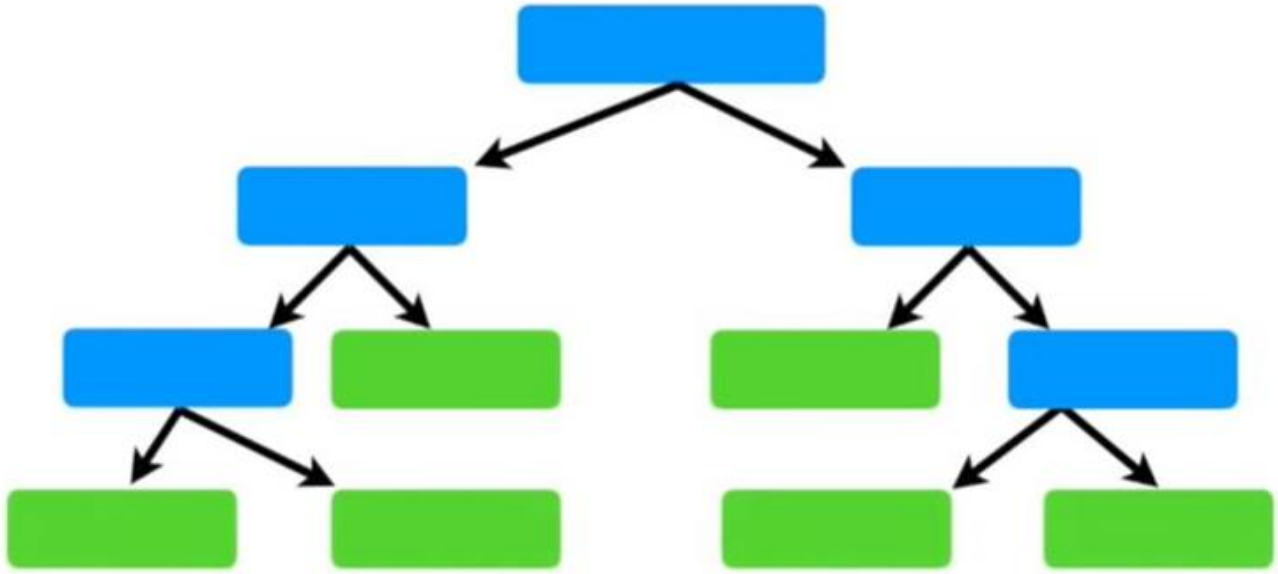
Introducción a la Ciencia de Datos

Árboles de Decisión

Nos clasifica, en forma de árbol, qué caminos o patrones sigue la data para que se pueda predecir cierta categoría (o cierto número)



Los elementos son:

- Root
- Nodes
- Leaf

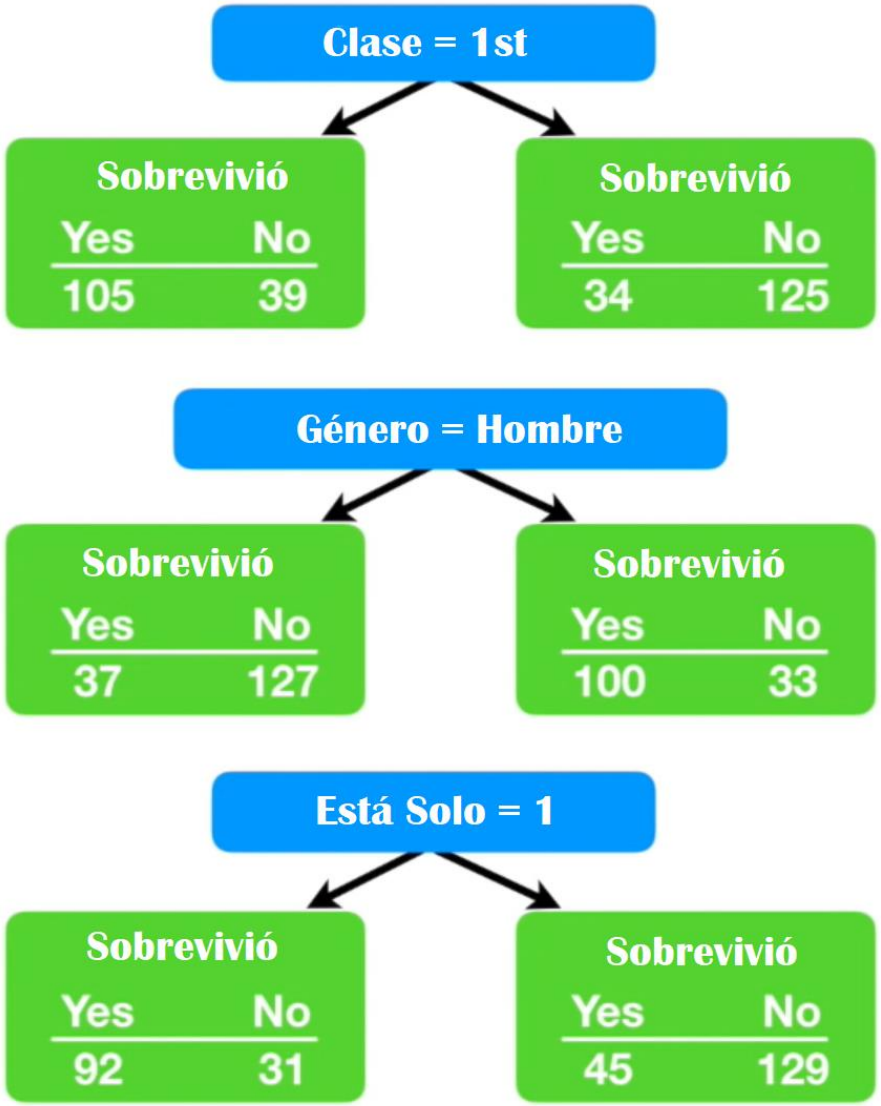
El criterio más importante es el grado de impureza (**Gini):** Determina qué tan bien alguna variable (columna) separar el target.

