Contents

1 Question 1

1.1 Q.i Explore the data set and calculate AUC

In this study case, the data set in `newthyroid.txt` contains five features that classify 185 samples into two classes, of which `n` and `h` stand for normal patients and patients who suffer from hyperthyroidism. The data set is imbalanced as the imbalance ratio is 4.29 with 150 `n` and 35 `h` samples. All the five variables are numeric, so boxplots are used to understand the data set and explore the distribution of samples for each feature. Figure 1 shows the five pairs of boxplots for five features (from `feature1` to `feature5`) associated with two classes `n` and `h`.

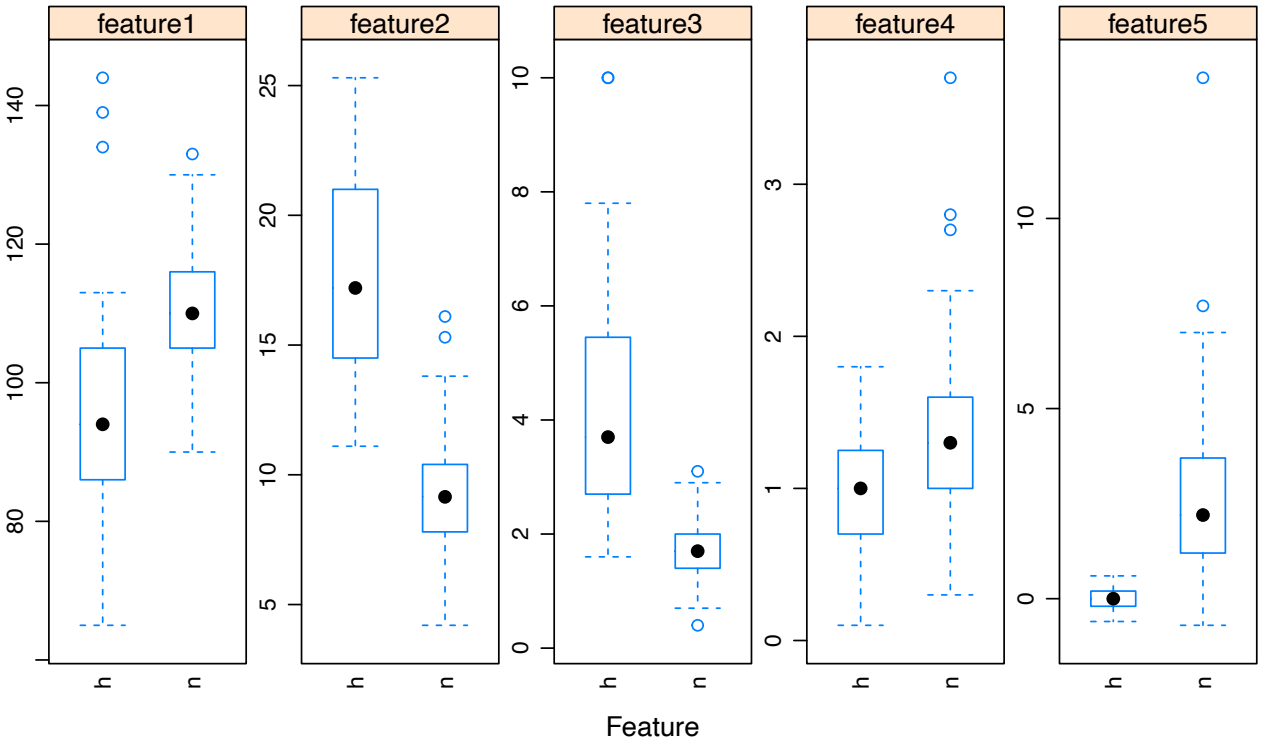

Figure 1: Boxplots of five features in the newthyroid data set.

