**1. Desempeño actual del equipo y jugadores**

Este bloque se centra en medir el estado de forma **actual** de los equipos y jugadores, combinando estadísticas avanzadas, estilo de juego y contexto individual.

**Estadísticas generales del equipo**

* **Expected Goals (xG):**
  + xG a favor y en contra (home/away).
  + Diferencia de xG (xG a favor - xG en contra).
* **Goles y tiros:**
  + Promedio de goles anotados y recibidos (últimos 5-10 partidos).
  + Tiros totales y tiros a puerta (efectividad ofensiva).
* **Posesión del balón:**
  + Promedio de posesión.
  + Relación entre posesión y resultados (equipos que controlan vs. equipos que contraatacan).
* **Córners:**
  + Promedio de córners a favor y en contra (indicador de presión ofensiva).

**Rachas y tendencias**

* **Rachas de victorias/derrotas/empates:**
  + Últimos 5-10 partidos.
  + Rachas como local y visitante.
* **Rachas de jugadores:**
  + Máximos goleadores en forma (goles y asistencias recientes).
  + Jugadores con tendencia a marcar contra ciertos equipos.
* **Jugadores clave en riesgo:**
  + Riesgo de expulsión (jugadores con muchas amarillas).
  + Jugadores a una tarjeta de suspensión.

**Factores individuales y estilo de juego**

* **Ausencias importantes:**
  + Lesiones y sanciones de jugadores clave (goleadores, porteros, defensas).
  + Dudas en la alineación.
* **Estilo de juego:**
  + Equipos que juegan con alta posesión vs. equipos de contraataque.
  + Equipos que dependen del juego aéreo (córners y centros).
  + Equipos que sufren en defensa a balón parado.
* **Rumores y crisis:**
  + Problemas internos (conflictos con el entrenador, crisis institucional).
  + Rumores de salida de jugadores clave.

**2. Desempeño del árbitro**

Este bloque evalúa la influencia del árbitro en el partido, especialmente en lo relacionado con tarjetas, penales y faltas.

**Estadísticas del árbitro**

* **Promedio de tarjetas por partido:**
  + Tarjetas amarillas y rojas totales.
  + Tarjetas al equipo local vs. equipo visitante.
* **Promedio de faltas sancionadas:**
  + Total de faltas señaladas por partido.
  + Faltas al equipo local vs. visitante.
* **Penales pitados:**
  + Frecuencia con la que el árbitro pita penales.

**Tendencias del árbitro**

* **Influencia del estadio:**
  + ¿El árbitro tiende a favorecer al equipo local?
* **Historial con los equipos:**
  + Estadísticas del árbitro arbitrando partidos de los equipos involucrados.
  + Promedio de tarjetas o penales pitados en esos partidos.

**3. Desempeño histórico entre los dos equipos (Head-to-Head)**

Este bloque mide cómo se han comportado históricamente los equipos entre sí, con énfasis en los enfrentamientos recientes.

**Estadísticas generales**

* **Resultados recientes:**
  + Últimos 5-10 enfrentamientos (victorias, empates, derrotas).
  + Resultados en el estadio actual (local/visitante).
* **Diferencia de goles:**
  + Promedio de goles anotados y recibidos en los enfrentamientos.
  + ¿Suelen ser partidos ajustados o con goleadas?

**Tendencias específicas**

* **BTTS (Both Teams To Score):**
  + Frecuencia con la que ambos equipos marcan en los enfrentamientos.
* **Goles en momentos clave:**
  + Goles en los primeros o últimos minutos.
  + Goles en la primera o segunda mitad.
* **Jugadores destacados:**
  + Jugadores que suelen marcar o destacar en estos enfrentamientos.

**4. Variables contextuales**

Este bloque tiene en cuenta factores externos que pueden influir en el rendimiento de los equipos durante el partido.

**Calendario y fatiga**

* **Partidos recientes:**
  + Equipos que jugaron entre semana (Champions, Europa League, Copa del Rey).
  + Fatiga acumulada por viajes largos.
* **Rotaciones:**
  + Probabilidad de descanso de jugadores clave debido a la acumulación de partidos.

**Condiciones climáticas**

* **Clima durante el partido:**
  + Lluvia, viento o calor extremo que pueden afectar el estilo de juego.
  + Equipos técnicos vs. equipos físicos en condiciones adversas.

