FINDINGS

PART 1

The provided (and sampled) data set contained 2000 rows of data containing details regarding if a site was a “Phishing” site or a “Legitimate” site. “Phishing sites” represented through a 1 in the “Class” variable (column 26 of dataset) and “Legitimate” sites were represented through a 0. The data set contains 25 predictors/variables (Labelled A01 - A25) that will be used to predict whether a site is “Legitimate” or not and compared to the provided “Class” data.

The proportion of “Legitimate” to “Phishing” sites is provided below:

Class	Proportion
0 (Legitimate)	64.2
1 (Phishing)	35.8

It can be found that in the provided dataset, 68.2% of rows of data represent “Legitimate sites” and 35.8% of rows of data represent “Phishing” websites.

The descriptions, including the mean, standard deviation, and five-number summary of the “real-valued” attributes are included in the table below (after NA values were omitted). “Real-valued” attributes were defined in this report as the predictors with more than 2 unique values within them as a few of the predictors had only 0 and 1 values within them.

Variable	Mean	SD	Min	Q1	Median	Q3	Max
A01	24.63	17.45	1	5	33	41	47
A02	0.2115	2.83	0	0	0	0	116
A04	2.755	0.5569	2	2	3	3	7
A05	0.009572	0.2829	0	0	0	0	12
A08	0.8518	0.2102	0.1579	0.6923	1	1	1
A11	0.04101	0.3986	0	0	0	0	11
A12	320.2	144.5	115	232	232	449	692
A13	0.007583	0.2236	0	0	0	0	9
A17	1.161	0.6125	0	1	1	1	8
A18	63.09	132.6	4	13	33	90	3738
A21	0.02174	0.1654	0	0	0	0	3
A22	0.05608	0.01065	0.0042	0.0511	0.0583	0.0632	0.0819
A23	70.87	68.2	0	11	100	107	1074
A24	0.2646	0.2513	0	0.0075	0.08	0.5229	0.5229
A25	0.0001724	0.00415	0	0	0	0	0.145

A few noteworthy aspects of this data are that most of the values within seem to be within a range of 0 to 10 with around 15 predictors containing only 0 and 1 values. Most of the attributes (A02, A05, A08, A11, A13, A21-22 and A24-35) have means and standard deviations close to 0.

Next, the number of NA values in each column/variable was found and reported below:

Predictor	A01	A02	A03	A04	A05	A06	A07	A08	A09	A10	A11	A12	A13
NA.Count	0	19	19	16	15	19	32	18	19	21	25	26	22
Predictor	**A14**	**A15**	**A16**	**A17**	**A18**	**A19**	**A20**	**A21**	**A22**	**A23**	**A24**	**A25**	**Class**
NA.Count	21	15	24	24	18	17	22	22	25	21	16	16	0

As can be seen from the table above, the number of NA values in each predictor is no more than 33 (with A07 having the highest at 32 NA Values). With 2000 rows of data in the provided data set, all the predictors have less that 2% of the total data as N/A values. Due to this very low significance, it is safe to keep all the predictors (not omit any predictors) from the proceeding analysis.

PART 2

A few pre-processing steps were needed to ensure that the data set was suitable for the following model fitting.

Firstly, al the NA values in the model needed to be removed as all the following model fitting cannot work with N/A data and N/A data can make the prediction of the models unpredictable and inaccurate. Therefore, all rows with an NA value were removed from the provided dataset. This reduced the total number of rows available in the data set from 2000 rows to 1567 rows.

After removing NA values in the data set, the proportion of “Legitimate” to “Phishing” sites is provided below:

Class	Proportion
0 (Legitimate)	65.5
1 (Phishing)	34.5

The proportions above remain similar to before removing NA values.

Next, the Class variable was changed from numerical attributes to a factor. This is because, the Class variable can only be 0 and 1 as they represent unique textual categories (“Phishing” and “Legitimate”) and cannot be a floating-point integer in-between. Therefore, for predicting the Class with classification models, the Class variable needed to have been a “factor”.

