S632 HW3

#1.

6. Suppose we collect data for a group of students in a statistics class with variables $X_1 =$ hours studied, $X_2 =$ undergrad GPA, and $Y =$ receive an A. We fit a logistic regression and produce estimated coefficient, $\hat{\beta}_0 = -6, \hat{\beta}_1 = 0.05, \hat{\beta}_2 = 1$.

   (a) Estimate the probability that a student who studies for 40 h and has an undergrad GPA of 3.5 gets an A in the class.

   (b) How many hours would the student in part (a) need to study to have a 50 % chance of getting an A in the class?

(a) From the formula (4.2) (p.134),

$$\hat{p}(X) = \frac{e^{-6+0.05X_1+X_2}}{1+e^{-6+0.05X_1+X_2}} = 0.3775$$

where $X_1 = 40$ and $X_2 = 3.5$.

(b) In the same way with (a),

$$\frac{e^{-6+0.05X_1+3.5}}{1+e^{-6+0.05X_1+3.5}} = 0.5.$$ From this, $$e^{-6+0.05X_1+3.5} = 1$$

and $-6+0.05X_1+3.5 = 0. \Rightarrow 0.05X_1 = 2.5$

$\therefore X_1 = 50$

#2.

7. Suppose that we wish to predict whether a given stock will issue a dividend this year ("Yes" or "No") based on X, last year's percent profit. We examine a large number of companies and discover that the mean value of X for companies that issued a dividend was $\bar{X} = 10$, while the mean for those that didn't was $\bar{X} = 0$. In addition, the variance of X for these two sets of companies was $\hat{\sigma}^2 = 36$. Finally, 80 % of companies issued dividends. Assuming that X follows a normal distribution, predict the probability that a company will issue a dividend this year given that its percentage profit was $X = 4$ last year.

Hint: Recall that the density function for a normal random variable is $f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2}$. *You will need to use Bayes' theorem.*

We substitute the given values for the Bayes' formula, i.e.,

$$P(\text{"Yes"} \mid X = 4)$$

$$= \frac{P(X=4 \mid \text{"Yes"}) \cdot P(\text{"Yes"})}{P(X=4)}.$$

So

$$P_1(4) = \frac{0.8\, e^{-(1/72)(4-10)^2}}{0.8\, e^{-(1/72)(4-10)^2} + 0.2\, e^{-(1/72)(4-0)^2}} = 0.752.$$

Therefore, the probability is 0.752.

ISLR Section4.8

HW_Q3. Exercise 14. a - g. •For b, use instead a logistic regression model and relevant methods/tests to select the best subset of regressors. •For c, use a random seed (for replication purposes) and use about 2/3 of data for training and 1/3 for testing. •h. Create a ROC curve for each method and plot them together for comparison. Explain your findings.

#14 a-g 14. In this problem, you will develop a model to predict whether a given car gets high or low gas mileage based on the Auto data set. (a) Create a binary variable, mpg01, that contains a 1 if mpg contains a value above its median, and a 0 if mpg contains a value below its median. You can compute the median using the median() function. Note you may find it helpful to use the data.frame() function to create a single data set containing both mpg01 and the other Auto variables.

```
#ggplot code can quickly become long if we constantly need to specify the characteris
tics of the theme we want to use. When we are making multiple plots and want them to
all have the same theme
#theme_set() to set the theme for all plots that are generated afterwards

library(ISLR2); library(tidyverse); library(ggthemes); library(GGally); library(knit
r); library(kableExtra); library(broom); library(dplyr)
```

```
## ── Attaching packages ─────────────────────────────────────────────────── ti
dyverse 1.3.2 ──
## ✔ ggplot2 3.3.6      ✔ purrr   0.3.4
## ✔ tibble  3.1.8      ✔ dplyr   1.0.10
## ✔ tidyr   1.2.0      ✔ stringr 1.4.1
## ✔ readr   2.1.2      ✔ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────────────────── tidyver
se_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
## Warning in !is.null(rmarkdown::metadata$output) && rmarkdown::metadata$output
## %in% : 'length(x) = 2 > 1' in coercion to 'logical(1)'
```

```
##
## Attaching package: 'kableExtra'
##
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
theme_set(theme_tufte(base_size = 15))
set.seed(1)
#Factors in R store categorical data.
data('Auto')
head(Auto)
```

```
##   mpg cylinders displacement horsepower weight acceleration year origin
## 1  18         8          307        130   3504         12.0   70      1
## 2  15         8          350        165   3693         11.5   70      1
## 3  18         8          318        150   3436         11.0   70      1
## 4  16         8          304        150   3433         12.0   70      1
## 5  17         8          302        140   3449         10.5   70      1
## 6  15         8          429        198   4341         10.0   70      1
##                        name
## 1 chevrolet chevelle malibu
## 2         buick skylark 320
## 3        plymouth satellite
## 4             amc rebel sst
## 5               ford torino
## 6          ford galaxie 500
```

```
At <- Auto %>%
    filter(!cylinders %in% c(3,5)) %>%
    mutate(mpg01 = factor(ifelse(median(mpg) < mpg, 1, 0)),
           cylinders = factor(cylinders,
                              levels = c(4,6,8),
                              ordered = TRUE),
           origin = factor(origin,
                           levels = c(1,2,3),
                           labels = c('American', 'European', 'Asian')))
#mutate function : create a variable, basically. The new variable needs a name and it
also needs a value that gets assigned to that name.
head(At)
```

```
##   mpg cylinders displacement horsepower weight acceleration year   origin
## 1  18         8          307        130   3504         12.0   70 American
## 2  15         8          350        165   3693         11.5   70 American
## 3  18         8          318        150   3436         11.0   70 American
## 4  16         8          304        150   3433         12.0   70 American
## 5  17         8          302        140   3449         10.5   70 American
## 6  15         8          429        198   4341         10.0   70 American
##                        name mpg01
## 1 chevrolet chevelle malibu     0
## 2         buick skylark 320     0
## 3        plymouth satellite     0
## 4             amc rebel sst     0
## 5               ford torino     0
## 6          ford galaxie 500     0
```

```
median(At$mpg)
```

```
## [1] 23
```

