CSC2062 AIDA – Assignment 3

Charles Collins

40256761

30/04/2021

Introduction

This report is about the analysis of various models to differentiate between different image types as well as determine what type different images may be. Each model uses a different classification method to classify the image based on the image's unique features. Cross-validation is also used for the models to evaluate the specific models on a limited data sample.

Section 1

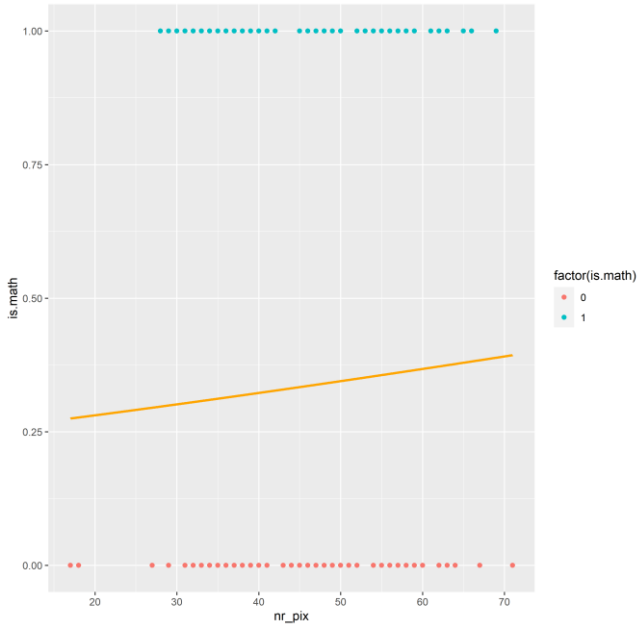
Section 1.1

To fit the logistic regression model for classification between math symbols and non-math symbols the images had to first be assigned a value to indicate its state as a math or non math symbol. Once the training values had been assigned these values it was possible to complete a binomial classification. The logistic regression model fitted with all 168 items using only the nr_pix feature to predict the probability of the "math symbol" category resulted in this table:

Coefficients:	Estimate	Std. Error	Z value	Pr(>|z|)
Intercept	-1.139338	0.710058	-1.605	0.109
Nr_pix	0.009946	0.015338	0.648	0.517

The estimated coefficient of beta 0 is -1.139338, while the estimated coefficient of beta 1 is 0.009946. This results in the formula of this logistic regression model being: $P(x) = e^{-1.139338+0.009946x}/1+e^{-1.139338+0.009946x}$ The std error is the standard errors of the coefficients. They can be used to construct the lower and upper bounds for the coefficient. The z-value is the ratio of the regression coefficient β to its standard error (z = coefficient ÷ standard error). The z statistic tests the hypothesis that a population regression coefficient is 0. If a coefficient is different from zero, then it has a genuine effect on the dependent variable. However, a coefficient may be different from zero, but if the difference is due to random variation, then the coefficient has no impact on the dependent variable. As the z value is the regression coefficient divided by the standard error it is an indicator of how much uncertainty surrounds the coefficient estimate, so a large z value shows a lower degree of uncertainty in the coefficient. My recorded z value is less than 1 showing a large degree of uncertainty in the coefficient estimate. Finally the P value indicates whether the corresponding coefficient has statistically significant predictive capability. It shows the probability the estimated coefficient is due to random variation rather than an actual link between the respective feature and the predicted factor. In this case there is an over 50% chance the coefficient is due to random variation instead of the predictive capability of nr_pix. As the p value is greater than the significance level (0.05) the association between nr_pix and image type is not statistically significant. The high p value indicates that nr_pix is an unreliable feature to use for this model as it does not have a significant level of predictive capability. This model showing that nr_pix is not an adequate feature to predict the type of symbol was expected as the nr_pix

feature is indicative of the size of the image rather than the overall shape of it which is what would determine its label and therefore its type.



