**UNIVERSIDAD PRIVADA ANTENOR ORREGO**

**FACULTAD DE INGENIERIA**

**PROGRAMA DE ESTUDIOS DE INGENIERIA DE SISTEMAS E INTELIGENCIA ARTIFICIAL**



**EXPOSICIÓN PRELIMINAR N° 01**

**ASIGNATURA:** APRENDIZAJE ESTADÍSTICO

**DOCENTE:** TEOBALDO SAGASTEGUI CHIGNE

**INTEGRANTES:**

* + - * MAXIMILIANO VILLANUEVA, JHON DENIS
      * SALIRROSAS VASQUEZ JHORDY
      * VASQUEZ CASTILLO JHERSON
      * VILLAJULCA QUISPE DIEGO
      * PAREDES CAJO JEFFRY
      * HARRISON CAMILO POMA

TRUJILLO, 25 DE JUNIO DE 2025

PERÚ

**I. Introducción**

**1.1 Titulo del Proyecto**

**PREDICCIÓN DE RESULTADOS EN LA LIGA PERUANA DE FÚTBOL MEDIANTE INTELIGENCIA ARTIFICIAL Y APRENDIZAJE ESTADÍSTICO PARA EL AÑO 2025.**

**1.2 Antecedentes**

En los últimos años, la aplicación de técnicas de inteligencia artificial (IA) y aprendizaje estadístico en el ámbito deportivo ha demostrado ser altamente eficaz para la predicción de resultados, análisis de rendimiento de jugadores y optimización de estrategias de juego.

En el contexto peruano, la Liga 1 (Primera División del fútbol peruano) se caracteriza por su competitividad y diversidad de estilos de juego. Sin embargo, la adopción de metodologías avanzadas para la predicción de resultados aún es incipiente en comparación con otras ligas internacionales.

La mayoría de los estudios previos sobre predicción de resultados deportivos se han centrado en ligas de alto perfil como la Premier League, La Liga Española o la Serie A Italiana. Sin embargo, existe un vacío importante en la aplicación de estas técnicas en ligas sudamericanas, especialmente en la Liga Peruana.

Mediante el uso de algoritmos de aprendizaje supervisado, como Regresión Logística, Máquinas de Soporte Vectorial (SVM), Redes Neuronales Artificiales (ANN) y Bosques Aleatorios, se busca desarrollar un modelo capaz de predecir con precisión los resultados de los partidos en la Liga Peruana

**1.3 Problema a resolver**

¿Cómo se puede diseñar un modelo de aprendizaje estadístico basado en inteligencia artificial que permita predecir con alta precisión los resultados de los partidos de la Liga Peruana de Fútbol, considerando variables históricas, estadísticas de jugadores, localía y otros factores contextuales?

**1.4 Objetivos**

**Objetivo General:**

Desarrollar un modelo de predicción de resultados de partidos en la Liga Peruana de Fútbol utilizando técnicas de aprendizaje estadístico e inteligencia artificial, capaz de analizar múltiples variables para generar pronósticos precisos.

* Recolectar y preprocesar un conjunto de datos históricos sobre la Liga Peruana de Fútbol, incluyendo resultados de partidos, estadísticas de equipos y jugadores, localía, condiciones climáticas, entre otros factores.
* Explorar y aplicar diferentes algoritmos de aprendizaje estadístico para la predicción de resultados.
* Comparar el desempeño de los distintos modelos mediante métricas de evaluación adecuadas (precisión, recall, F1-score, exactitud, etc.).
* Seleccionar el modelo más adecuado y optimizar sus parámetros para mejorar su rendimiento.
* Implementar un sistema de predicción que pueda ser actualizado con nuevos datos para mejorar su precisión continuamente.
* Proponer recomendaciones basadas en los resultados obtenidos y plantear posibles mejoras para futuros estudios.

**II. Requerimientos del sistema**

**2.1 Definición del Dominio**

El dominio de este proyecto es el análisis predictivo en el contexto del fútbol profesional peruano, específicamente la Liga 1 (temporada 2025). Se enfoca en la relación entre variables numéricas relacionadas con los goles y el resultado final de los partidos, utilizando técnicas de aprendizaje automático para la clasificación.

