

13 | 10/25

Término de aprendizaje de matrizes.

Punto 1

2025-2

1. Punto al modelo de regresión, buscamos resolver

diferentes modelos basados en el problema.

→ Definiciones como:

$f \in \mathbb{R}^N$; vector de salidas

$X \in \mathbb{R}^{N \times P}$; matriz de entradas

$\phi(X) \in \mathbb{R}^{N \times Q}$; matriz de características
con $\phi(x_n) \in \mathbb{R}^Q$

Finalmente $w \in \mathbb{R}^Q$ como vector

de parámetros.

y presentamos el problema y su

de optimización o inferencia mediante
asumiendo datos i.i.d.

i) Minimos cuadrados (OLS)

Buscamos minimizar el error cuadrático:

$$\text{f.g.: } J(w) = \|\Phi w - f\|^2$$

$$= (\Phi w - f)^T (\Phi w - f)$$

equivalente a maximizar verosimilitud bajo

error gaussiano con media cero

$$\int \text{varianza de } \hat{m}_n \sim N(m_n | 0, \sigma^2),$$

$$\text{En solución cerrada es } W_{ols} = (\Phi^T \Phi)^{-1} \Phi^T f \\ (\text{si } \Phi^T \Phi \text{ es invertible}). \quad \underline{\text{demonstración}}$$

puntual de $f = \Phi(x_n) w + m_n$ j. sin prior o bayes

$$\Rightarrow \|f - \Phi(x_n) w\|_2^2 \quad ; \text{ pero } J(w) = \|f - \Phi w^T\|_2^2$$

$$\Rightarrow \|f - \Phi w^*\|_2^2 = \langle f_n - \Phi w^* - f_n - \Phi w^* \rangle$$

$$\Rightarrow \|f - \Phi w^*\|_2^2 = (f_n - \Phi w^*)^T \cdot (f_n - \Phi w^*)$$

$$= f_n^T f_n - f_n^T (\Phi w^*) - (\Phi w^*)^T f_n + (w^T \Phi) (\Phi w^*)$$

$; f_n^T (\Phi w^*) - (\Phi w^*)^T f_n = 2 f_n (\Phi w^*)^T$

por do y se minimiza la f. de costo, e igualando
a cero, tenemos que:

$$\frac{\partial J(w)}{\partial w} = f_n^T f_n - 2 f_n^T \Phi w^* + (w^T \Phi)^T (\Phi w^*) = 0$$

j quitamos parámetros j restamos

$$\Leftrightarrow \frac{\partial J(w)}{\partial w} = f_n^T f_n - 2 f_n^T \Phi w^* + w^T \Phi^T \Phi w^*$$

$$\Rightarrow 0 - 2 f_n^T \Phi w^* + 2 w^T \Phi^T \Phi w^* = 0$$

y resolvendo para w:

$$2 w^T \Phi^T \Phi = 2 f_n^T \Phi$$

$$w = f_n^T \Phi (\Phi^T \Phi)^{-1}$$

$$\boxed{\tilde{w} = (\Phi^T \Phi)^{-1} \Phi^T f_n}$$

siempre y cuando $(\Phi^T \Phi)$ es
invertible

(vi) Minimizando los errores regularizados

(Ridge)

Se logra cuando se regulariza la

$$J(w) = \| \Phi w - f \|^2 + \lambda \| w \|^2$$

si $\lambda > 0$ el parámetro lambda

de regularización debe ser menor a cero,
es decir, asumiendo los N muestra y la matriz de
diseño $\Phi \in \mathbb{R}^{N \times m}$ y la matriz de
y en muestra sea: $\{(+ \Phi w^*), (- \Phi w^*)\}$

resolviendo en forma cuadrática en la f. a optimizar:

$$J(w) = (+ - \Phi w)^T (+ - \Phi w) + \lambda w^T w.$$

Ahora, para expandir la función, descomponemos el
producto cuadrático:

$$\Rightarrow (+ - \Phi w)^T (+ - \Phi w) = f^T f - \underbrace{f^T \Phi w - w^T \Phi^T f + w^T \Phi^T \Phi w}_{= f^T f - 2f^T \Phi w + w^T \Phi^T \Phi w.}$$

si $f^T f$ es de respecto a w , y añadiendo la
tercera:

$$J(w) = f^T f - 2f^T \Phi w + w^T \Phi^T \Phi w + \lambda w^T w$$

ahora, continuemos el gradiente ($\Delta J(w)$) y lo derivaremos a cero, esto con el fin de expresar la derivada del costo y condición de optimidad

$$\text{recordando, } \frac{\partial (w^T A w)}{\partial w} = (A + A^T) w \quad \text{y} \quad \frac{\partial (\lambda w^T w)}{\partial w} = \lambda w.$$

$$\sim -2f^T \Phi w \text{ es escalar} \rightarrow \frac{\partial (-2f^T \Phi w)}{\partial w} = -2\Phi^T f.$$

$$\sim \Phi^T \Phi \text{ es simétrica} \rightarrow \frac{\partial (\Phi^T \Phi)}{\partial w} = 2\Phi^T \Phi w.$$

$$\sim \lambda w^T w \text{ es simétrica} \rightarrow \frac{\partial (\lambda w^T w)}{\partial w} = 2\lambda w,$$

de esta forma, el término f^T desaparece al derivar.

$$\Rightarrow \Delta J(w) = -2\Phi^T f + 2\Phi^T \Phi w + 2\lambda w$$

e igualando a cero para optimizar:

$$\Delta J(w) = 0 = -2\Phi^T f + 2\Phi^T \Phi w + 2\lambda w \quad \text{y simplificando:}$$

$$\hookrightarrow \Phi^T \Phi w + \lambda w = \Phi^T f.$$

De donde, si factorizamos w ,

$$(\Phi^T \Phi + \lambda I)w = \Phi^T f, \quad \text{de donde, se } \Phi^T \Phi + \lambda I$$

es invertible (ie $\lambda > 0$) su única sol será:

$$\boxed{\tilde{w}_{\text{Ridge}} = (\Phi^T \Phi + \lambda I)^{-1} \Phi^T f}$$

(ii) Máximo Verosimilitud (MLE)

En este caso, asumimos que las observaciones f_n se generan según un modelo lineal con ruido gaussiano $f_n = \phi(X_n)^T w + \epsilon_n$; $\epsilon_n \sim N(0, \sigma^2)$ y el vector de señales presenta la siguiente distribución: $\phi(t | X, w) = N(t | \phi(X, w), \sigma^2 I)$

Es decir, cada señal f_n es una muestra de una distribución normal cuya media depende de w .

distribución normal cuya media depende

de w , y tiene varianza fija
y constante.

Una vez desembocada la función de verosimilitud,

se maximiza con respecto a w , lo

que equivale a minimizar el error

cuadrático en OLS.

procedemos con la verosimilitud para los parámetros w , siendo:

$$L(w) = \rho(t | X, w, \sigma^2) = (2\pi\sigma^2)^{-n/2} e^{-\frac{1}{2\sigma^2} \|t - \phi(X, w)\|^2}$$

donde, normalmente en el logaritmo de verosimilitud, se tiene:

$$f(w) = \log L(w) = -\frac{N \log(2\pi\sigma^2)}{2} - \frac{1}{2\sigma^2} \|t - \Phi w\|_2^2$$

para obtener el estimador de máxima verosimilitud (MLE) de w , se maximiza $f(w)$ respecto a w .

Vemos que el primer término no depende de w ,

$$\Rightarrow \tilde{w}_{MLE} = \underset{w}{\operatorname{argmax}} f(w) = \underset{w}{\operatorname{argmin}} \|t - \Phi w\|_2^2 \text{, esto equivale a minimizar el término cuadrático, siendo esto la formulación de OLS.}$$

procedemos expandiendo el término cuadrático:

$$\|t - \Phi w\|_2^2 = (t - \Phi w)^T (t - \Phi w) = t^T t - 2t^T \Phi w + \dots + w^T \Phi^T \Phi w$$

Derivando con respecto a w :

$$\frac{\partial f(w)}{\partial w} = \underbrace{-\frac{1}{\sigma^2}}_{= 0} (-2\Phi^T \Phi w) = 0.$$

simplificando y resolviendo:

$$\Phi^T \Phi w = \Phi^T t$$

entonces, la solución cerrada sería:

$$\tilde{w}_{MLE} = (\Phi^T \Phi)^{-1} \Phi^T t, \text{ podemos escribir } \sigma^2:$$

$$\sigma^2 = \frac{1}{N} \|t - \Phi \tilde{w}_{MLE}\|_2^2$$

(V) Máxima a posteriori (MAP)

En este caso, además del modelo de verosimilitud gaussiana, se introduce un conocimiento previo sobre los parámetros w , en forma de distribución $p(w) = \mathcal{N}(w \parallel 0, \sigma^2 I)$

• Esto indica que, antes de observar los datos, se cree que w es una variable aleatoria con distribución gaussiana y centrada en cero.