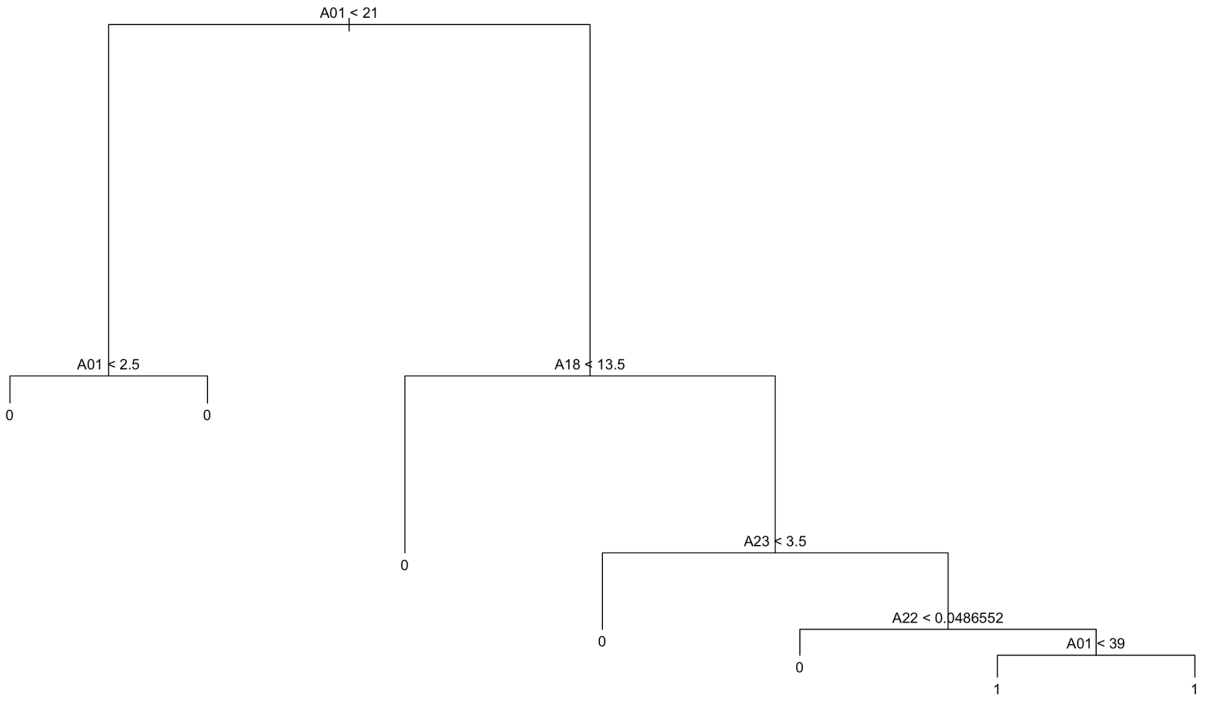
PART 3

The data was divided into 70% training set and 30% test set in R. See Appendix 13 – R-Code.

PART 4

RA classification model was fitted using 5 techniques (Decision tree, Naïve Bayes, Bagging, Boosting and Random Forest) using the newly created "training" data set, predicting Class through predictors A01 – A25. See Appendix 13 – R-Code.

The Decision Tree Plot is included below (Also in Appendix 4.1).



PART 5

Each of the models were then used to classify the newly created "test" data based on Predictors A01 – A25. The predicted classifications of the models were then compared against the actual data from the "Test" data, and a confusion matrix was created. The confusion matric and the accuracies of each of the models and shown below.

DECISION TREE

The confusion matrix for the Decision tree model is:

		Predicted	
Actual		0	1
	0	275	37
	1	49	110

The Accuracy for the Decision tree model is 81.7%.

NAÏVE BAYES

The confusion matrix for the Naïve Bayes model is:

		Predicted	
Actual		**0**	**1**
	0	1	311
	1	1	158

The Accuracy for the Naïve Bayes model is 33.8%.

BAGGING

The confusion matrix for the Bagging model is:

		Predicted	
Actual		**0**	**1**
	0	269	43
	1	47	112

The Accuracy for the Bagging model is 80.9%.

BOOSTING

The confusion matrix for the Boosting model is:

		Predicted	
Actual		**0**	**1**
	0	260	52
	1	62	97

The Accuracy for the Boosting model is 75.8%.

RANDOM FOREST

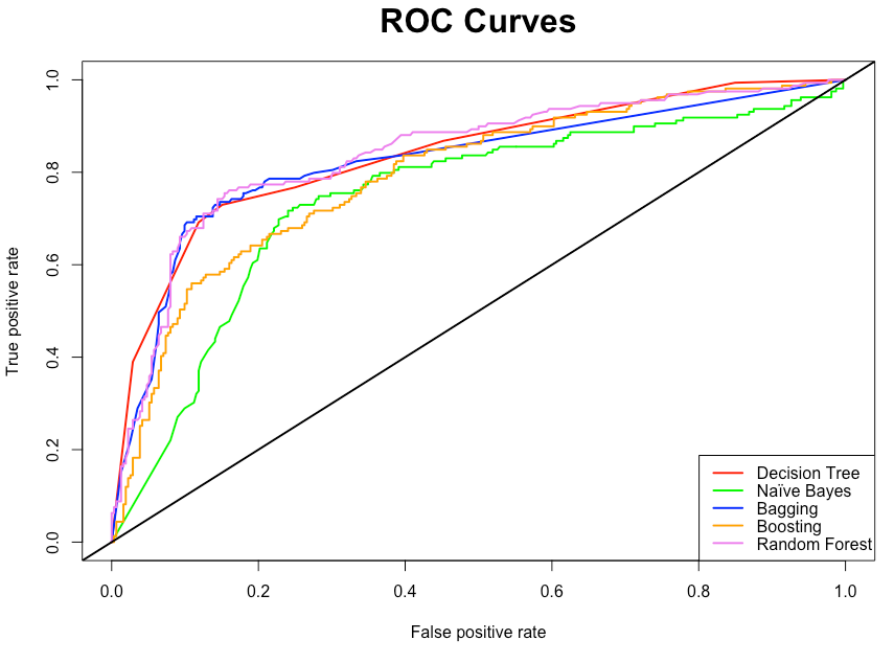
The confusion matrix for the Random Forest model is:

		Predicted	
Actual		**0**	**1**
	0	273	39
	1	50	109

The Accuracy for the Random Forest model is 81.1%.

PART 6

The confidence of predicting 'Phishing" for each case was found using the test data. An ROC curve was then constructed for each classifier and plotted on the same axis shown below (also in Appendix 6.1). The Area under the Curve (AUC) was also found for each classifier was also calculated and included in the table below.



Classifier	AUC
Decision Tree	0.837
Naïve Bayes	0.743
Bagging	0.822
Boosting	0.792
Random Forest	0.839

PART 7

A table was created comparing the Accuracy and AUC of each of the above classifiers to find a single "best" classifier. A column was added labelled "Average" that averaged the Accuracy and AUC scores (0.5 * Accuracy + 0.5 * AUC) to give a "final score" that will be used to compare the classifiers. All values are in a scale from 0 to 1.

