

**[NEXT]
[D'OR]
[MAN]**



SANTIAGO ANDRES CHAIN SANTOS
2200147

MIGUEL ANGEL MOLINA GARZÓN
2200161

Universidad
Industrial de
Santander



INTRODUCCIÓN

Se busca entrenar un modelo el cual prediga qué jugador de fútbol será el ganador del Balón de Oro en la actualidad y en los próximos años. Se utilizarán datos del videojuego FIFA que representan estadísticas de los jugadores. Se emplearán modelos de clasificación y regresión para realizar estas predicciones basadas en la información proporcionada. El futbolista que tenga mejor valoración general en el juego se considerará como el Balón de Oro.

**Modelo: GaussianNB,
RandomForestClassifier,
DecisionTreesClassifier, SVC y
DecisionTreesRegressor.**



DATASET

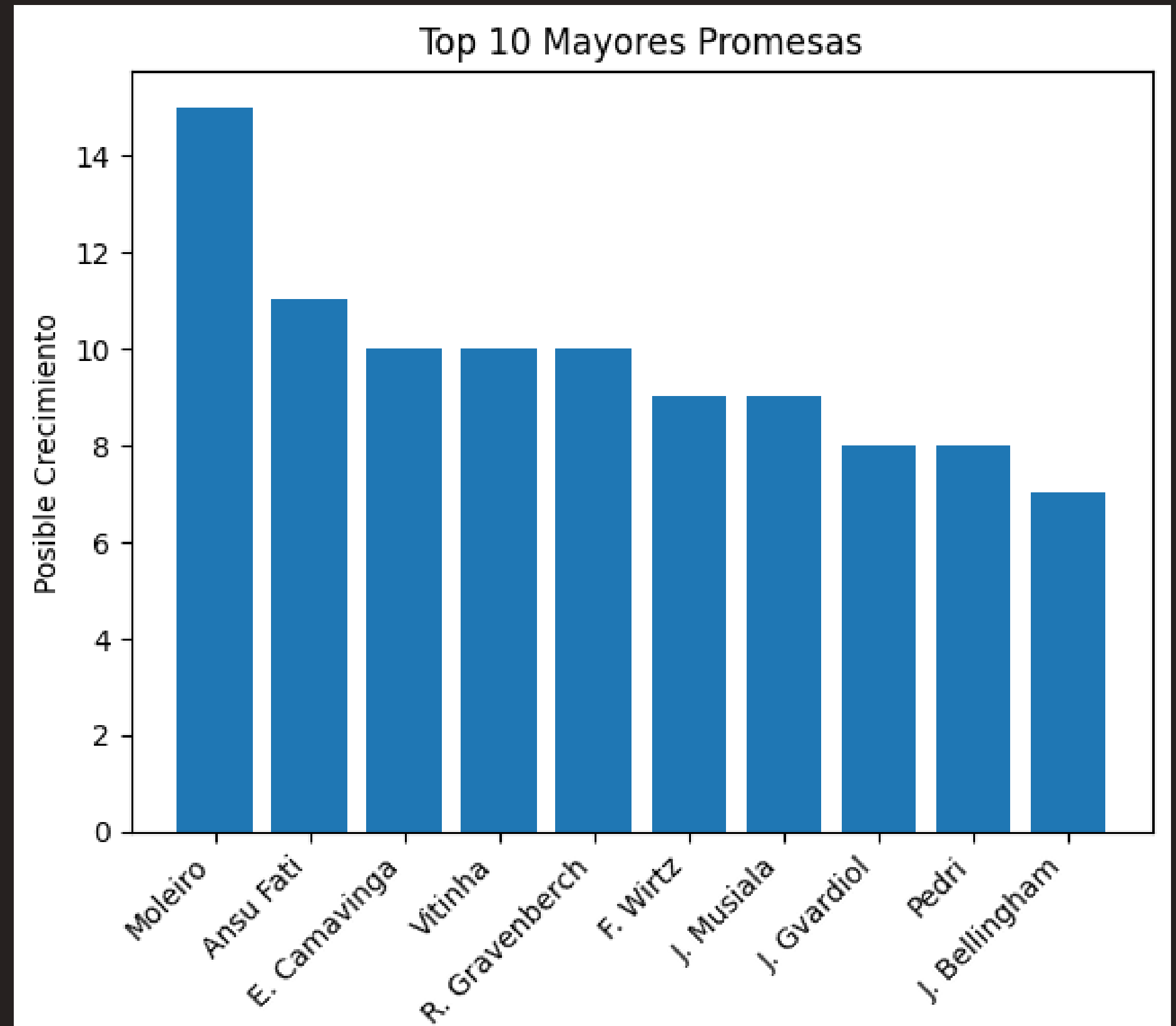
	Known As	Full Name	Overall	Potential	Value(in Euro)	Positions Played	Best Position	Nationality	Image Link	Age	...	LM Rating	CM Rating	RM Rating	LWB Rating	CDM Rating
0	L. Messi	Lionel Messi	91	91	54000000	RW	CAM	Argentina	https://cdn.sofifa.net/players/158/023/23_60.png	35	...	91	88	91	67	66
1	K. Benzema	Karim Benzema	91	91	64000000	CF,ST	CF	France	https://cdn.sofifa.net/players/165/153/23_60.png	34	...	89	84	89	67	67
