Comparison of classification methods

Logistic regression has a linear boundery:

$$\log(\frac{P(Y=1|x)}{1-P(Y=1|x)}) = \beta_0 + \beta_1 x$$

$P(Y=1|x) > 0.5$ is equivalent to $\beta_0 + \beta_1 x > 0$.
LDA has a linear log odds:

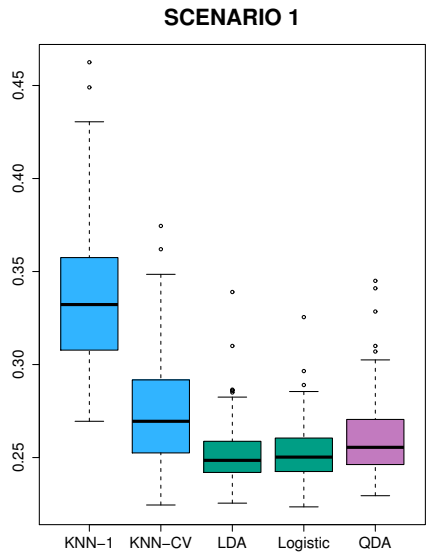
$$\log(\frac{P(Y=1|x)}{1-P(Y=1|x)}) = \frac{\mu_1 - \mu_0}{\sigma^2}x - \frac{1}{\sigma^2}(\mu_1 - \mu_0)^2 + \log\frac{\pi_1}{\pi_0}$$

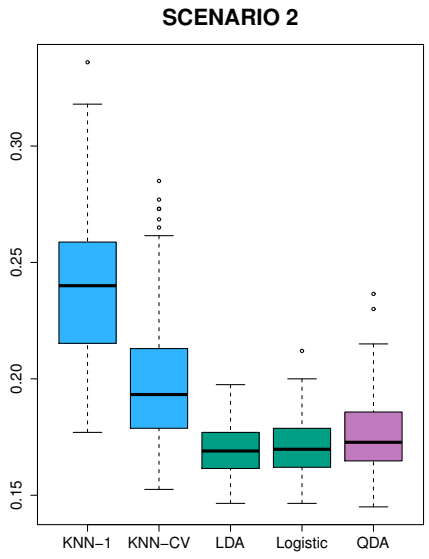
The difference between LDA and logistic regression: The linear coefficients are estimated differently. MLE for logistic models and estimated mean and variance based on Gaussian assumptions for the LDA. LDA makes more restrictive Gaussian assumptions and therefore expected to work better than logistic models if they are met.

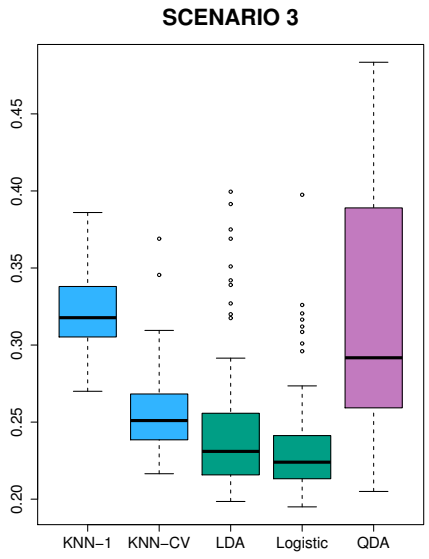
KNN is a completely non-parametric approach: no assumptions are made about the shape of the decision boundary. Therefore, we can expect this approach to dominate LDA and logistic regression when the decision boundary is highly non-linear. On the other hand, KNN does not tell us which predictors are important; we don't get a table of coefficients with p-values.

QDA serves as a compromise between the non-parametric KNN method and the linear LDA and logistic regression approaches. Since QDA assumes a quadratic decision boundary, it can accurately model a wider range of problems than can the linear methods. Though not as flexible as KNN, QDA can perform better in the presence of a limited number of training observations because it does make some assumptions about the form of the decision boundary.

Scenario 1: There were 20 training observations in each of two classes. The observations within each class were uncorrelated random normal variables with a different mean in each class.
Scenario 2: Details are as in Scenario 1, except that within each class, the two predictors had a correlation of -0.5.
Scenario 3: We generated X_1 and X_2 from the t-distribution, with 50 observations per class. In this setting, the decision boundary was still linear, and so fit into the logistic regression framework. The set-up violated the assumptions of LDA, since the observations were not drawn from a normal distribution.


