1. Logistic Regression
   1. We assume that the probability threshold is 0.5.  
      x will vote for party A if

x will vote for party B if

x will vote for party A or B with equal probability when

* 1. In general, g(x) can defined as a dot product of the weight vector w and the input x, i.e.,  
     Following the same procedure in c, a threshold, which is usually 0.5, of the probability p must first be defined to obtain a decision boundary of a more general form. We can expand the logistic regression model into an n-class classification model by applying n different g(x) functions where each g(x) computes the probability that x belongs to the corresponding class. As for the loss function, the original loss can be replaced by the cross-entropy loss function for classification problem as described in the lecture notes. The final model would be a feedforward neural network where the output layer has n nodes instead of one node only as in logistic regression model. This network allows us to predict which class x comes from out of the n class instead of just 2 class as in logistic regression model. The following functions defines the overall new model.  
     In the end, to classify x into one of the n classes, we could simply pick an i that maximize or .

1. Logistic Regression
   1. z will approach one-hot vector such that zj approaches 1 when , otherwise it will approach 0
   2. z will approach a uniform value, i.e., 1/K for all zj
   3. Case 1 ():  
      Case 2 ():  
      Combining the two cases, we have
2. Feedforward Neural Network  
   We can regard this new variant of classification problem as a K binary classification problem instead of K-class classification problem. Here, we assume that the y is a zero vector when x belongs to none of the K classes.  
   1. While there may be many ways to modify the original network to handle this variant of the classification problem, one approach is to use the original activation function and , but modify to be the logistic regression for binary classification as in the previous lecture notes in logistic regression, such as the logistic function below  
      This modification is sufficient to solve the new variant of problem given that we use the appropriate objective function below to perform K binary classification.
   2. Following the modification in a, we need to change the original loss function, which is sometimes called categorical cross-entropy loss, defined in lecture notes to the (Bernoulli or binary) cross entropy loss used in previous lecture notes in logistic regression, such as
   3. After defining a new network for K binary classification problem, we use the to classify x into class i given that is greater than 0.5 (as the default threshold). Otherwise, we will indicate ‘none of the above’ or ‘reject’.
3. Convolutional Neural Networks
   1. Conv (128, 32, 7, 7, 2)

      2. Conv1 (128, 16, 1, 1, 1)
      3. Conv1 (128, 16, 1, 1, 1)
      4. Conv2 (16, 32, 7, 7, 2)
      5. Conv2 (16, 32, 7, 7, 2)
      6. Conv1 + Conv2
4. Principal Component Analysis
   1. 100%. In other words, the variance is completely preserved. By using two principal components obtained by PCA to project S linearly onto another two-dimensional space spanned by the principal components together, 100% of the total variance can be explained by the two principal components together since S also in a two-dimensional space. Thus, there is no dimensionality reduction and it is equivalent to linear transformation that maps S to the same space.
   2. In PCA, the total variance is the sum of the variance of the principal component which is the eigenvalue as shown in the lecture notes. The maximum percentage of total variance that can be explained is expressed by the sum of eigenvalues of the used principal components divided by the sum of eigenvalues of all principal components. By using one principal components, the maximum percentage of total variance is .
5. Clustering – Partitional Clustering
   1. For simplicity, we partition S into k clusters or k new sets of data points  
      which contains all data points that belongs to cluster i according to b with i = 1, …, k. By rewriting the objective function, we have  
      Minimizing the original objective function is equivalent to minimizing the mean squared distance for each individual cluster given by  
      Taking the derivative, we have  
      Thus, which corresponds to the cluster mean obtained after updating according to the local optimization algorithm.
   2. Denote S1 and S2 as the cluster 1 and cluster 2 with mean m1 and m2 respectively.  
      S1 = {(x, y) | 1<x<9} S2 = {(x, y) | 11<x<17} m1 = (5, 1.5) m2 = (14, 1.5)  
      S1 = {(x, y) | 1<x<9} S2 = {(x, y) | 11<x<17} m1 = (5, 1.5) m2 = (14, 1.5)  
      🡺 stop  
      S1 has 10 data points while S2 has 8 data points.
6. Clustering – Hierarchical Clustering
   1. Diagram, engineering drawing

      Description automatically generated  
      three top-most level clusters: {2, 4, 5}, {20, 25}, and {39, 43, 44}
   2. Diagram, engineering drawing

      Description automatically generated   
      three top-most level clusters: {2, 4, 5}, {20, 25}, and {39, 43, 44}
   3. The two dendrograms have different hierarchical structure. However, the two partitions with three subsets each are the same.