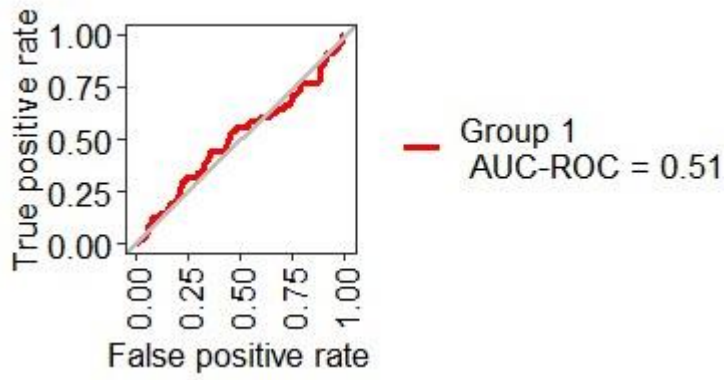
The plotted points on the graph are the training data and the line is modelled curve. As the graph shows the training data does not follow a pattern in regards to the nr_pix with the points being 1 or 0(yes or no) randomly across the range of number of pixels. The lack of correlation between nr_pix and the type of image is the reason the graph does not show a typical logistic regression curve. To produce a proper logistic regression curve the data points would have to show a more significant correlation between nr_pix and the image type as the current data shows that nr_pix is almost independent of the image type. The main assumption made when conducting this logistic regression was that the independent variable and the log odds were linearly correlated.

Section 1.2

Using the same model as in 1.1 while also using a decision threshold of 0.5 and evaluating the model with 5-fold cross-validation resulted in these values: accuracy = 0.6667, true positive rate(sensitivity) = 0, false positive rate = 0, precision = NA, recall = 0, F1-score = NA. By cross-validating the model from 1.1 we remove the possibility of overfitting the model, therefore giving results more in line with what the model would produce when used on a new dataset. The accuracy is 66.67% as none of the data fed to the model produces a probability of greater than 50% for being a maths symbol, so the model predicts none of the images are maths symbols which is correct for two thirds of the dataset, but this accuracy is not a good indicator of the true applicability of this model. The true positive rate or sensitivity is 1 and is calculated as TP/TP+FN, as no positives were predicted by the model the number of false and true positives are both 0. This sensitivity score shows the model is unable to detect math symbols using the given feature, nr_pix. The false positive rate is calculated by 1-specificity where specificity is TN/(TN + FP). The false positive rate is the indicator to how often the model incorrectly detects a maths symbol so a low false positive rate is indicative of a reliable model as it doesn't incorrectly non-math symbols as math symbols, so if looked at in isolation a false positive rate of 0 would indicate this model is extremely accurate. To achieve an accurate evaluation of the logistic regression model the true positive rate and false positive rate must be looked in conjunction with each other, when looking at the two rates together it is clear that the model is inaccurate as the true positive rate is 0 and the good false positive rate has merely been achieved by predicting false(non-math symbol) for all training examples. Recall is 0 and is the same as true positive rate or sensitivity. The precision is an indicator of how precise the model is, which means how often it has a correct prediction compared to how often it has an incorrect prediction, it is calculated by TP/TP+FP. As my model has no true or false positive

predictions the precision is NA. The F1 score is a measure of a model's accuracy on a dataset and is calculated by 2X((precisionXrecall)/(precision+recall)), as my precision is NA my F1 score is also NA. The f1 score would usually be useful if there was an uneven class distribution. One assumption of cross validation is that all rows are independent from one another

Section 1.3



The ROC curve shows the relationship between the Sensitivity and Specificity with curves that tend towards the upper left corner of the graph indicating a better performing model. The ROC is a probability curve while the area under the curve represents the degree of separability, meaning it is the measure of how capable the model is at differentiating between classes. The straight line through the origin is a baseline being a random classifier. As my ROC lies close to the baseline, sometimes falling below it, which results in the area under the curve equalling 0.51. A value of 0.51 means the used model is only marginally more accurate than a random classifier which would correctly classify the type of image 50% of the time on average. This result falls in line with expectations as previously stated in 1.1. Nr_pix is not an accurate predictor to use for a model to predict the type of the image, which was as expected.

