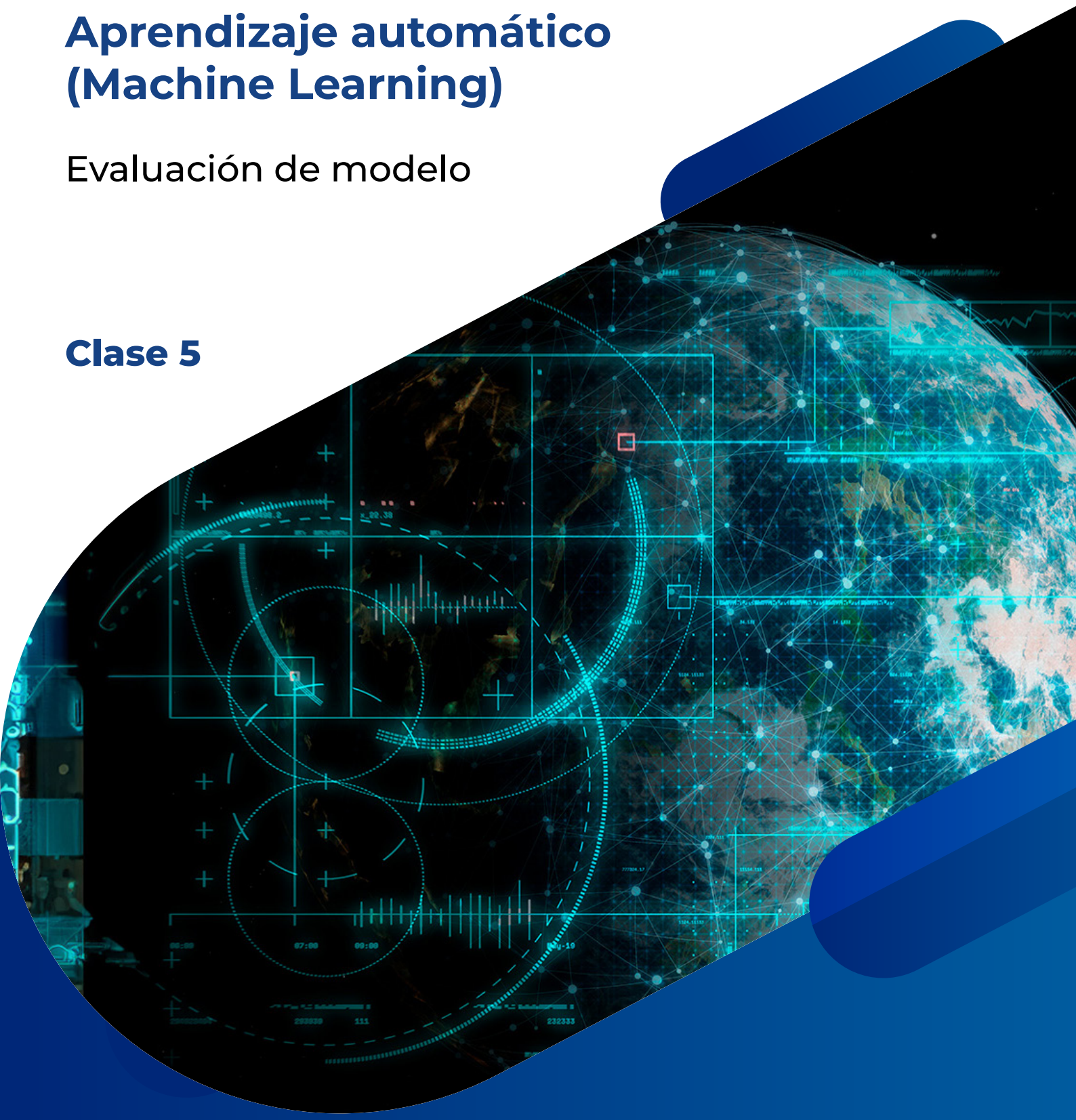
PUCE Virtual
EDUCACIÓN FLEXIBLE

Aprendizaje automático (Machine Learning)

