

# Laboratorio 1: Preparación de Datos, KNN y Bias-Variance

Inteligencia Artificial - CC3085

José Antonio Mérida Castejón

1 de febrero de 2026

## Task 1

*Responda las siguientes preguntas justificando su respuesta basándose en el material visto en clase. Se espera que demuestren análisis del caso y no solamente un “copy-paste“.*

### El Problema de la Convexidad (Regresión Logística)

*Durante la clase, se habló que no podemos usar el Error Cuadrático Medio (MSE) para la Regresión Logística. Con esto en mente considere el escenario donde usted asume el rol de Lead AI Engineer y un Junior le presenta un modelo de clasificación binaria que usa una función sigmoide, pero insiste en entrenarlo usando la fórmula MSE porque “es lo que usó en regresión lineal y funcionaba bien“.*

*Explíquele técnicamente (y gráficamente si es necesario) por qué su modelo probablemente se quedará atascado y no encontrará la solución óptima. Mencione el concepto de “mínimos locales“ vs “mínimo global“.*

Al entrenar modelos de machine learning, las funciones de pérdida que no tienen *solución de forma cerrada* se optimizan por medio de *descenso de gradiente*. Recordemos que la condición para tener una *solución de forma cerrada* implica poder aislar nuestros pesos ( $w$ ) utilizando un número finito de operaciones algebraicas. Es decir, podemos simplemente “despejar” para cada  $w$  y encontrar los valores de cada peso. En este caso, podemos intentar encontrar los *puntos críticos* de la función pérdida planteada por medio de la *gradiente*:

$$\sum(y_i - \sigma(w^T x_i))\sigma'(w^T x_i)x_i = 0$$

Observamos que los pesos  $w$  están atrapados dentro de la función sigmoide:

$$\sigma(w^T x_i) = \frac{1}{1 + e^{-w^T x_i}}$$

Esta es una ecuación trascendental. Debido a que  $w$  aparece en el exponente dentro de una sumatoria, no existe una operación algebraica finita que permita “despejar” o aislar los pesos. Por lo tanto, debemos utilizar un método numérico iterativo como el *descenso de gradiente*. Adicionalmente, aunque existiera una solución de forma cerrada tendríamos problemas al no tener una función

*convexa.* Recordemos que una función convexa es aquella donde para cualesquiera dos puntos  $x_1$  y  $x_2$  en su dominio, al trazar una línea recta entre estos puntos la función queda por debajo.

pgfplots

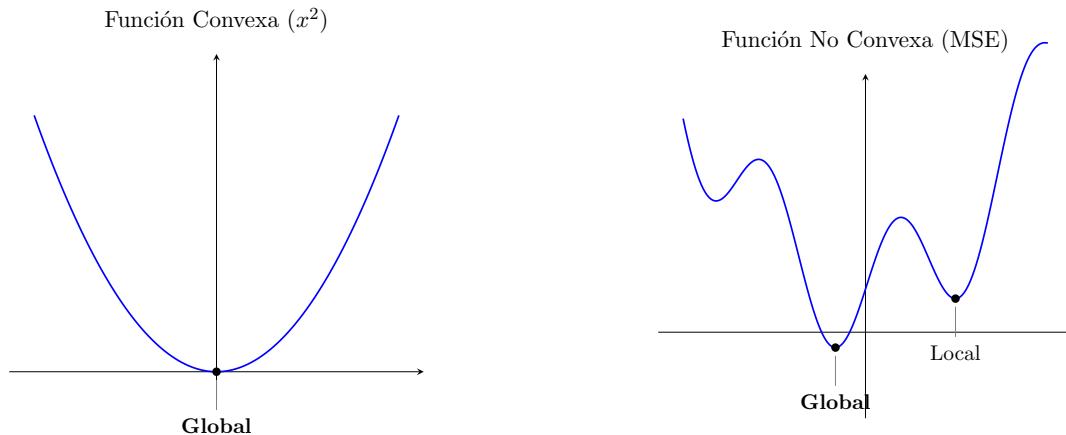


Figura 1: Comparativa de mínimos en funciones convexas y no convexas.