Section 2

Section 2.1

To fit the k nearest neighbour model I first had to create a subset of the features that would be used for the training of the model. Then the model was fitted multiple times with a range of values of k. When conducting k-nearest-neighbour classification with all odd k values between 1 and 25 the accuracy across the full set of 168 items, while using nr_pix, rows_with_2, cols_with_2, rows_with_3p, cols_with_3p and height as features to train the model, for each value of k is:

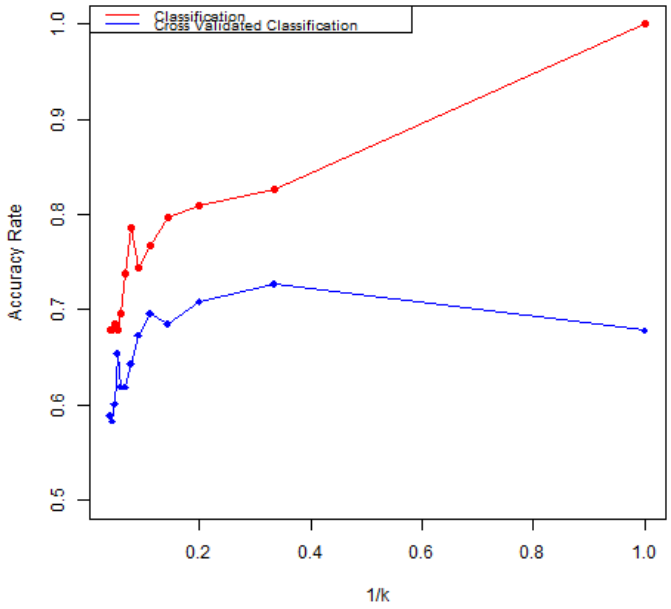
K value	1	3	5	7	9	11	13	15	17	19	21	23	25
Accuracy	1	0.827	0.809	0.798	0.768	0.744	0.786	0.738	0.696	0.679	0.685	0.679	0.679

This shows that the knn model when considering the first 6 features of the images produces the highest accuracy predictions when k is 1 and the model is least accurate when the k value is 23 or 25, where k is an odd number between 1 and 25 inclusively. This model is likely overfitted for the dataset due to the lack of cross-validation and the relatively low k values used, the effect of overfitting is elaborated on further in section 2.2. The KNN algorithm assumes that similar things exist in close proximity. The results were as expected due to the nature of non-cross-validated knn models and their susceptibility to overfitting.

Section 2.2

When conducting k-nearest-neighbour classification with all odd k values between 1 and 25 the 5 fold cross validated accuracy across the full set of 168 items for each value of k is:

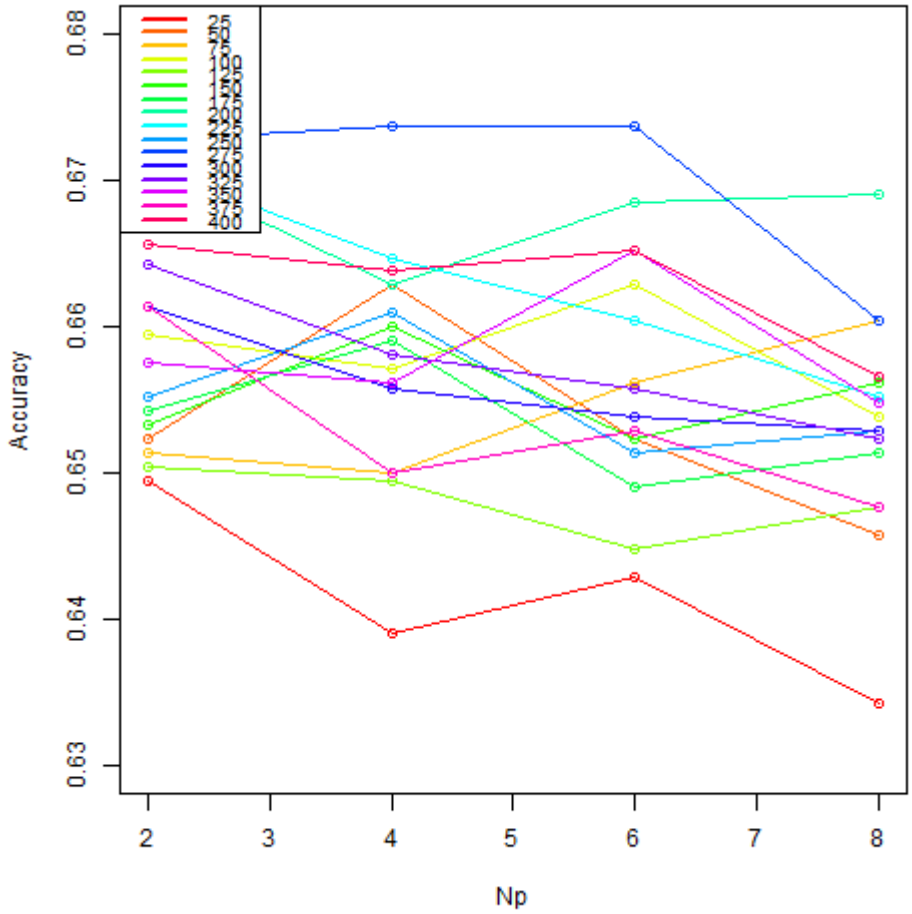
k	Accuracy
1	0.6661319
3	0.6969697
5	0.7028520
7	0.6729055
9	0.6424242
11	0.5951872
13	0.5593583
15	0.5948307
17	0.5534759
19	0.5707665
21	0.6067736
23	0.5593583
25	0.5775401



