

# Regularización del Modelo de Predicción de Salarios

*Jose Javier Martíb García*

*17/10/2019*

En esta práctica lo que he hecho ha sido regularizar el modelo de predicción de salarios con el cual ya había trabajado en la práctica anterior con la diferencia de que en el modelo de este informe he mantenido más variables que las que elegí en el modelo anterior pasando de 4 a 17 variables explicativas. A continuación voy a explicar que son cada una de dichas variables como trabajo de documentación del data set.

\*Age: es la edad del jugador.

\*G: son los partidos que ha jugado.

\*MP: Son los minutos jugados.

\*PER: ratio de eficiencia del jugador, mide un estandar de producción, el promedio esta en 15.

\*TS\_: es el porcentaje de tiros acertados tanto tiros libres, de dos puntos y triples.

\*X3PAr: Es el porcentaje de intentos desde la linea de 6.75 (triple).

\*FTr: es el porcentaje de intentos desde la linea de personal (tiros libres).

\*TRB\_: es el porcentaje total de rebotes, tanto ofensivos como defensivos que el jugador atrapa mientras esta en pista.

\*AST\_: es el porcentaje de asistencias que un jugador da a sus compañeros cuando estos aciertan en el tiro.

\*STL\_: es el porcentaje de balones robados al jugador contrario.

\*BLK\_: es el porcentaje de tapones.

\*TOV\_: es una estimacion de perdidas cometidas por cada 100 jugadas.

\*USG\_: es el porcentaje de las jugadas en equipo usadas mientras el jugador esta en pista.

\*WS: es el número de victorias estimadas en las que ha participado el jugador (Responsabilidad de Victorias).

\*BPM: posesiones en las que un jugador contribuye por encima de un jugador promedio de la liga traducido al equipo promedio.

\*VORP: posesiones de equipo en las que un jugador contribuye por encima de un reemplazo traducido a un equipo promedio y prorrateado a una temporada de 82 partidos.

\*NBA\_Draft: ronda y posición de la elección en el Draft de un jugador.

```
## Loading required package: tidy
```

```
## Loading required package: Matrix
```

```
##
```

```
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
```

```
##
```

```
## expand, pack, unpack
```

```
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-18

##
## 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

## corrplot 0.84 loaded

## [1] "C:/Users/Equipo/Desktop/CUNEF/Prediccion/Practicas"

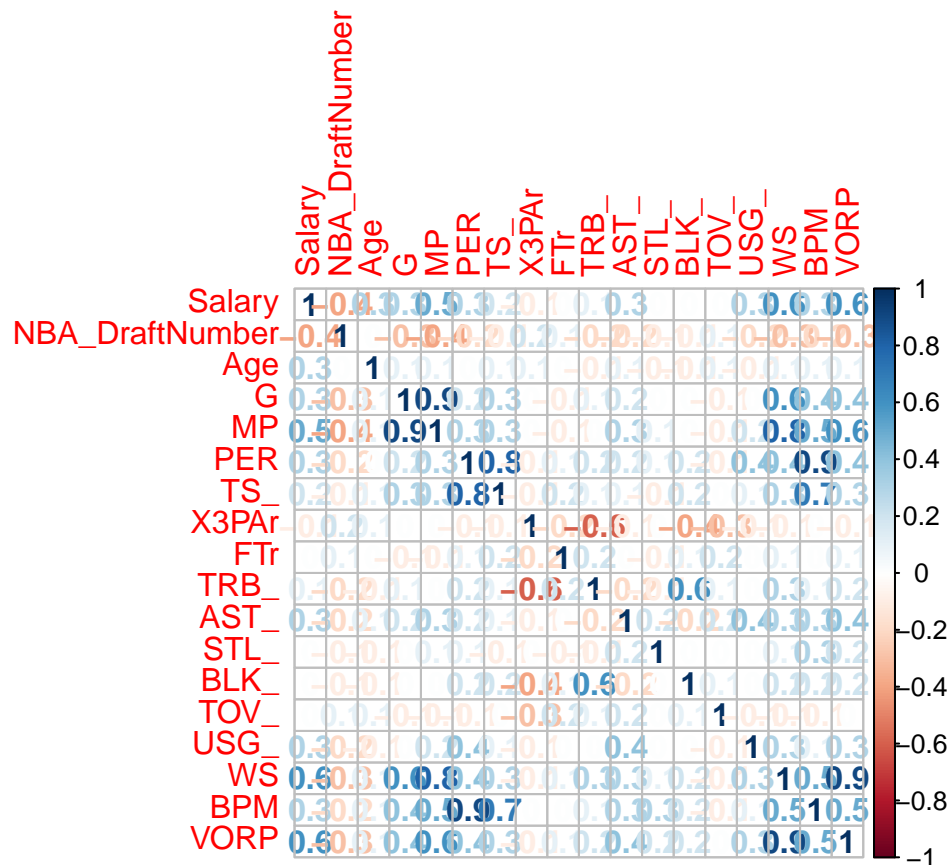
##
## Call:
## lm(formula = Salary ~ Age + AST_ + BLK_ + BPM + FTr + G + MP +
##     NBA_DraftNumber + PER + STL_ + TOV_ + TRB_ + TS_ + USG_ +
##     VORP + WS + X3PAr, data = nba)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14470123 -2956067  -393319   2185155  23002827
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3786227.3  3458768.9  -1.095   0.2742
## Age           523743.1    56031.1   9.347 < 2e-16 ***
## AST_         -12122.7    38890.0  -0.312   0.7554
## BLK_        -115018.1   181426.5  -0.634   0.5264
## BPM           351328.9   169116.5   2.077   0.0383 *
## FTr           104222.6   856248.2   0.122   0.9032
## G          -154502.2    24581.0  -6.285 7.53e-10 ***
## MP              5286.9     870.6   6.073 2.61e-09 ***
## NBA_DraftNumber -63434.4    12497.7  -5.076 5.59e-07 ***
## PER          -191543.8    104924.3  -1.826   0.0686 .
## STL_        -461043.9    296930.3  -1.553   0.1212
## TOV_        -15866.1     47756.5  -0.332   0.7399
## TRB_          100596.9     73866.3   1.362   0.1739
## TS_         -940901.6   4254534.6  -0.221   0.8251
## USG_          155804.0     73386.7   2.123   0.0343 *
## VORP           557730.1    611489.4   0.912   0.3622
## WS            541766.2    333364.7   1.625   0.1048
## X3PAr        -1515110.1   1493599.7  -1.014   0.3109
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5090000 on 465 degrees of freedom
## Multiple R-squared:  0.5434, Adjusted R-squared:  0.5267
## F-statistic: 32.56 on 17 and 465 DF,  p-value: < 2.2e-16
```

Aqui lo que he hecho ha sido quitar las variables que no quiero usar.

