HOMEWORK 3

Dario Placencio
908 284 6018

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 Repo: https://github.com/placenciohid/ECE760-Homework/blob/a07d70fbccf8c85da945eb97a608062c0fa8b51a/Homework%203/Homework%203%20-%20Dario%20Placencio.ipynb

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.

   Given we have 500 companies, we have 500 data points. These come with 4 features each: (1) Profit, (2) of Employyees, (3) Industry, so 3 variables. Also the label/target variable of the CEO Salary is present. The problem represents a Regression, since the predictions aims for an particular amount.

   (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.

   Here, we have 20 data points, with 13 features, and a problem that has a classification nature, since the intend is to predict is either success or failure, being a binary classification.

   (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.

   For this one, the data points are 54 (weeks in a year), the features are 4: USD, USD Market, British, German. And the problem has a regression nature, since the prediction aims towards a percentage change, looking the value itself.

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 calculated Euclidean distances from each point to the point of interest are as follows:

X_1	X_2	X_3	Distance
0	3	0	$\sqrt{9} = 3$
2	0	0	$\sqrt{4} = 2$
0	1	3	$\sqrt{10} \approx 3.16$
0	1	2	$\sqrt{5} \approx 2.24$
-1	0	1	$\sqrt{2} \approx 1.41$
1	1	1	$\sqrt{3} \approx 1.73$

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

The prediction with K = 1 is Green because the closest observation is Green, with a distance of 1.41.

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

The prediction with K = 3 would be Red, from the 3 closest observations, 2 are Red and 1 is Green. The vote would be Red, hence the prediction. The values of these distances are 1.41, 1.73, and 2.00.

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?

X is uniformly distributed on [0,1].
We will predict the response for a test observation using the range $[x - 0.05, x + 0.05]$
Given X is uniformly distributed on [0,1], the probability density function (pdf) is:

$$f(x) = \begin{cases} 1 & \text{if } 0 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

Now, calculating the probability of the range the observation within the 10

$$\begin{aligned} P(x - 0.05 \leq X \leq x + 0.05) &= \int_{x-0.05}^{x+0.05} f(x)dx \\ &= \int_{x-0.05}^{x+0.05} 1dx \\ &= x + 0.05 - (x - 0.05) \\ &= 0.1 \end{aligned}$$

So, the probability of the range the observation within the 10 percent range is 0.1.

(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?

Both X1, X2 are uniformly distributed on [0,1].
We will predict the response for a test observation using the range $[x_1 - 0.05, x_1 + 0.05]$ for X1 and $[x_2 - 0.05, x_2 + 0.05]$ for X2
Given X1, X2 are uniformly distributed on [0,1], the probability density function (pdf) is:

$$f(x_1, x_2) = \begin{cases} 1 & \text{if } 0 \leq x_1, x_2 \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

Now, calculating the probability of the range x1 [0.55, 0.65] and x2 [0.3, 0.4]:

$$\begin{aligned} P(0.6 - 0.05 \leq X_1 \leq 0.6 + 0.05, 0.35 - 0.05 \leq X_2 \leq 0.35 + 0.05) &= \int_{0.55}^{0.65} \int_{0.3}^{0.4} f(x_1, x_2) dx_1 dx_2 \\ &= \int_{0.55}^{0.65} \int_{0.3}^{0.4} 1 dx_1 dx_2 \\ &= \int_{0.55}^{0.65} x_1 \Big|_{0.3}^{0.4} dx_2 \\ &= \int_{0.55}^{0.65} 0.1 dx_2 \\ &= 0.1 \int_{0.55}^{0.65} dx_2 \\ &= 0.1 x_2 \Big|_{0.55}^{0.65} \\ &= 0.1(0.65 - 0.55) \\ &= 0.01 \end{aligned}$$

We will use 1 percent of the available observations to make the prediction when considering both features, with a 10 percent range.

(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 observations 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?

So, from the previous answer we know that:

For p=1, we use 10 percent or 0.1 of the observations.
For p=2, we use 10 percent or 0.01 of the observations.

These values are the product of the fractions of observations used in each individual feature's dimension. This is because, given independence and uniform distribution, the fraction of observations used in a multi-dimensional space is the product of the probabilities in each dimension.

Extending this to 100 dimensions, we get:

$$(0.1)^p = 1.0 \times 10^{-100}$$

So we will use 1.0×10^{-100} of the available observations to make the prediction when considering 100 features, with a 10 percent range.

