PREDICCIÓN DE PROBABILIDAD DE GANAR UN PARTIDO DE FÚTBOL

PROYECTO FINAL

FUNDAMENTOS DE DATA SCIENCE

PATRICIA LEMA

OBJETIVO

- Realizar un análisis de los partidos de futbol entre dos países, Ecuador y cualquier otro país, para poder predecir quién podría ser el ganador en su próximo encuentro.

RESUMEN EJECUTIVO

- Se pretende predecir la probabilidad de que Ecuador pueda ganar en un partido de fútbol, basándose en una base que almacena todos los partidos de fútbol desde 1872 hasta 2025.
- Con esto definido se presenta la pregunta esencial: Cuál es la probabilidad que Ecuador gane en el próximo partido?

CONTEXTO Y ALCANCE

- Actualmente Ecuador alcanzó los 23 puntos y se mantiene en la segunda posición de la tabla de las Eliminatorias Sudamericanas hasta la próxima fecha. Debe derrotar a Brasil.

Liga: **Eliminatorias Copa del Mundo** ▾ Temporada: **2023–25** ▾

	Club	PJ	G	E	P	GF	GC	DG	Pts
1	Argentina	14	10	1	3	26	8	18	**31**
2	Ecuador	14	7	5	2	13	5	8	**23**
3	Uruguay	14	5	6	3	17	10	7	**21**
4	Brasil	14	6	3	5	20	16	4	**21**
5	Paraguay	14	5	6	3	11	9	2	**21**

ENTENDIMIENTO DE LOS DATOS

FUENTES DE DATOS

- La información proviene de la plataforma Kaggle www.kaggle.com
- Esta información se actualiza semanal o diariamente depende de los partidos llevados a cabo.
- La base utilizada se la bajó un día después del partido Ecuador vs Chile, el 26 de marzo de 2025, y ya tenía la información actualizada del 25 de marzo de 2025.

DESCRIPCIÓN Y CALIDAD DE LOS DATOS

- El archivo es .csv
- Los tipos de datos son:

```
dtype: object
date           datetime64[ns]
home_team              object
away_team              object
home_score              int64
away_score              int64
tournament             object
city                   object
country                object
neutral                  bool
```

- De los campos indicados home_team, away_team, home_score y away_score son claves para el análisis
- Los datos son confiables y sin inconsistencias, se validaron los datos de los partidos para verificar su veracidad. Tampoco tiene duplicidad