En este punto he generado la matriz de correlación para ver como de correlacionadas están mis variables.

```
##          Salary NBA_DraftNumber Age      G      MP      PER      TS_ X3PAr
## Salary          1.00          -0.38  0.34  0.29  0.51  0.27  0.17 -0.07
## NBA_DraftNumber -0.38          1.00  0.01 -0.30 -0.37 -0.22 -0.14  0.19
## Age              0.34          0.01  1.00  0.07  0.07  0.01  0.08  0.08
## G                0.29          -0.30  0.07  1.00  0.87  0.23  0.29 -0.03
## MP               0.51          -0.37  0.07  0.87  1.00  0.29  0.26  0.00
## PER              0.27          -0.22  0.01  0.23  0.29  1.00  0.78 -0.15
## TS_              0.17          -0.14  0.08  0.29  0.26  0.78  1.00 -0.09
## X3PAr            -0.07          0.19  0.08 -0.03  0.00 -0.15 -0.09  1.00
## FTr              0.02           0.05 -0.04 -0.09 -0.06  0.11  0.19 -0.25
## TRB_             0.14          -0.16 -0.05  0.07  0.02  0.22  0.12 -0.55
## AST_             0.26          -0.15  0.08  0.18  0.31  0.23  0.06 -0.06
## STL_             0.03          -0.06 -0.06  0.02  0.06  0.09 -0.09 -0.05
## BLK_             0.04          -0.12 -0.06  0.05 -0.02  0.24  0.18 -0.41
## TOV_            -0.04          0.12  0.06 -0.13 -0.13 -0.14  0.04 -0.25
## USG_             0.29          -0.21 -0.06  0.02  0.24  0.39  0.11 -0.08
## WS               0.59          -0.34  0.11  0.63  0.78  0.43  0.34 -0.13
## BPM              0.31          -0.25  0.05  0.44  0.45  0.86  0.74 -0.01
## VORP             0.57          -0.28  0.08  0.40  0.61  0.40  0.26 -0.09
##          FTr  TRB_  AST_  STL_  BLK_  TOV_  USG_  WS  BPM
## Salary          0.02  0.14  0.26  0.03  0.04 -0.04  0.29  0.59  0.31
## NBA_DraftNumber 0.05 -0.16 -0.15 -0.06 -0.12  0.12 -0.21 -0.34 -0.25
## Age            -0.04 -0.05  0.08 -0.06 -0.06  0.06 -0.06  0.11  0.05
## G              -0.09  0.07  0.18  0.02  0.05 -0.13  0.02  0.63  0.44
## MP             -0.06  0.02  0.31  0.06 -0.02 -0.13  0.24  0.78  0.45
## PER            0.11  0.22  0.23  0.09  0.24 -0.14  0.39  0.43  0.86
## TS_            0.19  0.12  0.06 -0.09  0.18  0.04  0.11  0.34  0.74
## X3PAr          -0.25 -0.55 -0.06 -0.05 -0.41 -0.25 -0.08 -0.13 -0.01
## FTr            1.00  0.19  0.00 -0.09  0.10  0.18  0.01  0.06  0.05
## TRB_           0.19  1.00 -0.23 -0.10  0.57  0.11  0.00  0.26  0.14
## AST_           0.00 -0.23  1.00  0.21 -0.21  0.21  0.40  0.30  0.28
## STL_          -0.09 -0.10  0.21  1.00 -0.02 -0.03  0.00  0.08  0.26
## BLK_           0.10  0.57 -0.21 -0.02  1.00  0.08 -0.01  0.15  0.20
## TOV_           0.18  0.11  0.21 -0.03  0.08  1.00 -0.08 -0.11 -0.14
## USG_           0.01  0.00  0.40  0.00 -0.01 -0.08  1.00  0.29  0.10
## WS             0.06  0.26  0.30  0.08  0.15 -0.11  0.29  1.00  0.53
## BPM            0.05  0.14  0.28  0.26  0.20 -0.14  0.10  0.53  1.00
## VORP           0.08  0.25  0.38  0.16  0.17 -0.02  0.32  0.91  0.50
##          VORP
## Salary          0.57
## NBA_DraftNumber -0.28
## Age              0.08
## G                0.40
## MP               0.61
## PER              0.40
## TS_              0.26
## X3PAr            -0.09
## FTr              0.08
## TRB_             0.25
## AST_             0.38
## STL_             0.16
```

```
## BLK_      0.17
## TOV_     -0.02
## USG_      0.32
## WS        0.91
## BPM       0.50
## VORP      1.00
```



Al analizar la matriz de correlación podemos observar que la variable MP está muy correlacionada con la variable G, al igual que BPM lo está con PER y a su vez WS con VORP. Esto quiere decir que los valores de cada una de ellas varían sistemáticamente con respecto a los valores de la que hemos considerado aquí su pareja, es decir existe una relación directa. Como en este caso lo que queremos ver son qué variables son las más afectadas a la hora de predecir el salario, cabe señalar que el Salario está correlacionado positivamente con las variables WS y VORP en un 0.6 de coeficiente de correlación y con la variable MP en un 0.5 lo que nos quiere decir que al aumentar dichas variables el Salario aumenta proporcionalmente.

