Homework Assignment 1

CSE 251A: ML - Learning Algorithms

Due: April 11th, 2023, 9:30am (Pacific Time)

Instructions: Please answer the questions below, attach your code in the document, and insert figures to create **a single PDF file**. You may search information online but you will need to write code/find solutions to answer the questions yourself.

Grade: ____ out of 100 points

1 (10 points) Classification vs. Clustering

In this question, you are provided with several scenarios. You need to identify if the given scenario is better formulated as a *classification* task or a *clustering* task. You should also provide the reason that supports your choice.

1. Scenario 1: Assume there are 100 graded answer sheets for a homework assignment (scores range from 0 to 100). We would like to split them into several groups where each group has similar scores.

   Choice: clustering task

   Reason: we are grouping grades based off shared characteristics rather than explicitely defined groups.

2. Scenario 2: Assume there are 100 graded answer sheets for a homework assignment (scores range from 0 to 100). We would like to split them into several groups where each group represents a letter grade (A, B, C, D) following the criteria: A (90-100), B (75-90), C (60-75), D (0-60).

   Choice: classification task

   Reason: We are dividing up scores based on specific values and each group has a specific order such as numeric grades that a student grade can fall into ie. 80-90. 91 is similar to 90 but would not be classified under the 80-90 group.

2 (40 points) Basic Calculus

2.1 (20 points) Derivatives with Scalars

1. $f(x) = \frac{1}{2}(ax - b)^2$ where $a, b \in \mathbb{R}$ are constant scalars, derive $\frac{\partial f(x)}{\partial x}$.

$f(x) = \frac{(ax-b)^2}{2}$ $\quad f(x) = \frac{x}{2}^2$ $\quad f' = \frac{2x}{2} = x$

$g(x) = ax - b \quad g' = a$

$\frac{2(ax-b)}{2} \cdot a$

$a(ax-b)$ $\quad a^2x - ab$

2. $f(x) = \ln(1 + e^x)$, derive $\frac{\partial f(x)}{\partial x}$.

$f(x) = \ln(1+e^x)$ $\quad f(x) = \ln(x) \quad f'(x) = \frac{1}{x}$

$g(x) = 1 + e^x \quad g'(x) = e^x$

$\frac{1}{1+e^x} \cdot e^x = \frac{e^x}{1+e^x}$

2.2 (20 points) Derivatives with Vectors

Several particular vector derivatives are useful for this course. For matrix $\mathbf{A} \in \mathbb{R}^{M \times M}$, column vector $\mathbf{x} \in \mathbb{R}^M$ and $\mathbf{a} \in \mathbb{R}^M$, we have

- $\frac{\partial \mathbf{a}^\top \mathbf{x}}{\partial \mathbf{x}} = \frac{\partial \mathbf{x}^\top \mathbf{a}}{\partial \mathbf{x}} = \mathbf{a}$,
- $\frac{\partial \mathbf{x}^\top \mathbf{A}\mathbf{x}}{\partial \mathbf{x}} = (\mathbf{A} + \mathbf{A}^\top)\mathbf{x}$. If $\mathbf{A}$ is symmetric, $\frac{\partial \mathbf{x}^\top \mathbf{A}\mathbf{x}}{\partial \mathbf{x}} = 2\mathbf{A}\mathbf{x}$.
  A special case is, if $\mathbf{A} = \mathbf{I}$ (identity matrix), $\frac{\partial \mathbf{x}^\top \mathbf{x}}{\partial \mathbf{x}} = \frac{\partial \mathbf{x}^\top \mathbf{I}\mathbf{x}}{\partial \mathbf{x}} = 2\mathbf{I}\mathbf{x} = 2\mathbf{x}$.

The above rules adopt a *denominator-layout* notation. For more rules, you can refer to this Wikipedia page. Please apply the above rules and calculate following derivatives:

1. $f(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{a})^\top(\mathbf{x} - \mathbf{a})$ where $\mathbf{a} \in \mathbb{R}^M$ is a constant vector, derive $\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$.

