STAT 615 - 2020 Spring
Midterm Project

Why Using Different Classifiers for Different Problems is Necessary

Section 002: Group 6

March 29, 2020

Group members:
David White
Yuechuan Chen
Hongyi Liu
Yu Chen
Ruyan Zhou

Contents

1 Introduction

1.1 Background

The coronavirus is a serious problem for the whole world. We are a team of researchers working at the Center of Disease Control focusing on the coronavirus. We are doing our best to fight this pandemic.

This article is in response to questions by the new White House appointed management about why we cannot use the same classifier in every situation. We hope by the time you finish this article, you will understand that there is no classifier that will perform the best in all situations.

For every classifier that exists, there is a situation in which it will be outperformed by another classifier. We will demonstrate this by comparing the performance of 3 different classifiers: Logistic Regression, Quadratic Discriminant Analysis, and K-Nearest Neighbors. These are 3 very different classifiers, and each one excels in a niche situation. However, none of these classifiers is the best in all situations. It is up to the researcher to use their intuition about the problem they are facing to pick the appropriate classifier that they think will perform the best.

1.2 Simulating Problems

We created 3 different problems to test our classifiers. For each problem, we make certain assumptions about the world, and use these assumptions to simulate a population. From these simulated populations, we take two samples, one to train each classifier, and one to test each classifier. Since the populations are simulated, we know exact testing error for each classifier. Finally, we compare the performance of each classifier by comparing testing errors.

For each problem, a unique classifier performs the best. As you may know, some classifiers converged either uniformally or under certain conditions. Sample size is an important factor to take into consider when choosing a classifier. To show this, we train the classifiers with both small and big samples. Under large training sets, the classifiers still cannot perfectly classify the data, and some classifiers still work better than others. At the end of each example, we provide a very large sample to give readers an impression of what the distribution of our stimulated population looks like. All simulations are performed in R. To access the code, there is a website link provided in Appendix.

2 Basic Ideas of some Concepts

In this section we define and explain some concepts discussed in this paper.

2.1 Classification and classifier

Classification is the process of assigning a class or label to a set of data. Classification is usually performed by observing a set of data with known classes and labels, and using knowledge from that data to predict the class or labels of a new set of data. A classifier is the decision rules that classify new data.

2.2 Error rate

Error rate is the total number of misclassifications divided by the total attempts at classification. This is used as criterion to compare the performance of different classifiers.

2.3 Training set and testing set

The training set is a set of data with known classes. This set is used to train a classifier, i.e. create the rules for classification. A testing set is a set of data that may not have known classes. This set is used to test performance of the classifier.

2.4 KNN: (K-Nearest Neighbors Algorithm)

KNN is a classifier that the input consists of the k closest training examples in the feature space and the output is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k-nearest neighbors. In our problems, we set the classifier to look at the 8 nearest neighbors, as this is commonly accepted to be one of the best parameter values for this algorithm.

2.5 LR: (Logistic Regression Classifier)

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes, by using logistic regression. Logistic regression classifier can be viewed as minimizing some sort of "empirical risk" (actually an average-log loss function).

2.6 QDA: (Quadratic Discriminant Analysis)

QD is a classifier that models and classifies the categorical response Y with a non-linear combination of predictor variables X, to be exact, a quadratic function of X.

3 Example 1: Classifying coronavirus patients sent to an intensive care unit (ICU) based on age and average hours of exercise per week.

3.1 Background

Right now, with the COVID-9 epidemic ravaging the world, there are a lot of people who are being diagnosed with coronavirus. Suppose we have data on people who are diagnosed with coronavirus, their age (between 20 and 80 years old), and how much time do they work out every week (0-40 hours). Then, we can classify whether or not a coronavirus patient gets sent to the ICU based on these variables. We simulate this data under the assumption that both age and average duration of exercise have significant effect; with a positive linear relationship between the response and age, and a negative linear relationship between the response and average amount of exercise. Then, we create 4 samples from our simulated data: the training sets (Figure 3.1 & Figure 3.4) and the testing sets (Figure 3.2 & Figure 3.5). We will use the training set to train our classifiers and use the testing set to determine which classifier makes the fewest errors.

3.2 Case with small training set

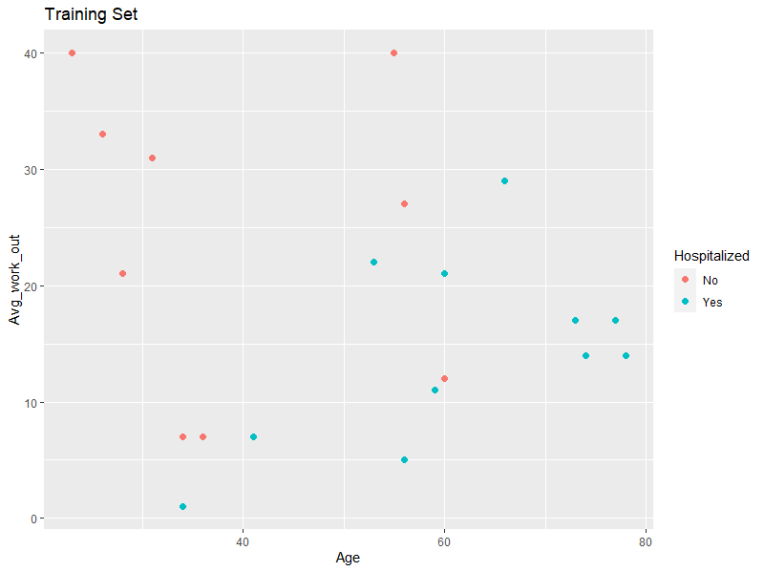

Figure 3.1: Small training data of coronavirus patients with Age and AWO

Let's say our training set and our test set are both 20 in size. We generate a training set as shown in Figure 3.1 above. And the generated testing set is shown in Figure 3.2 below.