Al contrario ocurre con la variable NBA\_Draft que respecto al salario existe una correlación negativa y eso es muy lógico ya que a posteriores rondas de selección en el Draft el salario disminuye ya que no es lo mismo ser elegido en primera ronda que en rondas posteriores y más aun cuando un jugador es elegido como primero en la primera ronda, el cual es considerado el mejor “universitario” del año.

En este punto he construido los training.

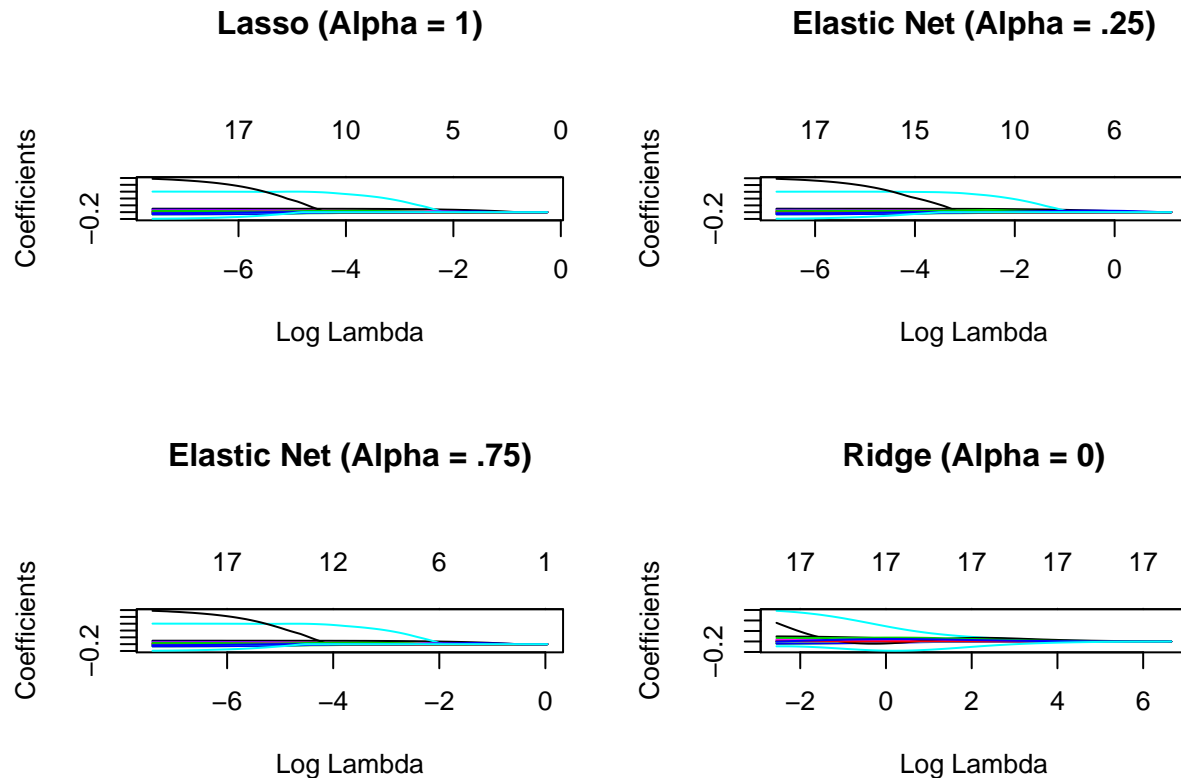
La finalidad de usar los training ha sido porque nos sirve para ajustar el modelo. Gracias al método de Cross-Validation hemos generado los training set y el test de prueba con el que evaluaremos la capacidad predictiva mediante el error de predicción.

Con esta matriz lo que vemos es que el número de observaciones ha sido reducido con respecto a la base de datos original ya que el “train” ha reducido la muestra.

```
## [1] 340 17
```

A partir de aquí me he dedicado a usar el método de regularización llamado Elastic Nets. Dicha técnica selecciona variables automáticamente y supera la eficacia del método Lasso. Es muy útil cuando el número de estimadores es mayor que el número de observaciones.

En el paso siguiente lo que tenemos que tener en cuenta es que cuanto mayor sea Lambda mayor es la penalización en los coeficientes de regresión estimados en función de dicho parámetro. También debemos tener en cuenta que lo más óptimo es elegir el valor más pequeño de Lambda con el cual se estabilizan dichos coeficientes.



Llegados a este punto, mediante técnicas de Cross-Validation he elegido que el parámetro K sea 10 (aunque también se puede utilizar  $k=5$ ) esto quiere decir que el conjunto de datos se ha dividido en 10 subconjuntos de tamaño similar para hacer más fácil el análisis.

Con el siguiente bucle se han ido asignando los valores que faltaban en `nba_grid`

