4123212_assignment_prediction

R Markdown

The dataset is a simple housing price dataset. The information it contains is restricted to houses in california. The columns it has are focused around its area, features such as population, households, location (in the form of coordinates). Furthermore, the dataset contains information about the house itself. Features such as amount of bedrooms, amount of rooms, median income of the household, house value. Furthermore, the feature in question, is the feature that describes the proximity of the house to a body of water. Either in the bay area, near the ocean or sea, or inland.

This assignment will apply classification techniques on the dataset. Three types will be attempted, for which the best will be chosen. k-nearest neighbors will be used, logistic regression and linear discriminant analysis as well. In order to do this, the column that captures the proximity to water has been transformed to a boolean type where all the classes that describe a house being in close proximity to the water as TRUE and the others as FALSE. This classification is then to be predicted.

```
rm(list = ls())
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 3.6.2
```

```
library(class)
library(ISLR)
library(tidyverse)
```

```
## Registered S3 methods overwritten by 'ggplot2':
##   method         from
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang

## Registered S3 method overwritten by 'rvest':
##   method            from
##   read_xml.response xml2

## -- Attaching packages ---------------------------------------------------------------------------------

## v ggplot2 3.1.1       v purrr   0.3.2
## v tibble  2.1.1       v dplyr   0.8.0.1
## v tidyr   0.8.3       v stringr 1.4.0
## v readr   1.3.1       v forcats 0.4.0

## -- Conflicts ----------------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()
```

```r
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 3.6.2
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

Load the dataset and show a short exerpt of it

```r
data <- read_csv("Data\\housing.csv")
```

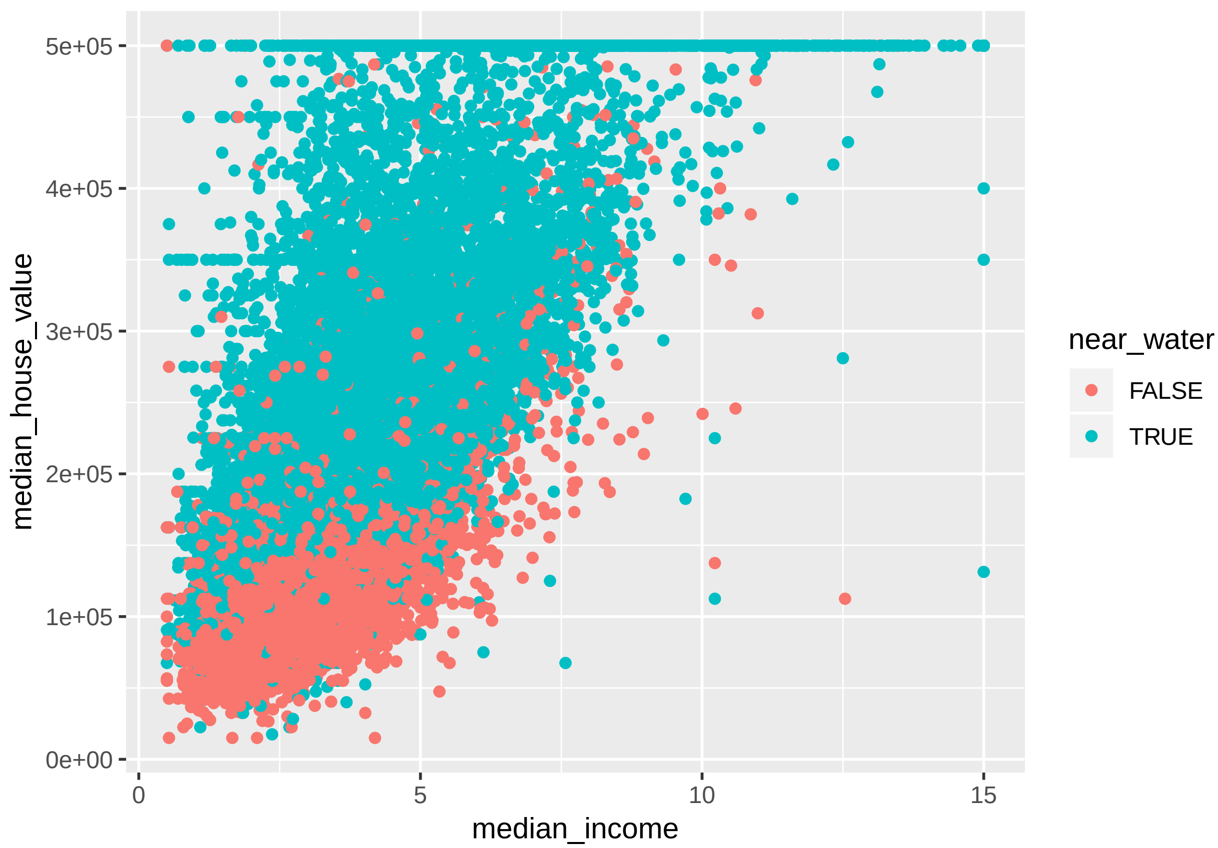
```
## Parsed with column specification:
## cols(
##   longitude = col_double(),
##   latitude = col_double(),
##   housing_median_age = col_double(),
##   total_rooms = col_double(),
##   total_bedrooms = col_double(),
##   population = col_double(),
##   households = col_double(),
##   median_income = col_double(),
##   median_house_value = col_double(),
##   ocean_proximity = col_character()
## )
```