2	R. Lewandowski	Robert Lewandowski	91	91	84000000	ST	ST	Poland	https://cdn.sofifa.net/players/188/545/23_60.png	33	...	86	83	86	67	69
3	K. De Bruyne	Kevin De Bruyne	91	91	107500000	CM,CAM	CM	Belgium	https://cdn.sofifa.net/players/192/985/23_60.png	31	...	91	91	91	82	82
4	K. Mbappé	Kylian Mbappé	91	95	190500000	ST,LW	ST	France	https://cdn.sofifa.net/players/231/747/23_60.png	23	...	92	84	92	70	66
5	M. Salah	Mohamed Salah	90	90	115500000	RW	RW	Egypt	https://cdn.sofifa.net/players/209/331/23_60.png	30	...	90	85	90	74	71
6	T. Courtois	Thibaut Courtois	90	91	90000000	GK	GK	Belgium	https://cdn.sofifa.net/players/192/119/23_60.png	30	...	34	35	34	32	34
7	M. Neuer	Manuel Neuer	90	90	13500000	GK	GK	Germany	https://cdn.sofifa.net/players/167/495/23_60.png	36	...	47	53	47	39	46
8	Cristiano Ronaldo	C. Ronaldo dos Santos Aveiro	90	90	41000000	ST	ST	Portugal	https://cdn.sofifa.net/players/020/801/23_60.png	37	...	87	81	87	65	62
9	V. van Dijk	Virgil van Dijk	90	90	98000000	CB	CB	Netherlands	https://cdn.sofifa.net/players/203/376/23_60.png	30	...	73	79	73	83	88