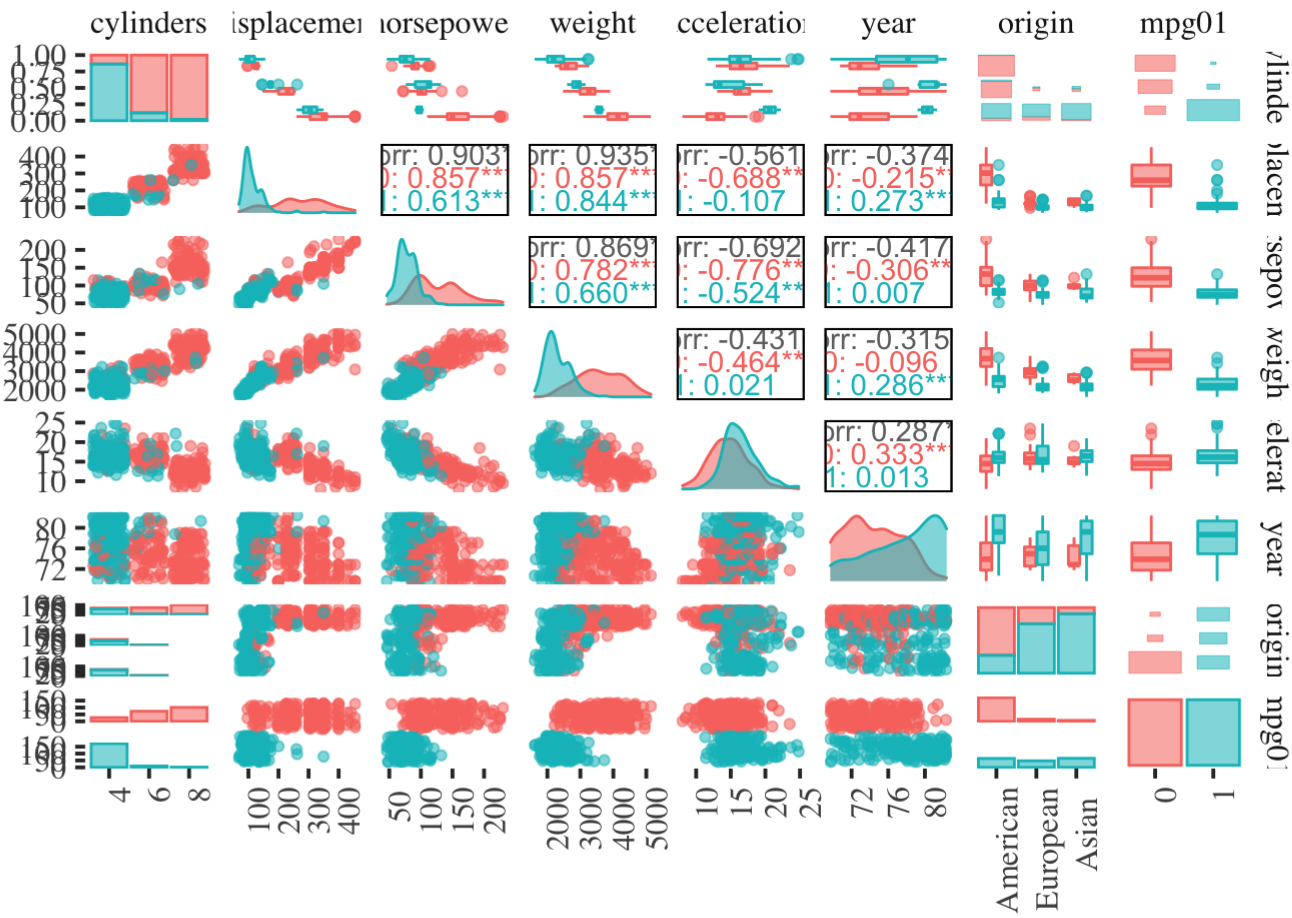
```
At %>%
    dplyr::select(mpg, mpg01) %>%
    sample_n(5)
```

```
##    mpg mpg01
## 1 29.8     1
## 2 23.0     0
## 3 25.0     1
## 4 28.4     1
## 5 17.0     0
```

```
#select() is a function from dplyr R package that is used to select data frame variab
les by name, by index, and also is used to rename variables while selecting, and drop
ping variables by name.
```

