

PRÁCTICA APRENDIZAJE AUTOMÁTICO

APA 2022-23



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona



Emch Boada, Andrés
Gotanegra Estañol, Miquel

Índice

1. Introducción	3
1.1 Descripción del conjunto de datos	3
1.2 Problemas iniciales	4
2. Preprocesamiento	5
2.1 Selección de características	5
2.2 Valores perdidos	5
2.3 Tratamiento de atributos categóricos	6
2.4 Valores atípicos	7
2.5 Análisis final de nuestro conjunto de datos preprocesados	8
2.6 División entrenamiento-prueba	12
3. Modelos clasificadores lineales/cuadráticos	13
3.1 LDA	13
3.2 QDA	14
3.3 KNN	15
3.4 SVM	17
4. Modelos clasificadores no lineales	18
4.1 MLP	18
4.2 SVM con kernel RBF	19
4.3 Gradient Boosting	19
4.4 VotingClassifier	21
5. Modelo final	22
6. Conclusiones	22
6.1 Acerca del conjunto de datos	22
6.2 Posibles extensiones	23
6.3 Comparación pesos lineales vs no-lineales	23
7. Bibliografía	24

1. Introducción

1.1 Descripción del conjunto de datos

Para desarrollar nuestra práctica el conjunto de datos que hemos escogido es un conjunto de datos sobre las estadísticas de todos los jugadores de la NBA en las temporadas entre los años 1998 y 2022 a pesar de que trabajaremos únicamente con la última temporada 2021-2022. El objetivo de nuestro predictor sería predecir la posición en la que juega (base, escolta, alero, ala-pívot y pívot) un determinado jugador a partir de sus estadísticas (clasificación). Por lo tanto, se trataría de un problema de clasificación. La fuente de nuestro dataset es el siguiente: <https://data.world/etocco/nba-player-stats>.

Hay variedad de tipología en el conjunto de datos, pero nos centraremos en datos continuos y no tanto en variables categóricas, a pesar de que planteemos un problema de clasificación. Los atributos de nuestro dataset serán los siguientes: Concretamente, los datos con los que vamos a trabajar serán los siguientes (un total de 20):

- Age: Edad
- G: Partidos jugados en la temporada
- GS: Partidos jugados formando parte del quinteto inicial
- MP: Minutos jugados por partido
- FG: Tiros por partido
- FGA: Intentos de tiro por partido
- 3P: Triples por partido
- 3PA: Intentos de triple por partido
- 2P: Tiros de dos por partido
- 2PA: Intentos de tiros de dos por partido
- FT: Tiros libres por partido
- FTA: Intentos de tiros libres por partido
- ORB: Rebotes ofensivos por partido
- DRB: Rebotes defensivos por partido
- AST: Asistencias por partido
- STL: Robos por partido
- BLK: Tapones por partido
- TOV: Pérdidas por partido
- PF: Faltas personales por partido
- PTS: Puntos por partido

Nuestro dataset contenía más pero hemos considerado que las anunciadas son las relevantes para nuestra práctica.

1.2 Problemas iniciales

Dado que la variable a predecir es categórica, es obvio que nos enfrentaremos a un problema de clasificación. Uno de los primeros problemas que podemos encontrar con sólo mirar el conjunto de datos sin procesar es que hay algunas variables que no dan información sobre la posición de nuestro jugador, como el equipo donde juega, su nombre y el año de la temporada de los datos promedios.

Además estos datos serían de índole categórica y, a no ser que sea imprescindible, vamos a intentar deshacernos al máximo de todos los datos que no sean continuos. También tenemos que considerar que los datos no son del todo independientes y se deben evaluar con criterio, pues no es lo mismo promediar 20 puntos/partido jugando una media de 15 minutos por partido que jugando 40. Los datos relativos a porcentajes no los vamos a considerar, pues estos se pueden calcular con el resto de atributos numéricos. Teniendo en cuenta las cuestiones que hemos mencionado, procedemos a preprocesar el conjunto de datos.

2. Preprocesamiento

2.1 Selección de características

De todo el conjunto de características hay algunas que creemos que no nos van a ser relevantes, como el equipo, el ranking del jugador (orden alfabético) o su nombre. También vamos a eliminar todas las estadísticas que se puedan derivar de otras columnas, como el porcentaje de tiro o de rebotes.

Además, como hemos escogido usar solo dos temporadas consecutivas, vamos a eliminar la variable Year, ya que no nos aporta prácticamente nada de información.

	Pos	Age	G	GS	MP	FG	FGA	3P	3PA	2P	...	FT	FTA	ORB	DRB	AST	STL	BLK	TOV	PF	PTS
74	PF	21	61	4	12.1	2.0	3.7	0.0	0.0	2.0	...	0.9	1.8	1.2	2.2	0.5	0.3	0.5	0.7	1.5	5.0
75	PG	24	7	0	2.6	0.1	1.1	0.0	0.3	0.1	...	0.0	0.0	0.0	0.4	0.3	0.0	0.0	0.0	0.1	0.3
76	C	27	58	58	27.7	3.3	5.3	0.0	0.1	3.3	...	1.0	2.3	3.7	5.2	1.9	0.9	0.7	1.3	1.9	7.6
77	C	23	64	64	33.5	7.1	12.5	0.0	0.1	7.1	...	4.4	5.5	2.2	6.7	5.4	1.2	1.0	2.6	2.3	18.7
78	C	35	26	23	25.9	5.4	11.4	1.2	3.1	4.2	...	1.6	1.8	0.7	3.8	1.9	0.4	1.1	1.0	1.8	13.5

2.2 Valores perdidos

Hemos hecho la comprobación de los posibles valores perdidos, para encontrar dichos posibles valores, hemos buscado valores '?', ' ' o '-1' y han sido convertidos a NaN. Posteriormente, hemos contado todos los valores equivalentes a NaN:

```
Missing Values
Pos      0
Age      0
G        0
GS       0
MP       0
FG       0
FGA      0
3P       0
3PA      0
2P       0
2PA      0
FT       0
FTA      0
ORB      0
DRB      0
AST      0
STL      0
BLK      0
TOV      0
PF       0
PTS      0
dtype: int64
```

Afortunadamente, comprobamos que ningún atributo contiene datos no válidos.

2.3 Tratamiento de atributos categóricos

Por suerte la única variable categórica que tenemos es la de respuesta. Si tuviéramos diversas temporadas, la variable Year tendría que convertirse en one-hot encoding.

```
SG      360
PF      293
PG      274
SF      273
C       265
SF-SG    8
SG-SF    7
SG-PG    6
C-PF     4
PF-SF    3
SF-PF    3
PG-SG    2
PF-C     1
Name: Pos, dtype: int64
```

Como podemos ver tenemos muchísimas posiciones diferentes en la base de datos, resultado de combinar las 5 posiciones base del baloncesto: Base o Point Guard (PG), Escolta o Shooting guard(SG), Alero o Small forward(SF), Ala-Pivot o Power Forward(PF) y Pivot o Center(C).

Para simplificar el proceso de determinar la posición de los jugadores, ya que algunos se pueden adaptar a diferentes posiciones según el equipo requiera, vamos a asignarle a cada uno su categoría principal.

```
Escolta  373
Ala-Pivot 297
Alero    284
Base     276
Pivot    269
Name: Pos, dtype: int64
```

