

SAA: Resumen

Aprendizaje No Supervisado

El aprendizaje no supervisado se utiliza cuando **no se dispone de las etiquetas y_i para cada individuo u objeto**. Los datos consisten en un vector de atributos x_i para cada individuo. Los problemas que resuelve el aprendizaje no supervisado incluyen el **análisis de clúster (agrupación de individuos)**, la **detección de anomalías** y la **reducción de la dimensionalidad**. El **análisis de clúster es central** para abordar los otros dos problemas.

El objetivo del análisis de clúster es **crear grupos** a partir de un conjunto de individuos u objetos. Un concepto fundamental en el análisis de clúster es la **distancia**, que permite definir si dos objetos dentro del mismo grupo están próximos entre sí.

Concepto de Distancia y Espacios

Una función de distancia entre dos objetos x e y debe cumplir las siguientes propiedades esenciales:

- Debe ser **no negativa** y valer cero solo si x es igual a y .
- Debe ser **simétrica** (la distancia de x a y es igual que la de y a x).
- Debe cumplir la **desigualdad triangular** (la distancia de x a y es menor o igual que la suma de la distancia de x a z y la distancia de z a y).

Funciones que cumplen estas propiedades definen un **espacio métrico**. Ejemplos comunes de distancias incluyen la **distancia euclídea** (la más usada), la **distancia de Manhattan** y la **distancia de Chebyshev**.

Un **espacio normado** es un caso particular de espacio métrico donde los vectores pueden sumarse y tienen una norma. La **norma** de un vector debe ser no negativa, escalar linealmente con la multiplicación escalar y cumplir la desigualdad triangular. Una norma **induce una distancia**, por ejemplo, la norma euclídea induce la distancia euclídea.

Algoritmos de Análisis de Clúster (No Supervisado)

1. K-Means

- Algoritmo iterativo que asigna puntos a centroides y actualiza la posición de los centroides.
- La **elección de los centroides iniciales es crucial** para evitar óptimos locales. Se proponen varias técnicas de inicialización (aleatoria, basada en jerárquico, densidad, etc.).
- Converge a un **óptimo local**, no garantizando encontrar el óptimo global. Se recomienda **inicializar el algoritmo varias veces** (ej. usando `n_init` en Python) y elegir la mejor solución (la de menor SSE - Sum of Squared Errors).
- Es importante **eliminar valores influyentes/outliers** de los grupos para que no perjudiquen a los centroides.
- La **escalabilidad de los datos** es muy importante para que la función objetivo (basada en distancias) sea objetiva.

- **Aplicación práctica:** Determinación de patrones de tráfico usando datos de velocidad de sensores. Requiere determinar el número de grupos (K). Permite identificar diferentes patrones (ej. sin congestión, moderada, congestionada) y analizarlos (días que se repite, periodos de congestión, velocidad media).
- Es necesario especificar el número de clusters (K).

2. Clustering Jerárquico

- El resultado es un **dendrograma**, un árbol que muestra la unión sucesiva de grupos. El eje Y representa la distancia interclúster. Un **corte en el dendrograma** a un nivel de distancia deseado permite obtener agrupamientos disjuntos.
- Queda definido por la **distancia entre objetos** y la **distancia entre clústeres**. Las distancias interclúster usuales son Vecino más próximo (Single), Vecino más alejado (Complete) y Distancia media (Average). Se utilizó Complete como ejemplo.
- Requiere una **matriz de distancia** entre todos los objetos. Se recomienda usar la distancia euclídea entre objetos.
- Se suele aplicar sobre **datos escalados**. PCA puede usarse para visualizar los resultados, pero el algoritmo se aplica sobre los datos escalados originales para no perder información.
- La **determinación del umbral de corte** en el dendrograma es manual y depende del problema.
- Permite obtener la etiqueta del grupo para cada observación. Se pueden caracterizar los grupos resultantes analizando los estadísticos descriptivos de las variables originales para los miembros de cada grupo.
- Tiene una **complejidad computacional alta** y puede ser lento.