**Asistencia al estadio**

* **Público y ambiente:**
  + Asistencia estimada al estadio.
  + Estadios llenos que dan ventaja al equipo local.
* **Presión para el equipo local:**
  + Equipos que sufren en partidos clave con alta asistencia (descenso, derbis).

**Importancia del partido**

* **Clasificación actual:**
  + Lucha por el título, puestos europeos o descenso.
* **Derbis o rivalidades históricas:**
  + Mayor presión y motivación.
* **Partidos finales de temporada:**
  + Equipos que ya no se juegan nada vs. equipos con objetivos claros.

**5. Nuevas categorías sugeridas**

Además de tus grupos, podrías considerar agregar **dos nuevas categorías**:

**5.1. Desempeño en minutos clave**

* Goles anotados/concedidos en los **primeros 15 minutos**.
* Goles anotados/concedidos en los **últimos 15 minutos**.
* Número de goles en **tiempo de descuento**.

Esto ayuda a identificar equipos que "empiezan fuerte" o que "se caen al final".

**5.2. Factores psicológicos y tácticos**

* **Presión psicológica:**
  + ¿El equipo local necesita ganar sí o sí?
  + Equipos con "miedo a perder" en partidos importantes.
* **Cambios tácticos del entrenador:**
  + Tendencias de sustituciones ofensivas o defensivas en la segunda mitad.
  + Efectividad de los cambios realizados.
* **Rendimiento en partidos grandes:**
  + Desempeño contra equipos del "top 6" o rivales directos.

**Resumen final por bloques**

1. **Desempeño actual:**
   * Expected Goals, goles, tiros, posesión, córners, rachas, lesiones, sancionados, crisis.
   * Estilo de juego y desempeño individual de jugadores clave.
2. **Desempeño del árbitro:**
   * Promedio de tarjetas, faltas y penales.
   * Tendencia a favorecer al local y estadísticas con los equipos involucrados.
3. **Desempeño histórico entre equipos:**
   * Resultados recientes, BTTS, diferencia de goles y jugadores destacados en enfrentamientos.
4. **Variables contextuales:**
   * Fatiga por calendario, condiciones climáticas, asistencia al estadio y presión por la clasificación.

**Variables adicionales a tener en cuenta:**

Si deseas profundizar aún más:

* **Movimiento de las cuotas de apuestas:** Variación antes del partido (puede reflejar información clave).
* **Promedios de goles en segmentos del partido:** Goles entre el minuto 0-15, 15-30, etc.
* **Penales fallados y anotados:** Historial de efectividad en penales.

Perfecto, vamos a diseñar un **pipeline de preprocesamiento** para preparar estas variables y dejarlas listas para entrenar un modelo. El pipeline se encargará de:

1. **Cargar y limpiar los datos.**
2. **Generar nuevas variables (feature engineering).**
3. **Tratar los valores faltantes y outliers.**
4. **Codificar variables categóricas.**
5. **Escalar o normalizar las variables numéricas.**
6. **Unir todo en un dataset final listo para entrenar.**

**Pipeline de Preprocesamiento Paso a Paso**

**1. Cargar y limpiar los datos**

* Cargar los datasets (partidos, árbitros, equipos, jugadores, etc.).
* Eliminar columnas innecesarias o duplicadas.
* Convertir las fechas a formato datetime para calcular intervalos.
* Unificar nombres de equipos, árbitros, etc., si hay inconsistencias.

**Ejemplo de código en Python**

python

Copier le code

import pandas as pd

# Cargar datos

matches = pd.read\_csv('matches.csv')

referees = pd.read\_csv('referees.csv')

players = pd.read\_csv('players.csv')

# Limpiar datos

matches.drop\_duplicates(inplace=True)

matches['date'] = pd.to\_datetime(matches['date'])

matches = matches.drop(columns=['unnecessary\_column\_1', 'unnecessary\_column\_2'])

**2. Generar nuevas variables (feature engineering)**

**A. Desempeño actual**

* Calcular medias móviles de las últimas 5 partidas:
  + Goles a favor, goles en contra, puntos, tiros, posesión, etc.

python

Copier le code

matches['home\_team\_last\_5\_points'] = matches.groupby('home\_team\_name')['home\_team\_points'].rolling(5).mean().reset\_index(0, drop=True)

* Calcular xG diferencial (goles reales vs esperados).
* Crear una variable binaria para **jugadores clave lesionados** o sancionados.

