1. Consider the following supervised learning problem: the input vector is a 3-dimensional vector where each entry is a positive integer in the range 1-9. The corresponding label is 1 if the sum of the feature dimensions is a prime number, else it is 0. I propose the following mapping as the classifier: if the first dimension is below 5 I will predict 1, else I will predict 0. Calculate the precision and recall of such a classifier (considering 1 as the target class).

   Approach: Total possibilities: 9**3=729. Possible prime numbers: 3,5,7,11,13,17,19,23. Count the number A of ways in which three integers can add up to these numbers: class 1. According to the predictor, number B of positive cases = 4*(9**2). For different values of x1 (1,2,3,4) count the total number C of ways that the total sum can be prime. This is the retrieved set. Precision = C/B, Recall = C/A.

2. Consider the following supervised learning problem: the input vector is a 4-dimensional vector where each entry is a positive integer in the range 1-9. The corresponding label is 1 if the sum of the feature dimensions is even, else it is 0. I propose the following mapping as the classifier: if the largest of the 4 feature values is below 5 I will predict 1, else I will predict 0. Calculate the class-wise and overall accuracy of my classifier.

   Approach: Total possibilities: 9**4=6561. Count the number A of ways in which four integers can add up to even value: class 1. According to the predictor, number B of positive cases = 4**4 (all values are below 5). Count the total number C of ways that the sum of four integers (1,2,3,4) can be even. This is the retrieved set. Precision = C/B, Recall = C/A.

3. Consider the following supervised learning problem: the input vector is a 3-dimensional vector where each entry is a real number in the range (-5,5). The corresponding label is 1 if all dimensions lie in the range (-2,2), else it is 0. I propose the following mapping as the classifier: if the first dimension is in the range (-2,2) I will predict 1, else I will predict 0. Calculate the precision and recall of such a classifier (considering 1 as the target class).

   Approach: Draw geometrically to solve the problem. The set of all values is a cube of volume 10**3, centered at the origin (0,0,0). The class 1 is a smaller cube of volume 4**3, also centered at the origin. The retrieved class has volume 4*(10**2), which subsumes the cube of class 1. Hence, the recall is 1. The precision = (4**3)/(4*(10**2))=0.16.

4. Consider the following supervised learning problem: the input vector is a 2-dimensional vector where each entry is a real number in the range (-5,5). The corresponding label is 1 if the sum of the feature values is positive, else it is 0. I propose the following mapping as the classifier: if the difference between the two feature values is in the range (-1,1) I will predict 1, else I will predict 0. Calculate the precision and recall of such a classifier (considering 1 as the target class).

   Approach: Draw the situation on 2D plane to solve the problem. The feature space is the rectangle bounded by the lines x1=5, x1=-5, x2=5 and x2=05. Half of this rectangle, lying above the line x1+x2=0 is the class 1. Consider the parallel lines x1-x2=-1 and x1-x2=1 and the area bounded between them by the rectangle. This is the retrieved class. Now calculate the overlap between class 1 and the retrieved class, and calculate precision and recall.

5. Consider the following supervised learning problem: the input vector is a 3-dimensional vector where each entry is a real number in the range (-2,2). The corresponding label is the sum of the three feature values. I propose the following mapping as the regressor: the value of the first feature will be the predicted label. Calculate the mean square error.

   Approach: y=x1+x2+x3, h=x1. SE at (x1,x2,x3) = (x2+x3)^2. Overall MSE: integration of SE over x2 (-2,2) and x3 (-2,2), normalized by total volume of the feature space 4**3.

6. Consider the following supervised learning problem: the input vector is a 3-dimensional vector where each entry is a real number in the range (-2,2). The corresponding label is defined by the relation y = x1+2x2+3x3. I propose the following mapping as the regressor: the value of the first feature (x1) will be the predicted label. Calculate the mean square error.

   Approach: Same as (5)

7. Consider the following supervised learning problem: the input vector is a 3-dimensional vector where each entry is a positive integer in the range 1-9. The corresponding label is 1 if the sum of the feature dimensions is a whole square number (eg. 1, 4, 9 etc), else it is 0. I propose the following mapping as the classifier: if the sum is below 10 I will predict 1, else I will predict 0. Calculate the precision and recall of such a classifier (considering 1 as the target class).