```r
head(data)
```

```
## # A tibble: 6 x 10
##   longitude latitude housing_median_~ total_rooms total_bedrooms population
##       <dbl>    <dbl>            <dbl>       <dbl>          <dbl>      <dbl>
## 1     -122.     37.9               41         880            129        322
## 2     -122.     37.9               21        7099           1106       2401
## 3     -122.     37.8               52        1467            190        496
## 4     -122.     37.8               52        1274            235        558
## 5     -122.     37.8               52        1627            280        565
## 6     -122.     37.8               52         919            213        413
## # ... with 4 more variables: households <dbl>, median_income <dbl>,
## #   median_house_value <dbl>, ocean_proximity <chr>
```

Transform the proximity to water to a boolean type, omit rows with missing values and plot the data using two variables: median income and median house value

```r
data <- data %>% mutate(near_water = ifelse(ocean_proximity =="INLAND", FALSE, TRUE))
data <- na.omit(data)
data %>% ggplot(aes(x = median_income, y = median_house_value, color = near_water)) + geom_point()
```



Create an accuracy function for easy accuracy calculation

```r
accuracy <- function(matrix)
  round((matrix[1,1]+matrix[2,2])/sum(matrix)*100, 2)
```

Split the data into train and test sets.

```r
amount_of_train <- round(0.8*length(data$longitude),0)

data <- data[,-10] %>%
  mutate(split = sample(rep(c("train", "test"), times = c(amount_of_train,
                              length(data$latitude)-amount_of_train))))

data_train <-
  data %>%
  filter(split == "train") %>%
  select(-split)

data_test <-
  data %>%
  filter(split == "test") %>%
  select(-split)
```

First classification method: K-nearest neighbors. this will be tried for 3-nearest, 5-nearest, 9-nearest

```
knn_3 <- knn(
  train = data_train[,-10],
  test  = data_test[,-10],
  cl    = as.factor(data_train$near_water),
  k     = 3
)

knn_5 <- knn(
  train = data_train[,-10],
  test  = data_test[,-10],
  cl    = as.factor(data_train$near_water),
  k     = 5
)

knn_9 <- knn(
  train = data_train[,-10],
  test  = data_test[,-10],
  cl    = as.factor(data_train$near_water),
  k     = 9
)
```

Create confusion matrices for each model and print the accuracies for each

```
confusion_matrix_3 <- table(true = data_test$near_water, predicted = knn_3)
confusion_matrix_5 <- table(true = data_test$near_water, predicted = knn_5)
confusion_matrix_9 <- table(true = data_test$near_water, predicted = knn_9)

paste("Accuracy is 3-nearest neighbor: ", accuracy(confusion_matrix_3), "%")
```

```
## [1] "Accuracy is 3-nearest neighbor:  82.77 %"
```

```
paste("Accuracy is 5-nearest neighbor: ", accuracy(confusion_matrix_5), "%")
```