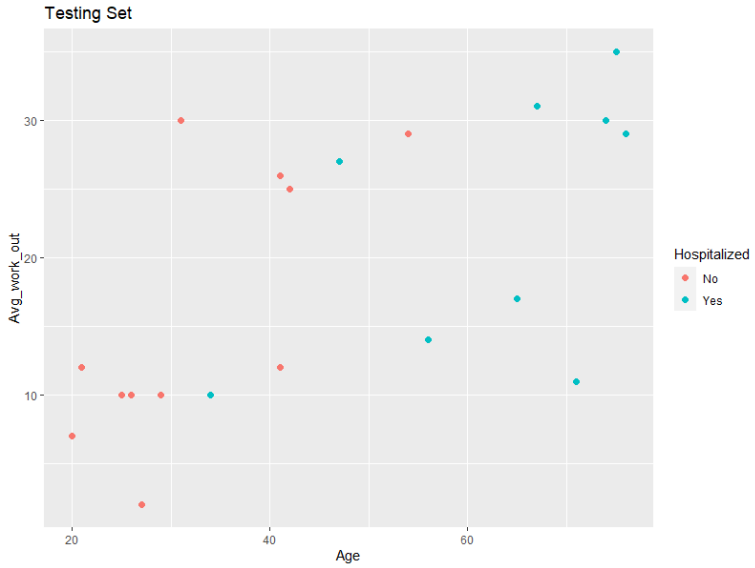

Figure 3.2: Testing data of coronavirus patients with Age and AWO

From these two data sets, we can see that the risk of being sent in ICU is positively affected by age and negatively affected by average duration of exercise.

Next, we use the training data to train our classifiers and use the testing data to see how well our trained classifier work. The results are summarized in Figure 3.3 below.

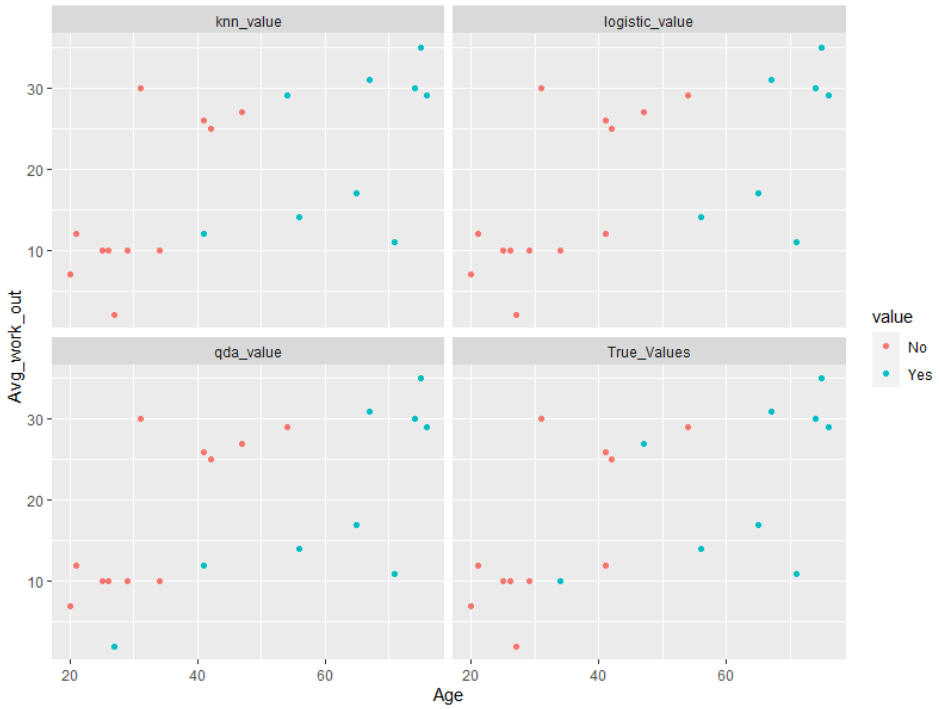

Figure 3.3: Classification results of different classifiers of coronavirus patients with Age and AWO

As you can see from Figure 3.3, 4 patients are misclassified (by misclassify, we mean that the classifier gives a wrong prediction) with the KNN algorithm, 2 patients are misclassified with LR and 4 patients are misclassified with QDA. To compare the efficiency of these classifiers, we use the concept of error rate explained in section 2. We have:

- The error rate for KNN is 0.2;
- The error rate for QDA is 0.2;
- The error rate for LR is 0.1.

It is clear that the Logistic Regression Classifier gives the best prediction.

3.3 Case with large training set

One may be aware that most classifiers converge either uniformlly or under certain conditions. Specifically, for our chosen classifiers, KNN strongly uniformally converges, and both LR and QDA may converge when the data is linearly separable. Since KNN will eventually become equivalent to the best classifier given enough training data, one may think we should just use KNN for relatively large sample sizes. Here, we demonstrate why this would be a mistake. We make a training set with size of 1000 and a testing set whose size is still 20.

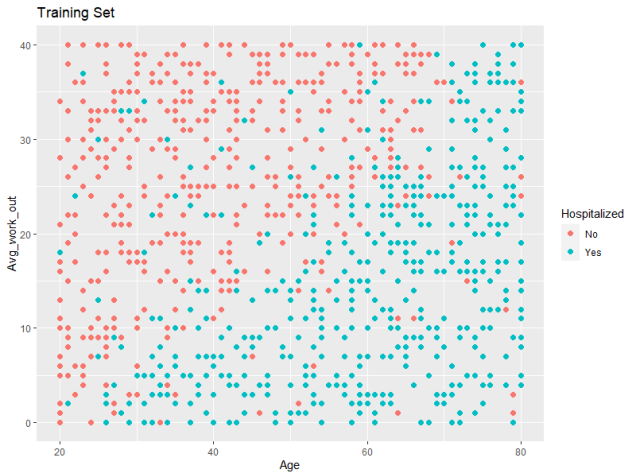

Figure 3.4: Large training data of coronavirus patients with Age and AWO

The generated training set and testing set are shown in Figure 3.4 and Figure 3.5 respectively. From Figure 3.4 we can see that with larger sample, the relationship between patients sent to ICU or not and age with average duration of exercise is even more convincing.

We train our classifiers with this larger training set and test them with the testing set in Figure 3.5. The results are summarized in Figure 3.6.

From the summarized results, we can see that 2 patients are misclassified with KNN algorithm, 1 patient is misclassified with LR and 1 patient is misclassified with QDA. Still, we use the concept of error rate explained in section 2 to compare the efficiency of these classifiers. We have:

- The error rate for KNN is 0.1;
- The error rate for QDA is 0.05;
- The error rate for LR is 0.05.

