Aprendizaje de Máquina práctica 04

Pablo Díaz - 30343 | Kevin Huerta - 30502 | Diego Zuazo - 30046 | Gerardo Hernandez - 29902

***Abstract*— Comprender y utilizar las funciones básicas en el lenguaje de programación R y Python para realizar la regresión logística y KNN.**

I. INTRODUCCIÓN

Este reporte tiene como objetivo explicar las metodologías aplicadas en los ejercicios resueltos de la práctica número 4, a través del uso de dos lenguajes de programación: Python y R. Donde el enfoque principal que tiene esta práctica fue hacer uso de la metodos de clasificacion en ambos lenguajes. Antes de entrar a la metodología, explicaremos algunos fundamentos que tuvimos que tener en cuenta para desarrollar nuestros algoritmos de resolución.

II. FUNDAMENTOS

En problemas de clasificacion, a diferencia de obtener una respuesta cualitativa como lo es en problemas de regresion, se quiere obtener una respuesta cuantitativa o categorica, con el fin de efectuar una clasificacion a la observacion (variables de entrada). Existen varias tecnicas de clasificacion, donde las mas comunes son regresion logistica, analisis lineal discriminante, y K vecinos mas cercanos.

A. Regresion Logistica

Regresion Logistica es un metodo de aprendizaje supervisado, que es utilizado con el proposito de modelar la probabilidad de pertenencia a una dicha clase. Claramente obteniendo respuetas con valores dentro del rango 0 y 1. El cual retoma la implementacion de un modelo lienal y lo aplica a una funcion de probabilidad. Dicha funcion es una funcion:

$$p(x) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}} = \frac{1}{1+e^{-(\beta_0+\beta_1 X)}}$$

Como se puede apreciar, dicha funcion de probabilidad usa un modelo lineal como exponente. La manera en que se logra reducir el error, es encontrando las 'Betas' optimas, donde preferiblemente ya no es utilizando el metodo de reducir el criterio de los minimos cuadrados. Para encontrar las 'Betas' optimas que generen el menor error posible, se opta mejor por un metodo llamado "maximum likelihood" por el principal motivo que tiene mejores propiedades estadisticas. La formula matematica que representa a este metodo es el siguiente:

$$l(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1-p(x_{i'}))$$

Donde el objetivo con este metodo es encontrar β_i que maximice dicha funcion, para posteriormente utilizar dichas constantes para construir nuestor modelo de regresion logistica.

B. KNN

KNN es un metodo de aprendizaje supervisado para resolver el problema de clasificación. La idea general de este método es 'ver' o 'conocer' cuales son tus K vecinos mas cercanos, para determinar a que clase pertenece el vecino nuevo o vecino desconocido. Por lo cual se puede intuir que este método se limita al K ser obligatoriamente impar.

C. Analisis Lineal Discriminante

En este metodo consiste modelar una distribucion de los predictores X separadamente para cada una de las clases que van a ser las respuestas Y. Todo esto para posteriormente usar el teorema de Bayes para voletear estas estimaciones para la probabilidad $Pr(Y = k|X = x)$. Cuando dichas distribuciones se asumen ser normales, tiene una gran semenjanza con la regresion logistica. Por lo cual primeramente se tiene que retomar el teorema de Bayes, donde lo que se busca con esto es asociar predictores aleatoriamente que pertenezcan a la clase k-th para formar una funcion de densidad probabilistica dentonada por:

$$f_k(x) = Pr(X = x|Y = k)$$

$$Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\Sigma_{l=1}^{K} \pi_l f_l(x)}$$

Donde la funcion de densidad en analisis lineal discriminante se asume que es una funcion normal o una funcion Gausiana.

$$f_k(x) = \frac{1}{\sqrt{2\pi\sigma_k}} exp(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2)$$

Por lo cual al hacer la coneccion con la manera en que se calcula la probabilidad en el teorema de Bayes, la funcion de probilidad queda de la siguiente manera:

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi\sigma_k}} exp(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2)}{\Sigma_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi\sigma_k}} exp(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2)}$$

III. METODOLOGÍA

La metodología del equipo fue la misma que en veces anteriores, nos dividimos en dos grupos para la implementación del código, un grupo de R y un grupo de Python. Como mencionamos en la práctica anterior, los equipos se alternaron de lenguaje para practicar. Cada ejercicio tuvo su nivel de dificultad, tuvimos que trabajar en equipo completo para encontrar el enfoque adecuada para algunas actividades. De forma mas puntual, la metodología fue la siguiente:

