Entendiendo los problemas de clasificación desde un marco teórico

Clase de Machine Learning aplicado a temas ambientales
Universidad Nacional de Colombia
2024

Profesor: Carlos Daniel Jimenez

Contenido de la clase

Agenda

- Introducción a la clasificación binaria
- Métricas de performance de los modelos
- Problemas de Múltiples clasificaciones
- Análisis del error

Bibliografía y web-grafía recomendada

- Curvas de aprendizaje
- Clasificación el contexto de Machine Learning
- Clasificando Iris

Clasificación binaría

Clasificación Binaria

- Consiste en evaluar dos alternativas a clasificar
- En contexto estadístico esto quiere decir que son problemas del orden bonimial
- Algunos modelos recomendados para trabajar con este tipo de problemas son :
  - Random Forest
  - SGD
  - SVM

Métricas de performance de los modelos

Métricas de performance de los modelos

- Primero a resaltar para trabajar con este tipo de problemas , al igual que el de regresión es : mezclar los datos antes de hacer el train and test split

- La primer forma de evaluar estos modelos tiene que ver con la implementación del cross validation

  Cross Validation: Es una forma de asegurarse de que el modelo que estamos entrenando no solo funciona bien con los datos con los que se entrenó, sino que también puede generalizarse bien a datos nuevos y desconocidos

Cross Validation

- **Dividir los Datos**: se divide el conjunto de datos en varias partes más pequeñas, llamadas "folds". Por ejemplo, si tenemos 100 datos y decidimos hacer una validación cruzada de 5 pliegues (5-fold cross-validation), dividiremos los datos en 5 partes de 20 datos cada una.

- **Entrenar y Evaluar**: Se toma una de esas partes como conjunto de prueba y las otras 4 partes como conjunto de entrenamiento. Se entrena el modelo con las 4 partes y se prueba con la parte restante. Este proceso se repite varias veces, cambiando cada vez la parte que se usa como prueba y las partes que se usan como entrenamiento.

- **Promediar Resultados**: Al final, se calculan las métricas de evaluación (como la precisión, el error, etc.) para cada una de las veces que se entrenó y probó el modelo, y se hace un promedio de estos resultados. Este promedio da una mejor idea de cuán bien funcionará el modelo con datos nuevos.

Métricas de performance de los modelos

- A veces el cross validation (en especial para el caso de las ciencias naturales) es necesario darle control sobre las muestras a nivel de representación y por ello se usa la API de StratifiedKfolds
- Esto suele ser porque hay algunas clases de muestras que no estan bien representadas
- Con la implementación de lo anterior se evita el sesgo a la hora de entrenar un modelo , a la par de evaluarlo

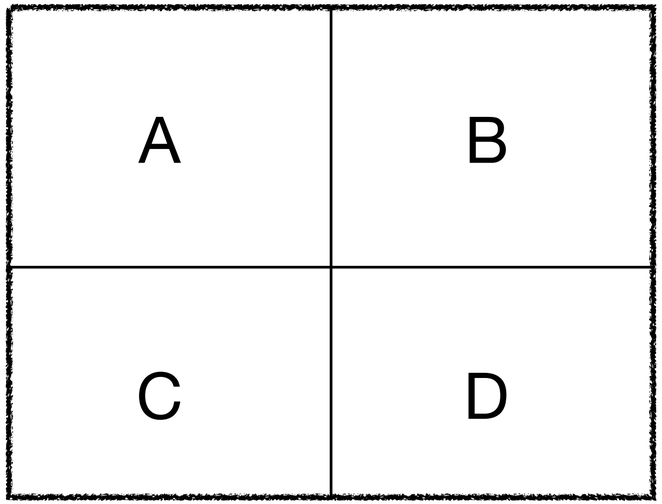
```
for train_index, test_index in skfolds.split(X_train, y_train):
    clone_clf = clone(sgd_clf)
    X_train_folds = X_train[train_index]
    y_train_folds = y_train[train_index]
    X_test_fold = X_train[test_index]
    y_test_fold = y_train[test_index]

    clone_clf.fit(X_train_folds, y_train_folds)
    y_pred = clone_clf.predict(X_test_fold)
    n_correct = sum(y_pred == y_test_fold)
    print(n_correct / len(y_pred))
```

