Seminario de Ingeniería Industrial IV
Curso Industrias Digitales

Trabajo Práctico - Exploración, Visualización de datos y Machine Learning
Predicción Goles Dataset

Nombre	Apellido	Padrón
Lucas Mateo	Gimenez	105237
Matias	Weinstabl	104014
Agustín	Buttini	104355
Luciano	Bianchi	104944
Pedro	García Rico	104050
Bautista	Pazos	104329

Grupo 2

Corrector: Ariel Schwartz

Introducción

El siguiente trabajo práctico consiste en explorar, limpiar, visualizar y analizar el dataset "Goles Esperados", en el cual se busca predecir si un tiro al arco terminará en gol o no. Para realizar el trabajo utilizaremos Python como herramienta que facilitará las tareas de exploración y limpieza de datos, análisis estadístico, visualización y modelado de Machine Learning.

Consignas

1) Explorar el dataset “Goles Esperados”, el cuál será utilizado para predecir si un tiro al arco terminará o no en gol.
a) Explicar la cantidad de filas y columnas que tiene el dataset, cuantas son numéricas y cuantas categóricas.

En el contexto de análisis de datos y machine learning, es fundamental distinguir entre columnas numéricas y categóricas porque afecta cómo se manipulan, analizan y modelan los datos.
Las columnas numéricas contienen datos que son números (pueden ser enteros o decimales).
Las columnas categóricas contienen datos que son categorías o etiquetas, utilizadas para clasificar y agrupar los datos.

El dataset “Goles Esperados” contiene 68194 filas y 25 columnas.

Columnas numéricas: 11.
Columnas categóricas: 14.

```
Cantidad de filas: 68194
Cantidad de columnas: 25
Columnas numéricas: Index(['Unnamed: 0', 'id', 'start_x', 'start_y', 'minute', 'end_x', 'end_y',
       'blocked_x', 'blocked_y', 'is_own_goal', 'match_id'],
      dtype='object')
Columnas categóricas: Index(['shot_outcome', 'player', 'date', 'area', 'from_throw_in', 'shot_type',
       'from_corner', 'from_freekick', 'from_open_play', 'from_fastbreak',
       'from_penalty', 'from_setpiece', 'from_cross_open_play',
       'shot_on_post'],
      dtype='object')
```

b) Responder las siguientes preguntas:
— ¿Existen valores vacíos?

```
Valores vacíos:
Unnamed: 0                0
id                        0
start_x                  30
start_y                  30
minute                    0
end_x                 68194
end_y                 68194
blocked_x             32419
blocked_y             32419
shot_outcome              0
player                 4604
is_own_goal           68023
match_id              53292
date                  53292
area                      0
from_throw_in             0
shot_type                 0
from_corner               0
from_freekick             0
from_open_play            0
from_fastbreak            0
from_penalty              0
from_setpiece             0
from_cross_open_play      0
shot_on_post              0
```

Hay un total de 10 columnas que contienen al menos un valor vacío.

— ¿Cuál es la media/desvío estándar/mediana/mínimo y máximo de cada variable numérica?

Se obtiene lo solicitado mediante el método describe() para las columnas relevantes.

	start_x	start_y	minute	blocked_x	blocked_y
count	68164.000000	68164.000000	68194.000000	35775.000000	35775.000000
mean	83.395236	50.025833	49.921430	92.233001	49.987617
std	9.262727	13.562249	28.101046	6.961193	9.576609
min	0.500000	0.400000	0.000000	1.200000	0.000000
25%	76.500000	40.575000	26.000000	87.500000	46.400000
50%	85.600000	50.000000	51.000000	94.600000	49.800000
75%	90.300000	59.400000	74.000000	98.200000	53.600000
max	99.800000	99.500000	146.000000	100.000000	100.000000

— ¿Cuántos valores únicos/distintos tiene cada una de las columnas?

```
Valores únicos por columna:
Unnamed: 0              68194
id                      63594
start_x                   646
start_y                   870
minute                    128
blocked_x                 262
blocked_y                 507
shot_outcome                5
player                   2114
is_own_goal                 1
match_id                  602
date                      192
area                        3
from_throw_in               2
shot_type                   3
from_corner                 2
from_freekick               2
from_open_play              2
from_fastbreak              2
from_penalty                2
from_setpiece               2
from_cross_open_play        1
shot_on_post                2
```

Unnamed = 68194.
id = 63594.

