Exercise set 2

Logistic regression and the K-nearest neighbors (KNN) classifier for a problem with two classes, zero and one

- In logistic regression (with one predictor), we model the conditional *probability* of $Y = 1$ given $X = x_0$.

$$P(Y = 1|X = x_0) = \frac{e^{\beta_0+\beta_1 x_0}}{1 + e^{\beta_0+\beta_1 x_0}} = \frac{1}{1 + e^{-\beta_0-\beta_1 x_0}}.$$

- The KNN classifier estimates the conditional probability of $Y = 1$ given $X = x_0$ by

$$P(Y = 1|X = x_0) = \frac{1}{K} \sum_{x_i \in \mathbb{N}_0} I(y_i = 1)$$

  where I is the indicator function and $\mathbb{N}_0$ are the K training-points that are closest to x_0.

Task 1

Consider the dataset

x	y
1	0
2	1
3	0
4	1
5	1
6	1
7	1

a) Fit a logistic regression of Y on X by

```
m1=glm(y~x,family="binomial")
summary(m1)
```

and write down the estimated equation for $P(Y = 1|X = x)$. Also, predict Y for $x_0 = 4$.

b) Compute the estimated probabilities and plot them on top of the observations by

```
plot(x,y)
m1=glm(y~x,family="binomial")
prob=predict(m1,type="response")
lines(x,prob,col="blue")
```

Which numbers do you get from the predict-function if you do not use the argument `type="response"`? Check by computing them yourself and by reading the help-function.

```
help(glm)
```

c) Use KNN with $K = 1, 3$ and 5 to estimate the conditional probability for $x_0 = 4$ by

```r
knn=function(x0,x,y,K)
{
  d=abs(x0-x)
  o=order(d)
  prob=mean(y[o[1:K]])
  return(prob)
}
```

Explain each row of the code. Why can we interpret the result as a probability?

d) Plot estimated probabilities, using KNN for $x = 1, 2, ..., 7$, on top of the observed data.

```r
prob_knn=matrix(0,7,1)
for(i in 1:7) prob_knn[i]=knn(x0=x[i],x,y,K=3)
plot(x,y)
lines(x,prob_knn)
```

Can you add lines of estimated probabilities for $K = 1$, $K = 5$ and from logistic regression?

e) Predict Y for $x = 1, 2, ..., 7$, i.e., in the training data, by logistic regression and KNN and produce a confusion matrix. Use the estimated probabilities and convert them to predicted values (zero or one). For logistic regression:

```r
pred=prob>0.5
table(y,pred)
prop.table(table(y,pred),margin = 1)
```

To produce predictions from KNN, use e.g.,

```r
pred_knn=prob_knn>0.5
```

Linear discriminant analysis and logistic regression

- Discriminant analysis, in general, uses the principle of allocating (predicting) an observation i to the class k with the largest estimated $p_k(x_i) = P(Y = k|X = x_i)$ among all classes k
- Assume that we have two classes and the predictors, $\boldsymbol{X}$, in all classes, is normally distributed, with different expectation but the same variance (and covariances in the case of a multivariate $\boldsymbol{X}$). Then $p_1(x_i) > p_2(x_i)$ implies that $\delta_1(x_i) > \delta_2(x_i)$, where

$$\delta_k(x_i) = \ln \pi_k + \frac{\mu_k x_i}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2}$$

  Since the *discriminant function* is a linear function of x_i, this approach is called *linear* discriminant analysis.

Task 2

Load the dataset Default in the R-package ISLR and do some descriptive data analysis to answer the following questions. The overarching aim is to predict the variable default.

a) Use both logistic regression and linear discriminant analysis to estimate the probability of default. Split the data 50/50 into a training and a test dataset. Start by

```r
library(ISLR)
n=nrow(Default)
ind=sample(1:n,size=floor(n/2))
train=Default[ind,]
test=Default[-ind,]
```

```
logreg=glm(default~.,data=train,family="binomial")
library(MASS)
lda=lda(default~balance+income,data=train)
```

b) Predict the test data using the estimated models. Compute confusion matrices. Do you get better predictions by removing some of the variables?

c) Are the X-variables student, balance and income, all suitable for linear discriminant analysis? Any other methods that can be more suitable?

Cross-validation methods

- In the *validation set approach*, the dataset is split randomly in a training and a test dataset.
- In the *leave-one-out (LOO) approach*, the prediction model is fitted to a training data consisting of all but one observations and evaluated on the left-out observation. This leave-one-out is repeated for all observations.
- In the *k-fold cross-validation approach*, the observations are split into k parts where one part is used as the test data. This, leave-one-group-out, is then repeated for all groups.
- The term *test error* is here used, generically, for any evaluation of a prediction in the test data, e.g. testMSE or testER.

Task 3

a) Consider the Auto dataset and create a class variable, y, which is "high" for observations where mpg is above its median and "low" when it is below. Convert it to a factor-variable. Compute age of the cars. Remove mpg, name and year.

```
library(ISLR)
Auto$y="low"
Auto$y[Auto$mpg>median(Auto$mpg)]="high"
Auto$y=as.factor(Auto$y)
Auto$age=83-Auto$year
Auto=Auto[,!(names(Auto) %in% c("mpg","name","year"))]
```

`mpg` is removed since it is substituted with `y`. `name` is removed since it contains too many unique values. `year` is removed since it is substiuted with the linear transformation `age`. Explain each line of the code.

We will now use the validation set approach to investigate how well classification models, based on other variables in the dataset, predicts y.

b) Split the data randomly into 50/50 training/test data, fit a logistic regression of y on all other variables and conclude which variables that you would like to try out as predictors.

c) Predict y in the test data and compute a confusion matrix.

d) Do the same thing with the leave-one-out apprach. Make sure that you understand what the code is doing.

```
n=nrow(Auto)
pred=matrix(0,n,1)
for(i in 1:n)
{
  train=Auto[-i,]
  test=Auto[i,]
  m1=glm(y~.,data=train,family="binomial")
  prob=predict(m1,newdata=test)
```

```
  pred[i]=prob>0.5
}
prop.table(table(Auto$y,pred),margin=1)
```

e) Do the same thing with 8-fold CV. Make sure that you understand what the code is doing.

```
n=nrow(Auto)
k=8
s=n/k # Size of each fold
pred=matrix(0,n,1)
ii=1
for(i in 1:k)
{
  train=Auto[-(ii:(ii+s-1)),]
  test=Auto[ii:(ii+s-1),]
  m1=glm(y~.,data=train,family="binomial")
  prob=predict(m1,newdata=test)
  pred[ii:(ii+s-1)]=prob>0.5
  ii=ii+s
}
prop.table(table(Auto$y,pred),margin=1)
```

f) Would you prefer to use all predictors (input variables) or only some of them?

The bootstrap

- In the bootstrap, we resample from our original sample, with replacement, in order to estimate the distribution of a sample quantity, such as a sample mean or an estimated regression coefficient.
- A 95% is an interval which covers the population value (true value) of the quantity of interest, e.g., a population mean.

Task 4

In this task we should compute a 95% confidence interval for the expected value of mpg, in the Auto-dataset.

a) Do this by assuming that the average of all observations of mpg is normally distributed. Does this assumption sound plausible?

b) Compute the 95% confidence interval using the bootstrap.