**2.2 Determinación de Requisitos (Requerimientos)**

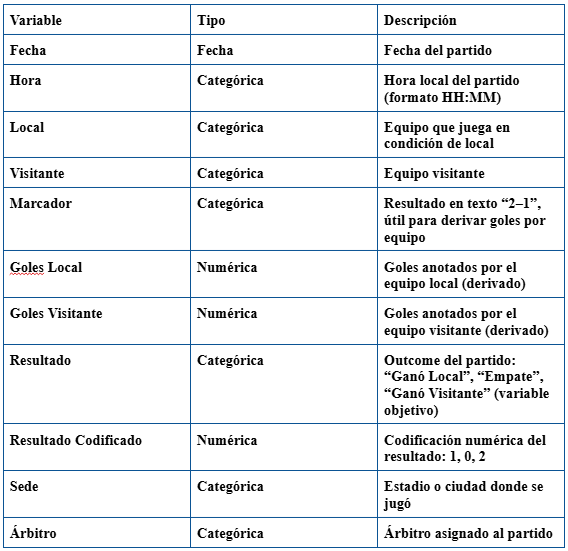
**III. Planteamiento del Data-Set (por Aprendizaje Supervisado o Aprendizaje no-Supervisado), PRE-PROCESAMIENTO y Normalización.**

**3.1 Medidas, Datos, Bases de Datos y Elaboración del Data-Set.**

El dataset contiene partidos oficiales de la Liga 1 Peruana (2025), con cada fila representando un encuentro. En total, incluye 137 registros (uno por partido observado).

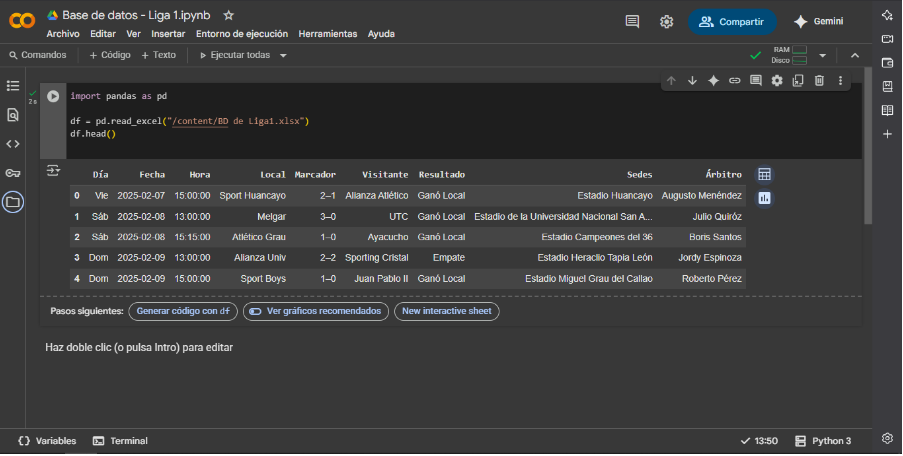
Las variables recopiladas abarcan aspectos de localía, goles, resultado y contexto del partido.

* Los datos fueron extraídos manualmente y parcialmente con herramientas de scraping desde sitios como *SofaScore* y *Transfermarkt*.
* El marcador, árbitro y sede fueron completados manualmente desde páginas oficiales.
* El dataset original está en formato Excel, y se transformó a DataFrame en Python para aplicar el preprocesamiento correspondiente.