(18538, 56)

EJEMPLO

Encontramos los jugadores mas prometedores.

La siguiente gráfica nos muestra aquellos jugadores los cuales tienen un rango de crecimiento mayor, dato a tener en cuenta en el estudio a futuro lejano.



CLASIFICACIÓN

Los jugadores de nuestro dataset serán clasificados según sus estadísticas individuales en si estos pueden ser posibles ganadores del Balón de oro.

Para esto agregamos una nueva columna la cual posee valores binarios siendo el 1 Potencial Ganador y el 0 No Potencial Ganador.

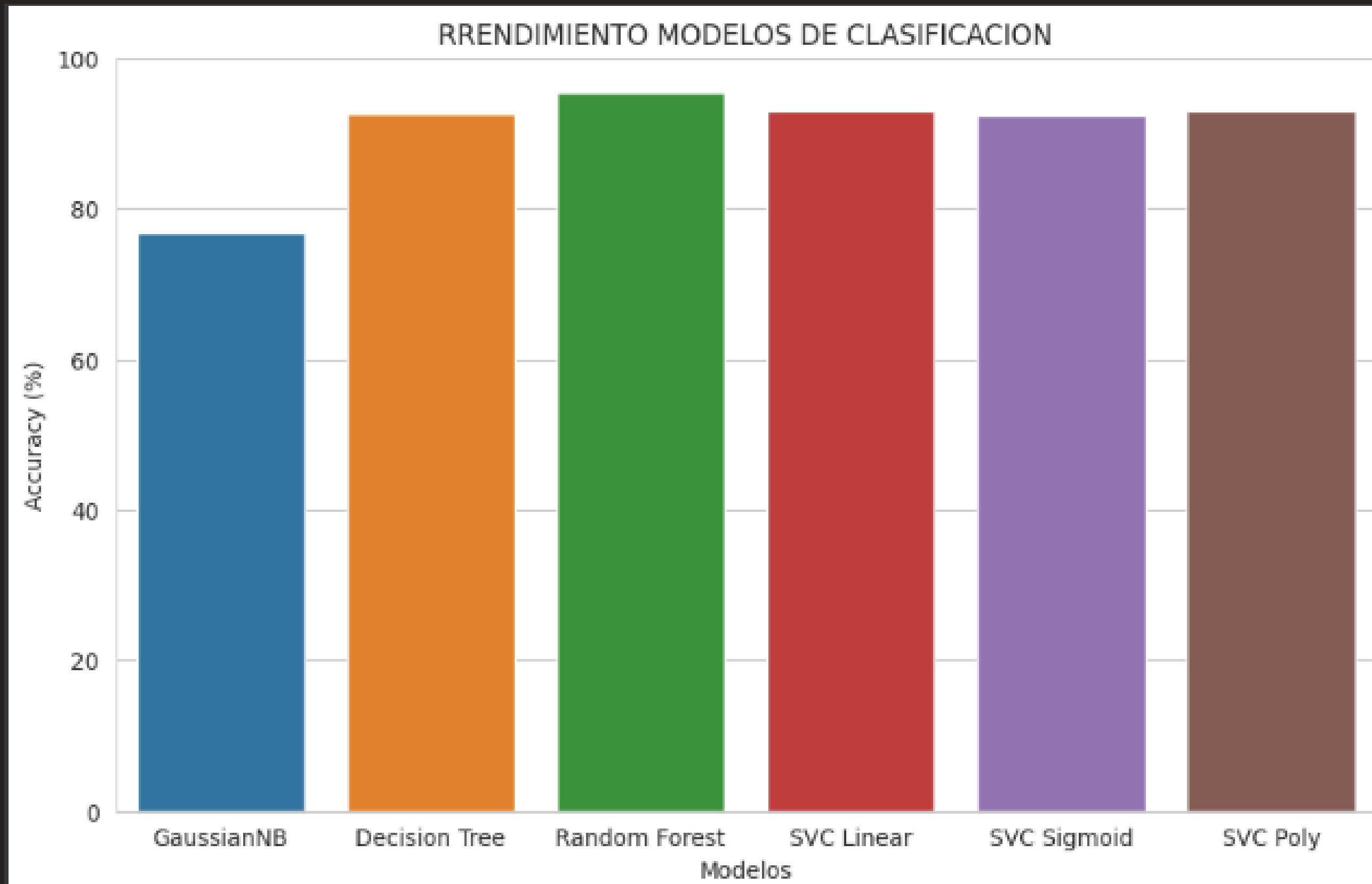


CLASIFICACIÓN

En el siguiente bloque se muestran todas las columnas elegidas a ser entrenadas y clasificadas, y a su vez definiendo nuestro groundtruth para el modelo de clasificación.

```
ent_col_clf=["International Reputation","Height(in cm)",  
            "TotalStats", "BaseStats","Preferred Foot" , "Pace Total" , "Shooting Total", "Passing Total",  
            "Dribbling Total", "Defending Total", "Physicality Total",'Crossing',  
            'Finishing', 'Heading Accuracy', 'Short Passing', 'Volleys',  
            'Dribbling', 'Curve', 'Freekick Accuracy', 'LongPassing', 'BallControl',  
            'Acceleration', 'Sprint Speed', 'Agility', 'Balance',  
            'Shot Power', 'Jumping', 'Stamina', 'Strength', 'Long Shots',  
            'Aggression', 'Interceptions', 'Positioning', 'Vision', 'Penalties',  
            'Composure', 'Marking', 'Standing Tackle', 'Sliding Tackle',"Potential_Winner"]  
  
#Se define el dataset a entrenar  
entrenamiento = df[ent_col_clf]  
  
X = entrenamiento.values[:, :-1]  
y = entrenamiento.values[:, -1]  
  
X, y = shuffle(X, y, random_state=200)  
  
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=200)
```