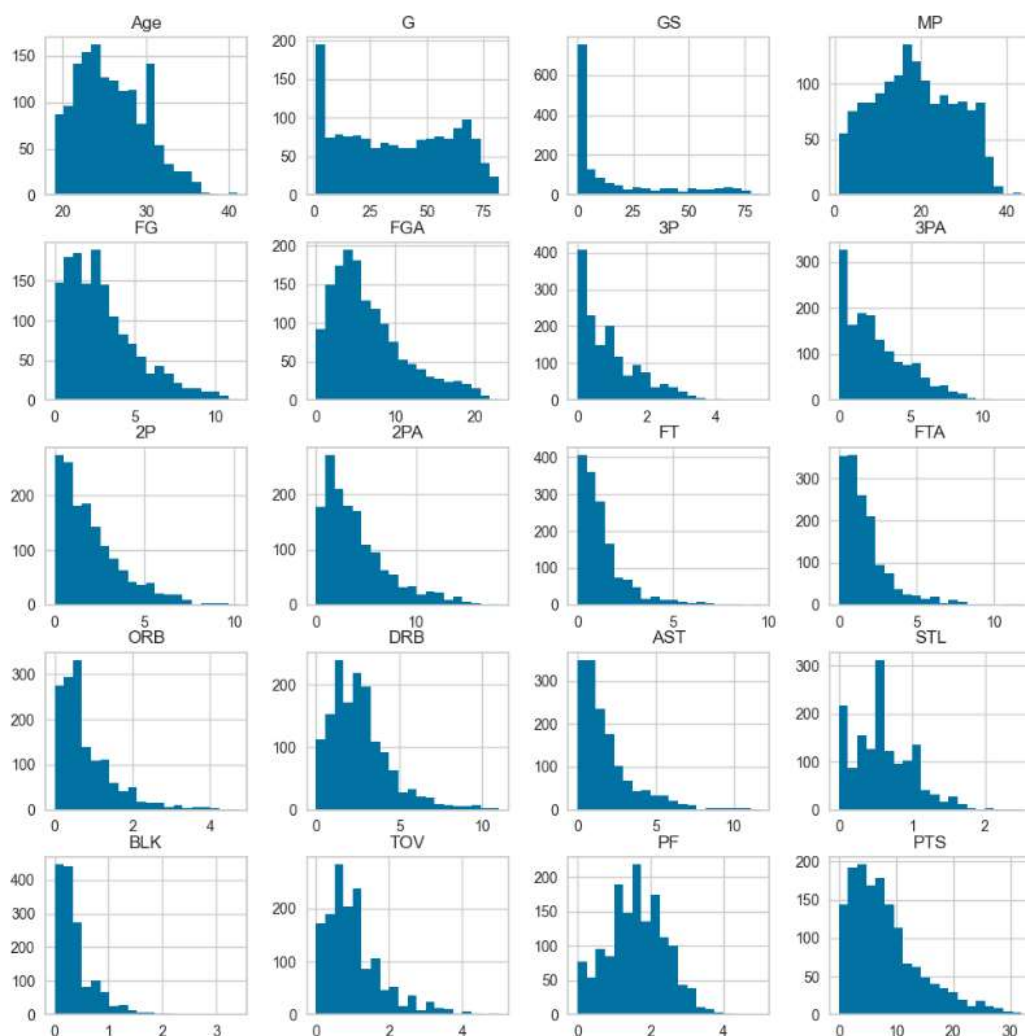
Podemos ver que la variable objetivo está más o menos balanceada entre las distintas categorías.

2.4 Valores atípicos

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Pos	1499	5	Escolta	373	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Age	1499.0	NaN	NaN	NaN	25.958	4.046	19.0	23.0	25.0	29.0	41.0
G	1499.0	NaN	NaN	NaN	36.949	23.878	1.0	15.0	37.0	58.0	82.0
GS	1499.0	NaN	NaN	NaN	16.757	22.832	0.0	0.0	5.0	27.0	82.0
MP	1499.0	NaN	NaN	NaN	18.771	9.443	1.0	11.3	18.2	26.35	43.5
FG	1499.0	NaN	NaN	NaN	3.002	2.256	0.0	1.3	2.5	4.1	11.4
FGA	1499.0	NaN	NaN	NaN	6.638	4.699	0.0	3.2	5.5	9.1	23.0
3P	1499.0	NaN	NaN	NaN	0.909	0.859	0.0	0.2	0.7	1.4	5.3
3PA	1499.0	NaN	NaN	NaN	2.626	2.217	0.0	0.8	2.1	4.0	12.7
2P	1499.0	NaN	NaN	NaN	2.096	1.785	0.0	0.8	1.6	2.9	10.2
2PA	1499.0	NaN	NaN	NaN	4.016	3.256	0.0	1.5	3.2	5.5	18.3
FT	1499.0	NaN	NaN	NaN	1.262	1.295	0.0	0.4	0.9	1.7	9.6
FTA	1499.0	NaN	NaN	NaN	1.646	1.59	0.0	0.6	1.2	2.1	11.8
ORB	1499.0	NaN	NaN	NaN	0.808	0.735	0.0	0.3	0.6	1.1	4.7
DRB	1499.0	NaN	NaN	NaN	2.63	1.802	0.0	1.3	2.3	3.5	11.0
AST	1499.0	NaN	NaN	NaN	1.863	1.83	0.0	0.6	1.3	2.4	11.7
STL	1499.0	NaN	NaN	NaN	0.596	0.41	0.0	0.3	0.5	0.9	2.5
BLK	1499.0	NaN	NaN	NaN	0.383	0.385	0.0	0.1	0.3	0.5	3.4
TOV	1499.0	NaN	NaN	NaN	1.022	0.818	0.0	0.5	0.8	1.3	5.0
PF	1499.0	NaN	NaN	NaN	1.591	0.797	0.0	1.0	1.6	2.1	5.0
PTS	1499.0	NaN	NaN	NaN	8.173	6.225	0.0	3.5	6.8	11.1	32.0

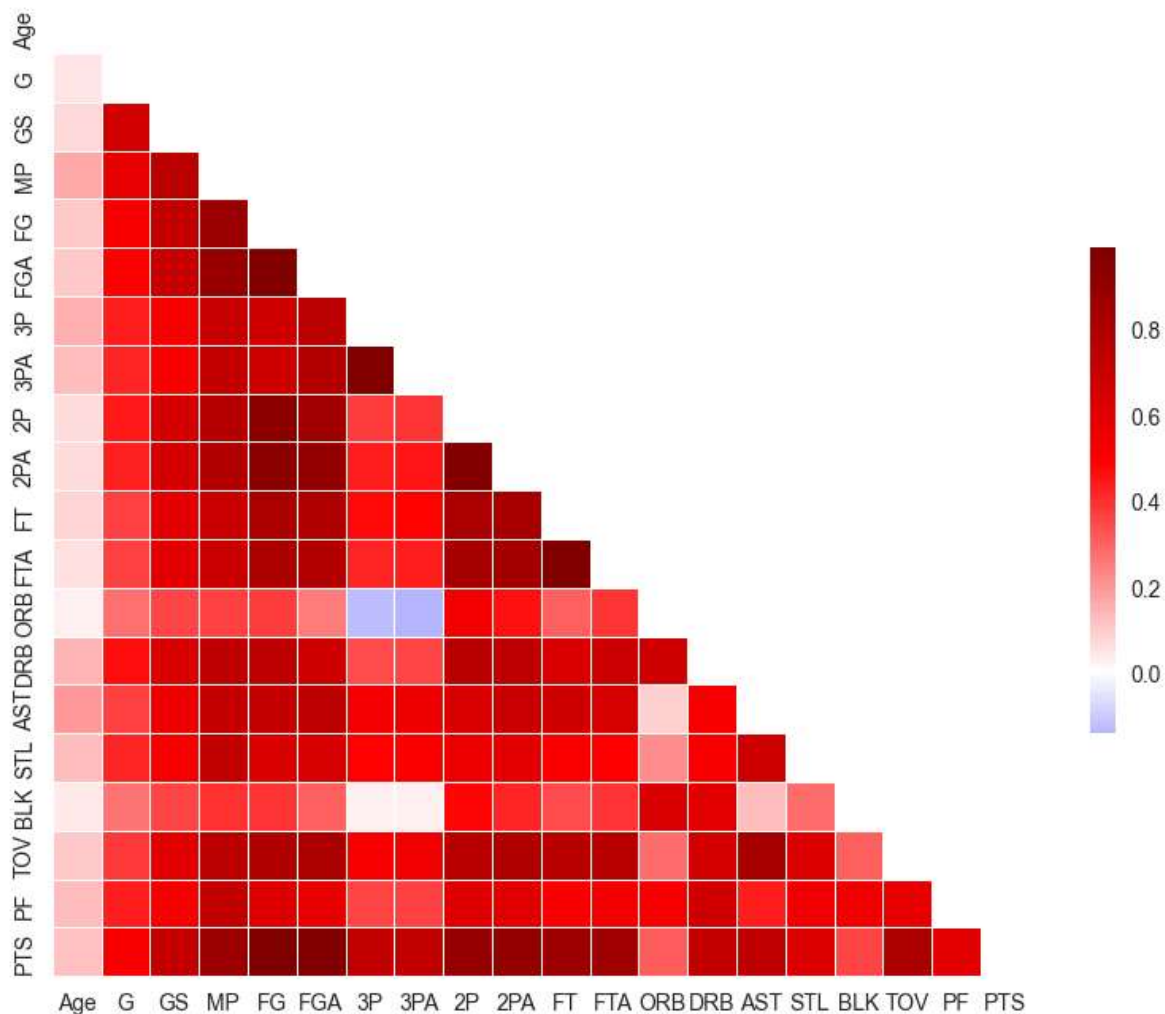
No parecen haber valores atípicos, los minutos jugados están dentro del rango (0-48 min), el rango de edades parece correcto y ninguna de las estadísticas está por debajo de 0 ni está fuera de un rango razonable.