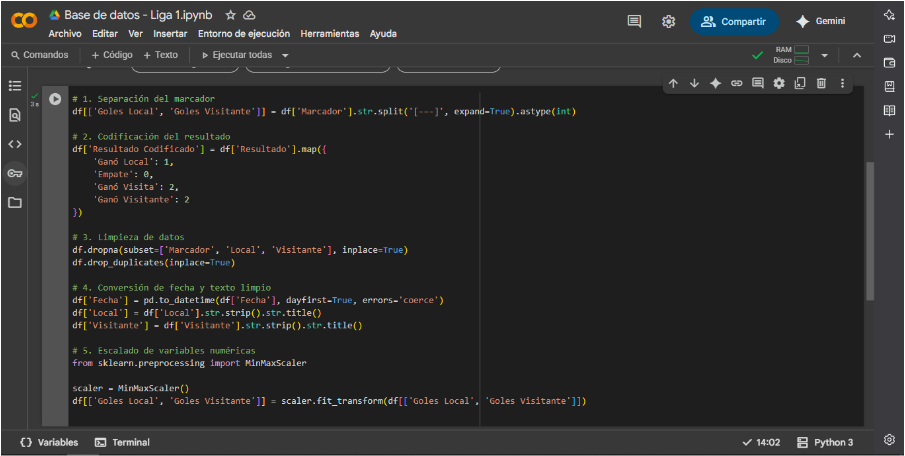


**3.2 Normalización y/o Filtrado de Datos**

* Carga del DataFrame:

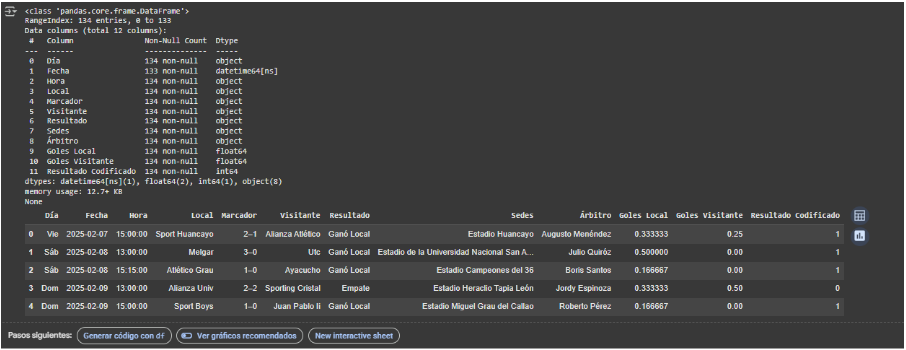


* Preparación del dataset:
* Se realizaron transformaciones necesarias para adaptar el dataset al modelo de aprendizaje:
* Separación del marcador en variables numéricas
* Codificación de resultados
* Limpieza de datos faltantes y duplicados
* Conversión de fechas y normalización de texto
* Escalado de variables numéricas



Luego de aplicar los pasos de preprocesamiento (separación de marcador, codificación de resultado, limpieza, conversión de formatos y escalado), el DataFrame final contiene columnas transformadas listas para su uso en el modelo.

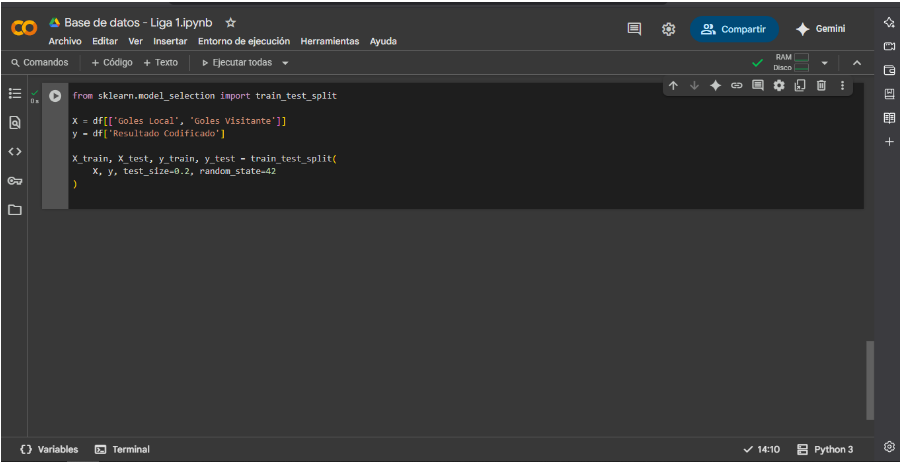
A continuación se muestra el resultado con df.info() y df.head():