Luego de la *demostración por dibujo*, podemos ver que una función convexa nos es sumamente útil al buscar optimizar nuestro modelo. Evitando formalidades y demostraciones matemáticas, tener una función convexa nos garantiza que el mínimo encontrado sea el mínimo global. Es decir, no tenemos “trampas” en dónde podamos caer al momento de buscar *puntos críticos* al tener solución de forma cerrada o utilizar *descenso de gradiente*.

Entonces, si fuésemos a utilizar MSE como función de pérdida para este modelo obtenemos una función “fea” dónde tenemos múltiples mínimos locales y puntos silla dónde se puede quedar *atascado* nuestro descenso de gradiente.

Recordemos que para el descenso de gradiente, iniciamos con pesos aleatorios y damos una cierta cantidad de *pasos* buscando la dirección opuesta a la dirección más *cuesta arriba* para buscar un mínimo. En este caso, es muy posible que estemos dando pasos para bajar y nos quedemos “atascados” sin dirección alguna porque ya no tenemos más inclinación hacia algún punto en específico.

## El problema de la K en KNN

*Refierase a las slides correspondientes al tema de KNN. Con esto, considere el escenario donde estamos clasificando enfermedades raras. Tenemos un dataset de 1,000 pacientes sanos (Clase A) y solo 10 pacientes con la enfermedad (Clase B). Responda:*

*Si usted elige un K=15 (un número mayor a la cantidad total de casos positivos), ¿qué es lo más probable que ocurra matemáticamente con todas las predicciones para nuevos pacientes enfermos? ¿Por qué el algoritmo de “votación mayoritaria” falla aquí y qué técnica simple de preprocesamiento (vista en el lab anterior) sería obligatoria antes de usar KNN?*

En este caso, lo más probable que suceda es que los pacientes enfermos serán clasificados incorrectamente como pacientes sanos. Matemáticamente, para que un nuevo punto sea clasificado como Clase B, necesitaría que al menos 8 de sus 15 vecinos más cercanos pertenezcan a esta clase. Dado que tenemos únicamente 10 casos positivos en el dataset, incluso si el paciente es idéntico a los pacientes enfermos 5 de sus 15 vecinos siguen siendo Clase A. Dado que no necesariamente todos

los pacientes enfermos serán idénticos, es muy poco probable que existan predicciones de Clase B tomando en cuenta el desbalanceo de clases.

La votación mayoritaria falla porque asume que la densidad local de los puntos es representativa de la clase, esto causa un sesgo hacia la clase mayoritaria. En este caso, si fuésemos a aumentar aún más el valor de K se volvería imposible que la Clase B ganara alguna votación.

La técnica de preprocesamiento a utilizar aquí sería undersampling, dónde buscaríamos tomar únicamente 10 registros de personas sanas de manera que queden balanceadas las clases. También existen otras técnicas como oversampling, SMOTE o *mandar al junior a conseguir más datos*.

## Polinomios y la Realidad

*Observe las gráficas de las diapositivas 38 – 40. Con esto considere y responda. Si su modelo de regresión polinomial tiene una pérdida (Loss) de casi 0.0 en el set de entrenamiento (orden 8), pero al probarlo en producción con datos nuevos el error es gigante:*

*Que nombre recibe este fenómeno?*

Este efecto recibe el nombre de overfitting, dónde el modelo empieza a memorizar ruido dentro del dataset de entrenamiento en lugar de poder captar los patrones subyacentes. Un modelo más complejo puede ajustarse mejor al set de entrenamiento, pero también debemos de tomar en cuenta su capacidad de generalizar.

*Que esta modelando su algoritmo que no deberia modelar*

En este caso, podemos decir que el algoritmo está modelando ruido / comportamientos únicos dentro del set de prueba en lugar de patrones subyacentes.