2.5 Análisis final de nuestro conjunto de datos preprocesados



Podemos ver que algunas variables como la edad, los minutos por partido, los intentos de tiro o las faltas personales siguen una distribución parecida a la normal. El resto parece tener una acumulación de jugadores en los rangos más bajos.

Una de las posibles interpretaciones que le podríamos dar es que los jugadores de baloncesto no balancean sus estadísticas, es decir, se centran en una o dos características compatibles con su rol. Es razonable pensar que un base no conseguirá tantos rebotes como un pívot, ni este conseguirá tantos tiros de 3 como el base.

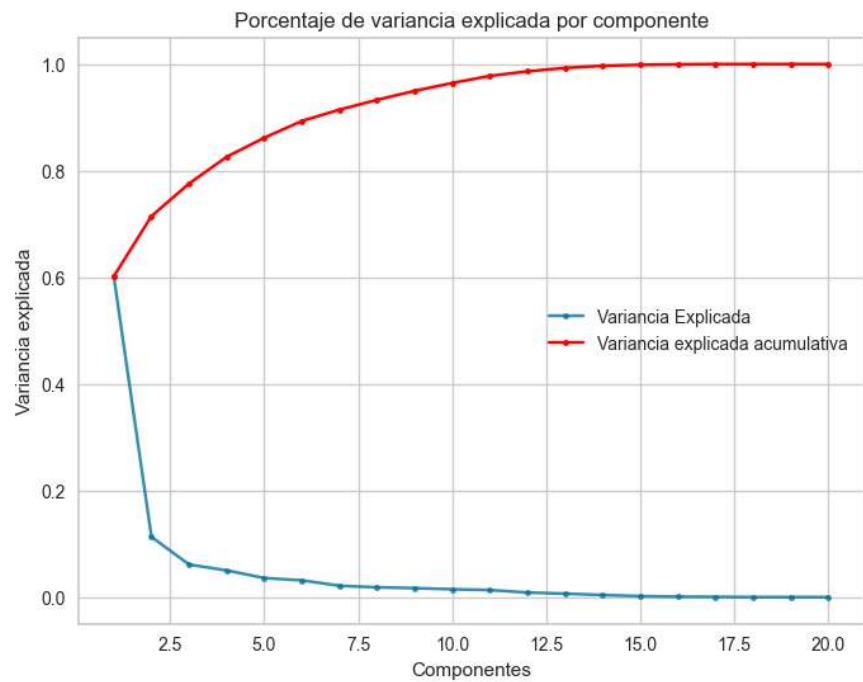


En la matriz de correlación podemos ver que casi todos los atributos están relacionados de forma positiva, lo cual tiene sentido, ya que si un jugador juega más partidos, tiene más ocasiones de tirar, de hacer asistencias o de coger rebotes; por tanto a mayor número de partidos mejores estadísticas.

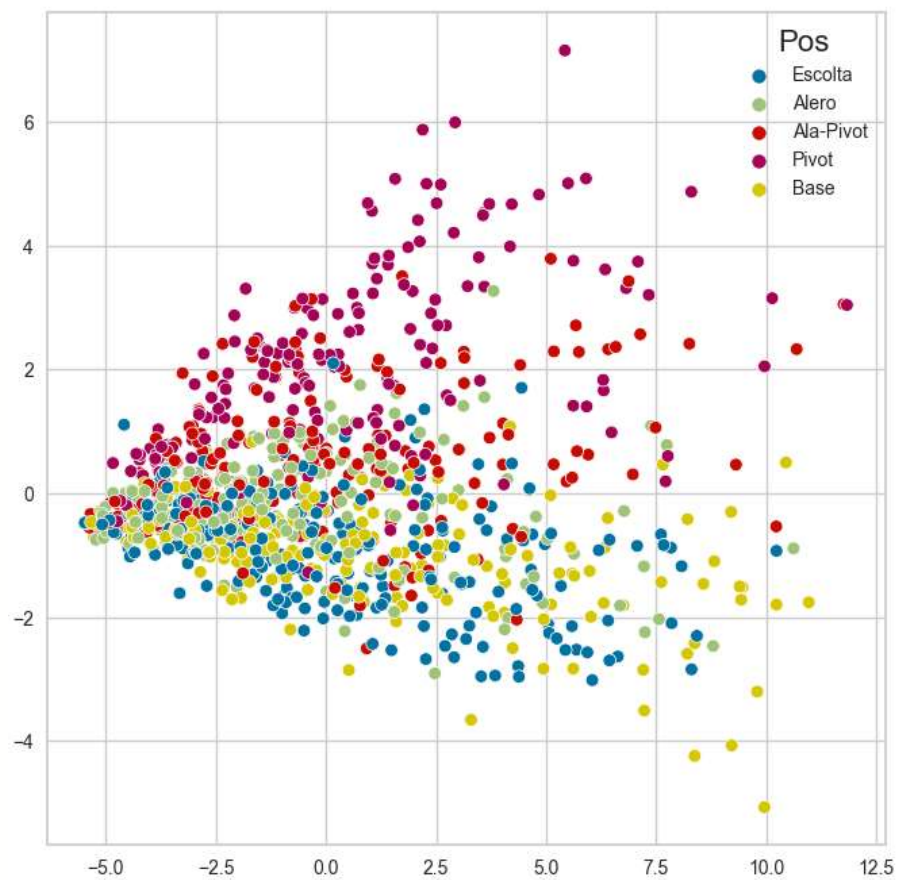
Cabe destacar que la edad de un jugador no parece tener un impacto enorme en sus estadísticas, aunque la relación ligeramente positiva nos indica que los jugadores con más experiencia suelen hacerlo un poco mejor.

Además podemos ver una correlación negativa entre los tiros de 3 puntos y los bloqueos y rebotes ofensivos, ya que estos suelen hacerlo los jugadores que juegan debajo del aro. También parece que estos suelen hacer menos asistencias, ya que BLK/ORB tiene una correlación débil con AST.

Para comprobar la separabilidad de las classes hemos usado una PCA:

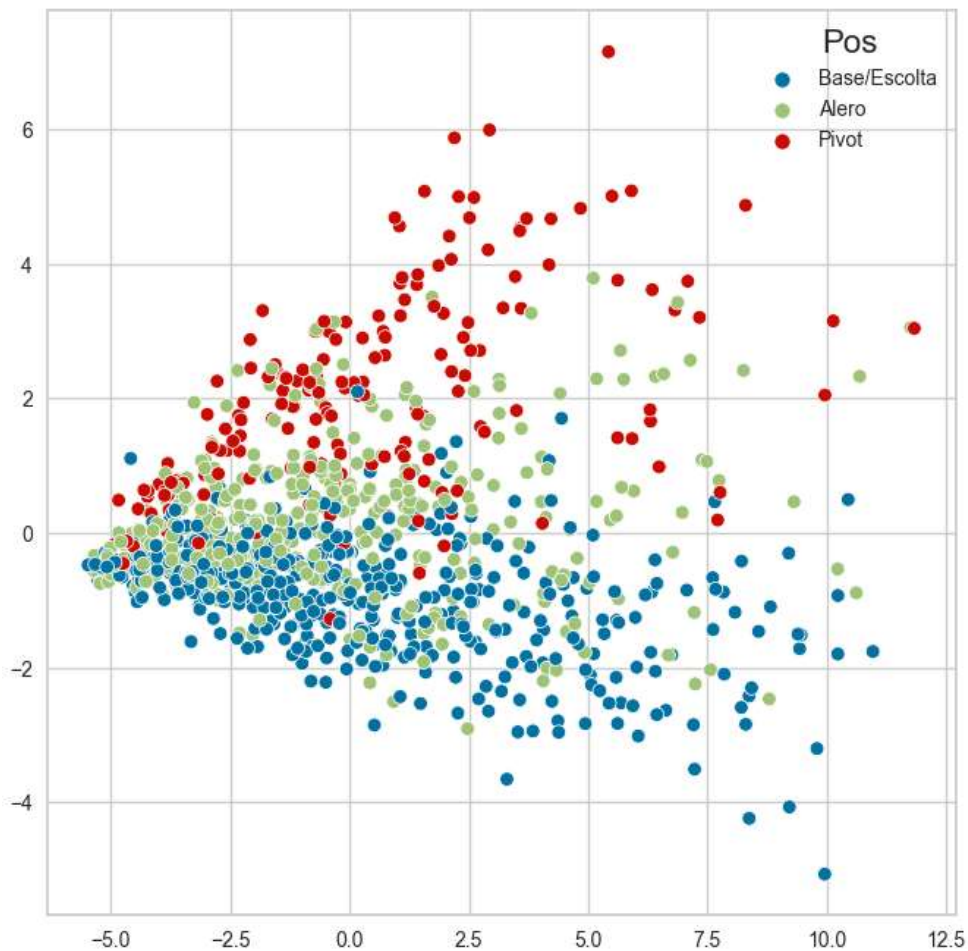


Tenemos el ~70% de la varianza en 2 componentes.