La diferencia, 68194 - 63594 = 4600, implica la cantidad de valores duplicados.

Realizamos un análisis notando que en los registros duplicados en uno figura el nombre y apellido del jugador en el campo “player” mientras que en el otro figura “NaN” en el campo “player”, valor nulo.

```
Valores duplicados en la columna id asociados a los jugadores:
                id          player
40664  2.311100e+09     Alan Varela
45268  2.311100e+09             NaN
40665  2.311100e+09  Agustín Obando
45269  2.311100e+09             NaN
40666  2.311100e+09     Alan Varela
...             ...             ...
45265  2.340768e+09   Brian Aguirre
49869  2.340770e+09             NaN
45266  2.340770e+09      José López
45267  2.340771e+09  Cristian Pavón
49870  2.340771e+09             NaN

[9200 rows x 2 columns]
```

2) Proceso de limpieza. Todo dataset debe ser limpiado y preparado para realizar un análisis. Por lo general existen columnas vacías, filas con datos faltantes, columnas de mayor interés que otras. ¿Qué trabajo realizarías? ¿Existen valores faltantes? Completar los "vacíos", removerlos o en caso de incluirlos explicar por qué se dejan en el dataset. Por otro lado, seleccionar las columnas que pueden ser de interés para el modelo de predicción y crear nuevas variables a partir de las ya existentes ¿Cuáles se te ocurren?.

El proceso de limpieza de un dataset en el contexto de machine learning se realiza para asegurar que los datos estén completos, correctos, coherentes y listos para ser utilizados en el análisis o modelos de machine learning.

Con respecto al dataset "Goles Esperados" se debe realizar un trabajo de limpieza para los 4600 casos de tiros duplicados, debido a que constituyen un número de registros considerable, capaz de sesgar el modelo. También, se eliminará el registro de 'id' duplicado que contiene valor nulo ("NaN") en el campo "player". De este proceso resultan 4 registros con este campo vacío.

En primer lugar, eliminaremos las columnas ***end_x*** y ***end_y***, ya que no se tiene ningún dato en estos campos ("NaN") y no podemos obtener información para el análisis, por lo tanto, no son útiles.

Adicionalmente, eliminaremos ***match_id*** y ***date***. A pesar de tener aproximadamente 18000 filas, se opta por omitir un estimado de xG por partido y/o por fecha.

Las columnas ***blocked_x*** y ***blocked_y*** indican las coordenadas donde el disparo fue bloqueado. A pesar de que en los tiros convertidos o errados el campo salga nulo ("NaN"), no podemos eliminar estas columnas ya que brindan información sobre los tiros atajados o bloqueados. Tampoco podemos llenar los campos nulos con algún valor porque implicaría afirmar que esos tiros fueron bloqueados cuando eso no ocurre.

Consideramos las siguientes variables de interés:

- ***shot_outcome***, que indica el resultado que tuvo el disparo efectuado, tendrá un peso considerable en el cálculo de xG de un tiro.
- ***start_x*** y ***start_y*** indican las coordenadas donde se realizó el disparo.
- ***Variables booleanas***: informan de donde provino el tiro, aclaran la situación de juego.
- ***shot_on_post***, que indica si un tiro pega en el palo, también aumentará el score de xG, dado que significa que un tiro tiene dirección al arco y no fue bloqueado.

Con respecto a nuevas variables, consideramos que ***angulo_de_tiro***, que indica el ángulo de las rectas que se forman entre la posición del tiro y los palos del arco (es mayor cuanto más cerca y centrado se encuentra), y ***distancia_de_tiro***, que serán relevantes para el cálculo de la probabilidad de gol.

