Laboratorio de Datos

Entrenamiento, testeo y validación

Primer Cuatrimestre 2025

Turno noche

Facultad de Ciencias Exactas y Naturales, UBA

Repaso: sistemas de ecuaciones lineales

Retomamos el ejemplo de inversiones. Sabemos que un fondo de inversión invirtió en acciones de YPF, Santander y Nvidia (y solo en estas acciones) pero no sabemos cuántas acciones compró de cada una. ¿Cómo podemos averiguarlo?

Suponemos que tenemos disponible:

- La valorización del fondo al final de cada día.
- El valor la acción de cada empresa al cierre de cada día.

Sistemas de ecuaciones lineales

Ponemos toda la información en la siguiente tabla.

Total	YPF	Santander	Nvidia
170262.00	20935	20100	37100.0
169929.50	21030	20500	36255.0
171064.00	20770	21700	36000.0
169637.35	20950	21000	35645.5
164625.45	20750	20316	33878.5

Cuadro: Valores diarios de acciones

Planteamos el sistema lineal

Llamamos c_1, c_2 y c_3 a la cantidad de acciones de cada tipo. Para calcular los valores, tenemos que resolver el siguiente sistema de ecuaciones:

$$
\begin{array}{lrcccc}
 & & YPF & Santander & Nvidia \\
 & & \downarrow & \downarrow & \downarrow \\
\text{Día 1} \rightarrow & 170262.00 &= 20935c_1 &+ 20100c_2 &+ 37100.0c_3 \\
\text{Día 2} \rightarrow & 169929.50 &= 21030c_1 &+ 20500c_2 &+ 36255.0c_3 \\
\text{Día 3} \rightarrow & 171064.00 &= 20770c_1 &+ 21700c_2 &+ 36000.0c_3 \\
\text{Día 4} \rightarrow & 169637.35 &= 20950c_1 &+ 21000c_2 &+ 35645.5c_3 \\
\text{Día 5} \rightarrow & 164625.45 &= 20750c_1 &+ 20316c_2 &+ 33878.5c_3
\end{array}
$$

Resolvemos el sistema

Como tenemos 3 incógnitas, nos alcanza con 3 ecuaciones:

$$\begin{aligned} 170262.00 &= 20935c_1 + 20100c_2 + 37100.0c_3 \\ 169929.50 &= 21030c_1 + 20500c_2 + 36255.0c_3 \\ 171064.00 &= 20770c_1 + 21700c_2 + 36000.0c_3 \end{aligned}$$

Triangulando la matriz y despejando, obtenemos los valores

$$c_1 = 3.2, \qquad c_2 = 2.0, \qquad c_3 = 1.7.$$

Estas son las cantidades de cada acción que tiene el fondo de inversión.

Validación del "modelo"

En base a los resultados que obtuvimos, ¿podemos confirmar que las acciones del fondo son las 3 acciones que usamos?

Si no estamos seguros si eran acciones de Santander o Galicia, ¿cómo podríamos asegurarnos?

Cambiamos los valores de Santander por los valores de Galicia:

$$
\begin{aligned}
170262.00 &= 20935c_1 + \mathbf{20100}c_2 + 37100.0c_3 \\
169929.50 &= 21030c_1 + \mathbf{19400}c_2 + 36255.0c_3 \\
171064.00 &= 20770c_1 + \mathbf{21900}c_2 + 36000.0c_3
\end{aligned}
$$

Sobreajuste

Resolvemos el sistema y obtenemos estos valores:

$$c_1 = 6.69507872, \qquad c_2 = 1.16828332, \qquad c_3 = 0.17838362$$

Los números no se ven menos redondos, pero eso no alcanza para decidir cuál es el modelo correcto.

Un sistema de 3 ecuaciones y 3 incógnitas en general **siempre** tiene solución.