**B. Desempeño del árbitro**

* Unir el dataset de árbitros con los partidos.
* Calcular medias: tarjetas amarillas/rojas, penales, sesgo local/visitante.

python

Copier le code

referee\_stats = referees.groupby('referee')['yellow\_cards', 'red\_cards'].mean()

matches = matches.merge(referee\_stats, on='referee', how='left')

**C. Desempeño histórico entre equipos**

* Calcular:
  + Puntos obtenidos en los últimos enfrentamientos (head\_to\_head\_last\_5\_matches).
  + Goles a favor y en contra en esos partidos.

**D. Variables contextuales**

* Calcular días desde el último partido (days\_since\_last\_match).
* Unir datos climáticos si están disponibles (weather\_conditions).
* Crear variable de **fatiga**:
  + Si un equipo jugó hace menos de 3 días, asignar "fatiga alta".

**3. Tratar valores faltantes y outliers**

* **Valores faltantes:**
  + Imputar con medias/medianas para variables numéricas.
  + Usar "Unknown" o "No data" para variables categóricas.

python

Copier le code

matches['weather\_conditions'].fillna('Unknown', inplace=True)

matches['home\_team\_xG'].fillna(matches['home\_team\_xG'].median(), inplace=True)

* **Outliers:**
  + Detectar usando métodos como **IQR** o **z-score** y tratarlos.

python

Copier le code

Q1 = matches['home\_team\_shots'].quantile(0.25)

Q3 = matches['home\_team\_shots'].quantile(0.75)

IQR = Q3 - Q1

matches = matches[(matches['home\_team\_shots'] >= Q1 - 1.5 \* IQR) & (matches['home\_team\_shots'] <= Q3 + 1.5 \* IQR)]

**4. Codificar variables categóricas**

* Variables como weather\_conditions, referee y home\_team\_name deben ser codificadas.
* Opciones:
  + **One-Hot Encoding** (para pocas categorías).
  + **Target Encoding** (para muchas categorías y si hay correlación con el target).

python

Copier le code

matches = pd.get\_dummies(matches, columns=['weather\_conditions'], drop\_first=True)

**5. Escalar o normalizar variables numéricas**

* Escalar las variables numéricas (como tiros, posesión, xG) para que tengan el mismo rango.
* Usar **StandardScaler** o **MinMaxScaler**.

python

Copier le code

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

numeric\_features = ['home\_team\_xG', 'away\_team\_xG', 'home\_team\_shots\_on\_target', 'away\_team\_shots\_on\_target']

matches[numeric\_features] = scaler.fit\_transform(matches[numeric\_features])

**6. Unir todo en un dataset final**

* Combinar todas las variables generadas y preprocesadas.
* El dataset final tendrá una estructura similar a:

| **match\_id** | **home\_team\_xG** | **away\_team\_xG** | **referee\_avg\_yellow\_cards** | **head\_to\_head\_home\_points** | **weather\_conditions\_Rain** | **days\_since\_last\_match** | **target (1=win, 0=lose/draw)** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1.5 | 0.8 | 4.2 | 9 | 1 | 3 | 1 |

**Pipeline Final Automatizado**

Una vez definidos todos los pasos, puedes automatizar el pipeline usando Pipeline de sklearn o Feature-engine.

python

Copier le code

from sklearn.pipeline import Pipeline

from sklearn.compose import ColumnTransformer

from sklearn.preprocessing import StandardScaler, OneHotEncoder

# Preprocesamiento numérico y categórico

numeric\_features = ['home\_team\_xG', 'away\_team\_xG', 'home\_team\_shots\_on\_target']

categorical\_features = ['weather\_conditions']

preprocessor = ColumnTransformer(

transformers=[

('num', StandardScaler(), numeric\_features),

('cat', OneHotEncoder(drop='first'), categorical\_features)

])

# Pipeline completo

pipeline = Pipeline(steps=[

('preprocessor', preprocessor),

])

# Aplicar pipeline

X = matches.drop(columns=['target'])

y = matches['target']

X\_preprocessed = pipeline.fit\_transform(X)

**Conclusión**

1. **Desglosa cada grupo** en variables específicas.
2. **Automatiza el pipeline** para limpiar, generar variables, escalar y codificar.
3. **Usa técnicas de selección de características** después del preprocesamiento para quedarte con las variables más importantes.

Si necesitas ayuda implementando alguna parte del pipeline o probando diferentes modelos, ¡dímelo! 🚀