3. Algoritmos Basados en la Densidad

- La idea clave es que los grupos se definen en **áreas de alta densidad** de datos, separadas por zonas poco densas.
- **DBSCAN (Density-based spatial clustering of applications with noise):**
 - Objetivo: Descubrir **clústeres con forma arbitraria** y ser eficiente en grandes bases de datos.
 - Solo requiere **dos hiperparámetros**: epsilon (radio del entorno) y minPts (número mínimo de puntos en el entorno para ser un punto núcleo).
 - Define la densidad de un punto como el número de puntos dentro de su entorno de radio epsilon.
 - Clasifica los puntos en **puntos núcleo** (alta densidad), **puntos alcanzables** (conectados a un punto núcleo) y **ruido** (no alcanzables desde ningún punto núcleo).
 - Introduce el concepto de **densamente alcanzable/conectado** para asegurar la simetría en la conexión entre puntos, a diferencia del concepto simple de directamente alcanzable.
 - Un clúster en DBSCAN es un conjunto de puntos densamente conectados donde todos los puntos son densamente alcanzables.
 - **Ventajas:** No necesita que se le indique el número de clústeres K. Puede encontrar clústeres no esféricos y de formas arbitrarias. Es **insensible a los outliers** al incorporar el concepto de ruido.

- **Desventajas:** Requiere la elección de epsilon y minPts.
- Ejemplo ilustrativo: Muestra cómo la elección de minPts afecta al resultado (pueden salir todos ruido o un cluster grande con puntos ruido).
- **DPC (Density Peak Clustering):**
 - Introducido en 2014.
 - También se basa en el concepto de densidad.
 - El objetivo fundamental es **identificar los centros de los clústeres**, que son puntos con mayor densidad.
 - Calcula la distancia entre todos los objetos y la densidad de cada punto (similar a DBSCAN o usando un kernel Gausiano).
 - Calcula delta, la distancia mínima a un punto con mayor densidad.
 - Los **centros de clúster tienen alta densidad y un valor de delta elevado**.
 - Propone un **grafo de decisión** (densidad vs. delta) para **identificar visualmente** los centros. Esta selección visual es una desventaja.
 - Una vez identificados los centros, los objetos se asignan al clúster del punto con mayor densidad más cercano a ellos. La asignación se realiza en una sola iteración.
 - Requiere definir umbrales para densidad y delta para seleccionar los centros.
 - **Ventajas:** Método rápido (asignación en una sola pasada).
 - **Desventajas:** La selección de centros es visual/manual. Puede **no detectar outliers** (un ejemplo mostró un outlier clasificado dentro de un grupo).

4. K-Means con Truco del Núcleo (Kernel K-Means)

- Motivación: El K-Means tradicional funciona bien si los datos son **linealmente separables** en el espacio original. Si no son linealmente separables, K-Means funciona mal.
- Solución: **Proyectar los datos a un espacio de características** (de mayor dimensión, a menudo infinita, como un espacio de Hilbert con núcleos reproductores - RKHS) donde podrían ser linealmente separables.
- Aplicar el algoritmo de **K-Means en este espacio de características**.
- Es necesario poder calcular distancias en el espacio de características, lo que requiere un producto escalar en ese espacio.
- El **Truco del Núcleo** permite calcular el producto escalar (y por tanto la distancia) entre los puntos proyectados en el espacio de características **sin necesidad de conocer explícitamente la función de proyección Φ** . Esto se hace evaluando una **función núcleo $K(x, y)$** sobre los datos originales.
- Ejemplos de funciones núcleo de Mercer: Lineal, Polinomial, Gaussiano.
- El algoritmo es similar al K-Means, pero los cálculos de distancia y la actualización de centroides se realizan en el espacio de características utilizando la matriz de Gram (K_{ij}) y el truco del núcleo.
- Permite agrupar datos que no están linealmente separados en el espacio original.

Reducción de la Dimensionalidad

1. Análisis de Componentes Principales (PCA):

- Técnica para **reducir el número de variables** de una base de datos simplificándola y agregando variables con atributos similares.
- Objetivo: Reducir la dimensionalidad manteniendo la **máxima representatividad/información** de los datos.
- Aplicaciones: **Visualizar datos multidimensionales** (proyectando a 2D o 3D). **Comprimir información** (imágenes, señales). **Acelerar algoritmos de aprendizaje automático** al reducir el número de variables.
- Proceso: Centrar y escalar los datos. Calcular la matriz de covarianza. Calcular los vectores propios (loadings) y valores propios. Ordenar los vectores propios por valor propio descendente. Seleccionar las m primeras componentes principales. Los **valores propios indican la varianza recogida** por cada componente. La **proporción de varianza explicada (PVE)** acumulada ayuda a elegir m. Elegir $m \leq 3$ permite la visualización gráfica.
- **Ventajas:** Fácil de realizar. Acelera algoritmos al reducir variables.
- **Desventajas:** Implica **pérdida de información**; hay que equilibrar reducción y pérdida. Elegir el número óptimo de componentes (m) es subjetivo. Puede verse como un análisis clúster dual (agrupar variables).