MÉTODOS DE CLASIFICACIÓN



FEATURE IMPORTANCES

	Feature	Importance
0	Dribbling Total	0.067238
1	BallControl	0.058766
2	BaseStats	0.049948
3	Dribbling	0.042424
4	TotalStats	0.033705
5	Interceptions	0.033295
6	Sliding Tackle	0.032866
7	Defending Total	0.030638
8	Composure	0.029817
9	Short Passing	0.029712
10	Standing Tackle	0.029046
11	Sprint Speed	0.028069
12	Vision	0.026641
13	Pace Total	0.025938
14	Acceleration	0.024374
15	Marking	0.024196
16	Shooting Total	0.024161
17	Passing Total	0.023743
18	Height(in cm)	0.023216

19	LongPassing	0.023117
20	Finishing	0.022510
21	Crossing	0.022305
22	Freekick Accuracy	0.021810
23	Long Shots	0.021134
24	Positioning	0.020989
25	Stamina	0.019918
26	Balance	0.019785
27	Heading Accuracy	0.019435
28	Jumping	0.019218
29	Volley	0.019138
30	Strength	0.018605
31	Physicality Total	0.018293
32	Curve	0.018082
33	Aggression	0.017528
34	Agility	0.017285
35	Penalties	0.016693
36	Shot Power	0.016041
37	International Reputation	0.007313
38	Preferred Foot	0.003009



REGRESIÓN



EN LA ACTUALIDAD



Construimos nuestro modelo de regresión basado en DecisionTreeRegressor, con el proposito de predecir la Valoración General (Overall) en base a las estadísticas con las cuales el modelo entrenó. Este proceso se relizara un número N de veces para una mayor precisión de los datos.