Sobreajuste (overfitting). Cuando tenemos igual cantidad de parámetros que observaciones, el sistema (casi) siempre va a tener solución pero no nos da ninguna información sobre si el modelo es correcto, no podemos usarlo para estimar otros valores.

¿Cómo elegir entre distintos modelos?

¿Qué estrategias se les ocurren para ver cuál modelo es mejor?

¿Cómo elegir entre distintos modelos?

¿Qué estrategias se les ocurren para ver cuál modelo es mejor?

1. Utilizar más días al plantear el sistema de ecuaciones.

¿Cómo elegir entre distintos modelos?

¿Qué estrategias se les ocurren para ver cuál modelo es mejor?

1. Utilizar más días al plantear el sistema de ecuaciones.
2. Verificar la fórmula que obtuvimos en otros días.

¿Cómo elegir entre distintos modelos?

¿Qué estrategias se les ocurren para ver cuál modelo es mejor?

1. Utilizar más días al plantear el sistema de ecuaciones.
2. Verificar la fórmula que obtuvimos en otros días.

Las dos estretegias son ideas centrales en la construcción de modelos:

1. Utilizar la mayor cantidad posible de datos para construir el modelo.
2. Probar el modelo en datos distintos a los que usamos para construir el modelo.

¿Cómo elegir entre distintos modelos?

Toda tarea de «aprendizaje automático», «machine learning» o «inteligencia artificial», consiste en:

1. Tomar un problema relevante del mundo material.
2. Elegir un modelo matemático que lo represente. El modelo en general va a depender de ciertos parámetros $\beta_1, \ldots, \beta_s$. Por ejemplo en un modelo de regresión lineal, estos parámatros son los coeficientes de cada variable.
3. Definir una forma de medir qué tan bueno es un modelo, en relación a la realidad (vía los datos disponibles). Esto suele hacerse mediante una *función de pérdida*.
4. ...

¿Cómo elegir entre distintos modelos?

Función de pérdida: Si fijamos los parámetros de nuestro modelo $\beta_1, \ldots, \beta_s$ y aplicamos el modelo resultante a un conjunto de datos X, la función de pérdida mide cuánto "perdemos" utilizando el modelo en lugar de los datos reales.

En regresión lineal, la función de pérdida más común es el *error cuadrático medio (MSE)*.

Vamos a llamar $L(\beta \mid X)$ a la función de pérdida para un conjunto de parámetros $\beta = (\beta_1, \ldots, \beta_s)$ y un conjunto de datos X.

¿Cómo elegir entre distintos modelos?

Toda tarea de «aprendizaje automático», «machine learning» o «inteligencia artificial», consiste en:

1. Tomar un problema relevante del mundo material.
2. Elegir un modelo matemático que lo represente. El modelo en general va a depender de ciertos parámetros $\beta_1, \ldots, \beta_s$. Por ejemplo en un modelo de regresión lineal, estos parámatros son los coeficientes de cada variable.
3. Definir una una *función de pérdida* $L(\beta \mid X)$.
4. "Aprender" los coeficientes $\beta_1, \ldots, \beta_s$. Es decir, encontrar valores $\beta_1, \ldots, \beta_s$ que minimicen la pérdida.

Nivel 1: Entrenar y evaluar sobre todo el conjunto de datos

Como primer acercamiento, entrenamos nuestros modelos con un conjunto de datos X y evaluamos la perfomance sobre *los mismos datos* X.

En este contexto, si un modelo M_1 es *más complejo* que un modelo M_0 (es decir tiene más variables o parámetros), entonces necesariamente el valor de la función de pérdida será menor o igual.

Nivel 1: Entrenar y evaluar sobre todo el conjunto de datos

Ejemplo:

- M_0 : gastos mensuales $\sim \beta_0 + \beta_1 \cdot$ sueldo $+ \beta_2 \cdot$ cantidad de hijos
- M_1 : gastos mensuales $\sim \beta_0 + \beta_1 \cdot$ sueldo $+ \beta_2 \cdot$ cantidad de hijos $+ \beta_3 \cdot$ altura

El modelo M_1 es más complejo que M_0. Podemos ver a M_0 como un caso particular de M_1 tomando $\beta_3 = 0$.