From the boxplots, it is observed that all the five variables contribute to the classification task of patients' classes (`n` and `h`) as the medians of each variable vary for different classes. For feature 1, 4 and 5, a normal patient tends to have a higher value, as the corresponding median is higher. For feature 2 and 3, patients who suffers from hyperthyroidism tend to have a higher value, compared to normal patients. The patient who suffers from hyperthyroidism

tends to have a wider spread in feature 1, 2 and 3 compared with the normal patient. On the contrary, normal patients tend to have a wider spread for features 4 and 5. It is also observed that among all the features, feature 4 is expected to have the smallest difference of median values between the two classes. This may indicate that feature 4 contributes less to the classification task compared to other features.

Pior to applying k-nearest-neighbours (kNN) method and linear discriminant analysis (LDA) to classifying the data set, random sampling is used to divide the data set into training and test sets. The R function `createDataParition()` is applied to choose 70% samples for training and 30% for testing, which contain 130 and 55 samples respectively. The random split is repeated 20 times so that we have distinct training and test sets in all the 20 random splits. Next, a for loop is then conducted to apply kNN and LDA for each training set and record the AUC value for predicting on the associated test set. For kNN, functions from `caret` package are used to tune k values based on AUC metric from (3,5,7,9,11,13,15,17,19,21). Specifically, 5-fold cross-validation method is conducted five times to choose the k value that results in the best AUC for each random split. For LDA, only one linear direction is obtained as there are two classes for each training set and functions from `caret` are used to conduct the analysis.

The function `roc()` from `pROC` package is used to create a `roc` object for each prediction on the test set and AUC can be extracted from that `roc` object. Two vectors are used to record the AUC value for both kNN and LDA methods in each iteration. The AUC for all the 20 random splits are shown in Table 1.

Table 1: The AUC values calculated on the 20 random splits using kNN and LDA.

	1	2	3	4	5	6	7	8	9	10
kNN	1.000	1.000	0.999	0.999	1.000	1.000	1.000	1.000	1.000	0.993
LDA	1.000	1.000	0.996	1.000	1.000	1.000	1.000	0.998	1.000	1.000
continued	11	12	13	14	15	16	17	18	19	20
kNN	0.997	0.996	0.998	0.998	1.000	0.998	0.996	0.996	1.000	0.999
LDA	0.996	0.996	0.996	0.998	1.000	0.998	0.996	1.000	0.993	1.000

1.2 Q.ii ROC curve

The ROC curves obtained from using kNN method and LDA are shown in Figure 2 based on the first random sampling. It is observed that the two ROC curves coincide with each other as the AUC values obtained from both methods are equal to 1. This could be seen from the table above. As AUC considers the area under curve, the same AUC values resulted from kNN and LDA lead to the identical areas under the ROC curve and thus the two ROC curves coincide for the first random split.