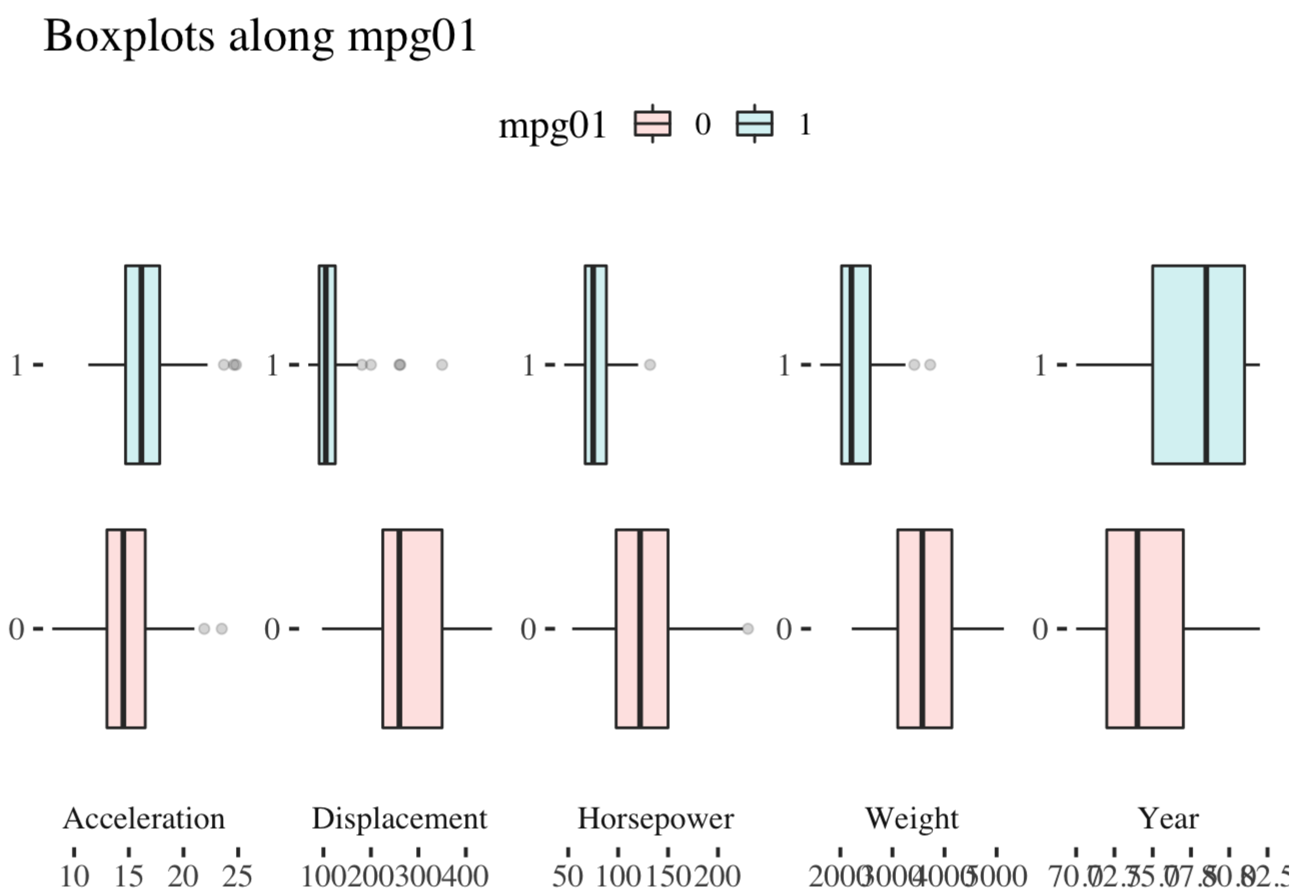
b. Explore the data graphically in order to investigate the association between mpg01 and the other features. Which of the other features seem most likely to be useful in predicting mpg01? Scatterplots and boxplots may be useful tools to answer this question. Describe your findings.

```
At %>%
    dplyr::select(-name, -mpg) %>%
ggpairs(aes(col = mpg01, fill = mpg01, alpha = 0.2),
        upper = list(combo = 'box'),
        diag = list(discrete = wrap('barDiag', position = 'fill')),
        lower = list(combo = 'dot_no_facet')) +
    theme(axis.text.x = element_text(angle = 90, hjust = 0.8))
```



```
At %>%
    dplyr::select(-name, -mpg, - origin, -cylinders) %>%
    gather(Variable, value, -mpg01) %>%
    mutate(Variable = str_to_title(Variable)) %>%
    ggplot(aes(mpg01, value, fill = mpg01)) +
    coord_flip() +
    theme(legend.position = 'top') +
    geom_boxplot(alpha = 0.2) +
    facet_wrap(~ Variable, scales = 'free', ncol = 5, switch = 'x') +
    labs(x = '', y = '', title = 'Boxplots along mpg01')
```

```
## Warning: 'switch' is deprecated.
## Use 'strip.position' instead.
## See help("Deprecated")
```



From the above observation by each color of mpg01, the values ‘cylinders’, ‘displacement’, ‘horsepower’, ‘weight’ and ‘year’ are well seperated via mpg01.

c. Split the data into a training set and a test set.

```
set.seed(3)
num_train <- nrow(At) * 0.75

inTr <- sample(nrow(At), size = num_train)

training <- At[inTr,]
head(training)
```

```
##      mpg cylinders displacement horsepower weight acceleration year   origin
## 261 17.5         8          318        140   4080         13.7   78 American
## 186 15.5         8          304        120   3962         13.9   76 American
## 140 26.0         4           97         78   2300         14.5   74 European
## 36  19.0         6          250         88   3302         15.5   71 American
## 384 28.0         4          120         79   2625         18.6   82 American
## 363 36.0         4          105         74   1980         15.3   82 European
##                     name mpg01
## 261     dodge magnum xe     0
## 186         amc matador     0
## 140          opel manta     1
## 36      ford torino 500     0
## 384         ford ranger     1
## 363 volkswagen rabbit l     1
```

```
testing <- At[-inTr,]
head(testing)
```

```
##    mpg cylinders displacement horsepower weight acceleration year   origin
## 11  15         8          383        170   3563         10.0   70 American
## 17  18         6          199         97   2774         15.5   70 American
## 18  21         6          200         85   2587         16.0   70 American
## 21  25         4          110         87   2672         17.5   70 European
## 25  21         6          199         90   2648         15.0   70 American
## 32  25         4          113         95   2228         14.0   71    Asian
##                    name mpg01
## 11 dodge challenger se     0
## 17          amc hornet     0
## 18       ford maverick     0
## 21         peugeot 504     1
## 25          amc gremlin     0
## 32       toyota corona     1
```

With 3 cylinders or 5 cylinders, we filter the cars.

d. Perform LDA on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
## The following object is masked from 'package:ISLR2':
##
##     Boston
```

```
fmla <- as.formula('mpg01 ~ displacement + horsepower + weight + year + cylinders')
lda <- lda(fmla, data = training)

pred <- predict(lda, testing)
table(pred$class, testing$mpg01)
```

```
##
##      0  1
##   0 40  2
##   1 10 45
```

```
1 - mean(pred$class == testing$mpg01)
```

```
## [1] 0.1237113
```

Prediction error is 0.124 and thus LDA model is well constructed.

e. Perform QDA on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

```
qda <- qda(fmla, data = training)

pred <- predict(qda, testing)
table(pred$class, testing$mpg01)
```

```
##
##      0  1
##   0 40  3
##   1 10 44
```

```
1 - mean(pred$class == testing$mpg01)
```

```
## [1] 0.1340206
```

Prediction error is 0.134 and thus QDA model is well constructed.

f. Perform logistic regression on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

```
log_reg <- glm(fmla, data = training, family = binomial)

pred <- predict(log_reg, testing, type = 'response')
pred_val <- round(pred)
table(pred_val, testing$mpg01)
```