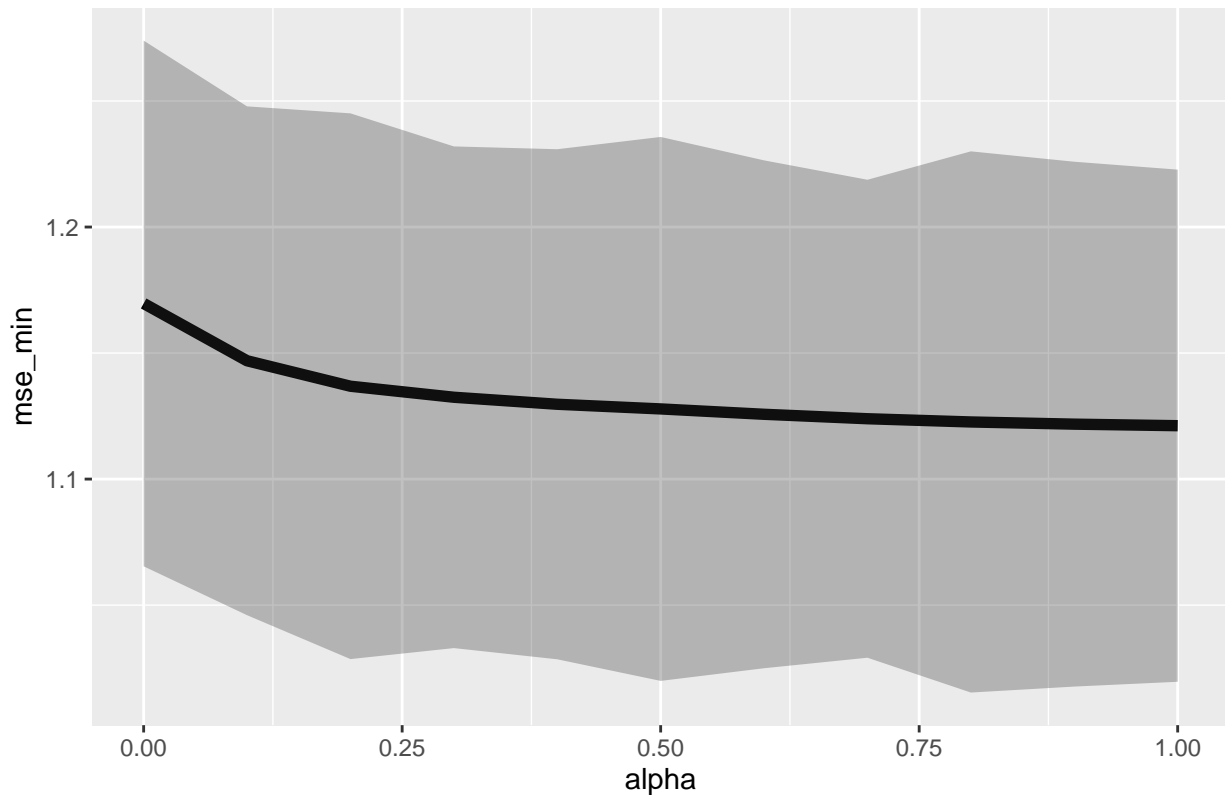
```
## # A tibble: 11 x 5
##   alpha mse_min mse_1se lambda_min lambda_1se
##   <dbl>   <dbl>   <dbl>     <dbl>     <dbl>
## 1 0       1.17    1.27    0.180    1.39
## 2 0.1     1.15    1.25    0.188    0.915
## 3 0.2     1.14    1.25    0.136    0.663
## 4 0.3     1.13    1.23    0.110    0.485
## 5 0.4     1.13    1.23    0.0990   0.400
## 6 0.5     1.13    1.24    0.0869   0.351
## 7 0.6     1.13    1.23    0.0724   0.292
```

```
## 8 0.7 1.12 1.22 0.0621 0.251
## 9 0.8 1.12 1.23 0.0543 0.241
## 10 0.9 1.12 1.23 0.0483 0.214
## 11 1 1.12 1.22 0.0434 0.193
```

Con esto vemos que para un alpha de coeficiente 1 asigna el valor minimo de lambda que en este caso es 0.434, lo cual es un valor muy pequeño y de esa manera la penalización es mínima.

El glmnet es un de estimación denominado Modelo Lineal en el cual los estimadores están generalizados.