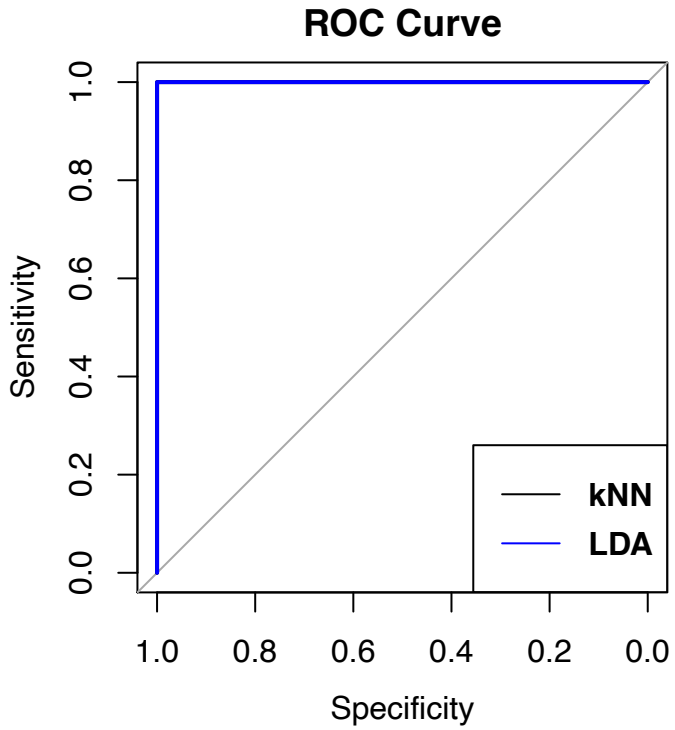

Figure 2: ROC curves based on kNN and LDA for the first random split

In the training process of applying classifier kNN on the first randomly selected samples, AUC is chosen as the metric to tune k. The optimal k value which maximises AUC value is 15, which means that we use 15NN to conduct the classification task on the test set. To introduce other metrics to assess the classification performance, confusion matrix is also computed for both kNN and LDA classifiers. The results are shown in Table 2.

From the confusion matrix, the diagonal elements indicate the number of correct classification outcomes for each class while the off-diagonal elements provide the misclassification. Thus,

Table 2: The confusion matrix for kNN and LDA classifiers on the test set.

	kNN		LDA	
	h	n	h	n
h	5	0	6	0
n	5	45	4	45

the classifier LDA produces a slightly larger number of correct classification outcomes, with the number of 51 compared to the number of 50 resulted from the kNN classifier. Other key metrics such as sensitivity and specificity are introduced and computed according to following formulas (1):

$$Sensitivity = TP/(TP + FN) \quad Specificity = TN/(TN + FP) \tag{1}$$

where TP stands for the number of true `h` correctly predicted as `h`, FP stands for the number of true `n` incorrectly predicted as `h`, FN stands for the number of true `h` incorrectly predicted as `n` and TN stands for the number of true `n` correctly predicted as `n`. For classifier kNN and LDA, the accuracy rates are 0.909 and 0.927 respectively, while sensitivity rates are 0.500 and 0.600 respectively and specificity rates are both 1.00. Thus, we conclude that the classifier LDA slightly improves classification results compared with the classifier kNN for slightly higher accuracy rate and higher sensitivity rate in this study.

1.3 Q.iii Boxplot of AUC

Based on the 20 AUC values calculated from 20 random splits, two boxplots for kNN and LDA are plotted in Figure 3. For classifier kNN, the AUC values have a median close to 0.999 with the spread from approximately 0.998 to 1 and an outlier located around 0.993. For classifier LDA, the AUC values have a median equal to 1 with larger spread from approximately 0.996 to 1. The boxplot for classifier LDA has a median higher than the median of boxplot for classifier kNN, while the spread of boxplot for LDA is wider than the spread of boxplot for kNN. This might result from the training process that we have used AUC-based cross-validation to tune the k value. That is, the training process of choosing k value with the

optimal AUC leads to narrowing down the spread of the boxplot, while the LDA method itself tends to produce more accurate prediction on the test set with higher median of AUC values.

