МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Факультет прикладної математики та інформатики

ЗВІТ

до індивідуального завдання №3

з дисципліни «Моделі статистичного навчання»

Виконав
студент групи ПМіМ-12:
Бордун Михайло

Перевірив:
Проф. Заболоцький Т. М.

Львів – 2021

Хід виконання

1.Аналіз даних Weekly, що містять 1 089 щотижневі дохідності за 21 рік, з початку 1990 р. до кінця 2010 року.

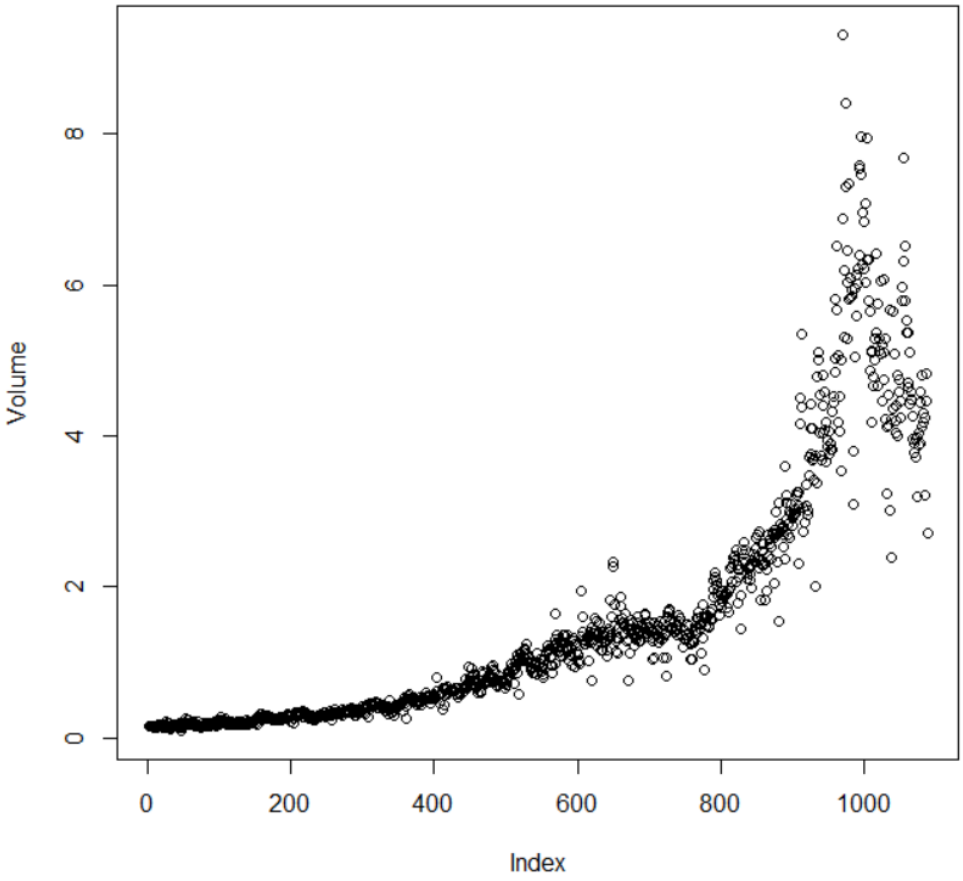
Розглянемо датасет Weekly. Для кожної дати наявна дохідність для попередніх 5 днів Lag1,…,Lag5. Змінна Volume містить дані про обсяг торгів попереднього дня у млн., Today – сьогоднішня дохідність, та змінна Direction, яка вказує чи зріс ринок чи впав.

```
     Year           Lag1               Lag2               Lag3          
Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950  
1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580  
Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410  
Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472  
3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090  
Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260  
     Lag4               Lag5              Volume            Today        
Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747   Min.   :-18.1950  
1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202   1st Qu.: -1.1540  
Median :  0.2380   Median :  0.2340   Median :1.00268   Median :  0.2410  
Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462   Mean   :  0.1499  
3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373   3rd Qu.:  1.4050  
Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821   Max.   : 12.0260  
Direction
Down:484
Up  :605
```

1.1 Розглянуто кореляції між змінними нашого датасету за допомогою функції cor().

```
              Year         Lag1        Lag2        Lag3         Lag4
Year    1.00000000 -0.032289274 -0.03339001 -0.03000649 -0.031127923
Lag1   -0.03228927  1.000000000 -0.07485305  0.05863568 -0.071273876
Lag2   -0.03339001 -0.074853051  1.00000000 -0.07572091  0.058381535
Lag3   -0.03000649  0.058635682 -0.07572091  1.00000000 -0.075395865
Lag4   -0.03112792 -0.071273876  0.05838153 -0.07539587  1.000000000
Lag5   -0.03051910 -0.008183096 -0.07249948  0.06065717 -0.075675027
Volume  0.84194162 -0.064951313 -0.08551314 -0.06928771 -0.061074617
Today  -0.03245989 -0.075031842  0.05916672 -0.07124364 -0.007825873
               Lag5      Volume       Today
Year    -0.030519101  0.84194162 -0.032459894
Lag1    -0.008183096 -0.06495131 -0.075031842
Lag2    -0.072499482 -0.08551314  0.059166717
Lag3     0.060657175 -0.06928771 -0.071243639
Lag4    -0.075675027 -0.06107462 -0.007825873
Lag5     1.000000000 -0.05851741  0.011012698
Volume  -0.058517414  1.00000000 -0.033077783
Today    0.011012698 -0.03307778  1.000000000
```

Як бачимо чітку кореляцію тільки між змінною Year та Volume. Можна представити це графічно за допомогою функції plot().



Бачимо, що змінна Volume зростає з плином часу, при чому загалом бачимо експоненціальний зріст.

1.2 Побудовано логістичну регресію, де Direction – залежна змінна, а п’ять зміщенних дохідностей та змінна Volume незалежні.

```
Call:
glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
    Volume, family = binomial, data = Weekly)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6949  -1.2565   0.9913   1.0849   1.4579

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.26686    0.08593   3.106   0.0019 **
Lag1        -0.04127    0.02641  -1.563   0.1181
Lag2         0.05844    0.02686   2.175   0.0296 *
Lag3        -0.01606    0.02666  -0.602   0.5469
Lag4        -0.02779    0.02646  -1.050   0.2937
Lag5        -0.01447    0.02638  -0.549   0.5833
Volume      -0.02274    0.03690  -0.616   0.5377
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1496.2  on 1088  degrees of freedom
Residual deviance: 1486.4  on 1082  degrees of freedom
AIC: 1500.4

Number of Fisher Scoring iterations: 4
```

Для нашої моделі бачимо, що найменше p-value відповідає змінній Lag2, тобто кореляція є суттєва. Тим паче, є наявний позитивний коефіцієнт біля цієї змінної, що вказує на те, що якщо позавчора ринок мав негативну дохідність, то сьогодні більше шансів зрости.

1.3 Побудовано матрицю помилок та обчислено загальну частку правильних прогнозів.

```
         Direction
pred.glm Down  Up
    Down   54  48
    Up    430 557
[1] "Correct predictions rate:  0.561065197428834"
[1] "Correct predictions rate when the market goes up:  0.920661157024793"
[1] "Correct predictions rate when the market goes down:  0.111570247933884"
```

Тобто, загальна частка правильних прогнозів становить 56.1%, що є досить поганим результатом прогнозування. Впродовж тижнів коли ринок йде вгору, модель правильно прогнозує у 92.1% випадків (557/(48+557))%. Впродовж тижнів коли ринок спадає, модель має рацію лише в 11.2% випадків (54/(54+430))%.

1.4 Побудовано модель логістичної регресії з використанням навчальних даних з 1990 по 2008 рр., з єдиним предиктором Lag2.

```
train = (Year >= 1990 & Year <= 2008)
Weekly.test = Weekly[!train, ]
Direction.test = Direction[!train]

cat("\n")
print(paste("Rows of train data: ", dim(Weekly.test)[1]))
```

```
[1] "Rows of train data:  104"

Call:
glm(formula = Direction ~ Lag2, family = binomial, data = Weekly,
    subset = train)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.536  -1.264   1.021   1.091   1.368

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.20326    0.06428   3.162  0.00157 **
Lag2         0.05810    0.02870   2.024  0.04298 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1354.7  on 984  degrees of freedom
Residual deviance: 1350.5  on 983  degrees of freedom
AIC: 1354.5

Number of Fisher Scoring iterations: 4
```

```
         Direction.test
pred.glm2 Down Up
     Down    9  5
     Up     34 56
[1] "Correct predictions rate:  0.625"
[1] "Correct predictions rate when the market goes up:  0.918032786885246"
[1] "Correct predictions rate when the market goes down:  0.209302325581395"
```

З огляду на матрицю помилок, бачимо, що модель схильна позначати більшу частину тестових даних як підйом ринку. Загалом частка правильних прогнозів становить 62.5%. Впродовж тижнів коли ринок йде вгору, модель правильно прогнозує у 91.8% випадків (56/(56+5))%. Впродовж тижнів коли ринок спадає, модель має рацію лише в 20.9% випадків (9/(34+9))%.

1.5 Побудовано модель лінійного дискримінантного аналізу з використанням навчальних даних з 1990 по 2008 рр., з єдиним предиктором Lag2.

```
Call:
lda(Direction ~ Lag2, data = Weekly, subset = train)

Prior probabilities of groups:
     Down        Up
0.4477157 0.5522843

Group means:
            Lag2
Down -0.03568254
Up    0.26036581

Coefficients of linear discriminants:
           LD1
Lag2 0.4414162

      Direction.test
       Down Up
  Down    9  5
  Up     34 56
[1] "Correct predictions rate:  0.625"
[1] "Correct predictions rate when the market goes up:  0.918032786885246"
[1] "Correct predictions rate when the market goes down:  0.209302325581395"
```

Як бачимо навчальні дані є досить добре розподілені і 55% вибірки позначено як зростання ринку. Ми також отримали групові середні, з яких бачимо, що позавчора коли ринок спадає, то дохідність є негативна і навпаки.

З огляду на матрицю помилок, бачимо, що модель схильна позначати більшу частину тестових даних як підйом ринку. Загалом частка правильних прогнозів становить 62.5%. Впродовж тижнів коли ринок йде вгору, модель правильно прогнозує у 91.8% випадків (56/(56+5))%. Впродовж тижнів коли ринок спадає, модель має рацію лише в 20.9% випадків (9/(34+9))%.

1.6 Побудовано модель квадратичного дискримінантного аналізу з використанням навчальних даних з 1990 по 2008 рр., з єдиним предиктором Lag2.

```
Call:
qda(Direction ~ Lag2, data = Weekly, subset = train)

Prior probabilities of groups:
     Down        Up
0.4477157 0.5522843

Group means:
            Lag2
Down -0.03568254
Up    0.26036581

      Direction.test
       Down Up
  Down    0  0
  Up     43 61
[1] "Correct predictions rate:  0.586538461538462"
[1] "Correct predictions rate when the market goes up:  1"
[1] "Correct predictions rate when the market goes down:  0"
```

Бачимо цікаву картину аналізуючи матрицю помилок. Тобто, модель позначає весь час тестові дані як підйом ринку. Загалом частка правильних прогнозів становить 58.7%. Впродовж тижнів коли ринок йде вгору, модель правильно прогнозує у 100% випадків і навпаки, коли ринок спадає, модель робить повністю хибні прогнози.

1.7 Побудовано класифікатор K-найближчих сусідів з K=1 з використанням навчальних даних з 1990 по 2008 рр., з єдиним предиктором Lag2.

Нижче наведений код для формування перших двох аргументів для функції прогнозування. Тобто формуємо матриці з предиктора Lag2, які пов’язані чисто з навчальними даними та тестовими. Set.seed(1) встановлено для випадку, якщо кілька спостережень розглядатимуться як найближчі сусіди, то R буде випадковим чином вибирати потрібну кількість.

```
train.X = as.matrix(Lag2[train])
test.X = as.matrix(Lag2[!train])

Direction.train = Direction[train]
set.seed(1)
pred.knn = knn(train.X, test.X, Direction.train, k = 1)
```

```
        Direction.test
pred.knn Down Up
    Down   21 30
    Up     22 31
[1] "Correct predictions rate:  0.5"
[1] "Correct predictions rate when the market goes up:  0.508196721311475"
[1] "Correct predictions rate when the market goes down:  0.488372093023256"
```

З огляду на матрицю помилок бачимо, що частка правильних прогнозів становить 50%. Впродовж тижнів коли ринок йде вгору, модель правильно прогнозує у 50.8% випадків (31/(30+31))%. Впродовж тижнів коли ринок спадає, модель має рацію в 48.8% випадків (21/(22+21))%.

1.8

```
"Test error rate table"
"Knn test error rate :  0.5"
"Qda test error rate :  0.413461538461538"
"Lda test error rate :  0.375"
"Glm test error rate :  0.375"
```

Вище наведено таблицю тестових помилок вищезгаданих моделей. Звідси очевидно, що найбільш відповідними для нашої вибірки виявились LDA та GLM моделі.

1.9 Поекспериментував з різними комбінаціями предикторів, використовуючи у тому числі можливі перетворення та взаємодії.

```
fit.glm3 =  glm(Direction ~ Lag2*Lag1, data = Weekly, family = binomial, subset = t
rain)
```

```
         Direction.test
pred.glm3 Down Up
     Down    7  8
     Up     36 53
[1] "Correct predictions rate:  0.576923076923077"
```

```
fit.lda2 = lda(Direction ~ Lag2:Lag1 + Lag1, data = Weekly, subset = train)
```

```
       Direction.test
        Down Up
  Down    4  4
  Up     39 57
[1] "Correct predictions rate:  0.586538461538462"
```

```
fit.qda2 = qda(Direction ~ Lag2 + abs(Lag2), data = Weekly, subset = train)
```

```
       Direction.test
        Down Up
  Down   11 12
  Up     32 49
[1] "Correct predictions rate:  0.576923076923077"
```

```
         Direction.test
pred.knn2 Down Up
     Down   15 20
     Up     28 41
[1] "Correct predictions rate:  0.538461538461538"
         Direction.test
pred.knn3 Down Up
     Down   20 20
     Up     23 41
[1] "Correct predictions rate:  0.586538461538462"
         Direction.test
pred.knn4 Down Up
     Down   19 23
     Up     24 38
[1] "Correct predictions rate:  0.548076923076923"
```

В наступних результатах подані матриці помилок для моделей К-найближчих сусідів відповідно із значеннями K= 5, 15, 30. Тобто для значення K=15 бачимо найкращу частку правильних прогнозів на тестових даних, а саме 58.7%, це найкращий результат прогнозування серед всіх попередньо розглянутих класифікаторів К-найближчих сусідів. Такого ж результату було досягнуто з використання LDA, де було використано змінну взаємодії та змінну Lag1. Але, бачимо, що використання змінних взаємодії не приносить покращень щодо прогнозування і базові моделі логістичної регресії та LDA мають найкращу точність (62.5%).

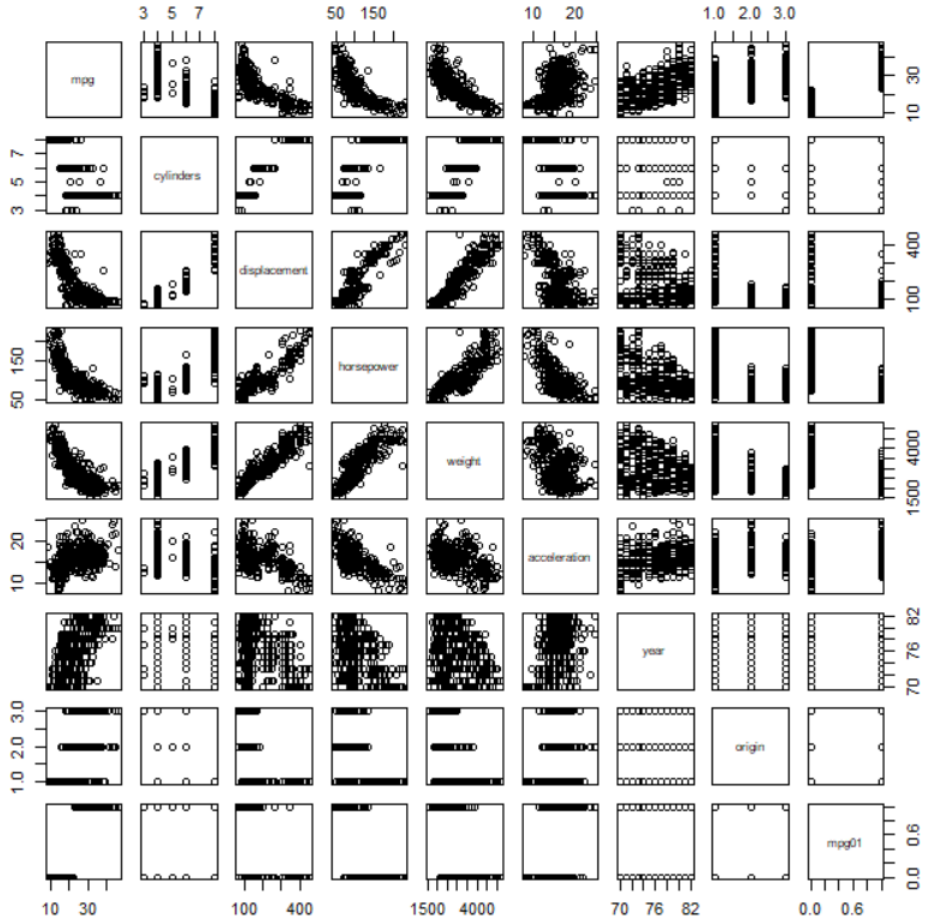
2. Модель класифікації для передбачення, чи вибране авто має велике або низьке споживання газу на базі даних Auto.

2.1 Було створено двійкову змінну mpg01, яка містить 1, якщо mpg містить значення вище медіани, і 0, якщо mpg містить значення нижче його медіани. А також створено єдиний набір даних , що містить як mpg01, так і інші змінні з датасету Auto.

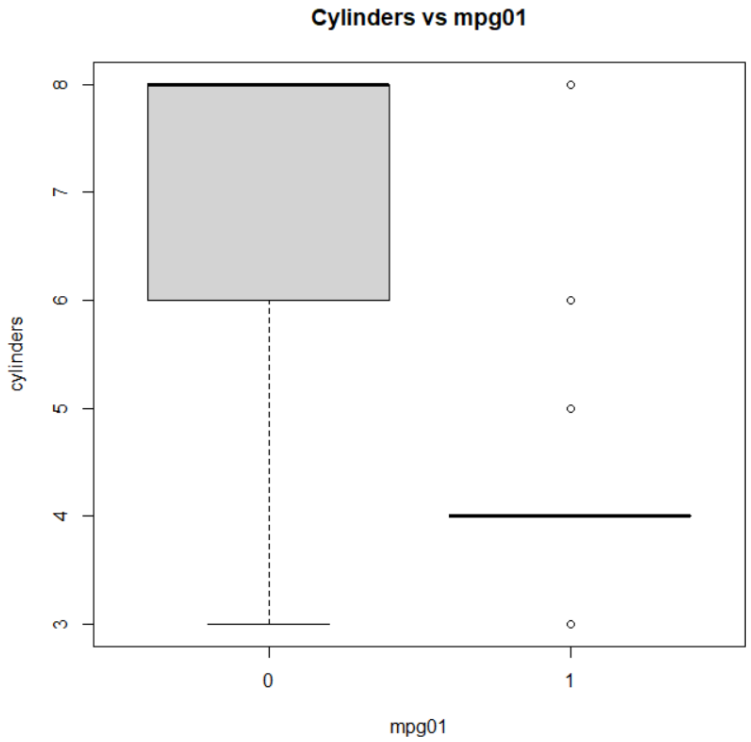
```
attach(autos)
mpg01 = rep(0, length(mpg))
mpg01[mpg > median(mpg)] = 1
autos = data.frame(autos, mpg01)
```

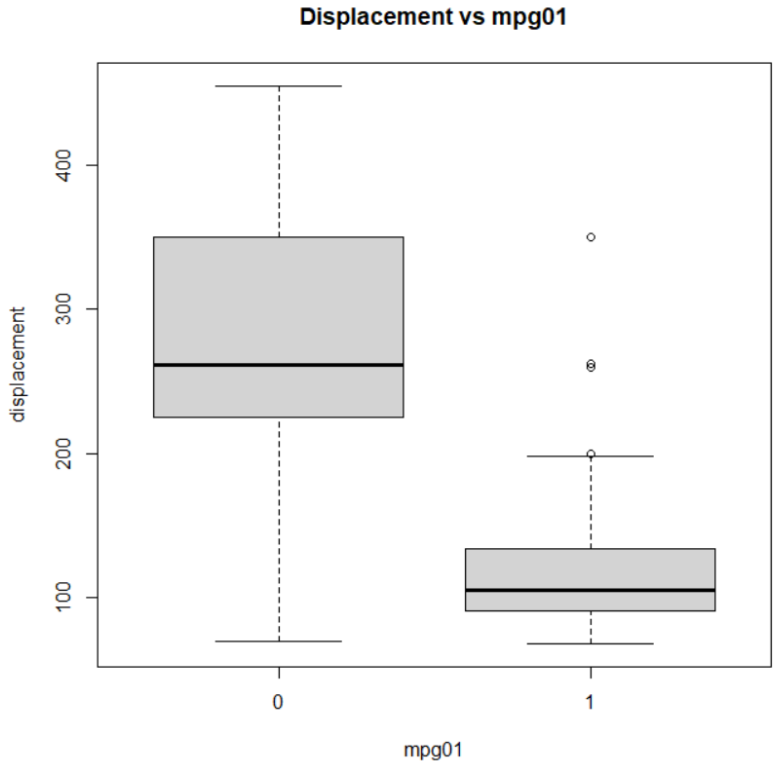
2.2 Дослідимо змінні та їх залежності з датасету Auto. Для початку використано функцію cor(), щоб побачити настільки сильна чи слаба кореляція між змінними. А також за допомогою функції pairs() виведено графічну залежність всіх змінних.

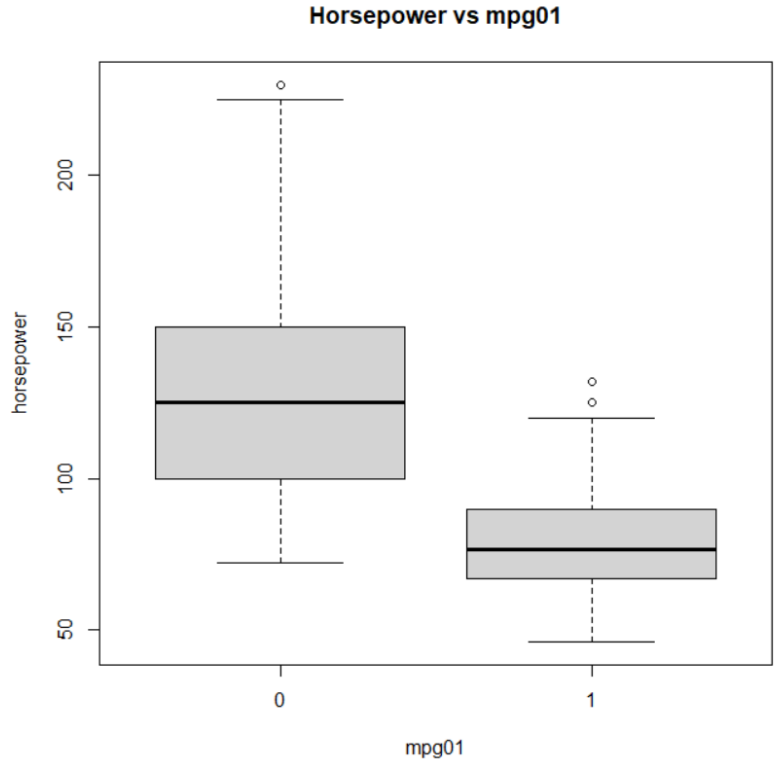
```
                mpg cylinders displacement horsepower weight acceleration  year
mpg            1.00     -0.78         -0.81      -0.78  -0.83         0.42  0.58
cylinders     -0.78      1.00          0.95       0.84   0.90        -0.50 -0.35
displacement  -0.81      0.95          1.00       0.90   0.93        -0.54 -0.37
horsepower    -0.78      0.84          0.90       1.00   0.86        -0.69 -0.42
weight        -0.83      0.90          0.93       0.86   1.00        -0.42 -0.31
acceleration   0.42     -0.50         -0.54      -0.69  -0.42         1.00  0.29
year           0.58     -0.35         -0.37      -0.42  -0.31         0.29  1.00
origin         0.57     -0.57         -0.61      -0.46  -0.59         0.21  0.18
mpg01          0.84     -0.76         -0.75      -0.67  -0.76         0.35  0.43
              origin mpg01
mpg             0.57  0.84
cylinders      -0.57 -0.76
displacement   -0.61 -0.75
horsepower     -0.46 -0.67
weight         -0.59 -0.76
acceleration    0.21  0.35
year            0.18  0.43
origin          1.00  0.51
mpg01           0.51  1.00
```

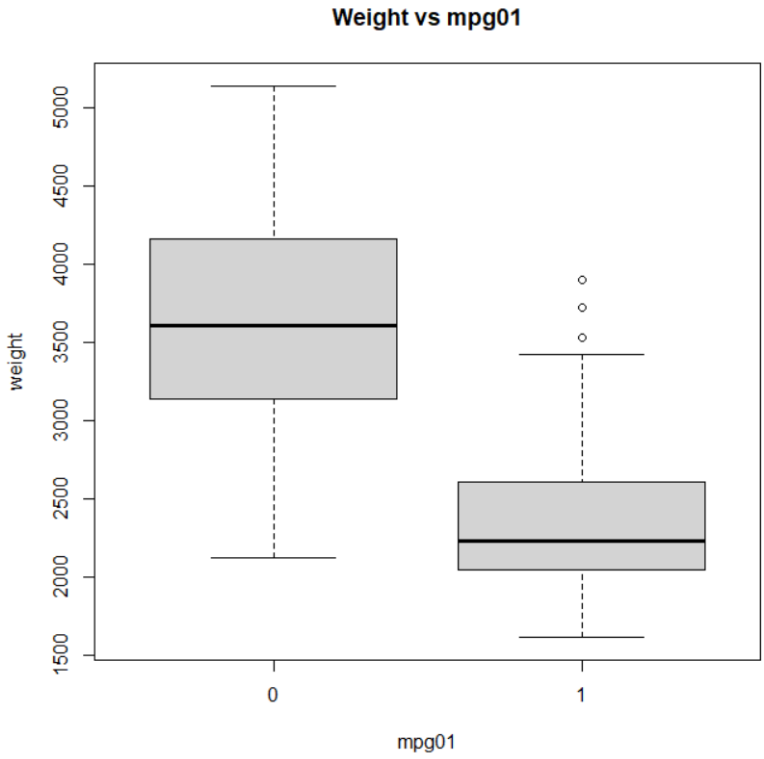


З числових даних бачимо, що є досить сильна залежність між mpg01 та змінними cylinders, displacement, horsepower та weight, але тут варто наголосити що це є від’ємна кореляція. Тобто далі будемо досліджувати детальніше вищезгадані змінні. Це було виконано з використанням функції boxplot().




Displacement vs mpg01
displacement
100
200
300
400
0
1
mpg01


Horsepower vs mpg01
horsepower
50
100
150
200
0
1
mpg01



Всі наведені вище коробчасті діаграми вказують, що справді є наявна від'ємна кореляція між змінними cylinders, displacement, horsepower, weight та нашою якісною змінною mpg01. Хоча варто сказати, що для всіх цих змінних існують значення при mpg01 = 1, які є більші за середнє при mpg01 = 0 (тобто коли mpg містить значення нижче його медіани), проте таких небагато.

2.3 Розбито дані на начальний та тестовий набори. Це було виконано взявши посортовану вибірку year та розділивши на дві половини, де перша – це тренувальний набір, а друга відповідно тестовий.

```
train = (year >= min(year) & year <= min(year) + (max(year) - min(year)) %/% 2)

autos.train = autos[train, ]
autos.test = autos[!train, ]
mpg01.test = mpg01[!train]
```

2.4 Застосовано лінійний дискримінантний аналіз на навчальних даних. В ролі предикторів було взято найбільш залежні від mpg01 змінні, тобто cylinders, displacement, horsepower та weight.

```
Call:
lda(mpg01 ~ cylinders + weight + displacement + horsepower, data = autos,
    subset = train)

Prior probabilities of groups:
        0         1
0.6635514 0.3364486

Group means:
  cylinders   weight displacement horsepower
0  6.830986 3672.106     282.0775  134.02817
1  4.055556 2228.125     105.5347   78.45833

Coefficients of linear discriminants:
                    LD1
cylinders    -0.3505402344
weight       -0.0009436055
displacement -0.0057104148
horsepower    0.0145160830

   mpg01.test
      0   1
  0  48  13
  1   6 111
[1] "Test error rate:  0.106741573033708"
```

Як бачимо, тільки третина навчальних даних відповідає тому, що mpg містить значення вище своєї медіани, тобто вибірка не є ідеально розподілена. Також цікаво наголосити, що коефіцієнт лінійного дискримінанту для змінної cylinders становить -0.35, що значною мірою буде впливати на правило для прийняття рішень щодо прогнозування. Тестова помилка отриманої моделі є 10%, при чому вона не сильно відрізняється в залежності чи значення mpg є нижче за медіану чи навпаки.

2.5 Застосовано квадратичний дискримінантний аналіз на навчальних даних. В ролі предикторів було взято найбільш залежні від mpg01 змінні.

```
Call:
qda(mpg01 ~ cylinders + weight + displacement + horsepower, data = autos,
    subset = train)

Prior probabilities of groups:
        0         1
0.6635514 0.3364486

Group means:
  cylinders   weight displacement horsepower
0  6.830986 3672.106     282.0775  134.02817
1  4.055556 2228.125     105.5347   78.45833

   mpg01.test
      0   1
  0  50  20
  1   4 104
[1] "Test error rate:  0.134831460674157"
```

Тестова помилка цієї моделі становить 13.5%, що є гіршим показником в порівнянні з LQA. Але тут як бачимо з матриці помилок прогнозування відбується дуже точно для mpg01=0, а саме тестова помилка становить (4/54)% = 7.4%, що звісно також означає про гірші показники прогнозування для значень змінної mpg01, які дорівнюють одиниці.

2.6 Застосовано логістичну регресію на навчальних даних. В ролі предикторів було взято найбільш залежні від mpg01 змінні.

```
Call:
glm(formula = mpg01 ~ cylinders + weight + displacement + horsepower,
    family = binomial, data = autos, subset = train)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.33108  -0.14028  -0.00528   0.23930   2.06076

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  11.894724   2.834852   4.196 2.72e-05 ***
cylinders     0.282914   0.711223   0.398   0.6908
weight       -0.002338   0.001198  -1.951   0.0511 .
displacement -0.027274   0.016790  -1.624   0.1043
horsepower   -0.035785   0.025278  -1.416   0.1569
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 273.343  on 213  degrees of freedom
Residual deviance:  85.446  on 209  degrees of freedom
AIC: 95.446

Number of Fisher Scoring iterations: 8

         mpg01.test
pred.glm  0  1
       0 52 37
       1  2 87
[1] "Test error rate:  0.219101123595506"
```

З огляду на наведені вище дані, бачимо що найменше p-value відповідає змінній weight, тобто кореляція є суттєва. Змінна cylinders виділяється за рахунок того, що є наявний позитивний коефіцієнт біля цієї змінної, однак через велике p-value неможливо говорити про зв’язок між залежною змінною в даній моделі. Всі інші коефіцієнти вказують на від’ємну кореляцію із залежною змінною. Тестова помилка становить 21.9%, що є найгіршим результатом прогнозування серед вищезгаданих моделей, при чому точність прогнозування для mpg01=0 сильно зростає, а саме (2/54)% = 3.7%.

2.7 Застосовано метод К-найближчих сусідів з різними значеннями для К на навчальних даних. Нижче наведений код для формування перших двох

аргументів для функції прогнозування. Тобто за допомогою функції cbind() утворюється матриці з предикторами, які пов'язані чисто з навчальними даними та тестовими.

```
train.X = cbind(cylinders, weight, displacement, horsepower)[train, ]
test.X = cbind(cylinders, weight, displacement, horsepower)[!train, ]
mpg01.train = mpg01[train]
set.seed(1)
```

```
        mpg01.test
pred.knn  0  1
       0 51 30
       1  3 94
[1] "Test error rate:  0.185393258426966"
         mpg01.test
pred.knn2   0   1
        0  51  24
        1   3 100
[1] "Test error rate:  0.151685393258427"
         mpg01.test
pred.knn3  0  1
        0 51 29
        1  3 95
[1] "Test error rate:  0.179775280898876"
```

В наступних результатах подані матриці помилок для моделей K-найближчих сусідів відповідно із значеннями K= 1, 5, 15. Тобто для значення K=5 бачимо найкращу тестову помилку, а саме 15.2%, що є гіршим результатом ніж у моделях LDA та QDA. З огляду на те, що більшість навчальних даних відповідає тому, що mpg містить значення нижче своєї медіани, то і знову бачимо що дуже висока точність для прогнозування значень, де mpg01=0, а саме тестова помилка становить (3/54)% = 5.6%.

3. Написання функцій.

3.1 Створено функцію Power(), що виводить результат піднесення 2 до 3-ої степені.

```r
Power = function() {2^3}
print(paste("raising 2 to the 3rd power: ", Power()))
```

```
"raising 2 to the 3rd power:  8"
```

3.2 Запрограмовано нову функцію Power2(), яка дозволяє передавати будь-які два числа, x і a, і виводить значення x^a.

```r
print(paste("x: 5, a: 3 -> power: ", Power2(5, 3)))
```

```
"x: 5, a: 3 -> power:  125"
```

3.3 Для демонстрації виконання функції Power2(), написано нижче наведений цикл з використання згенерованої вибірки значень для аргументів x та a.

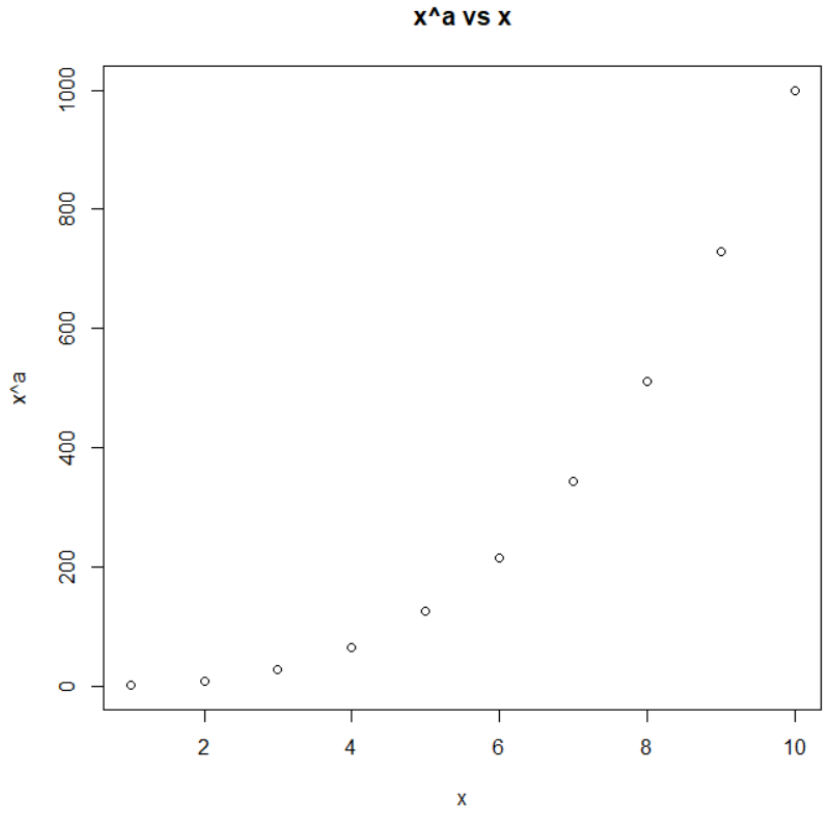
```r
for (x in sample(1:25, 3)) {
  a = sample(1:10, 1)
  print(paste("x: ", x, " a:", a, " -> power: ", Power2(x, a)))
}
```

```
"x:  18  a: 2  -> power:  324"
"x:  17  a: 2  -> power:  289"
"x:  16  a: 9  -> power:  68719476736"
```

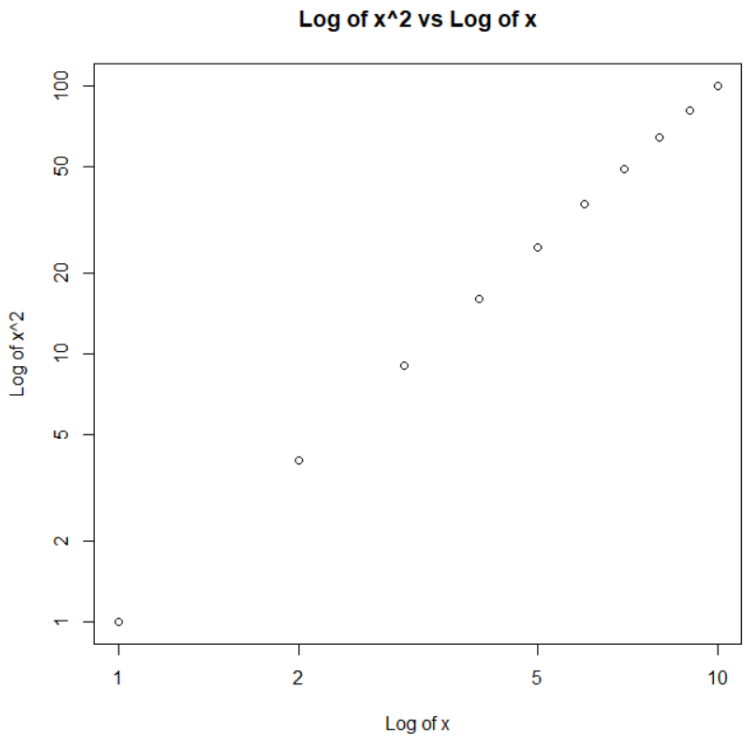
3.4 Написано нову функцію Power3(), яка фактично повертає результат x^a як об'єкт R, а не просто друкує його на екран.

```r
Power3 = function(x, a) {
    result = x^a
    return(result)
}
power_res_3_3 = Power3(3, 3)
```

3.5 Використовуючи функцію Power3(), побудовано графік $f(x) = x^2$. Як вибірку значень для осі абсцис взято діапазон цілих чисел від 1 до 10. Вісь ординат відповідно відображає x^2.

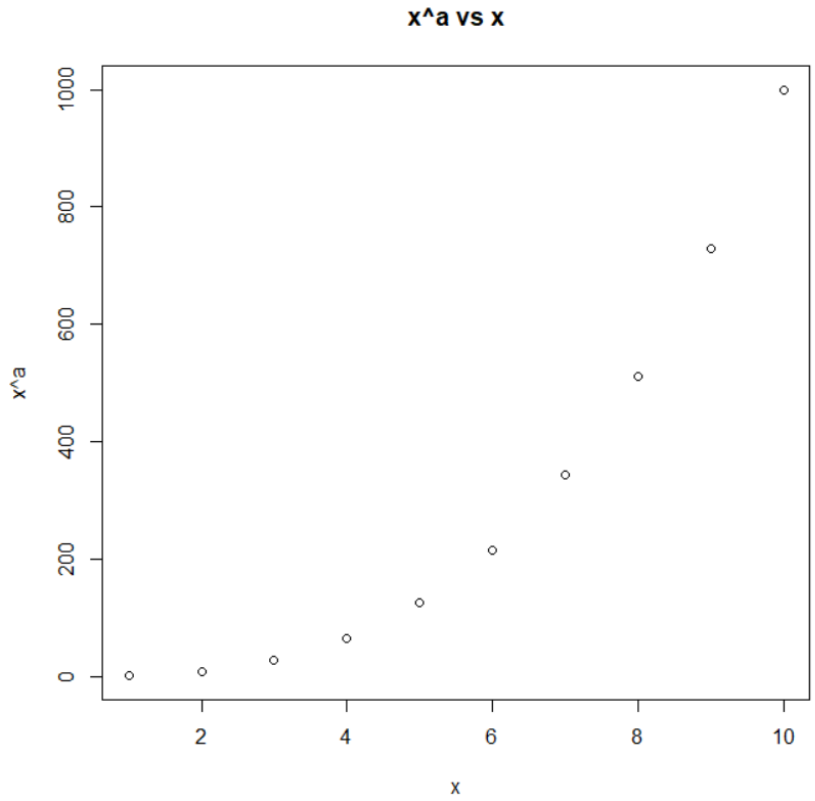


Також розглянув можливість відображення обох осей в логарифмічній шкалі використовуючи аргумент log=”xy” у функції plot().



3.6 Запрограмовано функцію PlotPower(), яка дозволяє будувати графік функції x^a для фіксованого a та для діапазону значень x. Виконано це завдяки використанню функції plot() та її аргументу для значення осі ординат як виклик функції Power3(x, a).

```
PlotPower = function(x, a) {
  plot(x, Power3(x, a), xlab = "x", ylab = "x^a", main = "x^a vs x")
}
```



4. Модель класифікації для передбачення у вибраному районі рівня злочинності більшого чи меншого за медіану на основі даних Boston.

Для початку нам потрібно визначити бінарну змінну crim01, яка містить 1, якщо crim містить значення вище медіани, і 0, якщо crim містить значення нижче його медіани. А також створено єдиний набір даних , що містить як crim01, так і інші змінні з датасету Boston.

Далі необхідно розбити дані на начальний та тестовий набори. Це було виконано взявши вибірку crim та розділивши на дві половини, де перша – це тренувальний набір, а друга відповідно тестовий.

Нижче також наведений код для дослідження логістичної регресії. В ролі предикторів було взято всі змінні, окрім crim01, crim, а також zn та rad для уникнення застережень про ідеальне розділення одиниць та нулів в target-змінній.

```r
train = (crim %in% crim[1:(length(crim) %/% 2)])

Boston.train = Boston[train, ]
Boston.test = Boston[!train, ]
crim01.test = crim01[!train]

fit.glm = glm(crim01 ~. - crim01 - crim - zn - rad,

 data = Boston, family = binomial, subset = train)

probs = predict(fit.glm, Boston.test, type = "response")

pred.glm = rep(0, length(probs))

pred.glm[probs > 0.5] = 1

print(table(pred.glm, crim01.test))

print(paste("Test error rate: ", mean(pred.glm != crim01.test)))
```

```
        crim01.test
pred.glm   0   1
       0  69  16
       1  21 147
[1] "Test error rate:  0.146245059288538"
```

Результат показує, що тестова помилка отриманої моделі є 14.6%, при чому з матриці помилок бачимо, що точність прогнозування у вибраному районі рівня злочинності більшого за медіану зростає, а саме помилка є (16/(147+16))% = 9.8%.

Розглянемо також моделі лінійного та квадратичного дискримінантного аналізу.

```
   crim01.test
      0   1
  0  81  24
  1   9 139
[1] "Test error rate:  0.130434782608696"

   crim01.test
      0   1
  0  82 100
  1   8  63
[1] "Test error rate:  0.426877470355731"
```

Верхня матриця помилок відноситься до LDA, нижня відповідно до QDA. Бачимо очевидну різницю, оскільки тестова помилка для LDA становить 13%, що робить цю модель також більш придатною в порівнянні з моделлю логістичної регресії для нашої задачі прогнозування. З іншого боку бачимо жахливу тестову помилку для моделі QDA, при чому знову розглядається сильний контраст у прогнозуванні, оскільки для рівня злочинності більшого за медіану набагато більше неправильних прогнозувань, тоді як для рівня злочинності меншого за медіану точність є більша ніж навіть у моделі LDA.

Далі розглянемо прогнозування з використанням моделі KNN. Нижче наведений код для формування перших двох аргументів для функції прогнозування. Тобто за допомогою функції cbind() утворюється матриці з предикторами, які пов’язані чисто з навчальними даними та тестовими.

```
train.X = cbind(indus, chas, nox, rm, age, dis,
 tax, ptratio, black, lstat, medv)[train, ]
test.X = cbind(indus, chas, nox, rm, age, dis,
 tax, ptratio, black, lstat, medv)[!train, ]
crim01.train = crim01[train]
set.seed(1)
```

```
        crim01.test
pred.knn   0   1
       0  85 115
       1   5  48
[1] "Test error rate:  0.474308300395257"
         crim01.test
pred.knn2   0   1
        0  85  42
        1   5 121
[1] "Test error rate:  0.185770750988142"
         crim01.test
pred.knn3   0   1
        0  85  23
        1   5 140
[1] "Test error rate:  0.110671936758893"
```

В наступних результатах подані матриці помилок для моделей K-найближчих сусідів відповідно із значеннями K= 1, 5, 15. Тобто для значення K=15 бачимо найкращу тестову помилку, а саме 11%, це найкращий результат прогнозування серед всіх попередньо розглянутих моделей. Найгірший результат прогнозування є для моделі з K=1, але це не означає, що є якась залежність між кількістю K та точністю, оскільки вже для значень K=100 результат був найгірший і складав точність майже 50%.