El gini se calcula: **1 - (probabilidad|1)^2 - (probabilidad|0)^2**

Arboles de Decisión:Un proceso iterativo

Nuestro árbol va a iterar por sobre cada columna y **calcular el Gini** en base a los diferentes valores que tengamos:

- Gini (Clase=1): 0.364
- Gini (G=Hombre):0.360
- Gini(EstáSolo=1):0.381



Arboles de Decisión:Un proceso iterativo

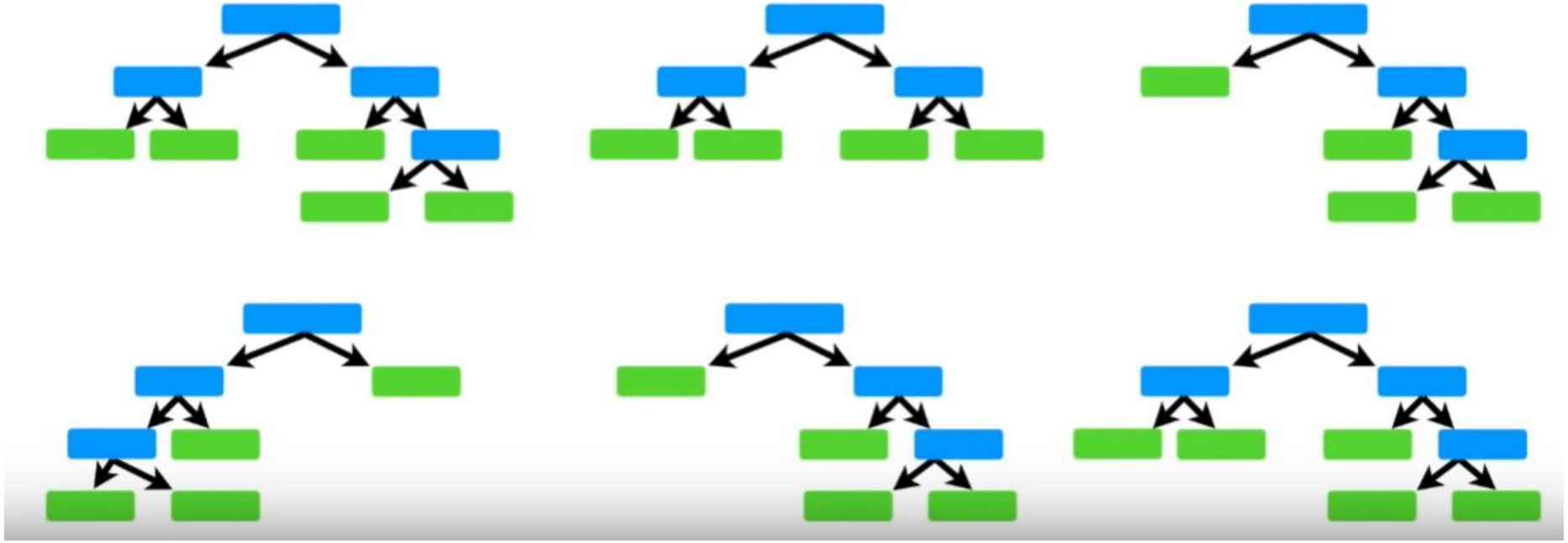
Básicamente, la dinámica del árbol se divide en estos pasos:

1.- Se calculan **todos los ginis de las variables predictoras** y se elige la variable que tenga **menor Gini como root** (o como nodo a partir de la segunda iteración).