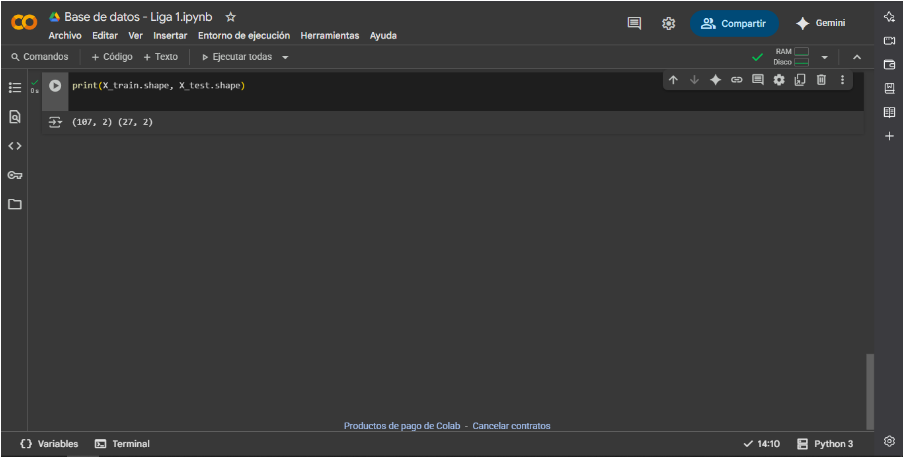
**3.3 Planteamiento de:**

**- Data-Set de entrenamiento (training)**

Se separó el conjunto de entrenamiento (80%) y de prueba (20%) usando train\_test\_split, con las variables numéricas como entrada (X) y la codificación del resultado como salida (y).

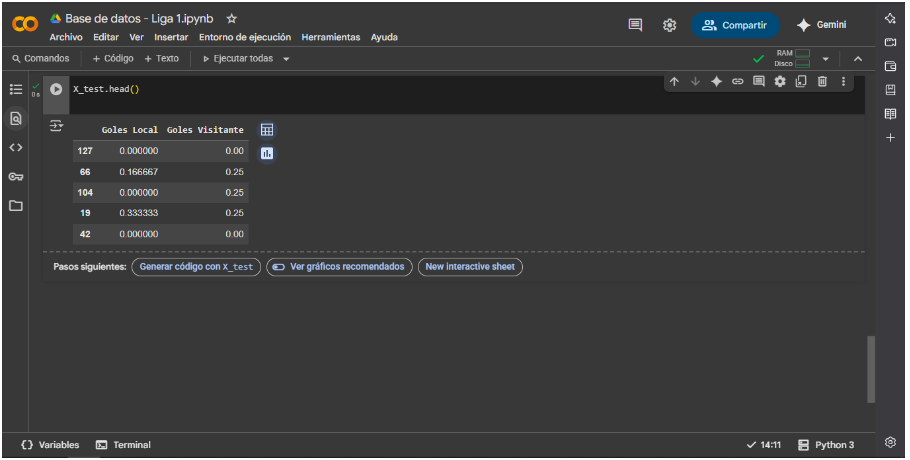


Tamaño de los conjuntos de entrenamiento y prueba:



**- Data-Set de Pruebas (test)**

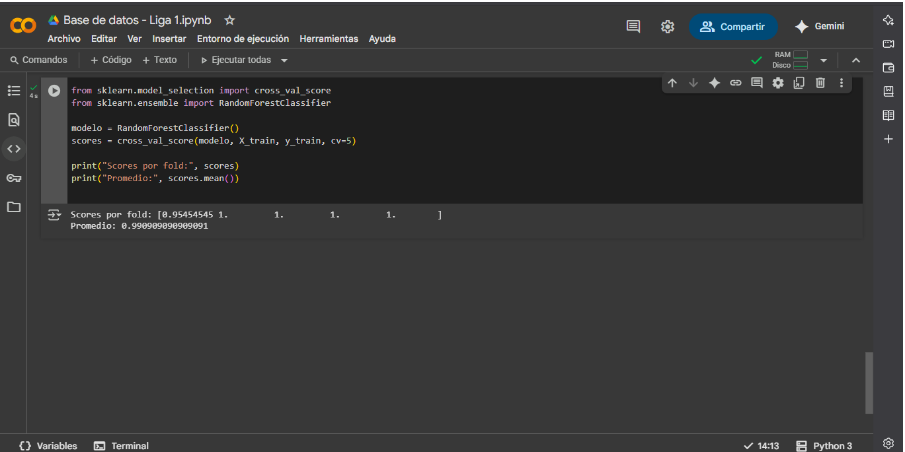
El conjunto de prueba (20% de los datos) se reserva para la evaluación del rendimiento final del modelo con datos no vistos.



Vista de los primeros registros del conjunto de prueba.

**- Set de Validación por "Cross-Validation"**

Se utilizó validación cruzada K-Fold con k=5 para mejorar la confiabilidad del entrenamiento, reduciendo la varianza causada por una única partición.



**IV. Aprendizaje del modelo del sistema**

**4.1 Planteamiento del Modelo de Aprendizaje.**

En este proyecto se plantea el uso de modelos de aprendizaje supervisado, específicamente modelos de clasificación, ya que el objetivo es predecir el resultado del partido en forma codificada (Resultado Codificado) en función de variables numéricas como los goles anotados por el equipo local y visitante.

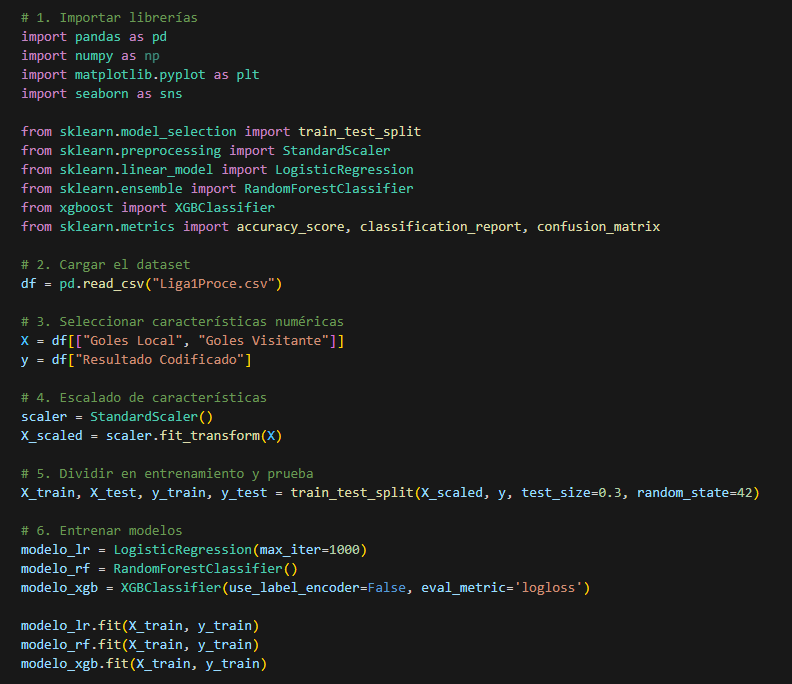
Los modelos seleccionados para el aprendizaje son:

* **Regresión Logística:** Modelo lineal para clasificación binaria. Útil como línea base por su simplicidad e interpretabilidad.
* **Random Forest:** Ensamble de árboles de decisión que mejora la precisión y reduce el sobreajuste.
* **XGBoost (Extreme** **Gradient Boosting):** Modelo de boosting altamente eficiente y preciso que suele superar a otros modelos en competiciones de ciencia de datos.

**4.2 Desarrollo e Implementación del Modelo**

Los modelos se desarrollaron e implementaron utilizando Python y las librerías scikit-learn y xgboost. El flujo seguido fue:

* Selección de variables predictoras (Goles Local, Goles Visitante) y variable objetivo (Resultado Codificado).
* Estandarización de los datos utilizando StandardScaler para evitar sesgos por magnitudes distintas.
* División del dataset en entrenamiento (70%) y prueba (30%) mediante train\_test\_split.
* Entrenamiento de tres modelos distintos: LogisticRegression, RandomForestClassifier y XGBClassifier.
* Cada modelo fue ajustado (fit) con los datos de entrenamiento para aprender la relación entre goles y resultado.



**V. Comprobación y Despliegue (Deploy) del Sistema**

**5.1 Entrenamiento del Modelo: Uso del Data-Set de entrenamiento**

El conjunto de entrenamiento fue utilizado para ajustar los parámetros internos de cada modelo. Durante este proceso, cada modelo ajustó sus pesos (en el caso de regresión logística), árboles de decisión (en Random Forest), o secuencia de árboles optimizados (en XGBoost) en función del patrón encontrado en los goles anotados por local y visitante.