Evaluación de modelo



Clase 5

MAESTRÍA EN
SISTEMAS DE INFORMACIÓN
Mención Data Science

La excelencia no se improvisa

INTRODUCCIÓN

La evaluación de modelos es un proceso esencial en *machine learning*, ya que permite medir el rendimiento de un modelo y comparar distintas opciones para seleccionar la más adecuada, según el problema que se de debe resolver (James et al., 2021). Un aspecto fundamental en la clasificación es el uso de la **matriz de confusión y métricas derivadas, como precisión, sensibilidad y F1-score**, que proporcionan una visión detallada del desempeño del modelo (Fawcett, 2006). Además, la función de decisión del clasificador influye en la predicción final y puede ajustarse mediante umbrales para optimizar resultados específicos (Murphy, 2012). Herramientas como la **curva ROC y la curva Precision-Recall** permiten evaluar el balance entre tasas de falsos positivos y verdaderos positivos; esto facilita la toma de decisiones en modelos de clasificación binaria (Provost & Fawcett, 1997).

En el caso de problemas multiclase es necesario emplear métricas adaptadas, como el promedio ponderado o el macro de las medidas de evaluación (Sokolova & Lapalme, 2009). Para tareas de regresión, se utilizan métricas como el error cuadrático medio (**MSE**) y el coeficiente de determinación (R^2), que miden la exactitud de las predicciones numéricas (Draper & Smith, 1998). La selección de modelos implica optimizar clasificadores en función de distintas métricas y calibrar su salida probabilística para mejorar la interpretación de resultados (Niculescu-Mizil & Caruana, 2005). Este proceso garantiza que el modelo elegido no solo tenga un buen rendimiento en datos de entrenamiento, **sino que también generalice adecuadamente datos no vistos**.

RDA 2: Evalúa los modelos de aprendizaje automático para identificar y cuantificar potenciales sesgos y limitaciones, empleando métricas adecuadas.

Clase 5. Evaluación de modelo

Evaluación y selección de modelos

Existen algunos puntos importantes de cara a evaluar los modelos de *machine learning;* uno de ellos es entender que el *accuracy* brinda una mirada parcial del *perfomance* de los modelos de **clasificación**, otro punto es entender la motivación del uso y definición de las métricas de evaluación en algoritmos de *machine learning*. Adicionalmente, es crucial entender las métricas de evaluación de los modelos de *machine learning*, de cara a seleccionar el **mejor modelo** y sus **mejores parámetros**.

A continuación, se muestra una ilustración que explica la importancia del **proceso de evaluación de los modelos**. A grandes rasgos, lo que se busca es llegar a tener los mejores modelos, en función del resultado que se obtenga en su evaluación.

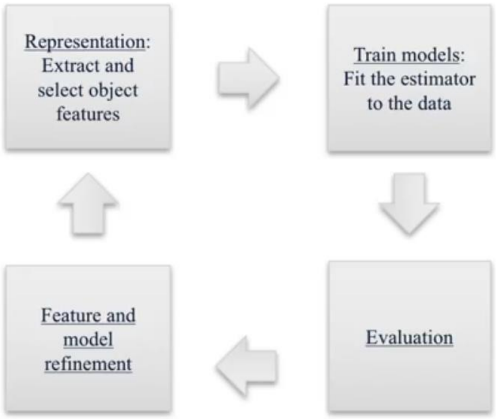


Figura 1. Refinamiento del ciclo de evaluación de modelos de *machine learning*

La métrica de evaluación del modelo puede variar, dependiendo del objetivo que tenga la implementación de este. La métrica *accuracy* es una de las más usadas, pero existen muchas otras, tales como:

- Índice de satisfacción del cliente (*Web search*)
- Revenue (e-commerce)
- Incrementar la tasa de supervivencia de pacientes (medicina)

