ECE 479: IoT and Cognitive Computing
Spring 2024, Homework

Due on Feb. 7, 2024 at 11:59:59 PM

Question 1: Edge Computing and Sensors (25 pts)

1. List two advantages and disadvantages of using Cloud Computing or Edge Computing for an IoT system. [**8 pts**]
   - Cloud Computing:
     - Advantages:
       (a) Centralized protection and Cost - Efficiency
       (b) Scalability
     - Disadvantages:
       (a) Latency and Response Time
       (b) Dependency on Network Connectivity
   - Edge Computing:
     - Advantages:
       (a) Low Latency and Support for mobility
       (b) Data Privacy
     - Disadvantages:
       (a) Limited Scalability
       (b) Data Abstraction and Complexity of Management

2. For the following applications, which paradigm (Cloud Computing or Edge Computing) is more suitable? Briefly explain why. [**6 pts**]
   - Health data collected by a smartwatch to track your daily activities.
     —Edge computing.
     Health data need to be updated everyday which require real-time processing.
     Edge computing have less latency and allow for immediate feedback.

- Temperature sensors are placed inside a refrigerated storage container to regulate the temperature during the shipping process.
  –Edge computing
  Temperature regulation in a refrigerated storage container requires quick and precise control. Edge computing is appropriate in this case as the temperature sensors can process and analyze data locally which facilitate quick adjustment to maintain temperature.
- License plate readers at toll plazas.
  – Cloud computing.

  Cloud computing can help to store and process data from license plate readers. Because we need to store toll record, cloud computing offer scalability and store a large amount of data, enabling operators to aggregate and analyze data from different toll plazas.
- Medical wearable devices that detect when you fall.
  –Edge computing.
  Edge computing can process sensor data locally, allowing for quick response without depend on cloud connectivity. Medical wearable devices need immediate feedback.

3. Wireless Sensors Network (WSN) Aggregation Strategy. [**11 pts**]

   All nodes except A are sensors, and node A is the sink node (gateway). The numbers on the edges are the corresponding transmission latency. To find out the shortest path between two connected nodes in a graph, you may need the help of the Dijkstra's algorithm.

   We want to collect data from sensors K, E, and G and transmit them back to sink node A. Please propose a plan, and calculate minimum **total** latency and the number of operations required in the transmission.

   Assume each aggregation at one node requires 5 milliseconds (ms) to complete. Justify your answer and fill in the following table. (You may not need to fill in all columns.)

# of Operations	K	E	G		
Transmission	2	2	2		
Aggregation	1	1	2		
Total	3	3	4		

Table 1: Wireless Sensors Network Traversing

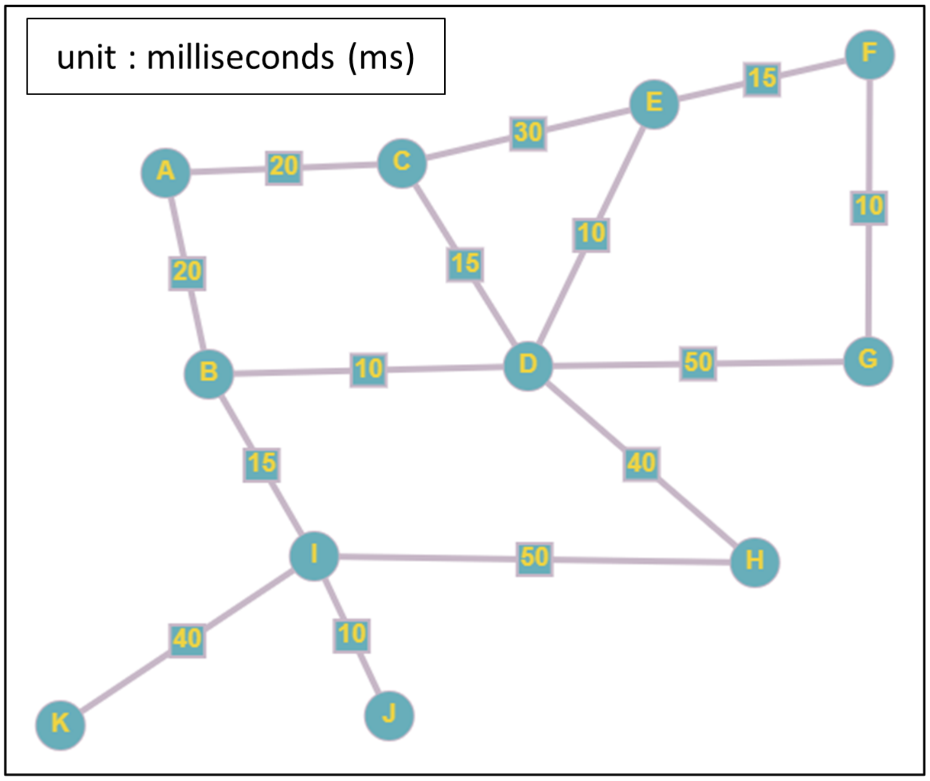

Figure 1: WSN topology

Question 2: Basic Python and NumPy programming (15 pts)