Una vez este modelo termine sus N iteraciones, verificamos cuales fueron los jugadores que tuvieron una mayor valoración general predecido en cada iteración. Con esta información se hallará la probabilidad de que este jugador sea ganador del Balón de oro.

```

#Copiamos y eliminamos los jugadores que no son potenciales ganadores
df_2 = entrenamiento.copy()
df_2 = df_2[df_2['Potential_Winner']!=0]

X = df_2.drop(['Overall'], axis=1)
y = df_2['Overall']
#
jugadores_valoracion = df_2.copy()
jugadores_valoracion['Full Name'] = df[['Known As']]
jugadores_valoracion['Valoracion_General'] = 0
jugadores_valoracion['Prob_BalonDeOro'] = 0
acc = []
for i in tqdm(range(1,101)):
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=i)
    reg_model = DecisionTreeRegressor(max_depth=20, min_samples_leaf=6)
    reg_model.fit(X_train, y_train)
    y_pred = reg_model.predict(X_test)
    r = r2_score(y_test, y_pred)
    acc.append(r)

    jugadores_valoracion['Valoracion_General'] += reg_model.predict(X)
    max_valoracion = jugadores_valoracion['Valoracion_General'].max()
    jugadores_valoracion['Prob_BalonDeOro'] += (jugadores_valoracion['Valoracion_General'] == max_valoracion).astype(int)

jugadores_valoracion['Valoracion_General'] /= i
jugadores_valoracion['Valoracion_General'] = np.round(jugadores_valoracion['Valoracion_General'], 3)
jugadores_valoracion['Prob_BalonDeOro'] = np.round((jugadores_valoracion['Prob_BalonDeOro'] / jugadores_valoracion['Prob_BalonDeOro'].sum())*100, 2)
jugadores_valoracion = jugadores_valoracion.sort_values('Valoracion_General', ascending=False)
jugadores_valoracion = jugadores_valoracion[['Full Name', 'Valoracion_General', 'Prob_BalonDeOro']]

print('\nR2:', np.mean(acc))

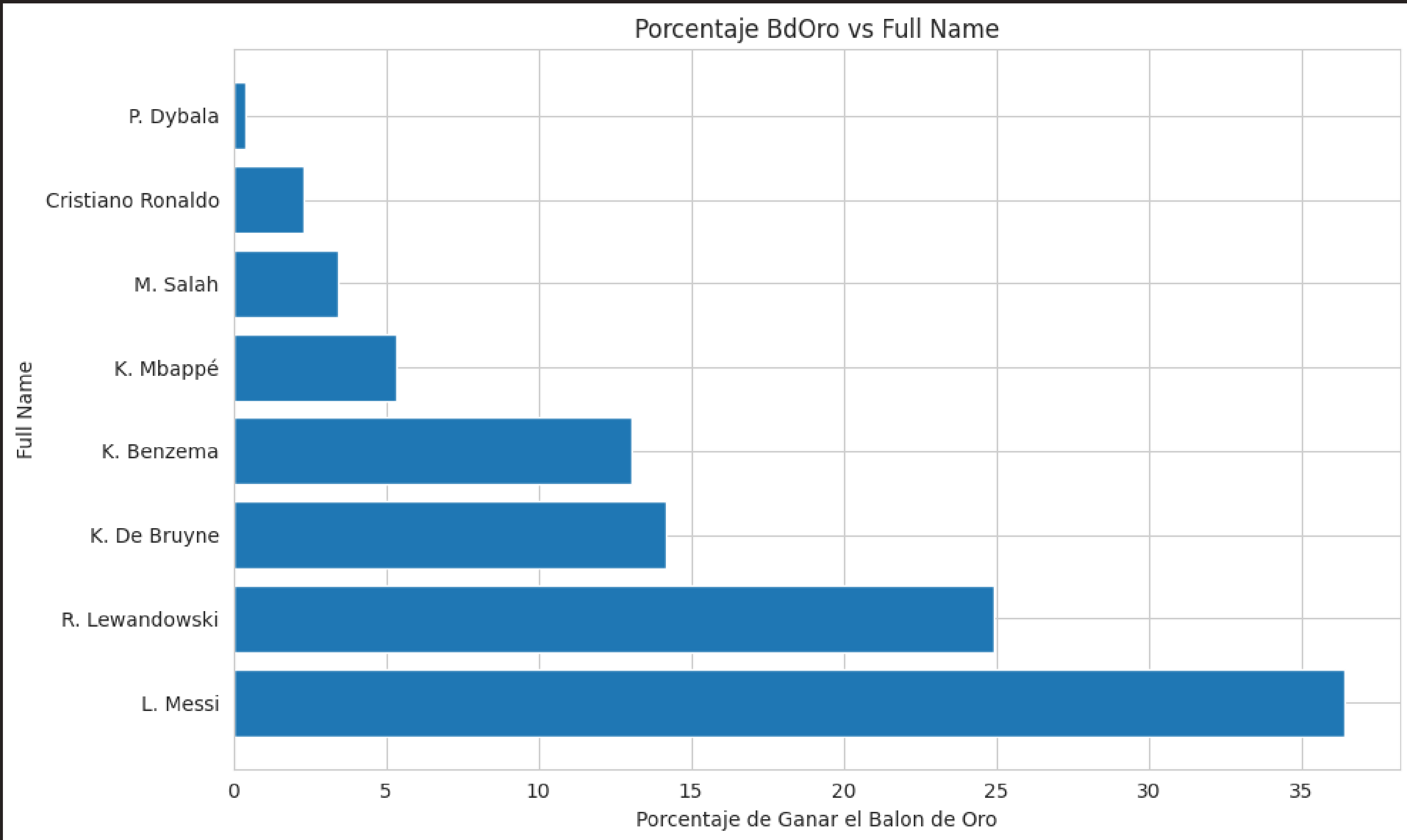
```

```

100%|██████████| 100/100 [00:02<00:00, 36.66it/s]
R2: 0.8967938176773853

```

RESULTADOS ACTUALIDAD



ESTUDIO A FUTURO



FUTURO CERCANO (5 AÑOS)



CRITERIOS DE ACTUALIZACIÓN DE ESTADÍSTICAS:

- Todo jugador $J > 33$ años decrecerá un 11% en todas sus estadísticas, y un 12% de Valoración.
- Todo jugador J entre 21-25 años elevará todas sus estadísticas en un 4%.
- Todo jugador $J < 21$ años elevará todas sus estadísticas en un 10% a excepción de
- Se agregan los 5 años que han "pasado".