***Accuracy* con clases no balanceadas**

Supongamos que tenemos dos clases (positiva y negativa) y tenemos 1 000 ítems, de los cuales solo 1 es de clase positivo, mientras que los restantes 999 son de clase negativa. Si el modelo clasifica a todas las clases como negativas, entonces tendrá un ***accuracy*** del 99,9%; pese a haber clasificado incorrectamente la única clase positiva.

$$accuracy \quad \frac{\#\ predicciones\ correctas}{total\ casos}$$

Un ejemplo de esta situación son las **transacciones fraudulentas** en una tarjeta de crédito, dado que hay muy pocos casos de transacciones fraudulentes en relación con el total.

En este punto es importante señalar que existen los ***dummy classifiers*****,** los cuales ignoran completamente los *input data* y se usan como un *sanity check*, para contrastar el *performance* del clasificador. Los *dummy classifiers* brindan una métrica nula, como *baseline*, y no deben ser usados para problemas reales, solo se usan como modelo de comparación básica con el modelo real.

Hay varias estrategias para el uso de los ***dummy classfiers*** en *scikit – learn:*

- La clase más frecuente
- Estratificado por clase
- Predicciones con distribución uniforme
- Valor constante

Algunas de las razones por las cuales un modelo puede tener un *accuracy* similar al de un ***dummy classifier*** son:

- *Features* inadecuadas o mal calculadas
- Una elección pobre de los hiperparámetros del modelo
- *Datasets* altamente desbalanceados

Por otro lado, existen los ***dummy regressors*** cuya funcionalidad es similar a la de los *dummy classifiers*; es decir, ignoran completamente los *data input* y se usan como *sanity check,* para contrastar el *performance* de un modelo de regresión. Los *dummy regressors* también brindan una métrica nula como *baseline,* y no deben ser usados para problemas reales, solo se usan como modelo de comparación básica con el modelo real.

Hay varias estrategias para el uso de los ***dummy regressors*** en *scikit – learn:*

- Media: predice la media de los *target* de entrenamiento
- Mediana: predice la mediana de los *target* de entrenamiento
- Cuantil: predice el cuantil (especificado por el usuario) de los *target* de entrenamiento
- Valor constante: especificado por el usuario.

Matriz de confusión y métricas básicas de evaluación

La matriz de confusión es una tabla que indica **'cuán confundido'** se encuentra el modelo en relación con los valores reales. A continuación, la Figura 2 explica este concepto sobre un modelo de clasificación binaria.

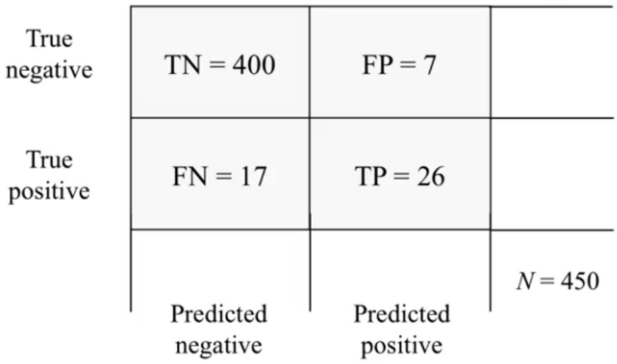


Figura 2. Matriz de confusión para un modelo de clasificación binaria.

La matriz de confusión combina las cuatro posibilidades de clasificación: 1) que el modelo clasifique correctamente la clase positiva ***true positive,*** 2) que clasifique incorrectamente la clase positiva ***false positive,*** 3) que clasifique correctamente la clase negativa ***true negative*** o, 4) que clasifique incorrectamente la clase negativa ***false negative.***

Accuracy: de la matriz de confusión se deriva el *accuracy,* que es la fracción de casos correctamente clasificados en relación con el total de casos:

$$accuracy = \frac{TN + TP}{TN + TP + FN + FP}$$

Para el ejemplo de la Figura 2 sería: $accuracy = \frac{400+26}{400+26+17+7} = 0.95$

