CPSC 532M Assignment 2

Lucas Palazzi, Student #79285755

1 Training and Testing

1.1 Training and Testing Error Curves

Figure 1 shows the generated plot for 1.1.

As the depth increases, the training error decreases to zero, hitting zero at depth 9.

The testing error decreases to 0.078 at depth 8, then increases to 0.0835 at depth 9 and doesn't change after depth 9.

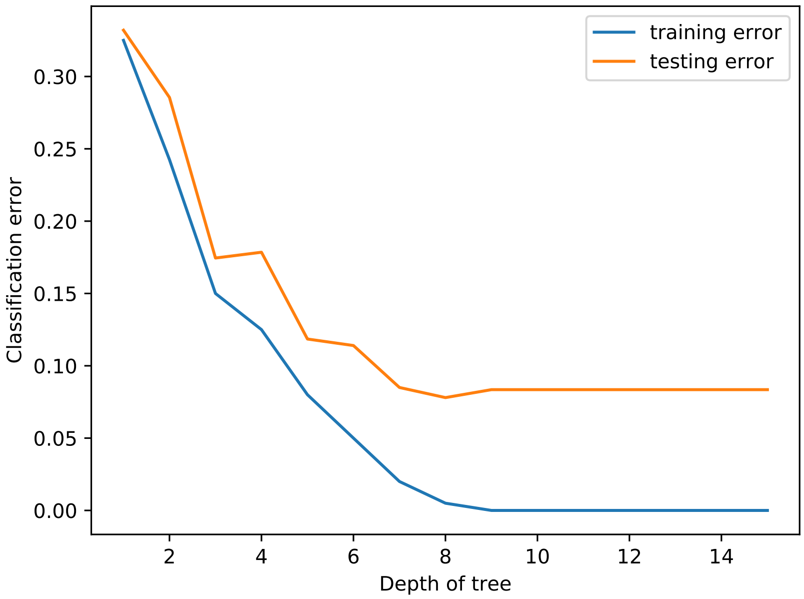


Figure 1: Classification error vs. Depth of tree (Question 1.1)

1.2 Validation Set

To minimize the validation set error, pick depth 8 for error of 0.1.

If the training and validation sets are switched, pick depth 6 for error of 0.085.

We could use more validation sets (smaller portion of training set) and use cross-validation, picking the depth with the best cross-validation score.

2 Naive Bayes

2.1 Naive Bayes by Hand

2.1.1 Prior probabilities

$p(\text{spam}) = 0.6$

$p(\text{not spam}) = 0.4$

2.1.2 Conditional probabilities

$p(x_1 = 1 \mid \text{spam}) = \frac{1}{6}$

$p(x_2 = 1 \mid \text{spam}) = \frac{5}{6}$

$p(x_3 = 0 \mid \text{spam}) = \frac{2}{6}$

$p(x_1 = 1 \mid \text{not spam}) = \frac{4}{4}$

$p(x_2 = 1 \mid \text{not spam}) = \frac{1}{4}$

$p(x_3 = 0 \mid \text{not spam}) = \frac{3}{4}$

2.1.3 Prediction

We predict spam if $p(y_i = \text{spam} \mid x_i) > p(y_i = \text{not spam} \mid x_i)$

$$\begin{aligned} p(y_i = \text{spam} \mid x_i) &= \prod_{i=1}^{3} p(\hat{x}_i \mid \text{spam}) p(\text{spam}) \\ &= p(\hat{x}_1 \mid \text{spam}) p(\text{spam}) p(\hat{x}_2 \mid \text{spam}) p(\text{spam}) p(\hat{x}_3 \mid \text{spam}) p(\text{spam}) \\ &= (0.6)^3 \left(\frac{1}{6}\right)\left(\frac{5}{6}\right)\left(\frac{2}{6}\right) \\ &= 0.01 \end{aligned}$$

$$\begin{aligned} p(y_i = \text{not spam} \mid x_i) &= \prod_{i=1}^{3} p(\hat{x}_i \mid \text{not spam}) p(\text{not spam}) \\ &= (0.4)^3 \left(\frac{4}{4}\right)\left(\frac{1}{4}\right)\left(\frac{3}{4}\right) \\ &= 0.012 \end{aligned}$$

Since

$$0.012 > 0.01$$
$$p(y_i = \text{not spam} \mid x_i) > p(y_i = \text{spam} \mid x_i)$$

Therefore the most likely label is “not spam”.

