Universidad Nacional Autonoma De Mexico

Facultad de Estudios Superiores Acatlán

Proyecto Módulo 5

Diplomado en Ciencia de Datos, Módulo 5

Carlos Francisco Javier Carreola Silva

Número de cuenta: 315023613

Fecha de entrega del informe: 20 de enero de 2023

Introducción

En el mundo del fútbol, estamos constantemente en la búsqueda de métodos para comprender y predecir los resultados de los encuentros, usando nuestro buen juicio, el azar, “expertos en apuestas”, etc.

Sabemos que el aprendizaje automático se ha convertido en una herramienta poderosa a medida que avanzamos en la era digital, y nos ayuda a analizar grandes cantidades de datos y extraer información valiosa.

Los modelos de aprendizaje automático nos podrían ayudar a predecir la manera en que se desarrolla un partido, dándonos predicciones como la cantidad de goles en un partido de fútbol, número de tiros de esquina, el ganador, etc. Esto puede ser particularmente útil para realizar apuestas, y obtener beneficios económicos a partir de estas predicciones.

Planteamiento del problema:

El objetivo principal de este proyecto es desarrollar un modelo predictivo capaz de clasificar si un partido de fútbol tendrá más de 2 goles (Over 2.5) o no (Under 2.5). Para lograr esto, se utilizarán datos recopilados de Transfermarkt, centrándose en la información de los equipos participantes y sus respectivas plantillas, con el propósito de capturar elementos que puedan influir en la cantidad de goles anotados durante un partido.

Estos modelos podrán ser utilizados para obtener estimaciones precisas y confiables del total de goles en partidos futuros, lo cual puede ser de gran utilidad para realizar apuestas el cual en primera instancia esta destinada unicamente para buscar generar utilidades de manera privada en casinos digitales por lo cual el producto final sera la prediccion de la apuesta over 2.5 goles de partidos futuros , sin embargo dependiendo de su exito podria convertirse en un producto destinado a otros usuarios con un despliegue mas formal y con mayor infraestructura.

Datos con los que contamos:

Los datos fueron recuperados de Transfermarkt.com, la información viene organizada en 5 tablas:

1. Club: Esta tabla incluye información relevante sobre cada equipo de fútbol, como su valor de mercado, nombre de director técnico, nombre de estadio, capacidad de su estadio y el identificador de Clubp_id, el cual ocuparemos para poder unirla con las demás tablas.

2. Club_games: Esta tabla contiene todos los datos de cada partido disputado de cada equipo, esta tabla la podemos relacionar con las otras a partir de las columnas Game_id y Club_id que son los identificadores de equipo y partido.

3. Game_events: Ésta es la tabla más grande, cada registro es un evento importante de cada partido, ya sea gol o sustitución, esta la podemos relacionar con las demás con el campo Game_id.

4. Games: Esta tabla contiene información general de cada partido incluyendo la posición de cada equipo, número de goles, id de equipo local y visitante, esta tabla la podemos relacionar con las demás con los campos de game_id, home id, visit_id, que son los identificadores de partido, equipo local y equipo visitante respectivamente.

Después de agrupar y realizar el proceso de combinación de tablas, se realizó un proceso de ingeniería de variables en el cual se generó la variable objetivo, así como diversos cálculos entre las variables continuas, posteriormente se realizó una modificación de las variables categóricas.

Ingenieria de Variable:

La unidad muestral para este caso sera cada partido, el cual tendra la etiqueta de 0 si no se anotaron mas de dos goles en dicho partido y de 1, si se anotaron mas de dos goles, la cual tambien se tuvo que crear y fue parte de la ingenieria de variables.

Se realizaron ventanas de tiempo que permitieran describir el numero de goles anotados, victorias, derrotas, empates, etc, de cada equipo en los ultimos 10 partidos, distinguiendo entre local y visita.

Análisis Exploratorios de datos:

Se realizó un histograma de todas las variables numéricas continuas y una gráfica de pie para las variables discretas, a continuación se muestran las más interesantes.

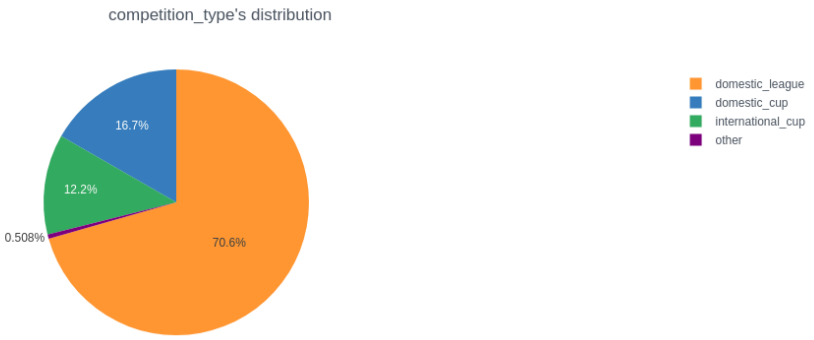

Figura 1. Distribución del tipo de competición, era de esperarse que la mayoría de los partidos fueran de la liga doméstica de cada equipo, únicamente el 12% de los partidos son internacionales.

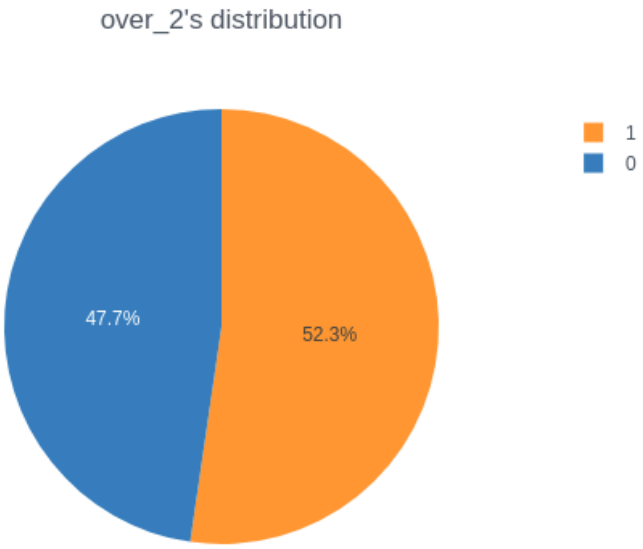

Figura 2. Distribución de la variable objetivo, se observa que no existe un desbalance de clases en nuestros datos.

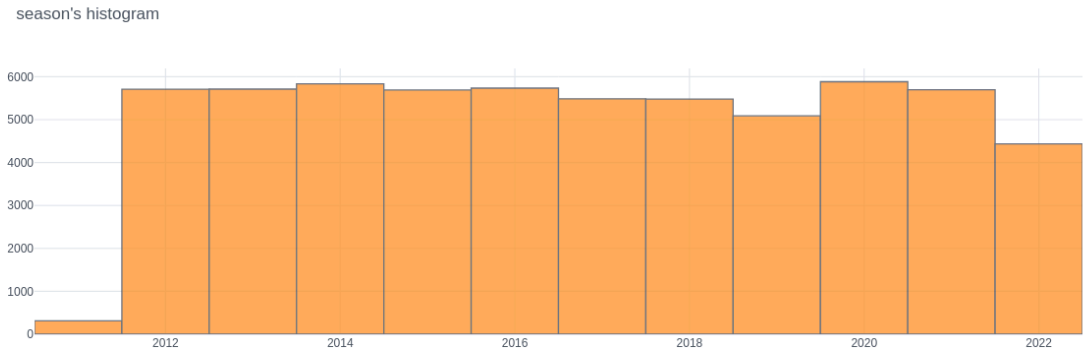

Figura 3. Histograma de las temporadas, podemos ver que los números de partidos que incluyen nuestros registros, son muy similares a partir del año 2012 y con una cantidad considerablemente menor en la temporada 2022-2023, que apenas está terminado, lo cual tiene bastante sentido por que aun no se sube esa información.

