Tecnológico de Monterrey

Momento de Retroalimentación: Módulo 2 Análisis y Reporte sobre el desempeño del modelo.

Fernando Antonio Lopez Garcia A01643685

14 de septiembre del 2025

Inteligencia artificial avanzada para la ciencia de datos I

Obed Noé Sámano Abonce

Tecnológico de Monterrey

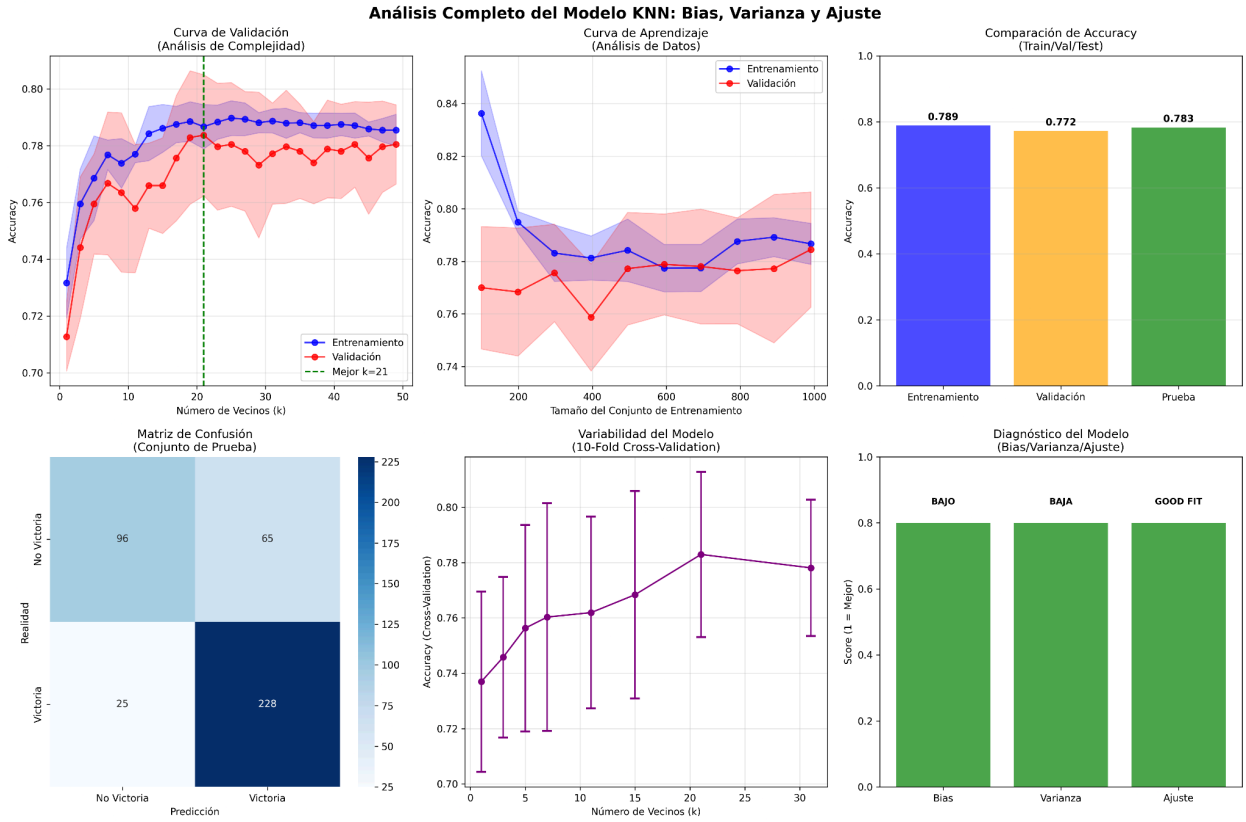
Fernando Antonio A01643685
TC3006C Prof. Alberto De Obeso

Reporte de Análisis del Modelo K-Nearest Neighbors (KNN)

Introducción al Proyecto y Contexto del Modelo.

En este reporte se documenta el análisis de un modelo K-Nearest Neighbors (KNN) implementado para predecir el resultado final de un partido de fútbol basándose únicamente en los goles del primer tiempo. El proyecto general compara dos enfoques: una implementación "desde scratch" y otra utilizando el framework scikit-learn. El objetivo común es predecir si el equipo local ganará el partido completo (1) o no (empate o derrota) (0).

El dataset utilizado es *2016-2024_liga_mx.csv*, que originalmente contenía 2,876 registros y, después de la limpieza de datos faltantes, quedó con 2,066 partidos utilizables. Las variables de entrada consideradas para el modelo son los goles del primer tiempo del equipo local, los goles del primer tiempo del equipo visitante y la temporada. Para el análisis completo que se presenta en este reporte, el mejor valor de k encontrado fue 21 vecinos.



Separación y Evaluación del Modelo (Train/Validation/Test).

