Method:

- The data **spam** found in library **kernlab** was first divided into 2 subsets nonspam_m and spam_m based on the value contained in the column "type". The data consists of 2788 non spam entries and 1813 spam entries.
- The random seed is set to 50, making the results reproducible. Then, the subsets are sampled to the size 150 randomly in order to make the process of further dividing, given no repetition assumption, easier.
- The subsets are then further resampled randomly to the size 100 and 50 in order to eliminate the chances of repetition in the data sets.
- The next step adopted was to eliminate the column "type" and include only the features.
- Class factors for the training & test data were then obtained.
- Classification using knn with k values 1, 9 and 25 was conducted.
- The models then are used to test the test data, and hence the accuracy of each model is obtained.
- The models then are used to test the train data, and hence the accuracy of each model is obtained
- The accuracy when tested with the train and test data are then plotted
- In this process, we go back to the pre-analysis step, and calculate mean and variances of each features.
- There features (see Appendix A and Appendix B or line 193-194 in R-script) — *capitalAve, capitalLong, and capitalTotal* — are identified to have mean and variances far away from those of others, and thus could undermine the validity of Euclidean distances, which underpin knn.
- Hence Standardisation of all features is applied before analysis
- The next step is the same as conducted in the unstandardised case
- The accuracies obtained from the standardised features are compared with those of unstandardised ones.

Analysis:

A. Unstandardised features:

- k = 1 - The obtained accuracy of the test data set is 69%
- k = 9 - The obtained accuracy of the test data set is 70%
- k = 25 - The obtained accuracy of the test data set is 75%

Summary A:

It seems when k = 1, the model suffers from overfitting, meaning the model may achieve considerable high accuracy for the training data set but score poorly for the unseen one, as the model is overly flexible (i.e., consider only the closest neighbor in terms of Euclidean distance). In contrast to other cases, we see all probabilities associated with each outcome equal to one, as only one point (its closest neighbor) is taken to account. Hence, it represents a distorted and not very useful number. Albeit when k increases to 9, the accuracy only marginally increases to 70% from a mere 69%. This may indicate a problem underlying the features that we have (e.g., unequal contribution to the Euclidean distance as the ranges of features differ vastly). Lastly, when k = 25, the accuracy considerably increases to 75%, which may indicate that when k = 1 and k = 9, the models are greatly overfitted. As expected, we notice several instances where probabilities associated with a predicted class are considerably below 1, as the model captures many nearest neighbours, including far away ones, which presumably belong to the other class. Indeed, the analysis above is strengthen by an accuracy plot (Appendix C), which shows that when k = 1, the accuracy tested with the train data is maximised (100%), a strong evidence of overfitting. The graph indicates that k = 25 is arguably the optimal choice among the three, as the accuracy increases massively from k = 9 to k = 25 for the test data set, while dropping just slightly in the train data set (from 79.5% to 76%). However, due to the lack of fine-tuning parameters, we cannot say conclusively whether k = 25 is optimal, as the optimal k could probably be any values above 9. Alternatively, it is still possible, mathematically, that it lies between 1 and 9. Lastly, overall, the models suffer more false negative cases — where it predicts non-spam, but in reality, it is spam — than false positive cases — where it predicts spam, but in reality it is non-spam (see line 98, 116, and 134 in the R-script for further detail). A reduction, in percentage, of false negative cases are more noticeable than false positive cases, when compared k = 25 to k = 1, and k =25 to k = 9.

B. Standardized features:

- k =1 - The obtained accuracy of the test data set is 80%
- k = 9 - The obtained accuracy of the test data set is 84%
- k = 25 - The obtained accuracy of the test data set is 81%

Summary B:

When k = 1, the model is probably overfitted, as a considerably higher accuracy is achieved when k = 9. However, contrast to the previous unstandardized case, when k = 25, the model's accuracy drops, which may be due to the fact that the model is underfitted, meaning it is overly inflexible (e.g., too linear). Hence, it is likely that optimal k lies between 1 and 25, albeit mathematically possible that it lies beyond 25. For the three tested k, k = 9 seems to be the most optimal, given highest accuracy in the test data set. However, some may rightly argue that k = 1 receives an overall superior accuracy (for both test and train data set). This conundrum is subjective and may require further investigation to make any concrete conclusions. Apart from similar analysis above which can be readily applied in this section, the models with standardized features clearly result in an overall better accuracy rates, both when tested with the test and train data set (see Appendix C and Appendix D). Lastly, similar to the unstandardized case, false negative cases are identified more than false positive cases (see line 216, 234, and 248 in the R-script for further detail). False positive cases are reduced more, in a percentage term, than false positive cases when k increase from 1 to 9; however when k decreases from 25 to 9, the false positive cases reduce more than its counterpart, albeit slightly.

C. Comparison of A and B - Summary C:

Overall, a noticeable improvement has been realized when we standardize the features. The result is unsurprising, as we have discussed, the anomaly of three mentioned features contribute disproportionately to the Euclidean distance. By standardizing all features, we mitigate the problem of the massively disproportionate Euclidean-distance contribution of a particular feature, particularly when all features are numeric, which lessens complications stem from categorical features; in cases we have categorical features, caution must be warned as we may have to use dummy features to represent categorical features. Albeit ranges of the optimal value of k of both instances are suggested, we cannot be more specific or conclusive without fine-tuning the parameters. In addition, both types are, in general, found to have more false negative cases; this may due to the intrinsic design of the email that tries to minimize false negative cases, as important emails are costly to identified as such but junk emails ended up in the inbox are not so much costly. Lastly, all the results obtained are far from conclusive as we take one random seed for granted, in addition to relying on one randomized test data set and train data set. A random split and K-fold cross-validation should be implemented to improve the confidentiality of the results.

