1.

a) The examples addressed using a supervised learning algorithm?

- (i) Given email labeled as spam or not spam, learn a spam filter: This is a classic example of a **classification** problem, where the goal is to predict a discrete output (spam or not spam) based on a set of input features (email text, sender, recipient, etc.). Supervised learning algorithms such as **Naive Bayes**, **Support Vector Machines (SVMs)**, and **Decision Trees** are all well-suited for this task.
- (ii) Given a set of news articles found on the web, group them into sets of articles about the same story: This is a **clustering** problem, where the goal is to group data points into clusters based on their similarity. Unsupervised learning algorithms such as **k-means clustering** and **hierarchical clustering** can be used for this task. However, supervised learning algorithms such as **k-nearest neighbors (KNN)** can also be used for clustering if the desired clusters are known beforehand.

b) The answer is **(iii) Reinforcement Learning**. Reinforcement learning is a type of machine learning where an agent learns by interacting with its environment and receiving rewards or penalties for its actions. It is different from supervised learning and unsupervised learning in that it does not require labeled data or specific goals.

c) (ii) False. Overfitting is attributed by **high variance** and **low bias**.

d) KNN is called a lazy learner algorithm because it does not explicitly learn a model from the training data. Instead, it stores all the training data and predicts the label for a new data point by finding the k nearest neighbors in the training data and using the majority vote of their labels. This means that the training time for KNN is minimal, but the prediction time can be slow, especially for large datasets.

e) Two advantages of Gradient Descent method over Least square method are:

- Gradient descent is a more general optimization algorithm that can be used to minimize any differentiable function, while least squares is specifically designed for linear regression problems.
- Gradient descent can be used to solve problems with non-convex cost functions, while least squares can only be used for convex cost functions.

I hope this helps!

2. (a) Explain at least 5 different metrics with appropriate mathematical expressions for assessing regression performance of machine learning models.

Here are five different metrics with appropriate mathematical expressions for assessing regression performance of machine learning models:

1.

Mean Squared Error (MSE):

This metric measures the average squared difference between the predicted values and the actual values.
Mathematically, it is defined as:

```
MSE = 1/n * Σ(y_i - y_hat_i)^2
```

Where:
n is the number of data points
y_i is the actual value of the i-th data point
y_hat_i is the predicted value of the i-th data point

Root Mean Squared Error (RMSE):

This metric is the square root of the mean squared error.

It is defined as:

```
RMSE = sqrt(MSE)
```

RMSE is easier to interpret than MSE because it is in the same units as the original data.

Mean Absolute Error (MAE):

This metric measures the average absolute difference between the predicted values and the actual values.
It is defined as:

```
MAE = 1/n * Σ|y_i - y_hat_i|
```

MAE is less sensitive to outliers than MSE because it uses absolute differences instead of squared differences.

R-squared:

This metric measures the proportion of variance in the dependent variable that is explained by the independent variable.
It is defined as:

```
R^2 = 1 - Σ(y_i - y_hat_i)^2 / Σ(y_i - y_mean)^2
```

Where:
y_mean is the mean of the actual values

Adjusted R-squared:

This metric is a modified version of R-squared that penalizes for the number of predictors in the model.
It is defined as:

```
Adjusted R^2 = 1 - (1 - R^2) * (n - 1) / (n - p)
```

Where:
p is the number of predictors in the model

These are just a few of the many metrics that can be used to assess regression performance. The best metric to use will depend on the specific problem and the characteristics of the data.

2. (b) Draw a comparison between Linear regression and Logistic Regression.

Feature	Linear Regression	Logistic Regression
Purpose	Predicts continuous target variables	Predicts binary or categorical target variables
Model	Linear relationship between the independent and dependent variables	Sigmoid function to map the linear combination of independent variables to a probability between 0 and 1
Output	Continuous value	Probability value between 0 and 1
Applications	Predicting house prices, stock prices, customer	Classifying emails as spam or not spam, predicting loan defaults

return rate

Here are some additional points to consider:

- Linear regression is more sensitive to outliers than logistic regression.
- Logistic regression can be used for multi-class classification problems with extensions like multinomial logistic regression.
- Linear regression is generally easier to interpret than logistic regression.

Sure, based on the image you sent, here are the answers to the questions:

2. (a) Give one real-life application each of supervised and unsupervised learning. Differentiate between the two learnings.