```
##
## pred_val  0  1
##        0 43  2
##        1  7 45
```

```
mean(pred_val == testing$mpg01)
```

```
## [1] 0.9072165
```

Prediction error is 0.092 and thus Logistic Regression model is well constructed.

g. Perform naive Bayes on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

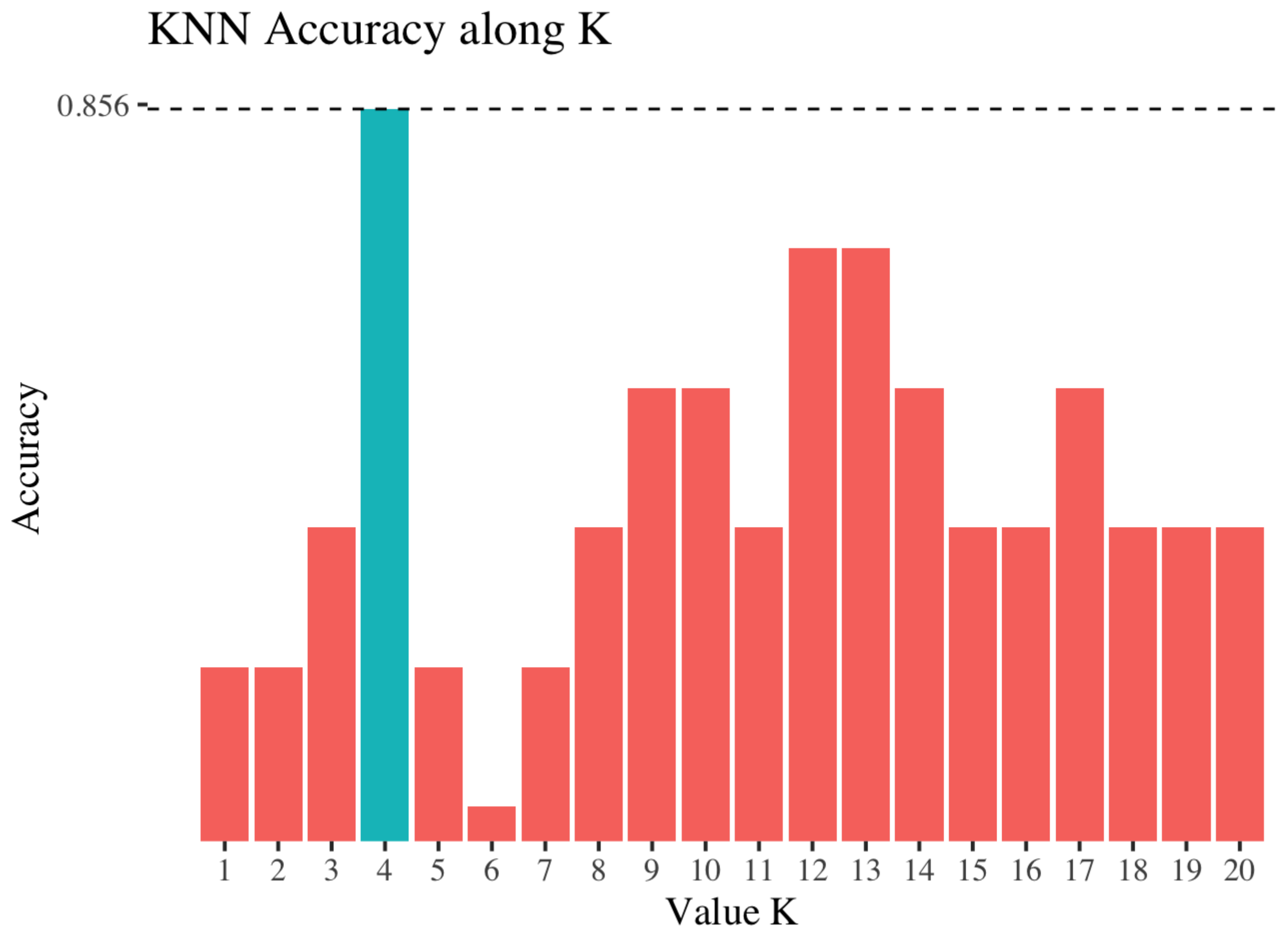
```
require(class)
```

```
## Loading required package: class
```

```
set.seed(3)
acc <- list()

x_tr <- training[,c('cylinders', 'displacement', 'horsepower', 'weight', 'year')]
y_tr <- training$mpg0
x_te <- testing[,c('cylinders', 'displacement', 'horsepower', 'weight', 'year')]

for (i in 1:20) {
    knn_pred <- knn(train = x_tr, test = x_te, cl = y_tr, k = i)
    acc[as.character(i)] = mean(knn_pred == testing$mpg01)
}
acc <- unlist(acc)

data_frame(acc = acc) %>%
    mutate(k = row_number()) %>%
    ggplot(aes(k, acc)) +
    geom_col(aes(fill = k == which.max(acc))) +
    labs(x = 'Value K', y = 'Accuracy', title = 'KNN Accuracy along K') +
    scale_x_continuous(breaks = 1:20) +
    scale_y_continuous(breaks = round(c(seq(0.90, 0.94, 0.01), max(acc)),
                                      digits = 3)) +
    geom_hline(yintercept = max(acc), lty = 2) +
    coord_cartesian(ylim = c(min(acc), max(acc))) +
    guides(fill = FALSE)
```

```
## Warning: `data_frame()` was deprecated in tibble 1.1.0.
## ℹ Please use `tibble()` instead.
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```



As a result, K=4 will be the most proper KNN models to use with accuracy of 0.9072 as in the previous problem.

HW_Q4. Exercise 16. •Do not use KNN model •Two or three subsets of predictors per model/method is good enough for comparison (obviously, try to select the most relevant subset) •Once you have determined which method (or methods) works best, create a sensitivity-specificity plot for this method(s) and explain your findings.

