Homework Assignment 1

CSE 151A: Introduction to Machine Learning

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 want to split submissions into groups based on similarity of scores, but we do not have any predefined classes. Since we are grouping submissions based on similarity, this is a clustering task.

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:
   Here, we have discrete labels (A, B, C, D) that we need to map the submissions to. Since we are assigning submissions to predefined classes, this is a clustering task.

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 $\dfrac{\partial f(x)}{\partial x}$.

$$\frac{d}{dx}\left[\frac{1}{2}(ax-b)^2\right] = 2 \cdot \frac{1}{2}(ax-b)^1 \cdot \frac{d}{dx}[ax-b]$$

$$= \boxed{a(ax-b)}$$

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

$$\frac{d}{dx}\left[\ln(1+e^x)\right] = \frac{1}{1+e^x} \cdot \frac{d}{dx}\left[1+e^x\right]$$

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

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

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

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 $\dfrac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$.

$$f(x) = \frac{1}{2}(x^T x - a^T x - x^T a + a^T a)$$
$$\frac{1}{2} \cdot \frac{d}{dx}\left[x^T x - a^T x - x^T a + a^T a\right]$$
$$= \frac{1}{2} \cdot \left[2x - a - a\right]$$
$$= \boxed{x - a}$$

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 $\dfrac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$.

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

$$f(x) = \frac{1}{2}(x^T A^T A x - x^T A^T b - b^T A x + b^T b)$$
$$\frac{1}{2} \cdot \frac{d}{dx}\left[x^T A^T A x - x^T A^T b - b^T A x + b^T b\right]$$
$$= \frac{1}{2} \cdot \left[2A^T A x - A^T b - A^T b\right]$$
$$= \boxed{A^T A x - A^T b}$$

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	5
Predicted Negatives ($\hat{y} = 0$)	10	180

Please follow the lecture notes to compute the metrics below:

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

$$\text{accuracy} = \frac{TP+TN}{TP+FP+TN+FN}$$

$$= \frac{5+180}{200}$$

$$= \boxed{0.925}$$

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

$$\text{precision} = \frac{TP}{TP+FP}$$

$$= \frac{5}{5+5}$$

$$= \boxed{0.5}$$

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

$$\text{recall} = \frac{TP}{TP+FN} = \frac{5}{5+10} = \frac{1}{3}$$

$$\text{F1 score} = \frac{2 \cdot \text{Prec} \cdot \text{Rec}}{\text{Prec} + \text{Rec}} = \frac{2 \cdot \frac{1}{2} \cdot \frac{1}{3}}{\frac{1}{2} + \frac{1}{3}}$$

$$= \frac{1/3}{5/6} = \boxed{0.4}$$

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, accuracy is only one metric of model performance. We must also take into consideration metrics such as precision, recall, and F1 score. As we see in the problem above, we have high accuracy (0.925), but poor precision (0.5), poor recall (0.333), and poor F1 score (0.4)

The low precision and recall means our model will result in a high number of false positives and false negatives. Subsequently, the F1 score which is a balance between precision and recall was also low.

