MATH3714 Coursework

Viet Dao
email: mm16vd@leeds.ac.uk

November 26, 2018

Contents

1 Introduction

We have been given a data frame $\mathbf{A}_{393x9}$ which is the table of different cars with mpg, cylinders, displacement, horsepower, weight, acceleration, year, origin and name for a given car. Our goals is to be make a model that is capable of predicting mpg from our data given. Now we plit up $\mathbf{A}_{393x9}$ into $\mathbf{Y}_{393x1}$ which contains only mpg and $\mathbf{Y}_{393x8}$ which contains everything in $\mathbf{A}_{393x9}$ apart from mpg. This sets up our responds and explainatory variable.

2 Initial Data Analysis

In this prelimatory stage we want to investigate outliers and possible missing data in our dataframe $\mathbf{A}_{393x9}$. The summary of our data is a useful point to start from:

```
> dat = read.table("http://www1.maths.leeds.ac.uk/~charles/math3714/Auto.csv",
 header = T)
> View(dat)
> summary(dat)
mpg             cylinders       displacement     horsepower
Min.   : 9.0    Min.   :3.000   Min.   : 68.0   Min.   : 46.0
1st Qu.:17.0    1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0
Median :23.0    Median:4.000    Median :151.0   Median : 94.0
Mean   :23.5    Mean   :5.468   Mean   :194.1   Mean   :104.5
3rd Qu.:29.0    3rd Qu.:8.000   3rd Qu.:267.0   3rd Qu.:125.0
Max.   :46.6    Max.   :8.000   Max.   :455.0   Max.   :230.0

weight          acceleration    year            origin
Min.   :1613    Min.   : 8.00   Min.   :18.00   Min.   :1.000
1st Qu.:2226    1st Qu.:13.70   1st Qu.:73.00   1st Qu.:1.000
Median :2807    Median :15.50   Median :76.00   Median :1.000
Mean   :2978    Mean   :15.52   Mean   :75.83   Mean   :1.578
3rd Qu.:3613    3rd Qu.:17.00   3rd Qu.:79.00   3rd Qu.:2.000
Max.   :5140    Max.   :24.80   Max.   :82.00   Max.   :3.000

name
amc matador       :  5
ford pinto        :  5
toyota corolla    :  5
amc gremlin       :  4
amc hornet        :  4
chevrolet chevette:  4
(Other)           :366
```

There are several problem with the data:

- First there is a problem with a data in the year. the summary says the earliest car made was in 18, is it 1918 or 2018?. On further inspection using View(dat) command we can see the name of that car is 'vw golf estate S 1.4 TSI' clearly from

2018 rather than 1918. This need to be changed from 18 to 118.

Solution

```
> dat$year[dat$name=='vw␣golf␣estate␣S␣1.4␣TSI'] = 118
> View(dat)
```

This fix the year releasing date for vw golf estate S 1.4 TSI.
Note: there should be a space between 'vw' and 'golf' instead of weird symbol.

- The second problem is the name of the cars. This problem lies in the make of the car and the name of the cars are in the same string hence we are not able to 'encode' this properly i.e. amc hornet, amc gremlin are almost identical but if we were to fit these values under the model it would be treated as different. From here, the name can be plit into two more groups, which is make of the car and name of the car. From there the make of the car can be encoded, similar to the origin of the car.

  Solution

  The first thing to notice in the 'name' header is that the first word is the 'make' of the car and the rest is the 'model' of the car. Now take the first word of the string and add it to make while for name remove the first word of the string.

```
dat = read.table("http://www1.maths.leeds.ac.uk/~charles/math3714/Auto.csv",
#---Addressing 2nd problem
#In order to achieved this I need to add an extra tag into the
#dataframe which is "stringAsFactors=F".
#adding a extra entry called make which stands for the maker of the car.
dat$make = dat$name

#changing the string into the first word of the sring.
#Then attaching the first word of the string to make table.
for(string in dat$make){
  substring = strsplit(string, "␣")[[1]]
  maker = substring[1]
  print(maker)
  dat$make[dat$make==string]=maker
}

#changing the string into every word apart from the first word.
for(string in dat$name){
  substring = strsplit(string, "␣")[[1]]
  print(paste(substring[-1], collapse='␣' ))
  dat$name[dat$name==string]=paste(substring[-1], collapse='␣' )
}
```

This should produce a new table with 'make' and 'name'.
NOTE: when importing the table the 'stringAsFactors=F' is a must else this wouldn't work.

- A problem that arise from spliting the ‘name‘ column into ‘name‘ and ‘make‘ is the fact that the ‘make‘ is a catergorical data and this need to be encoded i.e. convert catergory into integers, similarly to the origin which is a catergorical data but represented by 1-3.
  Solution: R-Code

```
>table(dat$make)
amc     audi    bmw     buick   cadillac        capri   chevroelt
27      7       2       17      2               1       1
chevrolet       chevy   chrysler        datsun
43              3       6               23
dodge   fiat    ford    hi      honda   maxda   mazda
28      8       48      1       13      2       10
mercedes mercedes-benz  mercury nissan
1        2              11      1
oldsmobile      opel    peugeot plymouth        pontiac renault
10              4       8       31              16      3
saab    subaru  toyota  toyouta triumph
4       4       25      1       1
vokswagen       volkswagen      volvo   vw
1               15              6       7
```

  The table produce is in the form of ‘maker‘ of the cars and directly below is the number of occurances in the table.

- Another problem lies in the fact that the data use several acronyms for the name make i.e. chevrolet and chevy, vw and volkswagen etc... This is a problem since it adds unwated complexity to our data. Therefore the data needs to be changed.
  Solution R-Code:

```
#---Problem 3
#Changing the make to a proper make.
for(string in dat$make){
  if(string=="chevroelt" | string=="chevy"){
    dat$make[dat$make==string]="chevrolet"
  }
  if(string=="maxda"){
    dat$make[dat$make==string]="mazda"
  }
  if(string=="mercedes-benz"){
    dat$make[dat$make==string]="mercedes"
  }
  if(string=="toyouta"){
    dat$make[dat$make==string]="toyota"
  }
  if(string=="vokswagen"|string=="vw"){
    dat$make[dat$make==string]="volkswagen"
  }
}
```

- The name of the vehicle is also a problem. This is beacause the vehical name is very unique and dependent on the maker of that car i.e. ‘100ls’ is dependent on audi since only ‘audi’ make cars with those names. This also poses the problem of that the name is so unique that it can cause over fitting. The solution is not delete the name column and only include the brand as one of our explainatory variable.

  ```
  #---Problem 4
  dat$name=NULL
  ```