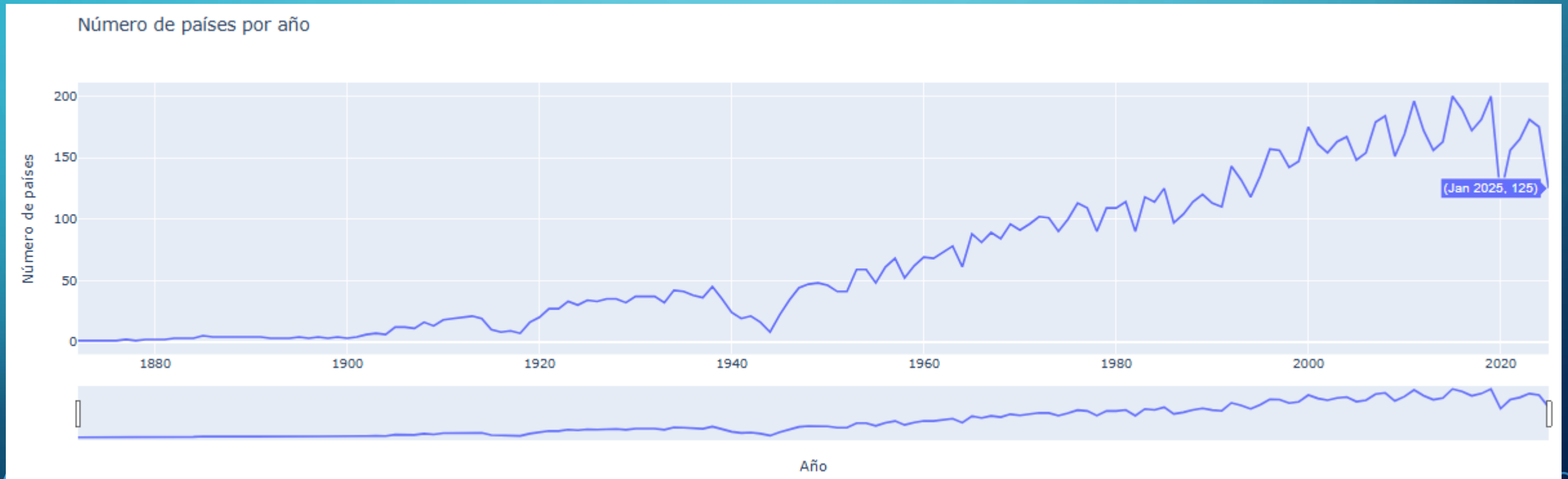
INFORME FINAL

CAMPOS NUEVOS

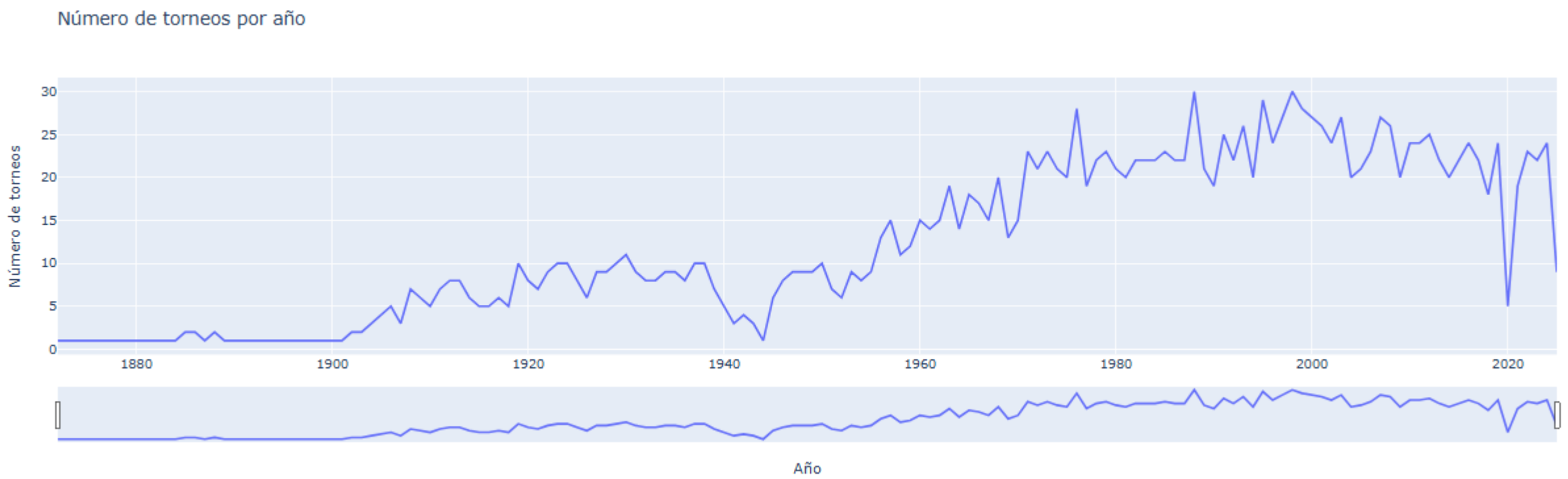
- porcentaje_ganado=0.5 lo ponemos por defecto
- porcentaje_ganado=partidos_ganados/partidos_anteriores

mes	año	resultado	oponente	partidos_previos	porcentaje_ganado
8	1938	empata	Bolivia	0	0.5
8	1938	gana	Colombia	0	0.5
8	1938	pierde	Peru	0	0.5
8	1938	gana	Venezuela	0	0.5
8	1938	pierde	Bolivia	1	0.0

