ADA442 - Project Report

Car Price Performance Categorization

Yağız Hikmet Karakuş, Furkan Özelge

2022-06-06 22:01:31

Table of Contents

Introduction

Nowadays, the car market and prices have been on the rise in Turkey. In this categorization process, we consider the features of the cars and determine whether their prices are reasonable or not. This is how we want to categorize the cars on the market. In other words, we can say that we measure whether cars are price-performance products or not. In this way, we will be able to interpret more accurately about the real values of the cars. We will be able to more accurately predict the actual prices of cars. We assume the tree model work better than logistic reggression Because of working with all categorical dataset.

Methodology

We will use Multinomial Classification Problem and Multinomial Logistic Regression. Since our problem is Multinomial Classification Problem, we used the most suitable Decision Tree and Multi Nominal Logistic Regression to make this classification. our data is all categorical and our response variable has 4 value thats why the multinominal logistic Regression is fit best for our dataset. # Data Description

We get our Data From UCI's machine leraning repository. Our Data Has 7 features but the last feature is our response feature

CAR car acceptability// response variable //independent variables and supersets

. PRICE overall price//superset . . buying buying price//independent variable . . maint price of the maintenance//independent variable . TECH technical characteristics//superset . . COMFORT comfort//superset . . . doors number of doors//independent variable . . . persons capacity in terms of persons to carry//independent variable . . . lug_boot the size of luggage boot//independent variable . . safety estimated safety of the car//independent variable

all of the variables are cathegorical. our Response varible is not a dummy varriable it has 4 class. because it has 4 class we cannot make binomial logistic reggression.

```
set.seed(44164)
url <- "https://archive.ics.uci.edu/ml/machine-learning-
databases/car/car.data"


data <- read.csv(url, header= FALSE)

colnames(data)<- c(
  "buying",
  "maint",
  "doors",
  "persons",
  "lug_boot",
  "safety",
  "response"
)
```

Explaratory Data Aanalysis (EDA) and Pre-Processing

```
dim(data)

## [1] 1728    7

summary(data)

##     buying             maint              doors             persons         
##  Length:1728        Length:1728        Length:1728        Length:1728       
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##    lug_boot            safety            response        
##  Length:1728        Length:1728        Length:1728       
##  Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character  

str(data)

## 'data.frame':    1728 obs. of  7 variables:
##  $ buying  : chr  "vhigh" "vhigh" "vhigh" "vhigh" ...
##  $ maint   : chr  "vhigh" "vhigh" "vhigh" "vhigh" ...
##  $ doors   : chr  "2" "2" "2" "2" ...
##  $ persons : chr  "2" "2" "2" "2" ...
```

```
##  $ lug_boot: chr  "small" "small" "small" "med" ...
##  $ safety  : chr  "low" "med" "high" "low" ...
##  $ response: chr  "unacc" "unacc" "unacc" "unacc" ...
```

we check the structure of data because all the variables are character and categorical we decide to make them factor.

```
data$response=as.factor(data$response)
data$buying <- as.factor(data$buying)
data$maint <- as.factor(data$maint)
data$doors <- as.factor(data$doors)
data$lug_boot <- as.factor(data$lug_boot)
data$safety <- as.factor(data$safety)
data$persons <- as.factor(data$persons)
str(data)
```

```
## 'data.frame':    1728 obs. of  7 variables:
##  $ buying  : Factor w/ 4 levels "high","low","med",..: 4 4 4 4 4 4 4 4 4 4
...
##  $ maint   : Factor w/ 4 levels "high","low","med",..: 4 4 4 4 4 4 4 4 4 4
...
##  $ doors   : Factor w/ 4 levels "2","3","4","5more": 1 1 1 1 1 1 1 1 1 1
...
##  $ persons : Factor w/ 3 levels "2","4","more": 1 1 1 1 1 1 1 1 1 2 ...
##  $ lug_boot: Factor w/ 3 levels "big","med","small": 3 3 3 2 2 2 1 1 1 3
...
##  $ safety  : Factor w/ 3 levels "high","low","med": 2 3 1 2 3 1 2 3 1 2
...
##  $ response: Factor w/ 4 levels "acc","good","unacc",..: 3 3 3 3 3 3 3 3 3
3 ...
```

after that we check if our data has na values

```
apply(is.na(data), 2, sum)
```

```
##   buying    maint    doors  persons lug_boot   safety response
##        0        0        0        0        0        0        0
```