#16 16. Using the Boston data set, fit classification models in order to predict whether a given census tract has a crime rate above or below the median. Explore logistic regression, LDA, naive Bayes, and KNN models using various subsets of the predictors. Describe your findings. Hint: You will have to create the response variable yourself, using the variables that are contained in the Boston data set.

```
library(ggthemes)
library(knitr); library(kableExtra); library(MASS); library(tidyverse); library(corrp
lot)
```

```
## corrplot 0.92 loaded
```

```
library(broom)
set.seed(3)
theme_set(theme_tufte(base_size = 14))
data('Boston')
head(Boston)
```

```
##      crim zn indus chas   nox    rm  age    dis rad tax ptratio  black lstat
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90  4.98
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90  9.14
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83  4.03
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63  2.94
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90  5.33
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12  5.21
##   medv
## 1 24.0
## 2 21.6
## 3 34.7
## 4 33.4
## 5 36.2
## 6 28.7
```

CRIM - per capita crime rate by town

ZN - proportion of residential land zoned for lots over 25,000 sq.ft.

INDUS - proportion of non-retail business acres per town.

CHAS - Charles River dummy variable (1 if tract bounds river; 0 otherwise)

NOX - nitric oxides concentration (parts per 10 million)

RM - average number of rooms per dwelling

AGE - proportion of owner-occupied units built prior to 1940

DIS - weighted distances to five Boston employment centres

RAD - index of accessibility to radial highways

TAX - full-value property-tax rate per $10,000

PTRATIO - pupil-teacher ratio by town

BLACK - B - 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town

LSTAT - % lower status of the population

MEDV - Median value of owner-occupied homes in $1000's

```
#CRIM is the predictor and we make new dummy variable for classification.
#The variable chas is a factor.
Bst <- Boston %>%
    mutate(chas = factor(chas),
           crime_factor = factor(ifelse(median(crim) < crim,
                                        'High', 'Low'),
                                 levels = c('High', 'Low')))
head(Bst)
```

```
##      crim zn indus chas   nox    rm  age    dis rad tax ptratio  black lstat
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90  4.98
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90  9.14
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83  4.03
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63  2.94
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90  5.33
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12  5.21
##   medv crime_factor
## 1 24.0          Low
## 2 21.6          Low
## 3 34.7          Low
## 4 33.4          Low
## 5 36.2          Low
## 6 28.7          Low
```

##Numerical computation

```
scalethis <- function(x) {
    (x - mean(x, na.rm = TRUE)) / sd(x, na.rm = TRUE)
}

t1 <- Bst %>%
          dplyr::select(zn:crime_factor) %>%
          gather(Variable, value, -crime_factor, -chas) %>%
          group_by(Variable) %>%
          mutate(value = scalethis(value)) %>%
          group_by(Variable, crime_factor) %>%
          summarize(Q10 = quantile(value, .10),
                    Q25 = quantile(value, .25),
                    median = median(value),
                    mean = mean(value),
                    Q75 = quantile(value, .75),
                    Q90 = quantile(value, .90))
```

```
## `summarise()` has grouped output by 'Variable'. You can override using the
## `.groups` argument.
```

```
kable(t1,
      digits = 3, format = 'html') %>%
    kable_styling(bootstrap_options = c('striped', 'hover', 'condensed')) %>%
    column_spec(1:2, bold = T) %>%
    scroll_box(height = '300px')
```

Variable	crime_factor	Q10	Q25	median	mean	Q75	Q90
age	**High**	-0.234	0.470	0.846	0.613	1.042	1.116
age	**Low**	-1.782	-1.317	-0.713	-0.613	0.118	0.750
black	**High**	-2.942	-0.298	0.292	-0.351	0.421	0.441
black	**Low**	0.222	0.356	0.405	0.351	0.441	0.441
dis	**High**	-1.108	-0.970	-0.786	-0.616	-0.355	0.098

Variable	crime_factor	Q10	Q25	median	mean	Q75	Q90
dis	**Low**	-0.631	-0.202	0.628	0.616	1.275	1.915
indus	**High**	-0.720	-0.180	1.015	0.603	1.015	1.231

```r
difference <- function(x) x[1] - x[2]

t2 <- t1 %>%
    group_by(Variable) %>%
    summarize(diff_Q10 = difference(Q10),
              diff_Q25 = difference(Q25),
              diff_med = difference(median),
              diff_mean = difference(mean),
              diff_Q75 = difference(Q75),
              diff_Q90 = difference(Q90))



kable(t2,
      digits = 3, format = 'html') %>%
    kable_styling(bootstrap_options = c('striped', 'hover', 'condensed')) %>%
    column_spec(1, bold = T) %>%
    add_header_above(c(' ' = 1, 'Between-Groups Differences' = 6)) %>%
    scroll_box(height = '300px')
```

	Between-Groups Differences					
Variable	**diff_Q10**	**diff_Q25**	**diff_med**	**diff_mean**	**diff_Q75**	**diff_Q90**
age	1.548	1.787	1.560	1.227	0.924	0.366
black	-3.165	-0.654	-0.113	-0.702	-0.020	0.000
dis	-0.477	-0.768	-1.414	-1.231	-1.630	-1.817
indus	0.586	0.952	1.816	1.205	1.391	0.984
lstat	0.235	0.674	0.933	0.906	1.122	1.474
medv	-0.883	-0.728	-0.565	-0.526	-0.598	-0.350
nox	0.844	0.975	1.510	1.445	1.597	1.645
ptratio	-0.231	0.831	1.016	0.507	0.508	0.370

```r
t3 <- t2 %>%
    gather(Measure, value, -Variable) %>%
    group_by(Variable) %>%
    summarize(`Absolute Mean Differences` = abs(mean(value))) %>%
    arrange(desc(`Absolute Mean Differences`))
```

```
## Warning: attributes are not identical across measure variables;
## they will be dropped
```