Métricas de performance de los modelos

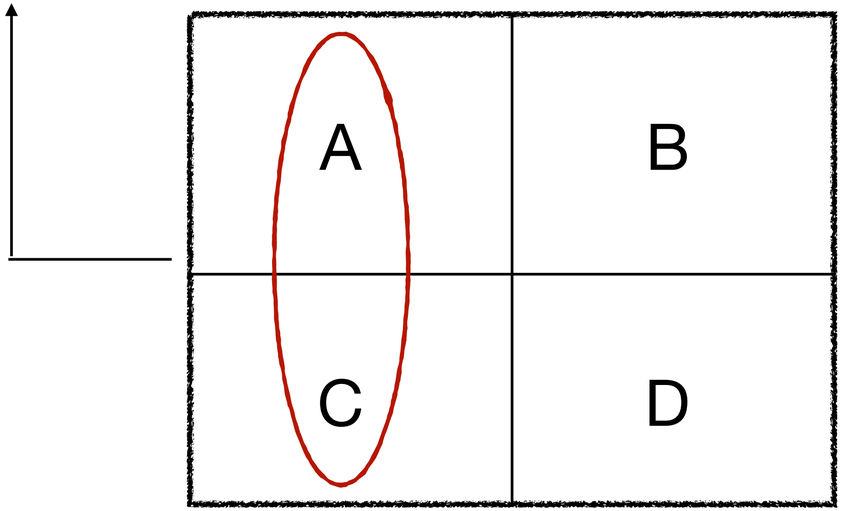
La matriz de confusión tiende a ser en términos prácticos el elementos que más información aporta a la hora de entender el comportamiento de un modelo.

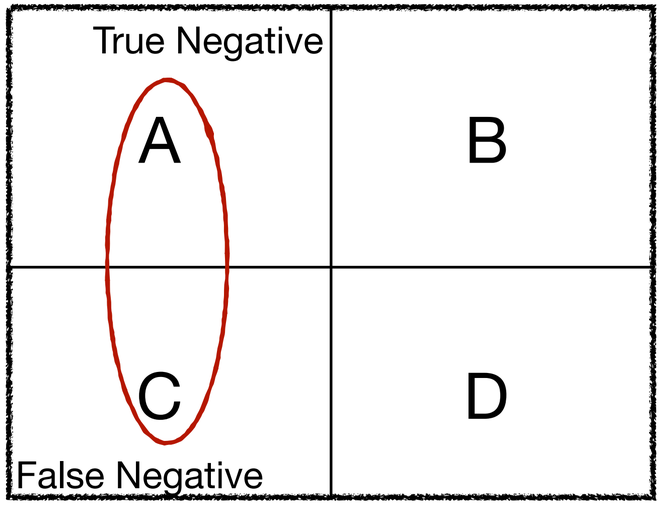
Esta trata de contar el número de veces que una instancia es clasificada en todas las categorías existentes

Para poder trabajar con está métrica es necesario tener las predicciones del modelo y enfrentarlas a los targets reales que tenemos almacenados en el y_test