   Approach: Total possibilities: 9**3=729. Possible perfect squares: 4,9,16,25. Count the number A of ways in which three integers can add up to these numbers: class 1. According to the predictor, number B of positive cases is the number of ways that three integers add up to 3,4, … 9. Count the total number C of ways that this sum of three integers can be 4 or 9. This is the retrieved set. Precision = C/B, Recall = C/A.

8. Consider the following supervised learning problem: the input vector is a 4-dimensional vector where each entry is a positive integer in the range 1-9. The corresponding label is 1 if the sum of the feature dimensions is even, else it is 0. I propose the following mapping as the classifier: if the smallest of the 4 feature values is below 3 I will predict 1, else I will predict 0. Calculate the class-wise and overall accuracy of my classifier.

   Total possibilities: 9**4=6561. Count the number A of ways in which four integers can add up to even value: class 1. According to the predictor, number B of positive cases = N-negative cases = N-7**4 (all values are in 3-9). Count the total number C of ways that the sum of four integers (3-9) can be even. The size of the retrieved set is then A-C. Precision = (A-C)/B, Recall = (A-C)/A.

9. Consider the following supervised learning problem: the input vector is a 3-dimensional vector where each entry is a real number in the range (-9,9). The corresponding label is 1 if all dimensions are positive, else it is 0. I propose the following mapping as the classifier: if the first dimension is in the range (-2,2) I will predict 1, else I will predict 0. Calculate the precision and recall of such a classifier (considering 1 as the target class).

   Approach: Draw the figure to solve the problem. The feature space is a cube of volume 18**3. The class 1 space is a cube of volume 9**3. The retrieved set has a volume 4*(18**2), which intersects with class 1 in the region 0<x1<2, 0<x2,x3<9, whose volume is 2*(9**2). Now calculate precision and recall.

10. Consider the following supervised learning problem: the input vector is a 2-dimensional vector where each entry is a real number in the range (-9,9). The corresponding label is 1 if the sum of the feature values is positive, else it is 0. I propose the following mapping as the classifier: if both the feature values are positive then I will predict 1, else I will predict 0. Calculate the precision and recall of such a classifier (considering 1 as the target class).

    Approach: Draw figure to solve the problem. Feature space: the rectangle bounded by the lines x1=-9, x1=9, x2=-9, x2=9. Class 1: area bounded by this rectangle above the line x1+x2=0. The retrieved class: the part of the rectangle lying in the first quadrant, which is subsumed by class 1. Hence precision=1, recall= 0.5.

11. Consider the following supervised learning problem: the input vector is a 4-dimensional vector where each entry is a real number in the range (-1,1). The corresponding label is the sum of the four feature values. I propose the following mapping as the regressor: the value of the first feature will be the predicted label. Calculate the mean square error.

    Approach: similar to (5).

12. Consider the following supervised learning problem: the input vector is a 4-dimensional vector where each entry is a real number in the range (-2,2). The corresponding label is defined by the relation y = x1+2x2+3x3+4x4. I propose the following mapping as the regressor: 10 times the value of the first feature (10x1) will be the predicted label. Calculate the mean square error.

    Approach: similar to (5)

13. Consider the following small dataset. There are 9 training examples with 2-dimensional feature vectors. Unfortunately, some of the feature values are missing. Additionally, you know a property of this dataset: if two points have a Euclidean distance of below 2, their class labels are definitely the same. You need to predict the label of the test example using the K-NN classification, and also estimate the missing value (in any way you like). Do the results change if you use K=1 and K=3? [Assume equal weight of all neighbors]

| ID | X1 | X2 | Y | ID | X1 | X2 | Y | ID | X1 | X2 | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | -1 | 1 | A | 2 | -1 | 0 | A | 3 | -2 | ? | A |
| 4 | 3 | 4 | B | 5 | ? | 3 | B | 6 | 6 | 3 | B |
| 7 | -3 | 0 | A | 8 | -4 | 2 | B | 9 | ? | 2 | B |
| Test | 0 | 2 | ? | | | | | | | | |

    Approach: Fill up the missing values in several ways. One way is to take the mean of feature values of all other points in same class, while making sure that the given property is satisfied. Note that the property does not mean that if two points have same label then they must be close, but it means that if two points are in different classes, they cannot be closer than 2.

