assignement2

Alban Tchikladze

13/12/2021

1) Description

Your assignment requires you to perform hypothesis testing, regression and classification tasks on the given data set

```
#library class to access the knn library
library(class)

#library C50 to access the C5.0 algorithm
library(C50)

#library neuralnet to access the NN algorithm
library(neuralnet)
```

2) Dataset

First download the dataset from Moodle's module page.

```
# read the csv file
housing.dataset <- read.csv("melbourne_housing_data.csv")

#summary of the data
summary(housing.dataset)
```

```
##        X              Suburb             Address              Rooms
##  Min.   :    1   Length:48433       Length:48433       Min.   : 1.000
##  1st Qu.:15797   Class :character   Class :character   1st Qu.: 2.000
##  Median :31587   Mode  :character   Mode  :character   Median : 3.000
##  Mean   :31562                                         Mean   : 3.072
##  3rd Qu.:47365                                         3rd Qu.: 4.000
##  Max.   :63021                                         Max.   :31.000
##      Type               Price              Method             SellerG
##  Length:48433       Min.   :   85000   Length:48433       Length:48433
##  Class :character   1st Qu.:  620000   Class :character   Class :character
##  Mode  :character   Median :  830000   Mode  :character   Mode  :character
##                     Mean   :  997898
##                     3rd Qu.: 1220000
##                     Max.   :11200000
##      Date              Postcode     Regionname         Propertycount
##  Length:48433       Min.   :3000   Length:48433       Min.   :   39
##  Class :character   1st Qu.:3051   Class :character   1st Qu.: 4280
##  Mode  :character   Median :3103   Mode  :character   Median : 6567
##                     Mean   :3123                      Mean   : 7566
##                     3rd Qu.:3163                      3rd Qu.:10412
##                     Max.   :3980                      Max.   :21650
##     Distance     CouncilArea
```

```
##   Min.   : 0.0   Length:48433
##   1st Qu.: 7.0   Class :character
##   Median :11.7   Mode  :character
##   Mean   :12.7
##   3rd Qu.:16.7
##   Max.   :55.8
```

Preparing the dataset to perform analysis

```
#removing columns that are useless for the kind of analysis that we will do
housing.dataset <- subset(housing.dataset, select = -c(X, Address, Date, Regionnam
e, CouncilArea))

#converting all the character columns into numeric ones
housing.dataset$Suburb <- as.numeric(as.factor(housing.dataset$Suburb))
housing.dataset$Type <- as.numeric(as.factor(housing.dataset$Type))
housing.dataset$Method <- as.numeric(as.factor(housing.dataset$Method))
housing.dataset$SellerG <- as.numeric(as.factor(housing.dataset$SellerG))
housing.dataset$Postcode <- as.numeric(as.factor(housing.dataset$Postcode))
```

3) Tasks

3.1. Task A

you have to define hypotheses, which will be tested. You should state at least 2 different hypotheses, each to test different data

Hypothesis one :

```
#1) Start with a well-developed question

#Let's imagine that someone which is running a real estate agency come to ask us f
or our help
#He have sold 500 flats and houses in the past year.
#And he want to now if he have sold them under the average price of the market



#2) Establish hypotheses, both null and alternative

#m will be the mean of the dataset price and m1 will be the mean of our sample
#the hypothesis is H1: m > m1 and we also have H0: m <= m1



#3) Determine appropriate statistical test and sampling distribution

#we look for a seller with approximately 500 sales
#countTable <- lapply(housing.dataset, table)
#countTable$SellerG
#in this case the seller 194 will do perfectly

#we select the price of the sales form this seller
samplePrice<-with(housing.dataset, Price[SellerG == 194])
```

```r
#we check his mean
mean(samplePrice)
```

```
## [1] 877406.3
```

```r
#4) Choose the significance level (α)

#we choose α = 0.05
Z <- qnorm(1-0.05/2)


#5) State the decision rule

#if m1 is in the trust interval of m, then we will say that the test is valid, oth
erwise it will not be valid


#6) Calculate test statistic

lowerBound <- mean(samplePrice) - Z*(sd(samplePrice)/sqrt(length(samplePrice)))
upperBound <- mean(samplePrice) + Z*(sd(samplePrice)/sqrt(length(samplePrice)))

#we print the interval
print(c(lowerBound,upperBound))
```

```
## [1] 849867.6 904944.9
```

```r
#we print the mean of all the price
mean(housing.dataset$Price)
```

```
## [1] 997898.2
```

```r
#7) State statistical conclusion

#The mean is not the 95% interval so the price for this seller are lower than the
 average market price


#8) Make the inference based on conclusion

#it means that, either he have sold his houses under market price, as he ask us to
compute for him.
#Or that he simply do not have access to the best houses of the market, an that he
have a good progression margin to get to the level of other sellers.

#to really answer his question about his sales and the average market price, we sh
ould have taken his houses parameters into account and select similar
#houses from the data set to compare with them, it would have been more accurate,
 and he would have known if he was selling them under the price
```