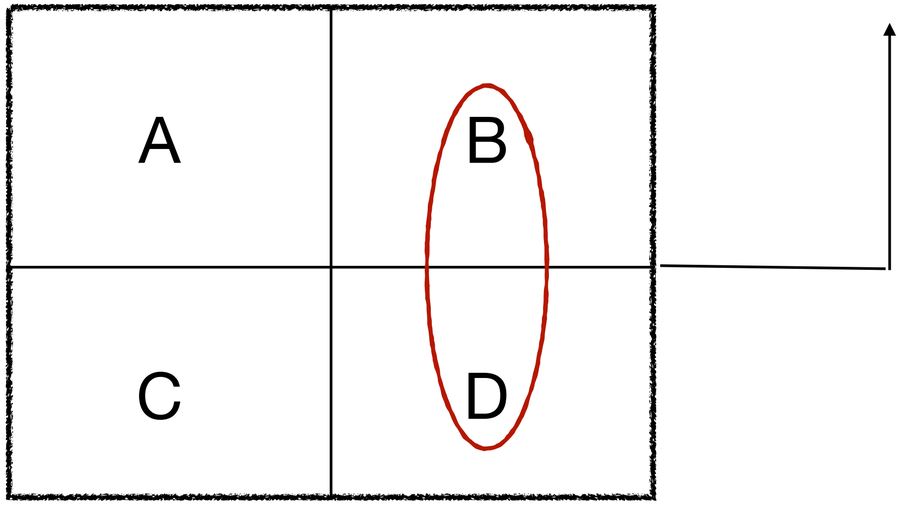

A
B
C
D

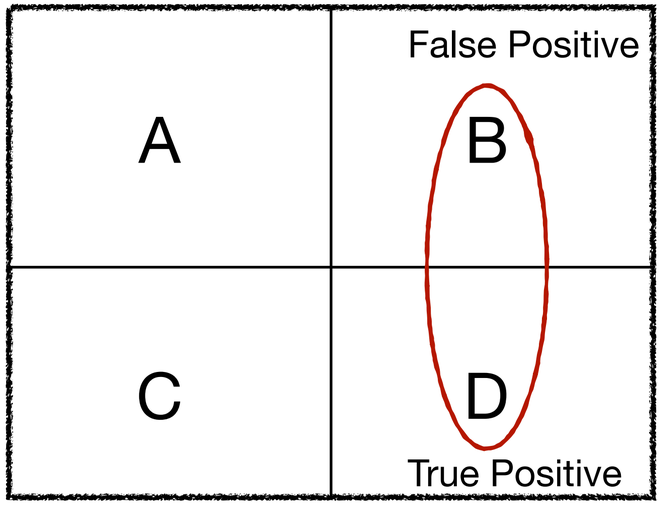
De esta lado siempre
están los negativos




True Negative
A
B
C
D
False Negative

De esta lado siempre
están los Positivos




A
False Positive
B
C
D
True Positive

Entendiendo la matriz de confusión

- True Negative: Correctamente clasificadas que no corresponden a un target especifico
- False Positive: Clasificadas mal, marcadas como el target objetivo
- False Negative: Clasificadas erróneamente que no corresponde al target especifico , cuando si corresponden
- True Positive : Clasificadas correctamente cuando corresponden al target objetivo

Información de la matriz de confusión

- Precision : Mide cuántas de las predicciones positivas del modelo son realmente correctas.

$$precision = \frac{TruePositive}{(TruePositive + FalsePositive)}$$

- Recall: Mide cuántos de los verdaderos positivos han sido capturados por el modelo

$$Recall = \frac{TruePositive}{(TruePositive + FalseNegative)}$$

Información de la matriz de confusión

- F1-score : Métrica que combina la precisión y el recall en una sola medida. Es especialmente útil cuando tienes un balance entre precisión y recall.

$$f1 = \frac{TruePositive}{TruePositive + \frac{(FalseNegative + FalsePositive)}{2}}$$

- Hay una característica importante a resaltar : Esta ecuación presenta un promedio armónico, lo cual resalta y eleva la representación de los valores bajos.

Trade Off Recall y Precisión

- Cuando trabajamos con modelos de clasificación, a menudo enfrentamos un dilema entre precisión y recall. Mejorar una métrica puede empeorar la otra, y viceversa. Este es el trade-off entre precisión y recall.

  Precisión Alta:
  i) Cuando la precisión es alta, significa que de todas las predicciones positivas, la mayoría son correctas.
  ii) Sin embargo, esto puede significar que el modelo es muy estricto al etiquetar algo como positivo, y podría estar perdiendo muchos verdaderos positivos, reduciendo el recall.

Recall Alto:
i) Cuando el recall es alto, significa que el modelo está capturando la mayoría de los verdaderos positivos.
ii) Sin embargo, esto puede significar que el modelo está siendo muy generoso al etiquetar algo como positivo, lo que puede aumentar el número de falsos positivos y reducir la precisión.