A. *Ejercicio número 1*

En el ejercicio 1 se hizo uso de un conjunto de datos llamado Smarket , este conjuinto de datos pertenece a la libreria ISLR. Se uso La función cor () produce una matriz que contiene todos los pares correlaciones entre los predictores en un conjunto de datos

B. *Ejercicio número 2*

Para este ejercicio , ajustamos un modelo de regresion logistica para predicir 'Direccion' usando de Lag1 hasta Lag5 y Volume. Se hizo uso de la funcion glm() se ajusta a modelos lineales generalizados. La sintaxis generalizada de la función glm () es similar a la de lm (), excepto que debemos pasar en modelo lineal el argumento familia = binomio para decirle a R que ejecute una regresión logística en lugar de algún otro tipo de modelo lineal generalizado.

C. *Ejercicio número 3*

Para el ejercicio 3 se uso LDA (Análisis discriminante lineal) en los datos de Smarket, para ajustar un modelo LDA se usa la funcion lda() esta funcion pertenece a la libreria MASS. Este modelo se ajusto usando solo las observaciones hechas antes del 2005

D. *Ejercicio número 4*

En el ejercicio numero 4 ajustamos un modelo QDA (Análisis discriminante cuadrático) igual que el ejercicio anterior se uso el conjunto de datos Smarket. Usamos la funcion gda() que igualmente pertenece a la libreria de MASS.

E. *Ejercicio número 5*

En este ejercicio realizamos KNN usando la funcion knn() , que pertenece a la libreria de clases. Esta funcion es diferente a las otras fuciones de ajustes que se han visto hasta ahora. La funcion knn() forma su prediccion solo utilizando un commando que requiere cuatro parametros. Tambien se uso la funcion cbind() para enlazar Lag1 y Lag2 en dos matrices.

F. *Ejercicio número 6*

Para el ultimo ejercicio aplicamos el enfoque dd KNN al conjuntos de datos Caravan que pertenece a la libreria de ISLR. Este conjunto de datos incluye 85 predictores que miden características demográficas de 5.822 personas

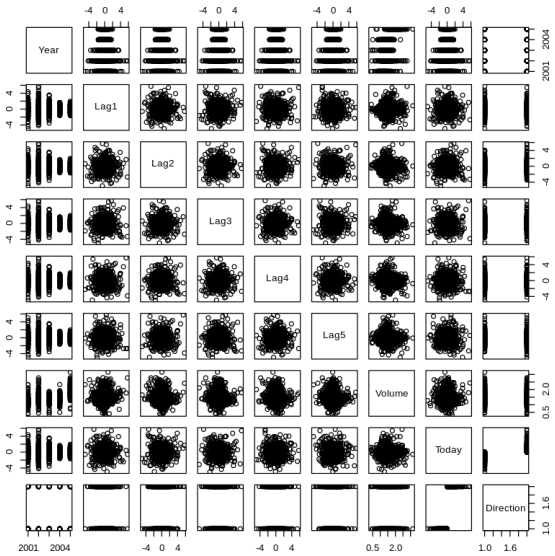
IV. RESULTADOS

Al igual que en las practicas pasadas, es complicado mostrar los datos de una forma digerible. Lo que haremos es mostrar los resultados mas relevantes obtenidos en las actividades en el orden tal y como lo arroja nuestro programa. En el caso de R, se muestran todos los resultados de R que especifica el libro y en el mismo orden. En el de Python también están en el orden que especifica el libro, pero hay algunos que no se pueden mostrar de la misma forma o con el mismo detalle debido a las restricciones que tiene Python en comparación con R. De igual forma, hay algunos pasos en la actividad que demuestran como se arrojan errores y después se arreglan, por lo que no incluiremos esos resultados ya que no son finales.