2.- Si el **nodo padre tiene menor impureza que los nodos hijos**, ese nodo padre “aborta” a sus hijos y se convierte en un **leaf,** es decir, este padre **ya no tendrá hijos.**

3.-Si el **nodo padre tiene mayor impureza**, entonces se queda y se vuelve a iterar con respecto a todas las variables restantes.

Random Forest



Es un conjunto de árboles entrenados con **diferentes partes de nuestra data**. Estos árboles **votan** si el **target final será 0 u 1,** dependiendo de si hay **más probabilidades** de que sean 0 o 1.

Random Forest

- **Ventaja**
  - Más robusto (evita el overfitting)
  - Nos dice mejor qué variables son más significativas para predecir el target (clase)

- **Desventaja**
  - No se puede seguir la lógica como en un árbol de decisión
  - El proceso de entrenamiento puede demorar mucho, lo que implica un mayor costo computacional

Matriz de confusión y sus métricas

$$recall = \frac{TP}{TP + FN}$$

		predicción	
		0	1
realidad	0	TN	FP
	1	FN	TP

Lesson Bonus_1: Matriz de Confusión

Una matriz de confusión o también conocida matriz de error, es una tabla que sirve para evaluar el rendimiento de un modelo de clasificación. El número de predicciones correctas e incorrectas se resumen con los valores de conteo y se desglosan por cada clase.

TP = True Positive (Verdadero Positivo)
TN = True Negative (Verdadero Negativo)
FP = False Positive (Falso Positivo)
FN = Falso Negative (Falso Negativo)

		predicción	
		0	**1**
realidad	**0**	TN	FP
	1	FN	TP

2x2

Falso Positivo: predijo que era positivo cuando en realidad era negativo.

Falso Negativo: predijo que era negativo cuando en realidad si era positivo.

---

https://www.datasource.ai/es/data-science-articles/comprension-de-la-matriz-de-confusion-y-como-implementarla-en-python

Lesson Bonus_2: Métricas de la Matriz de Confusión - Accuracy

Entre las principales métricas de una Matriz de confusión, podemos tener las siguientes:

Accuracy (exactitud)

Es la relación entre las predicciones correctas y las predicciones totales. Por lo tanto, es el cociente entre los casos bien clasificados por el modelo, y la suma de todos los casos.

Note: Cuando el dataset está desequilibrado, no es una métrica útil. Se recomienda utilizar la métrica **f1_score**

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

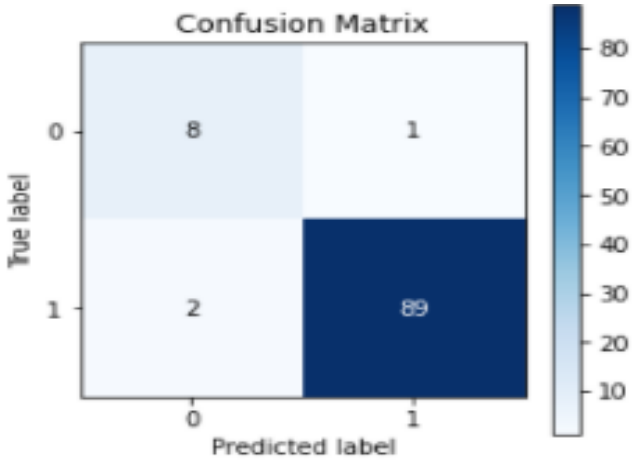
TP = True Positive (Verdadero Positivo)
TN = True Negative (Verdadero Negativo)
FP = False Positive (Falso Positivo)
FN = Falso Negative (Falso Negativo)

Lesson Bonus_2: Métricas de la Matriz de Confusión - Accuracy

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

TP = True Positive (Verdadero Positivo)
TN = True Negative (Verdadero Negativo)
FP = False Positive (Falso Positivo)
FN = Falso Negative (Falso Negativo)