3) Visualización y análisis de los datos. Responder y analizar.
— ¿Cuál es el promedio y mediana distancia de los disparos? Recuerden la fórmula de distancia entre 2 puntos para calcular la distancia de los disparos.

Realizando el cálculo en Python mediante las funciones mean() y median() para el campo “distancia_de_tiro” se obtiene:

- ➔ Media de distancia de tiro: 21,2 metros.
- ➔ Mediana de distancia de tiro: 20,9 metros.

— Sacar la cantidad de disparos que ocurrieron en el área chica, área grande y fuera del área y calcular cuantos de estos fueron gol y no (en cantidad y en porcentaje). (ayuda: usar groupby)

Mediante “group_by(‘area’)” se logra saber la cantidad de tiros al arco clasificados por el lugar de su ejecución.
Adicionalmente, se realiza el filtro “shot_outcome==‘Goal’’ para saber cuántos de estos tiros fueron convertidos, obteniendo así los siguientes resultados:

	Tiros	**Goles**	**Efectividad (%)**
Fuera del Área	28225	842	2,98
Área Grande	31541	3795	12,0
Área Chica	3481	1132	32,5

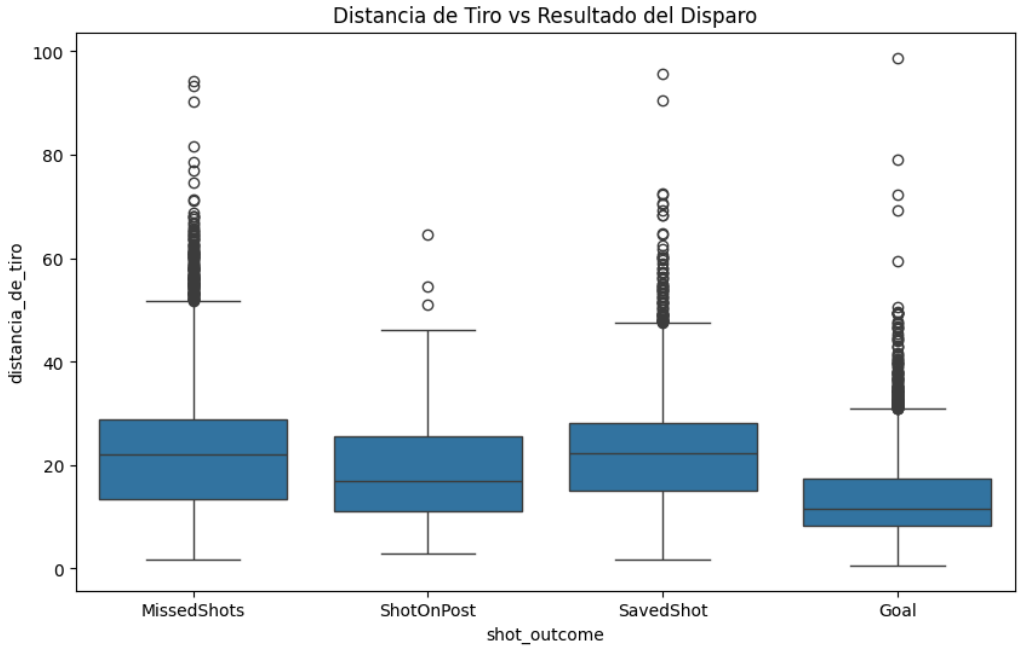
— Analizar y justificar si los goles en contra del dataset deben ser considerados o no para el modelo.

Para un modelo de goles esperados, no deben considerarse los goles en contra, ya que no consisten de una ocasión de gol generada por el equipo atacante, sino como un error del equipo que defiende. Por esto, toman un valor de cero en la suma de Goles Esperados (xG). No serán considerados por el algoritmo de predicción.

— ¿Cómo se podría visualizar si una variable impacta en el resultado de Y?

