Lab1

Jacob Krol

2026-01-29

Lab1: Comparing Classification Methods

Due: 11:59PM Friday, February 6 on Canvas as a knitted pdf file of your team's Quarto document

1. You will individually fit five classification models and compare their sensitivities and specificities.
2. You will interpret these models and make a prediction in your individual section.
3. As a team, you will create one visualization that summarizes the performance of the models.

Instructions for Lab1

In Lab0, the professor created a generating model for Y variables taking on 0 or 1 values based on the X1 and X2 inputs. You came up with the backstory for what Y, X1, and X2 were and why it was important to use X1 and X2 to predict Y.

In this lab, you will apply four newly learned statistical learning methods to continue your analyses for how to best predict or explain Y from X1 and X2. You should continue to use the Q1 qualitative context you developed in Lab0 (or you can develop a new one if you want).

Each individual will fit five classification models on a training dataset and then evaluate how well those models classify Y based on a test dataset. Individuals will make a prediction given (x1, x2) and then interpret their prediction and make a recommendation for action. Just as in Lab0, each individual will describe the Q1, Q2, and Q3 for this "project". Specifically, for Q1: What is Y, X1, and X2, and why should we care?. Train your five models on the training set, compare the sensitivity and specificity of the models on the test set, make a prediction given (x1, x2) (this is Q2). Then describe what actions (Q3) you recommend given your Q1 context and your Q2 results. Reflect on some ethical aspect of this project.

Generating Model

We have a logistic regression generating model. Given $x_1 \in [0,1]$ and $x_2 \in [0,1], Y \sim Ber(p)$, where p is related to x_1 and x_2 through the logistic link function: $\log(\frac{p}{(1-p)}) = x_1 - 2x_2 - 2{x_1}^2 + {x_2}^2 + 3x_1x_2$, where log is the natural log, sometimes written as ln.

The code for this is below.

```
library(class)
suppressPackageStartupMessages(library(tidyverse))
```

```
#Generative model
set.seed(200) #setting a random seed so that we can
#reproduce everything exactly if we want to

generate_y <- function(x1,x2) { #two input parameters to generate the output y
  logit <- x1 -2*x2 -2*x1^2 + x2^2 + 3*x1*x2
  p <- exp(logit)/(1+exp(logit)) #apply the inverse logit function
  y <- rbinom(1,1,p) #y becomes a 0 (with prob 1-p) or a 1 with probability p
}
```