Hypothesis two :

```r
#1) Start with a well-developed question

#Another real estate agency holder has come to ask us if there was a significant d
ifference between houses and townhouses.
#He told us that it was more advantageous to put some townhouses as houses because
the average price for houses his higher than for townhouses

mean(with(housing.dataset, Price[Type == 1]))
```

```
## [1] 1110587
```

```r
mean(with(housing.dataset, Price[Type == 2]))
```

```
## [1] 911148
```

```r
#And as we could see there his almost a 15% differences between those means

#we find out that maybe the distances could be the deciding factor to choose betwe
en twonhouses and houses
#So lets have a look at houses distances and see if we could add some townhouses i
nto the houses


#2) Establish hypotheses, both null and alternative

#m1 will be the mean of townhouses distances and m2 will be the mean of houses dis
tances
#the hypothesis is H1: m1 = m2 and we also have H0: m1 != m2


#3) Determine appropriate statistical test and sampling distribution

#we select the distance for this houses
sampleDistance<-with(housing.dataset, Distance[Type == 1])

#we sample a 1000 houses
sample_index <- sort(sample(length(sampleDistance), 1000))
sampleDistance <- sampleDistance[sample_index]

#we check the mean of Distance for our sample
mean(sampleDistance)
```

```
## [1] 14.0061
```

```r
#4) Choose the significance level (α)

#we choose α = 0.05
Z <- qnorm(1-0.05/2)


#5) State the decision rule
```

```
#if m2 is in the trust interval of m1, then we will say that the test is valid, ot
herwise it will not be valid


#6) Calculate test statistic

lowerBound <- mean(sampleDistance) - Z*(sd(sampleDistance)/sqrt(length(sampleDista
nce)))
upperBound <- mean(sampleDistance) + Z*(sd(sampleDistance)/sqrt(length(sampleDista
nce)))

#we print the interval
print(c(lowerBound,upperBound))
```

```
## [1] 13.50149 14.51071
```

```
#we print the mean of townhouses distances
distanceTownHouses <- with(housing.dataset, Distance[Type == 2])
mean(distanceTownHouses)
```

```
## [1] 11.52369
```

```
#7) State statistical conclusion

#we can see that they is a quite significant difference between the mean of townho
uses and the 95% trust interval
#It means that it's not really possible to put townhouses into houses
#and that the means of those two groups are enoughly spaced to say that they are d
ifferent


#8) Make the inference based on conclusion

#Plus, another factor that could have been taken into account to classify houses f
orm townhouses would have been the number of Rooms
#I'm sure that, with the mean that we have calculated for houses, there are some t
hat are near townhouses in the middle of the city
#But if you look at the size of the houses they are maybe quite bigger than townho
uses, which explain the mean differences

#but for some town houses that are near the lower bound of houses and that are big
enough, I bet you could still put them as houses.
#However, I'm not sure it will affect the price, or if you still do it, you maybe
 will have some difficulties to sell them
```

3.2. Task B

Preparing the dataset for regression

```
#Divide the dataset into training and test data. Use 75/25 split.
housing_index <- sort(sample(nrow(housing.dataset), nrow(housing.dataset)*.75))
```

```
housing_train<-housing.dataset[housing_index,]
housing_test<-housing.dataset[-housing_index,]
```