$y = x - a$

$f(x) = \frac{1}{2}(y)^T(y)$

$f(x) = \frac{y^2}{2}$

$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial y} \cdot \frac{\partial y}{\partial x}$

$\frac{\partial y^T}{\partial} = \frac{\partial y}{\partial x}$

$\frac{2y}{2} \cdot 1$

$x - a$

2. $f(\mathbf{x}) = \frac{1}{2}(\mathbf{A}\mathbf{x} - \mathbf{b})^\top(\mathbf{A}\mathbf{x} - \mathbf{b})$ where $\mathbf{A} \in \mathbb{R}^{M \times M}$ is a constant matrix and $\mathbf{b} \in \mathbb{R}^M$ is a constant vector, derive $\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$.

$(Ax-b)^T = Ax-b$ due to symmetry

Hint: Note that $(\mathbf{A}^\top \mathbf{A})^\top = \mathbf{A}^\top \mathbf{A}$, thus $\mathbf{A}^\top \mathbf{A}$ is a symmetric matrix.

$f(x) = \frac{1}{2}(Ax-b)^T(Ax-b)$ $\quad y = Ax-b$

$f(x) = \frac{1}{2}(y)^T(y)$ $\quad \frac{2y}{2} \cdot A$

$\frac{x^2}{2}$ $\quad (Ax-b)A$

$A^2x - Ab$

3 (20 points) Metrics

In machine learning, we have many metrics to evaluate the performance of our model. For example, in a binary classification task, there is a dataset $S = \{(\mathbf{x}_i, y_i), i = 1, .., N\}$ where each data point $(\mathbf{x}, y)$ contains a feature vector $\mathbf{x} \in \mathbb{R}^M$ and a ground-truth label $y \in \{0, 1\}$. We have obtained a classifier $f : \mathbb{R}^M \to \{0, 1\}$ to predict the label $\hat{y}$ of feature vector $\mathbf{x}$:

$$\hat{y} = f(\mathbf{x})$$

Assume $N = 200$ and we have the following *confusion matrix* to represent the result of classifier f on dataset S:

	Actual Positives ($y = 1$)	Actual Negatives ($y = 0$)
Predicted Positives ($\hat{y} = 1$)	5 TP	5 FP
Predicted Negatives ($\hat{y} = 0$)	10 FN	180 TN

Please follow the lecture notes to compute the metrics below:

1. Please compute the *accuracy* of the classifier f on dataset S.

$$\frac{TP + TN}{TP + FP + FN + TN} = \frac{5 + 180}{200} = .925 = 92.5\%$$

2. Please compute the *precision* of the classifier f on dataset S.

$$\frac{TP}{TP + FP} = \frac{5}{10} = .5 = 50\%$$

3. Please compute the *F1 score* of the classifier f on dataset S.

$$\frac{2 \cdot Prec \cdot Rec}{Prec + Rec}$$

$$Rec = \frac{TP}{TP+FN} = \frac{5}{5+10} = \frac{5}{15} = .33$$

$$\frac{.377}{.933} = .399$$

4. You may find the accuracy of current model very high. Does it mean the performance of this model is always very good? Why?

Hint: You may refer to other metrics you have computed.

No since our F1 score was low at .399. This means both recall and precision were low with 50% precision & 33% specificity. Only 50% of the time is a true positive found when comparing to the total # of positives detected. Recall is at 33% meaning there is a high percentage of False negatives that the model is not correctly labeling as TP. The model accuracy is high due to the amount of TN found in the dataset which says more about the low quality of the dataset than the model itself. To improve, a dataset with more TP could be used to ascertain the actual capabilities of the model.

4 (10 points) Loss functions

1. Prove the Rooted Mean Square Error (RMSE) is always greater than or equal to the Mean Absolute Error (MAE).

   Hint: You may use Cauchy-Schwarz inequality.

$RMSE^2 \cdot n = \|a\|\|b\|$

Cauchy: $(a,b) \leq \|a\|\|b\|$

$\frac{\hat{Y}-Y}{n} \leq \frac{RMSE^2 \cdot n}{n}$

$\sqrt{MAE} \leq RMSE$

2. Another commonly used loss function is called Max Error (ME), which is the maximum absolute difference between two vectors. Suppose we have two vectors $y \in \mathbb{R}^n$ and $\hat{y} \in \mathbb{R}^n$, then $ME = \max_{i=1,\cdots,n} |\hat{y}_i - y_i|$. Prove ME is always greater than or equal to RMSE.