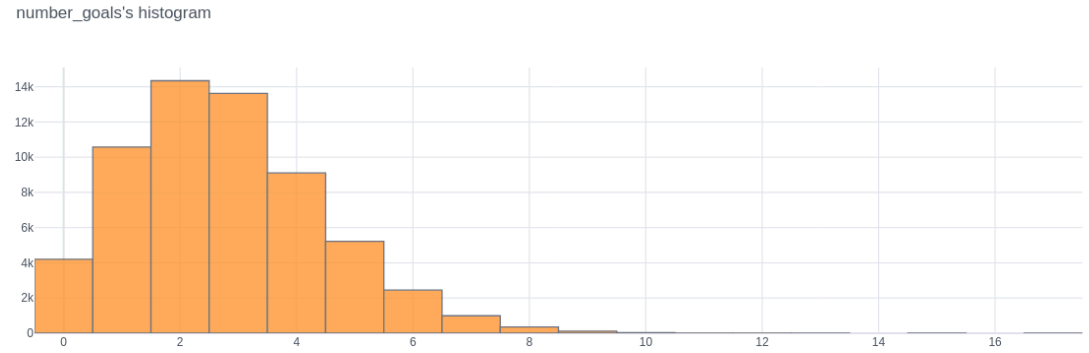

Figura 4. Distribución del total de goles, en los partidos, se observa que el total de goles más común es 2 goles. Recordemos que esta es la variable que queremos predecir.

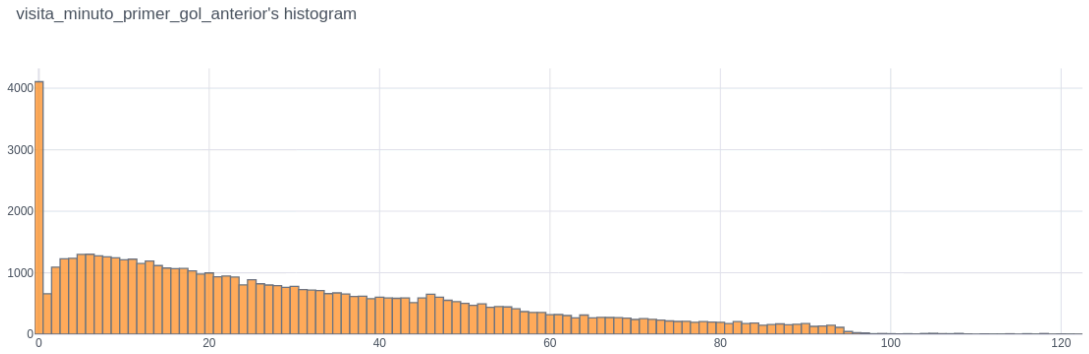

Figura 5. Distribución de los minutos en donde se anota el primer gol, se observa una clara tendencia a ser mayor en los primeros 20 minutos, en 0 significa que no hubo gol .

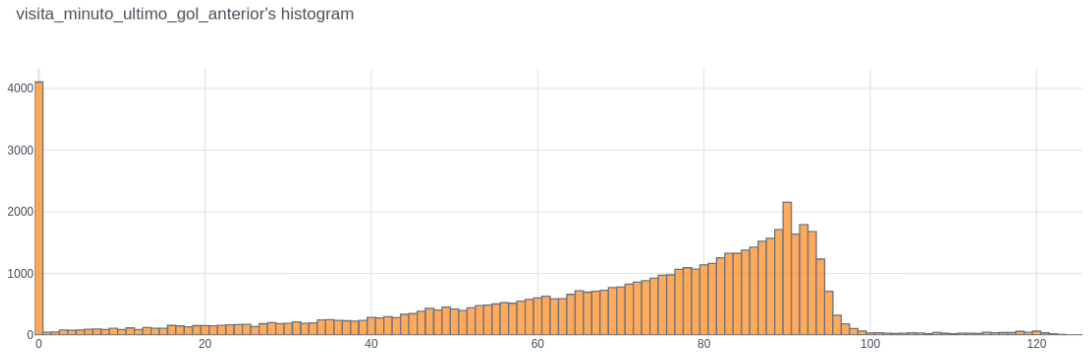

Figura 6. Distribución de los minutos en donde se anota el último gol, se observa una clara tendencia a ser mayor en los últimos 25-30 minutos, en 0 significa que no hubo gol .

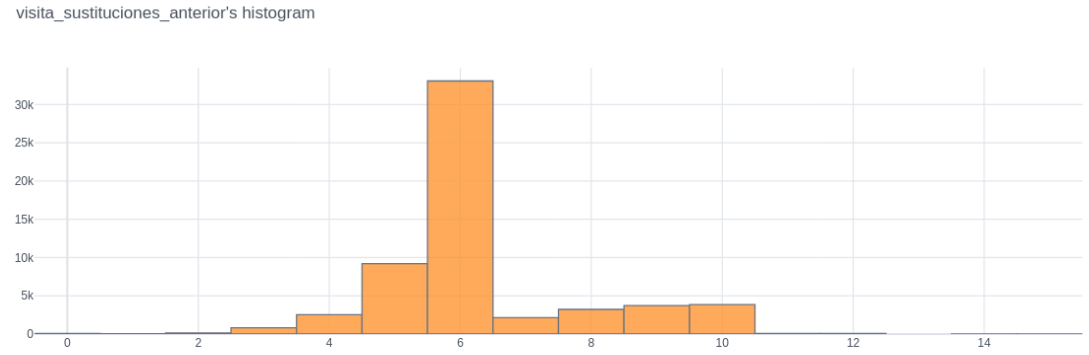

Figura 7. En general los partidos tuvieron 6 en total, lo cual sucede seguramente por que la regla anterior permite máximo a 3 por equipo, tal vez en algunos años este gráfico cambie.

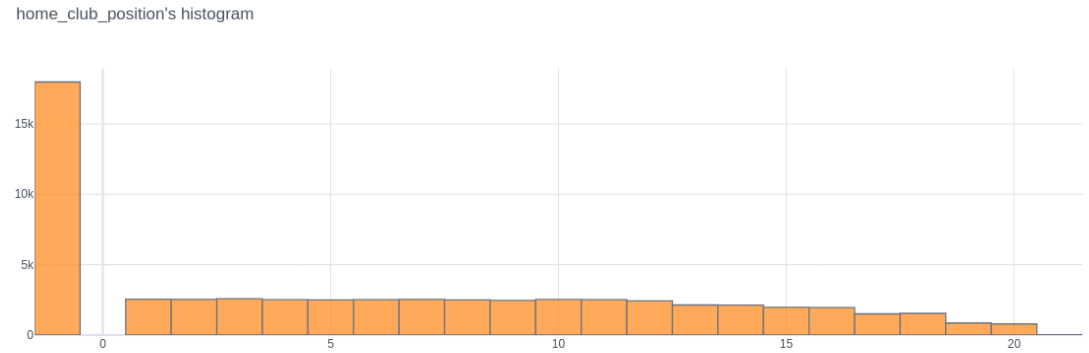

Figura 8. La distribución de la posición muestra que este campo tiene muchísimos valores que son -1, lo cual significa que no hay información, por lo que a esta variables se le tratará en la limpieza.

Así mismo se graficaron las distribuciones de las variables continuas dependiendo de la variable objetivo, pero en ninguna se encontraron diferencias considerables, por lo cual solo se incluye

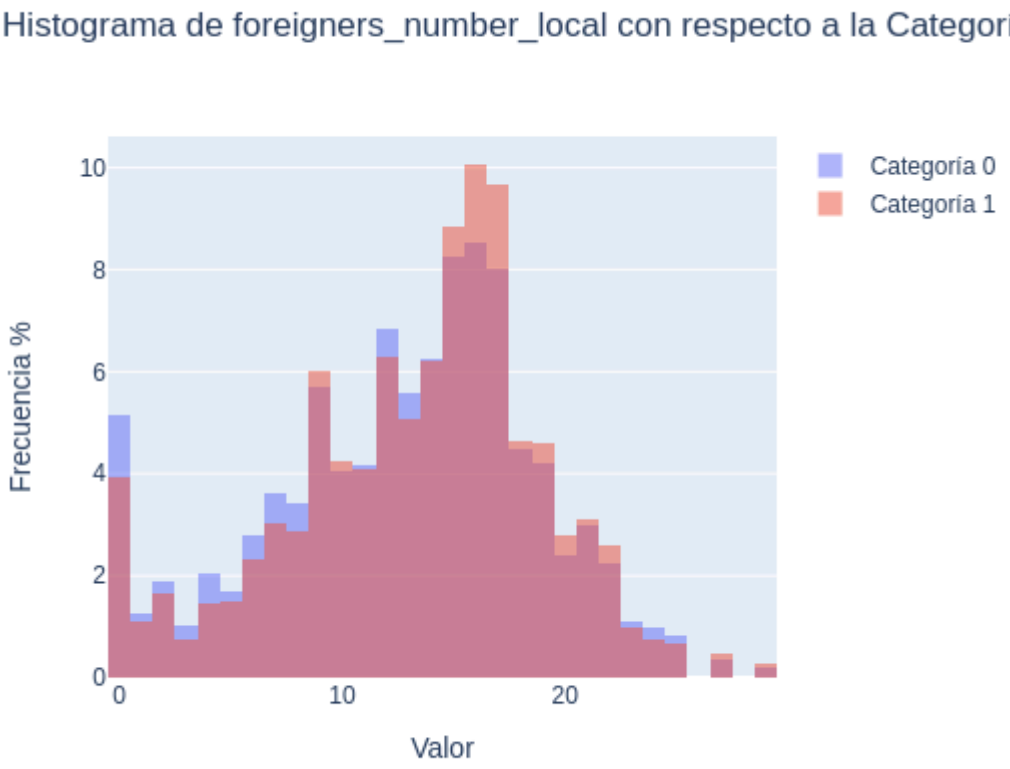

Figura 10. Distribución de la variable número de extranjeros del equipo local, con respecto a la variable objetivo.

