HOMEWORK 3

>>NAME HERE<<
>>ID HERE<<

Instructions: Use this latex file as a template to develop your homework. Submit your homework on time as a single zip (containing the pdf and the code) file to Canvas. Please, check Piazza for updates about the homework. It is recommended that you use Python for your solutions. You are allowed to use the libraries `numpy`, `matplotlib` and `pandas`.

1 Questions (60 pts)

1. (9 pts) Explain whether each scenario is a classification or regression problem. Additionally, 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.
   Solution goes here.

   (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.
   Solution goes here.

   (c) (3 pts) We are interested 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.
   Solution goes here.

2. (6 pts) The table below provides a training dataset 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 dataset 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, $X_1 = X_2 = X_3 = 0$. Solution goes here.

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

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

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) (2 pts) 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 the response of a test observation 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?

   Solution goes here.

   (b) (2 pts) Now suppose that we have a set of observations, each with measurements on $p = 2$ features, $X1$ and $X2$. We assume that $(X1, X2)$ are uniformly (evenly) distributed on $[0, 1] \times [0, 1]$. Associated with each observation is a response value. Suppose that we wish to predict the response of a test observation using only observations that 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?

   Solution goes here.

   (c) (2 pts) 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 the response of a test observation 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?

   Solution goes here.

   (d) (3 pts) 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.

   Solution goes here.

   (e) (3 pts) 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.

   Solution goes here.

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 Solution goes here.

   (b) (2 pts) Precision Solution goes here.

   (c) (2 pts) Recall Solution goes here.

5. (9 pts) Again, suppose that 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 an ROC curve based on the above table.
Solution goes here.

(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.
Solution goes here.

6. (8 pts) In this problem, we will walk through a single step of the gradient descent algorithm for logistic regression. Assume two-dimensional input. Recap:

$$f(\mathbf{x}; \mathbf{v}, b) = \sigma(\mathbf{v} \cdot \mathbf{x} + b)\ ,$$

where σ is the logistic function discussed in class.

$$\text{Cross-entropy loss: } L(\hat{y}, y) = -[y \log \hat{y} + (1-y)\log(1-\hat{y})]$$

$$\text{The single update step: } \mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla_{\mathbf{w}} L(f(\mathbf{x}; \mathbf{w}), y), \text{ where } \mathbf{w} = [\mathbf{v}_1, \mathbf{v}_2, b]^T$$

Now given

$$\text{Initial parameters : } \mathbf{v}_1 = b = 0, \mathbf{v}_2 = 1, (\Rightarrow \mathbf{w}^{(0)} = [0, 1, 0]))$$

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

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

(a) (4 pts) Compute the first gradient $\nabla_{\mathbf{w}} L(f(\mathbf{x}; \mathbf{w}), y)$.
Solution goes here.

(b) (4 pts) Compute the updated parameter vector $\mathbf{w}^{(1)}$ from a single update step.
Solution goes here.

7. (10 pts) In this problem, we consider a variant of linear regression with sparse structure. The only difference with standard linear regression is that the hidden regressor vector has at most k non-zero parameters, where the sparsity k is much smaller than the dimension d. For example, for $k = 2$ and $d = 4$, the following parameters are feasible solutions: $\mathbf{w} = [0, 0, 0, 0]$, $\mathbf{w} = [0, 0, 0, 7]$, $\mathbf{w} = [0, 14, 0, 21]$, while $\mathbf{w} = [1, 5, 2, 7]$ and $\mathbf{w} = [1, 0, 5, 8]$ are not.

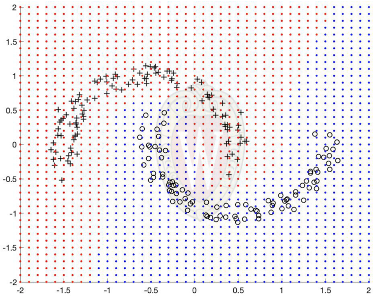
(a) (3 pt) Define the optimization problem for sparse linear regression, i.e., that of finding a sparse vector minimizing the square loss.
Solution goes here.