14. Consider the following small dataset. There are 9 training examples with 3-dimensional feature vectors. You need to predict the label of the test example using the K-NN classification. How do the results change as you change K? Assume weight of each neighbor as inversely proportional to Euclidean distance from test point.

| ID | X1 | X2 | Y | ID | X1 | X2 | Y | ID | X1 | X2 | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | -1 | 1 | A | 2 | -1 | 0 | A | 3 | -2 | -1 | A |
| 4 | 3 | 4 | B | 5 | 1 | 3 | B | 6 | 6 | 3 | B |
| 7 | -3 | 0 | A | 8 | -4 | 2 | B | 9 | 1 | 2 | B |
| Test | 0 | 2 | ? | | | | | | | | |

Approach: Increase K from 1 to at least 4 or 5. For each K, consider weighted means with normalization, i.e. sum of weights must be 1.

15. Consider the following small dataset. There are 9 training examples with 3-dimensional feature vectors. You need to predict the label of the test example using the 3-NN classification. How do the results change as you use Manhattan distance instead of Euclidean distance? Assume weight of each neighbor as inversely proportional to the distance from test point.

| ID | X1 | X2 | Y | ID | X1 | X2 | Y | ID | X1 | X2 | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | -1 | 1 | A | 2 | -1 | 0 | A | 3 | -2 | -1 | A |
| 4 | 3 | 4 | B | 5 | 1 | 3 | B | 6 | 6 | 3 | B |
| 7 | -3 | 0 | A | 8 | -4 | 2 | B | 9 | 1 | 2 | B |
| Test | 0 | 2 | ? | | | | | | | | |

Approach: consider weighted means with normalization, i.e. sum of weights must be 1.

16. Consider the following dataset of 2-dimensional examples. Apply the 3-nearest-neighbor algorithm to identify if any point in the training set is an outlier, i.e. labelled differently from its surrounding points.

| ID | X1 | X2 | Y | ID | X1 | X2 | Y | ID | X1 | X2 | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | -1 | 1 | A | 2 | -1 | 0 | A | 3 | -2 | -1 | A |
| 4 | 3 | 4 | B | 5 | 1 | 3 | A | 6 | 6 | 3 | B |
| 7 | -3 | 0 | A | 8 | -4 | 2 | B | 9 | 1 | 2 | B |

17. Consider the following dataset of 2-dimensional examples. Apply the 1-nearest neighbor algorithm to classify the test point. Now, identify some 2-3 clusters (in any way you like) of the training examples, and use 1-nearest neighbor algorithm to classify the test point using the cluster means.

| ID | X1 | X2 | Y | ID | X1 | X2 | Y | ID | X1 | X2 | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | -1 | 1 | A | 2 | -1 | 0 | A | 3 | -2 | -1 | A |
| 4 | 3 | 4 | B | 5 | 1 | 3 | B | 6 | 6 | 3 | B |
| 7 | -3 | 0 | A | 8 | -4 | 2 | B | 9 | 1 | 2 | B |
| Test | 0 | 2 | ? | | | | | | | | |

Approach: define clusters in any way, eg. by plotting on 2D plain. Test point cannot be part of a cluster. But must mention how to classify the test point based on clusters. One way is to calculate the centroids (means) of the clusters, and measure distance of test point from them. Another way is to compute average distance of the test point to different training points in each cluster.

18. We may use K-NN algorithm for regression. In the following dataset of 3-dimensional examples, use the weighted mean of the K neighbors as the predicted label of any point. Use the inverse Euclidean distance as the weight of each neighbor. What is the RMSE on the training set for K=1 and K=2?

| ID | X1 | X2 | X3 | Y | ID | X1 | X2 | X3 | Y |
|---|---|---|---|---|---|---|---|---|---|
| 1 | -1 | 1 | 0 | 1 | 2 | -1 | 0 | 3 | 5 |
| 3 | -2 | -1 | 2 | -1 | 4 | 3 | 4 | -2 | 11 |
| 5 | 1 | 3 | 1 | 11 | 6 | -4 | 2 | 0 | -3 |

Approach: Predict Y of any point as the weighted mean of the Y-values of its K nearest neighbor points. The weights should be normalized.

