1.1)

N = 1 Training error: 0.325

N = 1 Testing error: 0.332

N = 2 Training error: 0.242

N = 2 Testing error: 0.285

N = 3 Training error: 0.150

N = 3 Testing error: 0.174

N = 4 Training error: 0.125

N = 4 Testing error: 0.178

N = 5 Training error: 0.080

N = 5 Testing error: 0.118

N = 6 Training error: 0.050

N = 6 Testing error: 0.114

N = 7 Training error: 0.020

N = 7 Testing error: 0.085

N = 8 Training error: 0.005

N = 8 Testing error: 0.078

N = 9 Training error: 0.000

N = 9 Testing error: 0.084

N = 10 Training error: 0.000

N = 10 Testing error: 0.084

N = 11 Training error: 0.000

N = 11 Testing error: 0.084

N = 12 Training error: 0.000

N = 12 Testing error: 0.084

N = 13 Training error: 0.000

N = 13 Testing error: 0.084

N = 14 Training error: 0.000

N = 14 Testing error: 0.084

N = 15 Training error: 0.000

N = 15 Testing error: 0.084

Training error is 0 after N=9 and test error stops decreasing after that

1.2)

We would pick N=9 which yields in a min error of 0.1

And switching the validation and testing set changes the result now the min error is 0.085. The optimal depth now is: N=6

How could use more of our data to estimate the depth more reliably?(???)

We could use cross-validation. Each time splitting our data into 5 and picking one of the sets as the validation sets instead of giving up a half of our data. We could just pick the ceiling of the average minimizing tree depth

2.4)

The runtime for classifying t example is O(tdk). This is because we are trying to determine the cost of calculating p(x\_ij|y\_i). In order to determine this we require three for loops. The first for loop going through every single label k, the second one going through every single feature d and the third label going through every single example d. as these are nested, the runtime can be found to be O(tdk)

3)

3.2]

KNN test error: 0.065

N = 1 Training error: 0.000

KNN test error: 0.066

N = 3 Training error: 0.028

KNN test error: 0.097

N = 10 Training error: 0.072

3.3]

3.4]

When k=1, the algorithm predicts the own labels of X\_test, so there is no room for error.

3.5]

?

4)

4.1] Because we use bootstrapping. the fact that we use infinite depth tree does not overfit since the data we overfit on is randomly generated.

4.3]

Random Tree:

Training error: 0.148

Testing error: 0.515

Decision Tree:

Training error: 0.011

Testing error: 0.443

Random Forest:

Training error: 0.000

Testing error: 0.205

Discuss?

5)

5.1.2]

5.1.4]

5.2]

1) Because error function is related to the training set. We want to minimize the test\_error. Minimizing the training error might lead to overfitting

2) This is the violation of the golden rule. We should ignore the test data until we tune our model and the hyper-parameters. If we use the k value that minimizes the test\_error that is either violating the golden rule or treating the test set as training set; both of which are dubious methods.

4)The biggest change in slope happens at k=3

I think 3,4,5 could be reasonable guesses for the optimal k value because the higher the k value is the higher the risk for overfitting

5.3.1] Eps= 2 gives the 4 true clusters

5.3.2]Eps=5 gives 3 clusters

5.3.3]Eps = 13 gives 2 clusters

5.3.4]Eps = 20 and min\_sample=100 exclude the outliers and gives one cluster

6)

1. What is an advantage of using a boxplot to visualize data rather than just computing its mean and

variance?

The obvious advantage is you can see the outliers visually. Also, boxplot shows data in clusters which makes it easier to see where the bulk of data is compared to mean which might not be the perfect measure for certain data type such as a list of 10 -10s and 10 10s, the mean is 0 but there is no element which is close to 0

2. What is a reason that the the data may not be IID in the email spam filtering example from lecture?

Assuming this question is asked about word-label pairs: meaning word-label pairings are not i.i.d. If it is the case that some group of words are used together in spam messages than not, the i.i.d. assumption is violated

3. What is the difference between a validation set and a test set?

Validation set is part of our training set and we use it to check how well we could do in the test set. Whereas test set is not used until we are confident about our training set and it is not part of the training set it is a whole new set which we have not seen

4. Why can't we (typically) use the training error to select a hyper-parameter?

Because, we can overfit meaning try too many things and get one that works really well by chance or over-trying and get a training error of almost zero but it does not mean that we will do well in the test set.

5. What is the effect of n on the optimization bias (assuming we use a parametric model).

Optimization bias shrinks as the number of validation examples increases. So, given that we choose a fixed percentage of the training set (say 20%), then optimization bias will shrink as n increases.

6. What is an advantage and a disadvantage of using a large k value in k-fold cross-validation.

Disadvantage could be the runtime because as we increase the k, the number of times we run our algorithm increases in parallel to k

Advantage could be that we now try so many different cross-validations sets that it is very likely that our estimate will be close to the actual test error assuming our data is i.i.d

7. Why can we ignore p(xi) when we use naive Bayes?

Because we only need to compare the conditional probability of p(y\_i=” spam”|x\_i) to p(y\_i=”non-spam”|x\_i)

When we use the Bayes rule both denominators are p(x\_i) then we can just simplify these two so no need to calculate p(x\_i). An exception would be when p(x\_i) is 0 but we can overcome this with laplace smoothing

8. For each of the three values below in a naive Bayes model, say whether it's a parameter or a hyper-

parameter:

1. Our estimate of p(yi) for some yi.

?

1. Our estimate of p (xij j yi) for some xij and yi.

?

1. The value beta in Laplace smoothing.

hyperparameter

9. What is the effect of k in KNN on the two parts (training error and approximation error) of the

fundamental trade-off. Hint: think about the extreme values.

As k increases, the training error decreases and approximation error increases.

10. Suppose we want to classify whether segments of raw audio represent words or not. What is an easy

way to make our classifier invariant to small translations of the raw audio?

The blank background noise in the raw audios are likely to be the same across different segments of raw audio. So if we crop these parts out of all the segments we are only left with sounds that are different from silent parts or blank background noise. This accounts for the fact that there could be time translations.

11. Both supervised learning and clustering models take in an input xi and produce a label yi. What is

the key difference?

In clustering models, we do not know the labels y\_i

12. Suppose you chose k in k-means clustering (using the squared distances to examples) from a validation

set instead of a training set. Would this work better than using the training set (which just chooses

the largest value of k)?

Assuming the training sample is i.i.d, (training sample = training set+ validation set); the success of the model just depends on how big the training set and validation sets are. So, if they are equally split (50-50 split) then they would work just equally well. If the training set is bigger ,by the law of large numbers, the means are likely to be more accurate than the means that could be obtained by using the validation set.

13. In k-means clustering the clusters are guaranteed to be convex regions. Are the areas that are given

the same label by KNN also convex?

No, KNN areas can be concave, refer to the lecture demos