▶ #@title *Definimos una funcion para simular una temporada* { display-mode: "form" }

```
def Simular_5años(datasetprueba):
```

```
    #PARA LOS MAYORES DE 33
```

```
    columnas_excluir = ['Age', 'Known As', 'Full Name', 'Overall', 'Potential', 'Positions Played', 'Best Position',  
                        'Nationality', 'Height(in cm)', 'Club Name', 'Club Position', 'Contract Until',  
                        'Preferred Foot', 'International Reputation', 'Estilo', 'Potential_Winner', 'National Team Name', 'Avg_Score']
```

```
    condicion = datasetprueba['Age'] > 33
```

```
    datasetprueba.loc[datasetprueba['Age'] > 33, 'Avg_Score'] *= 0.88
```

```
    columnas_reducir = datasetprueba.columns[~datasetprueba.columns.isin(columnas_excluir)]
```

```
    datasetprueba.loc[condicion, columnas_reducir] *= 0.89
```

```
    #ENTRE 21-25
```

```
    columnas_excluir = ['Age', 'Known As', 'Full Name', 'Overall', 'Potential', 'Positions Played', 'Best Position',  
                        'Nationality', 'Height(in cm)', 'Club Name', 'Club Position', 'Contract Until',  
                        'Preferred Foot', 'International Reputation', 'Estilo', 'Potential_Winner', 'National Team Name', 'Avg_Score']
```

```
    condicion = (datasetprueba['Age'] >= 21) & (datasetprueba['Age'] <= 25)
```

```
    columnas_reducir = datasetprueba.columns[~datasetprueba.columns.isin(columnas_excluir)]
```

```
    datasetprueba.loc[condicion, columnas_reducir] *= 1.04
```

```
    #MENOS DE 21
```

```
    columnas_excluir = ['Age', 'Known As', 'Full Name', 'Overall', 'Potential', 'Positions Played', 'Best Position',  
                        'Nationality', 'Height(in cm)', 'Club Name', 'Club Position', 'Contract Until',  
                        'Preferred Foot', 'International Reputation', 'Estilo', 'Potential_Winner', 'National Team Name', 'Avg_Score']
```

```
    condicion = datasetprueba['Age'] < 21
```

```
    columnas_reducir = datasetprueba.columns[~datasetprueba.columns.isin(columnas_excluir)]
```