Se puede apreciar cierta separabilidad entre algunas de las clases, por ejemplo entre C y PG/SG/SF. Parece que nos va a costar distinguir entre PG/SG/SF y también, aunque un poco menos, entre C y PF.

Para simplificar el problema, ya que algunos roles del baloncesto hacen funciones muy similares y no hay separabilidad entre sus clases, vamos a agrupar los jugadores en 3 categorías generales: Base/Escolta, Alero y Pívot.



Redefinidas nuestras clases, podemos ver una separabilidad clara entre los Pívots y los Bases/Escoltas. Los Aleros siguen sin tener mucha separabilidad con ninguno de los dos. A pesar de ello, hemos considerado que esta división de las clases va a producir un beneficio en nuestros modelos.

2.6 División entrenamiento-prueba

Procedemos a hacer la partición de datos en conjunto de entrenamiento y conjunto de prueba.

```
Base/Escolta    460
Alero           408
Pivot           181
Name: Pos, dtype: int64
```

```
Base/Escolta    189
Alero           173
Pivot           88
Name: Pos, dtype: int64
```

Después de haber redistribuido las clases, se nos ha generado un desbalance en estas. Vamos a solucionar creando instancias sintéticas con SMOTE. Además también vamos a hacer undersampling del conjunto de test para tener el mismo número de instancias de cada clase, para así poder hacer una mejor del classification report.

```
Base/Escolta    460
Alero           460
Pivot           460
Name: Pos, dtype: int64
```

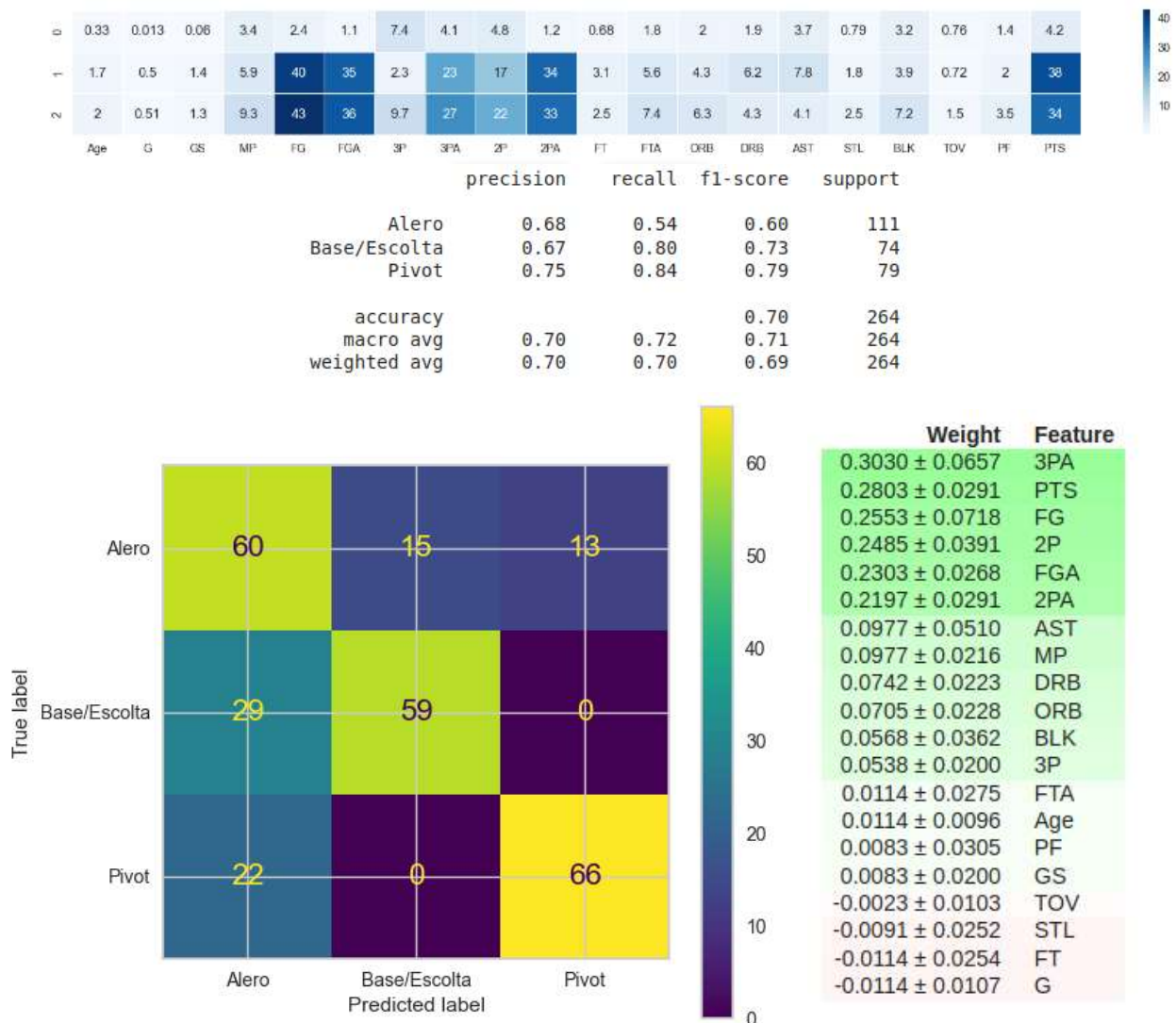
```
Alero           88
Base/Escolta    88
Pivot           88
Name: Pos, dtype: int64
```

3. Modelos clasificadores lineales/cuadráticos

3.1 LDA

Comenzamos haciendo la validación cruzada del método LDA:

Validación cruzada del método LDA:
0.7318840579710144



Observamos como los parámetros más relevantes son 3PA, PTS, FG, 2P y FGA. Podemos concluir que una eficiencia con una puntuación equivalente a 70 es equivalente a una puntuación medianamente buena, nuestro modelo no trabaja suficientemente bien con LDA.

A pesar de ello, no nos convence demasiado este modelo puesto que podemos observar cómo entre las posiciones Alero y Base/Escolta se ha

equivocado en una cantidad de elementos igual a la mitad de aciertos tanto de Alero como de Base/Escolta.

3.2 QDA

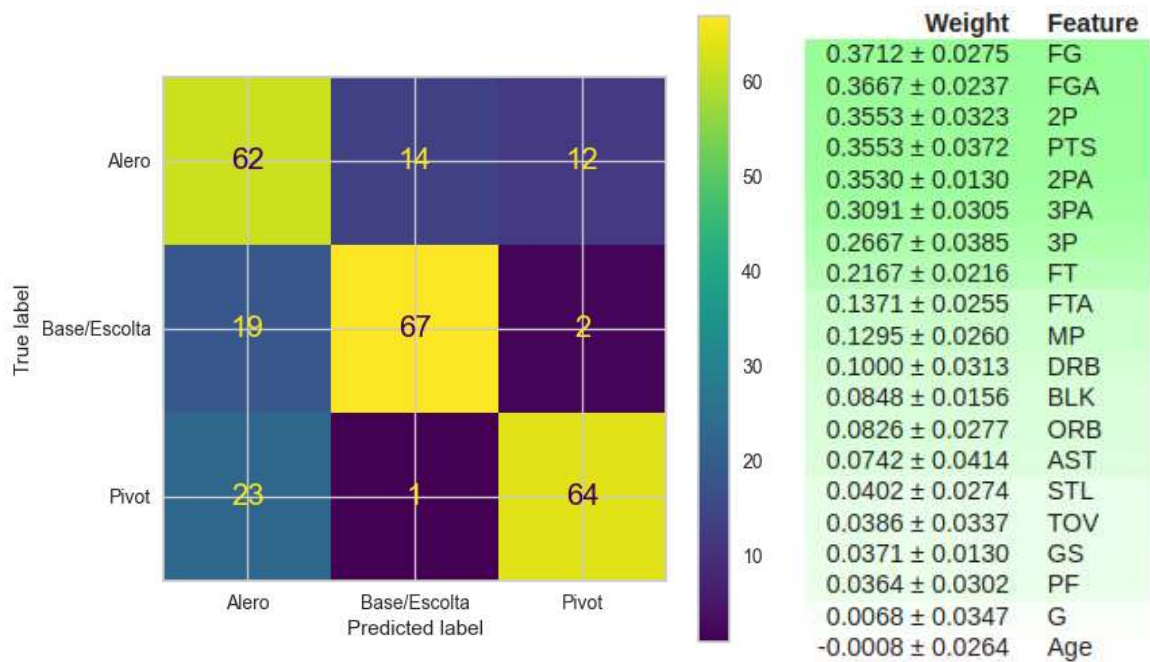
Comenzamos haciendo la validación cruzada del método QDA:

