Solutions

1. P(W=F|CLASS=T)=(1+1)/(2+1*2)=2/4=1/2
   P(W=F|CLASS=F)=(1+1)/(3+1*2)=2/5
   P(X=F|CLASS=T)=(1+1)/(2+1*2)=2/4=1/2
   P(X=F|CLASS=F)=(2+1)/(3+1*2)=3/5
   P(Y=T|CLASS=T)=(1+1)/(2+1*2)=2/4=1/2
   P(Y=T|CLASS=F)=(2+1)/(3+1*2)=3/5
   P(CLASS=T) = 2/5
   P(CLASS=F) = 3/5
   1/2 *1/2*1/2*2/5 = 1/20 = 0.05
   2/5*3/5*3/5*3/5 = 0.08

   The classification is CLASS = F

   If we add 100 new instances of CLASS = T, the priors change
   1/2 *1/2*1/2*102/105 = 0.12
   2/5*3/5*3/5*3/105 = 0.004
   Therefore, the classification changes and becomes CLASS = T

2. No, NB is not a good probability estimator due to 2 reasons
   a. It cannot compute the normalization factor and therefore only gives a value proportional to P(y|x)
   b. Even the un-normalized value that it generates is not accurate due to the Naïve assumption of conditional independence of features

3. Neural networks is the best choice here since it can represent any function and can handle continuous features.

4. 1-NN (there can be other right answers depending upon how you break ties)
   If I break the tie using the smaller value, i.e. suppose the instance is equidistant from two instances with values 3 and 4, and here I choose 3 as its neighbor
   Error = 2/10 = 0.2 (It makes mistakes for the fourth and eighth instances)

   3-NN (again can have other correct answers)
   Error = 2/10 = 0.2 (it makes mistakes for the fourth and eighth instances)

5. in 1-NN, the accuracy is 100% since the nearest neighbor of each point is itself. However for 3-NN this is not guaranteed. For example imagine a dataset where a “+” point is surrounded by “-“ points.

6. In KNNs, the learning phase just stores the training data, and in prediction/testing we use all the training data. Therefore, more training data implies more complexity for

predictions. In NB, during training, we store only the probabilities and not the training data. During predictions/testing, we simply use the learned probabilities. Thus, the computational complexity during predictions is not dependent on the size of the training data. Neural networks again predictions is easy but training is hard.

7. Yes neural networks can represent any function and that includes all Boolean functions. However, this does not always mean Neural networks are better since they may overfit the data.