SCENARIO 1
0.25
0.30
0.35
0.40
0.45
KNN−1
KNN−CV
LDA
Logistic
QDA

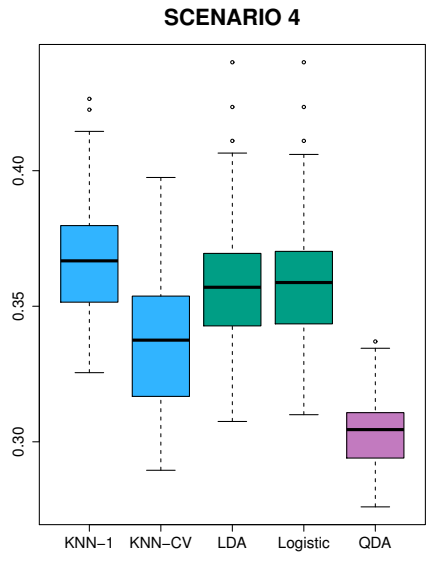

SCENARIO 2
0.15
0.20
0.25
0.30
KNN−1
KNN−CV
LDA
Logistic
QDA

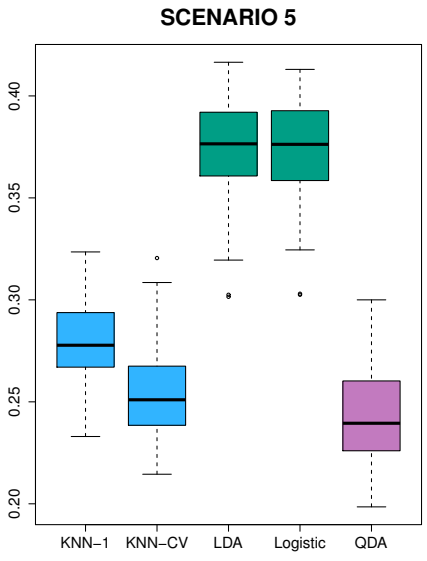

SCENARIO 3
0.20
0.25
0.30
0.35
0.40
0.45
KNN−1
KNN−CV
LDA
Logistic
QDA

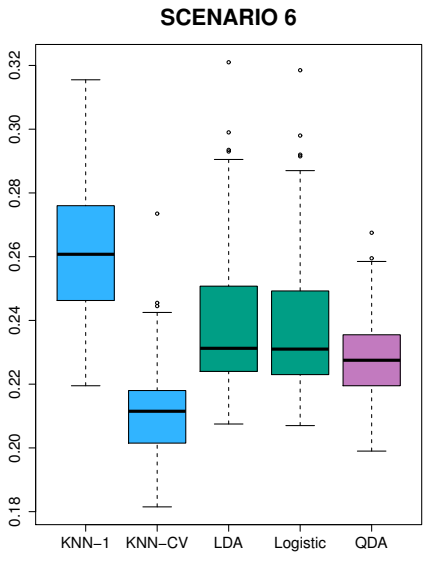
Scenario 4: The data were generated from a normal distribution, with a correlation of 0.5 between the predictors in the first class, and correlation of -0.5 between the predictors in the second class. This setup corresponded to the QDA assumption, and resulted in quadratic decision boundaries.

Scenario 5: Within each class, the observations were generated from a normal distribution with uncorrelated predictors. However, the responses were sampled from the logistic function using X_1^2, X_2^2, and $X_1 \times X_2$ as predictors. Consequently, there is a quadratic decision boundary.

Scenario 6: Details are as in the previous scenario, but the responses were sampled from a more complicated non-linear function. As a result, even the quadratic decision boundaries of QDA could not adequately model the data.