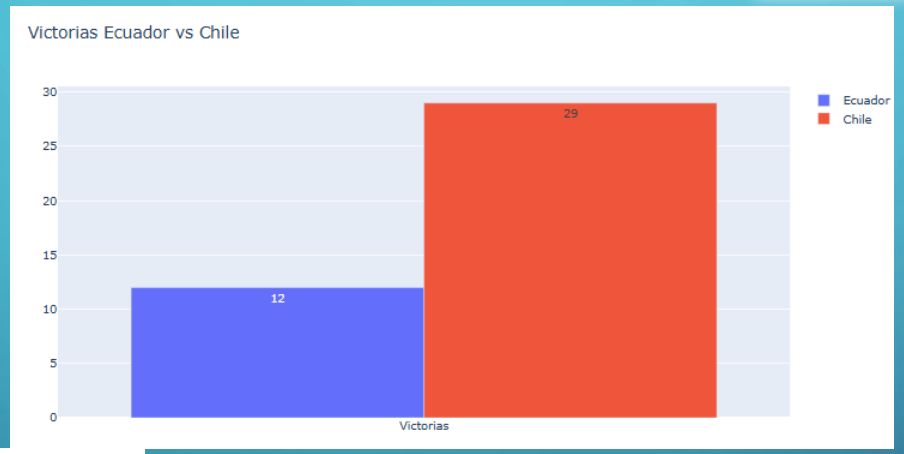
EVOLUCIÓN DE LOS PAÍSES PARTICIPANTES EN LOS PARTIDOS POR AÑO

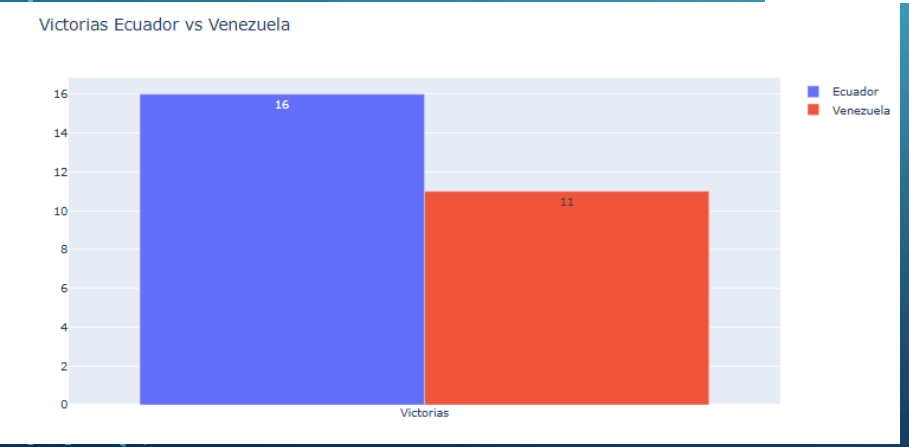


TORNEOS ÚNICOS JUGADOS POR AÑO



ULTIMOS OPONENTES





PREPARACIÓN DE DATOS

LIMPIEZA Y TRANSFORMACIÓN

- Se revisa tipo de dato de los campos
- Se verifica que no existan duplicados con la función duplicated
- Se cuenta las filas y columnas
- Se valida valores faltantes con la función isnull
- Se usa la función to_datetime para asegurarse que la fecha sea tipo date
- Se ordena por fecha con la función sort_values
- Se crear sumas de puntajes
- Se crea campos, año y mes

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48207 entries, 0 to 48206
Data columns (total 9 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   date        48207 non-null  object
 1   home_team   48207 non-null  object
 2   away_team   48207 non-null  object
 3   home_score  48207 non-null  int64 
 4   away_score  48207 non-null  int64 
 5   tournament  48207 non-null  object
 6   city        48207 non-null  object
 7   country     48207 non-null  object
 8   neutral     48207 non-null  bool  
dtypes: bool(1), int64(2), object(6)
memory usage: 3.0+ MB
None

Filas repetidas 0

Valores ausentes
date          0
home_team     0
away_team     0
home_score    0
away_score    0
tournament    0
city          0
country       0
neutral       0
dtype: int64
```

LIMPIEZA Y TRANSFORMACIÓN

- Se crea año y mes para las gráficas y agrupaciones
- Se crea resultado y oponente para los features
- Se crea partidos previos y porcentaje ganado como feature

mes	año	resultado	oponente	partidos_previos	porcentaje_ganado
8	1938	empata	Bolivia	0	0.5
8	1938	gana	Colombia	0	0.5
8	1938	pierde	Peru	0	0.5
8	1938	gana	Venezuela	0	0.5
8	1938	pierde	Bolivia	1	0.0

MODELADO

SELECCIÓN DE MODELOS

- Se usa clasificación y regresión
  Se usa get_dummies para crear las nuevas columnas en base a los oponentes con features = ['partidos_previos', 'porcentaje_ganado']
- Se usa: from sklearn.model_selection import train_test_Split para separar datos en entrenamiento y prueba y LabelEncoder para el resultado

- El 20% de los datos se asigna al conjunto de prueba
- El 80% restante se asigna al conjunto de entrenamiento
- Tenemos 463 de entrenamiento y 116 de test

	partidos_previos	porcentaje_ganado	oponente_Armenia	oponente_Australia	oponente_Belarus	oponente_Bolivia	oponente_Brazil	oponente_Bulgaria	oponente_Canada
20638	15	0.133333	False	False	False	True	False	False	False
31503	39	0.282051	False	False	False	False	False	False	False
18760	0	0.500000	False	False	True	False	False	False	False