Supervised learning:

- **Real-life application:** Spam filtering.
- **Explanation:** Supervised learning algorithms are trained on labeled data, where each data point has a corresponding label or output value. In spam filtering, the data points are emails, and the labels are "spam" or "not spam". The algorithm learns to identify patterns in the emails that are indicative of spam, and then uses these patterns to classify new emails.

Unsupervised learning:

- **Real-life application:** Customer segmentation.
- **Explanation:** Unsupervised learning algorithms are trained on unlabeled data, where the data points do not have any predefined labels or categories. In customer segmentation, the data points are customer profiles, and the goal is to group these profiles into different segments based on their similarities. Unsupervised learning algorithms can identify patterns in the data that can be used to group customers with similar characteristics together.

The key differences between supervised and unsupervised learning is the presence of labeled data:

Supervised Learning	Unsupervised Learning
Supervised learning algorithms are trained using labeled data.	Unsupervised learning algorithms are trained using unlabeled data.
Supervised learning model takes direct feedback to check if it is predicting correct output or not.	Unsupervised learning model does not take any feedback.
Supervised learning model predicts the output.	Unsupervised learning model finds the hidden patterns in data.
In supervised learning, input data is provided to the model along with the output.	In unsupervised learning, only input data is provided to the model.
The goal of supervised learning is to train the model so that it can predict the output when it is given new data.	The goal of unsupervised learning is to find the hidden patterns and useful insights from the unknown dataset.
Supervised learning needs supervision to train the model.	Unsupervised learning does not need any supervision to train the model.

Supervised learning can be categorized in **Classification** and **Regression** problems.	Unsupervised Learning can be classified in **Clustering** and **Associations** problems.
Supervised learning can be used for those cases where we know the input as well as corresponding outputs.	Unsupervised learning can be used for those cases where we have only input data and no corresponding output data.
It includes various algorithms such as Linear Regression, Logistic Regression, Support Vector Machine, Multi-class Classification, Decision tree, Bayesian Logic, etc.	It includes various algorithms such as Clustering, KNN, and Apriori algorithm.

3. (b)

Bias-variance trade-off:

- Bias: Refers to the underfitting of the model, where the model fails to capture the underlying trend in the data.
- Variance: Refers to the overfitting of the model, where the model captures too much noise in the data and does not generalize well to unseen data.

Achieving good generalized performance requires balancing bias and variance:

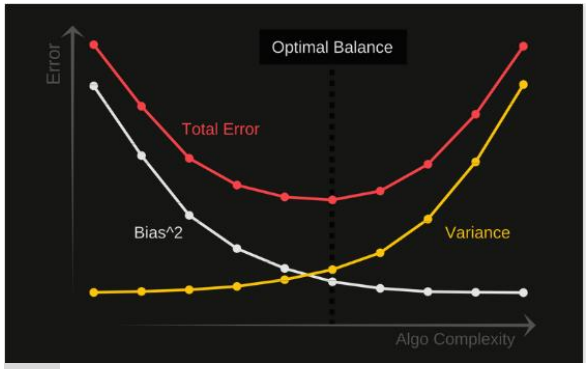
- Low bias: The model is able to capture the underlying trend in the data.
- Low variance: The model is not too sensitive to the specific training data and can generalize well to unseen data.

Visual representation:

Imagine a target function that represents the true relationship between the input and output variables. The goal is to learn a model that approximates this target function as closely as possible.

High bias: The model is represented by a straight line that does not fit the target function well. This indicates underfitting.
Low bias: The model is represented by a curve that closely follows the target function. This indicates good fit.
High variance: The model is represented by a curve that is too wiggly and captures noise in the data. This indicates overfitting.

The ideal scenario is to achieve a model with low bias and low variance, which is represented by a curve that closely follows the target function without being too wiggly.



4 a) Sure, I can help you with the questions in the image:

4. (a) Using K-NN algorithm and the given data set, predict the class label of the test data point (8.5), where K=3 and Euclidean distance.

Data set:

X	Y	Label
4.2	3.8	0
6.5	7.7	1
7.3	8.6	1
5.7	5.9	0
8.0	8.1	1
10.0	6.5	1

Solution:

1. **Calculate the Euclidean distance between the test data point (8.5) and each data point in the training set.**

X	Y	Label	Distance
4.2	3.8	0	4.74
6.5	7.7	1	1.58
7.3	8.6	1	0.28
5.7	5.9	0	2.83
8.0	8.1	1	0.40
10.0	6.5	1	1.70

1. **Sort the data points in ascending order of their distance from the test data point.**

X	Y	Label	Distance
7.3	8.6	1	0.28
8.0	8.1	1	0.40
6.5	7.7	1	1.58
4.2	3.8	0	4.74
5.7	5.9	0	2.83
10.0	6.5	1	1.70

1. **Select the K nearest neighbors (K=3 in this case).**

The three nearest neighbors are:

- (7.3, 8.6) with label 1
- (8.0, 8.1) with label 1
- (6.5, 7.7) with label 1

1. **Predict the class label of the test data point based on the majority vote of the nearest neighbors.**

Since all three nearest neighbors have the label 1, the predicted class label for the test data point (8.5) is **1**.