Para evitar que el modelo "pierda memoria" (es decir, para evitar que el modelo sea incapaz de generalizar bien), debemos buscar un equilibrio entre precisión y recall. Este equilibrio se puede lograr utilizando la métrica del **F1 Score**

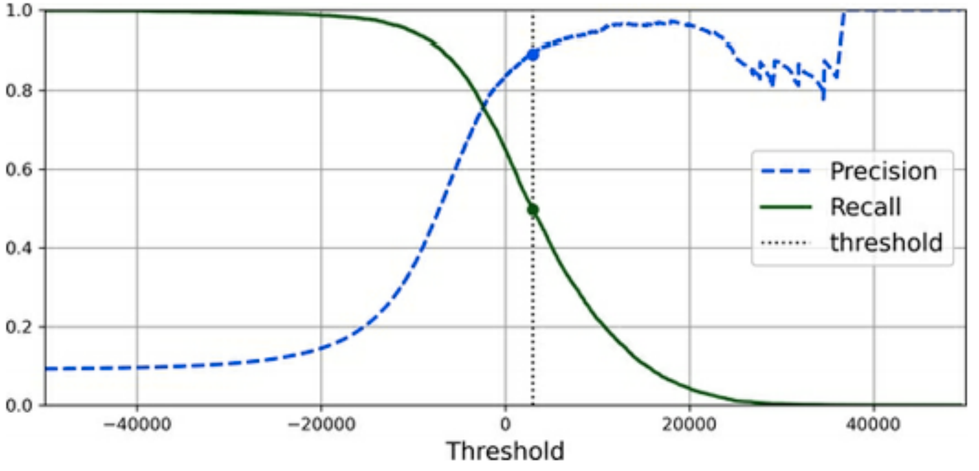
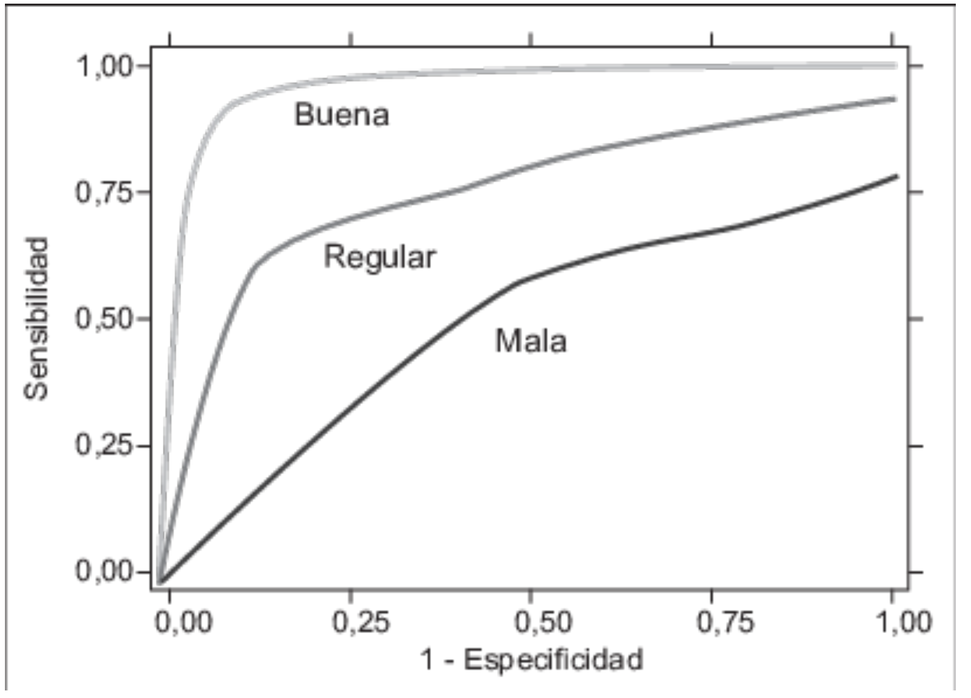


Figure 3-5. Precision and recall versus the decision threshold

Tomado de: Géron, Aurélien. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (p. 198). O'Reilly Media. Edición de Kindle.

Curva ROC

- La **Curva ROC (Receiver Operating Characteristic)** es una herramienta utilizada para evaluar el rendimiento de un modelo de clasificación binaria. Muestra la relación entre la **Tasa de Verdaderos Positivos** (TPR, también llamada Sensibilidad o Recall) y la **Tasa de Falsos Positivos** (FPR).
- La curva se traza al calcular el TPR y el FPR para varios umbrales de decisión diferentes.
- Un modelo perfecto tendría una curva que va directamente hacia la esquina superior izquierda y luego sigue la línea superior del gráfico.
- Un modelo aleatorio tendría una curva que sigue la diagonal (línea de no-discriminación).