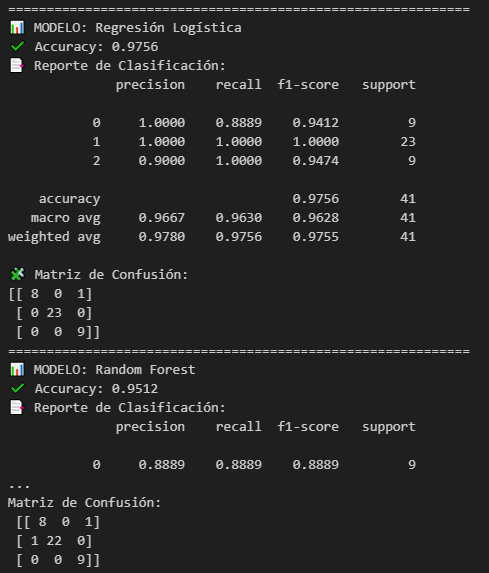
Se usó el siguiente particionamiento:

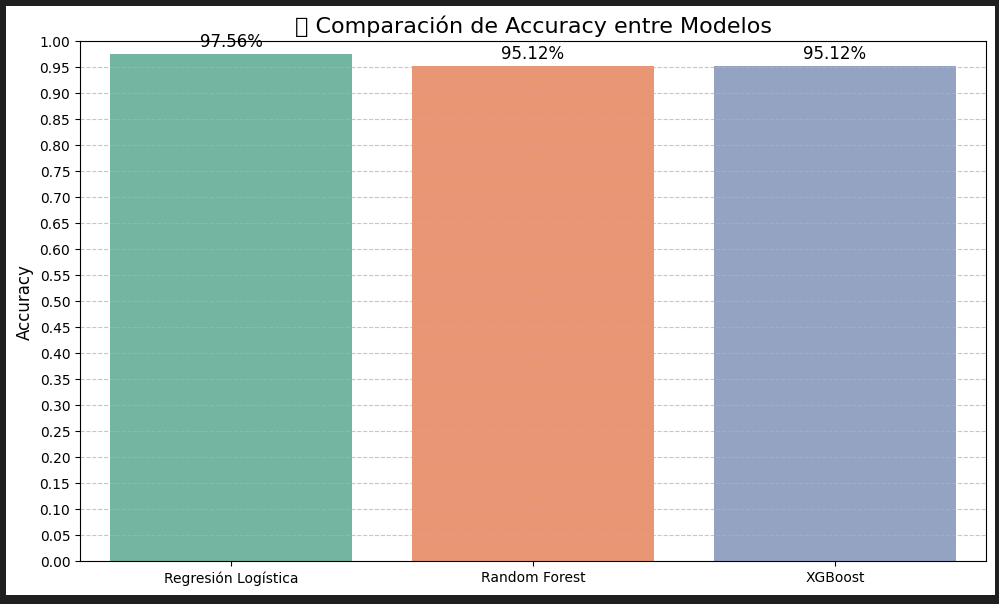
* **70%** del dataset para entrenamiento.
* **30%** para prueba.

**5.2 Ejecución y Pruebas del Modelo**

Luego del entrenamiento, los modelos fueron evaluados usando el conjunto de prueba. Se realizaron predicciones (predict) sobre los datos no vistos y se evaluaron los resultados usando métricas estándar de clasificación como:

* **Accuracy**: Proporción de aciertos totales.
* **Reporte de clasificación**: incluye precisión, recall y F1-score para cada clase.
* **Matriz de confusión**: visualización de verdaderos positivos, falsos negativos, etc.





**5.3 Ejecución de la validación del Modelo**

Con el objetivo de mejorar la confiabilidad del entrenamiento y evaluar la capacidad de generalización de los modelos, se aplicó una técnica de validación cruzada (K-Fold) con k=5.

Este enfoque consiste en dividir el conjunto de datos en 5 subconjuntos o "folds". En cada iteración, uno de estos subconjuntos se utiliza como conjunto de validación, y los otros cuatro como conjunto de entrenamiento. Este proceso se repite cinco veces, asegurando que cada subconjunto actúe como validación exactamente una vez. Al final, se calcula el promedio de las métricas de evaluación para obtener una estimación más robusta del desempeño del modelo.