A. *Ejercicios en R:*

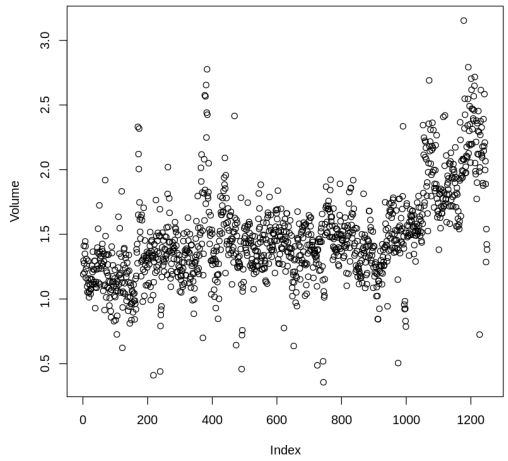
1) Actividad 1- Los resultados obtenidos para la Actividad 1 fueron los siguientes:

```
'Year' · 'Lag1' · 'Lag2' · 'Lag3' · 'Lag4' · 'Lag5' · 'Volume' · 'Today' · 'Direction'
1250 · 9
      Year          Lag1                Lag2                Lag3          
 Min.   :2001   Min.   :-4.922000   Min.   :-4.922000   Min.   :-4.922000  
 1st Qu.:2002   1st Qu.:-0.639500   1st Qu.:-0.639500   1st Qu.:-0.640000  
 Median :2003   Median : 0.039000   Median : 0.039000   Median : 0.038500  
 Mean   :2003   Mean   : 0.003834   Mean   : 0.003919   Mean   : 0.001716  
 3rd Qu.:2004   3rd Qu.: 0.596750   3rd Qu.: 0.596750   3rd Qu.: 0.596750  
 Max.   :2005   Max.   : 5.733000   Max.   : 5.733000   Max.   : 5.733000  
      Lag4                Lag5              Volume           Today          
 Min.   :-4.922000   Min.   :-4.92200   Min.   :0.3561   Min.   :-4.922000  
 1st Qu.:-0.640000   1st Qu.:-0.64000   1st Qu.:1.2574   1st Qu.:-0.639500  
 Median : 0.038500   Median : 0.03850   Median :1.4229   Median : 0.038500  
 Mean   : 0.001636   Mean   : 0.00561   Mean   :1.4783   Mean   : 0.003138  
 3rd Qu.: 0.596750   3rd Qu.: 0.59700   3rd Qu.:1.6417   3rd Qu.: 0.596750  
 Max.   : 5.733000   Max.   : 5.73300   Max.   :3.1525   Max.   : 5.733000  
 Direction
 Down:602 
 Up  :648 
```



A matrix: 8 × 8 of type dbl

	Year	Lag1	Lag2	Lag3	Lag4	Lag5	Volume	Today
Year	1.00000000	0.029699649	0.030596422	0.033194581	0.035688718	0.029787995	0.53900647	0.030095229
Lag1	0.02969965	1.000000000	-0.026294328	-0.010803402	-0.002985911	-0.005674606	0.04090991	-0.026155045
Lag2	0.03059642	-0.026294328	1.000000000	-0.025896670	-0.010853533	-0.003557949	-0.04338321	-0.010250033
Lag3	0.03319458	-0.010803402	-0.025896670	1.000000000	-0.024051036	-0.018808338	-0.04182369	-0.002447647
Lag4	0.03568872	-0.002985911	-0.010853533	-0.024051036	1.000000000	-0.027083641	-0.04841425	-0.006899527
Lag5	0.02978799	-0.005674606	-0.003557949	-0.018808338	-0.027083641	1.000000000	-0.02200231	-0.034860083
Volume	0.53900647	0.040909908	-0.043383215	-0.041823686	-0.048414246	-0.022002315	1.00000000	0.014591823
Today	0.03009523	-0.026155045	-0.010250033	-0.002447647	-0.006899527	-0.034860083	0.01459182	1.000000000



2) Actividad 2- Los resultados obtenidos para la Actividad 2 fueron los siguientes:

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

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.446  -1.203   1.065   1.145   1.326

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.126000   0.240736  -0.523    0.601
Lag1        -0.073074   0.050167  -1.457    0.145
Lag2        -0.042301   0.050086  -0.845    0.398
Lag3         0.011085   0.049939   0.222    0.824
Lag4         0.009359   0.049974   0.187    0.851
Lag5         0.010313   0.049511   0.208    0.835
Volume       0.135441   0.158360   0.855    0.392

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1731.2  on 1249  degrees of freedom
Residual deviance: 1727.6  on 1243  degrees of freedom
AIC: 1741.6

