True / False, Multiple Choice

- Linear regression is a useful model to make predictions, but is limited by the fact that we are unable to interpret the model to make inferences about the relationships between features and the output.

  True / **False**

- When given the choice of two models, the one that has smaller training error will always have smaller true error.

  True / **False**

- We expect a model with high variance to generalize better than a model with high bias.

  True / **False**

- When we use the same model complexity on a smaller dataset, overfitting is more likely.

  True / False

- In machine learning, bias is always a bigger source of error than variance

  True / **False**

- Given an infinite amount of noiseless training data, we expect the training error for decision stumps to go to 0.

  True / **False**

- As the number of iterations goes to infinity, boosting is guaranteed to reach zero training error.

  True / False

- Increasing k in k-NN increases bias and decreases variance.

  True / False

- To determine best value of k for k-means, it's sufficient to run the k-means algorithm once for each value of k you want to try.

  True / **False**

- Increasing the number of recommended items is more likely to increase the recall than decrease it.

  True / False

- With a large dataset, nearest neighbors is more efficient at test time than logistic regression.
  True / **False**

- K-means converges to a global optimum for the heterogeneity object.
  True / **False**

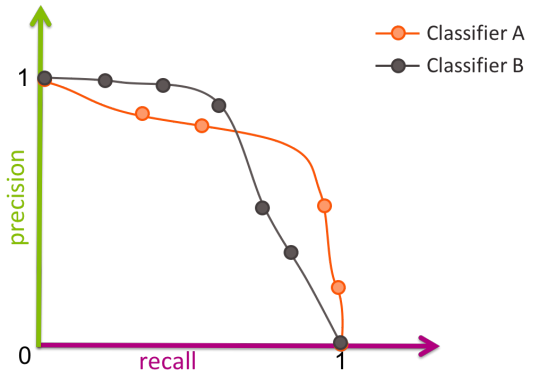
- To determine the best value of k for k-means, you only need to run the k-means algorithm once for each value of k you want to try.
  True / **False**

- Select all the following reasons why we would use LASSO over Ridge?
  - **It can help us identify which features are important**
  - It is faster to learn the weights for LASSO than for Ridge
  - LASSO usually achieves lower generalization error than Ridge
  - **If there are many many features, the model learned using LASSO can make predictions more efficient**

- To find the best set of coefficients for logistic regression, we use gradient descent to minimize the number of examples misclassified.
  True / **False**

- Which of the following are symptoms of a logistic regression model being overfit? Select all that apply
  - **Large estimated coefficients**
  - Good generalization to unseen data
  - Simple decision boundary
  - **Complex decision boundary**
  - **Overconfident predictions of class probabilities**

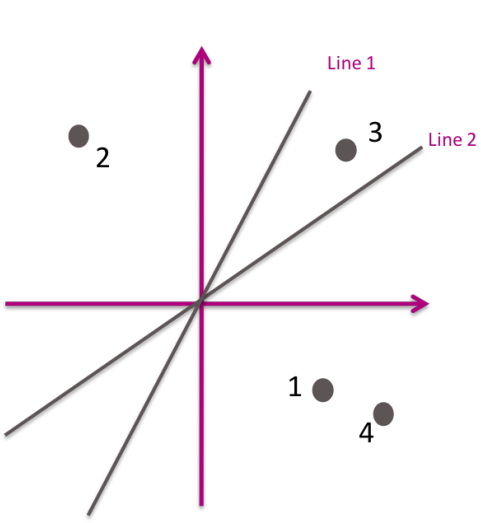
- For the following precision-recall curves for varying classification thresholds on different models A and B, which is the best model?
  - Model A
  - Model B
  - **Depends on the situation**