$(a,b) \leq \|a\| \|b\|$

$\|\hat{Y}-Y\|^2 \leq Max$

$ME = max(\hat{Y}-Y)$

$RMSE = \sqrt{sum\frac{(\hat{Y}-Y)^2}{n}}$

$\sqrt{(\hat{Y}-Y)(\hat{Y}-Y)} \leq \sqrt{\|\hat{Y}-Y\|^2}$

$\|\hat{Y}-Y\| \leq \sqrt{(\hat{Y}-Y)^2}$ $\sqrt{n}$ $RMSE^2 = \frac{1}{n}(\hat{Y}-Y)^2$

$\|\hat{Y}-Y\| \leq (\hat{Y}-Y)$ $RMSE^2 \cdot n = \hat{Y}^2 - Y^2$

$\|\hat{Y}-Y\| \leq Max$

$RMSE \cdot \sqrt{n} \leq Max$

$RMSE \leq \frac{Max}{\sqrt{n}}$

5 (10 points) Data Visualization

We will be using the UCI Wine dataset for this problem and Question 6. The description of the dataset can be found at https://archive.ics.uci.edu/ml/datasets/wine. You can load the dataset using the code below (recommended), or you can download the dataset ***here*** and load it yourself. You may refer the the Jupyter notebook `HW1-Q4-Q5.ipynb` for some skeleton code.

1. Show a scatter plot for the first 2 feature dimensions in 2-D space.

Some useful instructions are shown below:

- Import several useful packages into Python:

```
import matplotlib.pyplot as plt
from sklearn import datasets
```

- Load Wine dataset into Python:

```
wine = datasets.load_wine()
X = wine.data
Y = wine.target
```

Report ***your code*** and the ***scatter plot*** in Gradescope submission.

6 (10 points) Data Manipulation

We have already had a glimpse of the Wine dataset in Question 5. In this question, we will still use the Wine dataset. In fact, you can see the shape of array X is (178, 13) by running `X.shape`, which means it contains 178 data points and 13 features per data point. You may refer the the Jupyter notebook `HW1-Q4-Q5.ipynb` for some skeleton code. Here, we will calculate some measures of the array X and perform some basic data manipulation:

1. Show the first 2 features of the first 3 data points (i.e. first 2 columns and first 3 rows) of array X. (You can print the 3×2 array).

2. Calculate the mean and the variance of the 1st feature (the 1st column) of array X.

3. Randomly sample 3 data points (rows) of array X by randomly choosing the row indices. Show the indices and the sampled data points.

   Hint: You may use `np.random.randint()`.

4. Add one more feature (one more column) to the array X after the last feature. The values of the added feature for all data points are constant 1. Show the first data point (first row) of the new array.

   Hint: You may use `np.ones()` and `np.hstack()`.

Some useful instructions are shown below:

- Get a row or a column of the array X:

```
print X[0]           # Print the first row of array X.
print X[:, 0]        # Print the first column of array X.
                     # ':' here means all rows and '0' means column 0.
```

- Get part of the array:

```
print X[3:5, 1:3] # Print 4th and 5th rows, 2nd and 3rd columns.
print X[:3, :2]   # Print first 3 rows, first 2 columns.
```

- You may refer to a quick tutorial using NumPy here:
  http://cs231n.github.io/python-numpy-tutorial/

Report ***your code*** and the ***results of data manipulation*** in Gradescope submission.

Q4 Data Visualization

In [1]:
```
import matplotlib.pyplot as plt
from sklearn import datasets
```

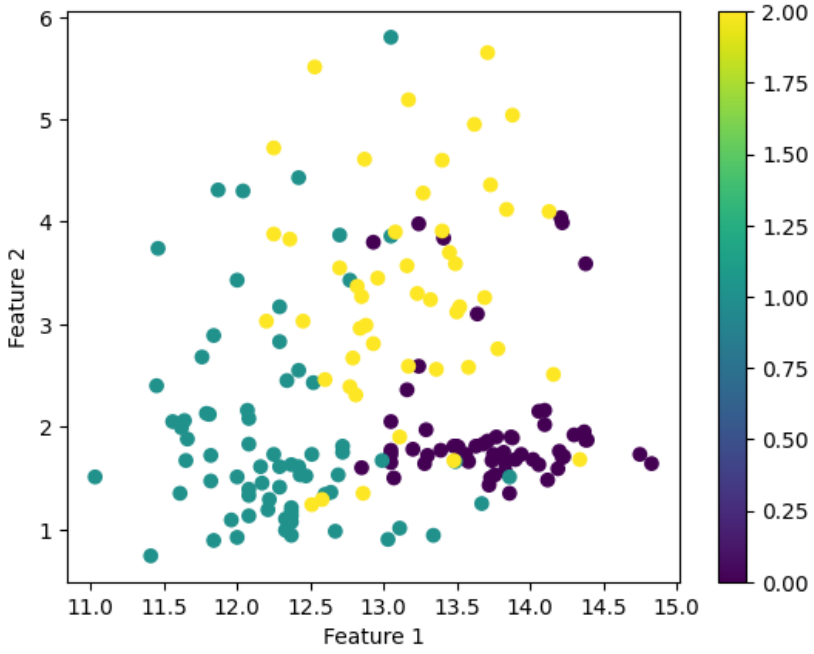
In [2]:
```
# Load data
wine = datasets.load_wine()
X = wine.data
Y = wine.target
print(X.shape, Y.shape)
```

```
(178, 13) (178,)
```

