price, and ten other variables

   n = 20, which is the number of products for which the data is collected

   p = 13, price, marketing budget, competition price, and ten 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.

   This is a 'Regression' problem as we are trying to predict the percent change in US dollar by understanding it's relationship to weekly changes in the world stock markets.

n = 52, as there are 52 weeks in a year

p = 3, the percent change in US, German, and British markets

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, $X_1 = X_2 = X_3 = 0$.

Euclidean distance is computed using the formula: $d(X^1, X^2) = \sqrt{\Sigma_i (X_i^1 - X_i^2)^2}$

X_1	X_2	X_3	Y	$Distance$
0	3	0	Red	3
2	0	0	Red	2
0	1	3	Red	$\sqrt{10}$
0	1	2	Green	$\sqrt{5}$
-1	0	1	Green	$\sqrt{2}$
1	1	1	Red	$\sqrt{3}$

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

With K = 1, we predict the class by simply selecting the class of the nearest observation to the test point, which in this case is the observation with the least distance of $\sqrt{2}(X1 = 1, X2 = 0, X3 = 1)$ The class of this observation is Green, and thereby the prediction for the test point is 'Green'

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

With K = 3, our prediction now becomes 'Red', since 2 of the 3 nearest neighbors, with distances $\sqrt{2}$, $\sqrt{3}$ and 2 have class 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 all x ϵ [0.05, 0.95], the observations used for prediction lie in the interval [x - 0.05, x + 0.05]. This represents a length of 0.1, which is 10% of the available observations. If x $<$ 0.05, then we will use

observations in the interval [0, x + 0.05], which has a length of x + 0.05 and represents a fraction of (100x + 5)%. If x $>$ 0.95, then we will use observations in the interval [x - 0.05, 1], which has a length of 1.05 - x and represents a fraction of (105 - 100x)%. Thereby, the following expression is evaluated to compute the average fraction:

$\int_{0.05}^{0.95} 10dx + \int_{0}^{0.05}(100x + 5)dx + \int_{0.95}^{1}(105 - 100x)dx = 9 + 0.375 + 0.375 = 9.75$

So 9.75% is the average fraction of available observations used to make 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?

Assuming X1 and X2 are independent, the fraction of available observations we will use to make the prediction is 9.75% 9.75% = 0.950625

(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?

Assuming all features are independent of each other, the fraction of available observations used to make the prediction is $9.75\%^{100} \approx 0$

(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.

As can be seen from above, as the number of features p is large, the fraction of data points used for predicting the class of any test point reduces significantly. This fraction is given by the formula $9.75\%^p$. As p $\to \infty$, the fraction $(9.75\%)^p \to 0$.

(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, we have l = 0.1, for p = 2, we have $l = 0.1^{1/2}$, and similarly for p = 100, we have $l = 0.1^{1/100}$. This also shows that the fraction/ volume of the available data points/observation space reduces significantly as the number of features increases. This directly impacts the performance of kNN and is known as the curse of dimensionality.

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
True Positive (TP) = 8
True Negative (TN) = 974
False Positive (FP) = 16
False Negative (FN) = 2
Accuracy = (TP+TN)/(TP+TN+FP+FN) = 982/1000 = 98.2%

(b) (2 pts) Precision
Precision = TP/(TP+FP) = 8/24 = 33.33%

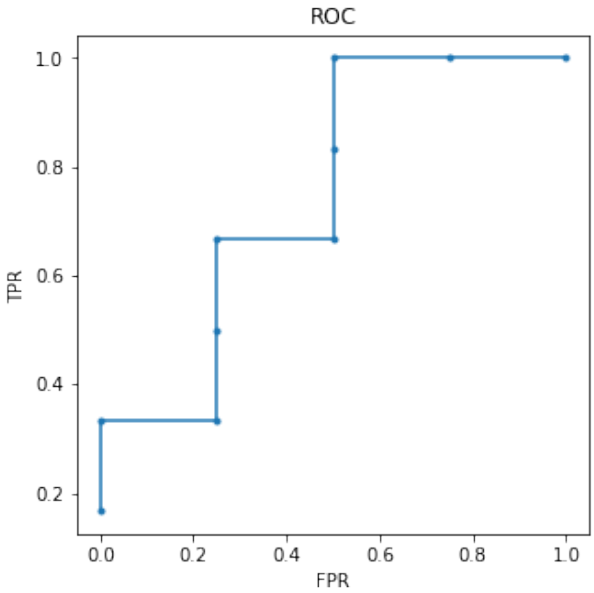
(c) (2 pts) Recall
Recall = TP/(TP+FN) = 8/10 = 80%

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.

Confidence positive	Correct class	TPR	FPR	Distance from ideal point(TPR=1 , FPR=0)
0.95	+	1/6	0	0.833
0.85	+	2/6	0	0.667
0.8	-	2/6	1/4	0.712
0.7	+	3/6	1/4	0.559
0.55	+	4/6	1/4	0.417
0.45	-	4/6	2/4	0.601
0.4	+	5/6	2/4	0.527
0.3	+	1	2/4	0.5
0.2	-	1	3/4	0.75
0.1	-	1	1	1



(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 threshold parameter must be chosen such that we maximize TPR but also minimize FPR simulatenously. This point corresponds to the point on the ROC curve closest to the the 'ideal point' which is given by TPR = 1.0 and FPR = 0.0. This corresponds to the point with confidence positive of 0.55 since it has the closest distance of 0.417 from the ideal point

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 ,$\nabla_\theta L(f(x;\theta), y) = x * (f(x;\theta) - y)$

$= x * (0.5 - 1) = -0.5 * x$

$\theta^{t+1} = \theta^t - \eta \nabla_\theta L(f(x;\theta), y)$

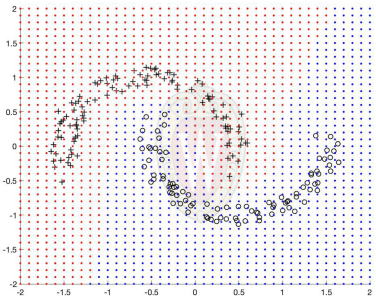
$\theta^1 = \theta^0 - \eta \nabla_\theta L(f(x;\theta), y)$

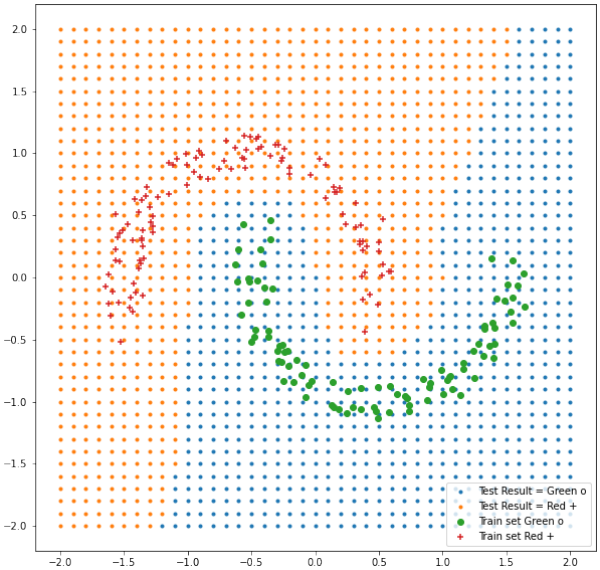
$= [0, 0, 0] - 0.1 * (-0.5 * x)$
$= [0, 0, 0] + 0.05 * [1, 3, 2]$
$= [0.05, 0.15, 0.1]$

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.





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)

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

- 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

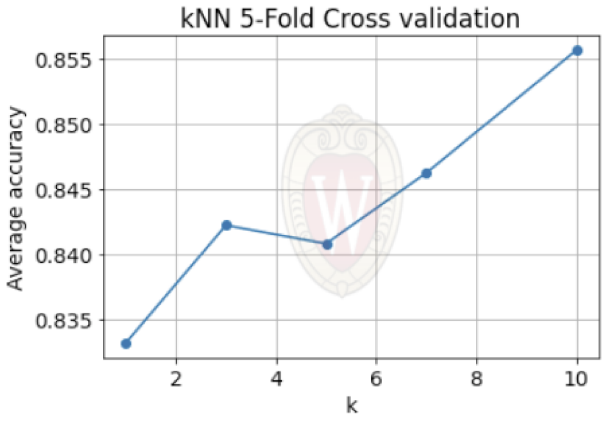
2. (8 pts) Implement 1NN, Run 5-fold cross validation. Report accuracy, precision, and recall in each fold.