```
## [1] "Accuracy is 5-nearest neighbor:  83.41 %"
```

```
paste("Accuracy is 9-nearest neighbor: ", accuracy(confusion_matrix_9), "%")
```

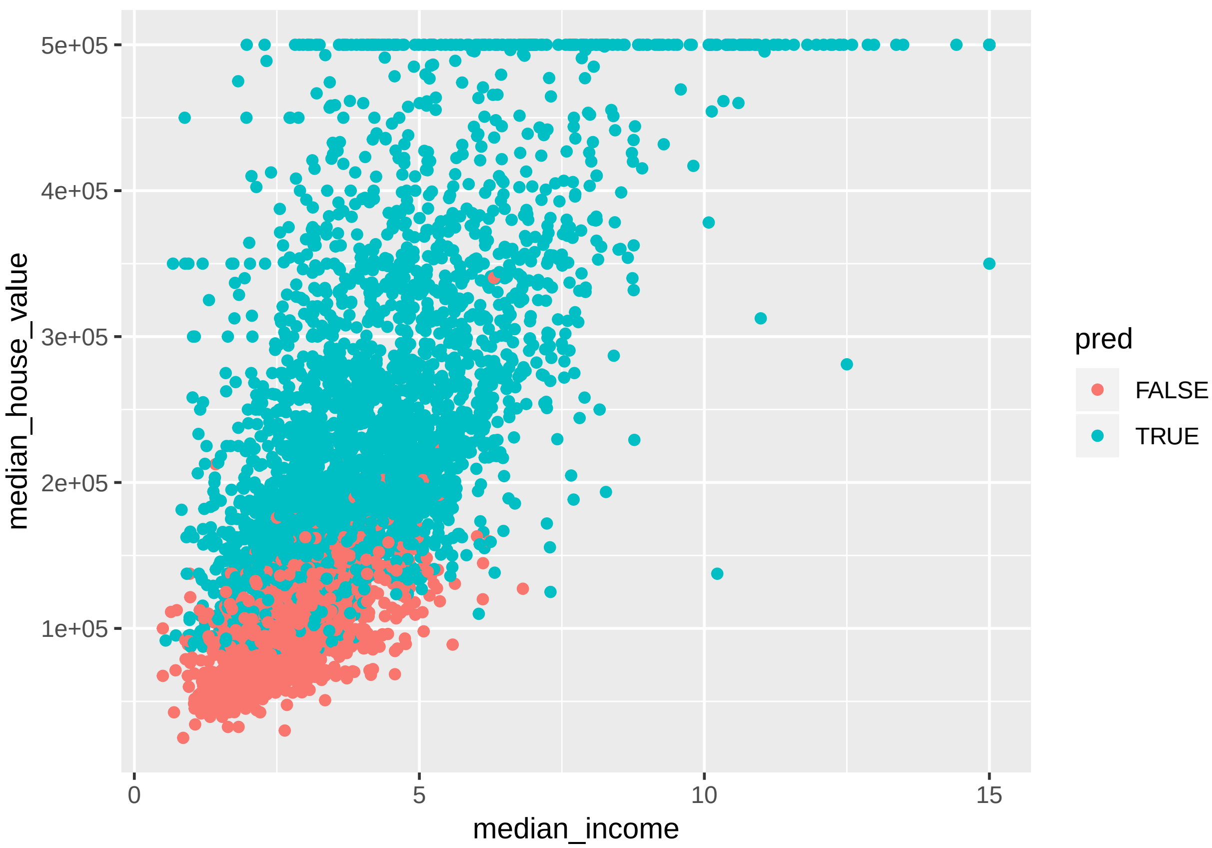
```
## [1] "Accuracy is 9-nearest neighbor:  83.9 %"
```

```
best_k_nearest <- accuracy(confusion_matrix_9)
```

It seems that the 9-nearest neighbor is the most accurate of the three models.

A plot of this model:

```
add_column(data_test, pred = knn_9) %>% ggplot(aes(x = median_income,
                                                   y = median_house_value,
                                                   colour = pred)) +
                                          geom_point()
```



Second method: Logistic Regression