Procesamiendto de datos realizado:

En este apartado vamos a encargarnos de que nuestros datos sean lo más confiables posibles, por lo que se realizarán algunos procesos que permitan garantizar esto.

Remoción de variables unarias:

Esto consiste en quitar las columnas que tengan el mismo valor para todos los registros, lo cual no nos aporta ninguna información relevante.

Haciendo un conteo de valores diferentes para cada variable, no se identifica ninguna variable que solo tenga un único valor.

Remoción de variables con alto contenido de valores ausentes:

Se hizo una prueba para identificar variables con un porcentaje mayor a 10% de valores ausentes.

Tratamiento de valores nulos:

Ahora seguimos teniendo variables con valores nulos pero en mucho menor proporción. Los demás valores nulos de tipo numérico haciendo un análisis, son el resultado de haber colocado los resultados de los partidos anteriores, esto produjo que los primeros 3 partidos de cada equipo tenga valores nulos ya que no tiene información anterior. Por lo que eliminar esos registros.

No se está imputando para este caso ningún valor nulo, por lo cual no fue necesario esperarnos a hacer la separación de conjuntos para hacer el tratamiento de los mismos.

Variables altamente correlacionadas:

Eliminar variables altamente correlacionadas en un conjunto de datos destinado a un modelo de machine learning ayuda a evitar redundancia, mejora la interpretación y eficiencia computacional, y puede aumentar la precisión predictiva del modelo, por lo que se hizo una prueba de correlación de todas las variables numéricas de nuestros datos para removerlas , pero no se encontró ninguna variable con correlación de 1, por lo que después de estas pruebas no fue necesario remover ninguna.

Tratamiento de valores extremos:

En este paso hicimos pruebas estadísticas que nos permitieran observar si tenemos registros de partidos atípicos que nos pudieran generar algún error, utilizamos la prueba de cuantiles para cada una de las variables de nuestra tabla, y se procedió a eliminar los registros que contenían más del 30% de sus variables detectada como atípica o rara.

Sabemos que no es muy común que haya 5 goles en tiempo de compensación por ejemplo, lo cual es muy probable que se deba a situaciones extremas muy particulares.

Una vez realizado este proceso de limpieza, se hizo la separación de nuestros conjuntos de entrenamiento y de prueba, se realizó un escalamiento tipo Min-Max scaler para que los modelos puedan ser más eficientes, terminando así el proceso de limpieza.

Selección de Variables:

Se decidió utilizar una selección de variables de acuerdo al poder predictivo con respecto a la variable objetivo, con la función Select K-best, seleccionando con la función f-classic, se hicieron varias pruebas modelando con regresión logística para ver cuántas variables nos ayudan a tener el mejor equilibrio entre varianza y sesgo, y después utilizaremos esas variables para entrenar modelos mucho más robustos.

El número de variables seleccionadas fue de 30.

1. attendance
2. own position
3. opponent position
4. visita goles anteriores recibidos
5. visita goles anteriores
6. visita_minuto_primer_gol_anterior
7. visita_minuto_ultimo_gol_anterior

8. visita_number_goals_anterior
9. visita_sustituciones_anterior
10. visita_goles_tiempo_añadido_anterior
11. visita_number_goals_anterior_2
12. suma_goles_recibidos
13. suma_goles_anotados
14. total_de_goles_anterior_visita
15. anotado_local_recibido_visita
16. anotado_visita_recibido_local
17. diferencia_sustituciones
18. suma_goles_anotados_2
19. total_de_goles_anterior_casa_2
20. total_de_goles_anterior_visita_2
21. anotado_visita_recibido_local_2
22. suma_goles_anotados_3
23. anotado_visita_recibido_local_3
24. visita_victoria_local_partidoanterior
25. over_2_anterior_visita
26. domestic_competition_id_local_GR1
27. domestic_competition_id_local_NL1
28. domestic_competition_id_visit_GR1
29. domestic_competition_id_visit_L1
30. domestic_competition_id_visit_NL1

Modelado:

Regresión Logistica:

El primer modelo que se probó con los datos fue una regresión logistica simple con parámetro “solver” Liblinear.

Obtuvimos las siguientes métricas:

Conjunto Train Conjunto Test

```
Roc Validate: 0.823
Acc Validate: 0.737
Matrix Conf Validate:
[[25798  9474]
 [ 9806 28204]]
              precision    recall  f1-score   support

           0       0.72      0.73      0.73     35272
           1       0.75      0.74      0.75     38010

    accuracy                           0.74     73282
   macro avg       0.74      0.74      0.74     73282
weighted avg       0.74      0.74      0.74     73282
```

```
Roc Validate: 0.824
Acc Validate: 0.736
Matrix Conf Validate:
[[12765  4747]
 [ 4914 14216]]
              precision    recall  f1-score   support

           0       0.72      0.73      0.73     17512
           1       0.75      0.74      0.75     19130

    accuracy                           0.74     36642
   macro avg       0.74      0.74      0.74     36642
weighted avg       0.74      0.74      0.74     36642
```

Podemos observar que es un modelo bastante estable obteniendo prácticamente las mismas métricas en ambos conjuntos, y tiene resultados bastante aceptables considerando el problema que se plantea.

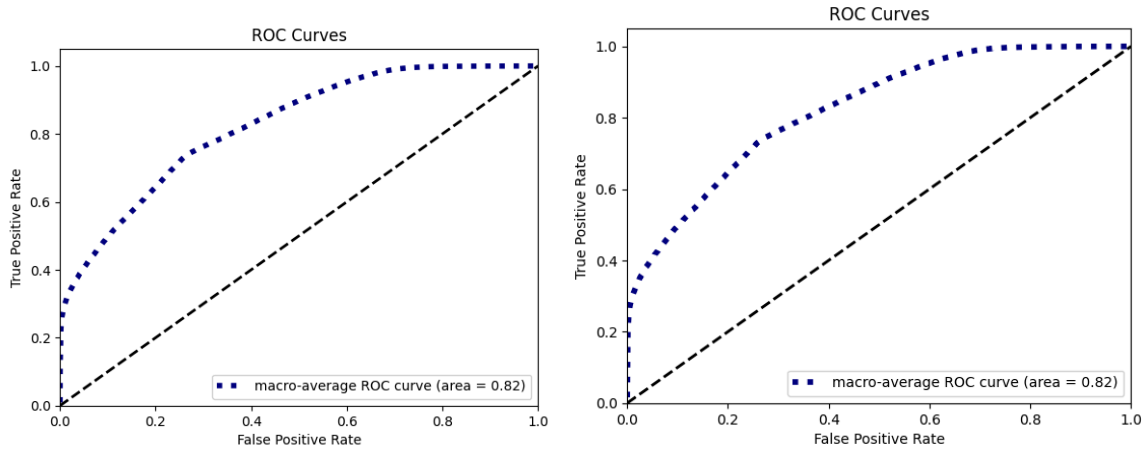

Figura 11. Curvas Roc para los conjuntos de entrenamiento y validación, respectivamente.

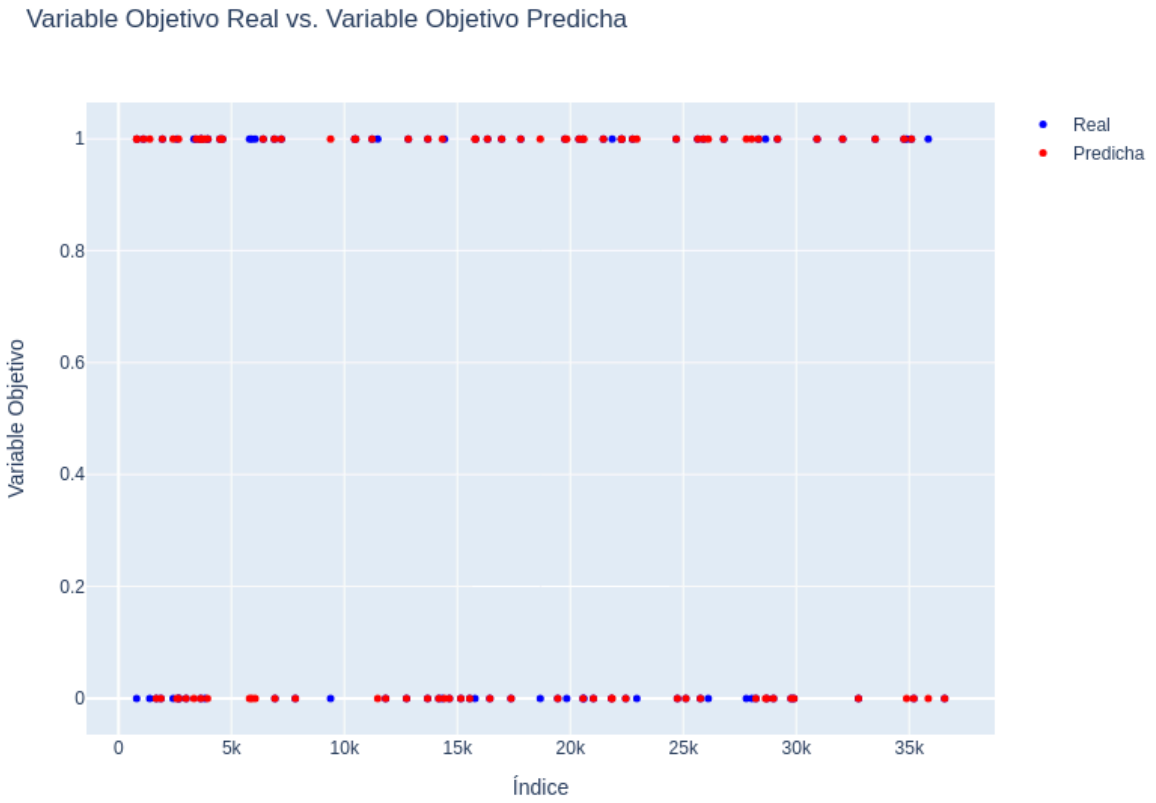

Figura 12. Gráfica de los valores predichos vs los valores reales.

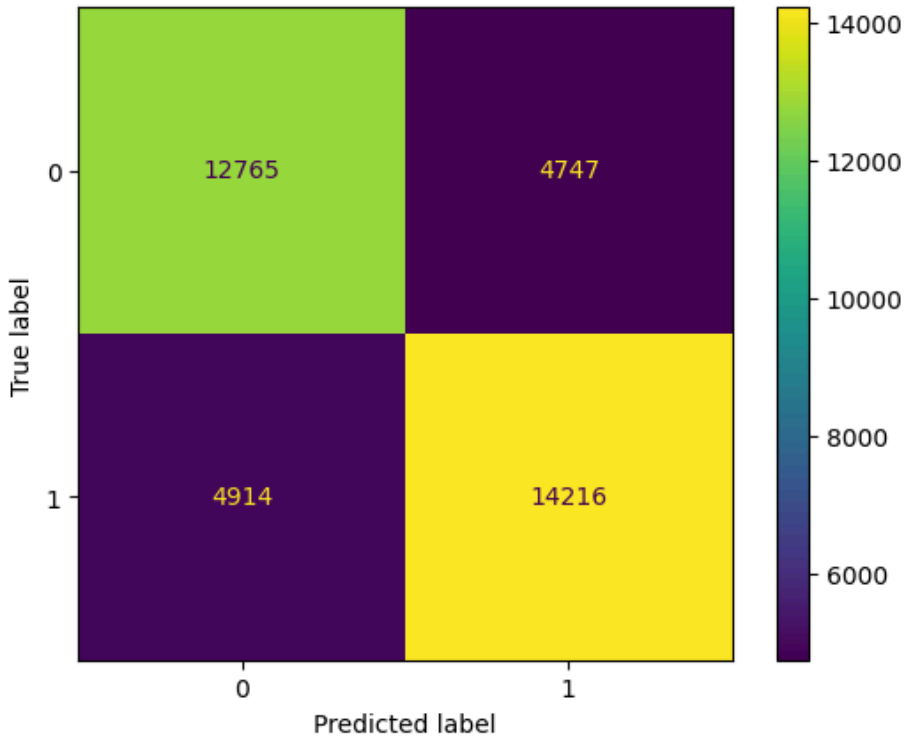

Figura 13. Matriz de confusión para el conjunto de validación.

XG Boost:

Para este caso, se utilizó hiper parametrización con búsqueda tipo Random Search para no tener tiempos de ejecución tan altos.

Obteniendo las siguientes métricas:

Conjunto Train

```
Roc Validate: 0.900
Acc Validate: 0.792
Matrix Conf Validate:
[[27308  7964]
 [ 7303 30707]]
              precision    recall  f1-score   support

           0       0.79      0.77      0.78     35272
           1       0.79      0.81      0.80     38010

    accuracy                           0.79     73282
   macro avg       0.79      0.79      0.79     73282
weighted avg       0.79      0.79      0.79     73282
```

Conjunto Test

```
Roc Validate: 0.873
Acc Validate: 0.764
Matrix Conf Validate:
[[13005  4507]
 [ 4158 14972]]
              precision    recall  f1-score   support

           0       0.76      0.74      0.75     17512
           1       0.77      0.78      0.78     19130

    accuracy                           0.76     36642
   macro avg       0.76      0.76      0.76     36642
weighted avg       0.76      0.76      0.76     36642
```

Podemos observar métricas ligeramente mejores, pero una diferencia entre conjunto de validación y entrenamiento mayor, siendo menos estable y tal vez se trate de un ligero sobreajuste.

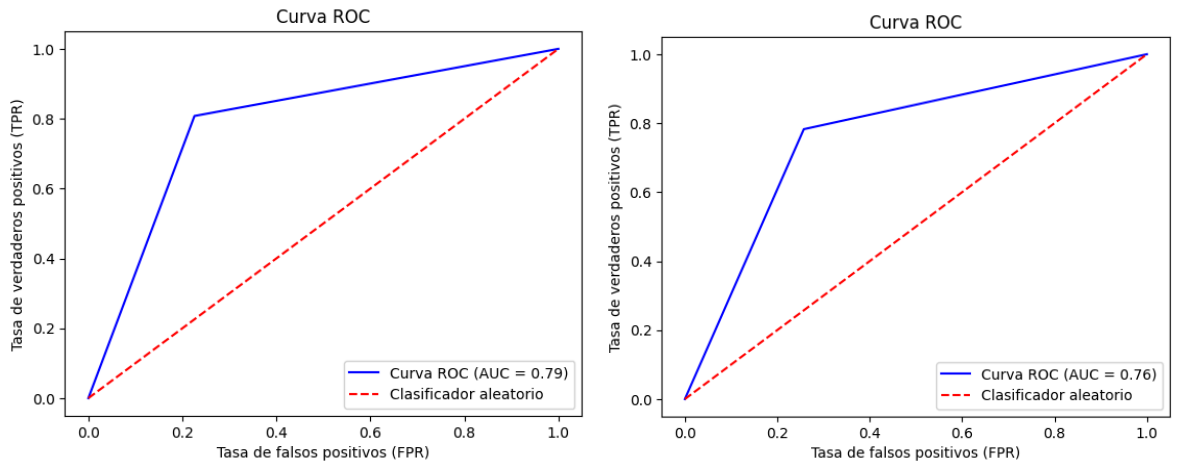

Figura 14. Curvas Roc del conjunto de entrenamiento y validación.

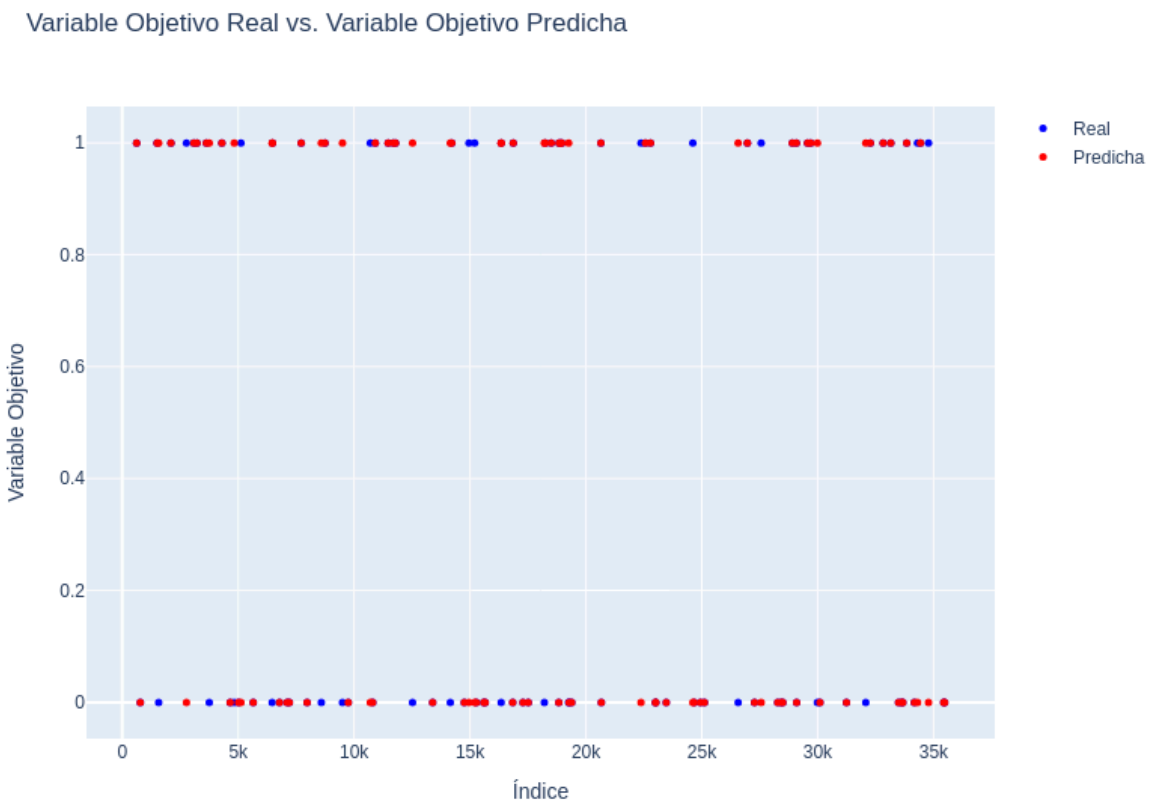

Figura 15. Gráfica de los valores predichos vs los valores reales.

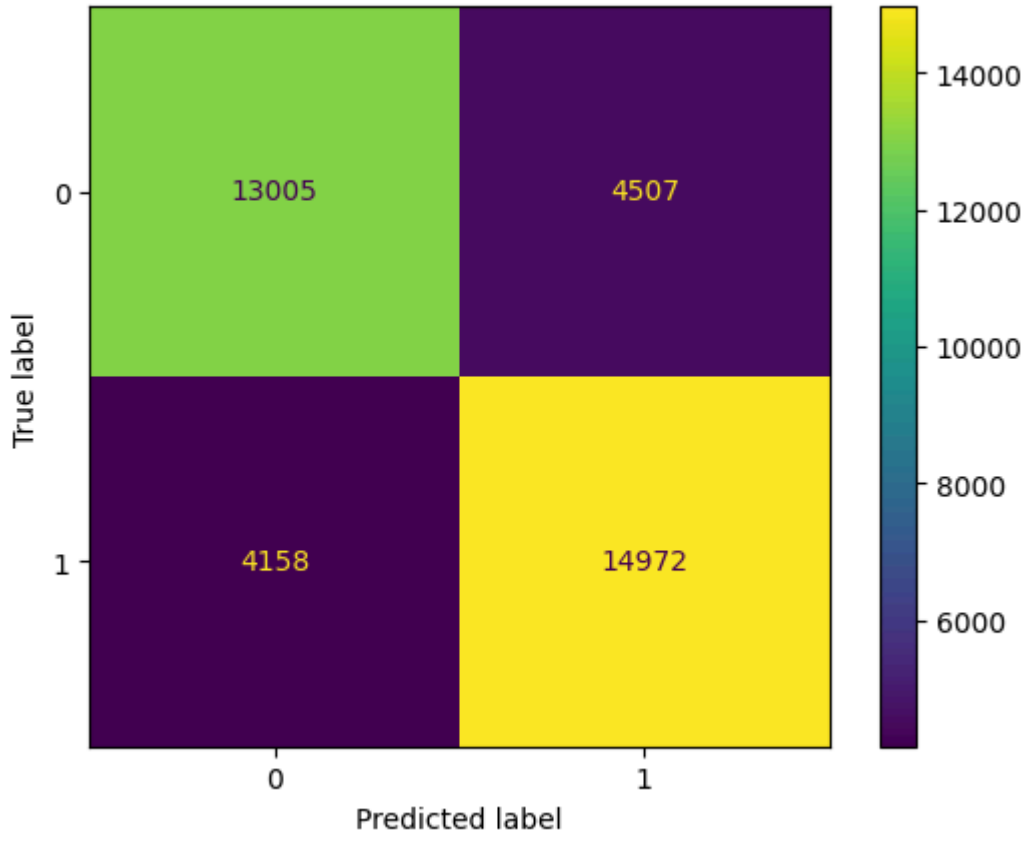

Figura 16. Matriz de confusión para el conjunto de validación

SGDC:

Para este modelo se utilizó una hiper parametrización con búsqueda de gradilla, siendo el modelo que más tardó en entrenar.

Obteniendo las siguientes métricas:

Conjunto Train

```
Roc Validate: 0.826
Acc Validate: 0.750
Matrix Conf Validate:
[[25806  9466]
[ 8883 29127]]
              precision    recall  f1-score   support

           0       0.74      0.73      0.74     35272
           1       0.75      0.77      0.76     38010

    accuracy                           0.75     73282
   macro avg       0.75      0.75      0.75     73282
weighted avg       0.75      0.75      0.75     73282
```

Conjunto Test

```
Roc Validate: 0.826
Acc Validate: 0.748
Matrix Conf Validate:
[[12766  4746]
[ 4470 14660]]
              precision    recall  f1-score   support

           0       0.74      0.73      0.73     17512
           1       0.76      0.77      0.76     19130

    accuracy                           0.75     36642
   macro avg       0.75      0.75      0.75     36642
weighted avg       0.75      0.75      0.75     36642
```

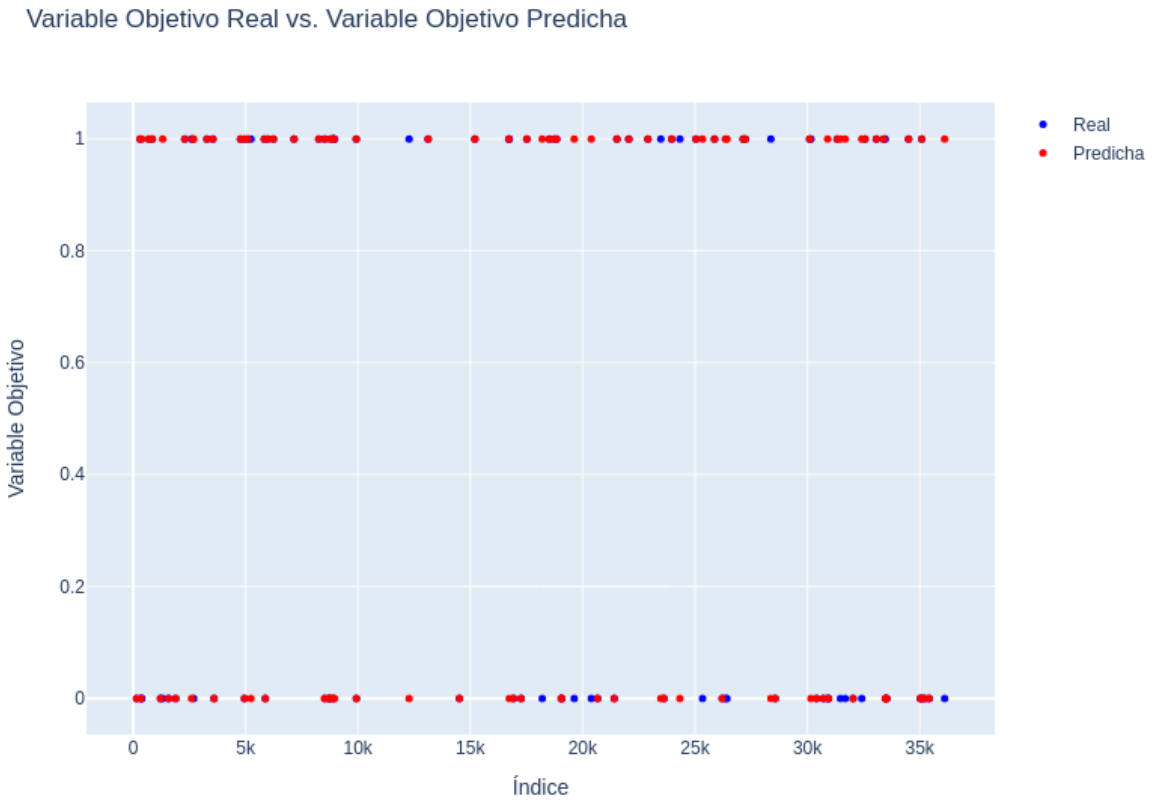

Figura 17. Gráfica de los valores predichos vs los valores reales.

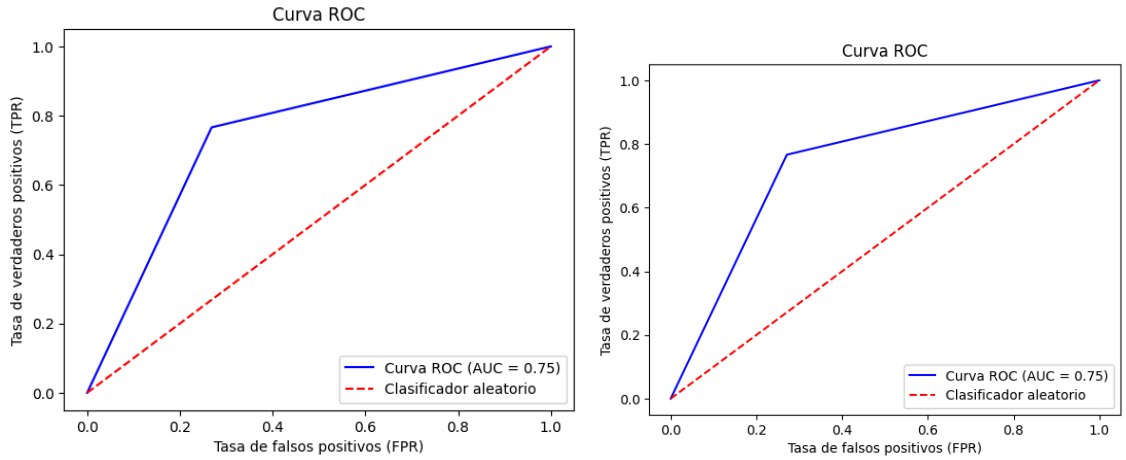

Figura 18. Curvas Roc para el conjunto de entrenamiento y validación.

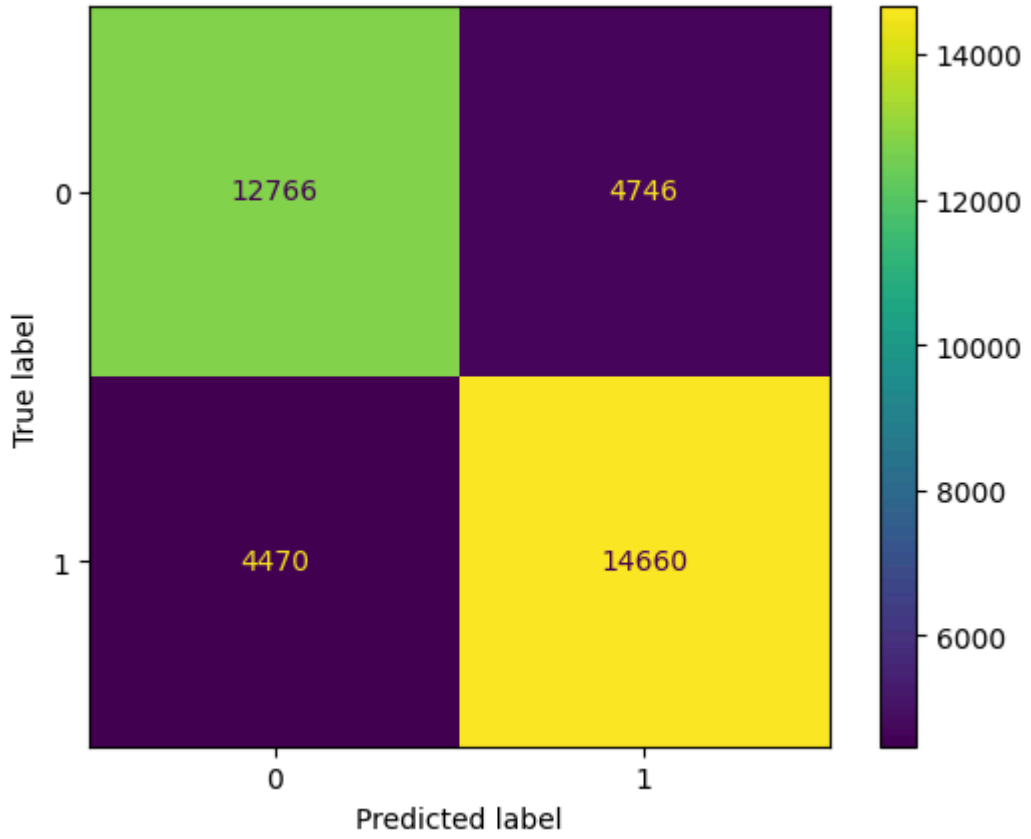

Figura 19. Matriz de confusión para el conjunto de validación.

Ridge Classifier:

Para este modelo se utilizó igualmente una hiper parametrización con búsqueda de gradilla, dando como mejores hiperparametros alpha=0.1 y solver=lsqr.

	precision	recall	f1-score	support
0	0.74	0.74	0.74	35272
1	0.76	0.76	0.76	38010
accuracy			0.75	73282
macro avg	0.75	0.75	0.75	73282
weighted avg	0.75	0.75	0.75	73282

	precision	recall	f1-score	support
0	0.74	0.74	0.74	17512
1	0.76	0.76	0.76	19130
accuracy			0.75	36642
macro avg	0.75	0.75	0.75	36642
weighted avg	0.75	0.75	0.75	36642

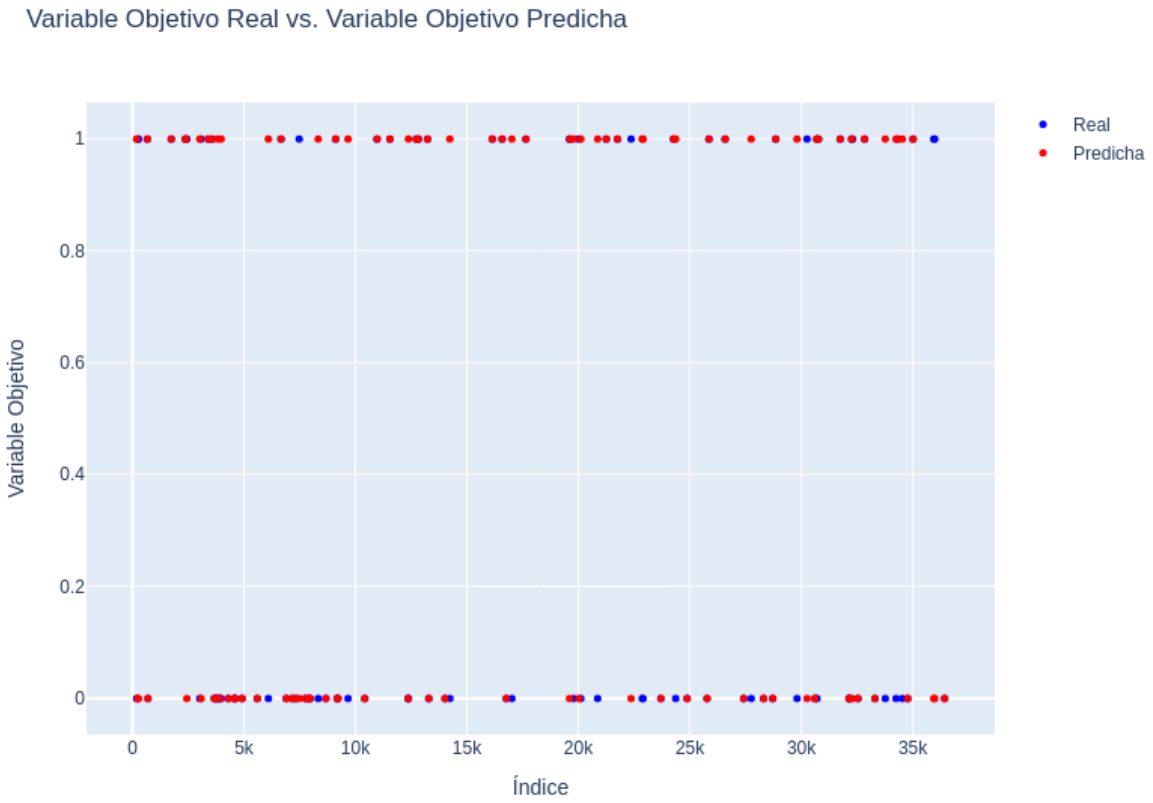

Figura 20. Gráfica de los valores predichos vs los valores reales.

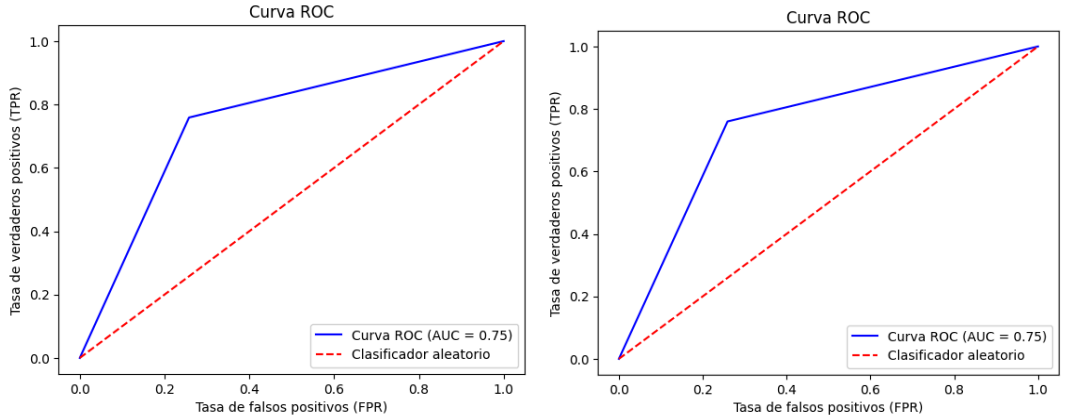

Figura 21. Curvas Roc para el conjunto de entrenamiento y validación.

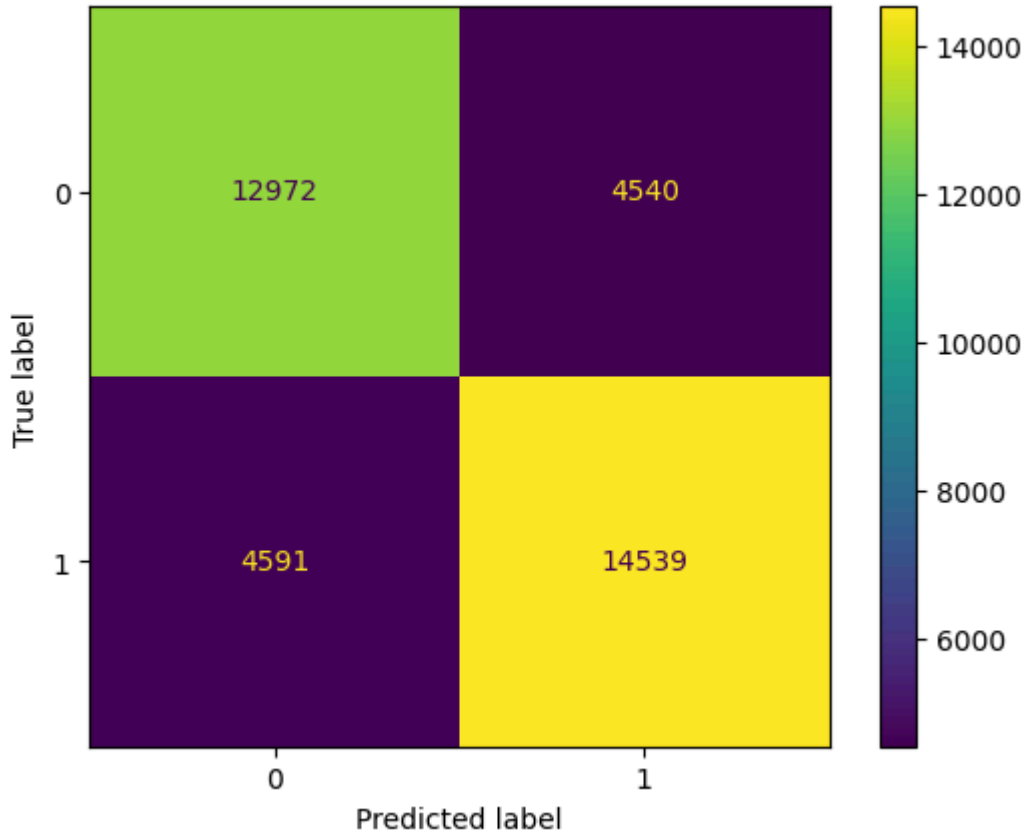

Figura 22. Matriz de confusión para el conjunto de validación.

Máquinas de vector soporte:

Para este modelo, se entrenó simplemente usando los valores que tiene la función por defecto.

Obteniendo las siguientes métricas:

Conjunto Train

	precision	recall	f1-score	support
0	0.75	0.75	0.75	35272
1	0.77	0.77	0.77	38010
accuracy			0.76	73282
macro avg	0.76	0.76	0.76	73282
weighted avg	0.76	0.76	0.76	73282

Conjunto Test

	precision	recall	f1-score	support
0	0.75	0.75	0.75	17512
1	0.77	0.77	0.77	19130
accuracy			0.76	36642
macro avg	0.76	0.76	0.76	36642
weighted avg	0.76	0.76	0.76	36642

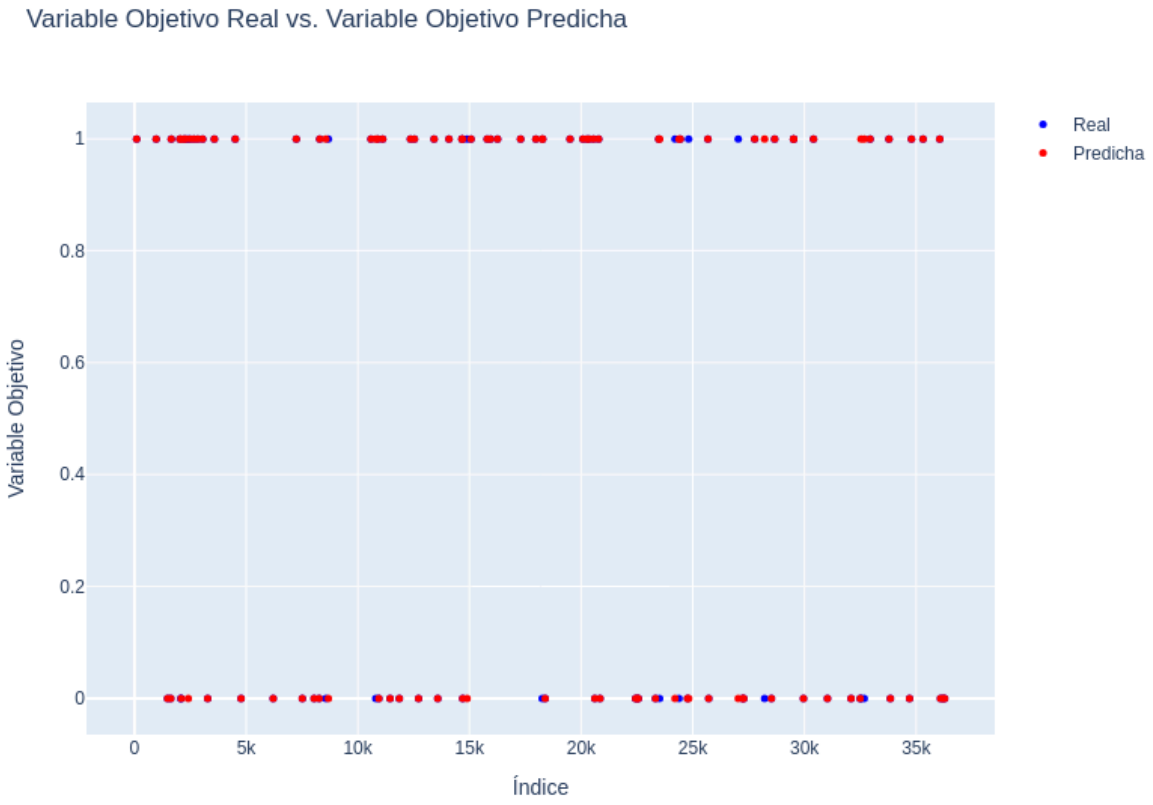

Figura 23. Gráfica del valor real vs el valor predicho.

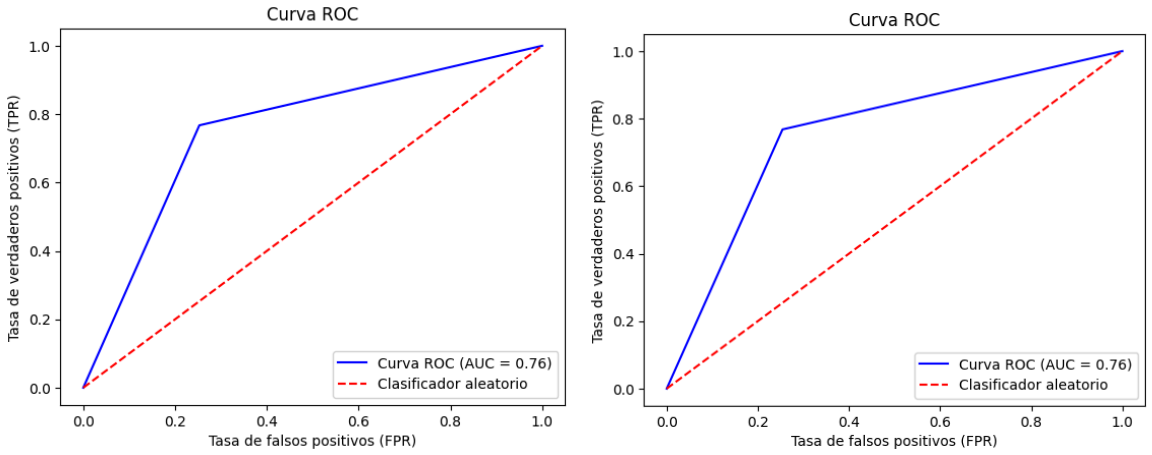

Figura 24. Curvas Roc para conjunto de entrenamiento y validación.

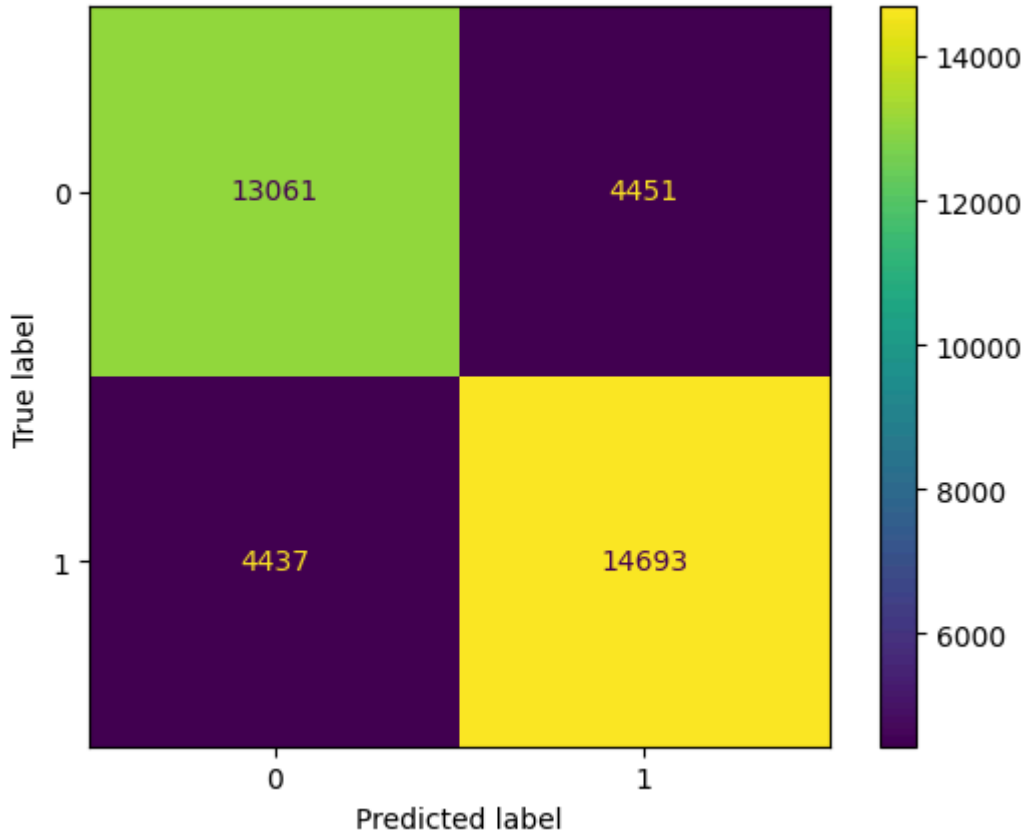

Figura 25. Matriz de Confusión para el conjunto de validación.

Resultados:

Elección del Modelo:

Todos los modelos probados nos dan métricas aceptables de acuerdo con la problemática planteada, dando un accuracy score entre 0.74 y 0.76, siendo los más altos el XG Boost y el Vector máquina soporte.

El mejor modelo probado fue el de **Vector máquina soporte**, ya que si bien tiene las mismas métricas en el conjunto test que el XG boost, este modelo tiene mayor estabilidad, presentando casi las mismas métricas en ambos conjuntos, a diferencia que el boost que tiene métricas ligeramente mayores para el conjunto de entrenamiento.

Implementación del modelo:

El modelo está concebido para su aplicación en la toma de decisiones relacionadas con apuestas en casinos digitales. El usuario final de este modelo es el investigador mismo, quien busca aprovechar la información recopilada y procesada para realizar apuestas informadas y, en consecuencia, obtener beneficios económicos.

La metodología de implementación del modelo se simplifica en el siguiente procedimiento. En primera instancia, se descargan los datos más recientes de Transfermarkt.com en formato CSV. Posteriormente, se realiza un procesamiento exhaustivo de los datos de la misma manera que fueron tratados los datos de entrenamiento. Luego, se emplea el modelo entrenado para realizar predicciones sobre la probabilidad de que un partido supere los 2 goles. Esta información se utiliza como base para la toma de decisiones en el ámbito de las apuestas. El ciclo de aplicación del modelo se renueva diariamente con la actualización de datos más recientes.

Local	Visita	over_2.5
Goverla Uzhgorod	Zorya Lugansk	1
Silkeborg If	Odense Boldklub	0
Zorya Lugansk	Goverla Uzhgorod	0
Odense Boldklub	Silkeborg If	0
Dynamo Kiew	Kryvbas Kryvyi Rig	1
Kryvbas Kryvyi Rig	Dynamo Kiew	0
Karpaty Lviv	Sk Tavriya Simferopol	1
Fc Nordsjaelland	Aalborg Bk	1
Fc Kopenhagen	Esbjerg Fb	1
Esbjerg Fb	Fc Kopenhagen	1
Aalborg Bk	Fc Nordsjaelland	0
Sk Tavriya Simferopol	Karpaty Lviv	1
Aarhus Gf	Brondby If	1
Brondby If	Aarhus Gf	1
Vorskla Poltava	Chornomorets Odessa	1
Metalurg Zaporizhya Bis 2016	Fk Mariupol	1

Chornomorets Odessa	Vorskla Poltava	1
Fc Midtjylland	Randers Fc	0
Fk Mariupol	Metalurg Zaporizhya Bis 2016	0
Dnipro Dnipropetrovsk	Arsenal Kiew	0
Randers Fc	Fc Midtjylland	0

Figura 25. Tabla con predicciones de partidos futuros, si el valor es cero significa que la apuesta es al under 2.5 y si es 1 la apuesta debe ser al over 2.5.

Conclusiones:

Los objetivos fueron cubiertos satisfactoriamente, toda vez que la problemática era muy compleja, al tratar de predecir un evento que depende de muchos factores y variables, incluida la suerte, y a simple vista como se muestra en el análisis exploratorio de datos, no se observan patrones obvios o fáciles de identificar a simple vista, por lo que se tuvo que crear un elevado número de variables para lograr que los modelos encontrarán la relación de los datos con la variable objetivo.

Dicho lo anterior, tener una confianza del 76% en que nuestra apuesta será correcta es un resultado más que favorable, y con un poco más de exhaustividad en la ingeniería de variables variables, limpieza, tratamiento de datos en conjunto con la implementación de modelos más robustos, se podrá ir incrementando esa confiabilidad en nuestros resultados gradualmente, pudiendo así ser aplicado en un futuro a apuestas en tiempo real, en una primer instancia para el over de 2 goles en específico, y así obtener beneficios económicos a partir de este proyecto.

Bibliografía:

Datos:

Transfermarkt. (2023). Recuperado de https://www.transfermarkt.com/