### MSE $\pm$ one standard error



Con el glmnet vemos el coeficiente de lambda que es el mismo que vimos en la tabla para un alpha de valor 1 que generamos a traves del bucle for y a su vez podemos ver la bondad de ajuste del modelo con el test de Cramer-Von Mises, el cual me ha dado un valor de error medio de validación cruzada de 1.136829 que es el valor de la media de los valores de error de cada uno de mis diez K.

```
## [1] 1.139033
```

```
## $lambda
```

```
## [1] 0.7771308896 0.7080927233 0.6451877174 0.5878710188 0.5356461778
## [6] 0.4880608477 0.4447028672 0.4051966901 0.3692001330 0.3364014108
## [11] 0.3065164366 0.2792863612 0.2544753306 0.2318684437 0.2112698902
## [16] 0.1925012555 0.1753999745 0.1598179242 0.1456201403 0.1326836484
## [21] 0.1208963988 0.1101562959 0.1003703142 0.0914536921 0.0833291982
## [26] 0.0759264619 0.0691813643 0.0630354826 0.0574355840 0.0523331650
## [31] 0.0476840309 0.0434479130 0.0395881202 0.0360712207 0.0328667529
## [36] 0.0299469611 0.0272865556 0.0248624932 0.0226537779 0.0206412787
## [41] 0.0188075644 0.0171367522 0.0156143704 0.0142272327 0.0129633246
```

```

## [46] 0.0118116986 0.0107623798 0.0098062797 0.0089351168 0.0081413457
## [51] 0.0074180909 0.0067590882 0.0061586294 0.0056115138 0.0051130023
## [56] 0.0046587773 0.0042449044 0.0038677988 0.0035241942 0.0032111145
## [61] 0.0029258479 0.0026659237 0.0024290904 0.0022132967 0.0020166735
## [66] 0.0018375178 0.0016742778 0.0015255395 0.0013900148 0.0012665297
## [71] 0.0011540147 0.0010514952 0.0009580833 0.0008729698 0.0007954176
## [76] 0.0007247549 0.0006603696 0.0006017042 0.0005482505 0.0004995454
##
## $cvm
## [1] 2.081950 2.007076 1.918228 1.830362 1.741329 1.667064 1.602322
## [8] 1.532423 1.463611 1.405906 1.357722 1.317543 1.283873 1.256135
## [15] 1.233595 1.214758 1.197788 1.182731 1.169999 1.159857 1.152284
## [22] 1.147503 1.144468 1.142172 1.140623 1.139570 1.139276 1.139151
## [29] 1.139033 1.139094 1.139269 1.139633 1.141093 1.142483 1.143218
## [36] 1.145557 1.149085 1.154310 1.160357 1.166377 1.172820 1.179273
## [43] 1.185832 1.193342 1.201156 1.209038 1.217042 1.224812 1.232649
## [50] 1.240070 1.246845 1.252791 1.258366 1.263548 1.268422 1.272884
## [57] 1.277011 1.280873 1.284553 1.288104 1.291410 1.294444 1.297269
## [64] 1.299821 1.302183 1.304377 1.306378 1.308241 1.309911 1.311414
## [71] 1.312803 1.314045 1.315237 1.316399 1.317435 1.318371 1.319200
## [78] 1.320034 1.320713 1.321311
##
## $cvstd
## [1] 0.2121784 0.2128822 0.2023441 0.1931578 0.1834823 0.1749840 0.1676276
## [8] 0.1608169 0.1550831 0.1499913 0.1454940 0.1415644 0.1381590 0.1352236
## [15] 0.1327821 0.1310287 0.1297513 0.1287838 0.1280358 0.1274302 0.1269698
## [22] 0.1269885 0.1273760 0.1278256 0.1283091 0.1287081 0.1290316 0.1293322
## [29] 0.1296608 0.1299386 0.1298972 0.1296397 0.1294721 0.1293854 0.1293378
## [36] 0.1292258 0.1291410 0.1291719 0.1294259 0.1296606 0.1299234 0.1304081
## [43] 0.1311957 0.1323253 0.1334151 0.1344627 0.1353320 0.1360374 0.1366945
## [50] 0.1374254 0.1381026 0.1387164 0.1394267 0.1402231 0.1410287 0.1418042
## [57] 0.1425354 0.1431928 0.1438250 0.1444525 0.1450527 0.1456146 0.1461638
## [64] 0.1466800 0.1471616 0.1476179 0.1480374 0.1484322 0.1487827 0.1490956
## [71] 0.1493837 0.1496350 0.1499092 0.1501249 0.1503770 0.1505769 0.1507500
## [78] 0.1509386 0.1510827 0.1512191
##
## $cvup
## [1] 2.294128 2.219959 2.120572 2.023520 1.924811 1.842048 1.769949
## [8] 1.693240 1.618694 1.555897 1.503216 1.459107 1.422032 1.391359
## [15] 1.366377 1.345787 1.327539 1.311515 1.298035 1.287288 1.279254
## [22] 1.274491 1.271844 1.269998 1.268932 1.268278 1.268307 1.268483
## [29] 1.268694 1.269032 1.269166 1.269272 1.270565 1.271869 1.272556
## [36] 1.274783 1.278226 1.283482 1.289783 1.296038 1.302743 1.309681
## [43] 1.317028 1.325667 1.334571 1.343501 1.352374 1.360849 1.369343
## [50] 1.377496 1.384948 1.391508 1.397792 1.403771 1.409451 1.414688
## [57] 1.419546 1.424066 1.428378 1.432556 1.436463 1.440058 1.443433
## [64] 1.446501 1.449345 1.451995 1.454415 1.456673 1.458694 1.460510
## [71] 1.462186 1.463680 1.465146 1.466524 1.467812 1.468948 1.469950
## [78] 1.470973 1.471796 1.472530
##
## $cvlo
## [1] 1.869771 1.794194 1.715884 1.637204 1.557847 1.492080 1.434694
## [8] 1.371606 1.308528 1.255915 1.212228 1.175978 1.145714 1.120912
## [15] 1.100813 1.083729 1.068036 1.053948 1.041963 1.032427 1.025314

```

```

## [22] 1.020514 1.017092 1.014347 1.012314 1.010862 1.010244 1.009819
## [29] 1.009372 1.009155 1.009371 1.009993 1.011620 1.013098 1.013880
## [36] 1.016331 1.019944 1.025138 1.030931 1.036716 1.042896 1.048865
## [43] 1.054637 1.061016 1.067741 1.074575 1.081710 1.088774 1.095954
## [50] 1.102645 1.108743 1.114075 1.118939 1.123325 1.127393 1.131080
## [57] 1.134475 1.137681 1.140728 1.143651 1.146357 1.148829 1.151105
## [64] 1.153141 1.155022 1.156759 1.158341 1.159809 1.161128 1.162319
## [71] 1.163419 1.164410 1.165328 1.166274 1.167058 1.167794 1.168450
## [78] 1.169095 1.169630 1.170092
##
## $nzzero
##  s0  s1  s2  s3  s4  s5  s6  s7  s8  s9 s10 s11 s12 s13 s14 s15 s16 s17
##   0   1   1   2   2   2   2   3   3   4   4   4   4   4   4   5   5   5
## s18 s19 s20 s21 s22 s23 s24 s25 s26 s27 s28 s29 s30 s31 s32 s33 s34 s35
##   5   5   5   6   6   7   7   6   6   7   7   7   7   7   7   7   8   8
## s36 s37 s38 s39 s40 s41 s42 s43 s44 s45 s46 s47 s48 s49 s50 s51 s52 s53
##   8   8   9   9   9  10  11  11  12  13  13  13  14  14  15  15  16  16
## s54 s55 s56 s57 s58 s59 s60 s61 s62 s63 s64 s65 s66 s67 s68 s69 s70 s71
##  16  16  17  17  17  17  17  17  17  17  17  17  17  17  17  17  17  17
## s72 s73 s74 s75 s76 s77 s78 s79
##  17  17  17  17  17  17  17  17
##
## $name
##               mse
## "Mean-Squared Error"
##
## $glmnet.fit
##
## Call:  glmnet(x = nba_train_x, y = nba_train_y, alpha = 1)
##
##           Df      %Dev   Lambda
## [1,]    0 0.00000 0.7771000
## [2,]    1 0.04922 0.7081000
## [3,]    1 0.09008 0.6452000
## [4,]    2 0.13600 0.5879000
## [5,]    2 0.17850 0.5356000
## [6,]    2 0.21380 0.4881000
## [7,]    2 0.24310 0.4447000
## [8,]    3 0.27890 0.4052000
## [9,]    3 0.31210 0.3692000
## [10,]   4 0.33970 0.3364000
## [11,]   4 0.36310 0.3065000
## [12,]   4 0.38260 0.2793000
## [13,]   4 0.39880 0.2545000
## [14,]   4 0.41230 0.2319000
## [15,]   4 0.42340 0.2113000
## [16,]   5 0.43400 0.1925000
## [17,]   5 0.44390 0.1754000
## [18,]   5 0.45210 0.1598000
## [19,]   5 0.45890 0.1456000
## [20,]   5 0.46460 0.1327000
## [21,]   5 0.46930 0.1209000
## [22,]   6 0.47330 0.1102000
## [23,]   6 0.47720 0.1004000

