# STAT318/462 Assignment 1

## Question 1 (4 marks: 1 for an advantage, 1 for a disadvantge, 2 for the conditions)

**Diadvantages** can be difficult to interpret, can be difficult for inference, may not be useful for noisy data, may not be useful when the training data set is small, can overfit the training data…

**Advantages** may be able to better capture nonlinear features, aviod underfitting the training data, better model complex problems, work well with larger training data sets, may provide more accurate predictions …

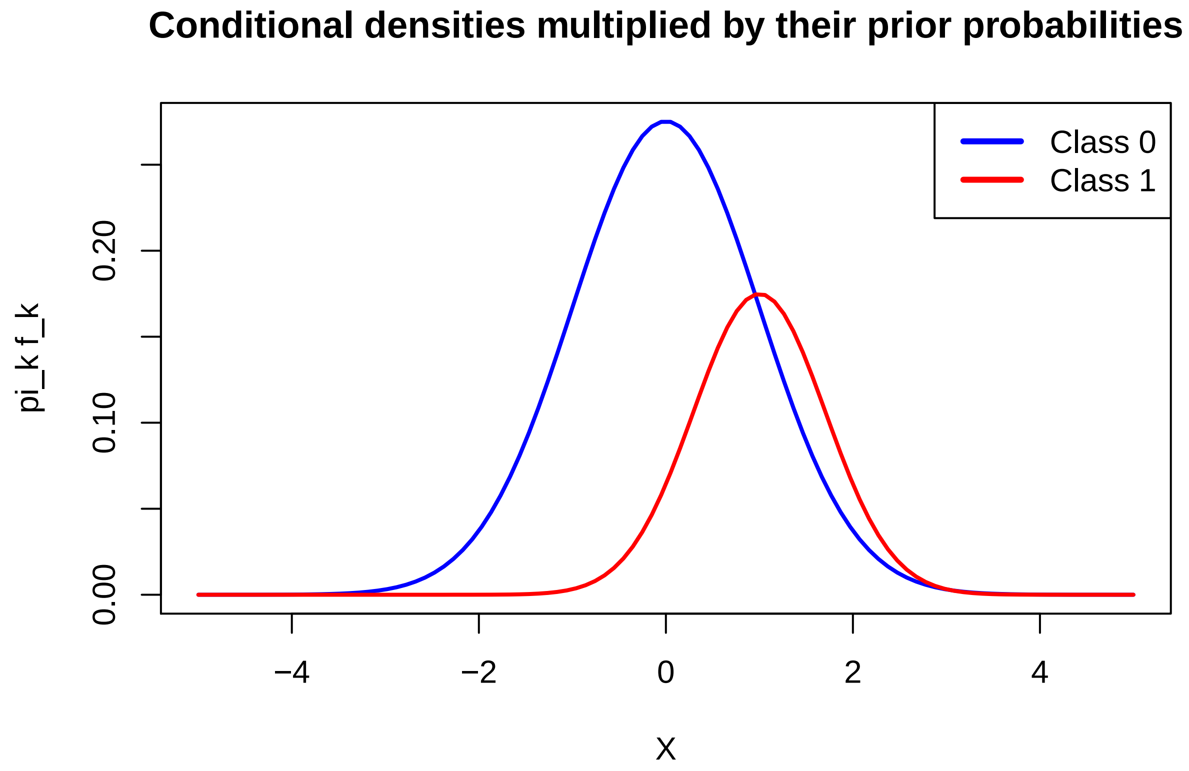
**Conditions** simple problems, relatively small training data sets, when you're interested in inference rather than just predictions, when the data is noisy, …

**As long as their basic arguments are correct give them the marks, but not if incorrect statements are made.**

## Question 2 (6 marks)

### (a) (2 marks: 1 for the plot and 1 for labelling)

```r
x = seq(-5,5,length=100)
plot(x,
     0.69*dnorm(x),
     col = "blue",
     type = "l",
     main = "Conditional densities multiplied by their prior probabilities",
     ylab = "pi_k f_k",
     xlab = "X",
     lwd=2)
points(x,
       0.31*dnorm(x,1,sqrt(0.5)),
       col="red",
       type="l",
       lwd=2)
legend("topright",
       legend = c("Class 0", "Class 1"),
       col = c("blue","red"),
       lwd = 3,
       text.col = "black",
       horiz = FALSE)
```



### (b) (2 marks: 1 for each point on the boundary)

```r
polyroot(c(log(0.69/(0.31*sqrt(2))) + 1,-2,1/2))
```

```
## [1] 0.9545773-0i 3.0454227+0i
```

### (c) (1 mark: 0.5 for correct classification and 0.5 for reason)

$X = 3$ is in class 1 because $\mathrm{P}(Y = 0|X = 3) < \mathrm{P}(Y = 1|X = 3)$ or $\pi_0 f_0(3) < \pi_1 f_1(3)$.