19. Consider the following small dataset. There are 9 training examples with 2-dimensional feature vectors. Unfortunately, some of the feature values are missing. Additionally, you know a property of this dataset: if two points have a Euclidean distance of below 2, their class labels are definitely the same. You need to predict the label of the test example using the K-NN classification, and also estimate the missing value (in any way you like). Do the results change if you use K=1 and K=3? [Assume equal weight of all neighbors]

| ID | X1 | X2 | Y | ID | X1 | X2 | Y | ID | X1 | X2 | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | -2 | 0 | A | 2 | -1 | -3 | A | 3 | -2 | ? | A |
| 4 | 3 | 1 | B | 5 | ? | 3 | B | 6 | 6 | 3 | B |
| 7 | 3 | 0 | B | 8 | -4 | 2 | A | 9 | ? | 2 | A |
| Test | 0 | 2 | ? | | | | | | | | |

Approach: same as 13

20. Consider the following small dataset. There are 9 training examples with 3-dimensional feature vectors. You need to predict the label of the test example using the K-NN classification. How do the results change as you change K? Assume weight of each neighbor as inversely proportional to Euclidean distance from test point.

| ID | X1 | X2 | Y | ID | X1 | X2 | Y | ID | X1 | X2 | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | -2 | 0 | A | 2 | -1 | -3 | A | 3 | -2 | 5 | A |
| 4 | 3 | 1 | B | 5 | 0 | 3 | B | 6 | 6 | 3 | B |
| 7 | 3 | 4 | B | 8 | -4 | 2 | A | 9 | -5 | 2 | A |
| Test | 0 | -2 | ? | | | | | | | | |

21. Consider the following small dataset. There are 9 training examples with 3-dimensional feature vectors. You need to predict the label of the test example using the 3-NN classification. How do the results change as you use Manhattan distance instead of Euclidean distance? Assume weight of each neighbor as inversely proportional to the distance from test point.

| ID | X1 | X2 | Y | ID | X1 | X2 | Y | ID | X1 | X2 | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | -2 | 0 | A | 2 | -1 | -3 | A | 3 | -2 | 5 | A |
| 4 | 3 | 1 | B | 5 | 0 | 3 | B | 6 | 6 | 3 | B |
| 7 | 3 | 4 | B | 8 | -4 | 2 | A | 9 | -5 | 2 | A |
| Test | 0 | 2 | ? | | | | | | | | |

22. Consider the following dataset of 2-dimensional examples. Apply the 3-nearest-neighbor algorithm to identify if any point in the training set is an outlier, i.e. labelled differently from its surrounding points.

| ID | X1 | X2 | Y | ID | X1 | X2 | Y | ID | X1 | X2 | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | -2 | 0 | A | 2 | -1 | -3 | A | 3 | 1 | 5 | B |
| 4 | 3 | 1 | B | 5 | 0 | 3 | B | 6 | 6 | 3 | B |
| 7 | 3 | 4 | A | 8 | -4 | 2 | A | 9 | -5 | 2 | A |

23. Consider the following dataset of 2-dimensional examples. Apply the 1-nearest neighbor algorithm to classify the test point. Now, identify some 2-3 clusters (in any way you like) of the training examples, and use 1-nearest neighbor algorithm to classify the test point using the cluster means.

| ID | X1 | X2 | Y | ID | X1 | X2 | Y | ID | X1 | X2 | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | -2 | 0 | A | 2 | -1 | -3 | A | 3 | -2 | 5 | A |
| 4 | 3 | 1 | B | 5 | 0 | 3 | B | 6 | 6 | 3 | B |
| 7 | 3 | 4 | B | 8 | -4 | 2 | A | 9 | -5 | 2 | A |
| Test | 0 | 2 | ? | | | | | | | | |