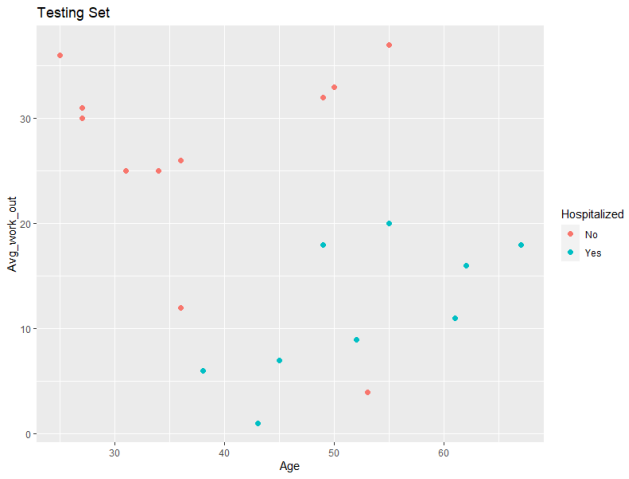

Figure 3.5: Testing data of coronavirus patients with Age and AWO

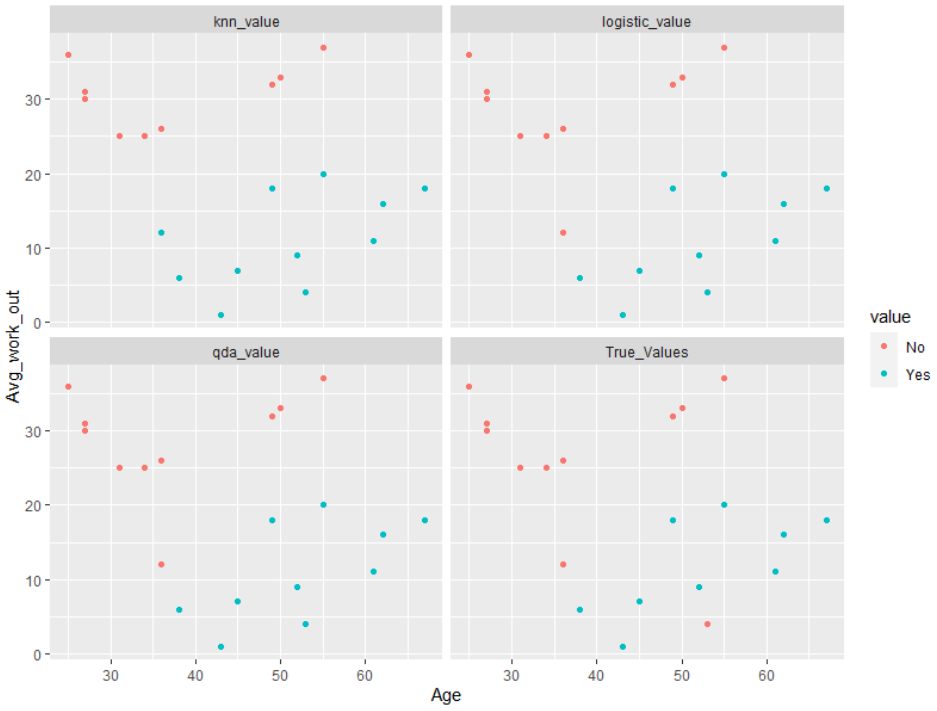

Figure 3.6: Classification results of different classifiers of coronavirus patients with Age and AWO

It is true that these classifiers all become more accurate as the training set gets larger. This shows the concept of convergence and how it works. However, we can see that the Logistic Regression classifier is still one of the best classifiers. QDA has the same error rate as LR. This may be in part due to the random nature of sampling, or also because both QDA and LR perform well with linearly separable data with large sample sizes, and in section 3.4 you'll see that the data we generated is linearly separable. What remains is that KNN is the worst algorithm among these three.

3.4 Conclusion and analysis

In Figure 3.7, you can get a sense of the simulated population for example 1. We generate a sample with size 100000. In order to make the problem more realistic, some amount of random error is including in the simulation, giving the figure a rougher boundary.

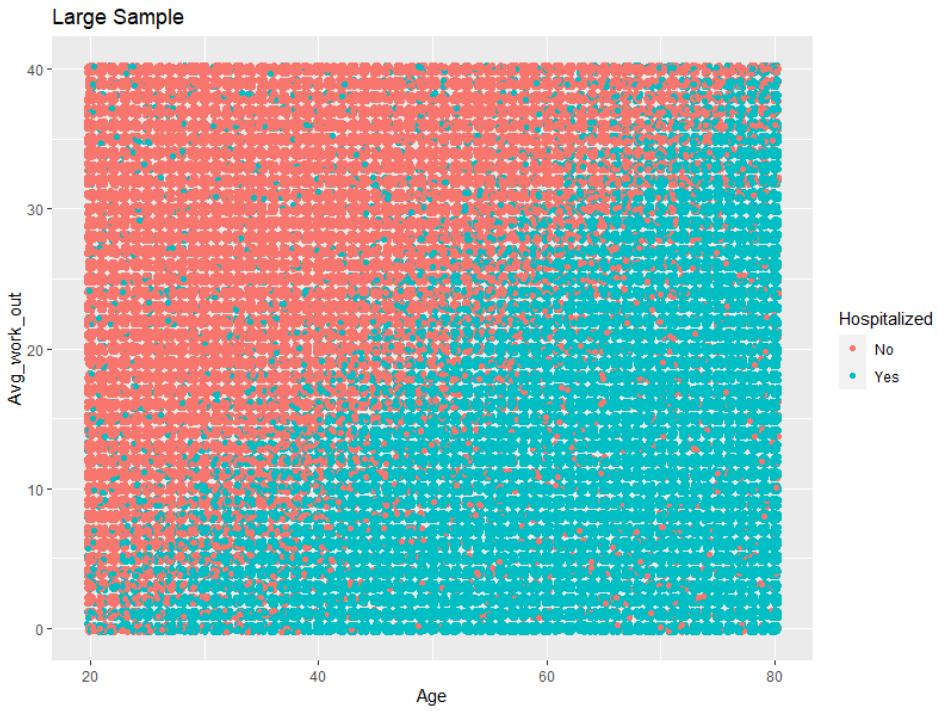

Figure 3.7: Large sample of coronavirus patients with age and AWO

In example 1, logistic regression is best. For small sample size with a linear boundary, logistic regression has an advantage because it matches the assumption of a linear boundary. QDA and KNN are more flexible with their decision boundaries, and as a result they do not perform as well with low sample sizes when the true decision boundary is linear. For large sample size, the linear boundary becomes more evident, QDA and LR may perform with little difference, but both of them are better than KNN.