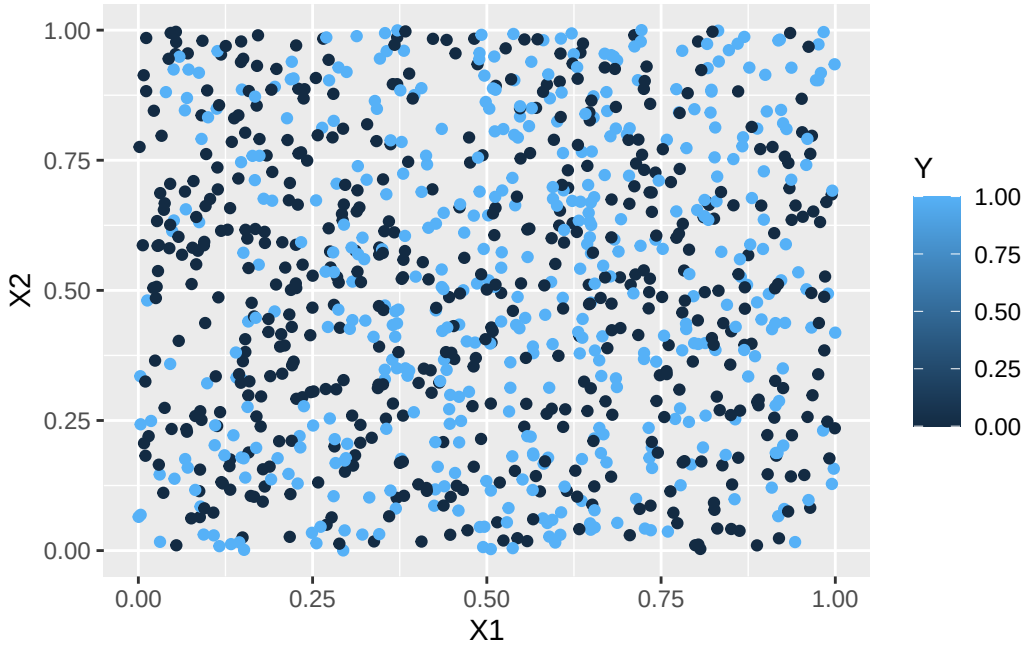
Training dataset

We are going to use our generating model to create a training dataset of 1000 predictors (x1, x2), and then 1000 outcomes. Then we plot all three variables to see what our training data looks like.

```
# Generate a training dataset with 1000 points
set.seed(200)
n = 1000
X1 <- runif(n,0,1)
X2 <- runif(n,0,1)

#I'm going to use a for loop to generate 1000 y's
Y <- rep(0,n) #initializing my Y to be a vector of 0's
for (i in 1:n) {
  Y[i] <- generate_y(X1[i],X2[i])
}

sum(Y) #How many 0's and 1's were predicted? In this
```

```
[1] 468
```

```r
#training set, almost 47% were 1's. When n is very large
# about 48% of the Y's are 1's. That's close enough to
# 50/50 so we shouldn't have issues with "imbalance"
# which is something we'll learn about later in the
# semester.

training <- cbind(X1,X2,Y) #combining all of my variables into a training dataset
ggplot(data=training, aes(x=X1, y=X2, color=Y)) +
  geom_point()
```



How well will various classifiers predict Y given new x1 and x2 values?

Test datasets

Each individual will generate a test set of 1000 predictors (x1, x2) and outcomes (y) that we will use as our "ground truth".

So, create your individual test dataset (using random seed=201, 202, 203, or 204; each teammate has a different test dataset).

What individuals need to do

1. Given the training set (seed=200), fit:

    1. logistic regression model
    2. linear discriminant analysis (LDA)
    3. quadratic discriminant analysis (QDA)
    4. naive Bayes
    5. KNN with k=optimal k from Lab0
2. Calculate the sensitivity, specificity, and overall error rate (misclassification rate) for each model given your test set (your seed=201 or 202 or 203 or 204).
3. Make a prediction for a new point (x1, x2) = (0.25, 0.25) or (0.25, 0.75) or (0.75, 0.25) or (0.75, 0.75) for each fitted model. Each individual will have a different point for their predictions.
4. Summarize the Q1, Q2, and Q3 aspects of this "project."

What teams need to do

1. Summarize the models' performance using all of the results from each individual.
2. Which is the best model to use in this situation?

Some intended outcomes from this assignment:

- You will individually get practice fitting a logistic regression, LDA, QDA, naive Bayes, and KNN
- You will compare the performance of these five models
- You will make predictions based on statistical learning models
- You will practice describing how the Q1 context affects your Q2 and Q3 outcomes, i.e., you will practice thinking about the whole problem (i.e., Q1Q2Q3, not just the Q2 quantitative parts)
- You will compare models visually
- You will gain experience collaborating with your teammates on an applied problem

Individual section Jake

Train and test setup

Recreating the training data set using given code.

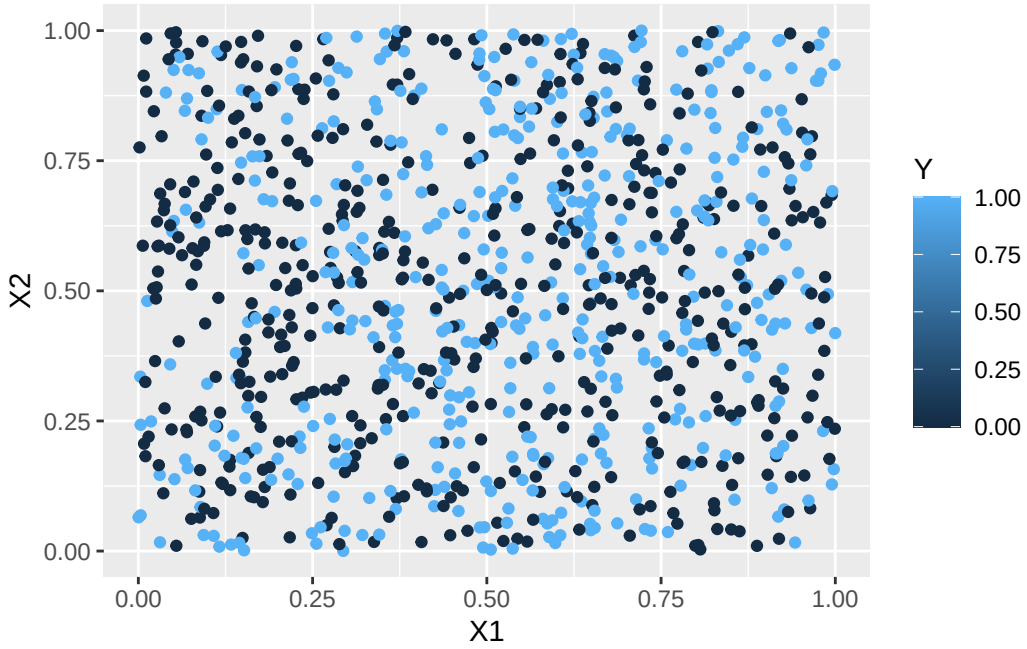
```
library(class)
suppressPackageStartupMessages(library(tidyverse))
```

```r
#Generative model
set.seed(200) #setting a random seed so that we can
#reproduce everything exactly if we want to

generate_y <- function(x1,x2) { #two input parameters to generate the output y
  logit <- x1 -2*x2 -2*x1^2 + x2^2 + 3*x1*x2
  p <- exp(logit)/(1+exp(logit)) #apply the inverse logit function
  y <- rbinom(1,1,p) #y becomes a 0 (with prob 1-p) or a 1 with probability p
}
```

```r
# Generate a training dataset with 1000 points
set.seed(200)
n = 1000
X1 <- runif(n,0,1)
X2 <- runif(n,0,1)

#I'm going to use a for loop to generate 1000 y's
Y <- rep(0,n) #initializing my Y to be a vector of 0's
for (i in 1:n) {
  Y[i] <- generate_y(X1[i],X2[i])
}

sum(Y) #How many 0's and 1's were predicted? In this
```

```
[1] 468
```

```r
#training set, almost 47% were 1's. When n is very large
# about 48% of the Y's are 1's. That's close enough to
# 50/50 so we shouldn't have issues with "imbalance"
# which is something we'll learn about later in the
# semester.

training <- cbind(X1,X2,Y) #combining all of my variables into a training dataset
ggplot(data=training, aes(x=X1, y=X2, color=Y)) +
  geom_point()
```