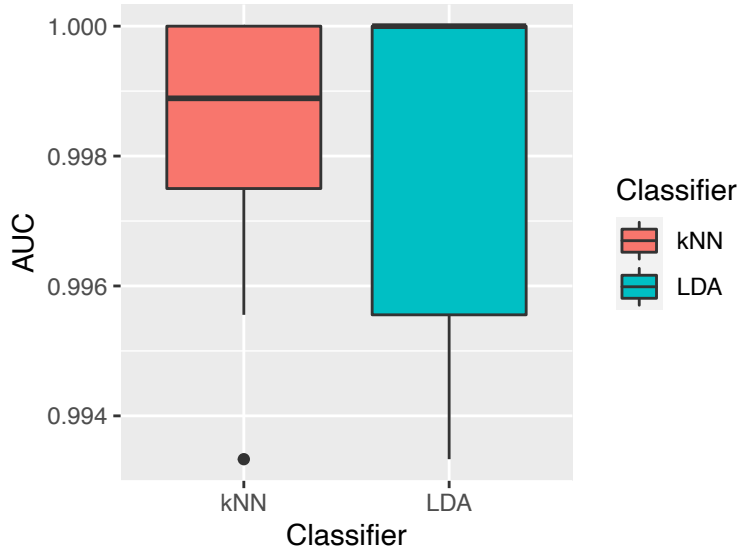

Figure 3: AUC obtained from using kNN and LDA for 20 random splits

1.4 Q.v Conclusions

As indicated in the analysis above, the sensitivity resulted from either kNN or LDA classifier is low, although both classifiers achieve rather high classification accuracy and AUC. This might be due to the imbalance in the data set as we have more `n` samples than `h` samples. However, it should be noticed that sensitivity needs to be an important metric in this `newthyroid` data set, as it may raise concerns if either method classifies patients who suffer from hyperthyroidism as normal patients. That being said, the low sensitivity means that either kNN or LDA classfier predicts a large number of patients with hyperthyroidism as normal patients, which may delay medical treatments for those patients.

Therefore, in order to improve the sensitivity of the classifier, we tune the value of k nearest neighbours according to sensitivity rather than AUC for the first random split. To conduct the comparison, `Sens` metric in the cross-validation process is specified in the `train()` function of `caret` package compared with `ROC` metric when tuning k based on AUC. The 3NN is chosen as it obtains the optimal sensitivity on the training set. The confusion matrix obtained

from predicting on the test set is given in Table 3, as well as the comparison with the confusion matrix obtained using 15NN that is tuned based on AUC for the first random sampling. Using 3NN, of which k is chosen based on the optimal sensitivity value in the training process, achieves sensitivity of 0.9 on the test set. There is a big improvement compared with 15NN chosen based on AUC. This means 3NN incorrectly classifies only one patient with hyperthyroidism as normal while 15NN predicts five incorrect test samples. Additionally, 3NN also achieves higher classification accuracy (0.9818), much higher Kappa (0.9364) and approximately the same AUC (0.999), indicating a better classification model compared to 15NN with 0.909 of accuracy and 0.621 of Kappa.

Table 3: The confusion matrix on the test set for 15NN and 3NN classifiers that are tuned using different metrics for the first random sampling.

	15NN (k tuned based on AUC)		3NN (k tuned based on sensitivity)	
	h	n	h	n
h	5	0	9	0
n	5	45	1	45

In conclusion, we should choose the metrics to assess the classification performance according to the specific data set. In this study, for example, sensitivity might be a better metric to train the classification model and assess the predicted results, because it is noticeable that wrongly classifying patients with hyperthyroidism as normal patients may lead to severe consequences in this classification task of diseases.

2 Question 2

2.1 Q.i `myFDA` function

Based on Fisher's Linear Discriminant Analysis, w could be computed by solving the following problem max $\frac{w^T S_B w}{w^T S_W w}$. The problem is equivalent to finding a direction w that maximises the ratio of between-class scatter and within-class scatter. By solving the problem, w is the direction of class mean difference normalized by within-class scatter S_w as shown in equation (2). The equation (3) is applied to compute the within-class scatter S_w via covariance matrix.

Based on the equations, a self-defined function is used to produce the linear discriminant for binary classification tasks.

$$w = S_w^{-1}(\mu_1 - \mu_2) \tag{2}$$

$$S_w = \sum_{c=1}^{2} \sum_{y_i=c} (x_i - \mu_c)(x_i - \mu_c)^T \tag{3}$$

2.2 Q.ii Cosine similarity

Cosine similarity is a method to measure the similarity between two vectors by observing the cosine angle of the vectors in a multidimensional space. Based on our calculation, w is the same linear discriminant either via `myFDA` function or via `LDA` function. The cosine similarity equals 1 which indicates that the linear discriminant calculated via these two methods has the same direction. Projections onto these two directions can be considered as the same. This result could be explained by the fact the FDA is equivalent to LDA if the assumptions of LDA are satisfied. Specially, LDA assumes that the distribution of a variable in every class is normal and the variance of a variable in every class is equal.