De esta forma, en la distribución posterior

se obtiene optimizando Bayes f.y.

$p(w) \propto p(+|w) p(w)$

de forma elijo, y optimizando.

Jaynes en el problema de optimización

busca al Ridge con $\lambda = \sigma^2 \alpha$.

De esta forma, $w_{MAP} = \arg \max_w P(w|+)$

$$p(w|+) \cdot p(w) = p(w|+) \cdot p(+); \text{ por Bayes}$$

Donde el posterior: $p(w|+) = \frac{p(+|w) \cdot p(w)}{p(+)}$

;

$p(+|w)$: Likelihood $\sim p(w)$: Prior.

De esta forma, sabemos que:

$$p(t_n | \varphi(x_n) w^\top, \sigma_m^2) = N(t_n | \varphi(x_n) w^\top, \sigma_m^2)$$

$$\Rightarrow w_{MAP} = \arg \max_w \log \left(\prod_{n=1}^N N(t_n | \varphi(x_n) w^\top, \sigma_m^2) \right) \prod_{q=1}^Q N(w_q | \varphi, \sigma_m^2)$$

resolviendo tenemos:

$$w_{MAP} = \underset{w}{\operatorname{arg\,min}} \frac{1}{2\sigma^2_m} \|f - \varphi w^\top\|_2^2 - \frac{1}{2\sigma^2_m} \|w\|^2$$

y como sabemos $\lambda = \frac{\sigma^2_m}{\sigma^2_w}$, finalmente
nuestro optimizador se define f.g así:

$$w_{MAP} = f^\top \varphi (\varphi^\top \varphi + \lambda I)^{-1}$$

r) Regresión lineal bayesiana (Gaussian model)

Aquí, consideramos w como una variable

aleatoria con prior gaussiano y muestreo de verosimilitud gaussiana.

De la forma que:

$$\text{Prior : } w \sim N(0, \alpha^{-1} I)$$

$$\text{Likelihood : } p(t|w) = N(t | \Phi w, \beta^{-1} I)$$

De igual manera, la posterior sobre w

es también gaussiana:

$$p(w|t) = N(w | m_w, s_w)$$

si m_w es el medio posterior

s_w es la variancia posterior.

De esta forma, hemos visto, para una nueva entrada

x^* , la predicción también sigue

una distribución normal y vectorial.

Esto proporciona, además de una predicción puntual,

también una medida de incertidumbre

. Hemos visto entonces como en el modelo lineal:

$$f_n = \phi(X_n)^T w + \epsilon_n ; \quad \epsilon \sim N(0, \sigma_n^2)$$

cuando f_n es una observación con ruido gaussiano.

en y, el vector de subtus:

$$f_n \in \mathbb{R}^n, \quad f = \phi w + \epsilon \quad ; \quad \epsilon \sim N(0, \sigma_n^2 I_n)$$

presenta una distribución condicional:

$$p(f|w) = N(f, \phi w, \sigma_n^2 I_n)$$

y de igual forma $\phi \in \mathbb{R}^{N \times n}$ es la matriz de diseño
con filas $\phi(X_n)^T$

$\Rightarrow p(w) = N(w | m_0, S_0)$, sera la distribución
a priori gaussiana para los pesos.

Esta expresión da creencia previa sobre los parámetros
antes de ver los datos.

Podemos aplicar Bayes:

$$\Rightarrow p(w|t) = \frac{p(t|w) \cdot p(w)}{p(t)} \propto p(t|w) p(w)$$

donde ambos términos son gaussianas, ergo, el posterior se define:

$$p(w|t) = N(w | m_n, s_n)$$

si m_n y s_n se determinan cumpliendo

el wadrado en el exponente.

Ahora, es necesario derivar el posterior.