Tomado de :https://www.researchgate.net/figure/Figura-5-Esquema-explicativo-de-distintas-posibilidades-de-curvas-ROC_fig5_224954042

Problemas de Múltiples clasificaciones

Clasificación Multiclase

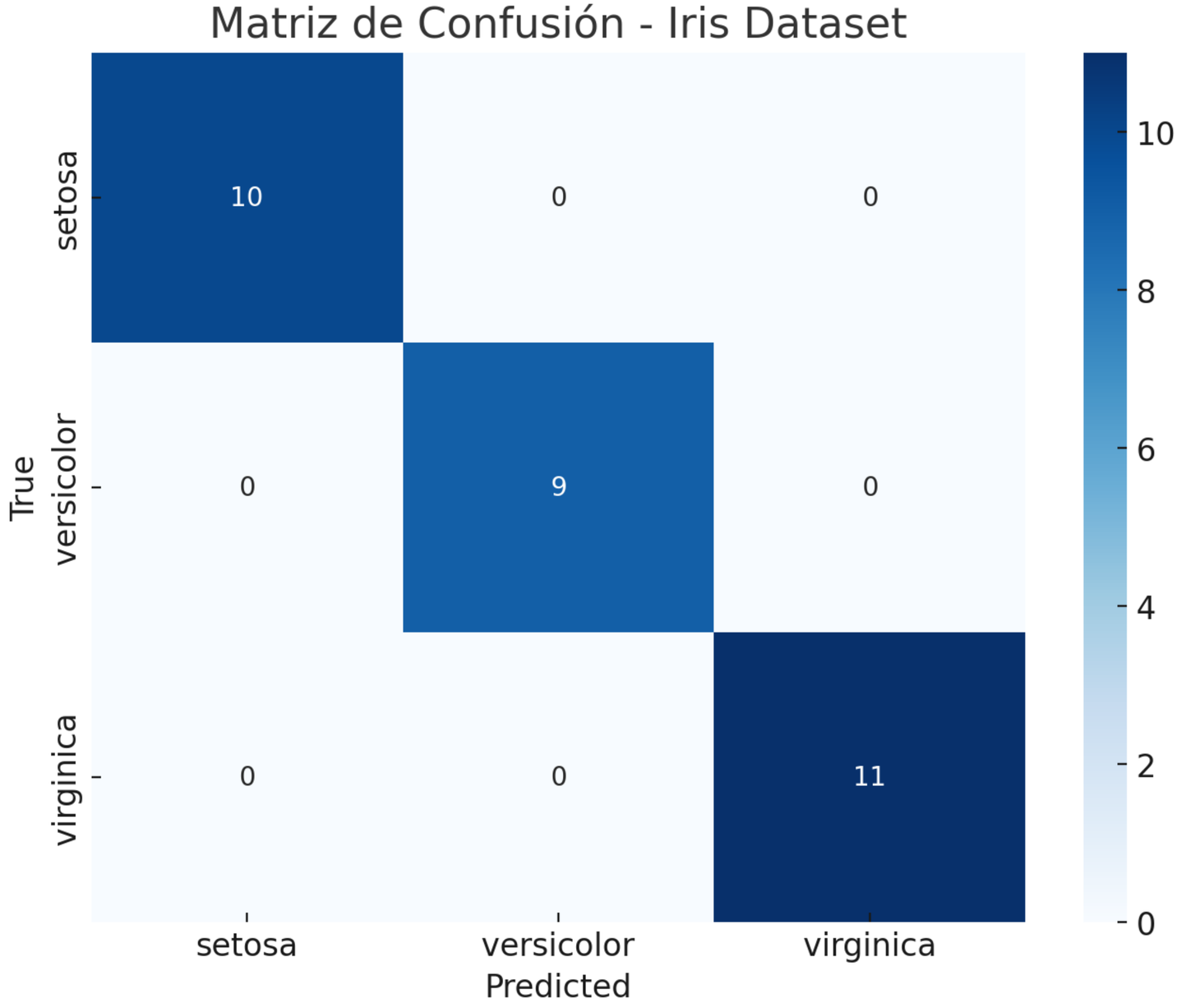
- Es un tipo de problema de clasificación en machine learning donde hay más de dos clases o categorías posibles para cada instancia.
- Estos problemas tienen un tratamiento especial en las métricas de desempeño, por lo que el componente visual será el más importante de analizar.