SCENARIO 4
0.30
0.35
0.40
KNN−1
KNN−CV
LDA
Logistic
QDA


SCENARIO 5
0.20
0.25
0.30
0.35
0.40
KNN−1
KNN−CV
LDA
Logistic
QDA


SCENARIO 6
0.18
0.20
0.22
0.24
0.26
0.28
0.30
0.32
KNN−1
KNN−CV
LDA
Logistic
QDA

Lab: Logistic Regression, LDA, QDA, and KNN

The **Smarket** data set consists of percentage returns for the S&P 500 stock index over 1, 250 days, from the beginning of 2001 until the end of 2005. For each date, we have recorded the percentage returns for each of the five previous trading days, **lag1** through **Lag5**. We have also recorded **Volume** (the number of shares traded on the previous day, in billions), **Today** (the percentage return on the date in question) and **Direction** (whether the market was Up or Down on this date)

```
> # The Stock Market Data
>
> library(ISLR)
> names(Smarket)
> dim(Smarket)
> summary(Smarket)
> pairs(Smarket)
> cor(Smarket[,-9])
> attach(Smarket)
> plot(Volume)
```

```
> # Logistic Regression
>
> glm.fit=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,
+             data=Smarket, family=binomial)
> summary(glm.fit)
> coef(glm.fit)
> summary(glm.fit)$coef
> summary(glm.fit)$coef[,4]
> glm.probs=predict(glm.fit,type="response")
> glm.probs[1:10]
> contrasts(Direction)
> glm.pred=rep("Down",1250)
> glm.pred[glm.probs>.5]="Up"
> table(glm.pred,Direction)
> (507+145)/1250
> mean(glm.pred==Direction)
```

```
> #create test data
> train=(Year<2005)
> Smarket.2005=Smarket[!train,]
> dim(Smarket.2005)
> Direction.2005=Direction[!train]
> glm.fit=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,
+         data=Smarket, family=binomial,subset=train)
> glm.probs=predict(glm.fit,Smarket.2005,type="response")
> glm.pred=rep("Down",252)
> glm.pred[glm.probs>.5]="Up"
> table(glm.pred,Direction.2005)
> mean(glm.pred==Direction.2005)
> mean(glm.pred!=Direction.2005)
```

```
> #With two predictors
> glm.fit=glm(Direction~Lag1+Lag2,data=Smarket,
+             family=binomial,subset=train)
> glm.probs=predict(glm.fit,Smarket.2005,type="response")
> glm.pred=rep("Down",252)
> glm.pred[glm.probs>.5]="Up"
> table(glm.pred,Direction.2005)
> mean(glm.pred==Direction.2005)
> 106/(106+76)
> predict(glm.fit,newdata=data.frame(Lag1=c(1.2,1.5),
+                   Lag2=c(1.1,-0.8)),type="response")
```

```
> # Linear Discriminant Analysis
> library(MASS)
> lda.fit=lda(Direction~Lag1+Lag2,data=Smarket,
+             subset=train)
> lda.fit
> plot(lda.fit)
> lda.pred=predict(lda.fit, Smarket.2005)
> names(lda.pred)
> lda.class=lda.pred$class
> table(lda.class,Direction.2005)
> mean(lda.class==Direction.2005)
> sum(lda.pred$posterior[,1]>=.5)
> sum(lda.pred$posterior[,1]<.5)
> lda.pred$posterior[1:20,1]
> lda.class[1:20]
> sum(lda.pred$posterior[,1]>.9)
```

```
> # Quadratic Discriminant Analysis
>
> qda.fit=qda(Direction~Lag1+Lag2,data=Smarket,
+             subset=train)
> qda.fit
> qda.class=predict(qda.fit,Smarket.2005)$class
> table(qda.class,Direction.2005)
> mean(qda.class==Direction.2005)
```

```
> # K-Nearest Neighbors
>
> library(class)
> train.X=cbind(Lag1,Lag2)[train,]
> test.X=cbind(Lag1,Lag2)[!train,]
> train.Direction=Direction[train]
> set.seed(1)
> knn.pred=knn(train.X,test.X,train.Direction,k=1)
> table(knn.pred,Direction.2005)
> (83+43)/252
> knn.pred=knn(train.X,test.X,train.Direction,k=3)
> table(knn.pred,Direction.2005)
> mean(knn.pred==Direction.2005)
```

```
> # An Application to Caravan Insurance Data
>
> dim(Caravan)
> attach(Caravan)
> summary(Purchase)
> 348/5822
```

This data set includes 85 predictors that measure demographic characteristics for 5,822 individuals. The response variable is Purchase, which indicates whether or not a given individual purchases a caravan insurance policy. In this data set, only 6 % of people purchased caravan insurance.

```
> standardized.X=scale(Caravan[,-86])
> var(Caravan[,1])
> var(Caravan[,2])
> var(standardized.X[,1])
> var(standardized.X[,2])
> test=1:1000
> train.X=standardized.X[-test,]
> test.X=standardized.X[test,]
> train.Y=Purchase[-test]
> test.Y=Purchase[test]
> set.seed(1)
> knn.pred=knn(train.X,test.X,train.Y,k=1)
> mean(test.Y!=knn.pred)
> mean(test.Y!="No")
> table(knn.pred,test.Y)
> 9/(68+9)
```

```
> knn.pred=knn(train.X,test.X,train.Y,k=3)
> table(knn.pred,test.Y)
> 5/26
> knn.pred=knn(train.X,test.X,train.Y,k=5)
> table(knn.pred,test.Y)
> 4/15
> glm.fit=glm(Purchase~.,data=Caravan,family=binomial,
+             subset=-test)
> glm.probs=predict(glm.fit,Caravan[test,],
+                   type="response")
> glm.pred=rep("No",1000)
> glm.pred[glm.probs>.5]="Yes"
> table(glm.pred,test.Y)
> glm.pred=rep("No",1000)
> glm.pred[glm.probs>.25]="Yes"
> table(glm.pred,test.Y)
> 11/(22+11)
```