Classifier	Accuracy	AUC	Avg
Decision Tree	0.817	0.837	0.827
Naïve Bayes	0.338	0.743	0.540
Bagging	0.809	0.822	0.816
Boosting	0.758	0.792	0.775
Random Forest	0.811	0.839	0.825

Purely based on Accuracy, the models ranked in order from "best performing" (highest accuracy) to worst (lowest accuracy) are Decision Tree, Random Forest, Bagging, Boosting and Naïve Bayes. It is notable that Decision Tree, Random Forest, Bagging and Boosting all have an accuracy of between 75% and 82% whilst the Naïve Bayes Accuracy is extremely low at 34%.

Purely based on AUC, models ranked in order from "best performing" (highest AUC) to worst (lowest AUC) are Random Forest, Decision Tree, Bagging, Boosting and Naïve Bayes. All 5 models have an AUC between 0.74 and 0.84.

Now looking at the average, the models ranked from “best” to worst” are Decision Tree, Random Forest, Bagging, Boosting and Naïve Bayes.

It would be incorrect to classify that the Decision Tree is the “single” best classifier (even though it had the highest average) because the average score of Random Forest is only two thousandths (0.002) lower than the average score of the Decision Tree. Based on the average score, Decision Tree and Random Forest are the best classifiers, followed closely by Bagging, followed by Boosting. Naïve Bayes seems to be the clear worst classifier as all other classifiers had an average of 0.77 or higher.

INVESTIGATIVE TASKS

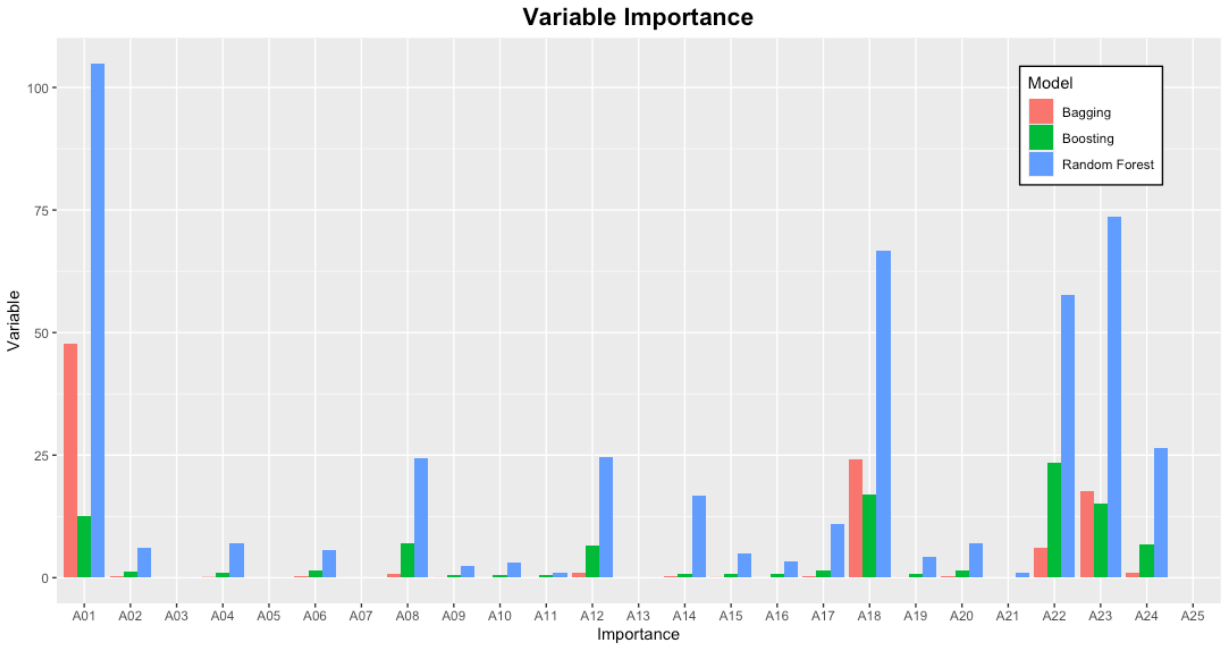
PART 8

DECISION TREE

For the Decision Tree model, the most “important” variables in predicting whether a web site will be phishing or legitimate are A01, A18, A23 and A22. This is because these are the only variables used in the decision tree to predict whether a website will be fishing or not as shows in the Decision Tree in Q4 (also in Appendix 8.1). For the decision tree model, the other variables can be omitted from the dataset without any impact on the performance of the decision tree as the other variables are not being used in the decision tree’s classification process at all.

ENSEMBLE METHODS

For the “Ensemble Classifiers” (Bagging, Boosting and Random Forest classifiers), the most important predictors in predicting whether a web site will be phishing or legitimate are shown in the graph below (also in Appendix 8.2).



For the **Bagging model**, the most important predictors, in order from most important to least important are A01 (importance value of 47.71), A18 (24.06), A23 (17.74), A22 (6.03), A12 (1.074), A24 (0.9055), A08 (0.7845), A06 (0.3955), A02 (0.3899), A20 (0.2757), A14 (0.2346), A17 (0.2114), A04 (0.08697), A15 (0.04695), A09 (0.02649) and A19 (0.02623). The most important predictors are **A01 (importance value of 47.71), A18 (24.06) and A23 (17.74).** The variables that have no importance (importance of 0) in the bagging model are A03, A05, A07, A10, A11, A13, A16, A21, A25. These variables (with no importance) can be omitted from the data without any (or very little) performance consequence as these variables are not being used by the Bagging model.