4 Example 2: Classifying carriers who infect at least 2 other people

4.1 Background

We know that the coronavirus is a dangerous infectious disease, so research of hour the virus spreads is important. It seems that the more you wash your hands and the less time you go to outside, the likely you are to spread the infection. We use these assumptions to simulate a new population. In this problem, we now have data of some virus carriers' average number hours spent out of their home per day and average number of times they wash their hands per day, and we want to use this data to classify a subject by whether they will infected at least 2 other people.

Using the features **Average time subject spent out of home per day (in hours, between 0 and 16)** and **Average number of times subject wash hands per day (between 0 and 15)**, we generate our training set and testing set. Then, use the training set to train our classifiers and use the testing set to compare their performance.

4.2 Case with small training set

First, let's consider a relatively small training set. Suppose both our training set and testing set are of size 25. We generate a training set in Figure 4.1 and a testing set in Figure 4.2.

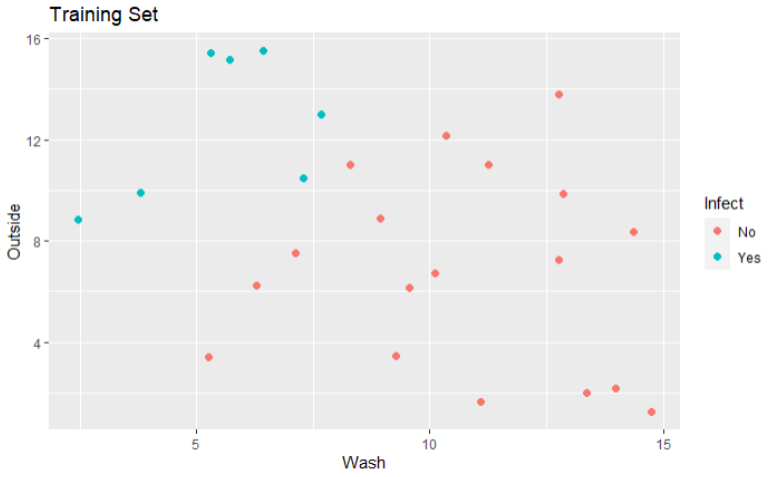

Figure 4.1: Training data of carriers with average home hours and average washing times

In the figures, the red points denote the carries who infect at least 2 other people and the blue points denote the opposite. This will hold for the rest of this section. It matches our intuition that the more a carrier washes his/her hands and the less he/she go out, the less probability he/she will infect other people.

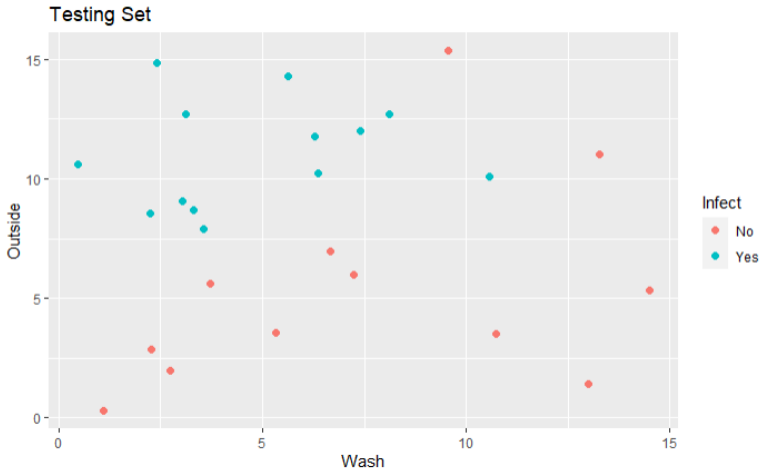

Figure 4.2: Test data of carriers with average home hours and average washing times

Next, we use the training data to train our classifiers and use the testing data to see how well our trained classifier work. The results are shown in Figure 4.3. In Figure 4.3, we can see that the KNN classification misclassifies several high-risk (blue) carriers into low-risk(red) which is really dangerous and the LR classification misclassifies several low-risk(red) carriers into high-risk(blue). It is clear that the QDA classification has less error on classification which reveals its good performance on this kind of data.

To compare the test with error rate, we can see that KNN misclassifies 7 points, LR misclassifies 6 points and QDA only misclassifies 3 points. Calculating the error rate, we get the results as follows:

- The error rate for KNN is 0.28;
- The error rate for QDA is 0.12;
- The error rate for LR is 0.24.

This also implies that QDA is best as it has lowest error rate.

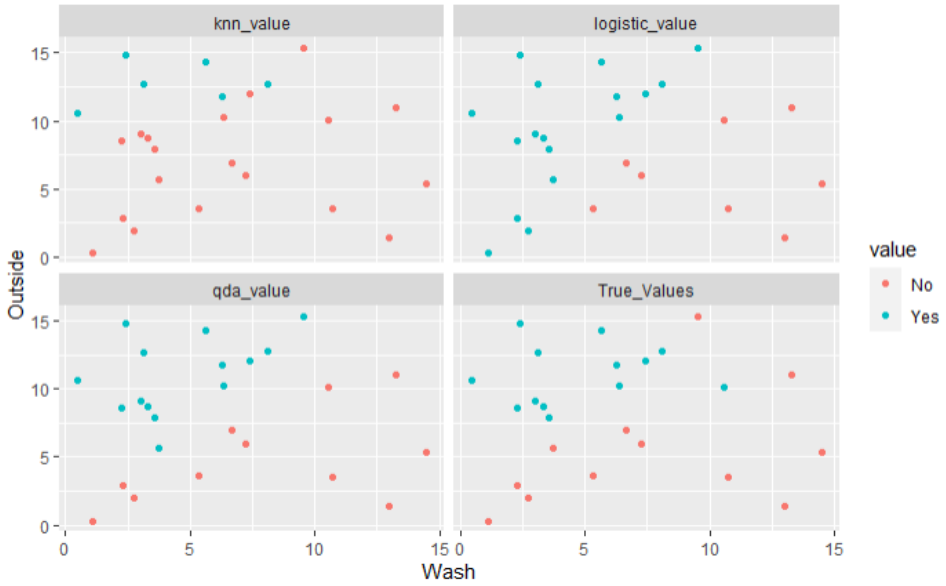

Figure 4.3: Classification results of different classifiers of carriers with average home hours and average washing times

4.3 Case with large training set

Again, one may say something about "convergence",e.g. once you have large sample, just use KNN. Let's consider a large training set under the background of this example.