Métricas de Performance en Clasificación Multiclase

- **Precisión por Clase**: Mide la precisión de las predicciones para cada clase individualmente.
- **Recall por Clase**: Mide el recall de las predicciones para cada clase individualmente.
- **F1 Score por Clase**: Combina la precisión y el recall para cada clase.
- **Accuracy General**: Mide la proporción de todas las predicciones correctas.

Selección de la métrica indicada

- **F1 Score**: Es útil cuando las clases están desbalanceadas, ya que considera tanto la precisión como el recall.
- **Accuracy**: Es útil cuando todas las clases están balanceadas.
- **Matriz de Confusión**: Proporciona una visión detallada de las predicciones correctas e incorrectas por clase.


Matriz de Confusión - Iris Dataset
True
setosa
versicolor
virginica
10
0
0
0
9
0
0
0
11
setosa
versicolor
virginica
Predicted
10
8
6
4
2
0

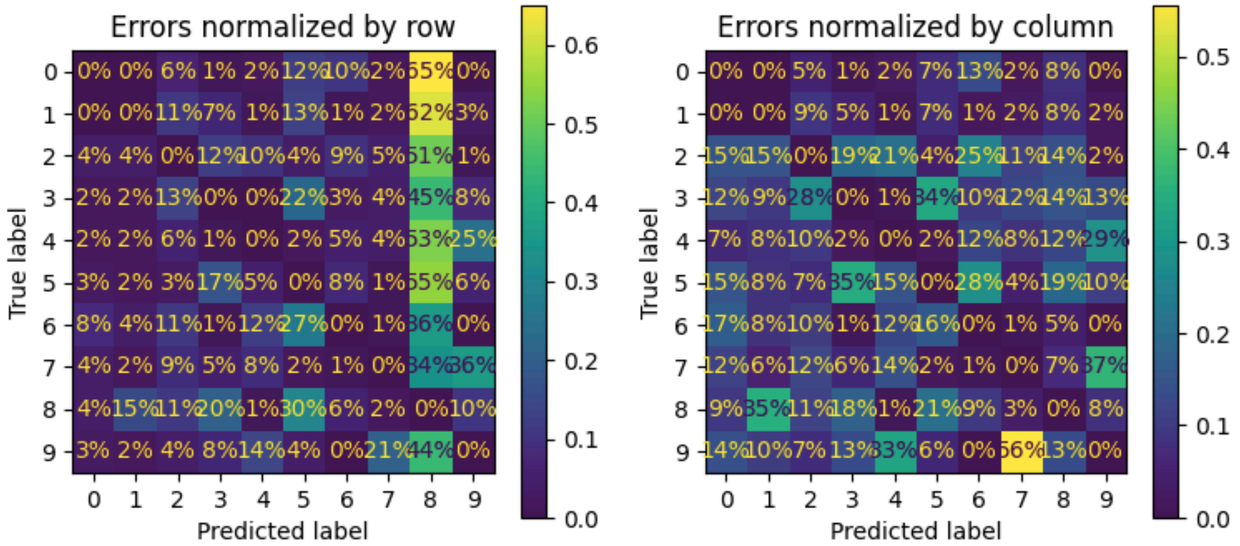
Reporte de métricas

	precision	recall	f1-score	support	
setosa	**1.00**	1.0	0 1.00		10
versicolor	**1.00**	1.0	0 1.00		9
virginica	**1.00**	1.0	0 1.00		11
accuracy			1.00		30
macro avg	**1.00**	1.0	0 1.00		30
weighted avg	**1.00**	1.0	0 1.00		30
Accuracy: 1.00					

Análisis del error

Análisis del error

- En este punto es importante analizar la matriz de confusión y entender el comportamiento de las diagonales
- Para entender los errores hay que normalizar los valores de la matriz, esto quiere decir que hay que dividir cada valor de la clasificación por el total de las imágenes de cada clase
- Un punto importante para abstraer la lógica de los errores es lograr visualizar en la matriz de confusión los valores que no acertaron


Errors normalized by row
True label
Predicted label
0 1 2 3 4 5 6 7 8 9
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Errors normalized by column
True label
Predicted label
0 1 2 3 4 5 6 7 8 9
0.5
0.4
0.3
0.2
0.1
0.0