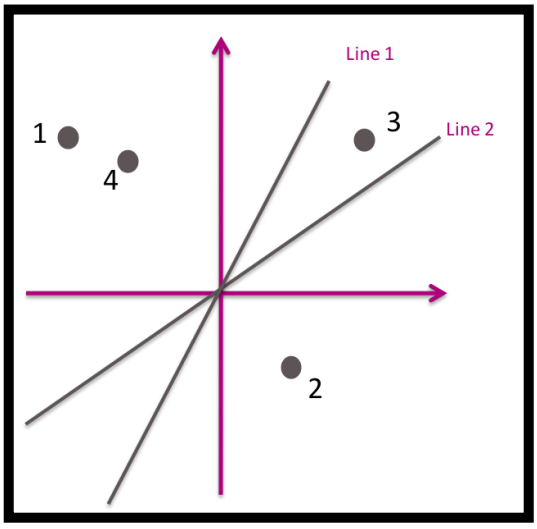


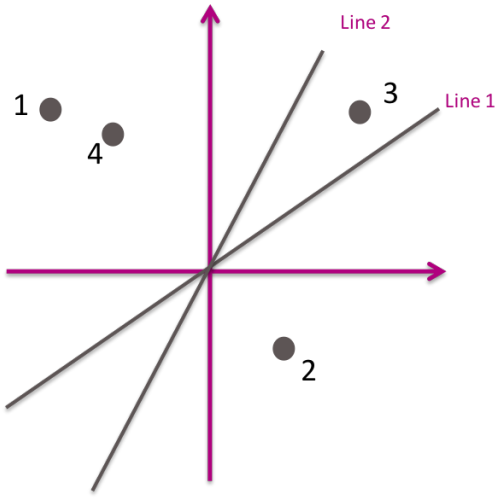
- Suppose we had the following bins for Locality Sensitive Hashing

Bin Index	0 0	0 1	1 0	11
Point Labels	{1, 4}		{3}	{2}

Which of the following plots show the lines and points that would result in this binning?







Short response

- Give examples of 3 examples of classifiers we have covered in class. Also give 3 examples of evaluation metrics we have used for classification tasks.

  Logistic Regression, Decision Trees, k-NN are some examples

  Accuracy, precision-recall, precision at k are some examples

- Describe for Ridge regression, how choosing a regularization penalty lambda that is too small will affect the following quantities. Assume, before adding the regularizer, we are using a model that is too complex for the amount of data we have.
    - Training error
      Training error will be low since the model can overfit without penalty.

    - Test error
      Test error will be high since the model can overfit to the training data.

    - Magnitude of coefficients
      The coefficients will be large since there is no penalty.

    - Number of 0 coefficients
      Coefficients are likely to be not zero since they are allowed to be large without penalty.

- Describe for Ridge regression, how choosing a regularization penalty lambda that is too large will affect the following quantities. Assume, before adding the regularizer, we are using a model that is too complex for the amount of data we have.
    - Training error
      Training error will be high since the model is penalized too much to make a hypothesis that reasonably matches the training data.

    - Test error
      Test error will be high for the same reason as training error being high.

    - Magnitude of coefficients
      The coefficients will have small magnitude since they are penalized highly for being large.

    - Number of 0 coefficients
      Ridge does not favor 0 coefficients so we would not expect any to be 0, just small.

- Describe one benefit and one drawback to using k-means++ instead of k-means.
  Benefit: k-means++ generally finds better clusterings with the better initialization
  Drawback: k-means++ takes a lot longer to initialize than k-means.

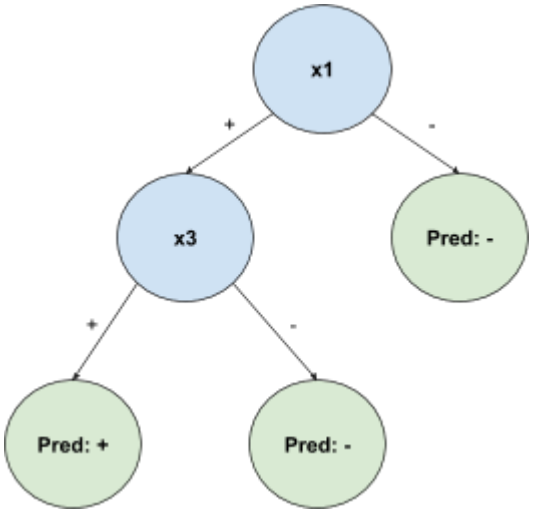
- What does PCA try to minimize in order to reduce the dimensionality of the dataset?
  Reconstruction error

Short work

- For the following dataset, construct a decision tree that the decision tree algorithm we discussed in class.

