**Question 1**

* We need to classify the following test example: (W=F, X=F, Y=T) using Naïve Bayes. Therefore, we need to compare the following probabilities under those assumptions:

1. P(W=F|C=T) \* P(X=F|C=T) \* P(Y=T|C=T) \* P(C=T)
2. P(W=F|C=F) \* P(X=F|C=F) \* P(Y=T|C=F) \* P(C=F)

Since, we must do this using Laplace-1 correction, we’re adding one instance to each of our features. So, for (1) we have 2/3 \* 2/3 \* 2/3 \* 3/6 = 12/81. While, for (2) we have 2/4 \* 3/4 \* 3/4 \* 4/6 = 3/16. Since. (2) > (1) we must classify this example as false.

* Just adding hundred more instances of class T, would skew the data towards class T. In our example, it would mean that P(C=F) would now be extremely small, and since these are products of fractions, overall probability would be diminutive. Therefore, it would be classified as True.

**Question 2**

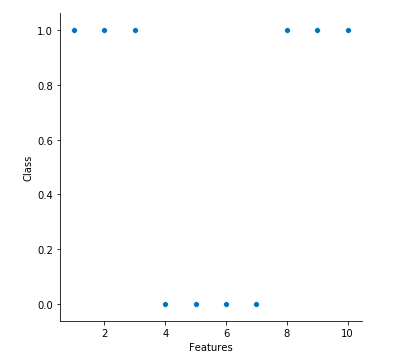
Naïve Bayes does not accurately calculate the class given features, i.e., P(*y*|X). Using Bayes’ formula, from P(*y*|Xi), we get ∏P(Xi|*y*) \* P(*y*). Therefore, the assumption is that the conditional probabilities are independent, an assumption that is real life data would be consistently violated. Secondly, the main issue with the Bayes algorithm is the value of P(X) which is nigh impossible to calculate, so its proportional rather than accurate.

**Question 3**

As the data set is not linearly separable, and contains continuous variables as features, Neural Networks will be the best algorithm to use in this instance. Unlike Decision Trees and Perceptron, it can learn highly complex functions which is almost necessary for data that is not linearly separable. It is also preferable for continuous data because of the way its trained, gradient descent for example, whereas Naïve Bayes is not applicable in the case of continuous data.

**Question 4**

Let’s plot the table for a better understanding



1. For NN-1, as it can be observed, leave-one-out cross validation error would be 20%. The data points will be correctly classified as the nearest neighbor is of the same class, except Feature 4 (nearest neighbor Feature 3) and Feature 8 (nearest neighbor Feature 7).
2. For NN-3, it may not be so obvious, but leave-one-out cross validation error will be 20%. Feature 4 and Feature 8 are of particular interest. The rest are fairly straightforward and correctly classified. For Feature 4 the 3-NNs are Feature 2 & Feature 3 and Feature 5, therefore it will be incorrectly classified. For Feature 8, Feature 6 and Feature 7 will ensure that it is incorrectly classified.

**Question 5**

Since, it is memory based when it comes to training, 1-NN will have 100% accuracy.

It cannot be guaranteed that accuracy on the training dataset, D, will be 100% accurate for 3-NN.

**Question 6**

1. For kNN, ‘Training’ doesn’t really involve much except for storing the training data which might require large memory in case of a huge dataset. Prediction is always going to be computationally more expensive than ‘learning’ for kNN.
2. Naïve Bayes algorithm is easily scalable and quick in prediction. Training is also relatively easy, as basically, all the instances of the features of a dataset are counted to calculate conditional probabilities. Even with gigantic datasets it should not be computationally expensive
3. Neural Networks because of their ability to learn complex functions, and possessing hidden layers, are notoriously expensive to train. However, these models make predictions with alacrity.

**Question 7**

Yes, Neural Networks can represent all functions that Decision Trees can. Neural Networks are largely better than Decision Trees, because not only can they learn far complex functions as compared to Decision Trees, they can be used for continuous data as well and are also less susceptible to noisy and “imbalanced” data. However, training a Neural Network can very computationally expensive, even more so than Decision Trees. Interpretation of the model is also near impossible compared to a Decision Tree which is straightforward. So, in theory Neural Networks may always be a better algorithm than Decision Trees but in practice it is not so.