

## Modelos :

A continuación, se presentan los modelos y problemas de optimización de los siguientes clasificadores, junto con los principales hiperparámetros a considerar para su ajuste posterior en el punto 2:

### 1. Naive Bayes (GaussianNB)

- **Modelo:** Es un clasificador probabilístico basado en el teorema de Bayes, que asume que las características son independientes entre sí.
- **Problema de optimización:** Maximizar la probabilidad a posteriori de cada clase dada una observación. Se ajustan los parámetros  $\mu$  (media) y  $\sigma$  (desviación estándar) de cada variable en cada clase.
- **Hiperparámetros clave:**
  - `var_smoothing`: Factor de suavizado para evitar divisiones por cero.

### 2. SGDClassifier

- **Modelo:** Implementa la regresión lineal o logística con optimización mediante descenso de gradiente estocástico (SGD).
- **Problema de optimización:** Minimizar la función de pérdida (por ejemplo, log-loss para regresión logística o hinge loss para SVMs) utilizando SGD para ajustar los coeficientes del modelo.
- **Hiperparámetros clave:**
  - `loss`: Tipo de función de pérdida (hinge, log, squared\_hinge, etc.).
  - `alpha`: Tasa de regularización.
  - `penalty`: Tipo de penalización (l1, l2, elasticnet).
  - `learning_rate`: Estrategia de actualización del gradiente.

### 3. LogisticRegression

- **Modelo:** Modelo estadístico que usa la función sigmoide para modelar la probabilidad de pertenencia a una clase.
- **Problema de optimización:** Minimizar la función de pérdida logarítmica (log-loss) utilizando métodos como Newton-CG, LBFGS o descenso de gradiente.
- **Hiperparámetros clave:**
  - `C`: Inverso de la fuerza de regularización.
  - `solver`: Algoritmo de optimización (lbfgs, liblinear, sag, etc.).
  - `max_iter`: Número máximo de iteraciones.

#### 4. Linear Discriminant Analysis (LDA)

- **Modelo:** Encuentra combinaciones lineales de variables predictoras que maximizan la separabilidad entre clases.
- **Problema de optimización:** Maximizar la relación entre la varianza inter-clase y la varianza intra-clase para encontrar la proyección óptima.
- **Hiperparámetros clave:**
  - solver: Algoritmo de optimización (svd, lsqr, eigen).
  - shrinkage: Parámetro de regularización cuando se usa lsqr o eigen.

#### 5. KNeighborsClassifier

- **Modelo:** Clasificación basada en la cercanía de los k vecinos más próximos en el espacio de características.
- **Problema de optimización:** Selección del número óptimo de vecinos (k) y la métrica de distancia adecuada (euclidiana, Manhattan, etc.).
- **Hiperparámetros clave:**
  - n\_neighbors: Número de vecinos a considerar.
  - metric: Métrica de distancia (euclidean, manhattan, etc.).
  - weights: Estrategia de ponderación de vecinos (uniform, distance).

#### 6. SVC (Support Vector Classifier)

- **Modelo:** Clasificador basado en la teoría de las máquinas de soporte vectorial (SVM), que encuentra un hiperplano óptimo para separar clases.
- **Problema de optimización:** Minimizar una función de margen suavizado con una penalización C para los errores de clasificación.
- **Hiperparámetros clave:**
  - C: Parámetro de regularización.
  - kernel: Tipo de kernel (linear, poly, rbf, sigmoid).
  - gamma: Parámetro del kernel en rbf, poly y sigmoid.

#### 7. RandomForestClassifier

- **Modelo:** Ensamble de múltiples árboles de decisión para mejorar la precisión y reducir el sobreajuste.
- **Problema de optimización:** Selección del número de árboles (n\_estimators), profundidad de los árboles y número de características consideradas en cada división.

- **Hiperparámetros clave:**
  - `n_estimators`: Número de árboles en el bosque.
  - `max_depth`: Profundidad máxima de los árboles.
  - `max_features`: Número de características consideradas en cada división.
  - `criterion`: Función de evaluación de la división (gini, entropy).

## 8. GaussianProcessClassifier

- **Modelo:** Modelo probabilístico basado en procesos gaussianos, que define una distribución sobre funciones.
- **Problema de optimización:** Ajustar los hiperparámetros del kernel, maximizando la verosimilitud marginal de los datos.
- **Hiperparámetros clave:**
  - `kernel`: Tipo de kernel utilizado (RBF, DotProduct, etc.).
  - `alpha`: Parámetro de suavizado.

Cada uno de estos clasificadores tiene desafíos particulares en la optimización de sus hiperparámetros y en la elección de la función de costo adecuada para obtener el mejor desempeño en la base de datos que trabajaremos a continuación.