Comenzando con el log. del producto: $p(t|w) \cdot p(w)$

$$\Rightarrow \log p(w|t) = \log p(t|w) + \log p(w) + C.$$

podemos sustituir las expresiones gaussianas

$$\Rightarrow \log p(t|w) = -\frac{1}{2} \frac{\|t - \varphi w\|^2}{\sigma_n^2} \sim$$

$$\log p(w) = -\frac{1}{2} (w - m_0)^T S_0^{-1} (w - m_0)$$

sumando ambos logs:

$$\log p(w|t) = -\frac{1}{2} \|t - \varphi w\|^2 - \frac{1}{2} (w - m_0)^T S_0^{-1} (w - m_0) + C$$

Si expandimos los términos cuadráticos en w :

$$\log p(w|t) = -\frac{1}{2} (t^T t - 2t^T \varphi w + w^T \varphi^T \varphi w) - \frac{1}{2} (w^T S_0^{-1} w - 2m_0^T S_0^{-1} w + m_0^T S_0^{-1} m_0)$$

$$\Rightarrow \log p(w|t) = -\frac{1}{2} w^T \left(\frac{1}{\sigma_n^2} \varphi^T \varphi + S_0^{-1} \right) w + w^T \left(\frac{1}{\sigma_n^2} \varphi^T t + S_0^{-1} m_0 \right) + C$$

Observemos, como la forma anterior es válida para en w ,

y sabemos que una gaussiana multivariada tiene exponente: $\left(\frac{-1}{2} w - m_n\right)^T S_n^{-1} (w - m_n)$

Agora que la posterior sobre w es:

$$p(w|t) = N(w | m_w, s_w)$$

$$; s_w = \left(S_0^{-1} + \frac{1}{\sigma_n^2} \varphi^T \varphi \right)^{-1}$$

$$m_w = S_w \left(S_0^{-1} m_0 + \frac{1}{\sigma_n^2} \varphi^T t \right)$$

Finalizamos con la predicción bayesiana,

para una nueva entrada x_* , donde ya vimos, el

objetivo es predecir f_* , marginalizando sobre la incertidumbre en w :

$$p(f_* | x_*, t) = \int p(f_* | x_*, w) p(w | t) dw.$$

y dado que ambos son gaussianas,

la integral produce la:

$$\Rightarrow p(f_* | x_*, t) = N(f_* | \mu_*, \sigma_*^2)$$

$$; \mu_* = \varphi(x_*)^T m_n \sim$$

$$\sigma_*^2 = \sigma_n^2 + \varphi(x_*)^T S_n \varphi(x_*)$$

; μ_* e σ_*^2 es la media y varianza predicción.

vi) Regresión ridge Kernel.

(Kernel ridge regression)

Este modelo permite modelar relaciones no lineales mediante el uso de funciones

Kernel. En lugar de usar directamente

los pesos w , expresamos el modelo

como una combinación de kernels:

$$f(x) = \sum_{n=1}^N \alpha_n k(x, x_n),$$

$$(w_n \text{ son los } \alpha = (K + \lambda I)^{-1} f)$$

K es la matriz de Gram

construida sobre el Kernel $k(x_i, x_j)$

vii) Proceso gaussiano para regresión (GPR)

Este modelo bayesiano NO paramétrico

asume que, todo función $f(x)$ es

distribuida como un proceso gaussiano

$$\text{f.g.: } f(x) \sim GP(0, k(x, x'))$$

Ahora, dado un conjunto de datos

se puede calcular la distribución condicional (predicción) para un nuevo punto.

$$\text{f.g.: } p(f_{x_*} | X, f, x_*) = \mathcal{N}(f_{x_*}, \text{var}(f_{x_*}))$$

cuya predicción presentar

también una incertidumbre.

Venus con el modelo observacional Andreev

$$\text{Gaussian} \quad f = f + \epsilon, \quad \epsilon \sim N(0, \sigma^2 \mathbb{I}_n)$$

Prob el conjunto de entrenamiento $(x_i, +)$ y una entrada x_* ,

el conjunto se define como:

$$\begin{bmatrix} t \\ f_* \end{bmatrix} \sim N \left(\begin{pmatrix} 0 \\ k_* + \alpha_n^2 \frac{1}{k_*} \end{pmatrix}, \begin{pmatrix} k_* & k_{**} \\ k_{**}^\top & k_{***} \end{pmatrix} \right)$$

;