Validación cruzada del método QDA:
0.7471014492753624

	Pos	Age	G	GS	MP	FG	FGA	3P	3PA	2P	...	FT	FTA	ORB	DRB	AST	STL	BLK	TOV	PF	PTS
0	Alero	0.323	0.434	0.198	0.402	0.241	0.264	0.159	0.195	0.201	...	0.122	0.131	0.178	0.273	0.118	0.247	0.109	0.176	0.385	0.235
1	Base/Escolta	0.331	0.446	0.202	0.444	0.284	0.330	0.221	0.264	0.217	...	0.138	0.139	0.095	0.217	0.224	0.306	0.068	0.240	0.373	0.280
2	Pivot	0.350	0.458	0.263	0.405	0.276	0.257	0.075	0.092	0.290	...	0.141	0.162	0.353	0.384	0.106	0.232	0.233	0.210	0.498	0.251

		precision	recall	f1-score	support
	Alero	0.70	0.60	0.65	104
	Base/Escolta	0.76	0.82	0.79	82
	Pivot	0.73	0.82	0.77	78
	accuracy			0.73	264
	macro avg	0.73	0.74	0.74	264
	weighted avg	0.73	0.73	0.73	264

fitted cross_val of qda: 0.7471014492753624



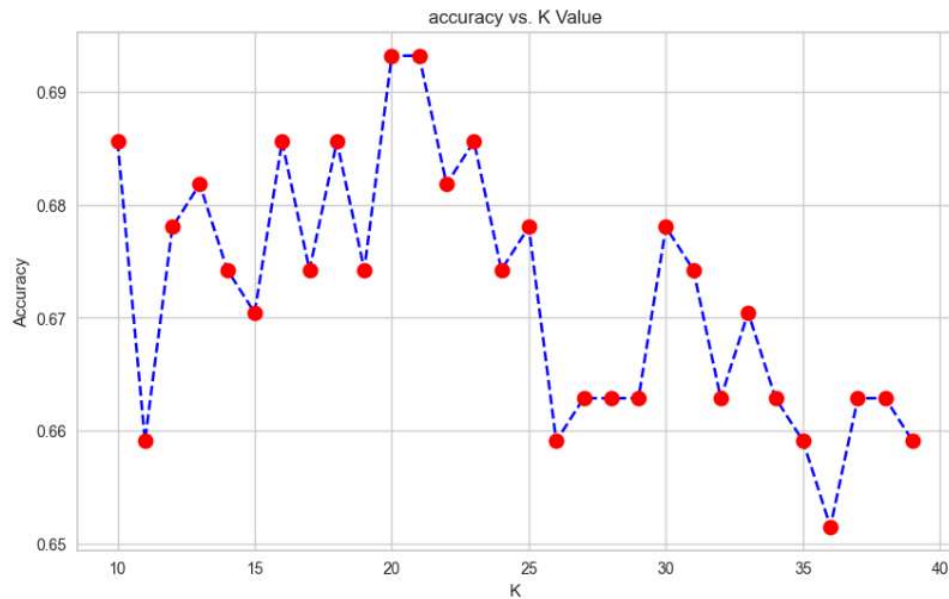
Vemos que la puntuación es ligeramente mejor que el modelo LDA. Se puede apreciar en la matriz un menor número de falsos positivos, así como un porcentaje de acierto algo mayor. Vemos también que los pesos son similares a los del LDA.

3.3 KNN

Comenzamos haciendo la validación cruzada del método KNN:

```
Validación cruzada del método KNN:  
0.7188405797101449
```

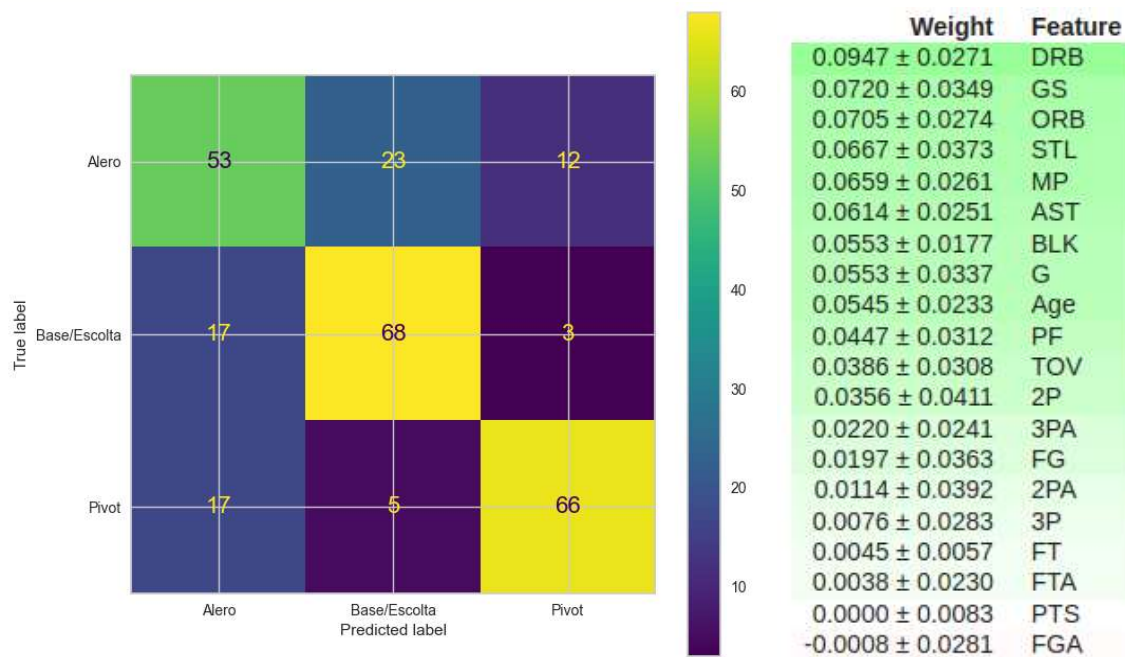
Antes de definir el modelo, hacemos una clasificación de KNN para saber cuáles son los mejor hiperparámetros:



Los hiperparámetros que mejor trabajan. En cuanto a la métrica de cálculo obtenemos 'manhattan' y el número de vecinos es 14.

```
fitted cross_val of knn_gs: 0.8057971014492754
```

	precision	recall	f1-score	support
Alero	0.60	0.61	0.61	87
Base/Escolta	0.77	0.71	0.74	96
Pivot	0.75	0.81	0.78	81
accuracy			0.71	264
macro avg	0.71	0.71	0.71	264
weighted avg	0.71	0.71	0.71	264



Podemos observar unos resultados similares a los dos anteriores, aunque cabe destacar que este modelo parece sufrir de overfitting, ya que la validación cruzada nos da casi un 10% más de acierto que el test. Esto, junto a que la probabilidad de acierto es algo menor a los demás, hace que este modelo no sea muy deseable.

Vemos que los pesos distan bastante de los obtenidos anteriormente, ya que a este modelo le parece darle más importancia a los rebotes o las veces que se empieza de titular.