Validación en Aprendizaje No Supervisado (Análisis de Clúster)

La validación se discute para problemas de clasificación, regresión y análisis de clúster. En análisis de clúster, existen métricas de validación **interna** (usando solo los datos) y **externa** (usando etiquetas de verdad fundamental - "ground truth").

Métricas de Validación Externa (requieren etiquetas de verdad fundamental):

- **Homogeneidad:** Cada clúster contiene solo miembros de una única clase.
- **Exhaustividad (Completeness):** Todos los miembros de una clase dada son asignados al mismo clúster.
- **V-measure:** Media armónica de homogeneidad y exhaustividad. Útil para analizar cualitativamente los errores.
- **Adjusted Rand Index (ARI):** Ajustado para el azar, más fiable para conjuntos de datos pequeños o muchos clústeres.
- **Adjusted Mutual Information (AMI).**
- Nota: En algunos casos, resultados aleatorios pueden no dar puntuaciones nulas para estas métricas, especialmente con un gran número de grupos. Se pueden calcular utilizando librerías como scikit-learn.

Aprendizaje Semisupervisado

El aprendizaje semisupervisado es un **híbrido entre aprendizaje supervisado y no supervisado**, utilizando **tanto datos etiquetados como no etiquetados** para el entrenamiento del modelo. Su objetivo es **mejorar el modelo que se obtendría si solo se usaran instancias etiquetadas**, aprovechando la información de los datos no etiquetados disponibles durante el entrenamiento. Los datos no etiquetados proporcionan **contexto útil**.

Es útil porque **etiquetar datos es costoso y difícil**, especialmente para tareas complejas como la segmentación de imágenes. Permite ser aplicado en casos con **pocas instancias etiquetadas y muchas sin etiquetar**, como en procesos industriales para detección de fallos.

Se basa en **asunciones** sobre los datos no etiquetados y la relación entre los puntos de diferentes clases. Las cuatro asunciones principales son:

- **Asunción de suavidad (o continuidad):** Dos instancias cercanas en el espacio de entrada deben tener la misma etiqueta.
- **Asunción de baja densidad:** El límite de decisión debe pasar a través de un espacio de baja densidad.
- **Asunción de la variedad (manifold):** El espacio de entrada de alta dimensión está compuesto por subespacios de menor dimensionalidad (variedades), y las instancias en la misma variedad deben tener la misma etiqueta.
- **Asunción de clúster:** Las instancias en el mismo clúster deben tener la misma etiqueta.

Una condición necesaria es que los datos no etiquetados sean **relevantes** para la tarea, es decir, que la distribución de los datos de entrada contenga información sobre la probabilidad condicional de la clase.

Clasificación de Métodos Semisupervisados:

- **Transductivos:** Utilizan las etiquetas disponibles para predecir etiquetas para un **conjunto específico de datos no etiquetados**, que luego pueden ser usados por un clasificador supervisado. No entrenan un modelo general, solo etiquetan los datos no etiquetados dados.
 - Ejemplo: **Propagación de etiquetas**, un algoritmo basado en grafos que propaga etiquetas basándose en la proximidad (distancia euclídea u otra) y las asunciones de suavidad y clúster.
- **Inductivos:** Tienen como objetivo **entrenar directamente un modelo general** (clasificación o regresión) utilizando datos etiquetados y no etiquetados. Se diferencian en cómo incorporan los datos no etiquetados:
 - **Métodos *wrapper*:** Primero entrenan un clasificador solo con datos etiquetados, generan **pseudo-etiquetas** para los datos no etiquetados, y luego vuelven a entrenar el clasificador (o clasificadores) usando datos etiquetados originales y pseudo-etiquetados. Usar múltiples clasificadores (diversificación) ayuda a reducir la tendencia a reforzar malas predicciones iniciales.
 - **Preprocesamiento no supervisado:** Utilizan datos no etiquetados en una etapa separada **antes** del entrenamiento supervisado final. Se usan para extraer características útiles, pre-agrupar datos (ej. usando DBSCAN, aprovechando la asunción de baja densidad), o establecer parámetros iniciales de un modelo supervisado (pre-entrenamiento, similar a tareas de pretexto en aprendizaje autosupervisado). El pre-entrenamiento o extracción de características a menudo implica reducción de dimensionalidad, aprovechando la asunción de la variedad.
 - **Intrínsecamente semisupervisados:** Incorporan directamente las instancias no etiquetadas en la **función objetivo** o de optimización del método de aprendizaje.
 - Ejemplo: **Máquinas de Vectores de Soporte Semisupervisadas (Semi-supervised SVMs)**. Mapean datos a un espacio de características de mayor dimensión usando kernels para separar categorías no linealmente separables, aplicando la asunción de baja densidad al maximizar el margen.