Training Data

x_1	x_2	x_3	y
+	−	−	−
+	−	+	+
+	+	+	+
−	−	+	−

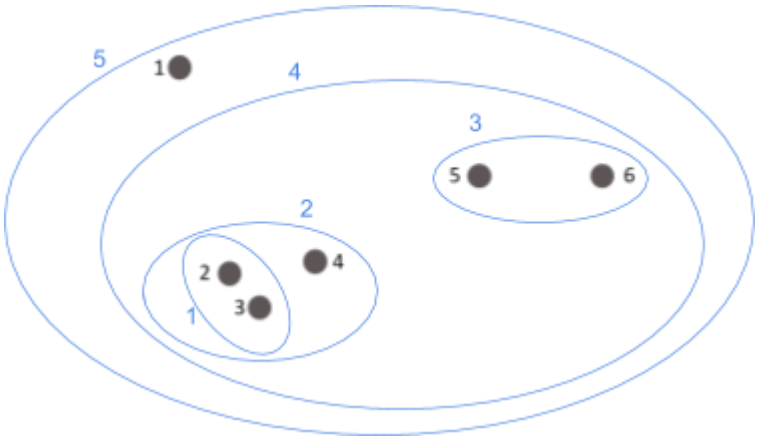


- Here is pseudocode that describes the decision tree algorithm for data where each feature takes on one of two values. We want to modify this pseudocode to allow for weighted datasets (like from boosting). Please circle which parts of the pseudocode must change and describe what needs to be changed for those parts. You don't need to write new pseudocode, just describe in a sentence what change needs to be made for each part you identify.

```
1:  function DecisionTree(Data)
2:      if all examples in Data have same label y then
3:          return Leaf(y)
4:      else
5:          for each feature h_i do
6:              Data_1, Data_2 = Split(Data, h_i)
7:              Error_i = ClassificationError(Data_1) + ClassificationError(Data_2)
8:          end for
9:          h* = choose feature h_i that has smallest Error_i
10:         Data_1, Data_2 = Split(Data, h*)
11:         return Branch(h*, DecisionTree(Data_1), DecisionTree(Data_2))
12:     end if
13: end function
```

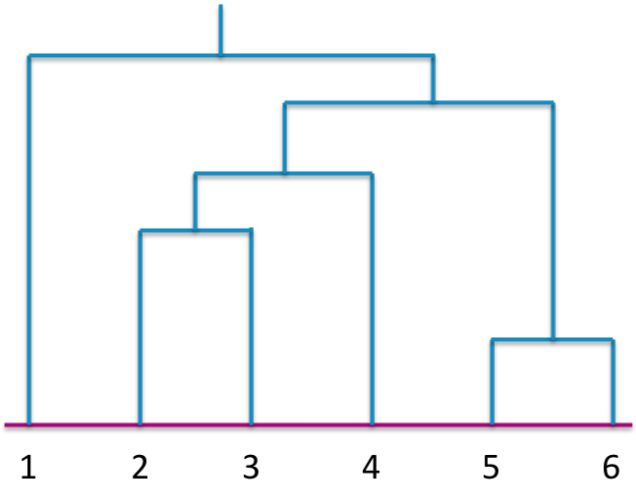
Need to compute weighted sum of misclassified points