Now we make a training set of size 1000, and the size of testing set remains 25. The data we generate are shown in Figure4.4 and Figure4.5.

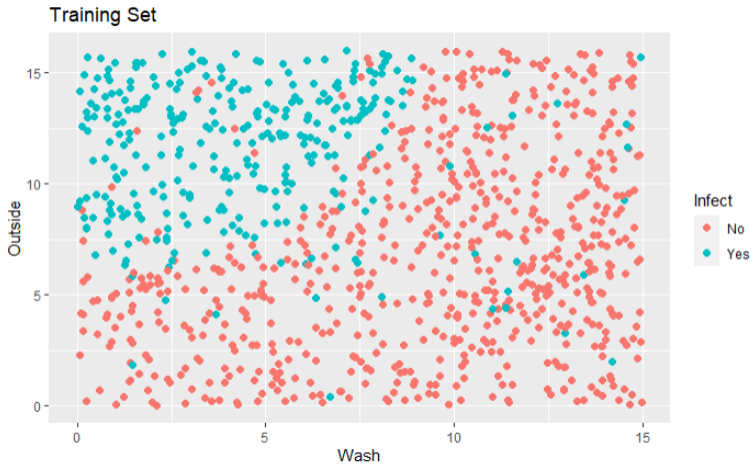

Figure 4.4: Training data of carriers with average home hours and average washing times

From Figure4.4, there is evidence that the boundary of our stimulated data is a convex curve with some noise.

Train these classifiers with the training set and get their prediction of the testing set. The results of these classifiers are shown in Figure4.6.

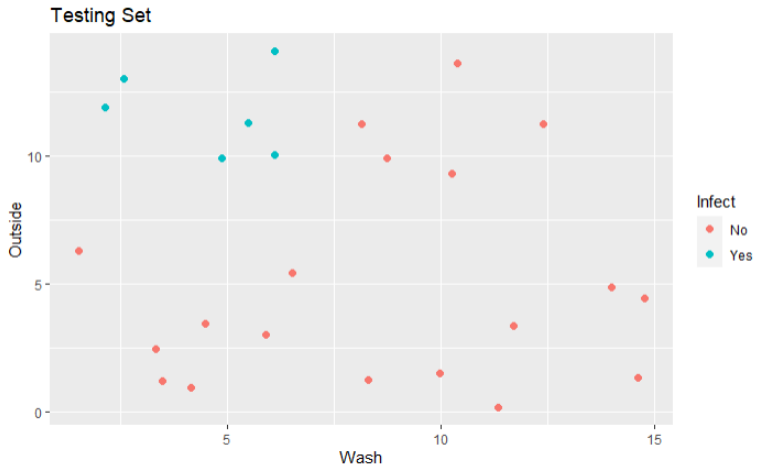

Figure 4.5: Testing data of carriers with average home hours and average washing times

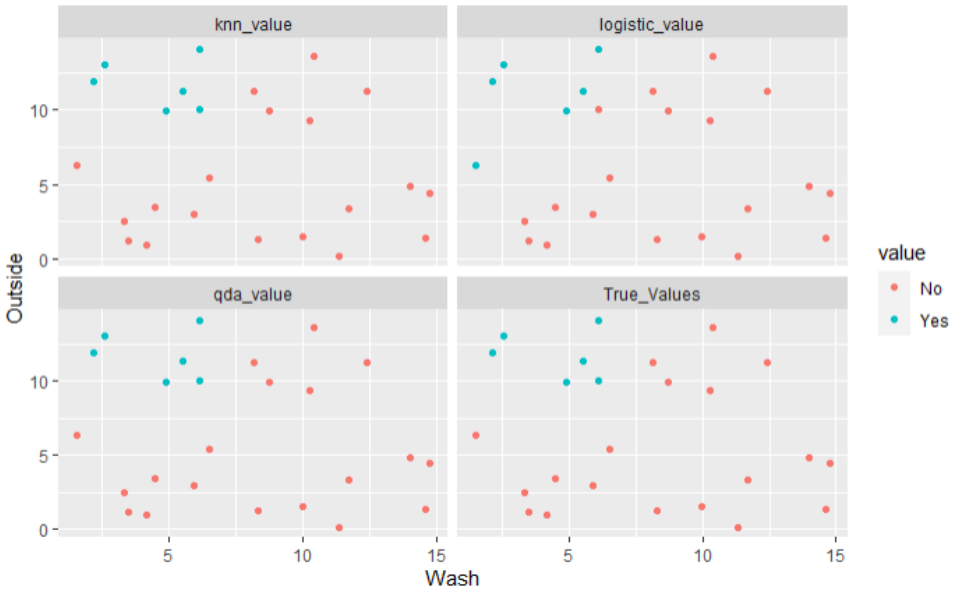

Figure 4.6: Classification results of different classifiers of carriers with average home hours and average washing times

From these figures above, we can see that KNN and QDA perfectly classify the testing data (by perfectly classify, we mean that the prediction for every point is correct) while LR has one point misclassified. Calculate the error rate, we have:

- The error rate for KNN is 0.00;
- The error rate for QDA is 0.00;
- The error rate for LR is 0.04.

Thus QDA is still one of the best classifiers as it has lowest error rate. Notice that KNN perfectly classifies our testing data just as QDA does.

4.4 Conclusion and analysis

Here is a look at the simulated population from example 2 which you can see in Figure4.7. We generated a sample with size 100000.