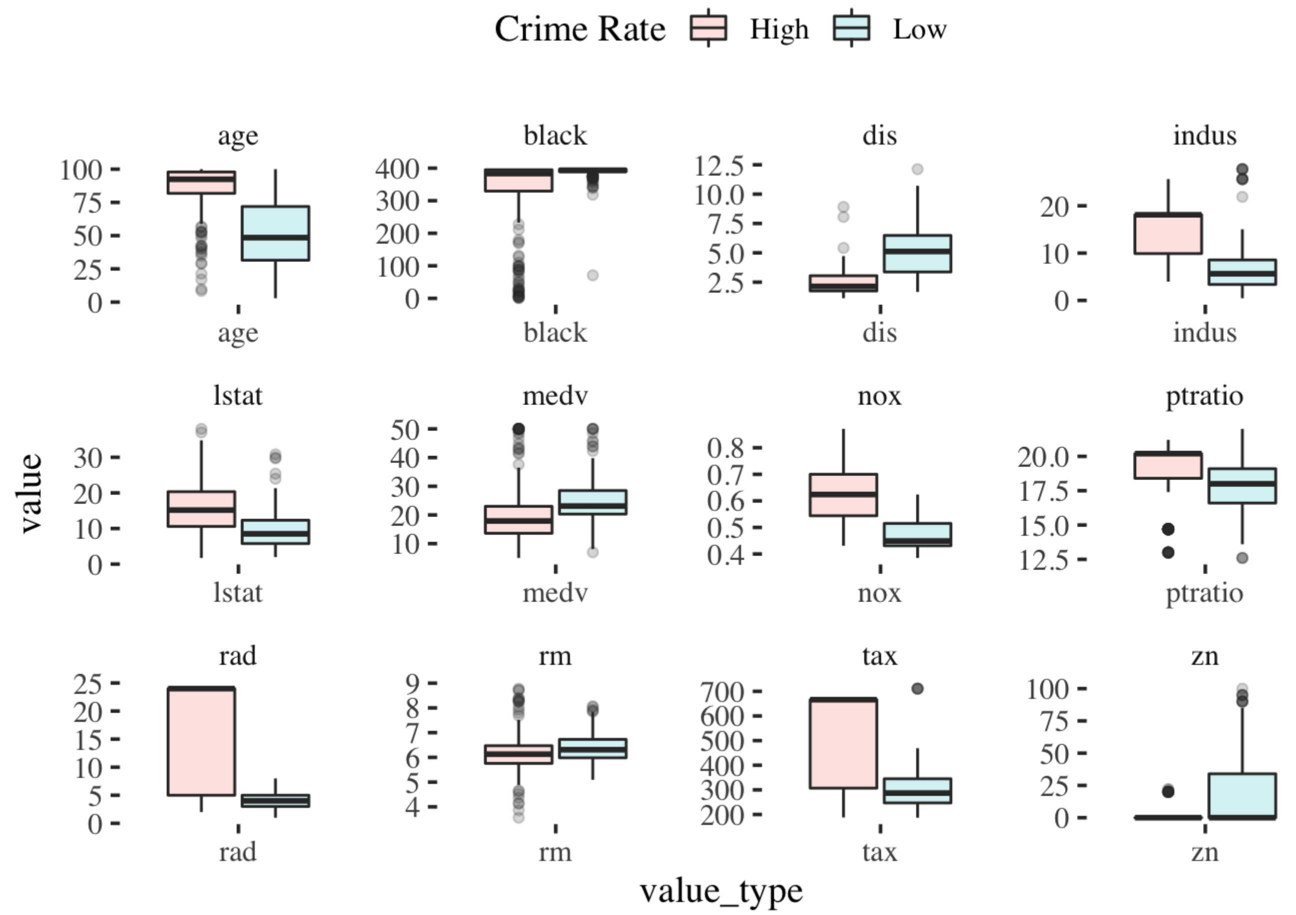
```
kable(t3,
      digits = 3, format = 'html') %>%
    kable_styling(bootstrap_options = c('striped', 'hover', 'condensed')) %>%
    column_spec(1, bold = T) %>%
    scroll_box(height = '300px')
```

Variable	Absolute Mean Differences
rad	1.374
nox	1.336
tax	1.300
age	1.235
dis	1.223
indus	1.156
zn	0.960
lstat	0.890
black	0.776

For the factor crim, normalize each variable to summarize for each group in the response variable. Also, from above, we can compare means from each group.

##Faceted Boxplots

```
Bst %>%
    dplyr::select(zn:crime_factor) %>%
    gather(value_type, value, -crime_factor, -chas) %>%
    ggplot(aes(value_type, value, fill = crime_factor)) +
    theme(legend.position = 'top') +
    geom_boxplot(alpha = 0.2) +
    facet_wrap(~value_type, scales = 'free') +
    scale_fill_discrete(name = 'Crime Rate')
```