```
logistic_regression <- glm(near_water ~ ., family = binomial, data = data_train)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
pred <- tibble(value = round(predict(logistic_regression, type = "response",
                                     newdata = data_test), 2))
```

Create an ROC plot in order to evaluate the model

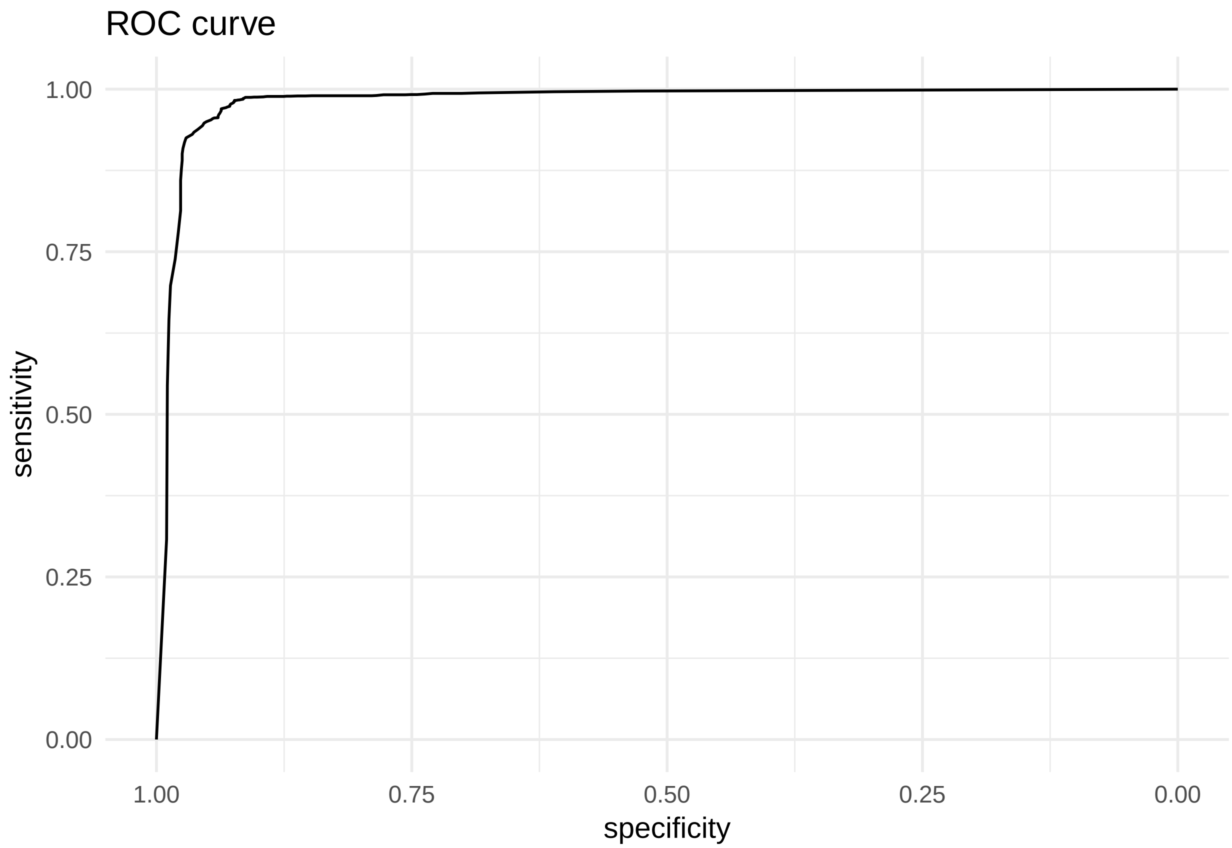
```
# Get the correct data
temporary_set <- data_test %>% mutate(near_water = ifelse(near_water == TRUE, 1, 0))