- Arquitecturas de Deep Learning propuestas incluyen redes en escalera, pseudo-conjuntos, conjunto temporal y modificaciones a GANs.

Validación de modelos

El material aborda la **validación** de sistemas de aprendizaje automático, dividiéndola principalmente según el tipo de problema a resolver: **Clasificación**, **Regresión**, y **Análisis Clúster**. La validación es crucial para evaluar la calidad de los modelos construidos.

Validación en Problemas de Clasificación

En los problemas de clasificación, especialmente la **clasificación binaria** (con dos clases de salida, positivo o negativo), la herramienta fundamental para entender el rendimiento del modelo es la **Matriz de Confusión**. Esta matriz relaciona las clases reales con las clases predichas. Permite identificar:

- **Verdaderos Positivos (TP)**: Aciertos en la clase positiva.
- **Verdaderos Negativos (TN)**: Rechazos correctos de la clase negativa.
- **Falsos Positivos (FP)**: Predicciones positivas que en realidad eran negativas.
- **Falsos Negativos (FN)**: Predicciones negativas que en realidad eran positivas. La importancia de cada uno de estos factores (TP, TN, FP, FN) depende completamente del problema específico.

A partir de la matriz de confusión, se derivan diversas métricas de validación:

- **Accuracy (Exactitud)**: La fracción o recuento de predicciones correctas $(TP + TN) / \text{Total}$. Es un buen primer indicador, pero puede tener sesgo con datos desbalanceados.
- **Precision**: Proporción de verdaderos positivos entre el total de positivos predichos $(TP / (TP + FP))$. Mide cuán precisos fuimos al predecir positivo.
- **Recall (Exhaustividad o Sensibilidad - TPR)**: Proporción de verdaderos positivos entre el total de positivos reales $(TP / (TP + FN))$. Mide cuántos de los positivos reales hemos detectado.
- **Métrica F ($F\beta$)**: Una media armónica ponderada de Precision y Recall. El F1 score es una métrica común que combina ambas de forma equilibrada.
- Otras métricas como el Coeficiente de Correlación de Matthews también son populares, especialmente en competiciones.

Otra herramienta gráfica muy útil para la clasificación binaria es la **Curva ROC (Receiver Operating Characteristic)**. Esta curva representa la relación entre la **Sensibilidad (TPR)** y la **Especificidad** (o $1 - \text{Tasa de Falsos Positivos, FPR}$) de un clasificador al variar el umbral de decisión.

- La tasa de falsos positivos (FPR) se define como $FP / (FP + TN)$.
- El punto ideal en la curva ROC es la esquina superior izquierda ($TPR=1, FPR=0$).
- El **Área bajo la curva ROC (AUC)** es una medida numérica que resume el rendimiento del clasificador. Un valor de AUC cercano a 1.0 indica un modelo perfecto, mientras que un valor cercano a 0.5 indica un rendimiento aleatorio. El AUC representa la probabilidad de que el clasificador puntúe una instancia positiva elegida al azar más alto que una instancia negativa elegida al azar.