Por otro lado, tenemos el error de clasificación *1 – accuracy*

$classError = \frac{FP+FN}{TP+TN+FP+FN}$

Para el ejemplo de la Figura 2 sería: $= classError = \frac{7+17}{400+26+17+7} = 0.060$

Recall (TPR): indica la fracción de todos los casos positivos que el modelo clasificó como positivos. El *recall* tambien es conocido como *True Positive Rate (TPR), Sensitivity o Probability of detection.*

$Recall = \frac{TP}{TP+FN}$ *Para el ejemplo de la figura 2 sería* $Recall = \frac{26}{26+17} = 0.60$

A mayor *recall,* menor cantidad de falsos positivos.

Precision: indica cuál es la fracción de predicciones positivas correctas.

$Precision = = \frac{TP}{TP+FP} = = \frac{26}{26+7} = 0.79$

False Positive Rate (FPR): indica qué fracción, de todas las instancias negativas, identificó el clasificador erróneamente como positivas También se conoce como especificidad.

$FPR = \frac{FP}{TN+FP} = \frac{7}{400+7} = 0.02$

Existe un *trade off* entre la *precision* y el recall. Algunas tareas de *machine learning* orientadas a *recall* son*:*

- Búsqueda y extracción de información en procesos legales

- Detección de tumores

Algunas de las tareas de *machine learning* orientadas a la precision son:

- *Search engine ranking*, sugerencias de consultas
- Clasificación de documentos

F1– score: El **F1–score** es una medida de evaluación que combina dos aspectos importantes de un modelo de clasificación: la **precision (precisión)** y el **recall (recuperación)**. Se utiliza especialmente cuando las clases en un problema de clasificación están desbalanceadas; es decir, cuando una clase es mucho más común que la otra.

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} = \frac{2 \cdot TP}{2 \cdot TP + FN + FP}$$

Figura 3. F1 Score

Classifier Decision Functions

Las **funciones de decisión en clasificadores** son componentes fundamentales de los modelos de clasificación en *machine learning*. Estas funciones determinan cómo asigna un clasificador una etiqueta a una instancia de datos, en función de las características de entrada. En términos sencillos, la función de decisión decide a qué clase o categoría pertenece una observación.

Existen diferentes tipos de clasificadores, y la forma en que la función de decisión actúa varía según el modelo. Por ejemplo, en un clasificador lineal, como la **regresión logística**, la función de decisión se basa en una combinación lineal de las características de entrada; mientras que en un clasificador basado en árboles de decisión, la función de decisión sigue reglas jerárquicas, que dividen el espacio de características en regiones de clase.

En modelos como la **máquina de soporte vectorial (SVM)**, la función de decisión se define a través de un hiperplano, que separa las clases. En este caso, el modelo asigna una clase a una instancia, según en qué lado del hiperplano se encuentra. La **distancia al margen** (o margen máximo) entre las instancias y el hiperplano también juega un papel crucial en la clasificación (Cortes & Vapnik, 1995).

En el contexto de modelos probabilísticos, como **Naïve Bayes**, la función de decisión no solo asigna una clase, sino que calcula la probabilidad de que una instancia pertenezca a cada clase; y elige la clase con la probabilidad más alta (Murphy, 2012). Esta probabilidad es crucial para tareas como la clasificación de textos o el análisis de sentimientos.

Finalmente, el rendimiento de la función de decisión puede evaluarse utilizando métricas como la **precision**, **recall** y el **F1– score**, que reflejan cuán bien está tomando decisiones el modelo, en relación con las clases positivas y negativas (Sokolova & Lapalme, 2009).

Curvas Precision-Recall

La curva Precision-Recall es una **representación gráfica** de cómo la **precisión** y el **recall cambian,** según se ajustan los umbrales de decisión del modelo. A medida que se disminuye el umbral, el modelo tiende a clasificar más instancias como positivas, lo que generalmente aumenta el *recall* pero puede reducir la precisión; ya que también aumenta el número de falsos positivos (Sokolova & Lapalme, 2009). Por el contrario, al aumentar el umbral, se puede lograr una mayor precisión, pero el *recall* disminuirá.