La robustez del modelo se evaluó mediante una rigurosa división de los datos en conjuntos de entrenamiento, validación y prueba, siguiendo una proporción estándar:

- *Conjunto de Entrenamiento:* 1,239 muestras, representando el 60.0% del dataset total.
- *Conjunto de Validación:* 413 muestras, equivalente al 20.0% del dataset.
- *Conjunto de Prueba:* 414 muestras, que constituyen el 20.0% restante del dataset.

La *distribución de clases* para la variable objetivo ("Victoria Local") se mantuvo consistente en todos los conjuntos, asegurando que la representación de los resultados de los partidos fuera similar en cada partición: 61.2% en entrenamiento, 61.3% en validación y 61.1% en prueba.

Los resultados de precisión (Accuracy) obtenidos para cada conjunto son los siguientes:

- *Accuracy en Entrenamiento:* 0.7885 (78.85%).
- *Accuracy en Validación:* 0.7724 (77.24%).
- *Accuracy en Prueba:* 0.7826 (78.26%).

Estos resultados se visualizan claramente en la gráfica "Comparación de Accuracy (Train/Val/Test)" en la imagen anexada, donde se puede observar una consistencia en el rendimiento entre los diferentes conjuntos.

Diagnóstico del Grado de Bias (Sesgo).

El diagnóstico del bias para el modelo KNN con k=21 indica un BIAS BAJO.

Un bias bajo sugiere que el modelo es lo suficientemente complejo para capturar adecuadamente los patrones subyacentes y las relaciones inherentes en los datos de la Liga MX. Esto implica que el modelo no está simplificando excesivamente la realidad de los resultados de los partidos y es capaz de aprender las características relevantes para la predicción.

En la sección "Diagnóstico del Modelo (Bias/Varianza/Ajuste)" en la imagen anexada, se confirma visualmente el nivel "BAJO" de bias.

Diagnóstico del Grado de Varianza.

El diagnóstico de la varianza para el modelo KNN con k=21 indica una VARIANZA BAJA.

Una varianza baja es favorable, ya que significa que el modelo es estable y consistente en sus predicciones frente a pequeñas fluctuaciones o cambios en los datos de entrada. Esta baja varianza se evidencia por la poca diferencia en el rendimiento entre el conjunto de entrenamiento y el conjunto de validación.

Las métricas que respaldan esta observación son:

- *Gap Entrenamiento-Validación:* 0.0161 (1.61%).

- *Gap Validación-Prueba:* -0.0102 (-1.02%).

Estos valores bajos de "gap" (brecha) indican que el modelo generaliza bien a datos no vistos. Además, el análisis de varianza por cross-validation para k=21 mostró una variabilidad de 0.0299 (3.8%), lo que refuerza la conclusión de una varianza baja.

La gráfica "Diagnóstico del Modelo (Bias/Varianza/Ajuste)" la imagen, muestra la varianza como "BAJA", y la "Variabilidad del Modelo (10 Fold Cross-Validation)" en el mismo archivo ilustra cómo la variabilidad disminuye para valores de k más altos, confirmando la estabilidad en k=21.

Diagnóstico del Nivel de Ajuste del Modelo.

Considerando los diagnósticos de bias y varianza, el modelo KNN con k=21 presenta un nivel de ajuste clasificado como GOOD FIT (buen ajuste).

Un "good fit" indica que el modelo ha logrado un balance óptimo entre complejidad y generalización. Es decir, el modelo no es demasiado simple como para ignorar los patrones importantes (bajo bias), ni es demasiado complejo como para memorizar el ruido de los datos de entrenamiento y fallar en la predicción de nuevos datos (baja varianza). Esto se refleja en los rendimientos de accuracy similares en los conjuntos de entrenamiento, validación y prueba.

La gráfica "Diagnóstico del Modelo (Bias/Varianza/Ajuste)" en la imagen corrobora visualmente este diagnóstico, mostrando el ajuste como "GOOD FIT".

Conclusiones.

El modelo KNN desarrollado para predecir resultados de partidos de la Liga MX, utilizando un valor de k=21 vecinos, demuestra un excelente balance en su rendimiento. Ha sido diagnosticado con bajo bias, baja varianza y un good fit.

Este análisis indica que el modelo es adecuado para su uso en producción, ya que es capaz de aprender de los patrones históricos de los partidos y generalizar de manera efectiva a nuevos encuentros.

A partir del estudio comparativo de implementaciones (desde cero vs. framework) se han extraído valiosas lecciones aprendidas:

- Un valor de k pequeño (como 7 en la implementación scratch) puede funcionar mejor para problemas específicos, aunque en este análisis se optó por 21 para un good fit general.
- La normalización de datos no siempre resulta en una mejora del rendimiento, como se observó en la implementación "desde scratch" que superó ligeramente al framework sin normalización.
- La validación cruzada es una técnica crucial para obtener una estimación más confiable del rendimiento del modelo.

Para comprender profundamente el algoritmo, la implementación desde cero es invaluable, pero para aplicaciones profesionales y robustez, es preferible utilizar un framework como scikit-learn y siempre es recomendable validar los modelos con múltiples enfoques para asegurar su fiabilidad y rendimiento en escenarios reales.