24. We may use K-NN algorithm for regression. In the following dataset of 3-dimensional examples, use the weighted mean of the K neighbors as the predicted label of any point. Use the inverse Euclidean distance as the weight of each neighbor. What is the RMSE on the training set for K=1 and K=2?

| ID | X1 | X2 | X3 | Y | ID | X1 | X2 | X3 | Y |
|---|---|---|---|---|---|---|---|---|---|
| 1 | -1 | 1 | 0 | 1 | 2 | -1 | 0 | 3 | 2 |
| 3 | -2 | -1 | 2 | -2 | 4 | 3 | 4 | -2 | 9 |
| 5 | 1 | 3 | 1 | 8 | 6 | -4 | 2 | 0 | 0 |

25. Consider the following dataset with three attributes. Which attribute will you choose to construct a Decision Stump and what will be its accuracy on the training set? Use information gain as splitting criteria. [Choose any splitting threshold that you find suitable for stump]

| ID No | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| X1 | -1 | 5 | -5 | -3 | 6 | 2 | -5 | 2 |
| X2 | 4 | 12 | -2 | -5 | 10 | 6 | 8 | 11 |
| X3 | 2 | -1 | 8 | 3 | -6 | -2 | 0 | 2 |
| Y | A | B | A | A | B | A | B | B |

Approach: must choose a threshold for each feature with some justification (eg. by plotting) and compare IG values of the different features.

26. Consider the following dataset to construct a decision tree of depth 3 (i.e. 2 rounds of splitting) using information gain as splitting criteria. Use any 6 examples for training and the rest for testing. What are the accuracies on the training and testing sets?

    Approach: Calculate IGs of different features and decide accordingly at both steps.

| ID No | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| X1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 |
| X2 | 1 | 1 | 2 | 2 | 1 | 1 | 2 | 2 |
| X3 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 |
| Y | A | A | B | A | B | B | A | B |

27. You are given 7 training datapoints with 3-dimensional features and real-valued labels. Create a regression stump by identifying the most discriminative feature, as done in decision tree. Use "variance" instead of "entropy" as a measure of homogeneity of a split dataset. What will be the predicted label on the test datapoint (ID No 8)?

    Warning: calculate the mean and variance of Y, not of X1/X2/X3 at any stage.

| ID No | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| X1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 |
| X2 | 1 | 1 | 2 | 2 | 1 | 1 | 2 | 2 |
| X3 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 |
| Y | 1.3 | 2.1 | 2.8 | 4.9 | 0.8 | 1.7 | 5.1 | |

28. Compare the results of 1-NN classification and Decision Stump (based on information gain) classification on the following dataset. [Choose any splitting threshold that you find suitable for stump]

| ID No | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| X1 | -1 | 5 | -5 | -3 | 6 | 2 | -5 | 2 | 2 | 3 |
| X2 | 4 | 12 | -2 | -5 | 10 | 6 | 8 | 11 | 8 | 5 |
| Y | A | B | A | A | B | A | B | B | ? | ? |

29. Compare the results of 3-NN regression and Regression Stump (based on variance) on the following dataset. [Choose any splitting threshold that you find suitable for stump]

Warning: calculate the mean and variance of Y, not of X1/X2 at any stage.

| ID No | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| X1 | -1 | 5 | -5 | -3 | 6 | 2 | -5 | 2 | 2 | 3 |
| X2 | 4 | 12 | -2 | -5 | 10 | 6 | 8 | 11 | 8 | 5 |
| Y | 2 | 22 | -12 | -13 | 22 | 10 | -2 | 15 | ? | ? |

30. Consider the following dataset. How does the classification accuracy on the test set change as we keep increasing the branching depth of a decision tree?

Approach: at each branching stage, compute information gain with respect to different features and compare them.

| ID No | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 (TEST) | 10 (TEST) |
|---|---|---|---|---|---|---|---|---|---|---|
| X1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 1 | 2 |
| X2 | A | A | B | B | A | A | B | B | B | A |
| X3 | 5 | 7 | 6 | -2 | 0 | 7 | 5 | 3 | 8 | -1 |
| Y | X | X | Y | X | Y | X | Y | Y | Y | Y |