2.1.4 Laplace smoothing

The below matrices show the original 10 training examples but with an added 4 examples, shown below the horizontal line, in order to give the estimates with Laplace smoothing.

$$X = \left[\begin{array}{ccc} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ \hline 1 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{array}\right] \qquad y = \left[\begin{array}{c} \text{spam} \\ \text{spam} \\ \text{spam} \\ \text{spam} \\ \text{spam} \\ \text{spam} \\ \text{not spam} \\ \text{not spam} \\ \text{not spam} \\ \text{not spam} \\ \hline \text{spam} \\ \text{spam} \\ \text{not spam} \\ \text{not spam} \end{array}\right]$$

2.2 Bag of Words

1. lunar
2. car, fact, gun, video
3. talk.*

2.3 Naive Bayes Implementation

See README.md for link to code.

$$\begin{aligned} \text{validation error obtained} &= 0.188 \\ \text{scikit-learn validation error} &= 0.187 \end{aligned}$$

2.4 Runtime of Naive Bayes for Discrete Data

Cost is $\mathcal{O}(tdk)$

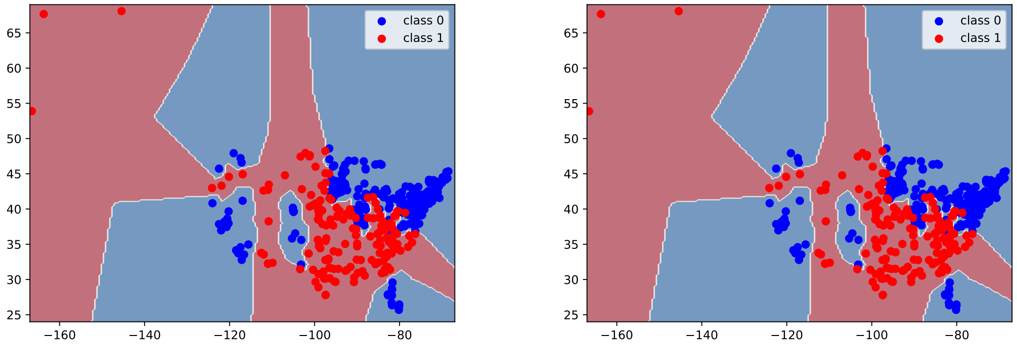
3 K-Nearest Neighbours

1. See README.md for link to code.
2.

k	training error	test error
1	0	0.0645
3	0.0275	0.0660
10	0.0725	0.0970

3. Figure 2 shows the generated plots for 3.3
4. The training error is 0 for $k = 1$ because since k is 1, the only "nearest neighbour" is itself. Therefore the predicted label will always be the actual label associated with that point.

5. Without an explicit test set, I'd use cross-validation to choose k (i.e., choosing different validation sets from my training set and using them as "test" sets, picking the k with the best cross-validation score).



(a) Using `knn.py` implementation

(b) Using scikit-learn implementation

Figure 2: Plotted KNN classifiers (Question 3.3)

4 Random Forests

1. The training error is not zero because it takes a bootstrap sample of the training data set, so it does not fit to the entire training set (i.e., some training examples are not considered).
2. See `README.md` for link to code.
3. The errors using 50 trees and no maximum depth are shown below.

$$\begin{aligned} \text{training error} &= 0 \\ \text{testing error} &= 0.189 \end{aligned}$$

   The Random Forest training error is zero, as is the case for the Decision Tree training error, which is less than the training error using a Random Tree.

   The Random Forest testing error is less than both the testing errors for Decision Tree and Random Tree.

   This is as expected since the Random Forest will fit the training set better than a single Random Tree (considers all of of the training set examples) but does not overfit to the training set as is the case for the Decision Tree model.

4.

$$\begin{aligned} \text{RandomForest time taken} &= 8.771621 \text{ seconds} \\ \text{RandomForestClassifier time taken} &= 0.078120 \text{ seconds} \end{aligned}$$

5 Clustering

5.1 Selecting among k-means Initializations

1. See `README.md` for link to code.

2. The value of the error decreases after each iteration. Once the error stops decreasing and doesn't change, the algorithm stops.
3. Figure 3 shows the generated plot for 5.1
4. `sklearn.cluster.KMeans` hyperparameters:
    - `n_clusters`: the number of clusters (means) to use
    - `init`: the method of initialization (i.e., how the inital means are picked), can be either `kmeans++` or `random`
    - `n_init`: the number of times to run k-means algorithm with different initial means
    - `max_iter`: maximum iterations of the k-means algorithm for a single run

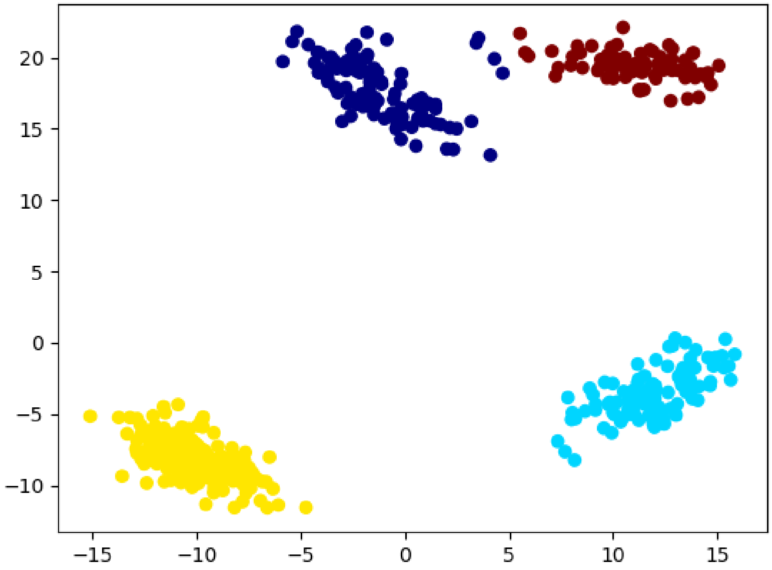


Figure 3: k-means clustering with lowest error from 50 runs (Question 5.1)