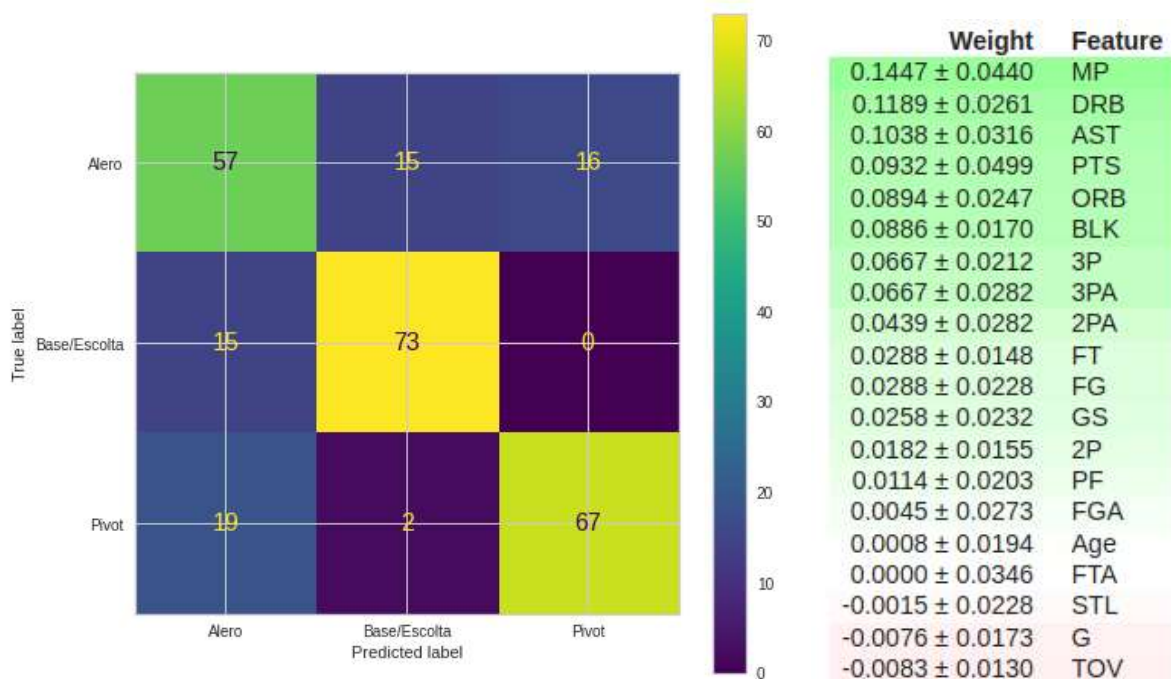
3.4 SVM

Comenzamos haciendo la validación cruzada del método SVM:

```
Validación cruzada del método SVM:  
0.7789855072463767
```

Para saber cuál es la mejor manera de definir nuestro modelo, procedemos a hacer un estudio de cuáles son los mejores hiperparámetros que mejor trabajan con nuestro conjunto de datos. En nuestro caso, ha resultado ser la regularización del parámetro C igual a 100 y trabajaremos con el kernel lineal ya que en este apartado trataremos modelos lineales.

```
F-score (mean):  
precision recall f1-score support  
Alero      0.63    0.65    0.64      88  
Base/Escolta 0.81    0.83    0.82      88  
Pivot      0.81    0.76    0.78      88  
  
accuracy          0.75    264  
macro avg         0.75    264  
weighted avg      0.75    264  
  
fitted cross_val of svm_fit: 0.7695652173913043
```



Observamos como la eficiencia de acierto es de un 75, y se observa como la matriz de confusión da resultados bastante buenos a nuestro parecer. Este modelo tiene el mayor porcentaje de acierto de todos los vistos hasta ahora. Parece ser que también nos encontramos con un caso de ligero overfitting. Los pesos son parecidos a los del knn.

4. Modelos clasificadores no lineales

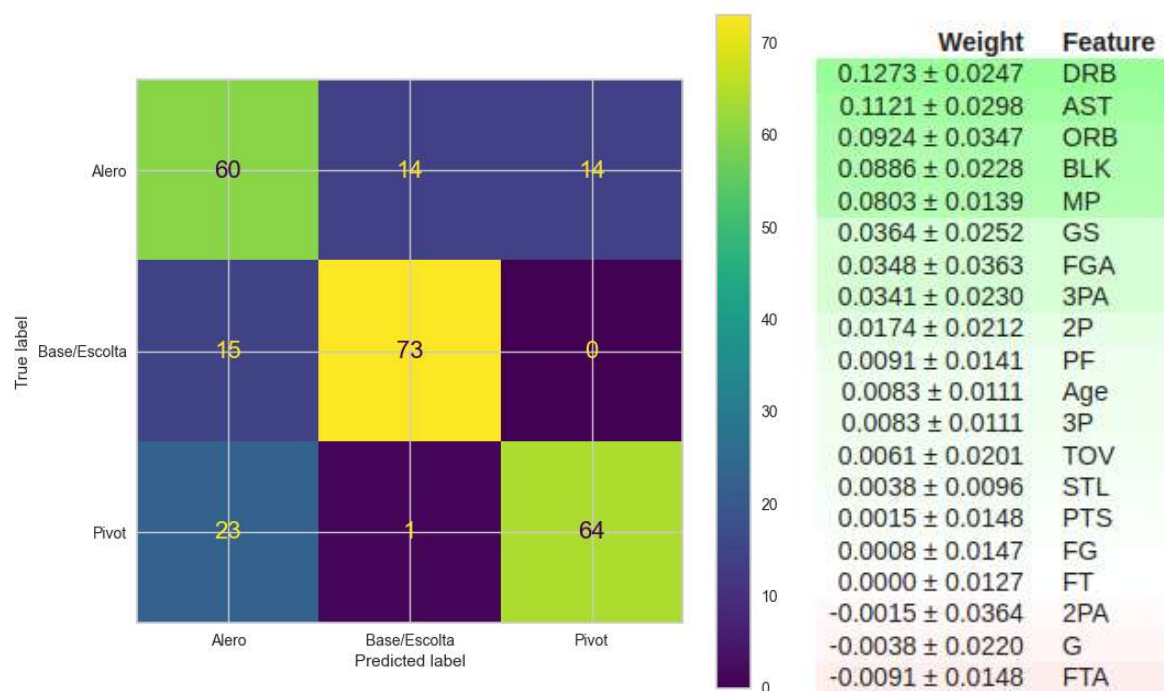
4.1 MLP

Mejores parámetros:

params	mean_test_score	rank_test_score
{'activation': 'relu', 'alpha': 0.001, 'hidden_layer_sizes': 100, 'learning_rate': 'constant', 'learning_rate_init': 0.01, 'momentum': 0.7, 'n_iter_no_change': 50}	0.768	1

fitted cross_val of mlp_fit: 0.7753623188405797

	precision	recall	f1-score	support
Alero	0.68	0.61	0.65	98
Base/Escolta	0.83	0.83	0.83	88
Pivot	0.73	0.82	0.77	78
accuracy			0.75	264
macro avg	0.75	0.75	0.75	264
weighted avg	0.74	0.75	0.74	264



El modelo parece tener muy clara la diferencia entre Base y Pivot, pero le cuesta más hacer la distinción entre Alero y las demás. Aun así tenemos una tasa de acierto del 75% en el test.

Estos pesos son parecidos a los del knn, pero con una distinción más clara entre los que son más importantes.

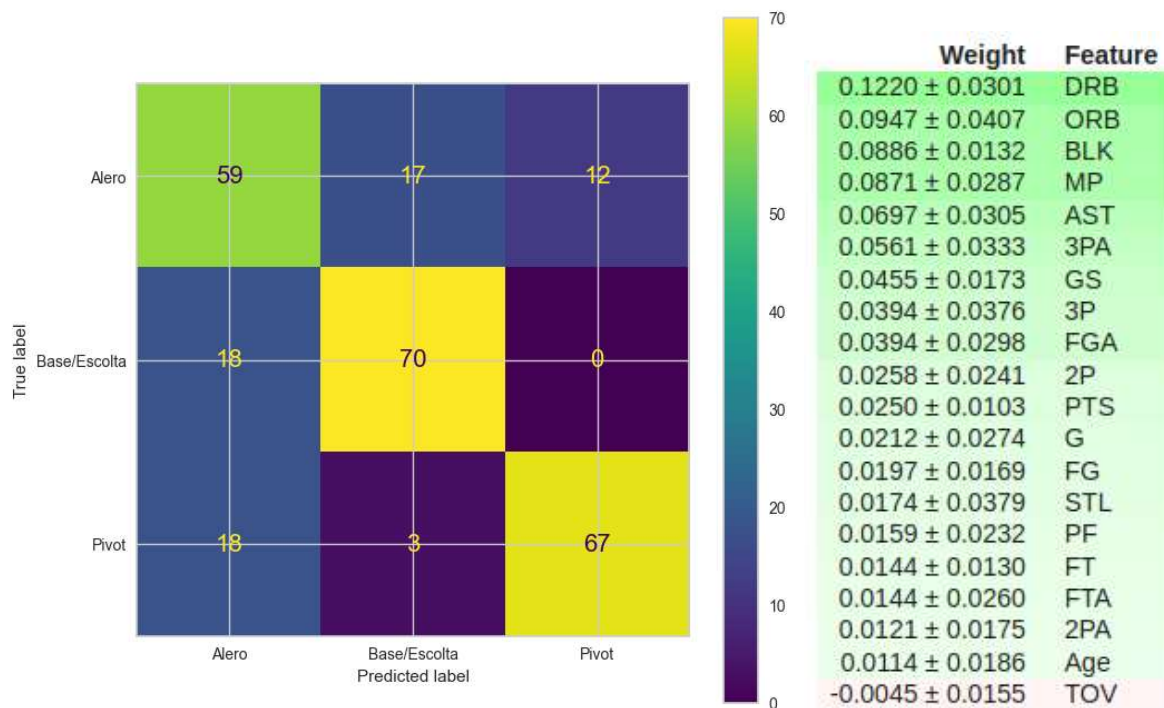
4.2 SVM con kernel RBF

Mejores parámetros:

params	mean_test_score	rank_test_score
{'C': 6.918309709189362, 'gamma': 'scale'}	0.799	1

fitted cross_val of svmRBF: 0.7985507246376812

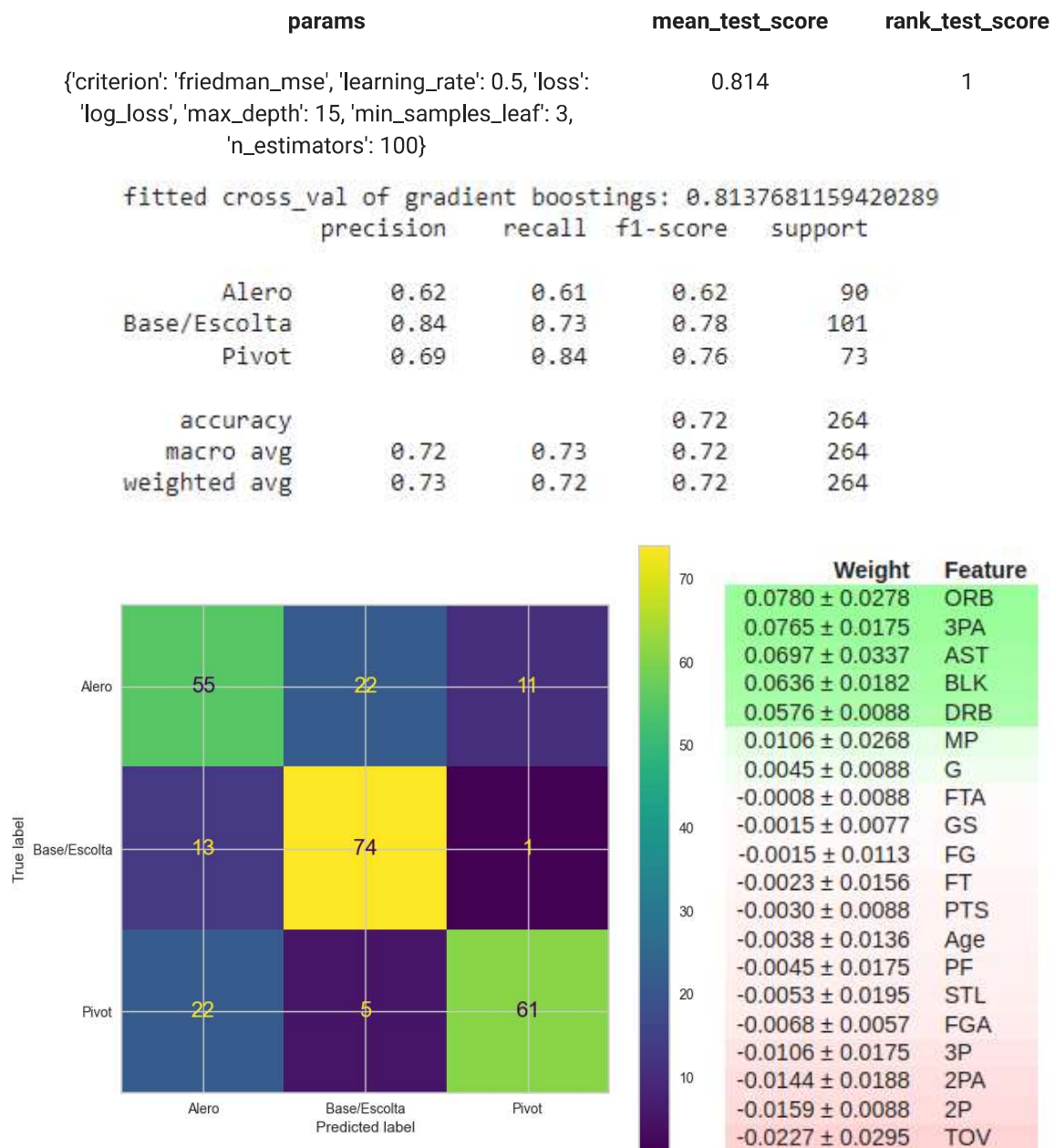
	precision	recall	f1-score	support
Alero	0.67	0.62	0.64	95
Base/Escolta	0.80	0.78	0.79	90
Pivot	0.76	0.85	0.80	79
accuracy			0.74	264
macro avg	0.74	0.75	0.74	264
weighted avg	0.74	0.74	0.74	264



La validación cruzada parecería indicar que este modelo es mejor que el anterior, pero al usarlo contra los datos de test vemos un resultado peor. Parece que este modelo sufre de overfitting.

Los pesos de los atributos son parecidos a los que obtuvimos con el kernel lineal.