Number of Fisher Scoring iterations: 3
```

```
 (Intercept)         Lag1         Lag2         Lag3         Lag4         Lag5
-0.126000257 -0.073073746 -0.042301344  0.011085108  0.009358938  0.010313068
      Volume
 0.135440659
```

A matrix: 7 × 4 of type dbl

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-0.126000257	0.24073574	-0.5233966	0.6006983
Lag1	-0.073073746	0.05016739	-1.4565986	0.1452272
Lag2	-0.042301344	0.05008605	-0.8445733	0.3983491
Lag3	0.011085108	0.04993854	0.2219750	0.8243333
Lag4	0.009358938	0.04997413	0.1872757	0.8514445
Lag5	0.010313068	0.04951146	0.2082966	0.8349974
Volume	0.135440659	0.15835970	0.8552723	0.3924004

```
(Intercept)        Lag1        Lag2        Lag3        Lag4        Lag5
  0.6006983   0.1452272   0.3983491   0.8243333   0.8514445   0.8349974
     Volume
  0.3924004
```

```
       1        2        3        4        5        6        7        8
0.5070841 0.4814679 0.4811388 0.5152224 0.5107812 0.5069565 0.4926509 0.5092292
       9       10
0.5176135 0.4888378
     Up
Down  0
Up    1
```

```
          Direction
glm.pred Down  Up
    Down  145 141
    Up    457 507
0.5216
0.5216
```

```
252 · 9
```

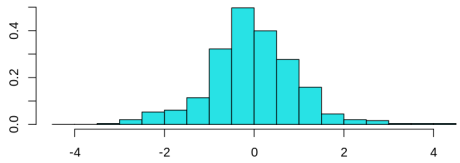
```
          Direction.2005
glm.pred Down Up
    Down   77 97
    Up     34 44
0.480158730158730
0.519841269841270
```

```
          Direction.2005
glm.pred Down  Up
    Down   35  35
    Up     76 106
0.559523809523810
0.582417582417582
```

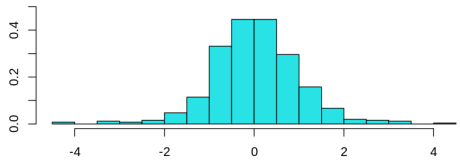
```
1:  0.479146239171912 2:  0.496093872956532
```

3) Actividad 3- Los resultados obtenidos para la Actividad 3 fueron los siguientes:

```
Call:
lda(Direction ~ Lag1 + Lag2, data = Smarket, subset = train)

Prior probabilities of groups:
    Down       Up
0.491984 0.508016

Group means:
            Lag1        Lag2
Down  0.04279022  0.03389409
Up   -0.03954635 -0.03132544

Coefficients of linear discriminants:
            LD1
Lag1 -0.6420190
Lag2 -0.5135293
```



group Down



group Up

```
'class' - 'posterior' - 'x'
```

```
             Direction.2005
lda.class Down  Up
     Down   35  35
     Up     76 106
0.55952380952381
```

```
70
182
```

```
     999      1000      1001      1002      1003      1004      1005      1006
0.4901792 0.4792185 0.4668185 0.4740011 0.4927877 0.4938562 0.4951016 0.4872861
    1007      1008      1009      1010      1011      1012      1013      1014
0.4907013 0.4844026 0.4906963 0.5119988 0.4895152 0.4706761 0.4744593 0.4799583
    1015      1016      1017      1018
0.4935775 0.5030894 0.4978806 0.4886331
Up · Up · Up · Up · Up · Up · Up · Up · Up · Up · Up · Down · Up · Up · Up · Up · Up · Down · Up · Up
▼ Levels:
'Down' · 'Up'
```

```
0
```

4) Actividad 4- Los resultados obtenidos para la Actividad 4 fueron los siguientes:

```
Call:
qda(Direction ~ Lag1 + Lag2, data = Smarket, subset = train)

Prior probabilities of groups:
    Down       Up
0.491984 0.508016

Group means:
            Lag1        Lag2
Down  0.04279022  0.03389409
Up   -0.03954635 -0.03132544
```

```
             Direction.2005
qda.class Down  Up
     Down   30  20
     Up     81 121
0.599206349206349
```

5) Actividad 5- Los resultados obtenidos para la Actividad 5 fueron los siguientes:

```
             Direction.2005
knn.pred Down Up
    Down   43 58
    Up     68 83
0.5
```

```
             Direction.2005
knn.pred Down Up
    Down   48 54
    Up     63 87
0.535714285714286
```

6) Actividad 6- Los resultados obtenidos para la Actividad 6 fueron los siguientes:

```
5822 - 86
No:        5474 Yes:        348
0.0597732737890759
```

```
165.037847395189
0.164707781931954
1
1
```

```
0.118
0.059
```

```
        test.Y
knn.pred  No Yes
     No  873  50
     Yes  68   9
0.116883116883117
```

```
        test.Y
knn.pred  No Yes
     No  920  54
     Yes  21   5
0.192307692307692
        test.Y
knn.pred  No Yes
     No  930  55
     Yes  11   4
0.266666666666667
```

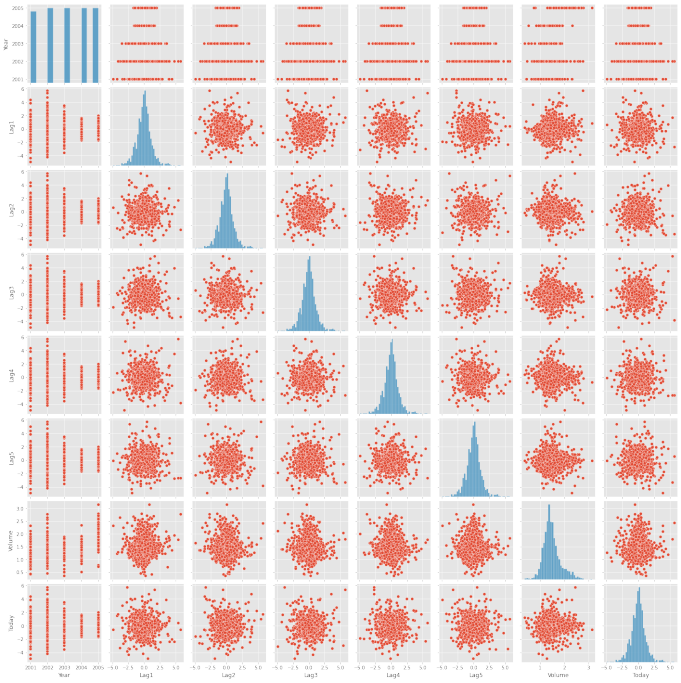
```
Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
        test.Y
glm.pred  No Yes
     No  934  59
     Yes   7   0
        test.Y
glm.pred  No Yes
     No  919  48
     Yes  22  11
0.333333333333333
```

B. Ejercicios en Python:

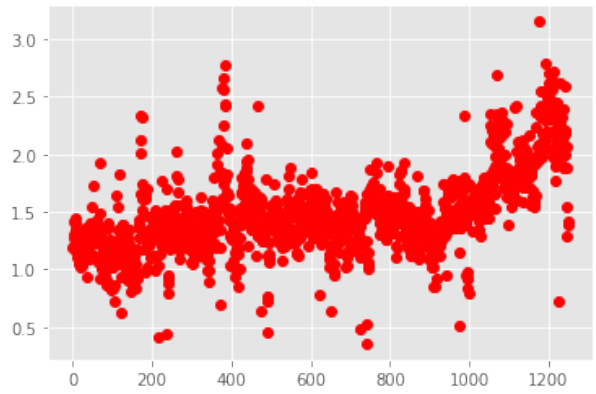
1) Actividad 1- Los resultados obtenidos para la Actividad 1 fueron los siguientes:

```
['Year', 'Lag1', 'Lag2', 'Lag3', 'Lag4', 'Lag5', 'Volume', 'Today', 'Direction']
(1250, 9)
```

	Year	Lag1	Lag2	Lag3	Lag4	Lag5	Volume	Today
count	1250.000000	1250.000000	1250.000000	1250.000000	1250.000000	1250.00000	1250.000000	1250.000000
mean	2003.016000	0.003834	0.003919	0.001716	0.001636	0.00561	1.478305	0.003138
std	1.409018	1.136299	1.136280	1.138703	1.138774	1.14755	0.360357	1.136334
min	2001.000000	-4.922000	-4.922000	-4.922000	-4.922000	-4.92200	0.356070	-4.922000
25%	2002.000000	-0.639500	-0.639500	-0.640000	-0.640000	-0.64000	1.257400	-0.639500
50%	2003.000000	0.039000	0.039000	0.038500	0.038500	0.03850	1.422950	0.038500
75%	2004.000000	0.596750	0.596750	0.596750	0.596750	0.59700	1.641675	0.596750
max	2005.000000	5.733000	5.733000	5.733000	5.733000	5.73300	3.152470	5.733000



	Year	Lag1	Lag2	Lag3	Lag4	Lag5	Volume	Today
Year	1.000000	0.029700	0.030596	0.033195	0.035689	0.029788	0.539006	0.030095
Lag1	0.029700	1.000000	-0.026294	-0.010803	-0.002986	-0.005675	0.040910	-0.026155
Lag2	0.030596	-0.026294	1.000000	-0.025897	-0.010854	-0.003558	-0.043383	-0.010250
Lag3	0.033195	-0.010803	-0.025897	1.000000	-0.024051	-0.018808	-0.041824	-0.002448
Lag4	0.035689	-0.002986	-0.010854	-0.024051	1.000000	-0.027084	-0.048414	-0.006900
Lag5	0.029788	-0.005675	-0.003558	-0.018808	-0.027084	1.000000	-0.022002	-0.034860
Volume	0.539006	0.040910	-0.043383	-0.041824	-0.048414	-0.022002	1.000000	0.014592
Today	0.030095	-0.026155	-0.010250	-0.002448	-0.006900	-0.034860	0.014592	1.000000




```
[[145 457]
 [141 507]]
0.5216
0.5216
```

```
(252, 9)
```

2) Actividad 2- Los resultados obtenidos para la Actividad 2 fueron los siguientes:

```
Optimization terminated successfully.
         Current function value: 0.691034
         Iterations 4
                           Logit Regression Results
==============================================================================
Dep. Variable:          Direction[Up]   No. Observations:                 1250
Model:                          Logit   Df Residuals:                     1243
Method:                           MLE   Df Model:                            6
Date:                Thu, 22 Oct 2020   Pseudo R-squ.:                0.002074
Time:                        00:48:12   Log-Likelihood:                -863.79
converged:                       True   LL-Null:                       -865.59
Covariance Type:            nonrobust   LLR p-value:                    0.7319
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.1260      0.241     -0.523      0.601      -0.598       0.346
Lag1          -0.0731      0.050     -1.457      0.145      -0.171       0.025
Lag2          -0.0423      0.050     -0.845      0.398      -0.140       0.056
Lag3           0.0111      0.050      0.222      0.824      -0.087       0.109
Lag4           0.0094      0.050      0.187      0.851      -0.089       0.107
Lag5           0.0103      0.050      0.208      0.835      -0.087       0.107
Volume         0.1354      0.158      0.855      0.392      -0.175       0.446
==============================================================================
Null Deviance = 1731.1747691164987
Residual Deviance = 1727.5840942032346
AIC: 1741.58
```

```
      Down  Up
Down    77  34
Up      97  44
0.4801587301587302
0.5198412698412698
```

```
Optimization terminated successfully.
         Current function value: 0.691034
         Iterations 4
Intercept   -0.126000
Lag1        -0.073074
Lag2        -0.042301
Lag3         0.011085
Lag4         0.009359
Lag5         0.010313
Volume       0.135441
dtype: float64
```

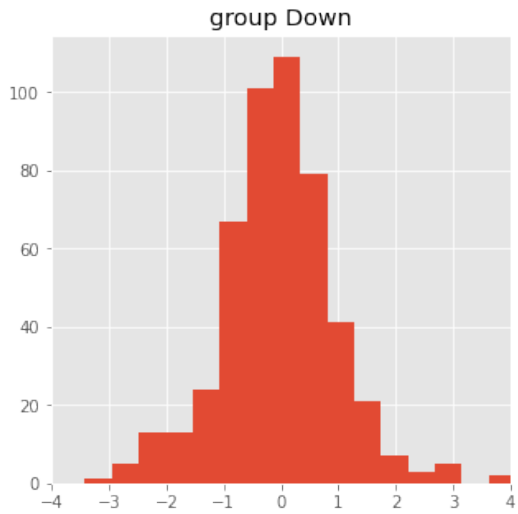
```
      Down   Up
Down    35   76
Up      35  106
0.5595238095238095
0.5824175824175825
```

3) Actividad 3- Los resultados obtenidos para la Actividad 3 fueron los siguientes:

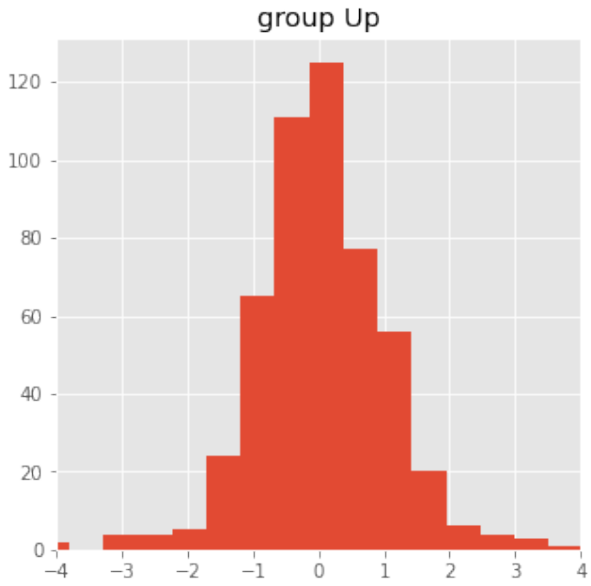
```
Optimization terminated successfully.
         Current function value: 0.691034
         Iterations 4
[0.50708413 0.48146788 0.48113883 0.51522236 0.51078116 0.50695646
 0.49265087 0.50922916 0.51761353 0.48883778]
```