The seed I'm using for generating test data is 203, and I will evaluate on the point (0.75, 0.25).

```
set.seed(203)
n = 1000

X1_test <- runif(n,0,1)
X2_test <- runif(n,0,1)

Y_test <- rep(0,n)
for (i in 1:n) {
  Y_test[i] <- generate_y(X1_test[i],X2_test[i])
}

test <- as.data.frame(cbind(X1_test,X2_test,Y_test))
point_eval <- data.frame(X1=0.75, X2=0.25)
head(test)
```

```
    X1_test   X2_test Y_test
1 0.1705013 0.4458760      0
2 0.4675975 0.4555522      0
3 0.4635794 0.3444861      0
4 0.1868930 0.1978852      1
```

```
5 0.4206904 0.2205762      1
6 0.2619604 0.7538692      0
```

```r
table(test$Y_test)
```

```
  0   1
528 472
```

The test data is fairly balanced with 528 0s and 472 1s.

Logistic regresssion (Jake)

```r
# fit
logistic_model <- glm(Y ~ X1 + X2, data = as.data.frame(training), family = binomial)
y_pred <- ifelse(predict(logistic_model, newdata = test, type = "response") > 0.5, 1, 0)
# eval
P <- sum(test$Y_test == 1)
N <- sum(test$Y_test == 0)
TP <- sum((y_pred == 1) & (test$Y_test == 1))
TN <- sum((y_pred == 0) & (test$Y_test == 0))
sensitivity_logistic <- TP / P
specificity_logistic <- TN / N
error_rate_logistic <- mean(y_pred != test$Y_test)
print(
    paste0(
        "Logistic Regression: Sensitivity: ",
        round(sensitivity_logistic, 4),
        ", Specificity: ",
        round(specificity_logistic, 4),
        ", Error Rate: ",
        round(error_rate_logistic, 4)
    )
)
```

```
[1] "Logistic Regression: Sensitivity: 0.3708, Specificity: 0.6667, Error Rate: 0.473"
```

```r
# predict
logistic_point_pred <- ifelse(predict(logistic_model, newdata = point_eval, type = "response"
print(paste0("Logistic Regression Prediction at (0.75, 0.25): ", logistic_point_pred))
```

```
[1] "Logistic Regression Prediction at (0.75, 0.25): 1"
```

For logistic regression, the test sensitivity was 0.3708, specificity was 0.6667, and overall error rate was 0.473. The evaluation point (0.75, 0.25) was predicted to be in class 1.