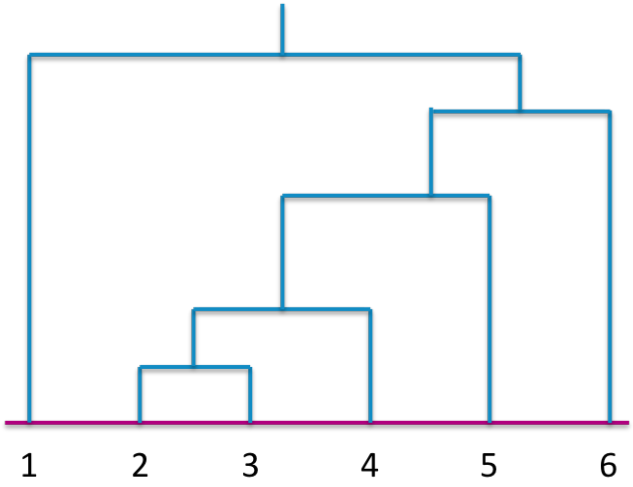
- For the following dataset

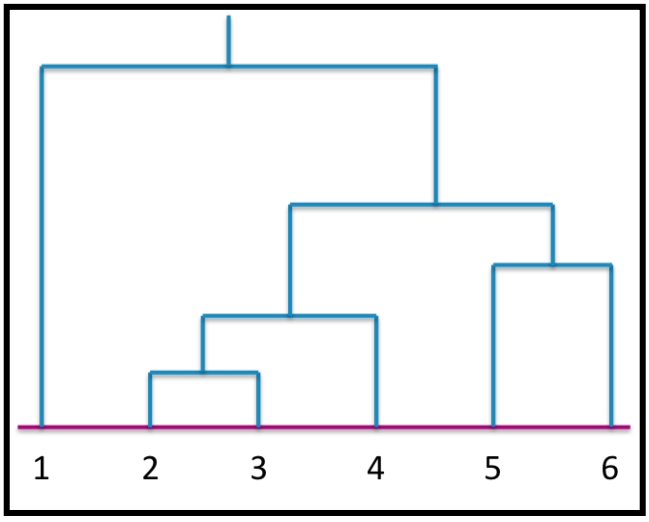


  - Draw the order the clusters are formed using agglomerative hierarchical clustering with single linkage. Circle the clusters and label them 1, 2, 3, 4, etc in the order the clusters are joined. Make sure to label the clusters clearly so it is not ambiguous. You do not need to show the stage when the points are in their own clusters to start.

  - Which of the following dendrograms matches the order in which the clusters are combined?