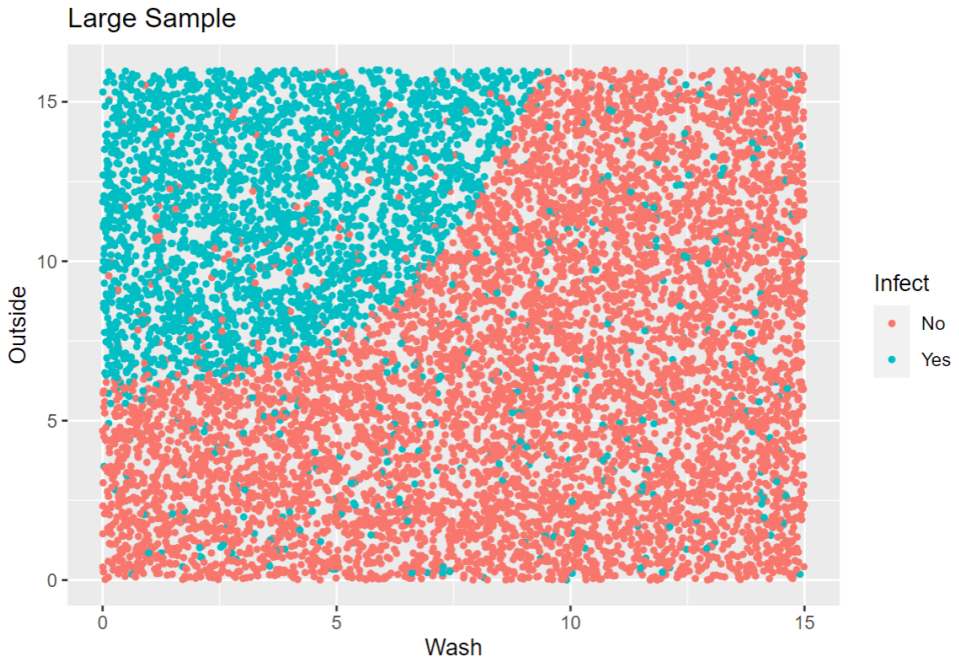

Figure 4.7: Large sample classification of QDA of carriers with average home hours and average washing times

In Example 2: QDA is best. In limited sample sizes with a curvilinear boundary, QDA is best because it assumes a curvilinear boundary. In example 1 we have already known that for linear boundary problem LR shows better performance. In fact, when the boundaries are moderately non-linear, QDA may give better results. In our large sample, QDA and KNN had equal performance. However, it is difficult to know just how large of a sample is large enough for KNN to perform the best. To make this determination, one must consider each problem individually.

Finally, for much more complicated decision boundaries, a non-parametric approach such as KNN can be superior which you can see in the next section.

5 Example 3: Classifying whether coronavirus is transmitted based on climate conditions

5.1 Background

Now, researchers in our team find that the infection rates are very different around the world. The biologist in our team thinks that climatic conditions may be influencing the spread of COVID-9. He believes that **relative humidity** and **temperature** are two very important factors affecting virus transmission. Thus, we want to explore this relationship. Say we start an experiment with guinea pigs to determine if infection of the coronavirus is affected by relative humidity and temperature. The following is our researchers' test design.

Test Outline: 1 infected guinea pig is caged with 4 non-infected guinea pigs at either 5 or 20 degrees Celsius and 35%, 50%, 65% and 80% relative humidity for 1 day. There are $2 \times 4 = 8$ combinations of temperature and relative humidity. After 1 day, we seperate the guinea pigs and count the number of non-infected guinea pigs who become infected.

Note: we will use knowledge from previous viral studies to simulate this experiment.

5.2 Case with small training set

First, let's consider a small training set. Suppose both our training set and testing set have the size of 100.

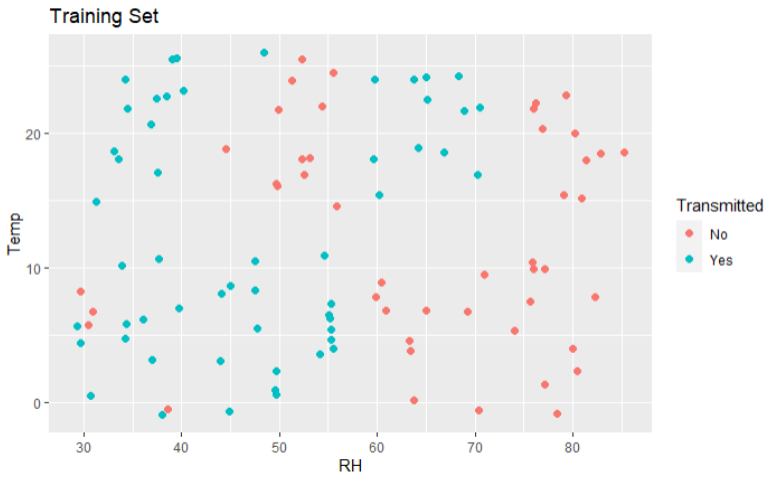

Figure 5.1: Training data of carriers with relative humidity and temperature

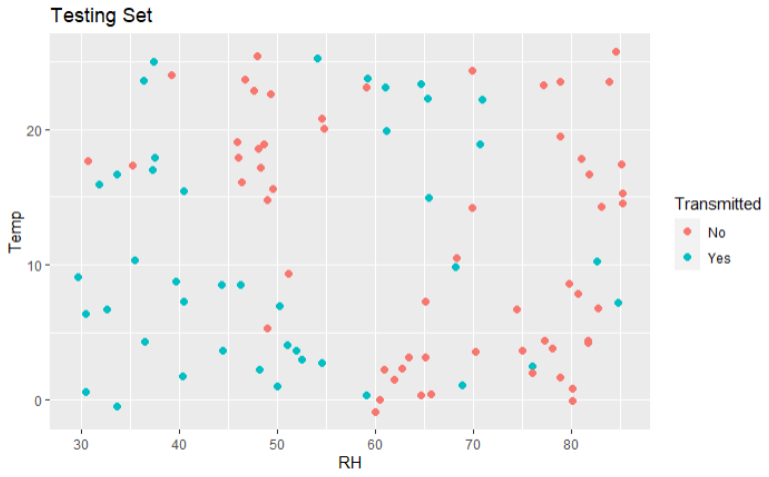

Figure 5.2: Testing data of transmission with relative humidity and temperature

We generate a training set which you can see in Figure 5.1 and a testing set which you can see in Figure 5.2. In these figures, the horizontal axis implies relatively humidity and the vertical axis denotes the temperature, blue points implies that the virus is transmitted among pigs in that cage and red points denote the opposite. Both relative humidity and temperature are measured at discrete values with only 8 unique combinations, therefore jitter plots are used to visualize the number of observations for each combination of relative humidity and temperature. Next, we classify the points in the testing set based on the training set.