roc_data <- roc(temporary_set$near_water, pred$value)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
ggroc(roc_data) + theme_minimal() + labs(title = "ROC curve")
```



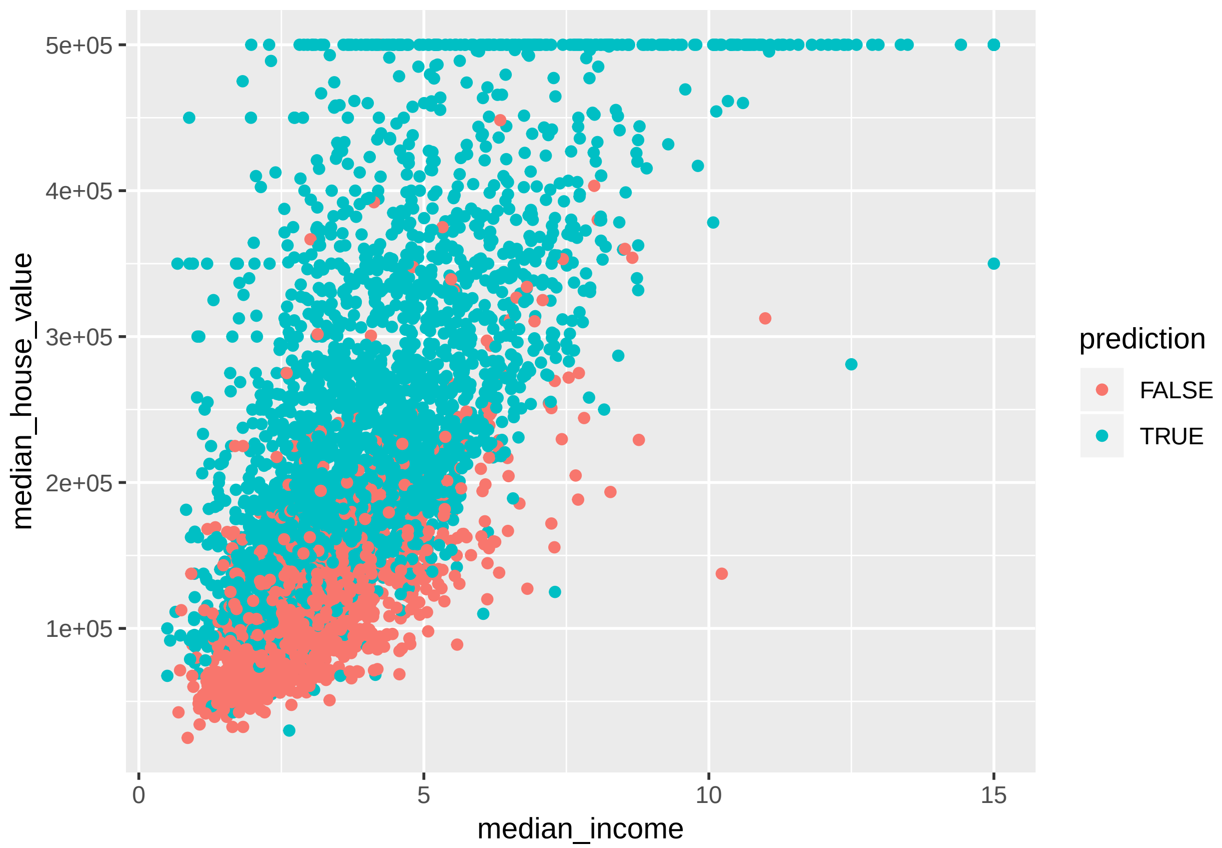

```
roc_data
```

```
##
## Call:
## roc.default(response = temporary_set$near_water, predictor = pred$value)
##
## Data: pred$value in 1309 controls (temporary_set$near_water 0) < 2778 cases (temporary_set$near_wate
## Area under the curve: 0.9811
```

The AUC of 0.98 and the ROC curve show us that the model performs quite well!

Transform the predictions to boolean and plot the values

```
pred <- pred %>% mutate(value = ifelse(value >= 0.5, TRUE, FALSE))
add_column(data_test, prediction = pred$value) %>% ggplot(aes(x = median_income,
                                                              y = median_house_value,
                                                              colour = prediction)) +
                                                    geom_point()
```



Plot the confusion matrix for this model

```
confusion_matrix <- table(true = data_test$near_water, predicted = pred$value )
confusion_matrix
```

```
##        predicted
## true     FALSE TRUE
##   FALSE   1200  109
##   TRUE      44 2734
```

Calculate the accuracy

```
best_log <- accuracy(confusion_matrix)
paste("Accuracy of logistic regression is: ", best_log, "%")
```

```
## [1] "Accuracy of logistic regression is:  96.26 %"
```

This seems much better than before. Let's try another method anyway!

Third method: Linear Discriminant Analysis

```
linear_discriminant <- lda(near_water ~ ., data = data_train)
linear_discriminant
```

```
## Call:
## lda(near_water ~ ., data = data_train)
##
## Prior probabilities of groups:
##     FALSE      TRUE
## 0.3173253 0.6826747
##
## Group means:
##       longitude latitude housing_median_age total_rooms total_bedrooms
## FALSE -119.7312 36.72831           24.29728    2713.380       532.6175
## TRUE  -119.5109 35.13680           30.71494    2596.143       540.3902
##       population households median_income median_house_value
## FALSE   1388.492   476.6665      3.205803           124787.8
## TRUE    1438.998   509.8572      4.174574           244841.2
##
## Coefficients of linear discriminants:
##                              LD1
## longitude          -1.691166e+00
## latitude           -1.780561e+00
## housing_median_age  9.213577e-03
## total_rooms        -2.286435e-04
## total_bedrooms      1.190170e-03
## population          7.600568e-05
## households         -2.620951e-04
## median_income       1.959869e-02
## median_house_value  2.793361e-06
```

Show the confusion matrix