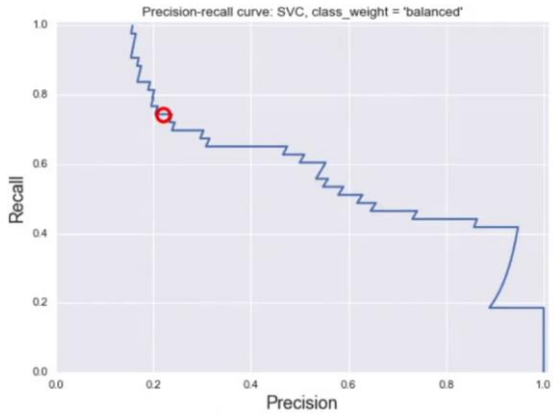


Figura 4. Curvas de Precision-Recall.

La curva se traza en un gráfico donde el eje X representa el **recall** y el eje Y la **precision**, mostrando cómo estas dos métricas interactúan en función de los umbrales seleccionados. Esta herramienta es particularmente útil cuando se enfrentan a problemas con **clases desbalanceadas**, ya que la **precisión global (*accuracy*) podría ser engañosa al no reflejar correctamente el desempeño** sobre la clase minoritaria (Powers, 2011).

Importancia de la curva:

- **Curva ideal**: en la mejor situación posible, la curva se acerca a la esquina superior derecha, donde tanto la precisión como el *recall* son máximos (Hand & Till, 2001).
- **Curva aleatoria**: en el caso de un modelo sin poder predictivo, la curva se acercará a una línea diagonal desde el origen hasta el extremo inferior derecho, indicando que el modelo tiene un rendimiento aleatorio.

La Figura 4 muestra un ejemplo de *Precision- Recall Curves*.

Curvas ROC

Las curvas **ROC** (Receiver Operating Characteristic) son herramientas fundamentales para evaluar el **desempeño** de los **modelos** de **clasificación**, especialmente en tareas de **clasificación binaria**. Esta técnica permite visualizar cómo varían las tasas de **verdaderos** y **falsos positivos,** según el umbral de decisión del clasificador, brindando una visión general de la capacidad del modelo para discriminar entre clases (Fawcett, 2006).

A continuación, se describe en detalle qué es una curva ROC, cómo se interpreta y su relevancia en el análisis de modelos predictivos.

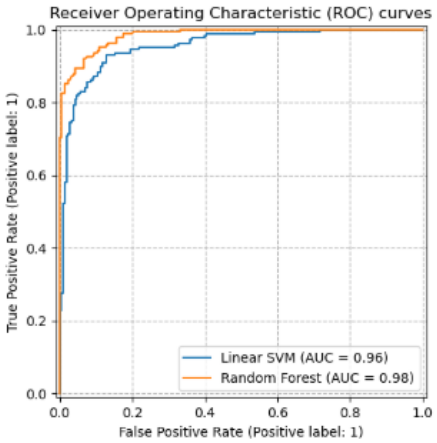


Figura 5. Curva ROC - Linear SVM y Random Forest.

Definición de Curvas ROC

La curva **ROC** es una representación gráfica que muestra el rendimiento de un clasificador binario a través de dos medidas principales: la **tasa de verdaderos positivos** (TPR, por sus siglas en inglés) y la **tasa de falsos positivos** (FPR, por sus siglas en inglés) (Fawcett, 2006). El objetivo principal de esta curva es ilustrar la capacidad de un modelo para clasificar correctamente las instancias positivas, mientras minimiza las instancias negativas clasificadas incorrectamente.