(b) (7 pts) Can we use (Stochastic) Gradient Descent to minimize this objective? Justify your answer.
Solution goes here.

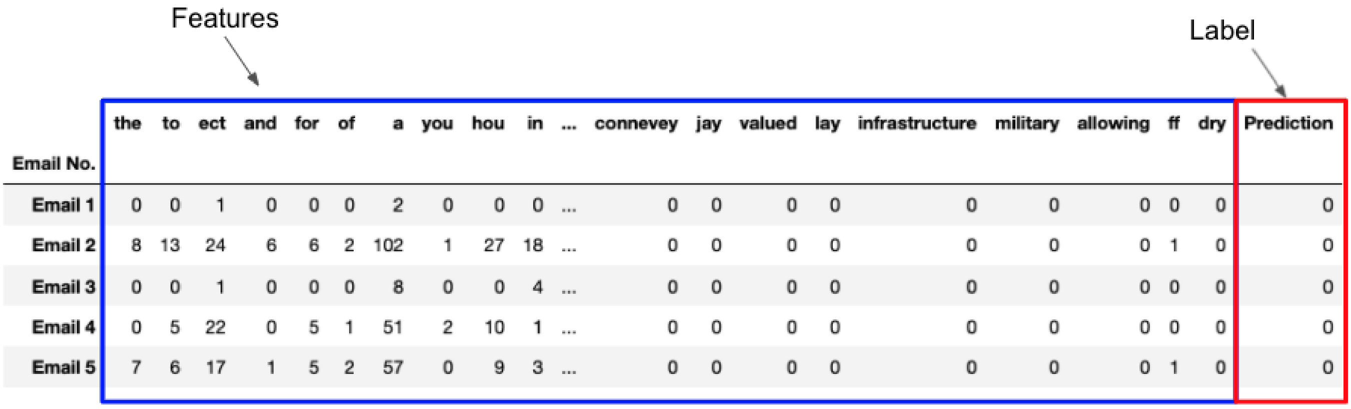
2 Programming (50 pts)

1. (10 pts) Use the entire `D2z.txt` as a training set. Use Euclidean distance (i.e., $\mathbf{A} = \mathbf{I}$). Visualize the predictions of 1NN on a 2D grid $[-2 : 0.1 : 2] \times [-2 : 0.1 : 2]$. 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 that 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.


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

- 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. (10 pts) Implement 1NN and run a 5-fold cross-validation. Report accuracy, precision, and recall in each fold.

   Solution goes here.

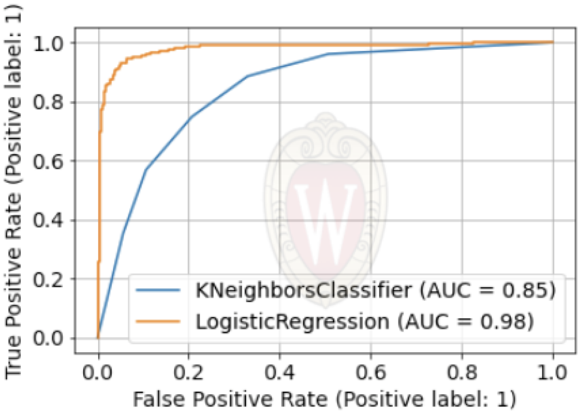
3. (10 pts) Implement logistic regression and run a 5-fold cross-validation. Report accuracy, precision, and recall in each fold.

   Solution goes here.

4. (10 pts) Run a 5-fold cross-validation with kNN varying k ($k = 1, 3, 5, 7$). Plot the average accuracy versus k, and list the average accuracy of each case. Solution goes here.

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 results may differ.

   Solution goes here.


True Positive Rate (Positive label: 1)
False Positive Rate (Positive label: 1)
0.0
0.2
0.4
0.6
0.8
1.0
KNeighborsClassifier (AUC = 0.85)
LogisticRegression (AUC = 0.98)