Este es la matriz de entrenamiento

$$j \quad k_* = [k(x_1, x_*) \dots, k(x_N, x_*)]$$

$$r_{**} = k(x_s, x_a)$$

Fishermetrie, de las propiedades gaussianas multivariantes, el posterior predictivo es:

$$\phi(f_*, x_*, \chi, t) = N(\mu_*, \sigma_*^2)$$

$$j \quad M_n = k_*^+ (k + \sigma_n^2 I)^{-1} f.$$

$$\text{and } \hat{\sigma}_n^2 = k_{*1} - k_*^\top (k_* + \hat{\sigma}_n^2 I)^{-1} k_*$$

Fineamente, presentamos una tabla comparativa para discutir similitudes y diferencias entre estos modelos:

| Modelo | Mátriz | Función de costo | Diferencias |
|-----------------------------|--------------------------------|--|---|
| OLS | Determinista | Igual a MLE sin prior | No tiene regularización |
| Mínimos cuadrados regulares | Matriz f_2 | similar a OLS con regularización | equivalente a MAP con prior gaussiano |
| MLE | Probabilístico gaussiano | coincide con OLS | Baixada de Maximo a post. |
| MAP | Probabilístico regularizado | similar a Ridge. ($\lambda = \sigma_n^2 / \sigma_w^2$) | MLE con autorregularización |
| Regresión Bayesiana | Probabilístico completo | Extiende el MAP. | Opción distribución completa: $\mu = w_{MAP}$ |
| Kernel Ridge | Determinista en espacio kernel | Igual a Ridge pero en espacio Kernel | No pasa interpretación probabilística |
| Proceso Gaussiano. | Bayesiano no paramétrico. | Media similar a KRR pero con varianza preditriva | Versión Bayesiana del KRR. |

De esto, podemos inferir que,

- (i) todos los modelos derivan de la formulación del OLS, variando la forma de manejar la incertidumbre
- (ii) Existe una similitud jerárquica a medida que el modelo se vuelve complejo.
- (iii) Modelos determinista \rightarrow F. de costo \neq Probabilístico \rightarrow Posterior

L. Con relación a los algoritmos de
 Regresión visto, discutimos algunos
 características de rendimiento y formulación
 matemática, de los algoritmos (Linear Reg.,
 Lasso, elastic net, Kernel Ridge, etc...)

| Regresión | Modelo matemático | Función de costo (J) | Optimización | Expresión de regresión | Finalidad |
|----------------|--|---|-------------------------------|----------------------------------|-----------|
| Linear Reg. | $f = \Phi w^T y$ | MSE (mean square error) | Análitica ((closed-form)) | OLS / ML (maximum likelihood) | Alta. |
| Lasso | Linear + penalización lineal | $MSE + \lambda \ w\ _1$ | Gradiente descendente | Regularización L1 MAP Laplace | Media |
| Elastic net. | Linear + penalización lineal + $\lambda_2 \ w\ ^2$ | $MSE + \lambda_1 \ w\ _1 + \lambda_2 \ w\ ^2$ | Grad. descendente | Interp. Lasso - Ridge | Media |
| Kernel Ridge. | $f(x) = \sum x_i k(x, x_i)$ | $\ f - K\alpha\ ^2 + \lambda \ K\alpha\ ^2$ (Dual (Kernel)) | Gradiente descendente | Ridge kernelizado / MAP. | Alta |
| SGD Regressor | $f = \Phi w^T y$ | MSE | Gradiente descendente | Iteración OLS | Alta |
| Gaussian Ridge | $y = Xw + b$ | Posterior sobre dos pesos | Regularización probabilística | Bayesian Ridge | Media |

| | | | | | |
|-----------------------------------|--|---|--------------------------------------|---|-------------------------------|
| Gaussian Process | $f(x) \sim GP(0, k)$ | • J oy variose imulat negativo | • Máximo verosimilitud | - Proceso Gaussiano | - Limitada por $O(n^3)$ |
| Support Vector regressor (SVR) | $f(x) = \sum_i (d_i + d'_i) k(x_i, x) + b$ | $\begin{aligned} & \min_{w, b, \epsilon, \epsilon'} \frac{1}{2} \ w\ ^2 + \\ & C \sum (\epsilon_i + \epsilon'_i) \end{aligned}$ | • Programación cuadrática (QP) | • Regulación de margen; (formulación dual correlacionada del OLS) | • Media (kernel de rbf) |
| Random Forest regressor | $\hat{f}(x) = \frac{1}{M} \sum_m h_m(x)$ | • F. de voto por cada nodo $MSE = \frac{1}{N_{\text{nodo}}} \sum (y_i - \hat{y}_{\text{nodo}})^2$ | • Bagging + partición | • Ensemble no paramétrico. | • Alta |
| Gradient Boosting / XGBoost | $\hat{f}(x) = \sum_{k=1}^K f_k(x)$ | $\begin{aligned} & \sum_i L(y_i, \hat{y}_i) + \sum_k Q(f_k) \\ & L(y_i, \hat{y}_i) = \frac{1}{2} (y_i - \hat{y}_i)^2 \end{aligned}$ | • Boosting por gradiente | • Ensemble aditivo (Gradiente en el espacio dimensional de predictores) | • Muy eficiente |