Linear Discriminant Analysis (Jake)

```
library(MASS)
```

```
Attaching package: 'MASS'

The following object is masked from 'package:dplyr':

    select
```

```
# fit
lda_model <- lda(Y ~ X1 + X2, data = as.data.frame(training))
y_pred <- predict(lda_model, newdata = test)$class
# eval
TP <- sum((y_pred == 1) & (test$Y_test == 1))
TN <- sum((y_pred == 0) & (test$Y_test == 0))
sensitivity_lda <- TP / P
specificity_lda <- TN / N
error_rate_lda <- mean(y_pred != test$Y_test)
print(
    paste0(
        "LDA: Sensitivity: ",
        round(sensitivity_lda, 4),
        ", Specificity: ",
        round(specificity_lda, 4),
        ", Error Rate: ",
        round(error_rate_lda, 4)
    )
)
```

```
[1] "LDA: Sensitivity: 0.3729, Specificity: 0.661, Error Rate: 0.475"
```

```r
# predict
lda_point_pred <- predict(lda_model, newdata = point_eval)$class
print(paste0("LDA Prediction at (0.75, 0.25): ", lda_point_pred))
```

```
[1] "LDA Prediction at (0.75, 0.25): 1"
```

For LDA, the test sensitivity was 0.3729, specificity was 0.661, and overall error rate was 0.475.

Again, the evaluation point (0.75, 0.25) was predicted to be in class 1.

Quadratic Discriminant Analysis (Jake)

```r
# fit
qda_model <- qda(Y ~ X1 + X2, data = as.data.frame(training))
y_pred <- predict(qda_model, newdata = test)$class
# eval
TP <- sum((y_pred == 1) & (test$Y_test == 1))
TN <- sum((y_pred == 0) & (test$Y_test == 0))
sensitivity_qda <- TP / P
specificity_qda <- TN / N
error_rate_qda <- mean(y_pred != test$Y_test)
print(
    paste0(
        "QDA: Sensitivity: ",
        round(sensitivity_qda, 4),
        ", Specificity: ",
        round(specificity_qda, 4),
        ", Error Rate: ",
        round(error_rate_qda, 4)
    )
)
```

```
[1] "QDA: Sensitivity: 0.3136, Specificity: 0.6894, Error Rate: 0.488"
```

```r
# predict
qda_point_pred <- predict(qda_model, newdata = point_eval)$class
print(paste0("QDA Prediction at (0.75, 0.25): ", qda_point_pred))
```

```
[1] "QDA Prediction at (0.75, 0.25): 0"
```

The QDA test sensitivity was 0.3136, specificity was 0.6894, and overall error rate was 0.488.

In contrast to logistic regression and LDA, the evaluation point (0.75, 0.25) was predicted to be in class 0 by QDA.

Naive Bayes (Jake)

```
library(e1071)
```

```
Attaching package: 'e1071'

The following object is masked from 'package:ggplot2':

    element
```

```
# fit
training_nb <- data.frame(
    X1=X1,
    X2=X2,
    Y=factor(Y)
)
test_nb <- data.frame(
    X1=X1_test,
    X2=X2_test,
    Y_test=factor(Y_test)
)
nb_model <- naiveBayes(Y ~ X1 + X2, data = training_nb)
y_pred <- predict(nb_model, newdata = test_nb)
# eval
TP <- sum((y_pred == 1) & (test_nb$Y_test == 1))
TN <- sum((y_pred == 0) & (test_nb$Y_test == 0))
sensitivity_nb <- TP / P
specificity_nb <- TN / N
error_rate_nb <- mean(y_pred != test_nb$Y_test)
print(
    paste0(
        "Naive Bayes: Sensitivity: ",
        round(sensitivity_nb, 4),
        ", Specificity: ",
        round(specificity_nb, 4),
```

```
        ", Error Rate: ",
        round(error_rate_nb, 4)
    )
)
```

```
[1] "Naive Bayes: Sensitivity: 0.5233, Specificity: 0.5587, Error Rate: 0.458"
```

```
# predict
nb_point_pred <- predict(nb_model, newdata = point_eval)
print(paste0("Naive Bayes Prediction at (0.75, 0.25): ", nb_point_pred))
```

```
[1] "Naive Bayes Prediction at (0.75, 0.25): 1"
```

For naive Bayes, the test sensitivity was 0.5233, specificity was 0.5587, and overall error rate was 0.458. The evaluation point (0.75, 0.25) was predicted to be in class 1.