### (d) (1 mark)

$\mathrm{P}(Y = 1|X = 2) = \frac{\pi_1 f_1(2)}{\pi_0 f_0(2)+\pi_1 f_1(2)}$

```r
(0.31/sqrt(pi)*exp(-(1^2)))/(0.69/sqrt(2*pi)*exp(-(2^2)/2) + 0.31/sqrt(pi)*exp(-(1^2)))
```

```
## [1] 0.6333126
```

## Question 3 (8 marks)

### (a) (3 marks: 1 for running kNN, 1 for the training error and 1 for test error)

```r
train = read.csv("AutoTrain.csv")
test = read.csv("AutoTest.csv")
k = c(2,5,10,20,30,50,100);
test.error = numeric(7);
```

```r
train.error = numeric(7);
for (i in 1:7){
  pred = kNN(k[i],
             train$horsepower,
             train$mpg,
             test$horsepower)
  test.error[i] = mean((pred - test$mpg)^2)
  pred = kNN(k[i],
             train$horsepower,
             train$mpg,
             train$horsepower)
  train.error[i] = mean((pred - train$mpg)^2)
}
test.error
```

```
## [1] 22.86349 19.56322 18.62914 17.31858 17.93018 19.57374 26.31542
```

```r
train.error
```

```
## [1] 11.67317 15.39669 17.38083 17.49457 18.99924 19.47530 26.25969
```

```r
kNN.error = min(test.error)
kNN.error
```

```
## [1] 17.31858
```

```r
kbest = k[which.min(test.error)]
kbest
```

```
## [1] 20
```

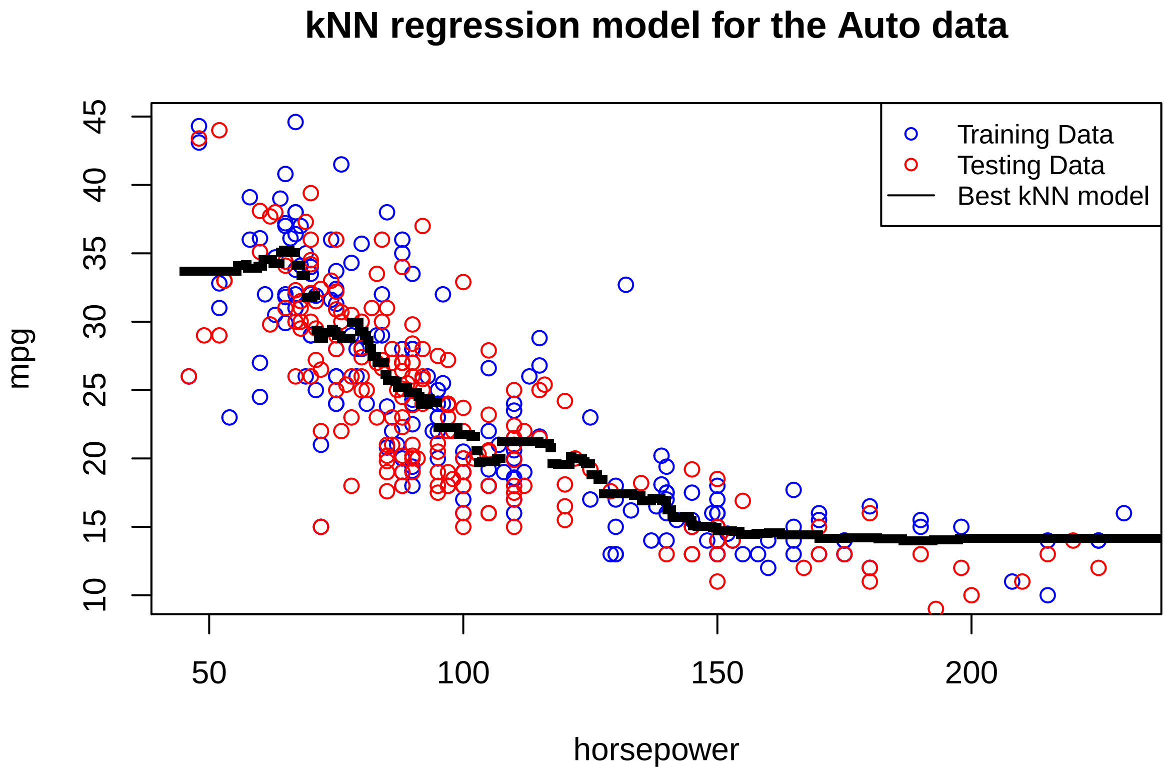
### (b) (1 mark: 0.5 for the best k and 0.5 for a valid reason)

The best value of $k$ is 20, because this gives the lowest test MSE.

### (c) (2 marks: 2 for plot. Deduct 0.5 if the labelling is poor, deduct 0.5 if the plots don't include predictions for all values of horsepower)

```r
plot(train$horsepower,
     train$mpg,
     col="blue",
     main = "kNN regression model for the Auto data",
     xlab = "horsepower",
     ylab = "mpg")
points(test$horsepower,
     test$mpg,
     col="red"
)
x.val = seq(45,250,length=2000)
kNN.val = kNN(kbest,
               train$horsepower,
               train$mpg,
               x.val)
points(x.val,
       kNN.val,
       pch=15,
       col="black",
       cex=0.6)

legend("topright",
       legend = c("Training Data", "Testing Data","Best kNN model"),
       col = c("blue","red","black"),
       pch= c(21,21,NA),
       lty=c(NA,NA,1),
       cex = 0.8,
       text.col = "black",
       horiz = FALSE)
```



### (d) (2 marks)

Larger neighbourhoods (ones that include many points) tend to produce models with high bias and low variance. Small neighbourhoods (ones that include few points) tend to produce models with low bias and high variance.