After that we check corrolation table to understand which complication we can face with and are there highly corrolated

```
library(magrittr) # needs to be run every time you start R and want to use
%>%
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag
```

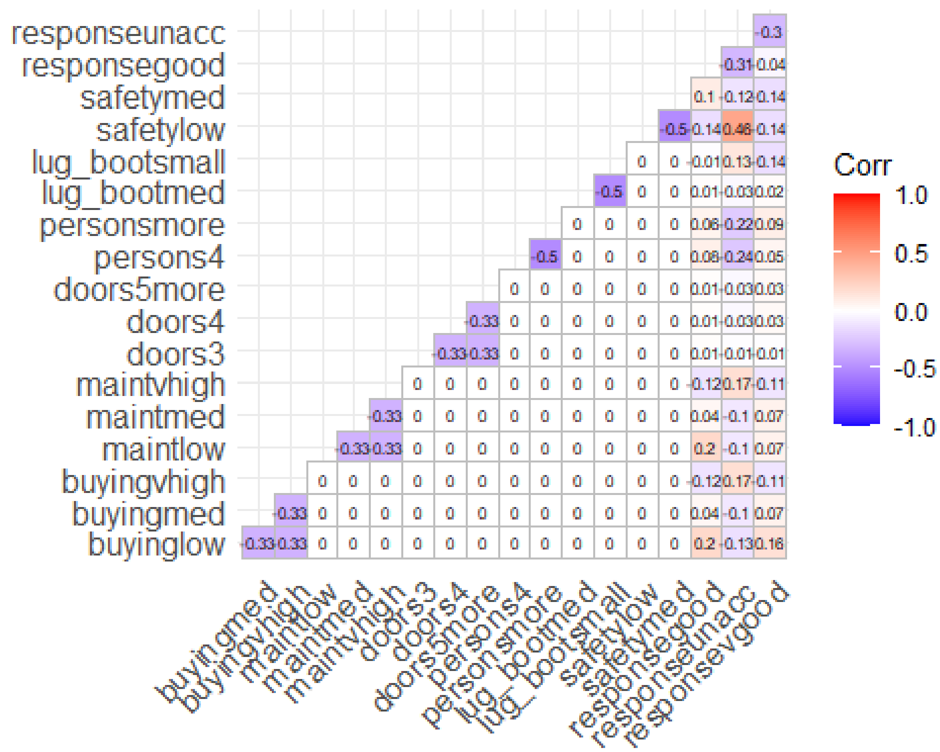
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggcorrplot)
```

```
## Zorunlu paket yükleniyor: ggplot2
```

```
model.matrix(~., data=data) %>%
  cor(method="spearman") %>%
  ggcorrplot(show.diag = F, type="lower", lab=TRUE, lab_size=2)
```

```
## Warning in cor(., method = "spearman"): the standard deviation is zero
```



Our corrolation table it doesn't seen any multicolonority problem

```
table(data$maint,data$buying)
```

```
##
##         high low med vhigh
##   high   108 108 108   108
##   low    108 108 108   108
##   med    108 108 108   108
##   vhigh  108 108 108   108
```

```
table(data$safety,data$buying)
```

```
##
##        high low med vhigh
```

```
##    high  144 144 144   144
##    low   144 144 144   144
##    med   144 144 144   144
```

```
table(data$doors,data$buying)
```

```
## 
##          high low med vhigh
##    2      108 108 108   108
##    3      108 108 108   108
##    4      108 108 108   108
##    5more  108 108 108   108
```

```
table(data$persons,data$buying)
```

```
## 
##         high low med vhigh
##    2     144 144 144   144
##    4     144 144 144   144
##    more  144 144 144   144
```

We can see our data distributed homogenously this can cause multicollinearity but before we check our model we don't want to intervention to this.

Data Partition

```
index=round(nrow(data)*0.8)

train_split=sample(nrow(data),size=index) #train indexes of the data
train=data[train_split,]
test=data[-train_split,]
```

Model Fit and Numerical Results

Multinominal Logistic Regression

```
# fit your model for the data
library(nnet)
model=multinom(response~. , data=train)
```

```
## # weights:  68 (48 variable)
## initial  value 1915.858807 
## iter  10 value 539.412524
## iter  20 value 397.285031
## iter  30 value 273.352848
## iter  40 value 202.158246
## iter  50 value 178.552587
## iter  60 value 175.883730
## iter  70 value 175.751211
## iter  80 value 175.740037
## iter  90 value 175.737649
```