Appendix

Appendix A

Train Dataframe

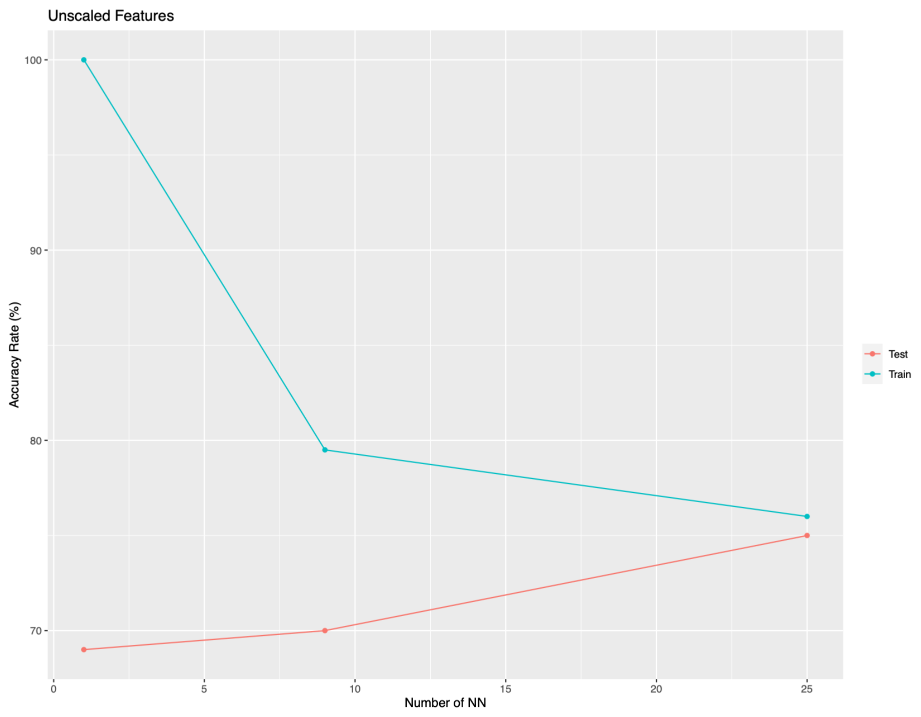
Variable	Mean	Standard Deviation
capitalAve	3.46	3.08
capitalLong	49.50	99.90
capitalTotal	300.00	496.00

Appendix B

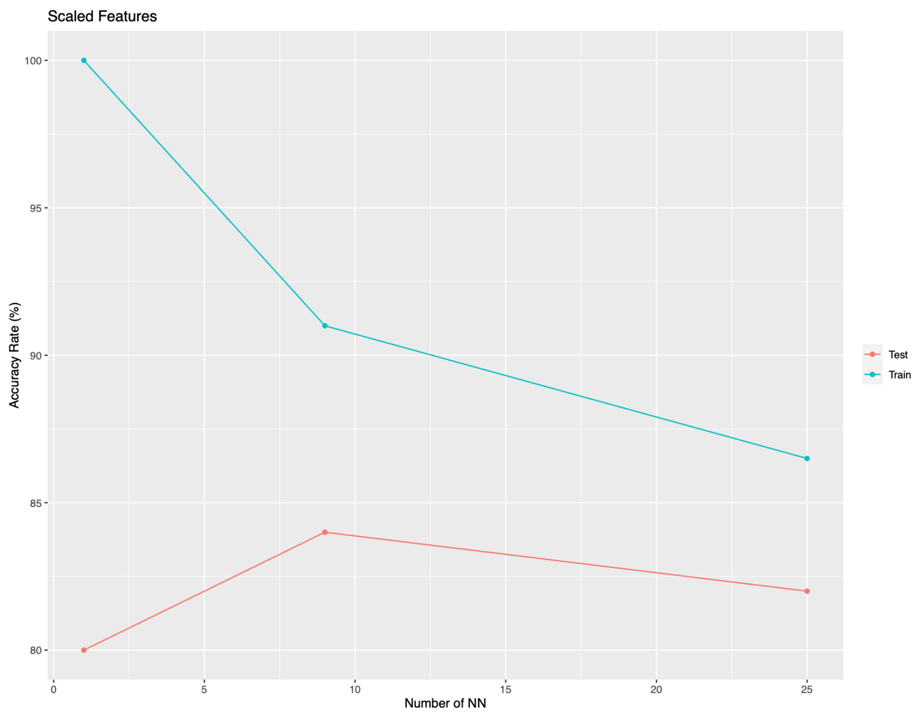
Test Dataframe

Variable	Mean	Standard Deviation
capitalAve	5.04	14.1
capitalLong	55.6	86.2
capitalTotal	297	524

Appendix C


Unscaled Features
Accuracy Rate (%)
100
90
80
70
0
5
10
15
20
25
Number of NN
Test
Train

Appendix D


Scaled Features
Accuracy Rate (%)
100
95
90
85
80
0
5
10
15
20
25
Number of NN
Test
Train