En problemas de **clasificación multiclase**, la Matriz de Confusión sigue siendo la herramienta más útil. Las filas representan las clases reales y las columnas las predichas. La diagonal principal muestra los aciertos. Es crucial tener en cuenta el **balanceo de las clases**, ya que los clasificadores tienden a favorecer las clases más frecuentes. La matriz de confusión normalizada ayuda a visualizar el rendimiento por clase independientemente de su frecuencia. Métricas como Accuracy, Precision y Recall se pueden extender a multiclase usando enfoques como el micro o macro average.

Validación en Problemas de Regresión

Los problemas de **regresión** (donde se predice un valor continuo) requieren métricas de validación diferentes a las de clasificación. La evaluación se basa en la diferencia entre el valor real y el valor predicho, conocida como **residuo o error**.

- Un valor agregado pequeño del error implica un buen modelo.
- El tratamiento de valores extremos o *outliers* es un problema importante en regresión.

Algunas métricas clave para evaluar modelos de regresión son:

- **Varianza Explicada**: Mide la proporción de la variación de los datos que el modelo explica. El mejor valor posible es 1.0.
- **Error Máximo (Max Error)**: Representa el error residual más grande. Captura el peor caso de error.
- **Error Absoluto Medio (MAE)**: El promedio de los errores absolutos $(|y_i - \hat{y}_i|)$. Es fácil de interpretar y todos los errores contribuyen proporcionalmente. Un MAE pequeño sugiere un buen modelo.
- **Error Cuadrático Medio (MSE)**: El promedio de los errores al cuadrado $((y_i - \hat{y}_i)^2)$. Penaliza más los errores grandes o valores atípicos que el MAE.
- **Error Cuadrático Logarítmico Medio (MSLE)**: El promedio de los errores logarítmicos al cuadrado $(\log(1+y_i) - \log(1+\hat{y}_i))^2$. Útil para objetivos con crecimiento exponencial. Penaliza más las subestimaciones.
- **Error Porcentual Medio Absoluto (MAPE)**: Promedio de los errores absolutos porcentuales $|y_i - \hat{y}_i| / |y_i|$. Es sensible a errores relativos y no cambia con un escalado global de la variable objetivo. Sin embargo, puede crecer inesperadamente con valores reales pequeños y está sesgado hacia las subestimaciones.
- **Coeficiente de Determinación (R^2)**: Representa la proporción de la varianza de la variable objetivo explicada por las variables independientes del modelo. Indica la bondad del ajuste. El mejor valor posible es 1.0 y puede ser negativo. El R^2 ajustado es una versión modificada que tiene en cuenta el número de variables. El R^2 describe mejor la capacidad del modelo para describir el fenómeno que para predecir nuevas muestras.

La selección de la métrica de error en regresión depende del objetivo del problema y de cómo se deben tratar los errores extremos.

Validación en Problemas de Análisis Clúster

La validación de los resultados de **clustering** es más difícil que en clasificación o regresión, ya que a menudo no se dispone de una verdad fundamental (*ground truth*). Los **índices de validez de los clústeres** se utilizan para evaluar qué tan bien se ajustan los resultados a las particiones naturales de un conjunto de datos.

- Una dificultad importante es que muchos algoritmos requieren definir previamente el número de grupos (K), el cual no siempre se conoce de antemano. Los índices de validez a menudo se usan para encontrar el número óptimo de grupos probando diferentes valores de K.
- Las medidas de validación se centran en la **compacidad** (proximidad de objetos dentro de un clúster) y la **separación** (aislamiento de los clústeres entre sí). Idealmente, los miembros de cada clúster están cerca y los clústeres están separados.

La validación del clustering se divide en:

- **Validación Interna:** Evalúa la calidad de una partición basándose únicamente en la estructura de los datos, sin referencia externa.
 - **Coefficiente Silhouette:** Mide cuán similar es un objeto a su propio clúster comparado con el siguiente clúster más cercano. Una puntuación más alta indica clústeres mejor definidos y separados. El rango es de -1 (agrupación incorrecta) a +1 (agrupación densa). Tiene un sesgo a favor de agrupaciones convexas. Se usa a menudo para elegir el K óptimo en k-means.
 - **Índice Calinski-Harabasz:** Basado en la relación de varianza (cociente entre la dispersión entre clústeres y la dispersión dentro de clústeres). Una puntuación más alta se relaciona con clústeres mejor definidos. También favorece clústeres convexas.
- **Validación Externa:** Evalúa la calidad comparando la estructura de los clústeres con una clasificación predefinida externa (*ground truth*). Requiere tener datos etiquetados.
 - **Rand Index (RI):** Compara pares de elementos para ver si están en el mismo grupo o en grupos diferentes en ambas particiones (la del modelo y la real). Su valor está entre 0 y 1.
 - **Adjusted Rand Index (ARI):** Una versión ajustada del Rand Index que descuenta el acuerdo esperado por azar. A diferencia del RI, un valor cercano a 0.0 indica una asignación aleatoria. Los valores cercanos a 1.0 indican un acuerdo significativo. Es más seguro usar ARI con menos muestras o muchos clústeres.
 - **Homogeneidad:** Mide si cada clúster contiene solo miembros de una única clase real. Un valor de 1.0 indica perfecta homogeneidad.
 - **Exhaustividad (Completeness):** Mide si todos los miembros de una clase real se asignan al mismo clúster. Un valor de 1.0 indica perfecta exhaustividad. Similar al recall en clasificación.
 - **V-measure:** La media armónica de Homogeneidad y Exhaustividad. Equilibra ambas medidas y ofrece una explicación intuitiva de los errores.
 - **Mutual Information (MI) / Normalized Mutual Information (NMI) / Adjusted Mutual Information (AMI):** Miden la concordancia entre las asignaciones de etiquetas, ignorando permutaciones. AMI se normaliza con respecto al azar, por lo que una asignación aleatoria tiene un valor cercano a 0.0, y un valor cercano a 1.0 indica un acuerdo significativo.

La mejor validación para clustering a menudo es la capacidad de los grupos identificados para resolver el problema real de negocio o la coherencia con el conocimiento del dominio.

Comparación de Algoritmos y Selección de Modelos

Los fuentes también mencionan que diferentes algoritmos de clasificación o clustering tienen fortalezas y debilidades distintas dependiendo de las características de los datos (por ejemplo, número de muestras, número de características, forma de las distribuciones, densidades).

- Por ejemplo, KNN funciona bien con suficientes datos, bajo ruido y pocas características. SVM es fuerte en clasificación binaria, especialmente usando kernels no lineales. Random Forest es bueno con muchos datos y características. DBSCAN es útil para detectar outliers y manejar formas diversas basadas en densidad. Los algoritmos jerárquicos son buenos para comprender la estructura de los datos con pocos elementos.
- La **selección del modelo** es un proceso clave. No hay una única solución para un problema. La elección depende de si se tienen datos etiquetados (problema de predicción: clasificación o regresión) o no (clustering). Dentro de cada categoría, se consideran las características de los datos y los objetivos del problema.
- La selección de modelos debe basarse en la definición del problema, incluyendo la **métrica de éxito** definida para el mismo. A menudo implica probar diferentes algoritmos y configuraciones (*hiperparámetros*) para encontrar el que mejor rendimiento ofrece según la métrica elegida en los datos de validación.

Distancia Euclídea

$$d^2(x, y) = \sqrt{\sum_{j=1}^p (x_j - y_j)^2}$$

SSE

$$\sum_{j=1}^p \sum_{k=1}^K \sum_{i \in C_k} (x_{ij} - \bar{x}_{ij})^2$$

Actualización centroides (C-means)

$$\bar{x}_k = \frac{\sum_{i=1}^n w_{ik}^m x_i}{\sum_{i=1}^n w_{ik}^m}$$

Actualización pesos (C-means)

$$w_{ik} = \frac{1}{\sum_{i=1}^K \left(\frac{d^2(x_i, \bar{x}_k)}{d^2(x_i, \bar{x}_s)} \right)^{\frac{2}{m-1}}}$$

Función BIC

$$BIC = -2 \sum_{k=1}^K \xi(k) + 2 \cdot K \cdot P \cdot \ln(n)$$

donde:

$$\xi(k) := -n_k \left(\sum_{j=1}^p \frac{1}{2} \ln(\hat{\sigma}_j^2 + \hat{\sigma}_{kj}^2) \right)$$

MSE

$$MSE(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{sample}-1} (y_i - \hat{y}_i)^2$$

MAE

$$MSE(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{sample}-1} |y_i - \hat{y}_i|$$

Varianza explicada

$$VarEx(y, \hat{y}) = 1 - \frac{Var(y_i - \hat{y}_i)}{Var(y)}$$