3. RAPIDS (GPU accelerated data science)

Comenzamos definiendo. "RAPIDS . ai" y lo que nos ofrece ..

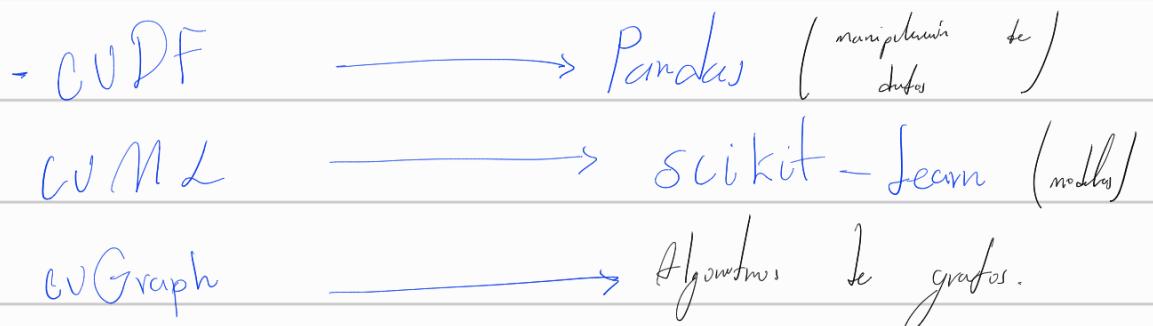
RAPIDS AI es un sistema de librerías de código abierto, esto gracia decir que está enfocada hacia el desarrollo libre de software eficiente; esto gracias a permitir su aceleración por GPU, desarrollando para NVIDIA principalmente.

Alyansus cyntes:

- Está basado en CUDA que es una forma de para el desarrollo de aplicaciones preferidas por GPU.
- Reduce el tiempo de cálculo usando GPU.
- Debido a la alta demanda se transportaron los "métodos" ó modelos de aprendizaje de las librerías Scikit-learn y numpy/pandas para el manejo de datos y tensores.

Una vez se revisó la documentación sobre el RAPIDS, se vio que los signos de equivalencias con otros modelos definidos anteriormente como el linear, logístico, bayesiano, etc..

→ Vemos que, para acceder a los modelos definidos en la librería "Cuml", simplemente se accede al método de muestra clase.



La ventaja de CML reside en su paralelismo para datasets grandes.

- Finalmente, vemos una tabla comparativa entre diferentes modelos, su implementación en RAPIDS, y algunas de las hipótesis más relevantes.
e.g., para los métodos linear regression, su implementación es mediante `Cuml - Linear Regression...`