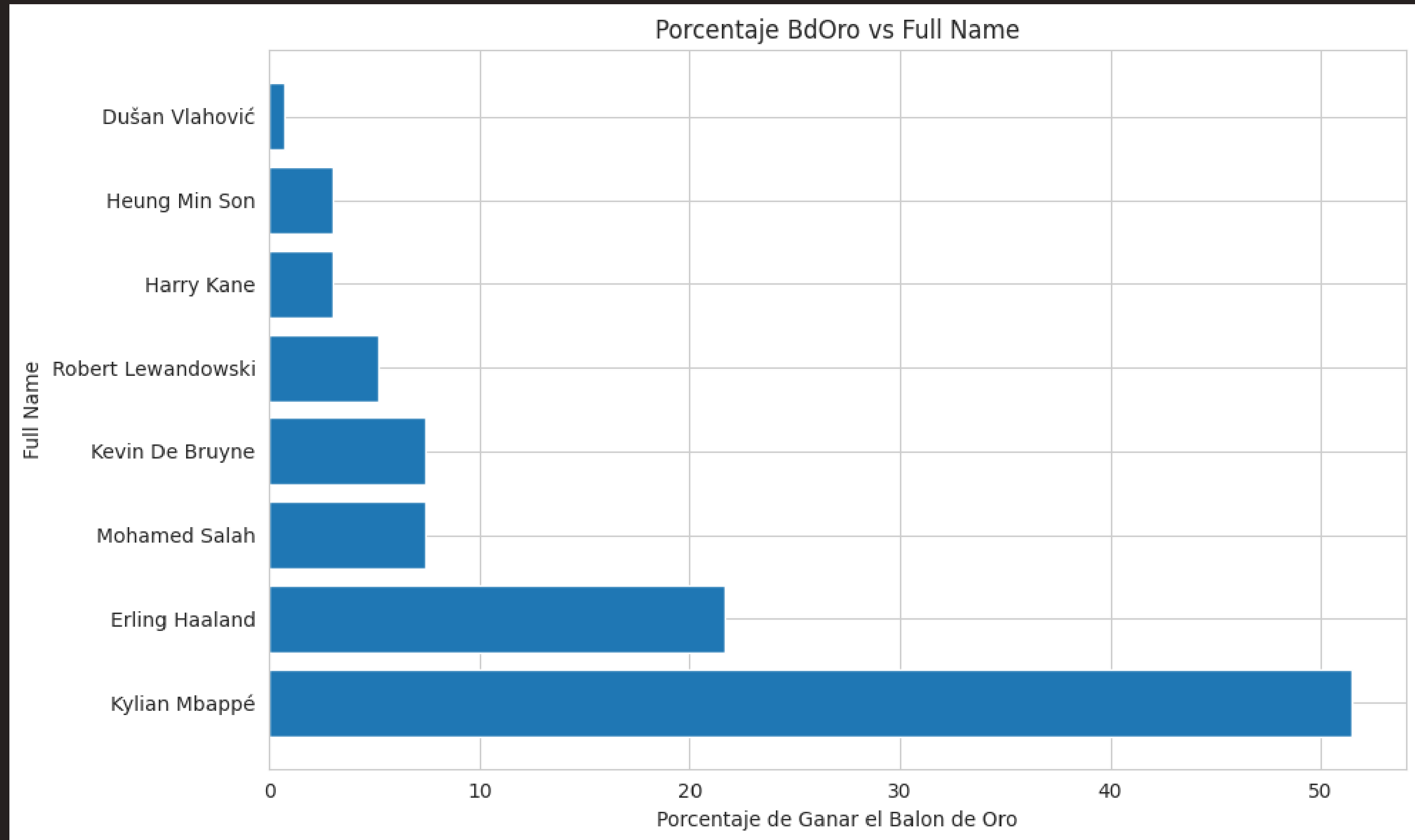
```
    datasetprueba.loc[condicion, columnas_reducir] *= 1.1
```

```
    datasetprueba.loc[datasetprueba['Age'] < 21, 'Avg_Score'] *= 1.2
```

```
    datasetprueba["Age"] += 5
```

```
    return datasetprueba
```


RESULTADOS FUTURO CERCANO





FUTURO LEJANO (10 AÑOS)