```



```

## [24,] 7 0.48050 0.0914500
## [25,] 7 0.48340 0.0833300
## [26,] 6 0.48590 0.0759300
## [27,] 6 0.48770 0.0691800
## [28,] 7 0.48980 0.0630400
## [29,] 7 0.49200 0.0574400
## [30,] 7 0.49390 0.0523300
## [31,] 7 0.49550 0.0476800
## [32,] 7 0.49680 0.0434500
## [33,] 7 0.49780 0.0395900
## [34,] 7 0.49870 0.0360700
## [35,] 8 0.49960 0.0328700
## [36,] 8 0.50050 0.0299500
## [37,] 8 0.50130 0.0272900
## [38,] 8 0.50190 0.0248600
## [39,] 9 0.50240 0.0226500
## [40,] 9 0.50320 0.0206400
## [41,] 9 0.50410 0.0188100
## [42,] 10 0.50480 0.0171400
## [43,] 11 0.50550 0.0156100
## [44,] 11 0.50620 0.0142300
## [45,] 12 0.50680 0.0129600
## [46,] 13 0.50750 0.0118100
## [47,] 13 0.50800 0.0107600
## [48,] 13 0.50850 0.0098060
## [49,] 14 0.50910 0.0089350
## [50,] 14 0.50970 0.0081410
## [51,] 15 0.51020 0.0074180
## [52,] 15 0.51070 0.0067590
## [53,] 16 0.51110 0.0061590
## [54,] 16 0.51160 0.0056120
## [55,] 16 0.51200 0.0051130
## [56,] 16 0.51240 0.0046590
## [57,] 17 0.51270 0.0042450
## [58,] 17 0.51290 0.0038680
## [59,] 17 0.51320 0.0035240
## [60,] 17 0.51340 0.0032110
## [61,] 17 0.51350 0.0029260
## [62,] 17 0.51370 0.0026660
## [63,] 17 0.51380 0.0024290
## [64,] 17 0.51390 0.0022130
## [65,] 17 0.51390 0.0020170
## [66,] 17 0.51400 0.0018380
## [67,] 17 0.51400 0.0016740
## [68,] 17 0.51410 0.0015260
## [69,] 17 0.51410 0.0013900
## [70,] 17 0.51420 0.0012670
## [71,] 17 0.51420 0.0011540
## [72,] 17 0.51420 0.0010510
## [73,] 17 0.51420 0.0009581
## [74,] 17 0.51420 0.0008730
## [75,] 17 0.51420 0.0007954
## [76,] 17 0.51430 0.0007248
## [77,] 17 0.51430 0.0006604

```

```
## [78,] 17 0.51430 0.0006017
## [79,] 17 0.51430 0.0005483
## [80,] 17 0.51430 0.0004995
##
## $lambda.min
## [1] 0.05743558
##
## $lambda.1se
## [1] 0.2318684
##
## attr(,"class")
## [1] "cv.glmnet"

## 18 x 1 sparse Matrix of class "dgCMatrix"
##              1
## (Intercept) 13.5226579807
## Age         0.0473575441
## AST_        .
## BLK_        .
## BPM         .
## FTr         .
## G           .
## MP          0.0005123538
## NBA_DraftNumber -0.0145709863
## PER         .
## STL_        .
## TOV_        .
## TRB_        .
## TS_         .
## USG_        .
## VORP        .
## WS          0.0170151887
## X3PAr       .
```

Con los resultados del modelo y a la vista de los coeficientes obtenidos podemos estimar el salario de un jugador multiplicando las variables de un jugador en concreto por el índice que aparece resultante del modelo Lasso para su variable correspondiente:

$Age = 0.0473575441$   $MP = 0.0005123538$   $NBA\_Draft = -0.0145709863$   $WS = 0.0170151887$

Podemos comprobar que el número de ronda del Draft es negativo por el motivo que explique anteriormente.

Otra cosa que hay que destacar es que los valores están en base logarítmica y para estimar el salario habría que usar una función exponencial.

Con la función predict vamos a predecir valores ajustados y otros coeficientes.

```
## [1] 19.17953

##              1
## 1  16.25343
## 2  20.64744
## 4  17.38302
## 7  16.22746
## 8  17.93406
```

## 12 17.57972  
## 13 20.29548  
## 16 17.20978  
## 18 18.09579  
## 22 18.99538  
## 31 21.40549  
## 36 20.48475  
## 41 18.39351  
## 42 19.46109  
## 46 18.31763  
## 47 17.24007  
## 50 18.85600  
## 53 18.35579  
## 56 17.99288  
## 59 17.98934  
## 60 15.74895  
## 75 18.71837  
## 76 19.03866  
## 77 15.79440  
## 78 21.95866  
## 79 16.60530  
## 81 17.42042  
## 83 16.54004  
## 84 20.30690  
## 85 16.29814  
## 86 16.86422  
## 91 17.93706  
## 95 18.80652  
## 99 21.40179  
## 100 16.86821  
## 108 20.14226  
## 109 18.24302  
## 112 17.55335  
## 115 18.36939  
## 116 17.90726  
## 121 18.45150  
## 123 17.67699  
## 124 17.74020  
## 130 16.55566  
## 135 18.33787  
## 136 16.31036  
## 138 16.41360  
## 139 17.03452  
## 145 18.27093  
## 147 21.44609  
## 151 19.84973  
## 163 18.95748  
## 165 25.10530  
## 166 20.61188  
## 171 15.11911  
## 172 16.45183  
## 176 19.39949  
## 189 13.24007  
## 191 18.32694