$\mathcal{L}_{\text{generalization}}$ CPV \leftrightarrow GPV

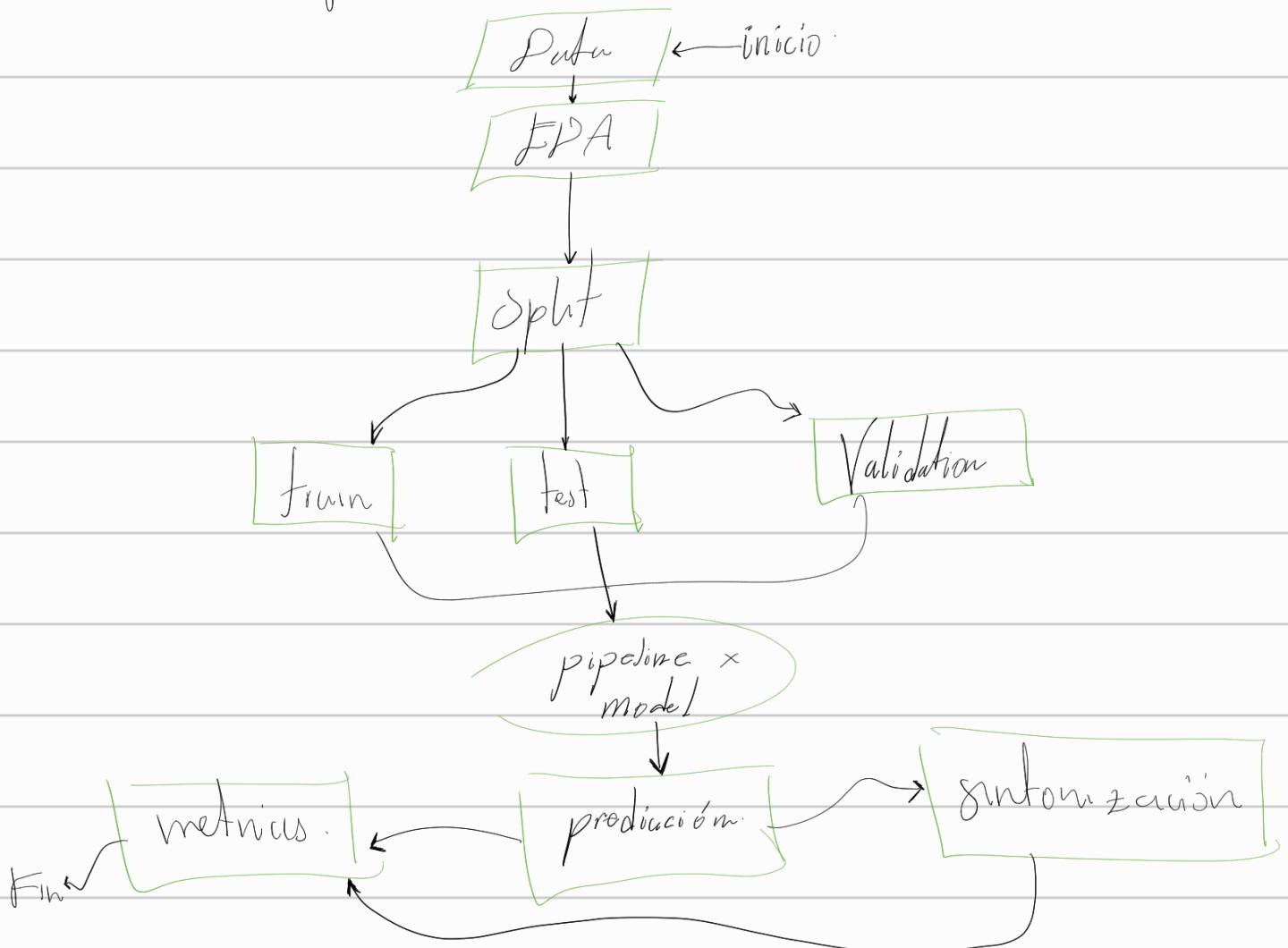
| Method | subset-learn | RAPIDS | Close | Hyperparameters |
|-----------------------|--------------------------------------|----------------------|--|--------------------------|
| Linear Reg | Cvm L. Linear Regression | Ols | fit_intercept | |
| Ridge | Lemppänen por Lasso / elastic net | — | CD | alpha, max_iter, tol. |
| Lasso | Cvm L. Lasso | CD | alpha, l1_ratio | |
| Elasticnet | Cvm L. Elastic Net | CD | alpha, l1_ratio | |
| Kernel Ridge | X | — | — | — |
| SGD Regressor | Cvml. SGD | SGD CPV - equivalent | lr, epochs, penalty | |
| Bayesian Ridge | X | — | — | — |
| Gaussian Process Reg | X | — | — | — |
| SVR | Cvml. SVR | SMO | C_epsilon, kernel_gamma | |
| Random Forest Reg | Cvml. Random Forest Regressor | Bayesian GPV | n_estimators, max_depth | |
| Gradient Boosting Reg | Cvml. XGB Regressor | Boosting | learning_rate, max_depth, subsample | |

A. Finalmente, trabajaremos en el nfl - big - data - bowl 2026 prediction, intentando implementar RAPIDS una vez comprendido el dataset.

a)

- Entender que el objetivo es predecir el movimiento de jugadores durante partidos en jugadas de la NFL usando datos de seguimiento a 10 Hz.

u.1 Procesos, componentes de la fase de datos para el EDA.



Apunta sobre el dataset y el
objetivo: