HOMEWORK 3

Hailey Park
hpark353@wisc.edu
https://github.com/hparkailey/760ML.git

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.

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, since the CEO salary, the dependent variable we are predicting is a numerical value without bounds. $n = 500$ and $p = 3$.

   (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, since we are predicting binary outcome variables, whether a new product will be a success or not. $n = 20$ and $p = 13$.

   (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.

   This is a regression problem, since the percent change in the US dollar in relation to weekly changes in the stock markets is a numerical value without bounds. Since there are 52 weeks in a year, $n = 52$ and $p = 3$.

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, $X_1 = X_2 = X_3 = 0$.

   Let Y_i be each of the training instance and Z be the test point where $X_1 = X_2 = X_3 = 0$. The Euclidean distance between Y_i and Z are as follows.

i	d(X_i,Z)
0	9
1	4
2	10
3	5
4	2
5	3

Table 1: Euclidean distance between X_i and Z

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

Green. Since we only consider one nearest neighbor, the Y label associated with X_4 is Green and we return Green as our prediction.

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

Red. We now consider three nearest neighbors, which are X_1, X_4, X_5, and taking the majority vote of their labels, we return 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?

10 %

(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?

$0.1 * 0.1 = 0.01$, so 1%

(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?

0.1^{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.

From parts (a)-(c), we can see that as the number of features (p) increases to infinty, the fraction of the available training set observations approaches 0 as in $\lim_{p\to\infty} 0.1^p = 0$. Thus, the drawback of KNN with large p is that there would only be a very few training observations near the given test observation.

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

For $p = 1$, the length would be 0.1, for $p = 2$, it would be $0.1^{\frac{1}{2}}$, and for $p = 100$, it would be $0.1^{\frac{1}{100}}$. $\lim_{p\to\infty} 0.1^{\frac{1}{p}} = 1$ so the length would approach 1 as p 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 $\frac{8+974}{8+16+2+974} = 0.982$

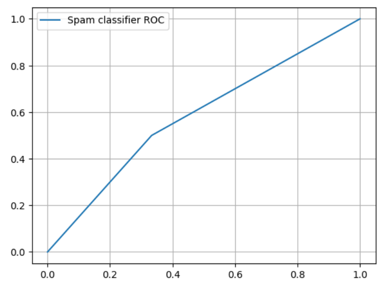
(b) (2 pts) Precision $\frac{8}{24}$

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

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.



(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.

Here, FP would be to misclassify non-spam emails as spams and TP would be to correctly identifying spam emails as spams. The cost associated with FP would be that we can potentially miss important non-spam emails and we would want to avoid this than focusing on correctly classifying spam emails. So the threshold 0.3 is where we can minimize the FPR while maintaining high TPR would be idealisstic in the ROC curve.

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

$$\begin{aligned}
\frac{\partial L(\hat{y}, y)}{\partial \theta} &= -(\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}})\frac{\partial \hat{y}}{\partial \theta} \\
&= -(\frac{y}{\sigma(\theta^\top x)} - \frac{1-y}{(1-\sigma(\theta^\top x))})\sigma(\theta^\top x))(1-\sigma(\theta^\top x)))x \\
&= -(\frac{y(1-\sigma(\theta^\top x))) - (1-y) * \sigma(\theta^\top x))}{\sigma(\theta^\top x)) * (1-\sigma(\theta^\top x)))}) * \sigma(\theta^\top x)) * (1-\sigma(\theta^\top x))x \\
&= -(y - \sigma(\theta^\top x)))x
\end{aligned}$$

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

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

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

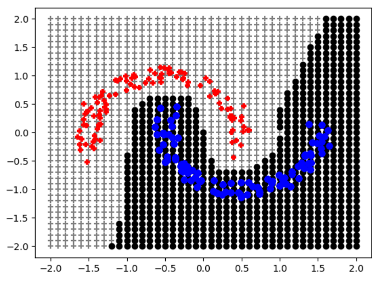
$$\text{data example : } x = [1, 3, 2], y = 1$$

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

$$\begin{aligned}
\theta^1 &= \theta^0 - \eta \nabla_\theta L(f(x;\theta), y) \\
&= [0, 0, 0] - 0.1 * (-(1 - \frac{1}{2})) * [1, 3, 2] \\
&= \frac{1}{20}[1, 3, 2]
\end{aligned}$$

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



Spam filter Now, we will use 'emails.csv' as our dataset.

- 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.

   Fold 0 Accuracy: 0.82 Recall: 0.82 Precision: 0.65

   Fold 1 Accuracy: 0.85 Recall: 0.87 Precision: 0.69

   Fold 2 Accuracy: 0.86 Recall: 0.84 Precision: 0.72

   Fold 3 Accuracy: 0.85 Recall: 0.82 Precision: 0.72

   Fold 4 Accuracy: 0.78 Recall: 0.76 Precision: 0.61

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.

   Fold 0 Accuracy: 0.76 Recall: 0.30 Precision: 0.67

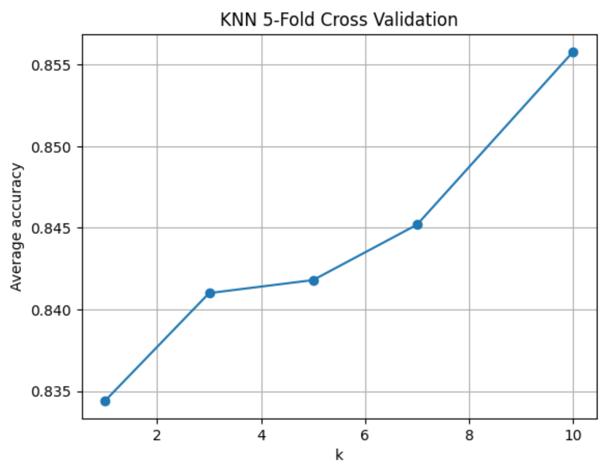
   Fold 1 Accuracy: 0.73 Recall: 0.13 Precision: 0.56

   Fold 2 Accuracy: 0.76 Recall: 0.29 Precision: 0.67

   Fold 3 Accuracy: 0.73 Recall: 0.21 Precision: 0.66

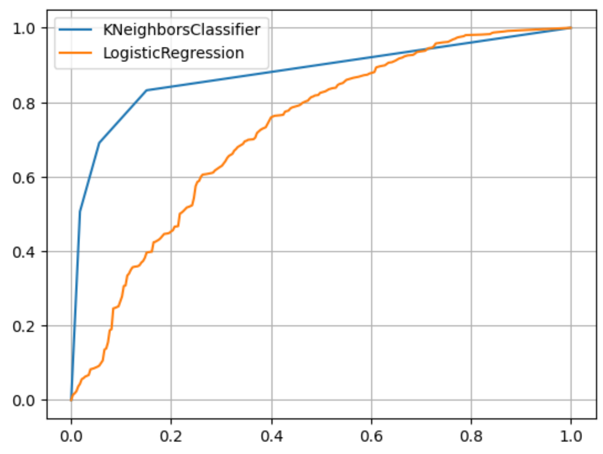
   Fold 4 Accuracy: 0.66 Recall: 0.74 Precision: 0.46

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.



   Average accuracies corresponding to each k are [0.8344, 0.841, 0.8418, 0.8452, 0.8558].

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.


KNeighborsClassifier
LogisticRegression
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0