1. **Python One-Liners: Basics [6 pts]**

   (a) Write down **a single line** Python expression that calculates $\sum_{n=0}^{i} n!$ using only Python built-in functions. [**1 pts**]
   result = sum(math.factorial(n) for n in range(i+1))

   (b) Write down **a single line** of Python code to sum the 0, 3, 6, 9,...,(3n)th elements from a list A using only Python built-in functions. [**1 pts**]
   result = sum(A[::3])

   (c) Write down **a single line** of Python code to sum the 2, 5, 8, 11, (3n+2)th elements from a list A using only Python built-in functions. [**1 pts**]
   result = sum(A[2::3])

   (d) Write down **a single line** of Python code to reverse given list A and save it to list B. [**1 pts**]
   A = B [::-1]

(e) Given two lists of numbers in Python, A and B, write down **a single line** of Python code to construct a new list of pairs of corresponding elements in two lists. For example, A = [1, 2, 3], B = ['a', 's', 'd', 'f'], then your code should give [(1, 'a'), (2, 's'), (3, 'd')]. **[1 pts]**

C = list((A[i], B[i]) for i in range(min(len(A), len(B))))

(f) Given two lists of numbers in Python, A and B, write down **a single line** of Python code to construct a new list of elements in A that appear in B as well. For example, A = [1, 2, 4, 4, 2, 1], B = [1, 4], then your code should give [1, 4, 4, 1] **[1 pts]**
C = [A[i] for i in range(len(A)) if A[i] in B]

2. **More one-liners: List and String Operations** [4 pts]
Use the following list and string methods:
`append(), count(), extend(), insert(), join(), remove(), split()`
Fill in the blank (parenthesis) to output the expected results. The comments are what you should expect after you have run the following line of code. You should only use one method in each blank.

```
# split string s0 by space and put into a list s0_list
# output: ['Oppenheimer', 'drinks', 'some', 'coffee', 'coffee',
   'coffee', 'coffee']
s0 = "Oppenheimer drinks some coffee coffee coffee coffee"
s0_list = ( s0.split( " " ) )

# count the occurrence of element 'coffee'
# output: 4
count = ( s0.count("coffee") )

# keep only the first four elements in the list
# output: ['Oppenheimer', 'drinks', 'some', 'coffee']
s0_list = ( s0.spilit( " " )[:4] )

# take out element 'some' from the list
# output: ['Oppenheimer', 'drinks', 'coffee']
s0_list.( remove("some") )

# put element 'much' in the list between 'drinks' and 'coffee'
# output: ['Oppenheimer', 'drinks', 'much', 'coffee']
s0_list.( insert(2, 'much') )

# put element 'everyday' at the end of the list
# output: ['Oppenheimer', 'drinks', 'much', 'coffee', 'everyday
   ']
s0_list.( append('everyday') )

# split string s1 by space and put into a list
# output: ['in', 'the', 'afternoon']
```

```
s1 = "in the afternoon"
s1_list = ( s1.split(" ") )

# put s1_list at the end of s0_list
# output: ['Oppenheimer', 'drinks', 'much', 'coffee', 'everyday
    ', 'in', 'the', 'afternoon']
s0_list.( extend(s1_list) )

# make a string from the list, connected with a space ' '
# output: Oppenheimer drinks much coffee everyday in the
    afternoon
s = ( ' '.join(s0_list) )
```

3. **NumPy Slicing [2 pts]**

   The **shape** of a NumPy array *tmp* is (5, 3, 4, 2), which corresponds to dimensions $[i, j, k, l]$.

   ```
   tmp = tmp[2:4, :-1, ::2, :1]
   ```

   What is the shape of the NumPy array *tmp* and which part of the original NumPy array *tmp* has been extracted after this slicing?