Therefore, the class label of the test data point (8.5) is 1.

4. (b) Discuss the important features of data partition in cluster analysis and discuss about K-means clustering.

Data partition in cluster analysis:

Data partitioning is a crucial step in many clustering algorithms, including K-means clustering. It involves dividing the data into a specific number of groups (clusters) based on their similarities. Here are some important features of data partition:

- **Number of clusters (K):** This parameter determines the granularity of the clustering and significantly impacts the results. Choosing an appropriate K value is essential for effective cluster formation.
- **Initialization:** The initial assignment of data points to clusters can influence the final clustering outcome. Various initialization methods exist, such as random selection or k-means++, to improve the convergence and quality of clusters.
- **Distance metric:** The chosen distance metric (e.g., Euclidean distance, Manhattan distance) determines how similarity between data points is measured. Selecting an appropriate metric is crucial for capturing the meaningful relationships within the data.

K-means clustering:

K-means clustering is a popular unsupervised learning algorithm for data partitioning. It works by iteratively performing the following steps:

1. **Initialization:** Randomly select K centroids (cluster centers) from the data points.
2. **Assignment:** Assign each data point to the nearest centroid based on the chosen distance metric.
3. **Update:** Recompute the centroid of each cluster by averaging the positions of its assigned data points.
4. **Repeat steps 2 and 3:** Until the centroids no longer change significantly, indicating convergence.

K-means clustering is efficient and easy to implement, making it a widely used technique for various applications like customer segmentation, image segmentation, and anomaly detection. However, it has limitations like being sensitive to the initial centroid selection and not working well with non-spherical clusters.

(a) Fitting a Straight Line Using Least Squares:

Given the data points:
Table

X	Y
5	16
10	19
15	23
20	20
25	30

We want to find the equation of the straight line ($Y = a + bX$) that best fits this data using the method of least squares.

1. Calculate the mean of X and Y:
    1. $\bar{X} = \frac{5 + 10 + 15 + 20 + 25}{5} = 15$
    2. $\bar{Y} = \frac{16 + 19 + 23 + 20 + 30}{5} = 21.6$
2. Calculate the deviations from the mean:
    1. $(X_i - \bar{X})$ and $(Y_i - \bar{Y})$ for each data point.
3. Calculate the product of deviations:
    1. $D = (X_i - \bar{X})(Y_i - \bar{Y})$ for each data point.
4. Calculate the sum of squared deviations of X:
    1. $S_{XX} = \sum (X_i - \bar{X})^2$
5. Calculate the sum of cross-products:
    1. $S_{XY} = \sum D$
6. Calculate the slope (b):
    1. $b = \frac{S_{XY}}{S_{XX}}$
7. Calculate the intercept (a):

14.

1. (a = \bar{Y} - b\bar{X})

Therefore, the equation of the best-fitting straight line is:

[Y = 0.8X + 12.8]

(b) Tuning Parameter (\lambda) in Regularized Linear Regression:

In Regularized Linear Regression (such as Ridge or Lasso), the tuning parameter (\lambda) controls the trade-off between fitting the data well and preventing overfitting. Here's how it works:

- **Ridge Regression (L2 regularization):**
  - Adds a penalty term (\lambda \sum_{j=1}^{p} \beta_j^2) to the loss function.
  - Shrinks the coefficients (\beta_j) towards zero.
  - Helps prevent overfitting by reducing the impact of individual features.
  - Larger (\lambda) leads to more regularization.
- **Lasso Regression (L1 regularization):**
  - Adds a penalty term (\lambda \sum_{j=1}^{p} |\beta_j|) to the loss function.
  - Encourages sparsity (some coefficients become exactly zero).
  - Useful for feature selection.
  - Larger (\lambda) leads to more regularization.

In summary, (\lambda) acts as a regularization parameter that balances model complexity and data fitting. It helps prevent overfitting by controlling the magnitude of the coefficients in the regression model.