La **tasa de verdaderos positivos** (TPR), también conocida como **recall**, mide la proporción de positivos verdaderos correctamente identificados por el clasificador. Se calcula como:

$$\text{TPR} = \frac{TP}{TP + FN}$$

- **TP** es el número de verdaderos positivos (instancias positivas correctamente clasificadas).
- **FN** es el número de falsos negativos (instancias positivas incorrectamente clasificadas como negativas) (Fawcett, 2006).

Por otro lado, la **tasa de falsos positivos** (FPR) mide la proporción de negativos reales que son erróneamente clasificados como positivos. Su fórmula es:

$$\text{FPR} = \frac{FP}{FP + TN}$$

- **FP** es el número de falsos positivos (instancias negativas clasificadas incorrectamente como positivas).
- **TN** es el número de verdaderos negativos (instancias negativas correctamente clasificadas como negativas) (Fawcett, 2006).

En una curva ROC, el eje **horizontal representa la FPR**, mientras que el eje vertical muestra la **TPR**. La curva se traza variando el umbral de decisión del clasificador, lo que permite observar cómo se modifican estas tasas a medida que cambia el punto de corte para clasificar una instancia como positiva o negativa. Cuanto más alejada esté la curva ROC del eje diagonal (que representa un modelo aleatorio), mejor será el rendimiento del clasificador (Fawcett, 2006).

El área bajo la curva ROC (AUC, por sus siglas en inglés) es una **medida cuantitativa del rendimiento global del modelo**. Un AUC de 1 indica un clasificador perfecto, mientras que un AUC de 0.5 sugiere que el modelo no es mejor que una clasificación aleatoria (Bradley, 1997).

Evaluación multiclase

La evaluación de problemas de clasificación multiclase es una extensión del caso binario. Se construye una matriz con el total de valores verdaderos vs. las predicciones binarias. En este tipo de problemas, la matriz de confusión es ampliamente usada junto con el *classification report,* que muestra una serie de métricas para problemas de clasificación.

Normalmente se calcula el promedio de métricas para todas las clases; sin embargo, hay diferentes formas de promediar los resultados multiclase:

- *Micro – macro average*
- *Support* (número de instancias) por clase; importante especialmente para clases desbalanceadas

La siguiente ilustración muestra un problema de clasificación de diez clases

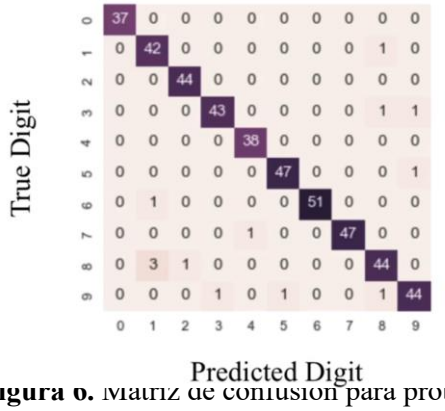


Figura 6. Matriz de confusión para problemas multiclase.

Micro vs. Macro Average:

La siguiente figura muestra el cálculo de *macro-average precision* para un problema de clasificación de 3 clases. Los aspectos importantes para considerar son los siguientes: cada clase tiene el mismo peso, se calcula la métrica por cada una de las clases y, finalmente, se calcula el promedio resultante.

Class	Predicted Class	Correct?
orange	lemon	0
orange	lemon	0
orange	apple	0
orange	orange	1
orange	apple	0
lemon	lemon	1
lemon	apple	0
apple	apple	1
apple	apple	1

Macro-average:

- Each class has equal weight.

1. Compute metric within each class
2. Average resulting metrics across classes

Class	Precision
orange	1/5 = 0.20
lemon	1/2 = 0.50
apple	2/2 = 1.00

Macro-average precision:
(0.20 + 0.50 + 1.00) / 3 = **0.57**

Figura 7. Micro Average Precision.

La siguiente figura muestra el cálculo de *macro-average precision* para un problema de clasificación de 3 clases. El aspecto importante para considerar es el siguiente: cada clase tiene el mismo peso; por lo tanto, la clase más poblada es la más influyente.