## 194 20.53604  
## 195 17.13014  
## 200 19.19802  
## 201 19.66224  
## 202 19.46242  
## 210 18.41964  
## 212 19.18968  
## 215 21.81684  
## 216 18.98723  
## 222 18.92470  
## 223 16.97485  
## 226 17.93107  
## 228 20.85858  
## 229 19.50825  
## 230 19.08062  
## 232 19.11435  
## 233 16.23385  
## 234 20.76594  
## 239 18.68791  
## 247 20.90864  
## 248 18.08783  
## 251 20.94622  
## 253 15.50258  
## 257 18.71329  
## 265 15.38116  
## 268 19.26059  
## 273 22.13409  
## 276 20.15985  
## 284 20.07633  
## 285 15.29661  
## 292 21.64082  
## 299 23.96796  
## 301 16.65385  
## 305 18.09808  
## 309 17.60567  
## 310 22.71306  
## 311 19.40960  
## 312 19.64806  
## 314 19.12251  
## 315 21.00764  
## 320 18.33514  
## 324 17.31511  
## 325 16.49564  
## 327 18.06670  
## 329 20.96151  
## 332 19.71137  
## 336 18.22932  
## 337 19.80724  
## 345 15.13856  
## 352 19.29898  
## 356 17.18992  
## 361 21.24713  
## 362 17.29261  
## 363 20.03513

```

## 366 19.41901
## 367 18.02397
## 368 18.05889
## 378 15.66922
## 380 17.88177
## 386 19.09739
## 392 15.75196
## 394 16.42373
## 404 17.84592
## 409 18.53090
## 411 19.01793
## 412 19.97524
## 416 17.31816
## 421 14.56511
## 424 19.51049
## 425 19.24013
## 431 17.94171
## 433 16.43365
## 439 17.85680
## 443 16.98808
## 449 19.77827
## 450 19.14740
## 461 19.26122
## 465 15.97808
## 474 16.41545
## 477 17.57747
## 480 19.16647
## 481 17.51650
## 482 18.90893
## 485 17.26414

```

Después de esto lo que he visto es el valor de predicción para cada uno de los elementos del `nba_test_X` y al hacer la media de la diferencia de cuadrados con el test y me da el valor cuadrático medio de la predicción.

Aquí lo que he hecho ha sido utilizar el método `Caret` para contrastar los coeficientes respecto al método `Elastic Nets`.

```

## Loading required package: lattice

## glmnet
##
## 340 samples
## 17 predictor
##
## Pre-processing: centered (17), scaled (17)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 306, 305, 306, 307, 307, 305, ...
## Resampling results across tuning parameters:
##
##   alpha  lambda      RMSE      Rsquared    MAE
##   0.1    0.0003590545  1.077315  0.4624347  0.8293374
##   0.1    0.0008294626  1.077315  0.4624347  0.8293374
##   0.1    0.0019161662  1.076984  0.4626954  0.8291452
##   0.1    0.0044265925  1.074907  0.4636620  0.8282941