   Please describe it in terms of i, j, k, l dimension. (for example : row 0 and row 1 of dimension i and .... of dimension j and ...)
   shape of tmp is (2, 2, 2, 1)
   Row 2 and Row 3 of dimension i
   Row 0 to Row 1 of dimension j
   Row 0 and Row 2 of dimension k
   Row 0 of dimension l

4. **NumPy Operations [3 pts]**
   We have `A = np.array([1, 2, 3, 4, 5, 6])` and a mystery NumPy array `B`. Printing `A*B` gives the output of `[2 14 9 20 30 24]`. What is this mystery NumPy array `B`?

   Now suppose that NumPy array `A1` and `B1` are transformed from A and B using NumPy array manipulation functions.

Please find out a way to perform these transformations from `A` and `B`, using NumPy array operations, such that

```
print(np.dot(A1, B1).ravel())
```

gives the output of [10 20 9 24 46 23 38 72 37]
Solution:
B = np.array([2,7,3,5,6,4])
B = np.array([2,7,3,5,6,4])
A = np.array([1, 2, 3, 4, 5, 6])
B = B[::-1]
A1 = A.reshape((3,2))
B1 = B.reshape((2,3))
output = np.dot(A1,B1).ravel()

Question 3: Data exploration (20 pts)

1. **Data visualization**
   Suppose you are analyzing a dataset that contains information about the sales performance of a retail company over the past year. You want to gain insights into the data and communicate your findings effectively.
   (a) How would you create a scatter plot to visualize the relationship between the company's advertising expenditure and its monthly sales revenue? What do the resulting patterns on this scatter plot tell you about this relationship? **[3 pts]**

   We plot expenditure on x-axis and monthly revenue on y-axis. We can use python matplotlib.pyplot module. We can use plt.scatter(expenditure, revenue) to display the relationship. The pattern can indicate whether expenditure influence revenue directly.
   If the points on the plot has a upward trend, indicating a positive correlation between advertising expenditure and revenue.
   (b) To understand the distribution of monthly sales revenue, what type of chart or graph would you use, and why? Provide a brief explanation of how you would construct this chart, and what insights it might provide.**[3 pts]**
   A histogram provides a visual representation of the frequency distribution of a continuous dataset We can use python module matplotlib.plt . We can use plt.hist to generate histogram. It can help to visualize revenue in different months which show the change in revenue. By dividing the range of sales revenue into interval and plotting the frequency we can see the distribution of a continuous dataset.

(c) You want to represent the contribution of each product category to the total annual revenue. What type of chart would be suitable for this purpose? Explain how you would create and interpret such a chart to effectively communicate the data.[**3 pts**]
pie chart. The pie chart can display the contribution of each product category. Firstly, we need to calculate the percentaafe of revenue of each category relative to the total revenue. Then we can create the pie chart. We can use python module matplotlib.plt . We can use plt.pie to visualize the data. You can easily compare the contributions of different product categories by observing the lengths of their respective segments.

(d) Finally, suppose you are interested in predicting future monthly sales revenue based on the past year's data. How would you go about fitting a regression line to the data points in the scatter plot you created earlier? Describe the steps involved and what information the regression line can provide.[**3 pts**]
1. Collect the data
2. Scatter Plot
3. Split the data to training data and test data
4. Regression analysis: Based on the scatter plot of the data, we should choose right model
5. Visualize the regression line and test the accuracy by calculating least squares regression.
6.Interpretation. Observe the regression line and find the relationship between expenditure and monthly sales revenue.
7. Predict the future data

2. **Fitting the Data**
Suppose that we have collected some data points for training and plotted them in the scattered plots, shown below. To find the trend of these data points, we constructed three regression models. Later, we want to use one of these models to describe the incoming unknown data points. Observe the following three graphs and answer the questions.

(a) Among the three models, which one do you think is the best? Why do you choose this model? [**2 pts**]
Model 2 fits the best. In model 1, many points are not on the line which is underfitting. In model 3, every points are on the line which is overfitting.

(b) What are the potential issues with the other two? Please explain in two to three sentences. [**2 pts**]
Over-fitting may generate complex model which is sensitive to specific training data, which has low performance in the test data. This will lead

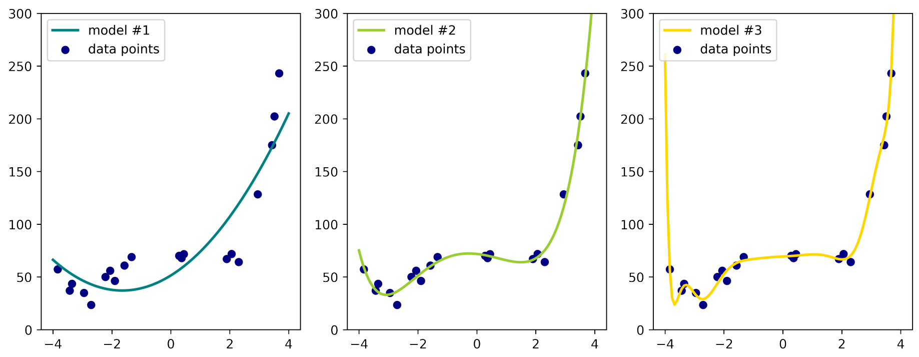

Figure 2: Three Different Models

to high variance and poor performance. Under-fitting model ignore some feature and too simple to fit data. It has high bias and poor predicted performance.

3. **Training and Validation**
   When developing a machine learning model, we usually need to evaluate the model's accuracy before testing it on real-world data. Since we only have access to the training dataset, we often have to split it into two parts, one for training and the other for validation.

   (a) In about three sentences, describe why we need a separate testing dataset to evaluate the trained model. [2 pts] **[2 pts]**
   This helps ensure that the model has not merely memorized the training examples but has learned patterns that can generalize to real-world scenarios, providing a more reliable measure.

   (b) Why do we need validation/cross-validation if the testing set is enough to evaluate the real-world performance of a model? That is, why can't we tune the model based on the testing dataset directly ? **[2 pts]**
   Using testing data directly may lead to over-fitting directly. Using validation/ cross-validation can help with model selection while didn't change the train test. Additionally, using validation set can help us adjust the hyperparameters and find the best module.

Question 4: PCA and Preprocessing of data (18 pts)

1. **Curse of dimensionality**[**3 pts**]
   Briefly explain "the curse of dimensionality". Why high dimensional data is a "curse"? What makes it difficult to model high-dimensional data?
   The curse of dimensionality refers to various phenomena that arise when analyzing

and organizing data in high-dimensional spaces . When the dimensionality increases, the volume of the space increases so fast that the available data become sparse. This may have many problem such as, overfitting, complex computation.

2. **Statistical interpretation of PCA[3 pts]**
   In the lectures, you have learned the statistical interpretation of PCA. Briefly describe it below.
   Standardize the n-dimensional original data and Construct the covariance matrix for features. Decompose the covariance matrix into its eigenvectors and eigenvalues. Sort the eigenvalues by decreasing order to rank the corresponding eigenvectors. Transform the n-dimensional original data into the m-dimensional space. The principal component capture variance in the data. Selecting a subset of principal component that represent main variance in the data, we can reduce its dimensionality.

3. **PCA on simple data[12 pts]** Suppose you have a tiny dataset with two features $X1$ and $X2$. You want to perform PCA on this dataset to reduce its dimensionality.

```
Data Point 1: X1 = 2, X2 = 4
Data Point 2: X1 = 3, X2 = 6
Data Point 3: X1 = 5, X2 = 10
Data Point 4: X1 = 7, X2 = 14
```

   Please follow the steps below and find the projection of this dataset onto the principal component.

   (a) Calculate the mean values of $X1$ and $X2$ and then center the dataset by subtracting these means from each data point.

   X1: 4.25 X2: 8.5

$$\begin{pmatrix} -2.25 & -4.5 \\ -1.25 & -2.5 \\ 0.75 & 1.5 \\ 2.75 & 5.5 \end{pmatrix}$$

   (b) Calculate the covariance matrix for the centered dataset. Show the full covariance matrix.

$$\begin{pmatrix} 4.91667 & 9.83333 \\ 9.83333 & 19.6667 \end{pmatrix}$$

   (c) Calculate the eigenvalues and eigenvectors of the covariance matrix you computed in the previous step. Provide both the eigenvalues and the

corresponding eigenvectors. Select the most significant principal component (eigenvector) based on the eigenvalues calculated.
eigenvalues : 0
eigenvector:

$$\begin{pmatrix} -0.89442719 & 0.44721 \end{pmatrix}$$

eigenvalues : 24.58333

$$\begin{pmatrix} 0.4472136 & -0.894427 \end{pmatrix}$$

eigenvalues: 24.58333 is the biggest. So the most significiant principal component is (0.4472136 , -0.894427)

(d) Project the centered data onto the chosen principal component. Show the projection of the data points along this component by taking the dot product of the centered data points and the principal component vector.

$$\begin{pmatrix} 3.01869176 \\ 1.67705098 \\ -1.0062305 \\ -3.68951215 \end{pmatrix}$$

Question 5: kNN and Linear Regression (22 pts)

1. **Simple regression with kNN [2 pts]**
   In your own words, describe how the kNN algorithm calculates the regression. You can assume that all the dependent and independent variables are numerical.
   We need to find k nearest neighbor of the train dataset and then averaging their target values to estimate the value of the target variable for the test data point.