To verify whether our sample meets the first assumption of normality in LDA, we decided to plot and visualize histograms. Histograms are a type of data visualization that shows the distribution of a variable. It gives us the frequency of occurrence per value in the data set, which is the nature of distributions. For each class, 4 histograms are plotted, resulting in a total of 8 histograms. From the figure 4, the multivariate normality assumption is verified for our sample as there is little deviation from the theoretical bell curve distribution in all these histogram plots. Last but not least, we verify the second assumption of equal variances by calculating the variances of each feature for two classes. The final results show that for same feature under two classes, the variances are almost equal. Thus, this assumption is also met. Considering the fact that both two assumptions are satisfied for the sample, FDA is equivalent to LDA so that `myFDA` function can help to solve LDA problem and achieve the same result as the `lda` function.

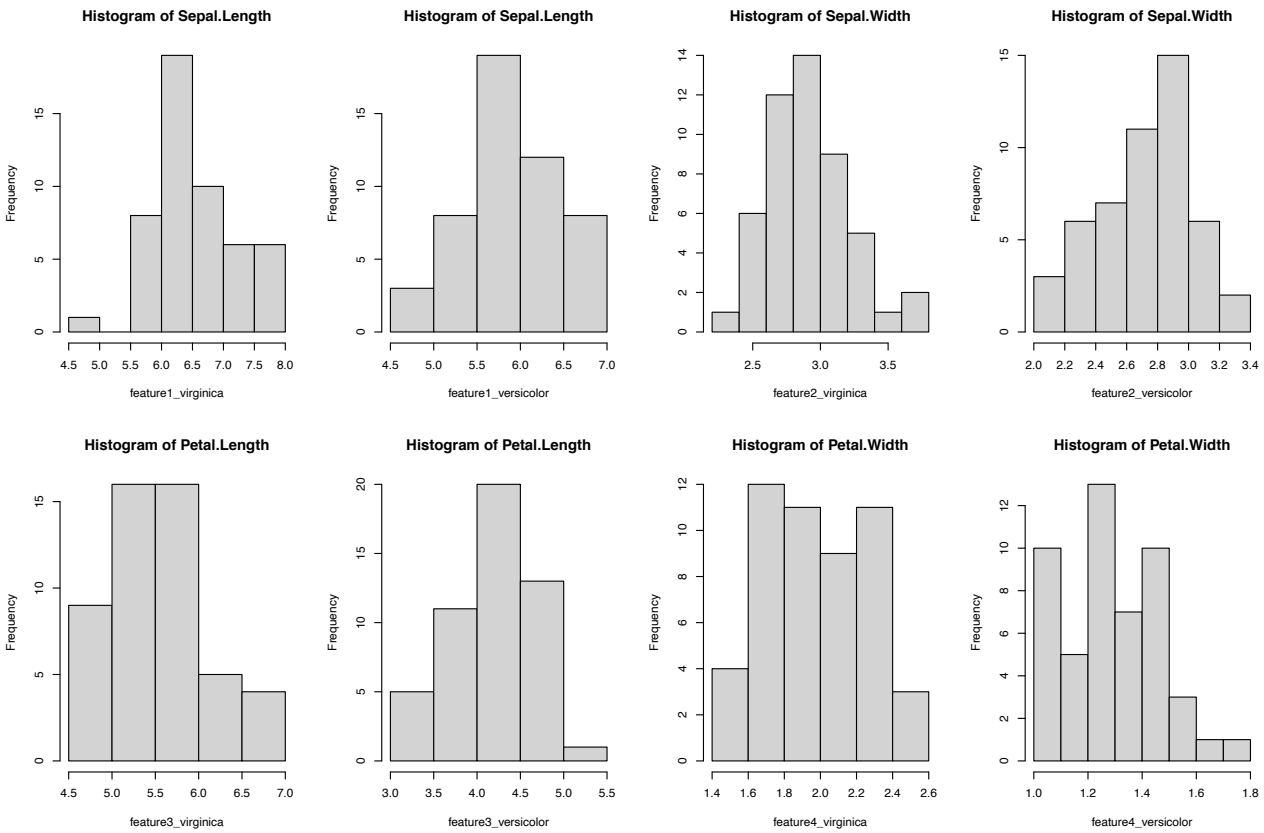

Figure 4: Histagrams of features for two classes in the iris data set.