```
Prior probabilities of groups:
      Down        Up
  0.491984  0.508016

Group means:
            Lag1      Lag2
Down   0.042790  0.033894
Up    -0.039546 -0.031325

Coefficients of linear discriminants:
            LD1
Lag1 -0.642019
Lag2 -0.513529
```

```
Optimization terminated successfully.
         Current function value: 0.691034
         Iterations 4
```




```
      Down       Up
0   0.490179  0.509821
1   0.479218  0.520782
2   0.466818  0.533182
3   0.474001  0.525999
4   0.492788  0.507212
5   0.493856  0.506144
6   0.495102  0.504898
7   0.487286  0.512714
8   0.490701  0.509299
9   0.484403  0.515597
10  0.490696  0.509304
11  0.511999  0.488001
12  0.489515  0.510485
13  0.470676  0.529324
14  0.474459  0.525541
15  0.479958  0.520042
16  0.493578  0.506422
17  0.503089  0.496911
18  0.497881  0.502119
19  0.488633  0.511367
['Up' 'Up' 'Up' 'Up' 'Up' 'Up' 'Up' 'Up' 'Up' 'Up' 'Up' 'Down' 'Up' 'Up'
 'Up' 'Up' 'Up' 'Down' 'Up' 'Up']
```




```
0
```

4) Actividad 4- Los resultados obtenidos para la Actividad 4 fueron los siguientes:

```
Prior probabilities of groups:
      Down        Up
  0.491984  0.508016

Group means:
            Lag1      Lag2
Down   0.042790  0.033894
Up    -0.039546 -0.031325
```

```
      Down   Up
Down    35   76
Up      35  106
0.5595238095238095
```

```
      Down   Up
Down    30   81
Up      20  121
0.5992063492063492
```

```
70
182
```

5) Actividad 5- Los resultados obtenidos para la Actividad 5 fueron los siguientes:

```
      Down  Up
Down    43  68
Up      58  83
0.5
```

```
      Down  Up
Down    48  63
Up      55  86
0.5317460317460317
```

6) Actividad 6- Los resultados obtenidos para la Actividad 6 fueron los siguientes:

```
(5822, 86)
No     5474
Yes     348
Name: Purchase, dtype: int64
0.05977327378907592
```

```
      No Yes
No   873  68
Yes   50   9
0.11688311688311688
```

```
      No Yes
No   921  20
Yes   54   5
0.19230769230769232

      No Yes
No   930  11
Yes   55   4
0.26666666666666666
```

```
      No Yes
No   934   7
Yes   59   0

      No Yes
No   919  22
Yes   48  11
0.3333333333333333
```

V. CONCLUSIONES

Esta practica resulto muy útil para ayudarnos a entender otros modelos lineales mas complejos y como aplicarlos. Dado que los ejercicios que nos enfrentábamos en la practica consistía en el la aplicaciones de estos. Entender como aplicar estos modelos es fundamental para el aprendizaje estadístico. Mediante esta practica no solo reforzamos nuestro conocimientos en modelos lineales pero si no también seguimos ampliando nuestro conocimientos en Python y en R.