```
pred <- predict(linear_discriminant, newdata = data_test)
confusion_matrix_LDA<-table(true = data_test$near_water, predicted= pred$class)
confusion_matrix
```

```
##        predicted
## true    FALSE TRUE
##   FALSE  1200  109
##   TRUE     44 2734
```

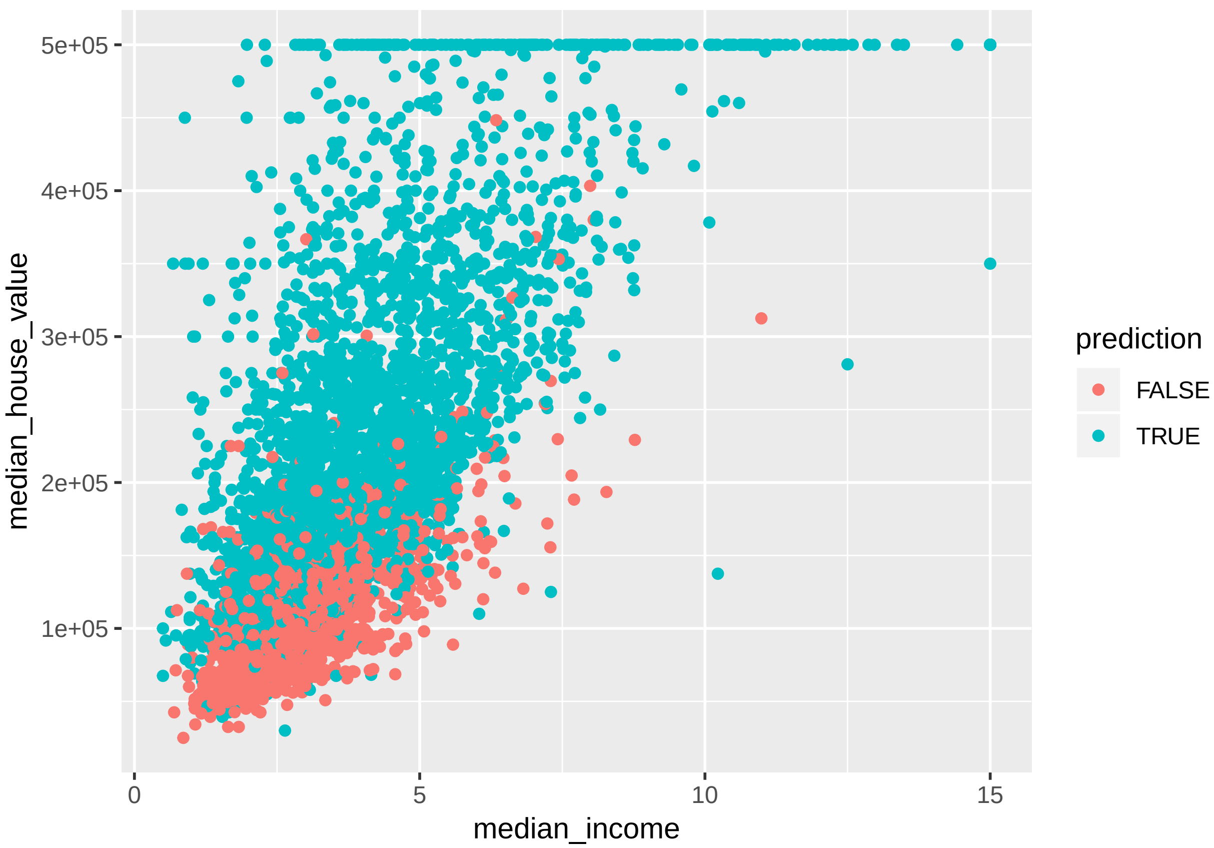
Calculate the accuracy

```
best_lda <- accuracy(confusion_matrix_LDA)
paste("Accuracy of linear discriminant analysis is: ", best_lda, "%")
```