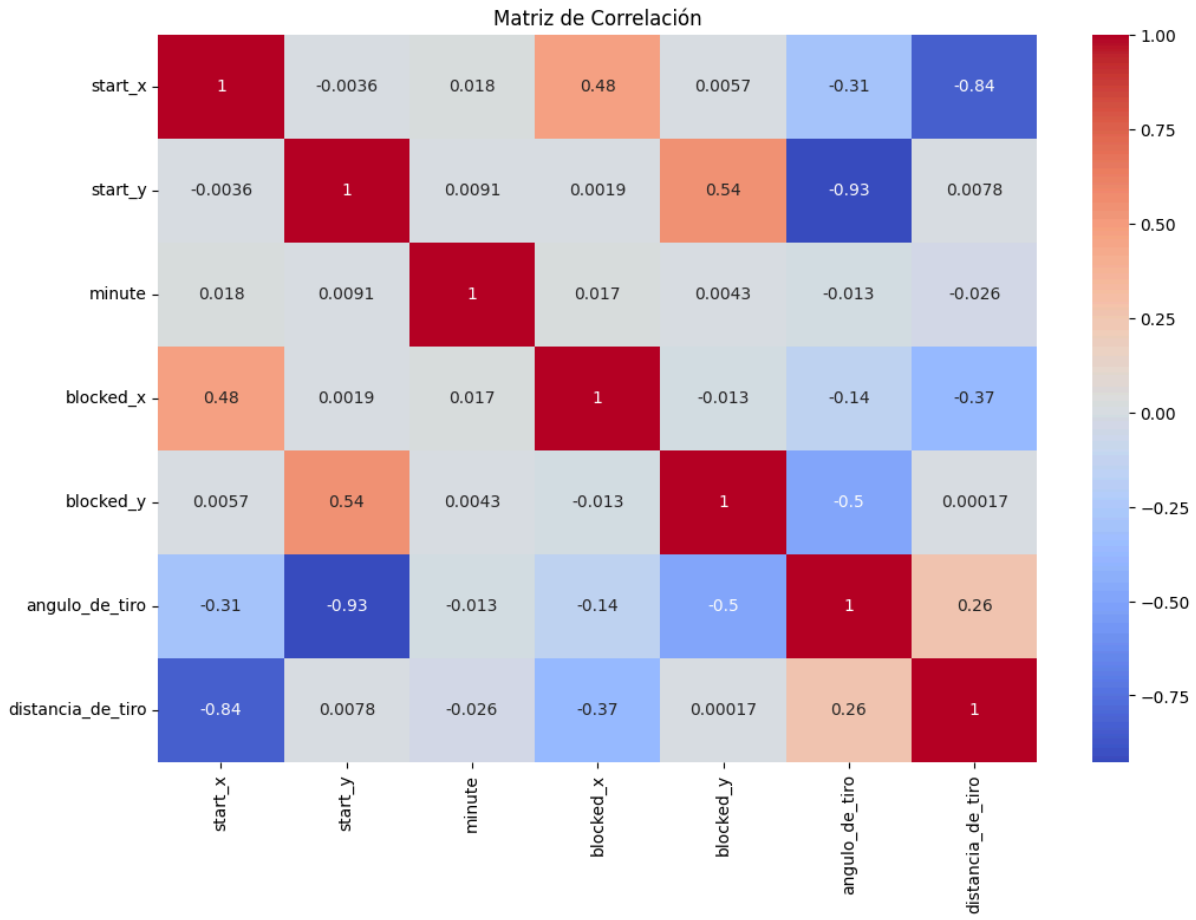
Para visualizar la relación entre una variable continua y el resultado del modelo (‘shot_outcome’), siendo éste una variable categórica, se utiliza un diagrama de

cajas (‘boxplot’), ya que permite observar la distribución de datos de las distintas categorías.
Si analizamos la variable “distancia_de_tiro”, es posible observar que los tiros que resultaron en gol tienen en promedio una menor distancia al arco que el resto de los resultados (tiro errado, tiro en el poste y tiro atajado).