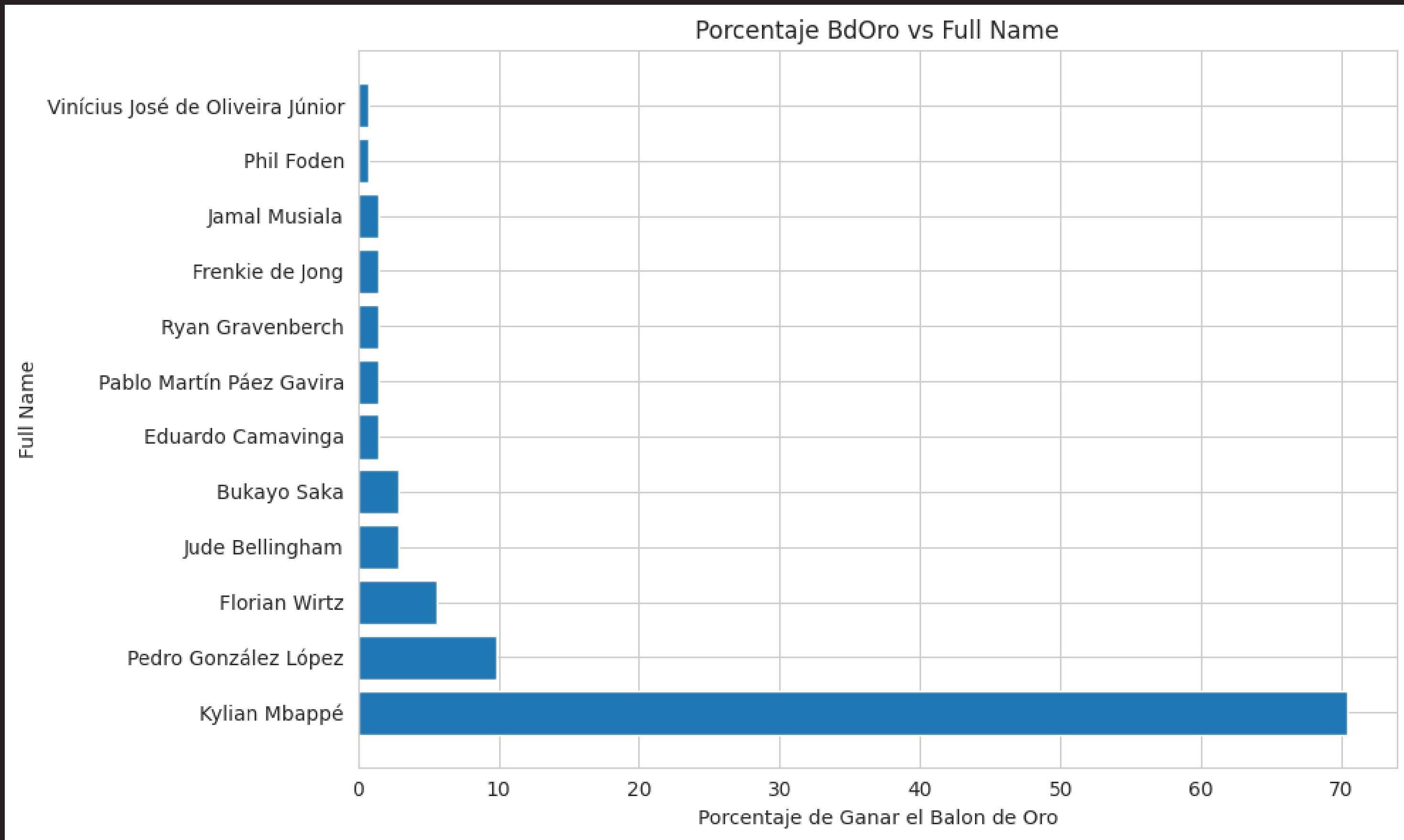


SE VUELVE A USAR LA FUNCIÓN QUE SIMULA EL PASO DE 5 AÑOS Y DA UNA ACTUALIZACIÓN DE ESTADÍSTICAS SEGÚN LA EDAD DEL FUTBOLISTA.

SE PROCURA QUE LOS JUGADORES MAS JOVENES ALCANCEN SU MÁXIMO POTENCIAL.



RESULTADOS FUTURO LEJANO



CONCLUSIONES

- **Al examinar los feature importances , se observó que el clasificador asignó un valor cercano ala mayoría de estadísticas entrenadas. Esto indica que los jugadores no tienen ventaja por la posición en la que jueguen, ya que el clasificador al otorga importancia equitativa a cada estadística.**
- **El proceso de simulación y estudio a futuro resultó ser un éxito, ya que los jugadores que identificamos previamente como promesas aparecen al final de la simulación como potenciales ganadores del Balón de Oro. Estos resultados respaldan la eficacia de nuestro enfoque de clasificación y regresión en la predicción de futuros talentos.**

CONCLUSIONES

- Durante la simulación a futuro, el accuracy del modelo de regresión decayó progresivamente debido a la variación en las estadísticas de los jugadores a lo largo del tiempo, afectando así la precisión de las predicciones a un futuro hipotético. Aunque el proceso proporcionó valiosa perspectiva, resalta la importancia de entender la naturaleza cambiante del fútbol.



GRACIAS

*'Que Deus nos abençoe e nos
proteja'*