For the **Boosting model**, the most important predictors, in order from most important to least important are A22 (importance value of 23.51), A18 (16.95), A23 (15.03), A01 (12.65), A08 (7.089), A24 (6.88), A12 (6.643), A17 (1.494), A06 (1.405), A20 (1.376), A02 (1.342), A04 (0.9264), A14 (0.8251), A19 (0.7436), A15 (0.7435), A16 (0.7106), A10 (0.6456), A11 (0.5455) and A09 (0.4933). The most important predictors are **A22 (importance value of 23.51), A18 (16.95), A23 (15.03) and A01 (12.65).** The variables that have no importance (importance of 0) in the Boosting model are A03, A05, A07, A13, A21, A25. These variables (with no importance) can be omitted from the data without any (or very little) performance consequence as these variables are not being used by the Boosting model.

For the **Random Forest model**, the most important predictors, in order from most important to least important are A01 (importance value of 104.8), A23 (73.65), A18 (66.62), A22 (57.66), A24 (26.42), A12 (24.49), A08 (24.3), A14 (16.84), A17 (11.05), A20 (6.997), A04 (6.905), A02 (6.018), A06 (5.731), A15 (4.943), A19 (4.207), A16 (3.264), A10 (3.178), A09 (2.471), A11 (1.036), A21 (1.018), A03 (0.01309), A05 (0.01031), A25 (0.007628), A13 (0.005243) and A07 (0.004949). The most important predictors are **A01 (importance value of 104.8), A23 (73.65), A18 (66.62), A22 (57.66), A24 (26.42), A12 (24.49), A08 (24.3), A14 (16.84) and A17 (11.05).** The random forest model does not have any predictors with an importance (Mean Decrease Gini) of 0, therefore it cannot be concluded that any predictors can be omitted from this data set without there being meaningful negative consequences to the Random Forest Model's performance.

The **top 10 overall most important predictors**, found through averaging the importance of variables found from the Bagging, Boosting and Random Forest models, ranked from most (average) important to least are A01 (average importance of 55.07), A18 (35.88), A23 (35.47), A22 (29.07), A24 (11.4), A12 (10.74), A08 (10.73), A14 (5.967), A17 (4.253), A20 (2.883). It is notable that A01, A18, A23 and A22 (top 4 important predictors) seem to be more important that rest by a factor of 18 or higher.

PART 9

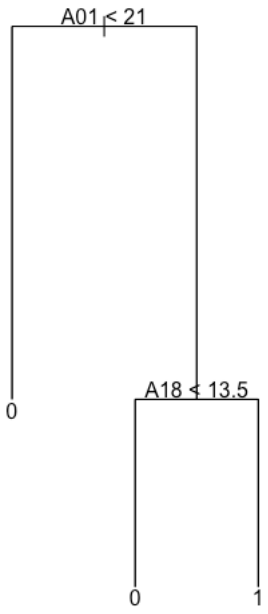
In order to create a classifier that is simple enough for a person to be able to be able to classify whether a site is phishing or legitimate by hand, the Decision Tree model was chosen due to its simple and greater "read ableness" over other ensemble methods. The Decision tree model was pruned in order to simplify the model. The level of pruning of the tree model was weighed against the perceived performance improvement of the decision tree in order to land on a final "best" and "simple" decision tree model.

The metric used are displayed in the table below (next page).

Classifier	Accuracy	AUC	Avg	Increase in AVG
Prune - 2	0.656	0.708	0.682	0
Prune - 3	0.756	0.773	0.764	0.082
Prune - 4	0.809	0.806	0.808	0.043

"Prune – 2" refers to the tree being pruned until there are only 2 leaf nodes and so on. Looking at the data above, the increase in average performance (**Increase in AVG**) score from a tree with 2 leaf nodes to a tree with 3 leaf nodes is 8%. However, the increase in average performance score from a tree with 3 leaf nodes to a tree with 4 leaf nodes is only 4%. Keeping in mind that as the number of leaf nodes in a tree increases, the "simplicity" of the tree decreases, it is imperative to choose the "simplest" tree that also has the highest performance. Therefore, the Prune – 3 tree model was selected as the "best" "simple" classifier in this instance, prioritising its "simpleness" (fewer leaf nodes) and its performance.

The "simple" tree model chosen in plotted and described below (also in Appendix 9.1).



The first criteria for prediction is the value of the A01 variable in the data. If the A01 variable in the data is less than 21, then that particular site should be classified as a "legitimate" site. If the A01 variable in the data is greater than or equal to 21, then the A18 variable in the data would have to be looked at. If the A18 variable (provided A01 is greater than or equal to 21) is less than 13.5, then that site should be classified as "legitimate". If the A18 variable is instead greater than or equal to 18.5 (provided A01 is greater than or equal to 21), then than site should be classified as "phishing".

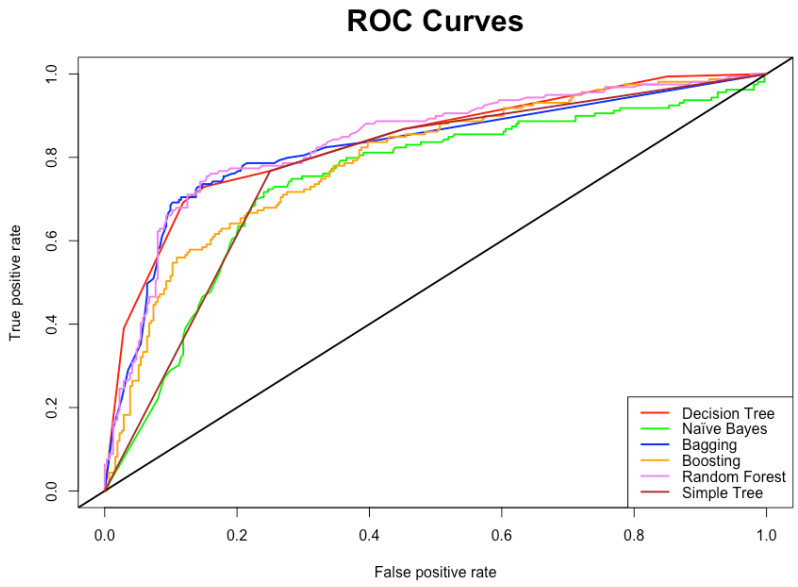
The two variables/attributes used here are A01 and A18. Looking at Part 8 above, it can be seen that A01 and A18 are the two most important predictors in terms of "average importance score" covering all three used ensemble methods. A01 was the most important predictor with an average importance value of 55.07 and A18 was the second most important predictor with an "average importance score" of 35.88. Therefore, these two variables were the ones that were selected and used for the "simple" classifier in order to achieve the maximum performance from the classifier.

The confusion matrix for the "simple" classification Tree based on the "test" data is below.

		Predicted	
Actual		0	1
	0	234	78
	1	37	122

The accuracy of this "simple" classification model is 75.6%.

The ROC Curve of the "Simple" classifier is shown below (also in Appendix 9.2), along with its AUC values, compared to the other classifiers from Part 4.



Classifier	AUC
Decision Tree	0.837
Naïve Bayes	0.743
Bagging	0.822
Boosting	0.792
Random Forest	0.839
Simple Tree	0.773

The AUC of the "Simple" Tree is 0.773, which is higher than the AUC of the Naïve Bayes model from Part 4. It is however slightly worse than the AUCs of the other classification model.

A table was created comparing the Accuracy and AUC of the above classifier to the other base classifiers from Q4. A column was added labelled "Average" that averaged the Accuracy and AUC scores (0.5 * Accuracy + 0.5 * AUC) to give a "final score" that will be used to compare the classifiers. All values are in a scale from 0 to 1.

Classifier	Accuracy	AUC	Avg
Decision Tree	0.817	0.837	0.827
Naïve Bayes	0.338	0.743	0.540
Bagging	0.809	0.822	0.816
Boosting	0.758	0.792	0.775
Random Forest	0.811	0.839	0.825
Simple Tree	0.756	0.773	0.764

Looking at the Accuracy of the Simple Tree model, it is much higher than the accuracy of the Naïve Bayes model. It has almost the same (slightly lower) accuracy as the Boosting Model and the Simple Tree has a lower accuracy than the other models.

Looking at the Average score (Avg), the "Simple" tree model has a better Average than the Naïve Bayes model but has a lower Avg score, and thus lower performance compared to the other (Decision Tree, Bagging, Boosting and Random Forest) model. However, the difference in performance (based on the Avg score) is very small here, at a level of 1% - 6% (0.01 – 0.06). It can be seen that the performance of the "simple" tree model is comparable to the more complex models (similar), whilst remaining simple enough for a person to be able to be able to classify whether a site is phishing or legitimate by hand.

PART 10

In order to create the best-tree based classifier possible, the random forest model was chosen as the base model, even though the Decision Tree had the highest Avg Model Performance. This is because the Random Forest model had a very similar Avg Model Performance to that of a Decision tree (only lower by 0.002 on a scale of 0 to 1) and the Random Forest model creates multiple data sets from the original training set using subset of data points and subset of attributes. This means that Random Forest can generally outperform individual trees (with a bit of fine-tuning) as there is no need to prune, Random Forest is not sensitive to outliers and overfitting is less of a problem in Random Forests compared to individual Decision Trees.

The main parameters that were changed in the creating of the best tree was the 'mtry' parameter and the 'ntree' parameter. The 'mtry' is the "number of variables randomly sampled as candidates at each split when creating a tree". Choosing the best mtry ensures that final model will have an improved accuracy/performance by making sure the model captures all the important variables at each split and reduces the chance of overfitting by capturing more than needed variables. The 'ntree' parameter "refers to the number of trees to grow in the forest". Increasing the 'ntree' parameter (up to a certain level) can increase the accuracy/performance of the Random Forest model as the model becomes better at averaging out the "noise" in the data across many trees and allows for the model to capture all the underlying patterns in the data that may be omitted with a lower 'ntree'.

The technique of Cross Validation, using the 'rfcv' function was used to find the best 'mtry' parameter that resulted in the lowest error (highest accuracy). The table below shows the different 'mtry' values tried by the 'rfcv' function (called n.var) and the corresponding error rates.

mtry (n.var)	**Error**
25	0.20072993
12	0.19890511
6	0.19708029
3	0.19616788
1	0.32846715

As can be seen from the table, the 'mtry' value with the lowest error seems to be 3 with an error of 19.6%. This 'mtry' value of 3 was chosen when refitting the Random Forest Model. The original random forest tree, fitted at Q4 used an 'mtry' value of 5. So, by using a lower 'mtry' value, the chance of overfitting the training data (which leads to a lower model performance in new data) is reduced. The new 'ntree' value chosen was 1000. This is double the default 'ntree' value of 500 and offers the new Random Forest model a better chance in capturing all underlying patterns in the data by averaging out the "noise" in the data through the use of many trees.

No attributes from the original training data set were omitted when fitting the new "best" Random Forest model. This is because, as can be seen from Part 8, the Random Forest model did not contain any attributes of 0 importance (importance value of 0) so it couldn't be concluded that those attributes could be omitted from this model without any meaningful

performance impact. The Random Forest model also uses the technique of creating multiple data sets using a subset of attributes and so by design is capable of choosing the "best" and most "important" attributes that increase model performance through combining the classifiers created for each subset of attributes and taking a majority vote for the final decision.

A new Random Forest model was created using the 'mtry' value of 3 and the 'ntree' value of 100 found above (suing the "train" data set).

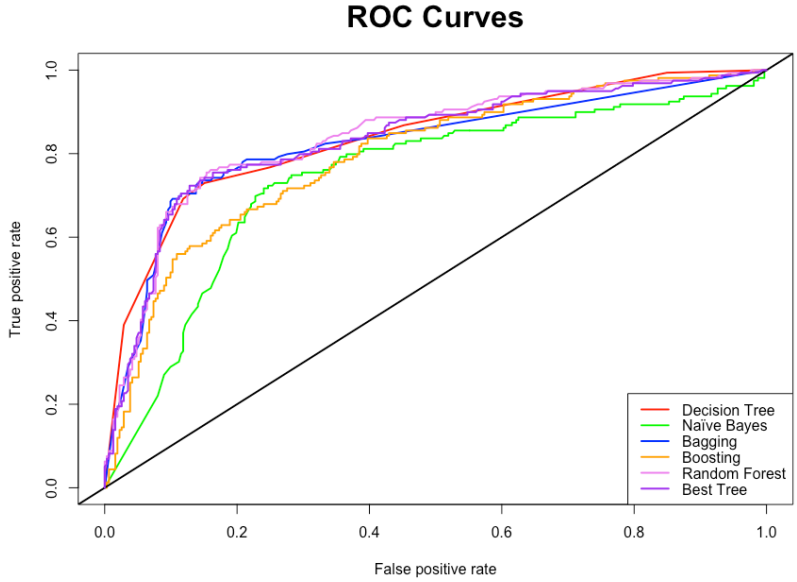
Looking at the new model, the 3 most important predictors are A01 (importance of 80.55), A23 (53.50) and A18 (47.28). Whilst the order of importance remains the same as the original Random Forest model, the importance values are lower in the new model comparatively.

The confusion matrix for the "best" tree-based model (new Random Forest model) based on the "test" data is below.

		Predicted	
Actual		0	1
	0	277	35
	1	48	111

The accuracy of this "best" tree-based (new Random Forest) classification model is 82.4%.

The ROC Curve of the new Random Forest classifier is shown below (also in Appendix 10.1), along with its AUC values, compared to the other classifiers from Part 4.



Classifier	**AUC**
Decision Tree	0.837
Naïve Bayes	0.743
Bagging	0.822
Boosting	0.792
Random Forest	0.839
Best Tree (new RF)	0.832

The AUC of the "Best" Tree (new Random Forest) is 0.832. This AUC value is very slightly lower than the AUC values of Decision Tree (lower by 0.005) and Random Forest (lower by 0.007). This difference in AUC is very marginal. However, the AUC value of the new Random Forest model is much higher than the AUC values of the other base models from Q4 suggesting the model's better comparative performance.

A table was created comparing the Accuracy and AUC of above classifier to the other base classifiers from Q4. A column was added labelled “Average” that averaged the Accuracy and AUC scores (0.5 * Accuracy + 0.5 * AUC) to give a “final score” that will be used to compare the classifiers. All values are in a scale from 0 to 1.

Classifier	Accuracy	AUC	Avg
Decision Tree	0.817	0.837	0.827
Naïve Bayes	0.338	0.743	0.540
Bagging	0.809	0.822	0.816
Boosting	0.758	0.792	0.775
Random Forest	0.811	0.839	0.825
Best Tree (new RF)	0.824	0.832	0.828

Looking at the Accuracy of the “Best” Tree-based model, it is higher than all the other base models from Part 4. It is much higher than the Naïve Bayes, Bagging and Boosting models, but only slightly higher than the Decision tree (higher by 0.007) and the Random Forest (higher by 0.013).

Looking at the Average score (Avg), the “Best” tree-based model has a better Average performance than all the other base models from Part 4. The Avg Performance of the new random forest model is considerable higher than the Naïve Bayes, Boosting and to an extent, even the Bagging model. However, the difference in performance (based on the Avg score) between the new Random Forest / “Best” Tree model and the Decision tree and the Random Forest models is very small here, at a level of 0.7% - 1.3% (0.007 – 0.0013). Even though the “Best” tree-based model technically has a higher performance metric, the difference is very marginal. This is mainly because the overall number of leaf nodes in the decision tree as well as the other tree-based models is quite small in this case. Cross Validation is mainly useful when there exists a large number of parameters to optimise or if the base model is highly prone to overfitting. In this case however, cross validation is not quite useful as the existing tree-based models created in Part 4 are already quite simple, and already avoid overfitting. So, the gains achieved through cross-validation and parameter optimisation is lower than anticipated, with the performance of the cross validated new Random Forest tree being almost the same as the base Decision Tree and Random Forest models.

PART 11

To implement an Artificial Neural Network (ANN) classifier to the “phishing” data to predict of a website is “phishing” or “legitimate”, pre-processing of the existing data needs to be done. Firstly, the data (predictors) needed to be normalised using the scale function. This is needed in order to ensure that the gradient-based optimisations used by neural networks work as intended and are not sensitised/biased due to differing scales of predictors.

All the predictor values were normalised, but the Class (Output Variable) was not scaled as the class needed to still be 0 or 1 as they represent “Legitimate” or “Phishing”. The class values were however converted to numeric values (they were originally factors). This is because Neural networks cannot work with factors and the values inputted / outputted need to be numerical values. The predictors were already numeric attributes.

One of the key advantages of a neural network is that they are able to "learn" the features from the data, meaning they can quickly identify which predictors are important and which are not. This means that Neural Networks can ignore "unimportant" variables by-design. Therefore, no attributes were omitted from the data whilst fitting the Artificial Neural Network. However, one consequence of this is that the model might not converge within the threshold due to a large number of predictors. This problem was not encountered during the fitting of the Neural Network, but if it were, the number of predictors used would have to be reduced. In order to choose the "best" predictors to use, their relative importance from the ensemble methods (Part 8) can be used. In Part 8 it was found that the top 10 overall most important predictors, found through averaging the importance of variables found from the Bagging, Boosting and Random Forest models, ranked from most (average) important to least are A01 (average importance of 55.07), A18 (35.88), A23 (35.47), A22 (29.07), A24 (11.4), A12 (10.74), A08 (10.73), A14 (5.967), A17 (4.253), A20 (2.883). These 10 variables would be the only predictors when training the Artificial Neural Network if the model did not converge. But, since the model did converge, all predictors were used from the original data set in order to avoid the possibility of the loss of important information from removing too many predictors, especially as the Neural Network already does a great job of identifying and the "important" predictors in the model.

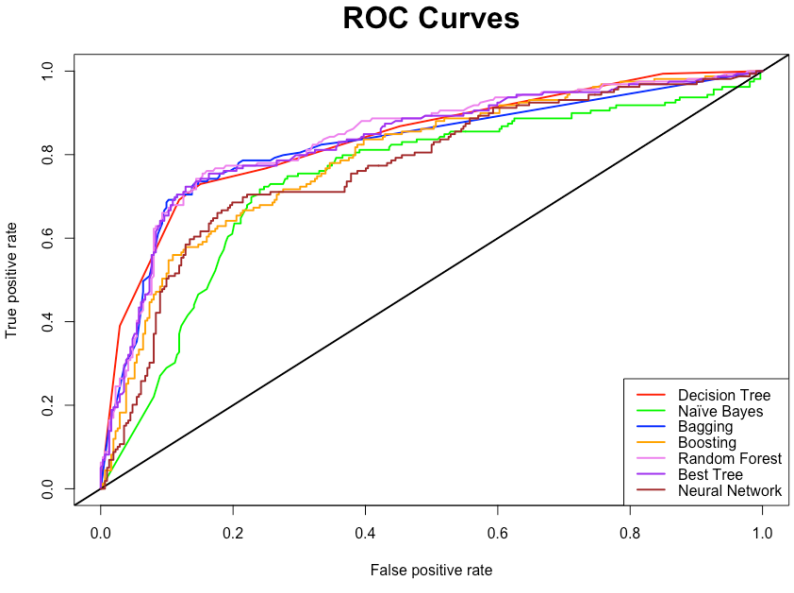
A 3x3 Artificial Neural Network was then fitted to the adapted training data set.

The confusion matrix for the Artificial Neural Network based on the "test" data is below.

		Predicted	
Actual		0	1
	0	274	38
	1	72	87

The accuracy of this Artificial Neural Network classification model is 76.6%.

The ROC Curve of the Artificial Neural Network is shown below (also in Appendix 11.1), along with its AUC values, compared to the other classifiers from Part 4 and Best Tree model.



Classifier	**AUC**
Decision Tree	0.837
Naïve Bayes	0.743
Bagging	0.822
Boosting	0.792
Random Forest	0.839
Best Tree (new RF)	0.832
Neural Network	0.771

is

however slightly worse than the AUCs of the other classification models by a factor of 2% - 6% (based on the model).

A table was created comparing the Accuracy and AUC of the above Artificial Neural Network and the base classifiers from Q4. A column was added labelled “Average” that averaged the Accuracy and AUC scores (0.5 * Accuracy + 0.5 * AUC) to give a “final score” that will be used to compare the classifiers. All values are in a scale from 0 to 1.

Classifier	Accuracy	AUC	Avg
Decision Tree	0.817	0.837	0.827
Naïve Bayes	0.338	0.743	0.540
Bagging	0.809	0.822	0.816
Boosting	0.758	0.792	0.775
Random Forest	0.811	0.839	0.825
Best Tree (new RF)	0.824	0.832	0.828
Neural Network	0.766	0.771	0.769

Looking at the Accuracy of the Artificial Neural Network, it is much higher than the accuracy of the Naïve Bayes model. It has a slightly higher accuracy compared to the Boosting Model, but has a lower accuracy compared to all the other models including Decision Tree, Bagging, Random Forest and Best Tree (new Random Forest from Q10)

Looking at the Average score (Avg), the Artificial Neural Network has a better Average (Avg Score) compared to the Naïve Bayes model. But it has a lower Avg score, and thus lower performance compared to the other (Decision Tree, Bagging, Boosting and Random Forest) models by a factor of 1% to 6% (0.01 – 0.06).

A few reasons for the Artificial Neural Network’s poor performance include that there is not enough data to train the neural network. Training a well-performing neural network typically requires tens of thousands oof data rows or more but to train this particular neural network, only 1097 rows of data were used. This level of data is not sufficient to create a high-performing model as using fewer data points can lead to overfitting as the model incorrectly assigns higher weights and biases to the small amount of data used. This is less of a problem with the other ensemble methods which use sampling to avoid overfitting. Neural Networks also have many different parameters (such as learbing rate, the width and depth o neuron layers etc) that significantly affect the performance of the model. For this implementation of Neural networks only a few neuron layers and depths were attempted (until 3 x 3) due to hardware limitations. Experimenting and analysing these parameters can significantly increase model performance. Furthermore, the data provided was somewhat biased with around 65% of the class values being 0 and only around 35% of the class values being 1. More information about the 1 class compared to the 0 class could have resulted in the higher False Negative rates seen in the model (compared to False Positives or True Positives).

PART 12

The new classifier chosen to be fit to this data and evaluated was **Support Vector Machines (SVM).** The package used to fit this classifier was **e1071** (https://cran.r-project.org/web/packages/e1071/e1071.pdf).

Support Vector Machine is a machine learning type algorithm used for regression and in this instance, classification tasks. It works by creating a line (or hyper plane when more than 2 dimensions) that tries to best separate the data points belonging to different classes, called the decision boundary. The main idea behind the algorithm is to maximise the margin between the decision boundary and the closest data points from each class (closest data points here are called support vectors). A larger margin means that there is a clearer separation between the classes, which means that the model is likely to perform better on new data.

The pre-processing required to fit a svm model is the same as the base models from Part 4. The NA values were omitted from the data set and the class column was converted from numerical to "factors" as 0 and 1 are unique values which represents "legitimate" and "phishing" sites respectively. All predictor values was also scaled (inside the svm function) to avoid overfitting similar to the ANN.

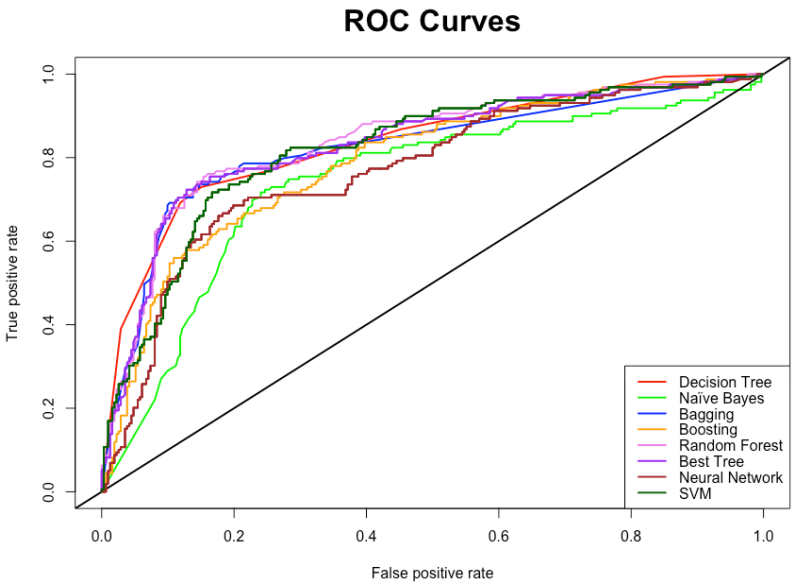
A svm model was then fitted to the created training dataset.

The confusion matrix for the Support Vector Machine model based on the test data is below.

		Predicted	
Actual		0	1
	0	269	43
	1	62	97

The accuracy of this Support Vector Machine model is 77.7%.

The ROC Curve of the Support Vector Machine model is shown below (also in Appendix 12.1), along with its AUC values, compared to the other classifiers from Part 4 and beyond.



Classifier	**AUC**
Decision Tree	0.837
Naïve Bayes	0.743
Bagging	0.822
Boosting	0.792
Random Forest	0.839
Best Tree (new RF)	0.832
Neural Network	0.771
SVM	0.819

The AUC of the Support Vector Machine model is 0.819, which is higher than the AUC of the Naïve Bayes model from Part 4 and the Artificial Neural Network from Part 11. It is however slightly worse than the AUCs of the other classification models by a factor of 1% - 2% (based on the model).

A table was created comparing the Accuracy and AUC of the above Support Vector Machine model, the base classifiers from Q4, Q10 and Q11. A column was added labelled "Average" that averaged the Accuracy and AUC scores (0.5 * Accuracy + 0.5 * AUC) to give a "final score" that will be used to compare the classifiers. All values are in a scale from 0 to 1.

Classifier	**Accuracy**	**AUC**	**Avg**
Decision Tree	0.817	0.837	0.827
Naïve Bayes	0.338	0.743	0.540
Bagging	0.809	0.822	0.816
Boosting	0.758	0.792	0.775
Random Forest	0.811	0.839	0.825
Best Tree (new RF)	0.824	0.832	0.828
Neural Network	0.766	0.771	0.769
SVM	0.777	0.819	0.798

Looking at the Accuracy of the Support Vector Machine model, it is much higher than the accuracy of the Naïve Bayes model. It has a slightly higher accuracy compared to the Boosting Model and the Neural Network, but has a lower accuracy compared to all the other models including Decision Tree, Bagging, Random Forest and Best Tree (new Random Forest from Q10)

Looking at the Average score (Avg), the Support Vector Machine model has a much better Average (Avg Score) compared to the Naïve Bayes model. The SVM model also has a slightly higher Avg score compared to the Boosting model and the Artificial Neural Network. A higher Avg score imples that the model is better performing overall. However, the SVM also has a lower Avg score, and thus lower performance compared to the Decision Tree, Bagging and both Random Forest models by a factor of 1% to 4% (0.01 – 0.04).

The reasons for the poorer performance of the SVM model compared to the Random Forest, Tree or Boosting models could be due to the relatively smaller amount of data used (only 1097 rows) as SVM is sensitive to data size and tends to perform better with larger data sets compared to the tree/ensemble methods.