K-Nearest Neighbors (Jake)

In lab 0, we found k=8 was most optimal.

```
library(class)
k <- 8
# fit and predict
y_pred <- knn(train = as.data.frame(training[,1:2]),
              test = as.data.frame(test[,1:2]),
              cl = as.factor(training[,3]),
                k = k)
# eval
TP <- sum((y_pred == 1) & (test$Y_test == 1))
TN <- sum((y_pred == 0) & (test$Y_test == 0))
sensitivity_knn <- TP / P
specificity_knn <- TN / N
error_rate_knn <- mean(y_pred != test$Y_test)
print(
    paste0(
        "KNN (k=8): Sensitivity: ",
        round(sensitivity_knn, 4),
        ", Specificity: ",
        round(specificity_knn, 4),
        ", Error Rate: ",
```

```
        round(error_rate_knn, 4)
    )
)
```

```
[1] "KNN (k=8): Sensitivity: 0.4809, Specificity: 0.5606, Error Rate: 0.477"
```

```
# predict
knn_point_pred <- knn(train = as.data.frame(training[,1:2]),
                      test = as.data.frame(point_eval),
                      cl = as.factor(training[,3]),
                      k = k)
print(paste0("KNN Prediction at (0.75, 0.25): ", knn_point_pred))
```

```
[1] "KNN Prediction at (0.75, 0.25): 1"
```

For KNN (k=8), test sensitivity was 0.4936, specificity was 0.5947, and overall error rate was 0.453. The evaluation point (0.75, 0.25) was predicted to be in class 1.

Cleanup environment.

```
rm(list=ls())
```

QQQ (Jake)

Q1

The problem context is that a hosptial wants to automate diagnosis of disease based on two factors: i) a patient's diagnostic test result

$$\in [0, 1]$$

(X1) and ii) doctor's belief that patient has disease

$$\in [0, 1]$$

(X2). The context here is that diseases can be challenging to diagnose and a model which integrates both test results and doctor's beliefs may improve diagnostic accuracy from solely relying on either factor alone. Furthermore, diseases often have serious health implications, so the accuracy, implementation, and consentual use of a disease diagnostic model has serious ethical considerations. Type I errors (false positives) could lead to unnecessary treatments

while type II errors (false negatives) could lead to a worsened condition of patient due to inadequate treatment.

Hypothetically, a hosptial's electronic health record (EHR) database houses patient data with diagnostic test results (X1), doctor's belief scores (X2), and confirmed disease diagnoses (Y). The goal is to evaluate various classification models to predict disease diagnosis (Y) based on X1 and X2 to consider whether a model may improve diagnostic accuracy in clinical practice. Again, there are many ethical considerations. Have the model developers received consent from patients and doctors to use their data to develop a diagnostic model? Which diseases will the model be used for? Which range of accuracies are acceptable for which diseases? Who will take the blame for incorrect model predictions?

Q2

The data would be securely extracted from the EHR system and processed into a 3 column table (X1, X2, Y) where X1 is diagnostic test result, X2 is doctor's belief score, and Y is confirmed disease diagnosis (0=no disease, 1=disease). The 2000 total records would be split in half into a training set and test set (1000 records each). Logistic regression, LDA, QDA, naive Bayes, and KNN models would be fit on the training set to predict Y from X1 and X2. Then, each model would be evaluated on the test set to calculate sensitivity, specificity, and overall error rate. Furthermore, we are interested in a specific case where a test point has X1=0.75 confidence in the disease, yet the doctor has only X2=0.25 confidence in the disease. With respect to the quantiative plan for this project, there are a few ethical considerations. Are more interpretable models, such as logistic regression, preferred compared to less interpretable models like KNN? How will data be secured during this project? Who will have access to the quantitative results of this project?

Q3

Conclusions will be drawn by comparing the sensitivity, specificity, and overall error rates of each model on the test set. Both quantitative performance and intepretability will be valued when considering whether a model is suitable for clinical implementation. The case where test and doctor's belief mismatch (X1=0.75, X2=0.25) will be analyzed across models to understand any patterns in agreement/disagreement across models. Perhaps, this will indicate whether a test's belief or doctor's belief is more effective in predicting disease diagnosis.

A final recommendation of whether to implement the diagnostic model in practice will be based on the test set metrics, interpretability, and ethical considerations. If the test set metrics are high for interpretable models, then a follow-up would be to discuss with clinicians and patients to understand their perspectives on using such a model in practice. If the metrics are low, then further data collection and model refinement would be recommended before considering clinical implementation. Ethical considerations such as patient consent, data security, and accountability for incorrect predictions will be central to any final recommendations.

Conclusions (Jake)

I believe the overall error rate was too high (>0.45) across all models to recommend clinical implementation. However, there was some agreement in the test point of interest (X1=0.75, X2=0.25). It was predicted to be in class 1 (disease) by all models except QDA, which predicted class 0 (no disease). QDA also had the highest error rate (0.488) and lowest sensitivity (0.3136). Again, predicting greater than 45% of cases incorrectly is not acceptable for clinical implementation, in my opinion.