Perform Linear Regression with Multiple Variables to predict the house price

```
#First, we have a look at the most correlated variables
cor(housing.dataset$Price, housing.dataset)
```

```
##          Suburb     Rooms       Type Price      Method     SellerG   Postcode
## [1,] -0.1272913 0.4124377 -0.3176301     1 0.009257079 -0.01927239 0.1664914
##      Propertycount   Distance
## [1,]   -0.06076933 -0.2536675
```

```
#then we do the Linear Regression with the most correlated variables
linearRegression <- lm(Price ~ Rooms + Suburb + Type + Postcode + Distance, data =
housing_train)

#R squared is 0.49, which is quite decent but not that much good
```

Report adjusted R squared (on training data). Use RMSE and correlation to report the prediction accuracy of the model on the test data

```
summary(linearRegression)
```

```
##
## Call:
## lm(formula = Price ~ Rooms + Suburb + Type + Postcode + Distance,
##     data = housing_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -7316730  -232216   -61221   146132  9391379
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  786716.98   13298.27   59.16   <2e-16 ***
## Rooms        257644.42    2822.10   91.30   <2e-16 ***
## Suburb         -383.28      21.41  -17.90   <2e-16 ***
## Type        -202581.02    3388.50  -59.78   <2e-16 ***
## Postcode       4692.54      47.33   99.14   <2e-16 ***
## Distance     -48677.40     346.54 -140.47   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 422500 on 36318 degrees of freedom
## Multiple R-squared:  0.4925, Adjusted R-squared:  0.4924
## F-statistic:  7049 on 5 and 36318 DF,  p-value: < 2.2e-16
```

```
#R squared = 0.49, which is not thatgreat
PricePred <- predict(linearRegression, housing_test)
#RMSE = 421 000
sqrt(mean((housing_test$Price - PricePred)^2))
```

```
## [1] 422973.4
```

```
#correlation = 0.70, of course because it's sqrt(R²)
cor(housing_test$Price,PricePred)
```

```
## [1] 0.7032033
```

Normalize the data and repeat the process of performing Linear Regression with Multiple Variables on normalized data to predict the house price.

```
#Normalize function
normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}

#normalize all the data rather than the two columns which were originaly int
housingTrain_norm <- as.data.frame(lapply(housing_train, normalize))
housingTest_norm <- as.data.frame(lapply(housing_test, normalize))
#housingTrain_norm$Price<-normalize(housingTrain_norm$Price)
#housingTrain_norm$Distance<-normalize(housingTrain_norm$Distance)

#another kind of normalization, that we will not use now
#housingTrain_norm$Price <- (housingTrain_norm$Price - mean(housingTrain_norm$Pric
e)) / sd(housingTrain_norm$Price)
#housingTrain_norm$Distance <- (housingTrain_norm$Distance - mean(housingTrain_nor
m$Distance)) / sd(housingTrain_norm$Distance)

#Repeating the linear regression process and analysis
linearRegression <- lm(Price ~ Rooms + Suburb + Type + Postcode + Distance, data =
housingTrain_norm)
summary(linearRegression)
```

```
##
## Call:
## lm(formula = Price ~ Rooms + Suburb + Type + Postcode + Distance,
##     data = housingTrain_norm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.65828 -0.02089 -0.00551  0.01315  0.84493
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0684741  0.0008110   84.43   <2e-16 ***
## Rooms        0.6953965  0.0076170   91.30   <2e-16 ***
## Suburb      -0.0127244  0.0007108  -17.90   <2e-16 ***
## Type        -0.0364518  0.0006097  -59.78   <2e-16 ***
## Postcode     0.0928798  0.0009368   99.14   <2e-16 ***
## Distance    -0.2443724  0.0017397 -140.47   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03801 on 36318 degrees of freedom
```

```
## Multiple R-squared:  0.4925, Adjusted R-squared:  0.4924
## F-statistic:  7049 on 5 and 36318 DF,  p-value: < 2.2e-16
```

```
PricePred <- predict(linearRegression, housingTest_norm)
sqrt(mean((housingTest_norm$Price - PricePred)^2))
```

```
## [1] 0.07257829
```

```
cor(housingTest_norm$Price,PricePred)
```

```
## [1] 0.6443488
```