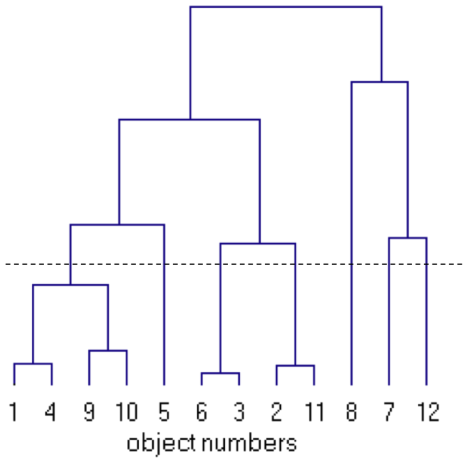


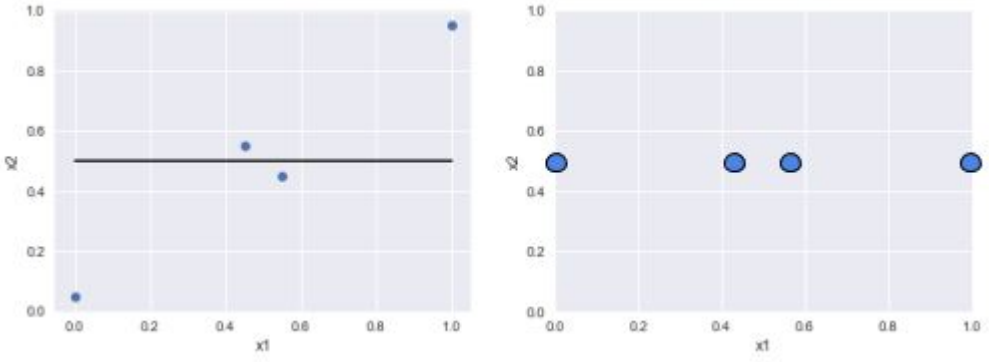


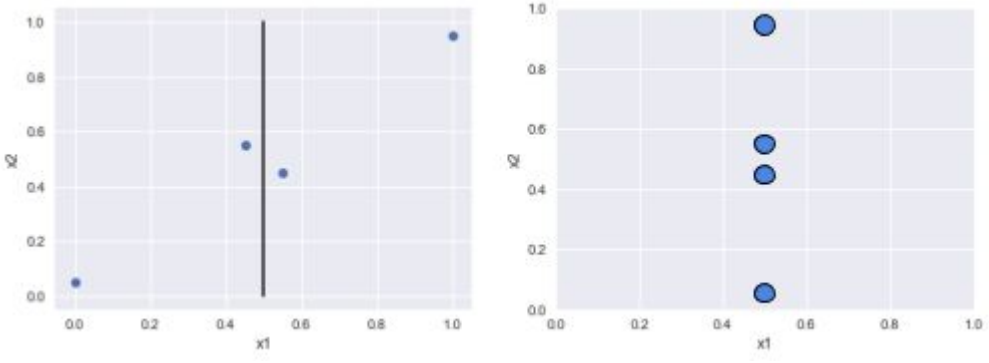


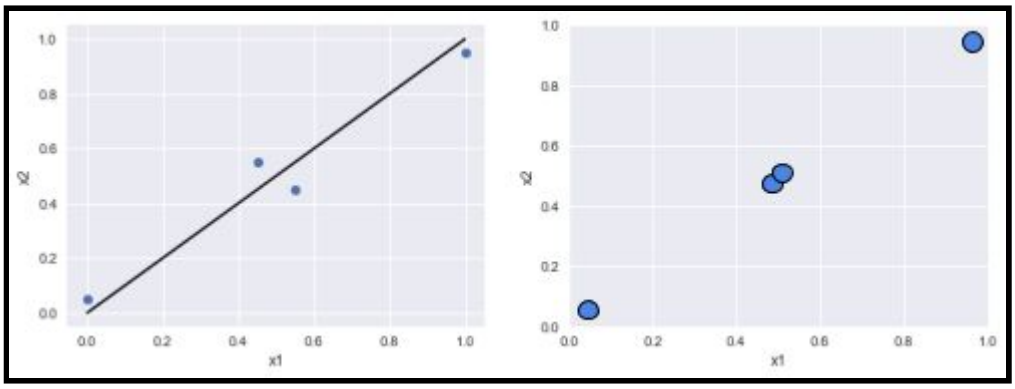
- Consider the following dendrogram (from a different dataset). How many clusters would we report if we were to slice the dendrogram
  with the dotted line on the tree.

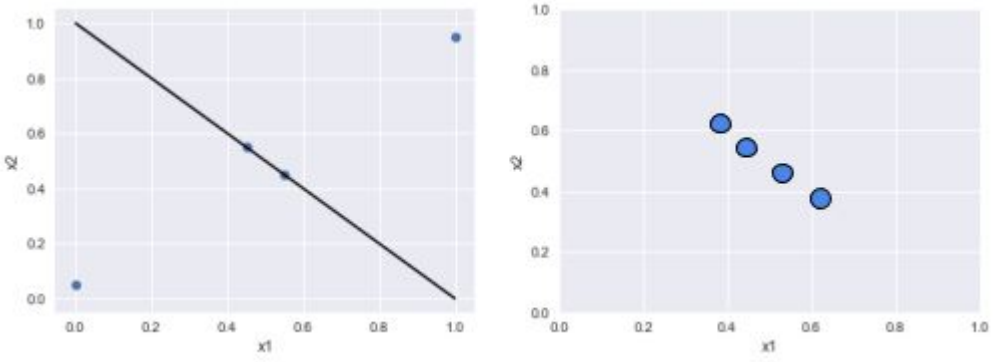
7



- For the following dataset we show 4 possible candidate lines to project the data onto for PCA. For each line, draw the reconstructed dataset after projecting to that line. Also identify which line we would use and state why.
  - 

  - 

- 

  - Minimizes reconstruction error
  - 

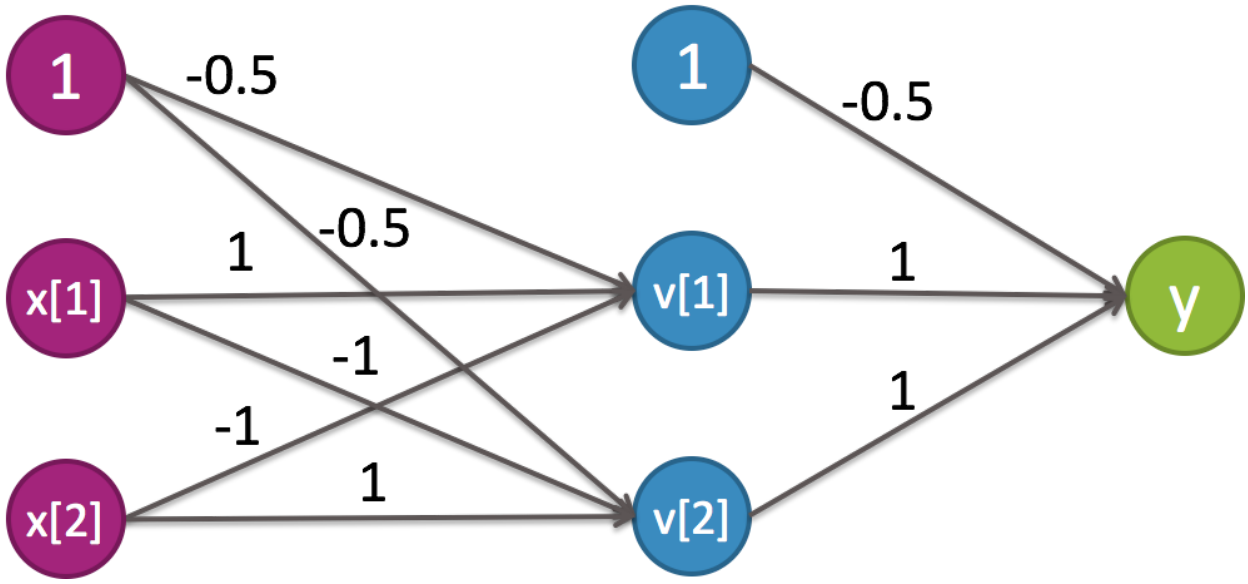
- Suppose we have a movie library consisting of 3 different movies and our system has 4 registered users. Also suppose we have already computed our best estimate for the ratings matrix factorization task and the matrices learned for users and movies are as shown below. Using this matrices, what movie would we recommend to User 2?

			Movie 1	Movie 2	Movie 3
User 1	1	2	1	2	1
User 2	1	0	0	1	2
User 3	0	2			
User 4	1	1			

$$\widehat{Rating}(U_2, M_1) = 1 * 1 + 0 * 0 = 1$$
$$\widehat{Rating}(U_2, M_2) = 1 * 2 + 0 * 1 = 2$$
$$\widehat{Rating}(U_2, M_3) = 1 * 1 + 0 * 2 = 1$$

We would recommend movie 2 because it has the highest predicted rating

- Suppose we have the following neural network, what would the output of the network be if we gave it the input x = [0, 1]. For simplicity, use the ReLu activation function g(x) = max(0, x) for every neuron in the network.



$$
\begin{aligned}
v[1] &= g(-0.5 + 1 * x[1] - 1 * x[2]) \\
&= g(-0.5 + 0 - 1) \\
&= g(-1.5) \\
&= max(0, -1.5) \\
&= 0 \\
v[2] &= g(-0.5 - 1 * x[1] + 1 * x[2]) \\
&= g(-0.5 - 0 + 1) \\
&= g(0.5) \\
&= max(0, 0.5) \\
&= 0.5 \\
y &= g(-0.5 + 1 * v[1] + 1 * v[2]) \\
&= g(-0.5 + 0 + 0.5) \\
&= g(0) \\
&= max(0, 0) \\
&= 0
\end{aligned}
$$

- For the following training dataset, draw the function learned by k-NN regression for the specified values of k.