4.3 Gradient Boosting

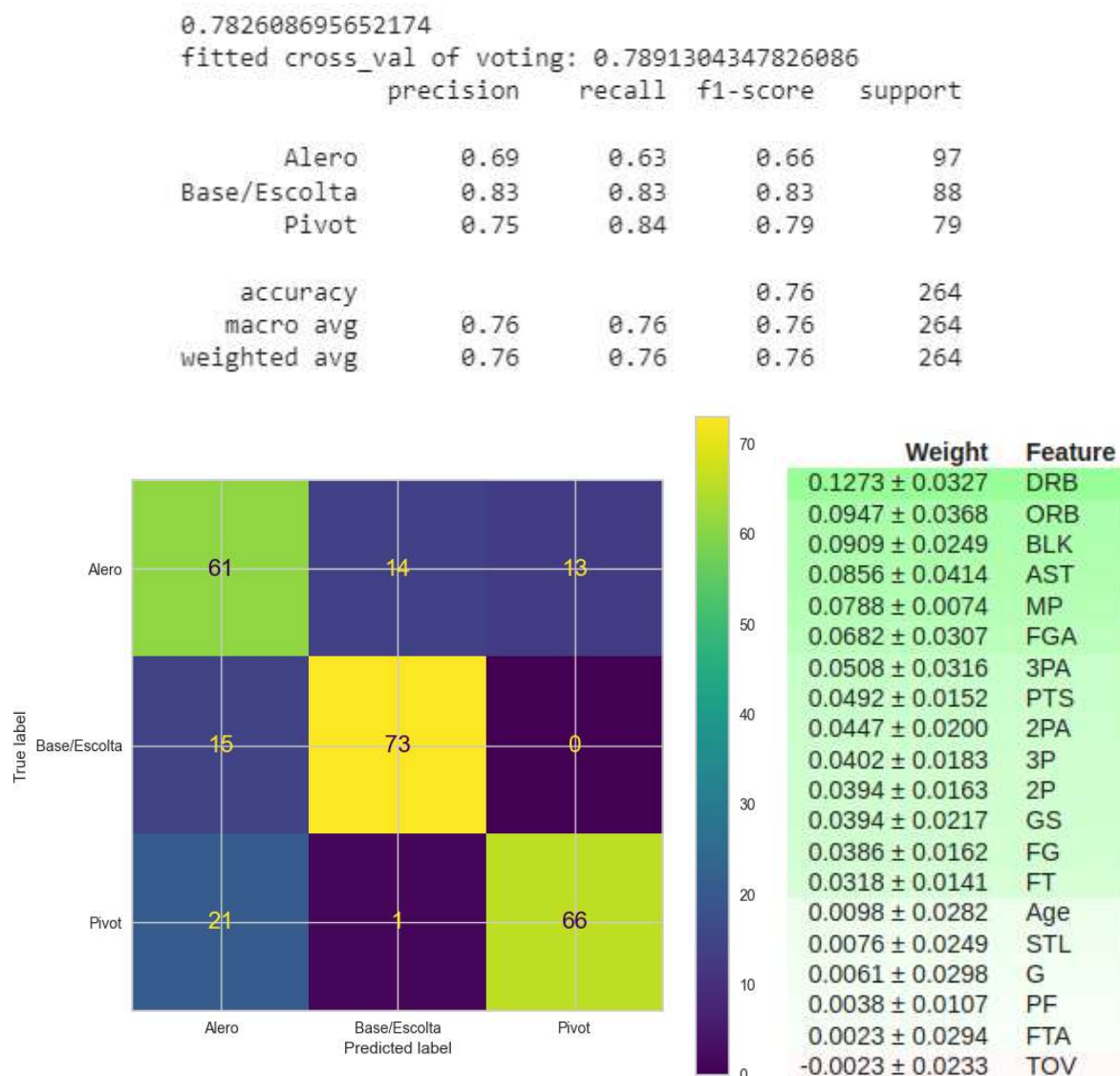


Este modelo nos da un resultado bastante mediocre, sin ninguna característica a destacar. También sufre de overfitting.

Los pesos son parecidos a los vistos hasta ahora.

4.4 VotingClassifier

Vamos a proceder a combinar los modelos que nos han parecido más útiles para entrenar un VotingClassifier. Vamos a utilizar MLP, que tiene el mayor acierto en Base/Escolta, SVM RBF, que tiene el mayor acierto en Pivots y QDA, que tiene la mejor puntuación de los Aleros. Además MLP está empatado en mayor puntaje global. Vamos a usar todos los modelos con sus mejores parámetros, definidos en los apartados anteriores. Asimismo estos modelos dan pesos diferentes a los atributos.



Si bien la confusión matrix de los datos de training está lejos de lo ideal, este modelo es el que obtiene los mejores resultados en el conjunto de test.

Vemos que aún habiendo usado un modelo lineal en la combinación, la distribución de los pesos sigue la línea del resto de los modelos no lineales.

5. Modelo final

Como modelo final hemos decidido quedarnos con el VotingClassifier, que combina SVM_RBF, MLP y QDA. Si bien este es el modelo más costoso de entrenar, ya que hay que ajustar los parámetros de los tres clasificadores de forma independiente, también es el que nos ofrece el mayor porcentaje de aciertos tanto global como por cada una de las clases.

	test acc	precision score (w)	recall score (w)	f1 score (w)
Voting RBF+MLP+QDA	0.758	0.765	0.758	0.760
SVM con kernel lineal	0.746	0.748	0.746	0.747
MLP	0.746	0.754	0.746	0.749
SVM con kernel RBF	0.742	0.749	0.742	0.745
QDA	0.731	0.745	0.731	0.735
Gradient Boosting	0.720	0.726	0.720	0.720
KNN	0.708	0.711	0.708	0.709
LDA	0.701	0.724	0.701	0.707

6. Conclusiones

6.1 Acerca del conjunto de datos

- Para la extracción de datos hemos creído oportuno usar solamente los dos últimos años del conjunto, que se remonta al 1998, ya que el baloncesto ha evolucionado bastante desde entonces y creemos que la misma posición se juega de distinta forma a como se hacía hace 20 años.
- Hemos tenido complicaciones a la hora de balancear los pesos ya que muchas posiciones de nuestro conjunto de datos eran mixtas, escolta y/o alero, pivot y/o ala-pivot, etc,... Es por eso que hemos tenido que redefinir estas posiciones ambiguas y, por ese motivo, se han observado en la matriz de confusión existían errores entre posiciones que podían ser compartidas por algunos jugadores.
- También hemos tomado la decisión de juntar las 5 posiciones naturales que existen en tres posiciones más generales aún para que los modelos fuesen más acertados. Tras haber probado alguna que otra combinación, hemos considerado la más adecuada a la vez que coherente a nivel baloncestístico, agrupar las posiciones en Base/Escolta, Alero/Ala-pivot y Pivot. Ya que es

más común que un escolta juegue de base que de alero y que un ala-pívot juegue de alero que de pívot.

- Hemos observado que algunos atributos han resultado muy determinantes para nuestros modelos a la hora de diferenciar posiciones con un rol muy diferenciado (por ejemplo: los rebotes ofensivos rara vez son estadísticas relevantes para posiciones diferentes a la del pívot)

6.2 Posibles extensiones

Como hemos comentado anteriormente, solamente hemos usado las dos últimas temporadas de la NBA. Creemos que en un futuro sería posible implementar un predictor que tuviera en cuenta el año como una variable más, o bien intentar predecir el año en el que jugó una persona teniendo en cuenta su posición y estadísticas.

6.3 Comparación pesos lineales vs no-lineales

La distribución de los pesos que podemos observar parece ir más ligada al modelo en sí que a si es lineal o no lineal

lda		knn		mlp	
Weight	Feature	Weight	Feature	Weight	Feature
0.3030 ± 0.0657	3PA	0.0947 ± 0.0271	DRB	0.1273 ± 0.0247	DRB
0.2803 ± 0.0291	PTS	0.0720 ± 0.0349	GS	0.1121 ± 0.0298	AST
0.2553 ± 0.0718	FG	0.0705 ± 0.0274	ORB	0.0924 ± 0.0347	ORB
0.2485 ± 0.0391	2P	0.0667 ± 0.0373	STL	0.0886 ± 0.0228	BLK
0.2303 ± 0.0268	FGA	0.0659 ± 0.0261	MP	0.0803 ± 0.0139	MP
0.2197 ± 0.0291	2PA	0.0614 ± 0.0251	AST	0.0364 ± 0.0252	GS
0.0977 ± 0.0510	AST	0.0553 ± 0.0177	BLK	0.0348 ± 0.0363	FGA
0.0977 ± 0.0216	MP	0.0553 ± 0.0337	G	0.0341 ± 0.0230	3PA
0.0742 ± 0.0223	DRB	0.0545 ± 0.0233	Age	0.0174 ± 0.0212	2P
0.0705 ± 0.0228	ORB	0.0447 ± 0.0312	PF	0.0091 ± 0.0141	PF
0.0568 ± 0.0362	BLK	0.0386 ± 0.0308	TOV	0.0083 ± 0.0111	Age
0.0538 ± 0.0200	3P	0.0356 ± 0.0411	2P	0.0083 ± 0.0111	3P
0.0114 ± 0.0275	FTA	0.0220 ± 0.0241	3PA	0.0061 ± 0.0201	TOV
0.0114 ± 0.0096	Age	0.0197 ± 0.0363	FG	0.0038 ± 0.0096	STL
0.0083 ± 0.0305	PF	0.0114 ± 0.0392	2PA	0.0015 ± 0.0148	PTS
0.0083 ± 0.0200	GS	0.0076 ± 0.0283	3P	0.0008 ± 0.0147	FG
-0.0023 ± 0.0103	TOV	0.0045 ± 0.0057	FT	0.0000 ± 0.0127	FT
-0.0091 ± 0.0252	STL	0.0038 ± 0.0230	FTA	-0.0015 ± 0.0364	2PA
-0.0114 ± 0.0254	FT	0.0000 ± 0.0083	PTS	-0.0038 ± 0.0220	G
-0.0114 ± 0.0107	G	-0.0008 ± 0.0281	FGA	-0.0091 ± 0.0148	FTA

LDA y QDA dan mucha más importancia a los puntos y a los tiros en general, mientras que knn, svm lineal y los modelos no lineales en general priorizan los rebotes, tanto ofensivos como defensivos, las asistencias y los bloqueos.

7. Bibliografía

1. Acerca de donde hemos extraído nuestro conjunto de datos:
<https://data.world/etocco/nba-player-stats>.
2. Acerca de las posiciones y los roles que desarrollan en la pista:
https://en.wikipedia.org/wiki/Basketball_positions
3. Acerca de los términos y estadísticas que se miden de cada jugador:
<https://www.sportingnews.com/es/nba/news/estadistica-avanzada-diccionario-terminos-que-es-que-sirve-nba-analytics-sabermetrics/1gc73ududvomy1s3mqiahg238c>
4. Documentación de la librería Scikit-learn, la librería principal de donde hemos sacado las funciones de los modelos: <https://scikit-learn.org/stable/#>