```
## [1] "Accuracy of linear discriminant analysis is:  93.52 %"
```

Plot the predicted plot

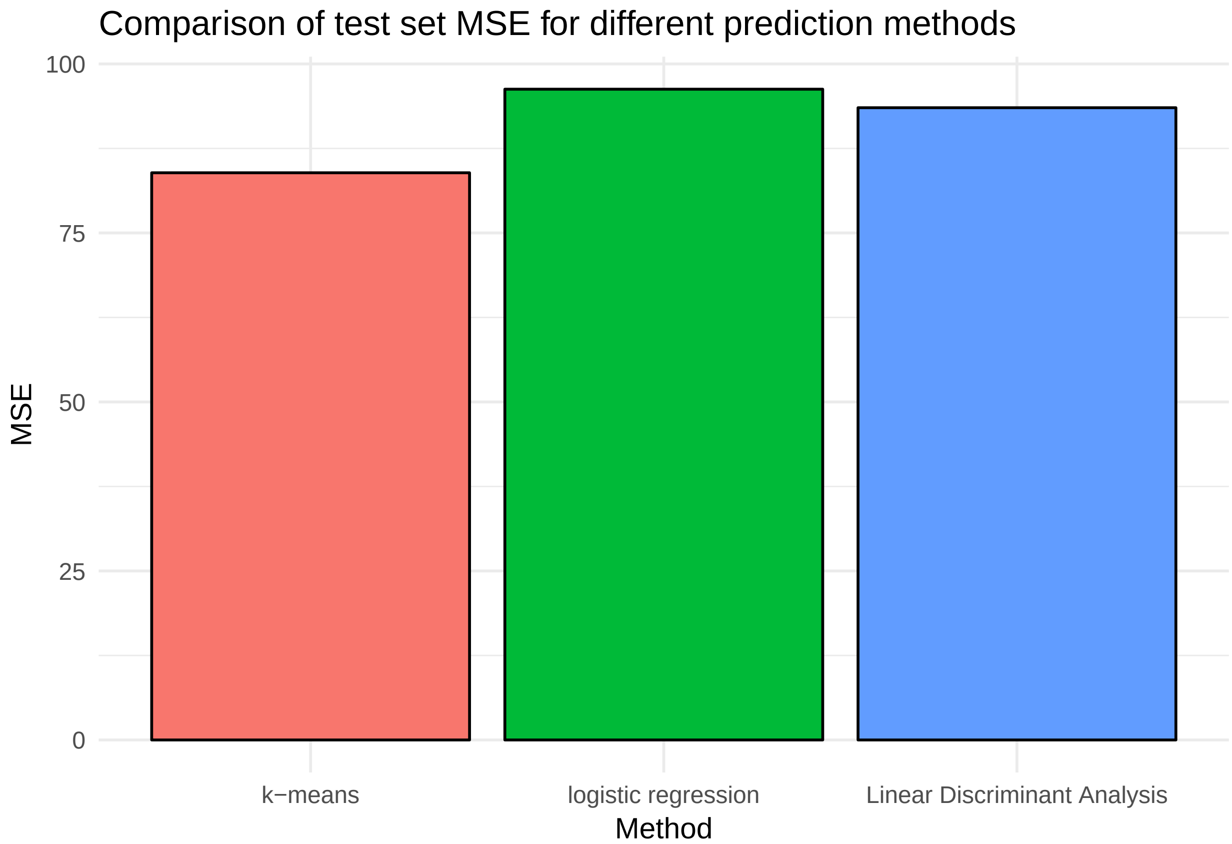
```
add_column(data_test, prediction = pred$class) %>% ggplot(aes(x = median_income,
                                                              y = median_house_value,
                                                              colour = prediction))+
                                                  geom_point()
```



Let's compare the best of all three models to see which one would be the best to use:

```
model_scores <- c(best_k_nearest, best_log, best_lda)

tibble(Method = as_factor(c("k-means", "logistic regression", "Linear Discriminant Analysis")), MSE = mo
  ggplot(aes(x = Method, y = MSE, fill = Method)) +
  geom_bar(stat = "identity", col = "black") +
  theme_minimal() +
  theme(legend.position = "none") +
  labs(title = "Comparison of test set MSE for different prediction methods")
```



It seems that the logistic regression model performed the best out of all three of the methods.