Class	Predicted Class	Correct?
orange	lemon	0
orange	lemon	0
orange	apple	0
orange	orange	1
orange	apple	0
lemon	lemon	1
lemon	apple	0
apple	apple	1
apple	apple	0

Micro-average:
- Each instance has equal weight.
- Largest classes have most influence

1. Aggregrate outcomes across all classes
2. Compute metric with aggregate outcomes

Micro-average precision:
4 / 9 = **0.44**

Figura 8. Macro Average Precision.

Algunas **conclusiones**:

- Si las clases tienen más o menos el mismo número de instancias, *macro-micro average* tenderán a ser las mismas.
- Si algunas clases tienen más instancias que otras y se quisiera establecer pesos en función de las **más numerosas,** es recomendable usar *micro average.*
- Si el *micro average* es más bajo que el *macro average,* entonces es recomendable examinar las clases mayoritarias para entender por qué se tiene ese nivel de *performance.*
- Si el *macro average* es más bajo que el *micro average,* entonces es recomenable examinar las clases minoritarias para entender por qué se tiene ese nivel de *performance.*

Evaluación para regresión

r_2 score suele ser suficiente para evaluar modelos de regresión, esta métrica indica cuán buena será la predicción para nuevas instancias. El mejor valor de predicción posible es 1.0, mientras que el valor de 0 indica una predicción con valor constante.

Algunas métricas alternativas son:

- **mae (Mean Absolute Error):** diferencia absoluta entre el *target* y el valor de predicción
- **mse (Mean Squared Error):** diferencia al cuadrado entre el *target* y el valor de predicción
- **Median Absolute Error:** robusta a *outliers; d*ado que usa la mediana de la distribución de los errores en lugar de la media.

También suele ser útil ocupar los *dummy regressors,* al igual que en problemas de clasificación, que comparan la clasificación de los modelos contra una predicción *dummy. Scikit - Learn* tiene una implementación para esto. Se puede usar cualquiera de las siguientes estrategias:

mean	Predice el promedio de los valores de *target* con los que se entrenó.
median	Predice la mediana de los valores de *target* con los que se entrenó.
quantile	Predice el cuantil especificado por el usuario.
constant	Predice un valor constante especificado por el usuario.

Figura 9. Estrategias para usar *dummy regressors.*

Selección de modelos: Optimización de clasificadores para diferentes métricas de evaluación

Train/Test con los mismos datos	**Separación Train/Test**	**K-fold cross validation**
Una única métrica	Una única métrica	K separaciones *train-test*
Usualmente presenta *overfitting* y no generaliza bien con nuevos datos.	Rápida y simple	Se tiene una métrica promedio sobre todos los *splits* (separaciones)
Puede ser útil para hacer un *sanity check.* Bajo *accuracy* en el conjunto de entrenamiento podría indicar problemas en la implementación	Falta de información sobre la varianza	Puede ser combinada con un *gridSearch*

Figura 10. Selección de modelos usando métricas de evaluación.

Usar únicamente *cross-validation* o *test set* para hacer selección de modelos podría llegar a ocasionar un sutil *overfitting* o una generalización altamente optimista. En lugar de eso, se recomienda hacer la siguiente separación de conjuntos de datos:

- *Training set* (para la construcción del modelo)
- *Validation set* (para seleccionar el modelo)
- *Test set* (para una evaluación final)

En la práctica, sería crear los *dataset* de *train*, *validation* y *test.* Hacer validación cruzada en los datos de entrenamiento para la selección del modelo, hacer la evaluación final del modelo con el conjunto de datos de *test.*

Referencias citadas en la Clase 5

Draper, N. R., & Smith, H. (1998). *Applied Regression Analysis* (3rd ed.). Wiley.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters, 27*(8), 861-874.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning* (2nd ed.). Springer.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.

Niculescu-Mizil, A., & Caruana, R. (2005). Predicting Good Probabilities with Supervised Learning. *Proceedings of the 22nd International Conference on Machine Learning (ICML-05)*, 625–632.