```

##	0.1	0.0102260031	1.070992	0.4655643	0.8266275
##	0.1	0.0236233941	1.064988	0.4690569	0.8232216
##	0.1	0.0545731057	1.057879	0.4736750	0.8177664
##	0.1	0.1260709554	1.052159	0.4784410	0.8106685
##	0.1	0.2912402656	1.053357	0.4827751	0.8128921
##	0.1	0.6728028042	1.080820	0.4745640	0.8337119
##	0.2	0.0003590545	1.076914	0.4627102	0.8291449
##	0.2	0.0008294626	1.076914	0.4627102	0.8291449
##	0.2	0.0019161662	1.076352	0.4630244	0.8288844
##	0.2	0.0044265925	1.073714	0.4643513	0.8277796
##	0.2	0.0102260031	1.068929	0.4667960	0.8257179
##	0.2	0.0236233941	1.061169	0.4721003	0.8206359
##	0.2	0.0545731057	1.053182	0.4777484	0.8133853
##	0.2	0.1260709554	1.045346	0.4864916	0.8054560
##	0.2	0.2912402656	1.056469	0.4844664	0.8141384
##	0.2	0.6728028042	1.101228	0.4721283	0.8476674
##	0.3	0.0003590545	1.076916	0.4627180	0.8291740
##	0.3	0.0008294626	1.076916	0.4627180	0.8291740
##	0.3	0.0019161662	1.075861	0.4633233	0.8286547
##	0.3	0.0044265925	1.072665	0.4649743	0.8273336
##	0.3	0.0102260031	1.066949	0.4680126	0.8248369
##	0.3	0.0236233941	1.057577	0.4750563	0.8181311
##	0.3	0.0545731057	1.049232	0.4815268	0.8094176
##	0.3	0.1260709554	1.043976	0.4899518	0.8050417
##	0.3	0.2912402656	1.062850	0.4833716	0.8181656
##	0.3	0.6728028042	1.126577	0.4680538	0.8669927
##	0.4	0.0003590545	1.076687	0.4627335	0.8291300
##	0.4	0.0008294626	1.076652	0.4627735	0.8291001
##	0.4	0.0019161662	1.075358	0.4636156	0.8284269
##	0.4	0.0044265925	1.071661	0.4655717	0.8268984
##	0.4	0.0102260031	1.065191	0.4693307	0.8238589
##	0.4	0.0236233941	1.054883	0.4772141	0.8158076
##	0.4	0.0545731057	1.044941	0.4861451	0.8052159
##	0.4	0.1260709554	1.045681	0.4901601	0.8054945
##	0.4	0.2912402656	1.070414	0.4820543	0.8228907
##	0.4	0.6728028042	1.157478	0.4625047	0.8932744
##	0.5	0.0003590545	1.077120	0.4626670	0.8292241
##	0.5	0.0008294626	1.076952	0.4628247	0.8291061
##	0.5	0.0019161662	1.074873	0.4639028	0.8282222
##	0.5	0.0044265925	1.070763	0.4660607	0.8265316
##	0.5	0.0102260031	1.063423	0.4707430	0.8226955
##	0.5	0.0236233941	1.052653	0.4790701	0.8135663
##	0.5	0.0545731057	1.041862	0.4897164	0.8021270
##	0.5	0.1260709554	1.047357	0.4899615	0.8053283
##	0.5	0.2912402656	1.079181	0.4801925	0.8276736
##	0.5	0.6728028042	1.195567	0.4471791	0.9263123
##	0.6	0.0003590545	1.077119	0.4626517	0.8291794
##	0.6	0.0008294626	1.076755	0.4629337	0.8289936
##	0.6	0.0019161662	1.074375	0.4642031	0.8280032
##	0.6	0.0044265925	1.069903	0.4665258	0.8261561
##	0.6	0.0102260031	1.061813	0.4719770	0.8215495
##	0.6	0.0236233941	1.050845	0.4805933	0.8117067
##	0.6	0.0545731057	1.039998	0.4922606	0.8013865
##	0.6	0.1260709554	1.048999	0.4900690	0.8052495

```

## 0.6 0.2912402656 1.088289 0.4783328 0.8327386
## 0.6 0.6728028042 1.239926 0.4129554 0.9664375
## 0.7 0.0003590545 1.077108 0.4626828 0.8291841
## 0.7 0.0008294626 1.076553 0.4630377 0.8289009
## 0.7 0.0019161662 1.073904 0.4644795 0.8277907
## 0.7 0.0044265925 1.069023 0.4670156 0.8257560
## 0.7 0.0102260031 1.060275 0.4731643 0.8205344
## 0.7 0.0236233941 1.049198 0.4821233 0.8099675
## 0.7 0.0545731057 1.039003 0.4940293 0.8010563
## 0.7 0.1260709554 1.051092 0.4896889 0.8060461
## 0.7 0.2912402656 1.097976 0.4769936 0.8389284
## 0.7 0.6728028042 1.279998 0.3786815 1.0011358
## 0.8 0.0003590545 1.077134 0.4627288 0.8291665
## 0.8 0.0008294626 1.076316 0.4631868 0.8287901
## 0.8 0.0019161662 1.073413 0.4647737 0.8275756
## 0.8 0.0044265925 1.068128 0.4675402 0.8253523
## 0.8 0.0102260031 1.058722 0.4743927 0.8194833
## 0.8 0.0236233941 1.047335 0.4840045 0.8082346
## 0.8 0.0545731057 1.038958 0.4947426 0.8011928
## 0.8 0.1260709554 1.053699 0.4889256 0.8074544
## 0.8 0.2912402656 1.109620 0.4743820 0.8482187
## 0.8 0.6728028042 1.314239 0.3582867 1.0258237
## 0.9 0.0003590545 1.077068 0.4628131 0.8290981
## 0.9 0.0008294626 1.076080 0.4633214 0.8286772
## 0.9 0.0019161662 1.072924 0.4650600 0.8273448
## 0.9 0.0044265925 1.067210 0.4680642 0.8249506
## 0.9 0.0102260031 1.057190 0.4756409 0.8184089
## 0.9 0.0236233941 1.045233 0.4861205 0.8065484
## 0.9 0.0545731057 1.039859 0.4945776 0.8019216
## 0.9 0.1260709554 1.056766 0.4878102 0.8095769
## 0.9 0.2912402656 1.123148 0.4699203 0.8608788
## 0.9 0.6728028042 1.351062 0.3210853 1.0521293
## 1.0 0.0003590545 1.077033 0.4628083 0.8291110
## 1.0 0.0008294626 1.075840 0.4634655 0.8285653
## 1.0 0.0019161662 1.072463 0.4653268 0.8271444
## 1.0 0.0044265925 1.066360 0.4686269 0.8245701
## 1.0 0.0102260031 1.055695 0.4768411 0.8171841
## 1.0 0.0236233941 1.043099 0.4882860 0.8046640
## 1.0 0.0545731057 1.041019 0.4939559 0.8027618
## 1.0 0.1260709554 1.060144 0.4865133 0.8123672
## 1.0 0.2912402656 1.138675 0.4628030 0.8757541
## 1.0 0.6728028042 1.383774 0.3006406 1.0758935
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were alpha = 0.8 and lambda
## = 0.05457311.

```

Despues de hacer el metodo Caret (utilice este porque me parecia mas practico que el H2O) he de decir que la lambda que me da es mucho mayor que la que obtengo con el metodo de Elastic Nets en un principio llegue a pensar que ambos modelos harían lo mismo pero a raíz de los resultados, reconozco que me surgen dudas, porque la verdad es que hay mucha diferencia entre ambos valores de lambda.

El alpha optimo para este modelo es 1, el cual es el valor maximo para distribuir la penalizacion. Por tanto me quedare con el metodo de Elastic Nets el cual me produce un modelo con un nivel muy aceptable de

prediccion, por ello acabare diciendo que las variables elegidas predicen bien el salario a pagar a los jugadores de la NBA.

FUENTES:

[([http://rpubs.com/Joaquin\\_AR/406480](http://rpubs.com/Joaquin_AR/406480))[1], (<http://idus.us.es/xmlui/bitstream/handle/11441/43746/Carrasco%20Carrasco%2C%20Mar%C3%ADa%20TFG.pdf?sequence=1&isAllowed=y>)[2], ([http://www.basketball-reference.com/leagues/NBA\\_2018\\_advanced.html](http://www.basketball-reference.com/leagues/NBA_2018_advanced.html))[3]