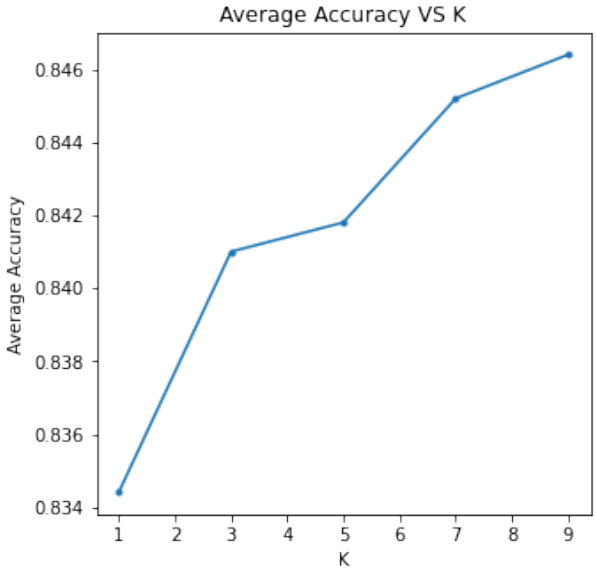
	Fold	Accuracy	Precision	Recall
0	1	0.825	0.653631	0.821053
1	2	0.855	0.689655	0.841549
2	3	0.863	0.722054	0.841549
3	4	0.854	0.721557	0.819728
4	5	0.775	0.605195	0.761438

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.

```
Testing Complete
Learning rate        : 0.01
Number of iterations : 1000
Fold 1 accuracy  : 0.94
Fold 1 precision : 0.8947368421052632
Fold 1 recall    : 0.8947368421052632
Fold 2 accuracy  : 0.967
Fold 2 precision : 0.9236111111111112
Fold 2 recall    : 0.9602888086642599
Fold 3 accuracy  : 0.934
Fold 3 precision : 0.92578125
Fold 3 recall    : 0.8345070422535211
Fold 4 accuracy  : 0.944
Fold 4 precision : 0.8940397350993378
Fold 4 recall    : 0.9183673469387755
Fold 5 accuracy  : 0.919
Fold 5 precision : 0.8299120234604106
Fold 5 recall    : 0.9248366013071896
```

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.