The classification results are shown in Figure5.3. From these figures, it doesn't appear that you can draw a line in the figure and perfectly classify the data. Unlike LR and QDA, KNN does not need to classify with a linear boundary, and therefore has an advantage in this situation, even with a small sample size. We have the following error rates for each classifier.

- The error rate for KNN is 0.15;
- The error rate for QDAis 0.33;
- The error rate for LR is 0.33.

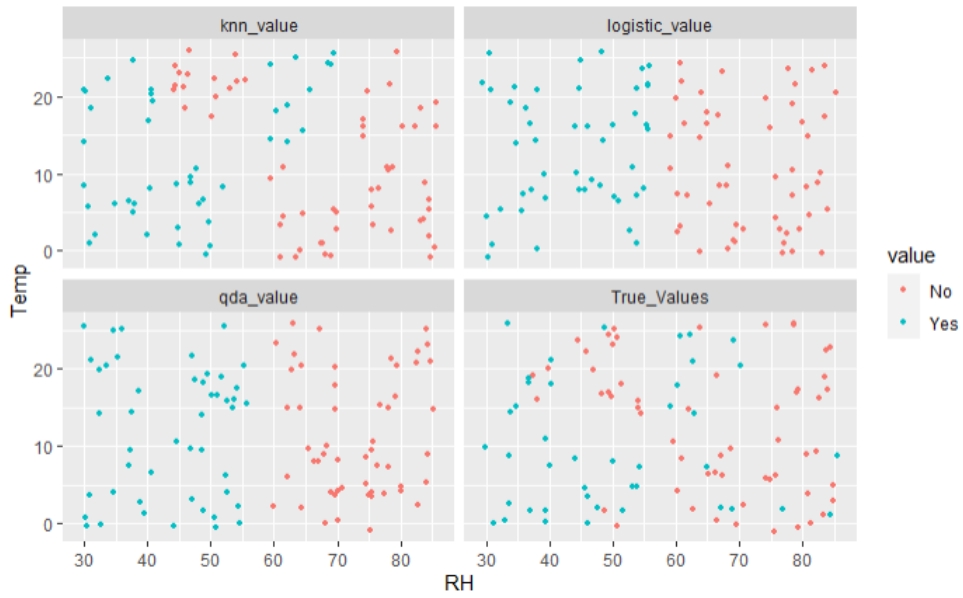

Figure 5.3: Classification results of different classifiers of transmission with relative humidity and temperature

It is clear that KNN is the best classifier in this case.

5.3 Case with large training set

Again, we will test on a larger sample. In this case we will find that only KNN works well while the others are performing bad. This is because the data we generate are not linearly separable.

Now we make a training set of size 1000 and a testing set of size 100. The data we generated are shown in Figure5.4 and Figure5.5.

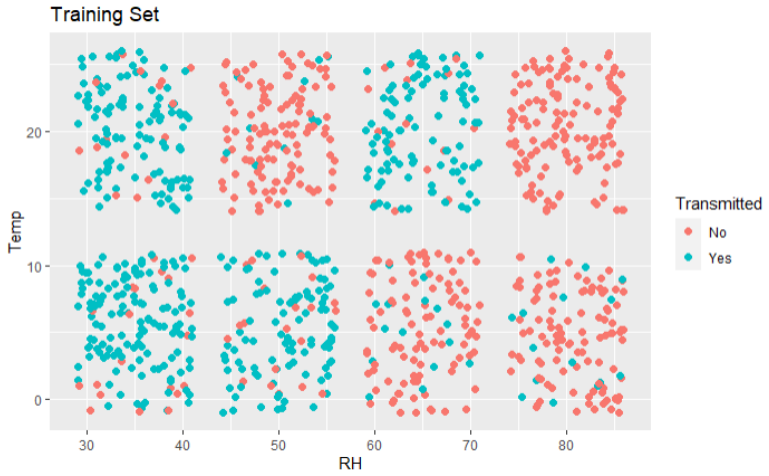

Figure 5.4: Training data of carriers with relative humidity and temperature

The classification results are shown in Figure5.6. From these figures, we can see that the results are very similar as the case with small training set. KNN is able to classify the data in clusters while LR and QDA attempt to classify with a linear boundary. We have the following error rates for each classifier.

- The error rate for KNN is 0.12;