2. **kNN Classification**
   Given a dataset containing points in a 3D space belonging to one of three classes (Class A, Class B, and Class C), with each point having three features (x, y, z), calculate the Euclidean distance between a new point $P(4, 4, 4)$ and all the points in the dataset. The dataset is as follows:

   - Class A: $(1, 2, 1)$, $(2, 3, 2)$, $(3, 3, 1)$, $(2, 2, 2)$
   - Class B: $(7, 2, 1)$, $(8, 3, 2)$, $(9, 1, 3)$, $(7, 4, 1)$
   - Class C: $(5, 7, 7)$, $(7, 2, 5)$, $(6, 5, 3)$, $(5, 5, 6)$

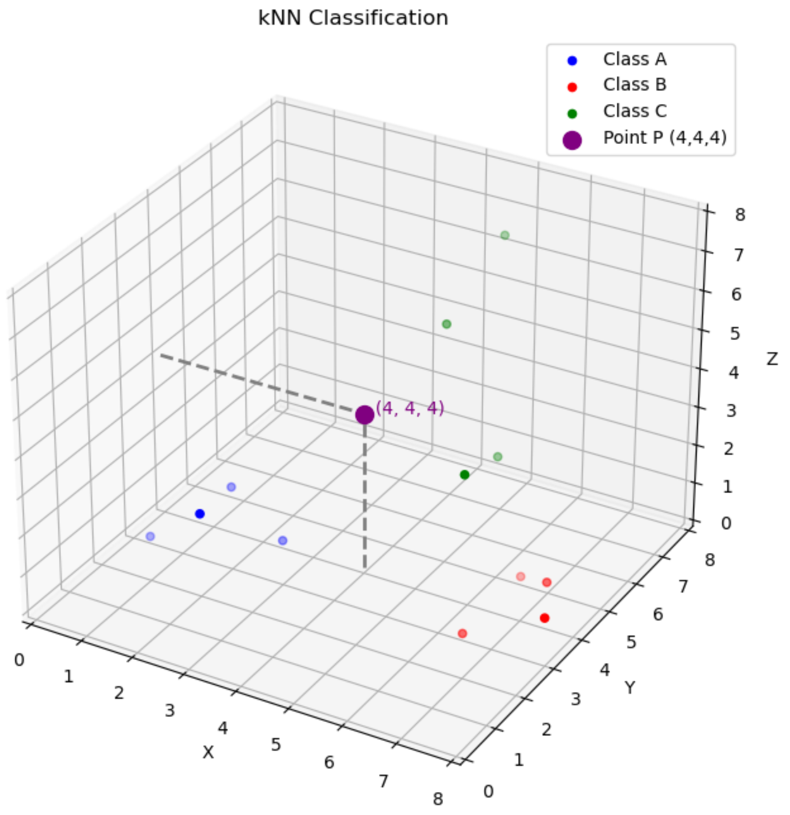

Figure 3: Demonstration of the dataset

(a) **Calculate the Euclidean Distance [4 pts]**
Calculate the Euclidean distance between the new point $P(4, 4, 4)$ and all the points in the dataset using the formula:

$$d(P, Q) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Class A: [4.69041576, 3. , 3.31662479, 3.46410162]
Class B: [4.69041576, 4.58257569, 5.91607978, 4.24264069]
Class C: [4.35889894, 3.74165739, 2.44948974, 2.44948974]

(b) **Determine the Optimal k [4 pts]**
Based on the calculated distances, classify point P using kNN for $k = 3$, $k = 4$, and $k = 5$. With a tie between Class A and Class C for $k = 4$, additional criteria (e.g., weighted distance) might be needed for classification. Which value of k provides the most reasonable classification for point P? Discuss the impact of different k values on the classification outcome. Plot the results of the selected optimal k to verify your choice.

K = 3 2 nearest points in Class C and 1 in Class A P belong to C

K = 4 2 nearest points in Class C and 2 in Class A. I deasign the following weighted method. Firstly, we add up distances of nearest points in class A and class B respectively. Comparing to Class A, point P is close to points in class C. Point P belong to C.
K = 5 2 nearest points in Class C and 3 in Class A P belong to A
K = 5 provide the most reasonable classification for P. Because it will consider larger amount of nearest points which can lead to low bias and reduce local variation.
Smaller K is more sensitive to local variation. While larger K will consider the whole dataset and help to deal with local variation and noise.

3. **Linear regression in greater depth [12 pts]**
Linear regression is one of the simplest, yet also an extremely effective algorithm. Interestingly, it can be interpreted from 2 different perspectives:

   (a) The first interpretation of this problem is straightforward: we simply want a linear function that can produce the minimal error of a given form, and for linear regression, we usually pick the mean squared error (MSE). To begin with, we denote the set of parameters (weights) with $\boldsymbol{w}$ and the features of the i-th sample with $\boldsymbol{x}^{(i)}$. So the prediction for the i-th sample is:

$$\hat{y}^{(i)} = \boldsymbol{w}^T \boldsymbol{x}^{(i)}$$

   Given the information above, write down the equation of the mean squared error (MSE) to be minimized, the first part of the equation is given to you. **[4 pts]**

$$\text{Minimize}_w f(w) = \frac{1}{i} \sum_{i=0}^{n} (y^{(i)} - \hat{y}^{(i)})^2$$

   (b) The linear regression can also be explained as a geometry/linear algebra problem. From this perspective, $\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{w}}$ has an alternative interpretation: a linear combination of the columns of matrix $\boldsymbol{X}$. For example, a column vector with three elements can be written as:

$$\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$$

   where v_1, v_2, and v_3 are the elements of the vector.
   In this interpretation, the vector of truth values $\boldsymbol{y}$, can be seen as a vector that points out from the hyper-plane. So, minimizing the error becomes minimizing the distance between the vector constructed with the vectors on the plane $\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{w}}$ and the ground truth vector $\boldsymbol{y}$, as demonstrated in

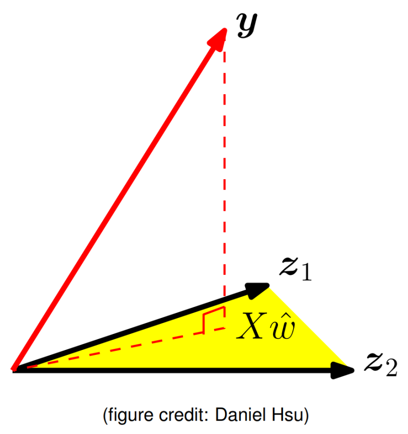

Figure 4: Demonstration of the least squares

the following figure extracted from Lecture 6. In geometry, this is the same as minimizing the magnitude of the vector $\boldsymbol{y} - \hat{\boldsymbol{y}}$. From pure geometry knowledge, we know that such a vector is shortest when it is perpendicular to the surface, in which case it will be orthogonal to all the vectors in that plane. Using this relationship, we can then derive the analytical expression of $\hat{\boldsymbol{w}}$, as explained in Lecture 6.

$$\boldsymbol{X}^T(\boldsymbol{X}\hat{\boldsymbol{w}} - \boldsymbol{y}) = 0 \quad (1)$$

$$\boldsymbol{X}^T\boldsymbol{X}\hat{\boldsymbol{w}} = \boldsymbol{X}^T\boldsymbol{y} \quad (2)$$

$$\hat{\boldsymbol{w}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \quad (3)$$

i. In your own words, describe how you establish Equation (1) above and interpret this relationship. **[4 pts]**

We can consider its geometric interpretation of linear regression. we view $\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{w}}$ as plane formed by linear combination of columns of matrix X. Y is true value which point out from plane. $\boldsymbol{x}^T$ represent transpose of feature matrix of X. $\boldsymbol{X}\hat{\boldsymbol{w}}$ denote predicted value. $\hat{\boldsymbol{y}} - \boldsymbol{X}\hat{\boldsymbol{w}}$ denote the difference between predicted value and truth ground. When this equation is zero, indicating that Xw - y is orthogonal to hyper-plane.

ii. Equation (3) takes a very different form from the answer in Part (a). Why do they describe the same problem? Think about the geometric interpretation and explain your understanding. **[4 pts]**

It means that the error vector is perpendicular to hyper-plane. It indicate that the error is minimized when it is orthogonal to the plane. It leads to the shortest distance between plane and value.