(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.

This is one of the challanges impose by the "curse of dimensionality", as the number of features (or dimensions) increases, so does the distance between the data points. KNN is approach that leverages on spatial proximity to make predictions, so as the distance between the data points increases, the accuracy of the predictions decreases.

Even on data sets with a large number of observations, if the number of features is large, is unlikely that the data points will be close to each other, because the distance is a exponential function of the number of 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 what happens to the length of the sides as $\lim_{p \to \infty}$.

In order to calculate the lenth of the hypercube on each one of the p values, let's first define the volume of the hypercube as:

$$V = L^p$$

Where L is the length of each side of the hypercube and p is the number of dimensions.
We know that percentage of observations used in each individual feature's dimension is 0.1, so:

$$L^p = 0.1$$
$$L = 0.1^{\frac{1}{p}}$$

Replacing the values of 1,2, and 100 for p, we get:

$$L_{p=1} = 0.1^{\frac{1}{1}} = 0.1$$
$$L_{p=2} = 0.1^{\frac{1}{2}} = 0.316$$
$$L_{p=100} = 0.1^{\frac{1}{100}} = 0.977$$

This shows that as the value of features increases, so does the length of the sides of the hypercube, getting closer to 1, representing the entire range of the feature. This means that with more features, we progresively need to consider a larger range of values to make predictions, making the "neighborhood" of the KNN model larger, and therefore less accurate.

4. (6 pts) Suppose 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

$$\begin{aligned} Accuracy &= \frac{TP + TN}{TP + TN + FP + FN} \\ &= \frac{8 + 974}{8 + 974 + 16 + 2} \\ &= \frac{982}{1000} \\ &= 0.982 \end{aligned}$$

(b) (2 pts) Precision

$$\begin{aligned} Precision &= \frac{TP}{TP + FP} \\ &= \frac{8}{8 + 16} \\ &= \frac{8}{24} \\ &= 0.333 \end{aligned}$$

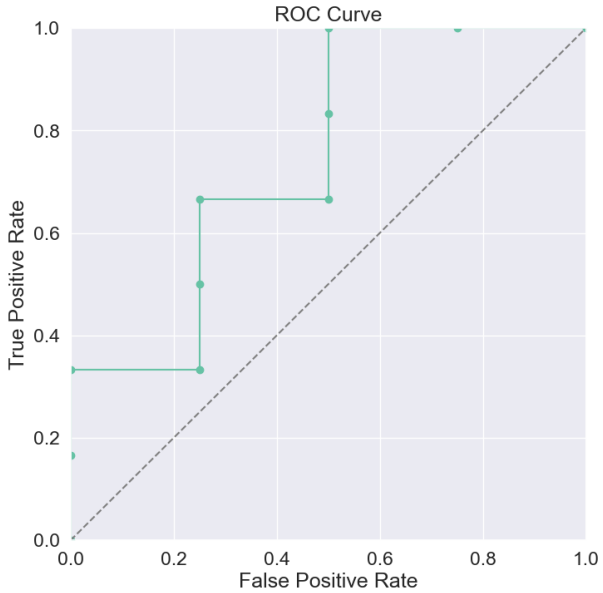
(c) (2 pts) Recall

$$\begin{aligned} Recall &= \frac{TP}{TP + FN} \\ &= \frac{8}{8 + 2} \\ &= \frac{8}{10} \\ &= 0.8 \end{aligned}$$

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.

Considering the context, what I would like to avoid is to have False Positives, emails that are actually not Spam, and that I would be missing from my inbox because of missclassification.
So, I would choose a threshold that would minimize the False Positive Rate (FPR), based on the values of the ROC curve, this treshold would be .85, which would give me a FPR of 0, but in comparison a little bit of a better classification of the True Positives (TPR) than the threshold of 0.95.