Highlight the difference in prediction accuracy of both models

```
#Their is no differences in R squared, as we are not changing the distribution of
 the data, only the scale of them.
#A thing that I'm not sure about is why do the correlation is lower after the norm
alization, because I was thinking that R squared was the squared root of the corre
lation
#And if R squared is not change, why do the correlation changes, anyway, it's a bi
t of a mystery for me but the value do not changed that much
#and RMSE plummet, but it's normal because we are calculating distances, and after
the normalization, all the values are between 0 and 1 and so the distances are ver
y very tiny, and if you add a squared operation on top of that, it well be even lo
wer, that's why we jump from 420 000 to 0.07. because the distances are very tiny
 in a scale from 0 to 1.

#Anyway a correlation of 0.7 is not that good, but it maybe because of the initial
transformation of the data that we have done in the beginning.
```

3.3. Task C

Preparing data for classification

```
#Divide the data set into training and test data. Use 80/20 split.
housing_index <- sort(sample(nrow(housing.dataset), nrow(housing.dataset)*.8))
housing_train<-housing.dataset[housing_index,]
housing_test<-housing.dataset[-housing_index,]
```

#Use kNN to classify houses into appropriate types based on their features

```
#normalize the data set to help for analysis
housing.dataset<- as.data.frame(lapply(housing.dataset, normalize))

#dividing into test and training set
housingType_train <- as.numeric(as.factor(housing.dataset[housing_index,3]))
housingType_test <- as.numeric(as.factor(housing.dataset[-housing_index,3]))

#delete the Type column for the data
housing_train <- subset(housing_train, select = -c(Type))
housing_test <- subset(housing_test, select = -c(Type))
```

```
#launching the model
prevision <- knn(housing_train,housing_test,cl=housingType_train,k=3)

#putting the data together
table <- table(prevision,housingType_test)

#function to calculate accuracy of the model
accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) * 100}

accuracy(table)
```

```
## [1] 71.27078
```

```
#approximately 79% of accuracy, so almost correct
```

Use C5.0 to classify houses into appropriate types based on their features.

```
#modifying the training and test set
housingType_train <- as.factor(housing.dataset[housing_index,3])

#launching the model
model <-  C5.0( housing_train, housingType_train, trials=10 )

#making prediction using the model
prevision <- predict( model, housing_test, type="class" )

table <- table(prevision,housingType_test)

#calculating accuracy of the model
accuracy(table)
```

```
## [1] 84.18499
```

```
#approximately 85% of accuracy, so acceptable and better than the other one
```

Use ANN to classify houses into appropriate types based on their features.

```
#modifying the training and test set
housing.dataset<- as.data.frame(lapply(housing.dataset, normalize))
housing_train<-housing.dataset[housing_index,]
housing_test<-housing.dataset[-housing_index,]


housingTrain_index <- sort(sample(nrow(housing_train), nrow(housing_train)*.3))
housing_train<-housing_train[housingTrain_index,]

housingTest_index <- sort(sample(nrow(housing_test), nrow(housing_test)*.3))
housing_test<-housing_test[-housingTest_index,]

#launching the model
nn=neuralnet(Type~.,data = housing_train)
#without hidden = 5, which was too heavy for my PC (Intel Core i5-7300, 8 Go of RA
```

```
M, GTX 1050 graphic card)
#even when taking only 30% of the dataset was not enough, R is simply not as effic
ient as it's needed

#prediction of the data
pred = compute(nn, housing_test)
result = pred$net.result

#accuracy
cor(result, housing_test$Type)
```

```
##            [,1]
## [1,] 0.7265572
```

```
#74% of accuracy, so not that good but because of 1 hidden neuron was used
```

Evaluate and compare the (best) performance of each classifier

```
#we can see that all model are relatively accurate with more than 70% of accuracy
 for all of them
#The best model is C5.0 with 84% of accuracy
#Followed by KNN with 79%
#And last one ANN with 74%

#C5.0 is the fist one because decision tree are very well optimized for this task

#KNN is also a very efficient algorithm, but as I used only k = 3 with my function
form the class packages, it could be more optimized by getting a higher k and fusi
oning the results

#The ANN is the last one because I used only 1 hidden neuron and it's the minimum
 that you can do to generate a neural network, usually they are way bigger and wit
h hidden 5 it would have been surely more accurate but, R is not adapted to deep l
earning as Python or other languages could be
#so that why it's underperforming, with python and Google Colab, you could be sure
that the neural network is more efficient in this classification task
```