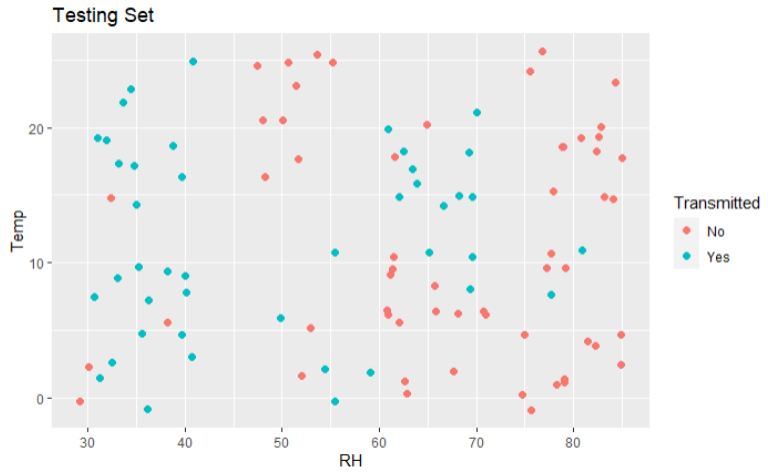

Figure 5.5: Testing data of transmission with relative humidity and temperature

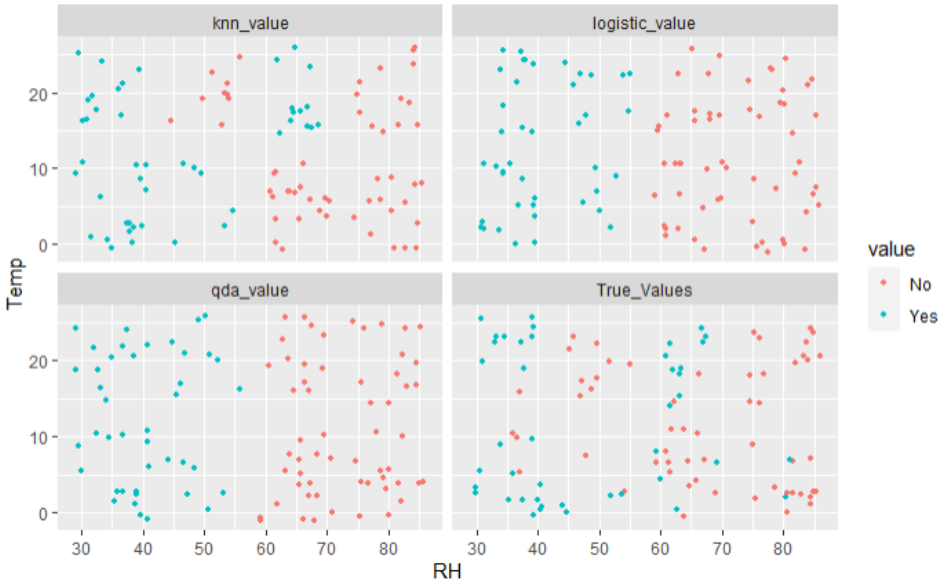

Figure 5.6: Classification results of different classifiers of transmission with relative humidity and temperature

- The error rate for QDAis 0.31;
- The error rate for LR is 0.31.

In the larger sample, KNN is again the best classifier. Also, notice that the difference between the error rate by KNN with small training set and the error rate by KNN eith large training set is small. By expanding the training data tenfold, the accuracy was improved by only a fifth. So, the convergence rate is very small. In this case, it may not be worth it to spend money on a larger sample.

5.4 Conclusion and analysis

In example 3, KNN performed the best. For complicated boundaries and/or with sufficient sample size, KNN performs either better than, or as well as, the other 2 classifiers.

In addition, here is a look at the simulated population for example 3 which you can see in Figure 5.7:

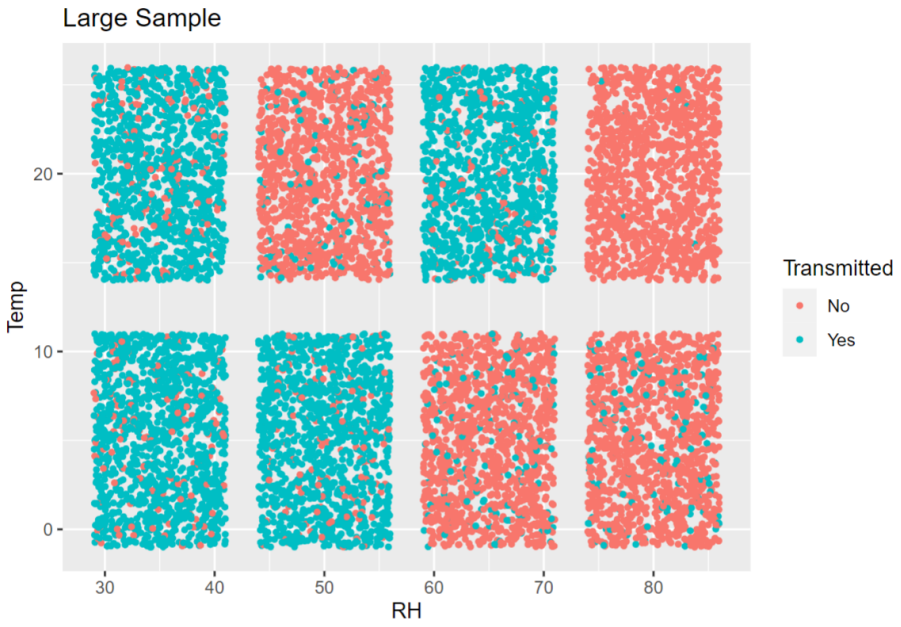

Figure 5.7: Large sample classification of KNN of transmission with relative humidity and temperature

Figure 5.7 reveals the distribution of the sample space. This figure shows the population of transmission with relative humidity and temperature, we can see more details: the groups of combination (temp=20, RH=50), (temp=5, RH=65), (temp=5, RH=80), (temp=20, RH=80), show a lower infect rate for most of the observations in these groups are not transmitted. The boundary of transmitted and non-transmitted is not linear or curvilinear, it shows more complexity, and therefore QDA and LR do not perform as well as KNN.

6 Conclusion

Different classifiers have different advantages and disadvantages. For instances, KNN is a general model that can be applied to the data from any distribution. For example, data does not have to be separable with a linear boundary, as in example 3. A downside of KNN is that although it performs well generally, with enough data, it can be outperformed by models that work well in specific situations. Furthermore, changing the parameter K can change the resulting predicted class label. In Example 1 we use $K = 8$ and it lead to bad results.

Logistic Regression excels when the data has a linear classification bound, especially for binary data. While the other two models may catch up to the performance of LR given enough data, LR will generally perform as well, or better than LR and KNN given limited sample sizes. This was seen in example 1, when LR showed really good classification. However, for data with a very non-linear boundary, logistic regression is not the best classifier to pick, as can be seen in examples 2 and 3.

QDA performs the best when the decision boundary is curvilinear. While KNN and LR may catch up given enough sample size, with a curvilinear boundary, QDA will generally perform as well or better than LR and KNN given a limited sample size. Meanwhile, when the boundary is not curvilinear, like in examples 1 and 3, QDA is not the best classifier to choose.

In conclusion, there is no single best classifier for all problems. One must consider the specifics of each problem and use their knowledge to decide which classifier will likely perform the best in that situation.

7 Appendix

To get our code, please check this website: https://github.com/RuyanZhou/NoFreeLunchThm.git.