In [3]:
```
###### To be filled ######
# Hint: You may use plt.scatter() to draw the plot. You may need to set the 'c' argument
#       in order to have different color for the data points with different classes in Y.

x_features = X[:, :2]

plt.scatter(x_features[:,0], x_features[:,1], c=Y)
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.colorbar()
plt.show()
```



Q5 Data Manipulation

In [4]:
```
import numpy as np
```

1. Show the first 2 features of the first 3 data points.

In [5]:
```
###### To be filled ######

x_features = X[:3, :2]
print(x_features)
```

```
[[14.23  1.71]
 [13.2   1.78]
 [13.16  2.36]]
```

2. Calculate the mean and the variance of the 1st feature (the 1st column) of array X.

In [6]:
```
###### To be filled ######
# Hint: You may use np.mean() and np.var()

feature_one_mean = np.mean(X[:, :1])
feature_one_variance = np.var(X[:, :1])

print(f'mean: {feature_one_mean}')
print(f'variance: {feature_one_variance}')
```

```
mean: 13.00061797752809
variance: 0.6553597304633255
```

3. Randomly sample 3 data points (rows) of array X by randomly choosing the row indices. Show the indices and the sampled data points.

In [7]:
```
###### To be filled ######
# Hint: You may use np.random.randint().

indices = np.random.randint(low=0, high=X.shape[0], size=3)
print("Randomly chosen indices:", indices)

sampled_data = X[indices, :]
print("\nSampled data points:\n", sampled_data)
```

```
Randomly chosen indices: [ 40 106  96]

Sampled data points:
 [[1.356e+01 1.710e+00 2.310e+00 1.620e+01 1.170e+02 3.150e+00 3.290e+00
  3.400e-01 2.340e+00 6.130e+00 9.500e-01 3.380e+00 7.950e+02]
 [1.225e+01 1.730e+00 2.120e+00 1.900e+01 8.000e+01 1.650e+00 2.030e+00
  3.700e-01 1.630e+00 3.400e+00 1.000e+00 3.170e+00 5.100e+02]
 [1.181e+01 2.120e+00 2.740e+00 2.150e+01 1.340e+02 1.600e+00 9.900e-01
  1.400e-01 1.560e+00 2.500e+00 9.500e-01 2.260e+00 6.250e+02]]
```

4. Add one more feature (one more column) to the array X after the last feature. The values of the added feature for all data points are constant 1. Show the first data point (first row) of the new array

In [8]:
```
###### To be filled ######
# Hint: You may use np.hstack() and np.ones()

ones = np.ones((X.shape[0], 1))

hstack_value = np.hstack((X, ones))

print("First data point (with added feature):\n", hstack_value[0])
```

```
First data point (with added feature):
 [1.423e+01 1.710e+00 2.430e+00 1.560e+01 1.270e+02 2.800e+00 3.060e+00
 2.800e-01 2.290e+00 5.640e+00 1.040e+00 3.920e+00 1.065e+03 1.000e+00]
```

Submission Requirement

Please combine your code, plot and results for Q4, Q5 with your answers for Q1, Q2, Q3 together as a single PDF and submit it through Gradescope.

A easy way is, you can save your completed Jupyter notebook as a PDF (e.g. in Chrome, right click the web page -> Print ... -> Save as PDF) and then merge it with your answers for Q1, Q2, Q3.