INFORME EJECUTIVO DEL PROYECTO FINAL DE DEEP LEARNING 2025

Predicción de Resultados Deportivos mediante Modelos Secuenciales

José Alfredo Martínez Valdés
Facultad de Ingeniería – Ingeniería de Sistemas
Universidad de Antioquia
Fundamentos de Deep Learning
Medellín, Colombia
jose.martinez7@udea.edu.co

***Abstract*—Este informe ejecutivo presenta la solución desarrollada para el proyecto final del curso Deep Learning 2025. El trabajo consiste en la construcción, experimentación y comparación de modelos secuenciales (LSTM y Transformer) para predecir resultados deportivos a partir de series temporales generadas desde un conjunto de datos históricos. Se describen el contexto de aplicación, el objetivo de *machine learning*, la estructura completa de los notebooks entregados, la arquitectura del sistema, los procesos de preprocesado, las iteraciones realizadas y los resultados obtenidos. Finalmente, se exponen consideraciones de reproducibilidad, limitaciones del estudio y líneas de trabajo futuro.**

I. Introducción

El análisis predictivo aplicado al deporte es un campo relevante dentro del aprendizaje automático, debido a la abundancia de datos históricos y la complejidad inherente del rendimiento competitivo. Este proyecto desarrolla un *pipeline* completo para la predicción del resultado del próximo partido desde la perspectiva del equipo local, utilizando modelos secuenciales capaces de capturar patrones temporales.

Los componentes principales de la solución son:

- Exploración y limpieza del conjunto de datos.
- Construcción de secuencias temporales.
- Entrenamiento de modelos LSTM y Transformer [1], [2].
- Evaluación comparativa mediante métricas de clasificación.
- Documentación reproducible mediante notebooks ejecutables en Google Colab.

El informe responde de manera explícita a los requisitos de la asignatura tanto para la Entrega 1 como para la Entrega 2: documentación ejecutiva, notebooks numerados, modelos entrenados, datos documentados y enlace al video explicativo del proyecto.

II. Contexto de Aplicación y Objetivo de Machine Learning

A. Contexto

El proyecto aborda la predicción de resultados de fútbol profesional. Aunque los resultados deportivos están sujetos a un alto grado de variabilidad, la evolución temporal de un equipo contiene patrones relevantes, incluyendo rachas de victorias/derrotas y desempeños recientes.

B. Objetivo

El objetivo se formula como un problema de clasificación multiclase:

> *Predecir el resultado del próximo partido del equipo local (victoria, empate o derrota) utilizando una ventana fija de partidos previos y características derivadas.*

Desde la óptica práctica, se busca construir un modelo que supere un baseline simple y sirva como punto de partida para futuros análisis predictivos más avanzados.

III. Datos y Disponibilidad

A. Tipo y tamaño de los datos

El conjunto de datos está compuesto por registros tabulares de partidos históricos:

- 7 000–8 000 partidos luego del filtrado.
- Variables: fecha, equipos implicados, goles anotados, marcador, condición local y diferencias de goles.
- Variable objetivo: `win`, `draw` o `loss`.

B. Distribución de clases

La proporción observada es:

- Victoria: 43%
- Empate: 27%
- Derrota: 30%

Esto implica un desbalance moderado que justifica el uso de métricas macro-promediadas.

C. Disponibilidad y preparación

Los datos están incluidos en el repositorio en formato procesado. Los notebooks:

- **01_exploracion_datos.ipynb**: descripción, consistencia y exploración.
- **02_preprocesamiento.ipynb**: limpieza, ordenamiento temporal, generación de etiquetas y construcción de secuencias.

IV. Estructura de Notebooks Entregados

Los notebooks están organizados numéricamente según lo requerido:

- **01**: Exploración de datos.
- **02**: Preprocesamiento.
- **03**: Baseline no secuencial.
- **04**: Modelo LSTM.
- **05**: Modelo Transformer.
- **06**: Iteraciones experimentales.

Todos los notebooks están verificados para ejecución en Google Colab.

V. Descripción de la Solución

A. Preprocesado

Incluye:

- Ordenamiento cronológico.
- Construcción de etiqueta objetivo.
- Normalización y creación de características derivadas.
- Generación de secuencias de longitud fija.
- División temporal en entrenamiento, validación y prueba.

B. Modelo Baseline

Incluye un clasificador simple basado en características agregadas. Su propósito es establecer una línea base mínima.

C. Modelo LSTM

Modelo recurrente con:

- Una capa LSTM de 64 unidades.
- Capa densa intermedia.
- Activación `softmax`.
- Regularización por *dropout*.

D. Modelo Transformer

Basado en mecanismos de atención [2]:

- Codificación posicional.
- Bloques encoder con *multi-head attention*.
- Proyección final multiclase.

VI. Iteraciones Realizadas

Cada iteración está documentada en los notebooks:

- Baseline inicial.
- LSTM simple.
- LSTM regularizado.
- Primer Transformer.
- Ajustes de dimensiones, *heads* y funciones de activación.
- Optimización final y selección del mejor modelo.

VII. Métricas de Desempeño

Se utilizaron métricas técnicas y de negocio:

- Accuracy.
- Precisión macro.
- F1 macro.
- Matrices de confusión.

Modelo	Accuracy	Precisión macro	F1 macro
Baseline	0.45	0.41	0.39
LSTM	0.57	0.55	0.53
Transformer	0.61	0.59	0.58

TABLE I
Comparación de desempeño entre modelos.

VIII. Resultados

Los resultados confirman que el Transformer supera al LSTM, y ambos superan al baseline.

IX. Limitaciones del Estudio

- El conjunto de datos no incluye características contextuales (localía formal, ranking FIFA, efectos de torneo, alineaciones).
- Solo se utilizaron secuencias basadas en diferencia de goles.
- No se optimizaron hiperparámetros mediante búsqueda exhaustiva.
- El dominio es altamente ruidoso por naturaleza.

X. Trabajo Futuro

- Incorporación de variables adicionales: ranking, superficie, clima, torneos.
- Modelos híbridos CNN–LSTM o Transformer–MLP.
- *Transfer learning* con modelos preentrenados para series temporales.
- Optimización avanzada de hiperparámetros.

XI. Reproducibilidad y Entorno Experimental

- Todos los notebooks se ejecutan en Google Colab sin dependencias locales.
- El archivo `requirements.txt` describe las librerías principales.
- Los modelos entrenados están almacenados en `models/`.
- Semillas fijadas para controlar la aleatoriedad.

XII. Conclusiones

Se demostró que:

- Los modelos secuenciales capturan patrones temporales relevantes.
- El Transformer obtiene el mejor desempeño general.
- El pipeline desarrollado es modular, reproducible y extensible.

Enlace al Video del Proyecto

https://youtu.be/3QCbO28zlaQ

References

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. DOI: 10.1162/neco.1997.9.8.1735.

[2] A. Vaswani et al., "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.