El valor mínimo de pérdida de M_1 es menor o igual que el valor mínimo de pérdida de M_0:

$$L_{M_1}(\beta_0^*, \beta_1^*, \beta_2^*, \beta_3^* \mid X) \leq L_{M_0}(\beta_0^*, \beta_1^*, \beta_2^* \mid X)$$

(recordemos que al menos en teoría, cuando hacemos mínimos cuadrados obtenemos el mínimo absoluto)

Nivel 1: Entrenar y evaluar sobre todo el conjunto de datos

En este esquema, un valor de pérdida menor no nos asegura que el modelo M_1 sea mejor que M_0 (tenga mejor capacidad predictiva).

Que un modelo alcance un error muy pequeño durante el entrenamiento, puede ser tanto mérito propio del modelo como un síntoma de una excesiva parametrización, que le permite "interpolar" o "memorizar" los datos.

Ejemplo: Si tenemos n observaciones y tomamos un polinomio de grado $n-1$ o un modelo con n variables (independientes), vamos a poder obtener un modelo exacto, la pérdida será 0.

Nivel 2: Separar en conjuntos de entrenamiento («train») y prueba («test»)

Para evitar el *sobreajuste* (overfitting) de los modelos, es habitual dividir el conjunto de datos en dos partes mutuamente excluyentes (y conjuntamente exhaustivas, ¡nada se tira!).

Por ejemplo, utilizamos un $80\,\%$ de los datos para entrenamiento y un $20\,\%$ para testeo.

Entrenaremos cada modelo con los mismos datos de train para obtener los β óptimos, pero seleccionaremos como "mejor" a aquel modelo que minimice la pérdida sobre el conjunto de test.

Nivel 3: Split entrenamiento - validación - prueba

Como antes argumentamos que el modelo que minimiza el error de entrenamiento puede estar sobreajustándose a los datos, es igualmente posible que aquél que minimiza el error de prueba esté sobreajustándose a los datos de prueba: al fin y al cabo, así fue como definimos nuestra regla de selección (minimizar el error de prueba).

Dicho de otra forma, si probamos muchos modelos distintos, es posible que el mejor modelo funcione bien en los datos de prueba por casualidad.

Nivel 3: Split entrenamiento - validación - prueba

Para evitar este problema, se suele dividir el conjunto de datos en tres partes: entrenamiento, validación y test:

1. Entrenamos los modelos minimizando L en X_{train} (los datos de entrenamiento).
2. Una vez entrenados los modelo, medimo la pérdida L en X_{val} y seleccionamos el modelo con menor pérdida.
3. Evaluamos su performance "en el mundo real" con L en X_{test}.

Nivel 4: Validación Cruzada en k Pliegos ("K-fold CV")

En un esquema tripartito «train-val-test», el error de test sólo sirve para reporte, y achica el tamaño efectivo de la muestra.

Más aún, como hay un único conjunto de test, y todo el proceso está atravesado por ruido estocástico, la selección de modelos sigue teniendo un fuerte componente de azar.

Para evitar posibles sesgos en la selección de los conjuntos de entranamiento y validación, resulta conveniente realizar varias repeticiones del experimento con distintos conjuntos de datos. ¿Pero de dónde sacamos los datos para ello?

¡Pues los reutilizamos!

Nivel 4: Validación Cruzada en k Pliegos ("K-fold CV")

En validación cruzada de k pliegos ("k-fold cross-validation"), dividimos primero el conjunto de datos sólo en train y test.

Luego, partimos train en k partes iguales, que se rotarán el papel de validación: entrenamos y evaluamos el modelo k veces, cada vez dejando uno distinto de los k pliegos como *val* y el resto para *train*.