The best k value for the cross-validated model is 5, while the least accurate is 17. The overall accuracy of the model decreased after it was 5 fold cross-validated. The non-cross-validated line shows an overall positive linear correlation with a slight outlier where 1/k is 1/13. The cross-validated line shows a non-linear relationship with 1/k resulting in a right skewed distribution. This trend is also shown in the above graph with the majority of the base classification plot being above the cross validated classification plot. This is because the non cross validated model has likely been overfitted with the dataset, meaning the model has captured the noise of the data making it very accurate for the training set, but it will give poor results for any new data sets. 5 fold cross validation splits the training data into five partitions and runs the analysis on each partition, then averaging the overall error estimate. This has lower accuracy as it prevents the overfitting of the model so the model has greater applicability to new datasets. As stated the highest accuracy for the non-cross-validated model corresponded to a k value of 1, however due to this k value being low it is more significantly effected by noise within the data, hence why the accuracy is so high. The accuracy being extremely high for the low values of k implies the model is overfitted and that the accuracies for the higher values of k are closer to the accuracy the model would achieve if applied to a new dataset. This is further shown by the points on the graph being closer together for the base model and the cross-validated model at the higher values of k (lower values of 1/k). The results fall in line with prior expectations due to the previously known weaknesses of knn models and the effective strategies to reduce said weaknesses. The trajectory of the cross-validated classification plot shows that even after reducing the effect of overfitting through the use of 5 fold cross-validation the lower k values are still more susceptible to “noise” within the data lowering the overall reliability of the model for wider use. Despite the seemingly best k value being 5 due to this number being relatively low the chance of it being effected by “noise” is rather high, so it’s likely the accuracy at the higher k values will be more consistent with results produced by the model with new designated test datasets. When cross-validating the model I assume that all rows are independent from one another.

Section 3

Section 3.1



The graph shows the cross-validated accuracy of a random forest model against the number of predictors used in the model with each line corresponding to the number of trees used for each model. As the plots show the random forest model had the greatest cross-validated accuracy predicting the label/category of an image when the number of trees was 275 and the number of predictors was 4. The least accurate model used 25 trees and made use of 8 predictors. The graph does not show a reliable trend between number of trees and accuracy, nor does it show a consistent trend of number of predictors and accuracy. The highest accuracy achieved was 0.6738095, because the model has been cross-validated the likelihood of the accuracy being skewed due to overfitting is unlikely. The higher the number of trees used the more reliable the resultant model is, but as the number of trees increases the degree of improvement decreases and eventually the improvement in reliability is outweighed by the increase in computation time. This reasoning leads me to believe despite 275 being the optimum number of trees for this model when applied to this dataset, it is likely that the accuracy of the model will be closer to 0.6657143 as that corresponds to 400 trees. The results are slightly outside what was expected as I assumed the higher tree counts would produce more accurate models, but I seemingly underestimated how much the “noise” within the data, likely due to Nr_pix, affected the model. The assumption when conducting cross validation is that all rows are independent from one another.

Section 3.2

Using the best random forest model from 3.1, a Nt of 275 and Np of 4, to account random variance caused by the random elements of both cross-validation and random forest models I refitted the stated model 15 times and took the average and standard deviation of the refitted accuracies. The mean accuracy after refitting the model was 0.661, while the standard deviation of accuracies was 0.00479. This accuracy should be closer to the true accuracy of the model when applied to other datasets, as random variance would have been accounted for and removed by independently refitting the model.

Section 3.3

To construct the best possible model first I had to decide on which method to use. I decided to test both the k-nearest-neighbour and random forest method as both are effective classification models when using a large number of training examples. Knn classifies the image label by finding the distance between the query and all the examples in the data, selecting k number of examples closest to the query, it then votes for the most frequent label to decide what to predict. As previously mentioned The KNN algorithm assumes that similar things exist in close proximity. Random forest on the other hand is a collection of Nt number of decision trees which each complete a class prediction and the class with the most votes becomes the overall random forest model's prediction. This method works well because the trees protect each other from individual error as even if some trees are incorrect, many other trees will be right, so as a group the trees are able to move in the correct direction. This means the prerequisites for the random forest model to perform are there needs to be actual correlation between the features and the class so the trees perform better than random guessing and the other one is that the prediction made by the individual trees need to have low correlations with each other. A strength of random forest is that it uses bootstrap aggregation which is a machine learning ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression. It also reduces variance and helps to avoid overfitting. The bootstrap aggregation causes the random forest model to suffer less from variance than the knn model, reducing variance allows the model to be more consistent in its predictions. To determine which features to use for the knn and rf models I fitted multiple models for each method with each one using different combinations of features, through conducting this method I was able to decide which features contributed to a greater accuracy for the models and the features I settled on using were in line with my expectations. For the final model of both methods I used all features except the nr_pix feature. The nr_pix feature was omitted as when used it decreased the overall accuracy of the models, this is likely because it is not indicative of the shape of the image and therefore has almost no correlation with the image label. When fitting the models first I had to determine the optimum value for k for the knn method, and Nt and Np for the random forest method. To do this I fitted the model with multiple values for k, Nt and Np finally using the values that resulted in the highest accuracy. Then to account for the random variance caused by the cross-validation and random forest method I refitted both models 5 times and took the average accuracy. These accuracies were 0.653 for knn and 0.670 for rf. These results allow me to conclude that the best model for classifying between the 21 image categories is a random forest model that uses all features except nr_pix, has a Nt value of 400 and a Np value of 2. This better model being rf was expected from looking at the results of 2.2 and 2.2, as well as the information gathered prior to fitting the models such as the reduced variance and lesser susceptibility to overfitting. The optimum values for the rf model were also expected like the number of tree being the maximum as previously discussed in 3.1, which further back up the idea that the result in 3.1 was significantly effected by the "noise" caused by nr_pix. The low Np value was expected because by using less predictors the model is less likely to select and be dragged down by poor predictor features. To further improve the final model I could test with a larger range of Nt values as the most accurate model from the tested Nt values used the largest number of trees available, so there is potentially a more accurate model that uses more trees than 400. Another improvement would be to train the model with a larger dataset as by using more training data the model should become more accurate for more varied datasets as any features and correlations would be accounted for by the larger training set.

Finally a more effective way to determine which features to use would be to first use a classification model to separate the dataset by the image type(math, digit and letter) as some features may be better at differentiating between letters than digits. Once the data has been divided into subsets the used features can be decided by using statistical tests to evaluate their importance. Another way to improve the model would to devise more features from the images as some features that are unaccounted for could potentially have greater predictive capabilities than the features used in the current model. I could've also fitted more models using different methods which could have been more accurate than the knn and rf models I fitted.

Conclusions

Throughout this report I have analysed different classification methods of data and I have come to the conclusion that each method brings with it its own strengths and weaknesses. In section 1 the strength of logistic regression was slightly undermined due to the data, but it was useful for determining the usefulness of the used feature. Section 2 showed knn is susceptible to noise within the data for lower values of k, showing higher values of k are more reliable. Finally section 4 showed the effectiveness random forest and how for higher numbers of trees it provides a more reliable result.