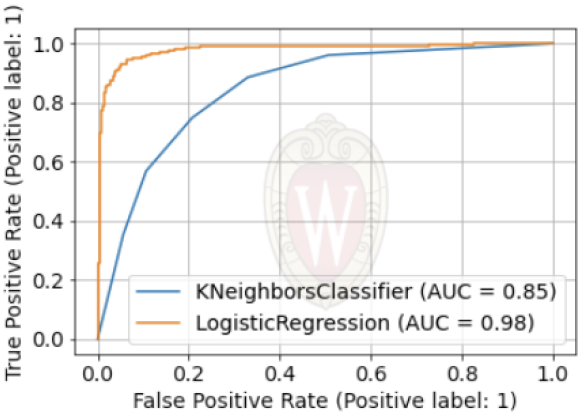


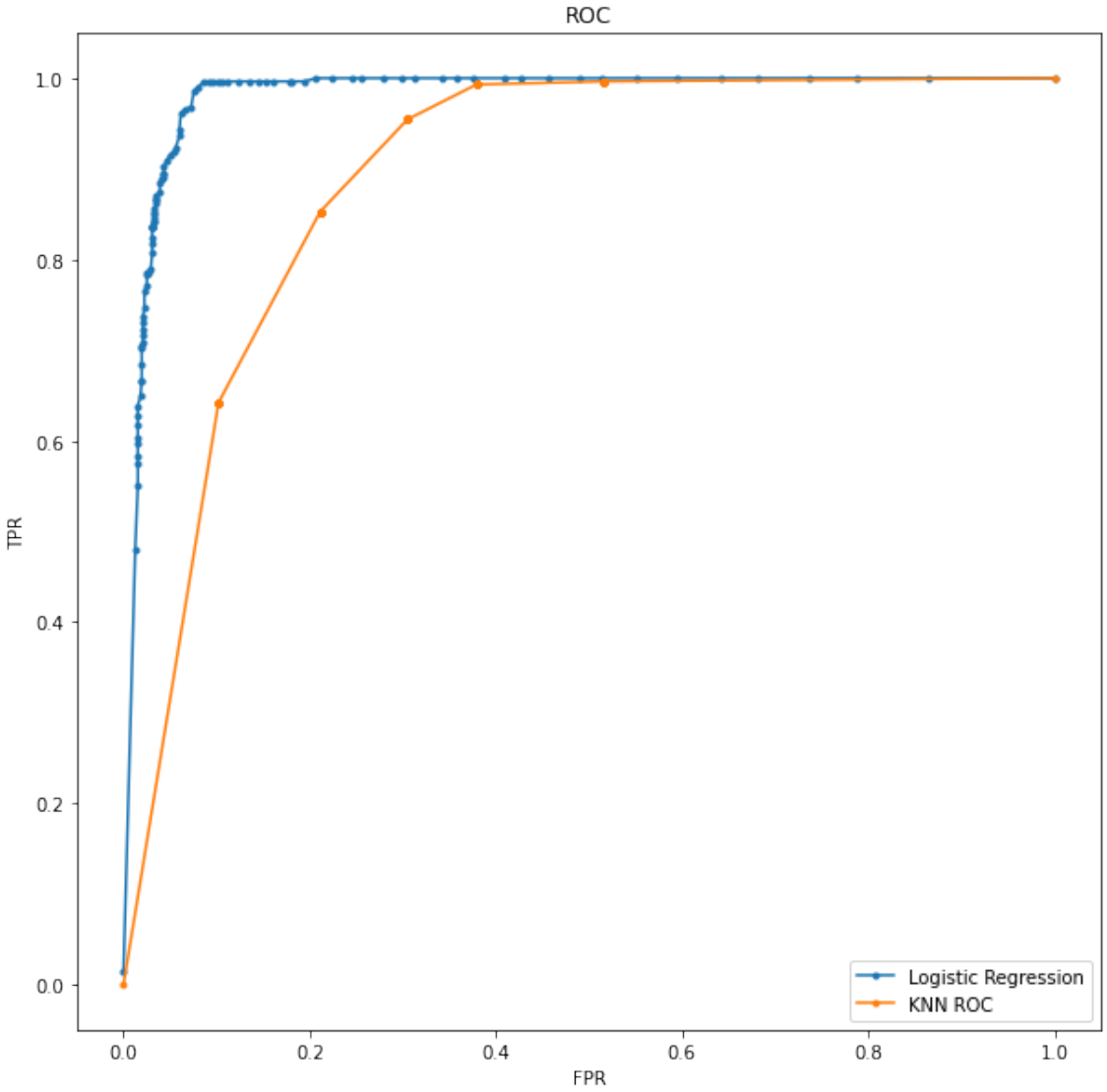


	K	Accuracy1	Accuracy2	Accuracy3	Accuracy4	Accuracy5
0	1	0.825	0.855	0.863	0.854	0.775
1	3	0.846	0.850	0.856	0.880	0.773
2	5	0.837	0.852	0.871	0.869	0.780
3	7	0.837	0.861	0.875	0.874	0.779
4	9	0.842	0.858	0.871	0.882	0.779

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.

   The ROC curves - KNN vs Logistic Regression is as shown


True Positive Rate (Positive label: 1)
False Positive Rate (Positive label: 1)
KNeighborsClassifier (AUC = 0.85)
LogisticRegression (AUC = 0.98)


ROC
TPR
FPR
Logistic Regression
KNN ROC