— ¿Existe correlación entre variables? ¿Hay mejores variables para predecir la variable Y? (ayuda: usar matriz de correlación y un heatmap).

Una matriz de correlación en el contexto de machine learning es una tabla que muestra las correlaciones entre todas las variables de un dataset.
Esta matriz es fundamental para comprender cómo están relacionadas las diferentes características entre sí dentro de los datos.
A continuación, se aprecia la matriz de correlación para este ejercicio:



Coeficientes relevantes de correlación:

blocked_x - start_x = 0,48. Se ve una correlación positiva moderada entre la posición de salida de un tiro y la posición de bloqueo (en caso de ser bloqueado). Esto es lógico, ya que si se realiza un disparo cerca del arco que resulta bloqueado, la posición del bloqueo será incluso más cerca del arco que el mismo tiro.

blocked_y - start_y = 0,54. Se aprecia análogamente el caso de Y. Cabe aclarar que un disparo puede ser bloqueado más cerca de su posición de salida cuando se está próximo al área/arco, y es posible que sea por esta razón que el factor de correlación no sea tan elevado.

angulo_de_tiro - start_y = -0,93. Al alejarse del arco en el eje Y, es esperable tener menos ángulo de tiro. Por esto la correlatividad negativa es muy elevada.

angulo_de_tiro - blocked_y = -0,5. La correlación puede explicarse a partir de que los tiros bloqueados en un Y mayor tienen menor ángulo de tiro.

distancia_de_tiro - start_x = -0,84. Evidentemente, mientras aumenta la distancia desde el punto de partida al arco, la coordenada x disminuye, lo que explica la alta correlatividad.

distancia_de_tiro - blocked_x **=** -0,37. De la misma manera que el punto de partida, el punto de bloqueo también tendrá una coordenada menor en promedio cuando aumente la distancia.

— Se piden realizar gráficos scatterplot para ver correlación, histogramas, gráficos de barras o de torta, boxplots e incluso crear métricas como los ejemplos vistos en clase (y cualquier otro gráfico que se te ocurra!).

4) Pre-processing para modelado de Machine Learning.
Se debe preparar el dataset para realizar modelos de clasificación (predicción si el disparo fue gol o no). Preparar el dataset teniendo en cuenta los siguientes puntos:
— ¿Qué variables se van a incluir en el modelo? Eliminar las que no correspondan.

Consideramos incluir en el modelo todas las variables eliminando las columnas no relevantes, que son: ***id***, ***player***, ***match_id***, ***date***. Estas variables no aportan información que pueda aprovecharse para determinar la probabilidad de gol de un tiro.

— ¿Hay variables categóricas? ¿Cómo se deben preparar estas columnas?

Las variables categóricas incluidas en el dataset se convierten a un formato adecuado para el modelo, mediante one-hot encoding, donde no es necesario considerar la multicolinealidad, teniendo en cuenta que se trabaja un modelo de aprendizaje automático.
Como fue comentado anteriormente, el dataset utilizado tiene 14 columnas categóricas.

— Estandarizar variables numéricas y explicar por qué se realiza este proceso.

Este proceso debe realizarse para mantener un rango de valores consistente en las variables numéricas, lo que facilita el entrenamiento del modelo, teniendo valores de media 0 y desvío estándar 1.

— Realizar Train y Test split. Explicar por qué se realiza este proceso.

Se divide al dataset en dos: un 80% destinado al entrenamiento del modelo predictivo y un 20% a evaluar la performance del modelo.
Si el score del train es bueno y mayor que el score del test, podemos decir que el modelo está overfitteando.
Si el score del train es parecido al score del test, podemos decir que el modelo está generalizando bien.