5.2 Selecting k in k-means

1. We do not choose k by taking the value that minimizes the `error` function because this is done only on the training data and thus only minimizing the function for the training dataset (overfitting).
2. Even evaluating the `error` function on test data is not a suitable approach because you could run into the same problem of overfitting to the test data.
3. Figure 4 shows the generated plot for 5.2
4. According to the *elbow method* I would choose k to be between 2 and 4, as this is the range of k that seems to make the sharpest "elbow".

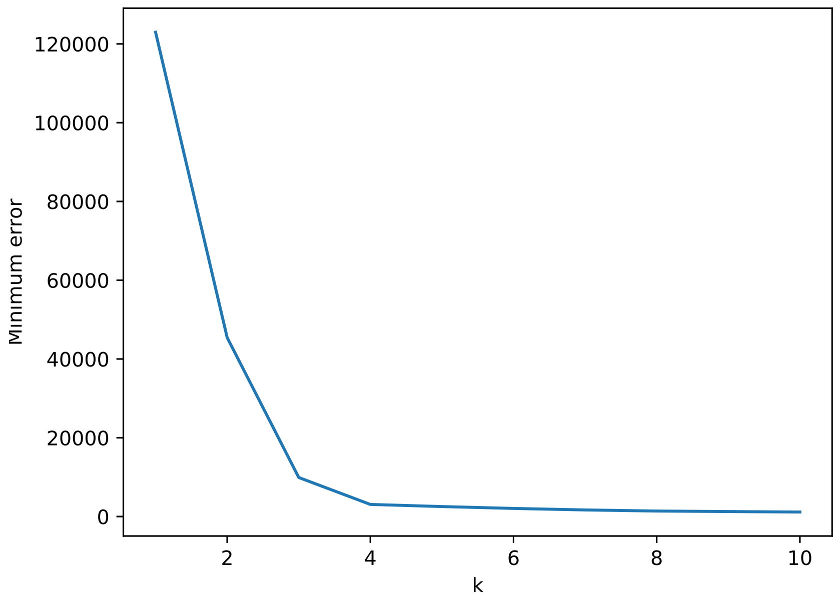


Figure 4: Minimum error across 50 initializations, as a function of k (Question 5.2)

5.3 Density-Based Clustering

Clusters	`eps`	`minPts`
4 “true” clusters	3	3
3 clusters	12	2
2 clusters	15	2
1 cluster	16	2

6 Very-Short Answer Questions

1. Boxplots visually depict the quartiles, which can’t be visualized using only mean and variance.
2. Under the IID assumption the order of the examples should not matter, however the order in which one receives emails does influence the data in the emails (e.g., receiving an email confirmation for an online shopping order makes it more likely to receive a shipping notification email a day or two later).
3. A validation set is a subset of the training set, where a test set is separate from the training set.
4. We can’t typically use the training error to select a hyperparameter because by only minimizing the training error we risk overfitting the model to the training data.
5. The optimization decreases with the number of examples n in the training dataset.
6. An advantage of using a large k value in k-fold cross-validation is that you cross validate your model more times, thus reducing optimization bias. The disadvantage is that it requires longer computation because the error must be computed k times.
7. We can ignore $p(x_i)$ since after the conditional probabilities in the inequality are substituted using Bayes rule, $p(x_i)$ is a denominator on both sides of the inequality and can therefore be

ignored.

8. (a) parameter

   (b) parameter

   (c) hyperparameter

9. (a) increasing $k \rightarrow$ training error $\uparrow$, approximation error $\downarrow$

   (b) decreasing $k \rightarrow$ training error $\downarrow$, approximation error $\uparrow$

10. To make the classifier invariant to small translations of the raw audio, we can add translated data to the training set.

11. In supervised learning, the training dataset has labels y_i that are given (i.e., we know the correct labels for the training set beforehand), while in unsupervised learning we do not know the correct labels for the training dataset.

12. By choosing k from a validation set as opposed to a training set, you are effectively limiting the largest value of k that can be chosen (validation set is a subset of the training set, therefore in a validation set the largest possible number of clusters is smaller than in the training set). This would be a better option since, assuming the validation set is randomly selected from the training set, the clusters will still be labelled but with lower chance of optimization bias.

13. The areas given by the same label by KNN are not convex.