Provost, F., & Fawcett, T. (1997). Analysis and visualization of classifier performance: Comparison under imprecise class distributions. *KDD*, 43–48.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management, 45*(4), 427-437.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273-297.

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition, 30*(7), 1145-1159.

Powers, D. M. W. (2011). Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies, 2*(1), 37-63.

Definición de los términos citados en la Clase 5

***Precision*.**
La **precision (precisión)** de un modelo de clasificación en *machine learning* es una métrica que indica cuántas de las predicciones positivas realizadas por el modelo son correctas. Se calcula como la proporción de **verdaderos positivos** (instancias correctamente clasificadas como positivas), sobre el total de instancias clasificadas como positivas (la suma de verdaderos positivos y falsos positivos).

***Recall*.**
El **recall** de un modelo de clasificación en *machine learning* es una métrica que mide la capacidad del modelo para identificar correctamente todas las **instancias positivas reales**. En otras palabras, indica qué **porcentaje** de las **instancias positivas** que realmente existen en los datos fueron correctamente clasificadas como positivas por el modelo.

Profundización Clase 5
Trade off* entre *Precision* y *Recall

Umbral de Decisión	***Precision (Precisión)***	***Recall***
0.1	*Baja*	*Alta*
0.2	*Moderada*	*Alta*
0.3	*Moderada*	*Alta*
0.4	*Alta*	*Media*
0.5	*Alta*	*Baja*
0.6	*Muy alta*	*Baja*
0.7	*Muy alta*	*Muy baja*

0.8	*Muy alta*	*Muy baja*
0.9	*Muy alta*	*Extremadamente baja*

- **Umbral bajo (cerca de 0.1).** Cuando el umbral de decisión es bajo, el modelo clasifica muchas instancias como positivas, lo que **aumenta el recall** (más positivos detectados). Sin embargo, esto puede **disminuir la precisión,** porque el modelo también clasifica erróneamente más instancias como positivas (más falsos positivos).
- **Umbral alto (cerca de 0.9).** Cuando el umbral es alto, el modelo es más selectivo y clasifica menos instancias como positivas, lo que **reduce el recall** (menos positivos detectados). Pero esto **aumenta la precisión,** porque las instancias clasificadas como positivas tienen mayor probabilidad de ser correctas (menos falsos positivos).

Este **trade-off** muestra que no es posible maximizar ambas métricas simultáneamente. El ajuste del umbral depende de la importancia que se le dé a la **precision** o al **recall,** según el contexto del problema (por ejemplo, en detección de fraudes puede ser más importante el recall).

Cross validation

A continuación, se muestra un ejemplo de implementación de *cross validation* sobre el *dataset load_digits* de *sklearn* usando diferentes métricas de performance

```
# Cargamos las librerías
from sklearn.model_selection import cross_val_score
from sklearn.datasets import load_digits
from sklearn.svm import SVC

# Cargamos el dataset digits
dataset = load_digits()

# Convertimos el problema multiclase a uno binario
X, y = dataset.data, dataset.target == 1
# Definimos el clasificador
clf = SVC(kernel='linear', C=1)

# accuracy
print('Cross-validation (accuracy)', cross_val_score(clf, X, y, cv=5))
# AUC
print('Cross-validation (AUC)', cross_val_score(clf, X, y, cv=5, scoring = 'roc_auc'))
# recall
print('Cross-validation (recall)', cross_val_score(clf, X, y, cv=5, scoring = 'recall'))
```

```
Cross-validation (accuracy) [ 0.91944444  0.98611111  0.97214485  0.97493036  0.96935933]
Cross-validation (AUC) [ 0.9641871   0.9976571   0.99372205  0.99699002  0.98675611]
Cross-validation (recall) [ 0.81081081  0.89189189  0.83333333  0.83333333  0.83333333]
```

PUCE Virtual
EDUCACIÓN FLEXIBLE

La excelencia no se improvisa

síguenos

MDASC - 12 - 241