```
Accuracy: 0.5172
Precision: 0.3338
Recall: 0.4415
F1-score: 0.3755
ROC AUC Score: 0.6136
```

- También se usa:
- from sklearn.metrics import **accuracy_score, precision_score, recall_score, f1_score, roc_auc_score**

- Análisis:
- Accuracy: El modelo tiene una exactitud del 51.72%, lo que significa que predice correctamente un poco más de la mitad de las veces. Esto puede indicar que el modelo no es muy preciso y hay margen de mejora. Precision: Un valor de 0.3338 sugiere que cuando el modelo predice gana, solo acierta alrededor del 33.38% de las veces. Esto indica una alta tasa de falsos positivos. (+FP)
- Recall: Con un valor de 0.4415, el modelo está capturando alrededor del 44.15% de los casos positivos reales. Esto implica que hay una cantidad significativa de falsos negativos. (PR)(+FN)
- F1-score: Un valor de 0.3755 indica un rendimiento moderado, pero con margen de mejora en la precisión y la recuperación.
- ROC AUC Score: Un valor de 0.6136 sugiere un rendimiento aceptable, mejor que una clasificación aleatoria (0.5), pero aún con espacio para mejorar. Un valor más cercano a 1 indicaría un mejor rendimiento.

EVALUACIÓN E INTERPRETACIÓN DE RESULTADOS

- A pesar de devolver valores no muy acertados se presentan los siguientes resultados
- Se necesitaría más features para obtener una mejor predicción pero la base es muy limitada

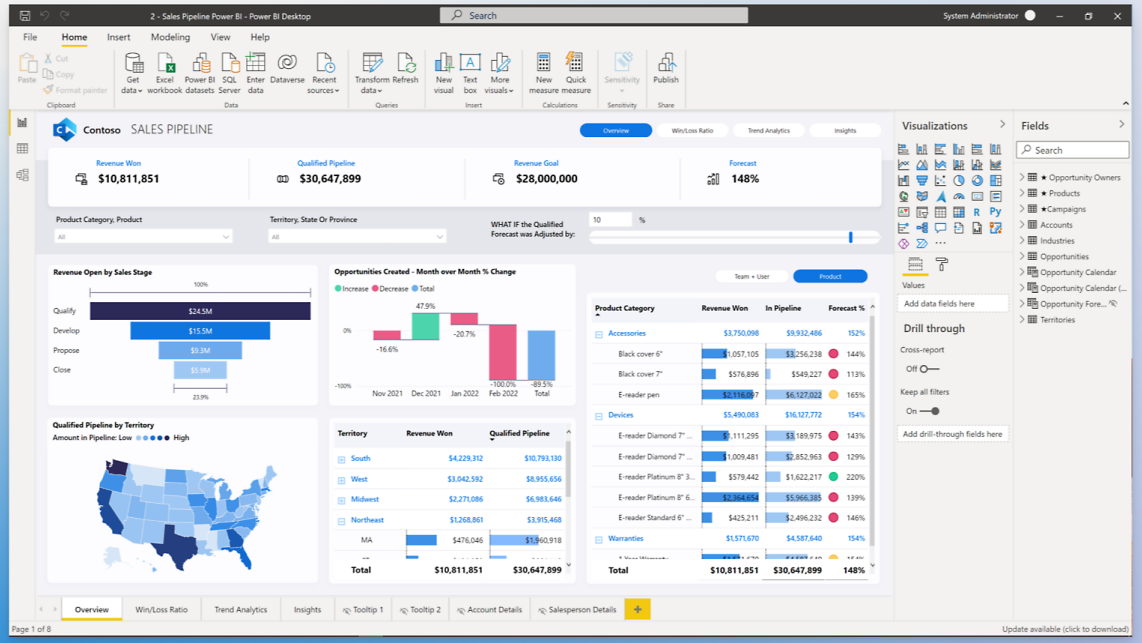
```
#Se puede poner cualquir país de la lista
prediccion_oponente = df_prediction[df_prediction['oponente_Venezuela'] == 1]
```

```
probabilidad_de_ganar_oponente = prediccion_oponente['probabilidad_de_ganar'].values[0]
print(f"Probabilidad de que Ecuador gane al oponente: {probabilidad_de_ganar_oponente:.4f}")
```

```
Probabilidad de que Ecuador gane al oponente: 0.4568
```

PLAN DE IMPLEMENTACIÓN

- Los resultados se podría presentar en un Power BI free



CONCLUSIONES, PRÓXIMOS PASOS Y RECOMENDACIONES

- Si se desea obtener mayor precisión se puede optar por conseguir más features como información del rendimiento de los jugadores, el clima, altura del territorio en el cuál se va a llevar a cabo el encuentro.