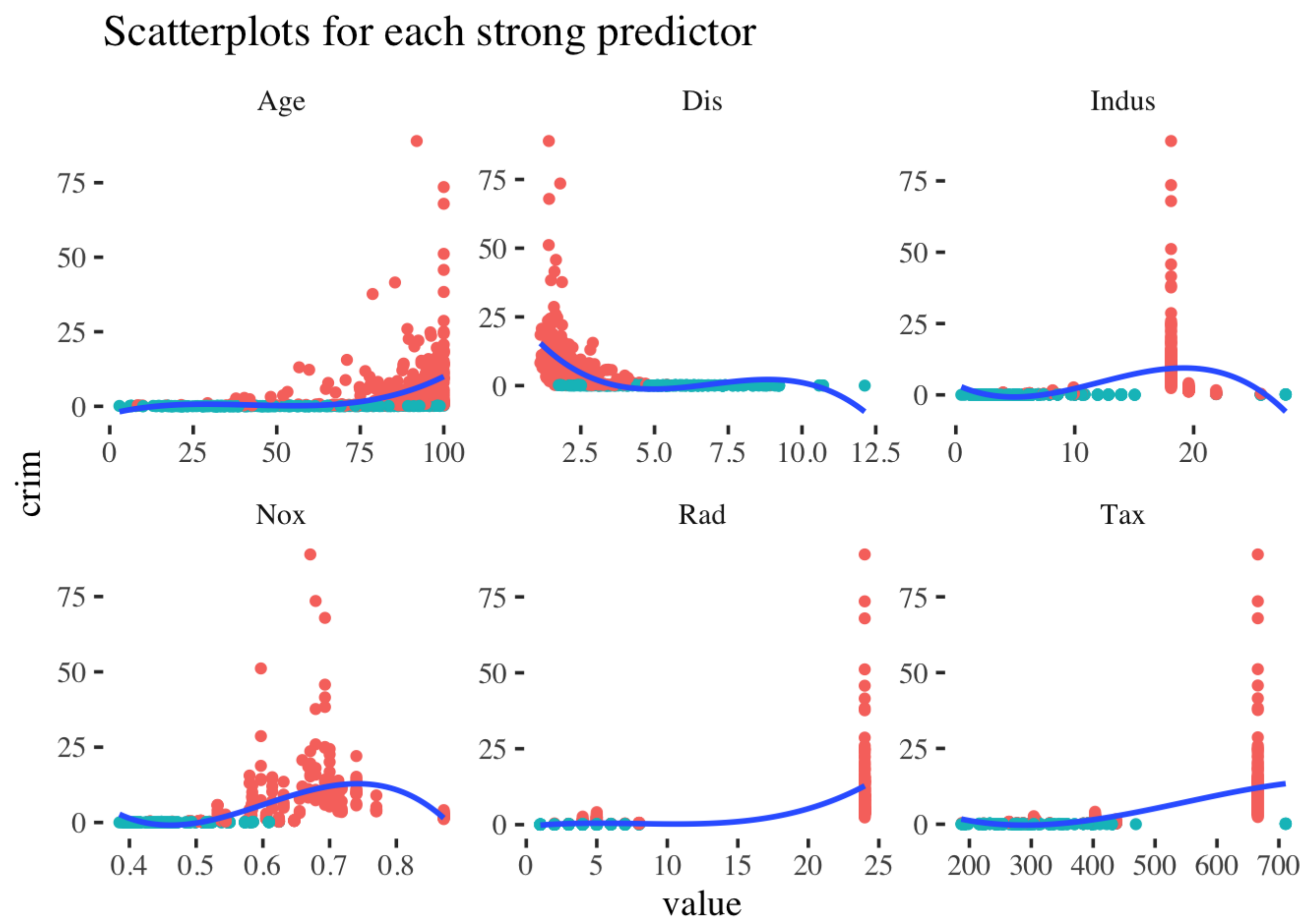


From the above boxplots, ‘rad’, ‘age’, ‘indus’, ‘nox’ and ‘tax’ are well separated by covariages. Therefore, these are strong variables.

So, let’s plot these variables versus crim, ased on the median.

```
Bst %>%
    dplyr::select(crim, crime_factor, rad, nox, tax, age, dis, indus) %>%
    gather(Variable, value, -crim, -crime_factor) %>%
    mutate(Variable = str_to_title(Variable)) %>%
    ggplot(aes(value, crim)) +
    guides(col = FALSE) +
    labs(title = 'Scatterplots for each strong predictor') +
    geom_point(aes(col = crime_factor)) +
    facet_wrap(~ Variable, scales = 'free') +
    geom_smooth(method = 'lm', formula = y ~ poly(x, 3), se = FALSE)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```



#Seperate

```
set.seed(3)
trainsize <- nrow(Bst) * 0.75
inTrain <- sample(1:nrow(Boston), size = trainsize)
training <- Bst[inTrain,]
testing <- Bst[-inTrain,]

log_reg <- glm(crime_factor ~ rad + nox + tax +
                   age + dis + indus, data = training, family = binomial)

log_reg %>%
    tidy %>%
    kable(digits = 3)
```

term	estimate	std.error	statistic	p.value
(Intercept)	21.596	4.141	5.215	0.000
rad	-0.628	0.134	-4.685	0.000
nox	-38.345	7.767	-4.937	0.000
tax	0.008	0.003	2.960	0.003
age	-0.006	0.010	-0.597	0.551
dis	-0.052	0.166	-0.316	0.752
indus	0.030	0.050	0.587	0.557

For the covariages, there are measurement of the significance.

```
pred <- predict(log_reg, testing, type = 'response')
pred_value2 <- ifelse(pred >= 0.5, 'Low', 'High')
acc <- mean(pred_value2 == testing$crime_factor)

table(pred_value2, testing$crime_factor) %>%
    kable(format = 'html') %>%
    kable_styling() %>%
    add_header_above(c('Predicted' = 1, 'Observed' = 2)) %>%
    column_spec(1, bold = T) %>%
    add_footnote(label = acc)
```

Predicted	Observed	
	High	Low
High	51	3
Low	13	60

[a] 0.874015748031496

##LDA, QDA models

```
ldamodel2 <- lda(crime_factor ~ poly(rad, 3) + poly(nox, 3) +
                   poly(tax, 3) + poly(age, 3) + poly(dis, 3), data = training)
pred <- predict(ldamodel2, testing)
acc <- mean(pred$class == testing$crime_factor)

table(pred$class, testing$crime_factor) %>%
    kable(format = 'html') %>%
    kable_styling() %>%
    add_header_above(c('Predicted' = 1, 'Observed' = 2)) %>%
    column_spec(1, bold = T) %>%
    add_footnote(label = acc)
```

Predicted	Observed	
	High	Low
High	62	2
Low	2	61

[a] 0.968503937007874

```
ldamodel2 <- lda(crime_factor ~ rad + nox + tax + age + dis, data = training)
pred <- predict(ldamodel2, testing)
acc <- mean(pred$class == testing$crime_factor)

table(pred$class, testing$crime_factor) %>%
    kable(format = 'html') %>%
    kable_styling() %>%
    add_header_above(c('Predicted' = 1, 'Observed' = 2)) %>%
    column_spec(1, bold = T) %>%
    add_footnote(label = acc)
```

Predicted	**Observed**	
	High	**Low**
High	46	2
Low	18	61

[a] 0.84251968503937

```
qda_model <- qda(crime_factor ~ poly(rad, 3) + poly(nox, 3) +
                   poly(tax, 3) + poly(age, 3) + poly(dis, 3), data = training)
pred <- predict(qda_model, testing)
acc <- mean(pred$class == testing$crime_factor)

table(pred$class, testing$crime_factor) %>%
    kable(format = 'html') %>%
    kable_styling() %>%
    add_header_above(c('Predicted' = 1, 'Observed' = 2)) %>%
    column_spec(1, bold = T) %>%
    add_footnote(label = acc)
```

Predicted	**Observed**	
	High	**Low**
High	59	0
Low	5	63

[a] 0.960629921259842