Q4 Data Visualization

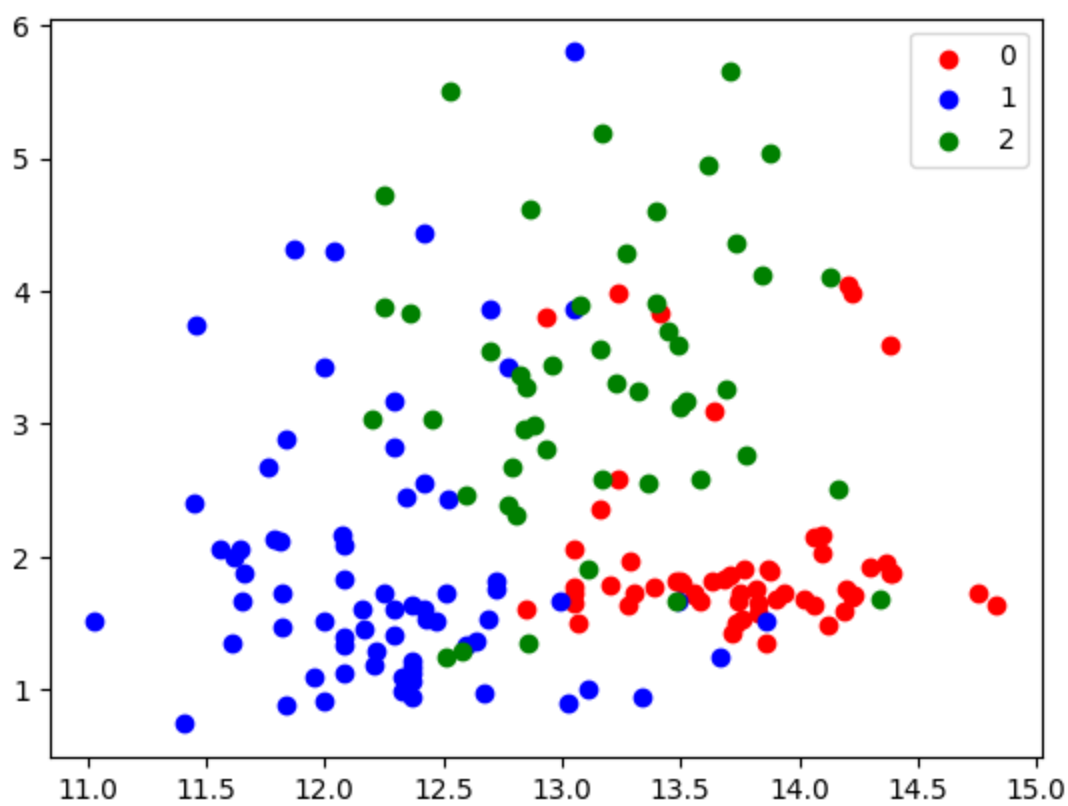
In [79]:
```
from matplotlib import pyplot as plt, colors
import numpy as np
from sklearn import datasets
```

In [80]:
```
# Load data
wine = datasets.load_wine()
X = wine.data
Y = wine.target
print(X.shape, Y.shape)
```

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

In [81]:
```
###### 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
cdict = {0: 'red', 1: 'blue', 2: 'green'}

fig, ax = plt.subplots()
for t in np.unique(Y):
    ix = np.where(Y == t)
    ax.scatter(X[ix,0], X[ix,1], c = cdict[t], label = t)
ax.legend()
plt.show()
```



Q5 Data Manipulation

In [82]:
```
import numpy as np
```

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

In [83]:
```
print(X[:3,:2])
```

```
[[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 [84]:
```
###### To be filled ######
# Hint: You may use np.mean() and np.var()
print("Mean:", np.mean(X[:,0]))
print("Variance:", np.var(X[:,0]))
```

```
Mean: 13.000617977528089
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 [85]:
```
###### To be filled ######
# Hint: You may use np.random.randint().
idxs = np.random.randint(len(X), size=3)
for idx in idxs:
    print("Index", idx, ":", X[idx])
```

```
Index 91 : [ 12.     1.51   2.42  22.    86.     1.45   1.25   0.5    1.63   3.6
   1.05   2.65 450.  ]
Index 54 : [1.374e+01 1.670e+00 2.250e+00 1.640e+01 1.180e+02 2.600e+00 2.900e+00
 2.100e-01 1.620e+00 5.850e+00 9.200e-01 3.200e+00 1.060e+03]
Index 103 : [1.182e+01 1.720e+00 1.880e+00 1.950e+01 8.600e+01 2.500e+00 1.640e+00
 3.700e-01 1.420e+00 2.060e+00 9.400e-01 2.440e+00 4.150e+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 [86]:
```
###### To be filled ######
# Hint: You may use np.hstack() and np.ones()
col = np.ones((len(X), 1))
X = np.hstack((X, col))
print(X[0])
```

```
[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.

In [ ]:
```

```