```
## final  value 175.737533
## converged
```

```
summary(model)   #produces NAN's value
```

```
## Call:
## multinom(formula = response ~ ., data = train)
##
## Coefficients:
##       (Intercept) buyinglow buyingmed buyingvhigh  maintlow  maintmed
## good    -52.54385 33.243600 28.462587    2.009403 31.025963 26.767222
## unacc    48.42931 -4.830939 -3.850598    1.895756 -3.550647 -3.201397
## vgood   -60.53174 46.247345 39.887621    5.865216 16.403399 11.364294
##       maintvhigh    doors3    doors4 doors5more   persons4 personsmore
## good   -6.430643  2.868591  4.129263   4.083610   1.081586    1.124317
## unacc   2.607158 -2.082662 -2.531467  -2.546854 -49.741516  -49.185046
## vgood -36.046578  4.919278  7.164352   9.488344  13.979582   14.933399
##       lug_bootmed lug_bootsmall  safetylow  safetymed
## good    -2.863896     -9.630863  -7.205096  -8.765721
## unacc    1.364482      4.187777  45.774930   2.881661
## vgood   -6.244323    -35.449082 -17.931578 -32.363183
##
## Std. Errors:
##       (Intercept)   buyinglow   buyingmed buyingvhigh   maintlow
maintmed
## good    87.199903 174.8327967 174.8269168  422.244834 44.0277380
44.0252418
## unacc    0.349784   0.6289035   0.5390937    0.417802  0.5523148
0.5143468
## vgood  149.464116 224.1952746 224.1902876  597.491878  3.0687216
2.3148384
##         maintvhigh    doors3    doors4 doors5more   persons4 personsmore
## good  6.248637e-02 1.1170621 1.2791324  1.2981517 43.6059851  43.6063375
## unacc 4.450171e-01 0.4730773 0.4723378  0.4866316  0.2328833   0.2376061
## vgood 5.848077e-08 1.6490280 1.7800525  3.0472189 74.7340782  74.7335689
##       lug_bootmed lug_bootsmall    safetylow  safetymed
## good     1.118186   2.191369467 1.292671e-06 1.98136774
## unacc    0.413396   0.510334042 1.502906e-06 0.39641872
## vgood    1.580874   0.001844381 2.341705e-07 0.02143997
##
## Residual Deviance: 351.4751
## AIC: 447.4751
```

```
z=abs(summary(model)$coefficients)/abs(summary(model)$standard.errors)
p <- (1 - pnorm(abs(z), 0, 1)) * 2
p
```

```
##       (Intercept)    buyinglow    buyingmed  buyingvhigh     maintlow
## good    0.5467963 8.491954e-01 8.706725e-01 9.962030e-01 4.810024e-01
## unacc   0.0000000 1.576517e-14 9.150458e-13 5.693823e-06 1.287288e-10
## vgood   0.6854835 8.365710e-01 8.587869e-01 9.921678e-01 9.024121e-08
```

```
##            maintmed   maintvhigh       doors3       doors4   doors5more
## good  5.431894e-01 0.000000e+00 1.022937e-02 1.245829e-03 1.656839e-03
## unacc 4.840242e-10 4.669086e-09 1.070738e-05 8.347938e-08 1.662056e-07
## vgood 9.139039e-07 0.000000e+00 2.853096e-03 5.702419e-05 1.847124e-03
##        persons4 personsmore  lug_bootmed lug_bootsmall safetylow
safetymed
## good  0.9802116   0.9794301 1.043120e-02  1.108207e-05         0
9.685611e-06
## unacc 0.0000000   0.0000000 9.645554e-04  2.220446e-16         0
3.614886e-13
## vgood 0.8516154   0.8416199 7.817811e-05  0.000000e+00         0
0.000000e+00
```

our variables are significant.

```
# Testing the performance of the fitted model
pred_log_reg <- predict(model, test, type = "class")
confusion_matrix=table(test$response, pred_log_reg)

accuracy <- sum(diag(confusion_matrix))/sum(confusion_matrix)



accuracy

## [1] 0.9248555
```

we have really high acuracy. Multinomial Logistic Regression seems like enough for this categorization. We think The NaN values can be caused of the multicollinarity. in order to solve that we want to drop the problematic column and try to test our new model when we examine when we check our corrolation matrix we see ther are the highest corrolation on saftey so we decide to drop that independent variable

```
# fit your model for the data
library(nnet)
model2=multinom(response~.- safety , data=train)

## # weights:  60 (42 variable)
## initial  value 1915.858807
## iter  10 value 819.911167
## iter  20 value 755.667006
## iter  30 value 715.444143
## iter  40 value 707.967207
## iter  50 value 706.878985
## final  value 706.873698
## converged

summary(model2)

## Call:
## multinom(formula = response ~ . - safety, data = train)
##
```

```
## Coefficients:
##       (Intercept)  buyinglow  buyingmed buyingvhigh   maintlow   maintmed
## good    -36.71751 18.3328048 17.2648947  -0.6073226 18.2104723 17.2164125
## unacc    20.79720 -0.3927297 -0.4254737   0.6973950 -0.4186582 -0.5399779
## vgood   -21.06281 18.9240826 18.1774262   0.5121546  1.1078702  0.9181443
##         maintvhigh     doors3     doors4 doors5more    persons4 personsmore
## good    -1.9260932  0.2660360  0.5542596  0.2106382   0.5570556   0.5816897
## unacc    0.6356377 -0.3613855 -0.5242026 -0.4478328 -20.2843951 -20.1976717
## vgood -15.5164069  0.2678235  0.6607107  0.3680412   1.2387177   1.4932549
##       lug_bootmed lug_bootsmall
## good   -0.0896135    -0.1925813
## unacc   0.1753914     0.8617613
## vgood  -0.9812742   -20.1653663
##
## Std. Errors:
##       (Intercept) buyinglow buyingmed  buyingvhigh  maintlow  maintmed
## good    0.1787536 0.1921071 0.2124105 1.619976e-08 0.1984244 0.2115378
## unacc   0.1728823 0.2194315 0.2086095 2.121296e-01 0.2182036 0.2124766
## vgood   0.2488518 0.2081391 0.2275586 4.414225e-08 0.4562202 0.4364809
##         maintvhigh    doors3    doors4 doors5more  persons4 personsmore
## good  4.884148e-09 0.4985719 0.4940909  0.5055892 0.1911405   0.2003177
## unacc 2.134115e-01 0.2181814 0.2159543  0.2184573 0.1144193   0.1162294
## vgood 1.939387e-07 0.5136498 0.4967952  0.5072546 0.2096842   0.2184855
##       lug_bootmed lug_bootsmall
## good    0.4303202  4.332814e-01
## unacc   0.1834821  1.928172e-01
## vgood   0.3707571  3.511361e-09
##
## Residual Deviance: 1413.747
## AIC: 1497.747
```

We see our NaN values dissapear but AIC and Deviance values increase rapidly it will affect our accuracy in order to check we train our data

```
pred_log_reg2 <- predict(model2, test, type = "class")
confusion_matrix2=table(test$response, pred_log_reg2)

accuracy <- sum(diag(confusion_matrix2))/sum(confusion_matrix2)



accuracy

## [1] 0.6936416
```

We decide that to go on with our first model. because of high accuracy and lower Residual deviance and AIC values.

Decision Tree

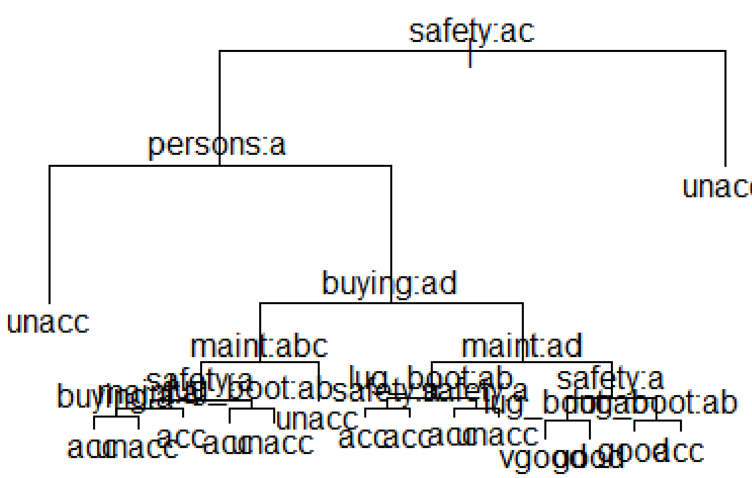
```
library(tree)
```

```
## Registered S3 method overwritten by 'tree':
##   method     from
##   print.tree cli
```

```
tree.model <- tree(response ~ ., data=train)
summary(tree.model)
```

```
##
## Classification tree:
## tree(formula = response ~ ., data = train)
## Variables actually used in tree construction:
## [1] "safety"   "persons"  "buying"   "maint"    "lug_boot"
## Number of terminal nodes:  16
## Residual mean deviance:  0.3189 = 435.6 / 1366
## Misclassification error rate: 0.06585 = 91 / 1382
```

```
plot(tree.model)
text(tree.model)
```




```
pred.tree <- predict(tree.model, test, type = "class")

tree_confMat <- table(test$response,pred.tree)
tree_confMat
```

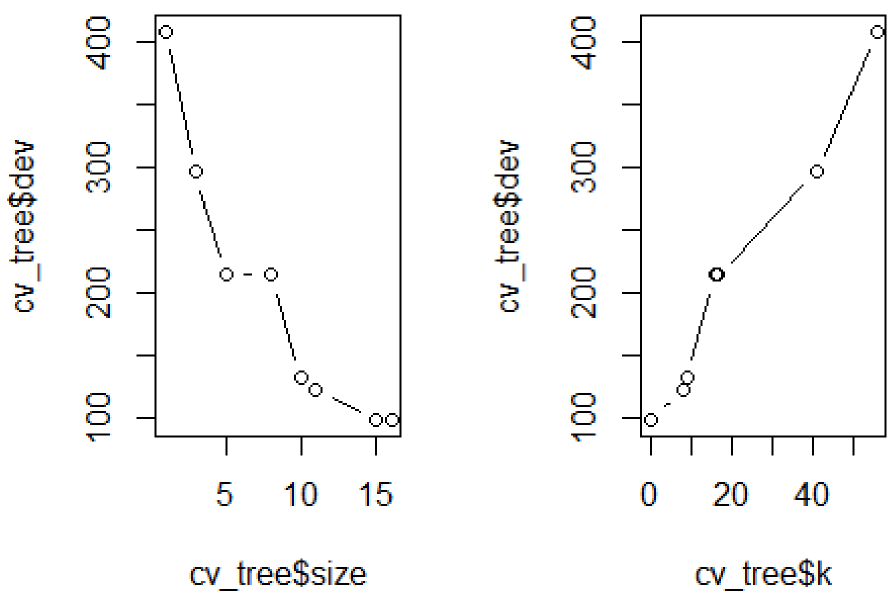
```
##        pred.tree
##          acc good unacc vgood
##   acc     67    7     2     2
##   good     0   14     0     4
```

```
##    unacc   14     0    222      0
##    vgood    1     0      0     13
```

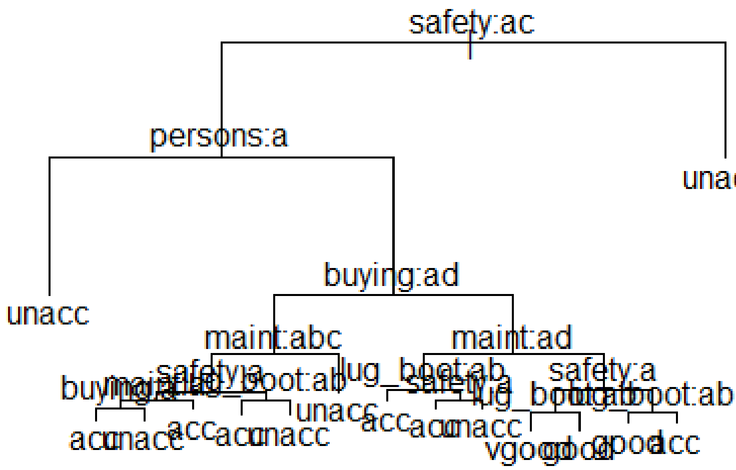
```
tree_accuracy <- sum(diag(tree_confMat))/sum(tree_confMat)
tree_accuracy
```

```
## [1] 0.9132948
```

```
cv_tree <- cv.tree(tree.model, FUN = prune.misclass)
par(mfrow = c(1, 2))
plot(cv_tree$size, cv_tree$dev, type = "b")
plot(cv_tree$k, cv_tree$dev, type = "b")
```




```
new.tree <- prune.misclass(tree.model, best = 13)
plot(new.tree)
text(new.tree)
```




```
pred.tree_prune2 <- predict(new.tree, test, type = "class")

tree_confMat2 <- table(test$response,pred.tree_prune2)
tree_confMat2

##        pred.tree_prune2
##         acc good unacc vgood
##   acc    67    7     2     2
##   good    0   14     0     4
##   unacc  14    0   222     0
##   vgood   1    0     0    13

tree_accuracy2 <- sum(diag(tree_confMat2))/sum(tree_confMat2)
tree_accuracy2

## [1] 0.9132948
```

Our Accuracy and confusion matrix didn't change and we see we can get the same result with less nodes.

```
summary(new.tree)

##
## Classification tree:
## snip.tree(tree = tree.model, nodes = 44L)
## Variables actually used in tree construction:
## [1] "safety"   "persons"  "buying"   "maint"    "lug_boot"
## Number of terminal nodes:  15
```

```
## Residual mean deviance:  0.3381 = 462.1 / 1367
## Misclassification error rate: 0.06585 = 91 / 1382
```

```
summary(model)
```

```
## Call:
## multinom(formula = response ~ ., data = train)
##
## Coefficients:
##       (Intercept) buyinglow buyingmed buyingvhigh  maintlow  maintmed
## good    -52.54385 33.243600 28.462587    2.009403 31.025963 26.767222
## unacc    48.42931 -4.830939 -3.850598    1.895756 -3.550647 -3.201397
## vgood   -60.53174 46.247345 39.887621    5.865216 16.403399 11.364294
##       maintvhigh    doors3    doors4 doors5more   persons4 personsmore
## good   -6.430643  2.868591  4.129263   4.083610   1.081586    1.124317
## unacc   2.607158 -2.082662 -2.531467  -2.546854 -49.741516  -49.185046
## vgood -36.046578  4.919278  7.164352   9.488344  13.979582   14.933399
##       lug_bootmed lug_bootsmall  safetylow  safetymed
## good    -2.863896     -9.630863  -7.205096  -8.765721
## unacc    1.364482      4.187777  45.774930   2.881661
## vgood   -6.244323    -35.449082 -17.931578 -32.363183
##
## Std. Errors:
##       (Intercept)   buyinglow   buyingmed buyingvhigh   maintlow
maintmed
## good    87.199903 174.8327967 174.8269168  422.244834 44.0277380
44.0252418
## unacc    0.349784   0.6289035   0.5390937    0.417802  0.5523148
0.5143468
## vgood  149.464116 224.1952746 224.1902876  597.491878  3.0687216
2.3148384
##         maintvhigh    doors3    doors4 doors5more   persons4 personsmore
## good  6.248637e-02 1.1170621 1.2791324  1.2981517 43.6059851  43.6063375
## unacc 4.450171e-01 0.4730773 0.4723378  0.4866316  0.2328833   0.2376061
## vgood 5.848077e-08 1.6490280 1.7800525  3.0472189 74.7340782  74.7335689
##       lug_bootmed lug_bootsmall    safetylow  safetymed
## good     1.118186   2.191369467 1.292671e-06 1.98136774
## unacc    0.413396   0.510334042 1.502906e-06 0.39641872
## vgood    1.580874   0.001844381 2.341705e-07 0.02143997
##
## Residual Deviance: 351.4751
## AIC: 447.4751
```

The residual deviance of the decison trr model seems more than multinominal logistic regression also when we check their accuracy level.Even if there is a small difference Multinominal Logistic Regression is better than Decsion Tree for categrize our dataset. # Conclusions

We worked with a data set that was difficult to work with. For this reason, although we achieved what we wanted, we faced some problems. We worked really hard to make the

Classification, but sometimes, no matter how well we work with the right method, things don't go the way we want. The Decision Tree and Multi Nominal Regression we used to make the Classification were sufficient for us, but we still ran into a problem. This problem was caused by our dataset. When we crosstable the independent variables from our dataset, we saw that they were distributed very homogeneously. In fact, they all had the same value. For this reason, we realized that our data is more suitable for decision tree than logistic regression. But when we test our data with different models we found ou Multinominal Logistic Regression is better to apply for our data this reason we choose the multinomial. logistic regression for our final conclusion # References

- Marko Bohanec (1997, June 01). Car Evaluation Data Set. Retrieved January 20, 2022, from https://archive.ics.uci.edu/ml/datasets/Car+Evaluation