```
qda_model <- qda(crime_factor ~ rad + nox + tax + age + dis, data = training)
pred <- predict(qda_model, testing)
acc <- mean(pred$class == testing$crime_factor)

table(pred$class, testing$crime_factor) %>%
    kable(format = 'html') %>%
    kable_styling() %>%
    add_header_above(c('Predicted' = 1, 'Observed' = 2)) %>%
    column_spec(1, bold = T) %>%
    add_footnote(label = acc)
```

Predicted	Observed	
	High	**Low**
High	48	3
Low	16	60

[a] 0.850393700787402

```
qda_model <- qda(crime_factor ~ poly(rad, 3) + poly(nox, 3) +
                   poly(tax, 3) + poly(age, 3) + poly(dis, 3) + poly(indus, 3),
                 data = training)
pred <- predict(qda_model, testing)
acc <- mean(pred$class == testing$crime_factor)

table(pred$class, testing$crime_factor) %>%
    kable(format = 'html') %>%
    kable_styling() %>%
    add_header_above(c('Predicted' = 1, 'Observed' = 2)) %>%
    column_spec(1, bold = T) %>%
    add_footnote(label = acc)
```

Predicted	Observed	
	High	**Low**
High	61	0
Low	3	63

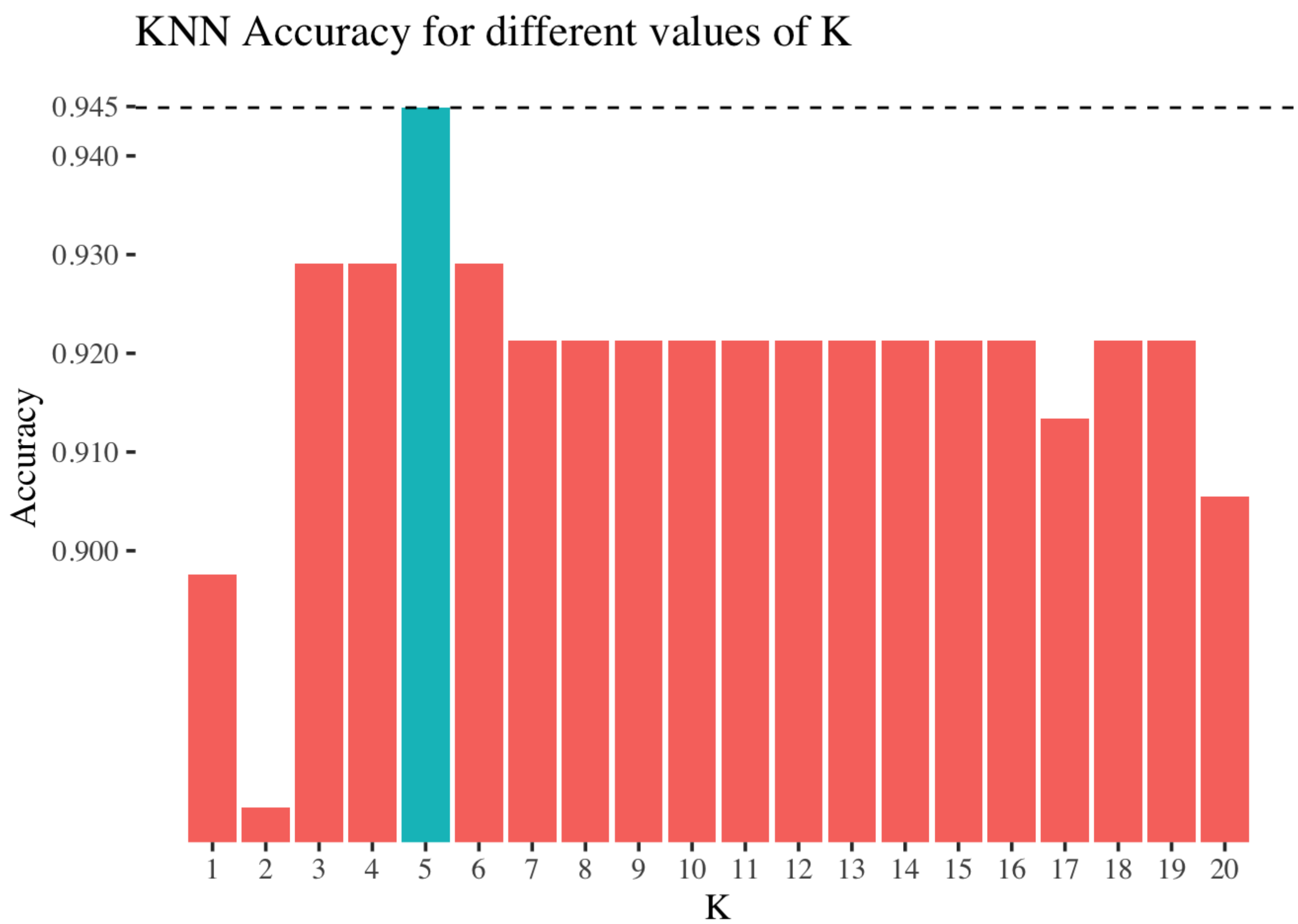
[a] 0.976377952755906

The last PDA model is the beset model amont whole models, based on the error 0.024 from above.

##KNN

```r
require(class)
variables <- c('rad', 'nox', 'tax', 'age', 'dis', 'zn', 'indus')

xtrain <- training[, variables]
ytrain <- training$crime_factor
xtest <- testing[, variables]
acc <- list()

for (i in 1:20) {
    knn_pred <- knn(train = xtrain, test = xtest, cl = ytrain, k = i)
    acc[as.character(i)] = mean(knn_pred == testing$crime_factor)
}

acc <- unlist(acc)

data_frame(acc = acc) %>%
    mutate(k = row_number()) %>%
    ggplot(aes(k, acc)) +
    geom_col(aes(fill = k == which.max(acc))) +
    labs(x = 'K', y = 'Accuracy', title = 'KNN Accuracy for different values of K') +
    scale_x_continuous(breaks = 1:20) +
    scale_y_continuous(breaks = round(c(seq(0.90, 0.94, 0.01), max(acc)),
                                      digits = 3)) +
    geom_hline(yintercept = max(acc), lty = 2) +
    coord_cartesian(ylim = c(min(acc), max(acc))) +
    guides(fill = FALSE)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```



In

the above, KNN has accuracy of .945 and K = 5. At this setting and dataset, logistic regression model with third order polynomial as in the above fitting.