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

The first gradient of the loss function is:

$$
\begin{aligned}
\nabla_\theta L(f(x;\theta),y) &= \frac{\partial}{\partial\theta} L(f(x;\theta),y) \\
&= \frac{\partial}{\partial\theta} - [y\log\hat{y} + (1-y)\log(1-\hat{y})] \\
&= -[y\frac{\partial}{\partial\theta}\log\hat{y} + (1-y)\frac{\partial}{\partial\theta}\log(1-\hat{y})] \\
&= -[y\frac{\partial}{\partial\theta}\log\sigma(\theta^\top x) + (1-y)\frac{\partial}{\partial\theta}\log(1-\sigma(\theta^\top x))] \\
&= -[y\frac{1}{\sigma(\theta^\top x)}\frac{\partial}{\partial\theta}\sigma(\theta^\top x) + (1-y)\frac{1}{1-\sigma(\theta^\top x)}\frac{\partial}{\partial\theta}\log(1-\sigma(\theta^\top x))] \\
&= -[y\frac{1}{\sigma(\theta^\top x)}\sigma(\theta^\top x)(1-\sigma(\theta^\top x))x + (1-y)\frac{1}{1-\sigma(\theta^\top x)}(-\sigma(\theta^\top x)(1-\sigma(\theta^\top x))x)] \\
&= -[y(1-\sigma(\theta^\top x))x + (1-y)(-\sigma(\theta^\top x)x)] \\
&= -[yx - y\sigma(\theta^\top x)x - \sigma(\theta^\top x)x + y\sigma(\theta^\top x)x] \\
&= -[yx - \sigma(\theta^\top x)x] \\
&= -[y - \sigma(\theta^\top x)]x \\
&= -[y - f(x;\theta)]x
\end{aligned}
$$

(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.
We know that:

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

So, we need to first calculate $\hat{y}$ - $f(x;\theta)$. We know that:

$$z = \theta^\top x$$

So, for the given data example, we have:

$$
\begin{aligned}
z &= \theta^\top x \\
&= [0,0,0]^\top [1,3,2] \\
&= 0
\end{aligned}
$$

And, we know that:

$$
\begin{aligned}
\hat{y} &= \sigma(z) \\
&= \sigma(0) \\
&= frac11 + e^{-0} \\
&= 0.5
\end{aligned}
$$

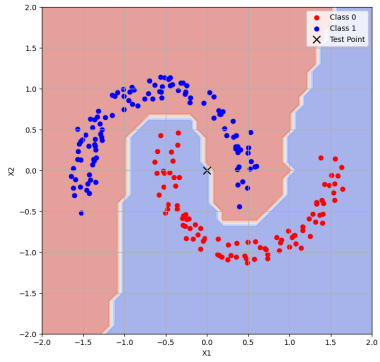
Now, we can calculate the updated parameter vector θ^1:

$$
\begin{aligned}
\theta^1 &= \theta^0 - \eta\nabla_\theta L(f(x;\theta),y) \\
&= [0,0,0] - 0.1(-[y - f(x;\theta)]x) \\
&= [0,0,0] - 0.1(-[1-0.5][1,3,2]) \\
&= [0,0,0] - 0.1(-[0.5][1,3,2]) \\
&= [0,0,0] - 0.1(-[0.5,1.5,1]) \\
&= [0,0,0] - [-0.05,-0.15,-0.1] \\
&= [0.05,0.15,0.1]
\end{aligned}
$$

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

   Answer



   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 performance metrics across the 5 folds are as follows:

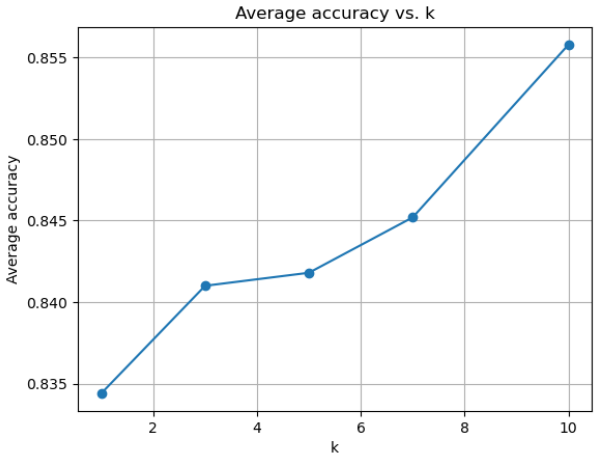
Fold	Accuracy	Precision	Recall
1	0.8250	0.6545	0.8175
2	0.8530	0.6857	0.8664
3	0.8620	0.7212	0.8380
4	0.8510	0.7164	0.8163
5	0.7750	0.6057	0.7582

   Table 1: Performance Metrics for 5-Fold Cross Validation

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	Alpha	Accuracy	Precision	Recall
1	0.001	0.918	0.8859	0.8175
2	0.001	0.930	0.9027	0.8375
3	0.001	0.930	0.9315	0.8134
4	0.001	0.954	0.9366	0.9048
5	0.001	0.941	0.8997	0.9085

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.



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.