Por último, si el score del train y del test es malo, podemos decir que hay underfitting.
Esto es importante en machine learning porque permite evaluar de manera objetiva y precisa cómo se desempeñará el modelo para datos nuevos. Esto promueve la construcción de modelos más efectivos y confiables que puedan generalizar bien a situaciones del mundo real más allá de los datos de entrenamiento.

5) Machine Learning.
Investigar el set de datos sobre todos los disparos y predecir si la jugada marcada terminó en gol o no. Se deben probar al menos 3 modelos de Machine Learning de clasificación.
— Para cada modelo realizar una matriz de confusión y analizar resultados (Evaluar las métricas de Accuracy, Precision, Recall).
— Seleccionar el modelo ganador y explicar por qué fue seleccionado.

Se han evaluado 3 modelos de machine learning para realizar la predicción de goles esperados:

- **K-Nearest Neighbors (KNN)**: asigna una etiqueta a un punto de datos nuevo en base a las etiquetas de sus K vecinos más cercanos en el espacio de características. Es un algoritmo de aprendizaje supervisado y no paramétrico, fue elegido por su simplicidad y efectividad en problemas de clasificación y regresión.
- **Random Forest**: es un algoritmo de Machine Learning que combina outputs de varios árboles de decisión para concluir en un único resultado. Se utiliza para mejorar la precisión y controlar el sobreajuste.
- **XGBoost**: “extreme gradient boosting" (refuerzo de gradientes extremo). Este método se basa en árboles de decisión y supone una mejora sobre el Random Forest. Funciona bien con datasets grandes y complejos al utilizar varios métodos de optimización.

Matriz de Confusión

KNN		Estimado	
		Negativo	Positivo
Real	Negativo	TN = 11336	FP = 148
	Positivo	FN = 280	TP = 886

Random Forest		Estimado	
		Negativo	Positivo
Real	Negativo	TN = 11397	FP = 87
	Positivo	FN = 261	TP = 905

XGBoost		Estimado	
		Negativo	Positivo
Real	Negativo	TN = 11412	FP = 72
	Positivo	FN = 261	TP = 905

Métricas utilizadas

MCC (Matthews Correlation Coefficient): Es una medida de la calidad general del modelo de clasificación que considera tanto los verdaderos positivos como los verdaderos negativos, teniendo en cuenta el desbalance de clases. Un valor más cercano a 1 indica un modelo perfecto, 0 indica predicciones aleatorias, y -1 indica un modelo completamente incorrecto.

Log Loss: También conocida como entropía cruzada logarítmica, mide la precisión de las predicciones de probabilidad del modelo. Valores más bajos indican una mejor precisión y confianza en las predicciones del modelo.

AUC-ROC (Area Under the Receiver Operating Characteristic curve): Es una medida de la capacidad del modelo para distinguir entre clases. Un valor de 1 indica un modelo perfecto, mientras que 0.5 indica predicciones aleatorias. Cuanto más alto sea el valor, mejor es la capacidad del modelo para discriminar entre las clases.

Specificity: También conocida como True Negative Rate, mide la proporción de negativos verdaderos que el modelo clasifica correctamente como negativos. Es útil en problemas donde identificar los negativos verdaderos es crítico.

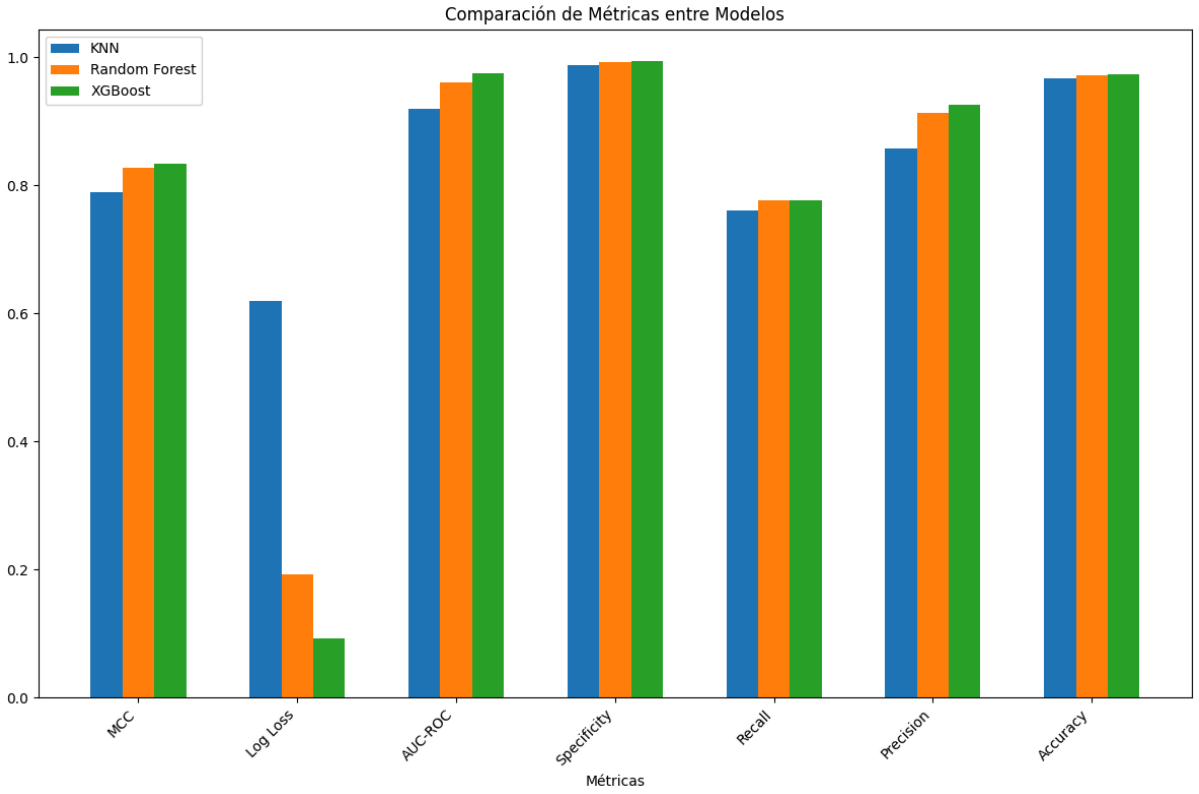
Precision: Es la proporción de instancias positivas predichas correctamente por el modelo entre todas las instancias positivas predichas. Indica cuán confiables son las predicciones positivas del modelo.

Recall: Es la proporción de instancias positivas que fueron correctamente predichas por el modelo sobre el total de instancias positivas reales. También se conoce como

sensibilidad o True Positive Rate. Es útil cuando el costo de los falsos negativos es alto.

Accuracy: Es la proporción de predicciones correctas (tanto positivas como negativas) realizadas por el modelo sobre el total. Es una métrica general de desempeño que indica qué tan bien el modelo clasifica todas las clases.

Se ha realizado una comparación de estas métricas con el fin de seleccionar un algoritmo para implementar en el modelo predictivo

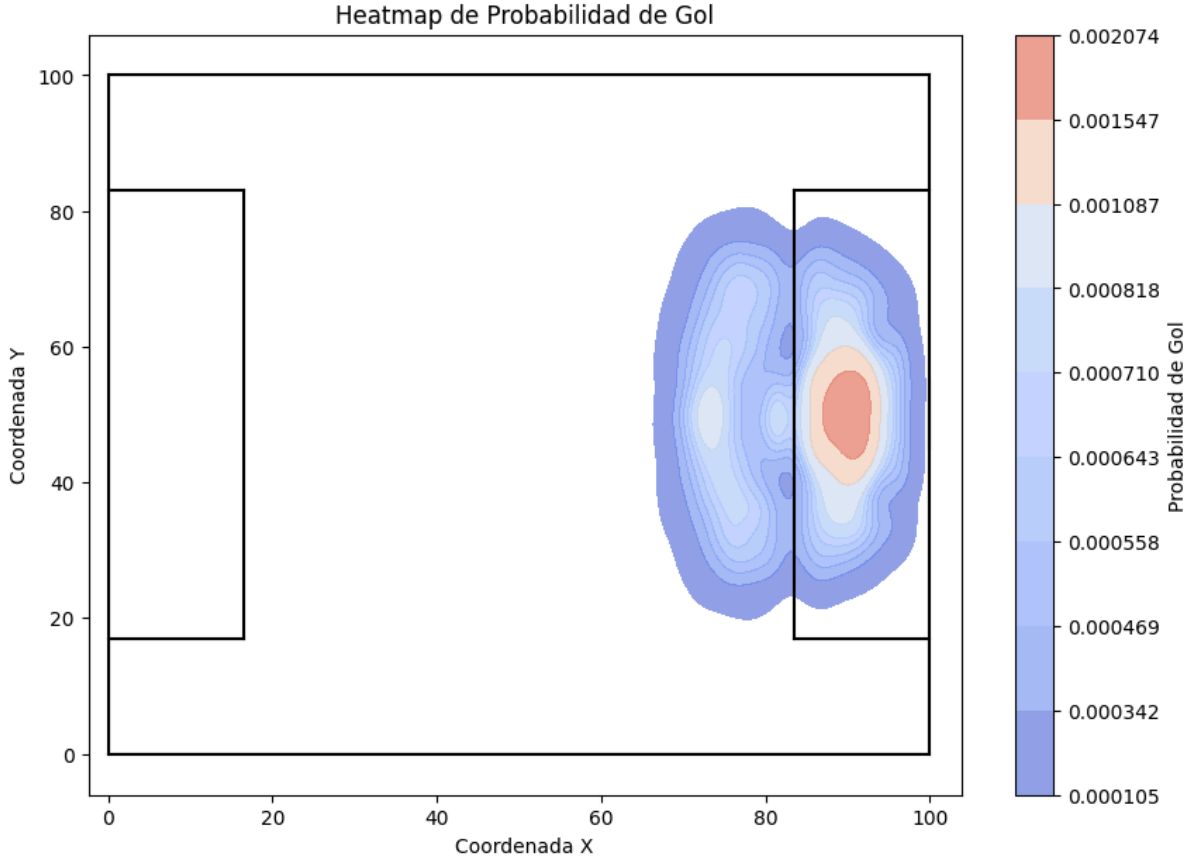


	KNN	Random Forest	XGBoost
MCC	0.789	0.827	0.834
Log Loss	0.620	0.193	0.093
AUC-ROC	0.920	0.961	0.975
Specificity	0.987	0.992	0.994
Recall	0.760	0.776	0.776
Precision	0.857	0.912	0.926
Accuracy	0.966	0.972	0.974

Selección del modelo

Se selecciona el algoritmo **XGBoost**, obtiene el mejor valor en cada una de las métricas evaluadas. Random Forest se le aproxima, sin embargo es levemente inferior. Esto tiene sentido, debido a que a grandes rasgos, XGBoost es una mejora sobre Random Forest. Si bien K-Nearest Neighbors posee buenas métricas, se encuentra un escalón por debajo de los otros métodos.

Como parte del análisis, a modo de demostración, se añadió la columna ‘xg’ al dataset con el algoritmo XGBoost, que representa la probabilidad de convertir un disparo en gol. Se confeccionó un mapa de calor que muestra la probabilidad de gol dadas las coordenadas iniciales del tiro. Este mapa de calor, basado en la columna ‘xg’, proporciona una representación visual para observar las áreas con mayor probabilidad de gol en los disparos al arco.



Como es de esperar, hay una concentración de alta probabilidad en la zona del punto de penal, desde ahí disminuye a medida que se aleja del arco. Se puede remarcar que hay dos zonas sobre el borde frontal del área donde también disminuye la probabilidad de gol, posiblemente debiéndose a la presencia de los defensores centrales del equipo que recibe el disparo.