Verdaderos Negativos	Falsos Positivos
Falsos Negativos	Verdaderos Positivos



Accuracy = (89+8)/(89+1+8+2) = (97/100) = 0.97 = 97%

```
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(val_y, y_pred1)
accuracy
```

Métricas de la Matriz de Confusión - Recall

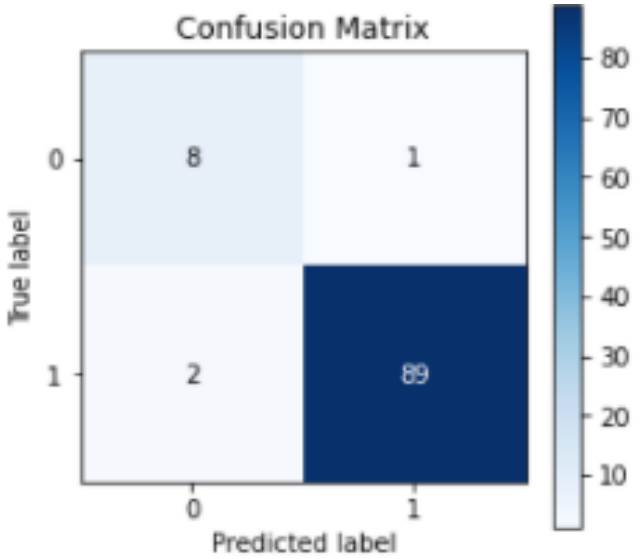
Recall (sensibilidad)

Recall representa la tasa de verdaderos positivos. Es decir, es la proporción entre los casos positivos bien clasificados por el modelo, respecto al total de positivos.

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

Verdaderos Negativos	Falsos Positivos
Falsos Negativos	Verdaderos Positivos

TP = True Positive (Verdadero Positivo)
TN = True Negative (Verdadero Negativo)
FP = False Positive (Falso Positivo)
FN = Falso Negative (Falso Negativo)



Recall = (89)/(89+2) = 89/91 = 0.978
= 97.8%

```
from sklearn.metrics import recall_score
recall = recall_score(val_y, y_pred1)
recall
```

Métricas de la Matriz de Confusión - Precision

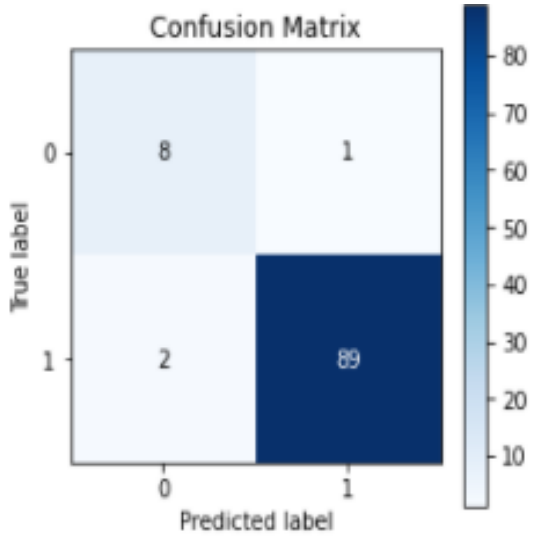
Precision (Precisión)

La precisión también se conoce como valor predictivo positivo. Es la proporción de instancias relevantes entre las instancias recuperadas.

$$Precision = \frac{TP}{TP + FP}$$

Verdaderos Negativos	Falsos Positivos
Falsos Negativos	Verdaderos Positivos

TP = True Positive (Verdadero Positivo)
TN = True Negative (Verdadero Negativo)
FP = False Positive (Falso Positivo)
FN = Falso Negative (Falso Negativo)



Precision = (89)/(89+1) = 89/90 = 0.98888
= 98.9%

```
from sklearn.metrics import precision_score
precision = precision_score(val_y, y_pred1)
precision
```

Lesson Bonus_2: Métricas de la Matriz de Confusión - Specificity

Specificity (Especificidad)

Specificity representa la tasa de verdaderos negativos. Es decir, es la proporción entre los casos negativos bien clasificados por el modelo, respecto al total de negativos.

$$Specificity = \frac{TN}{TN + FP}$$

TP = True Positive (Verdadero Positivo)
TN = True Negative (Verdadero Negativo)
FP = False Positive (Falso Positivo)
FN = Falso Negative (Falso Negativo)

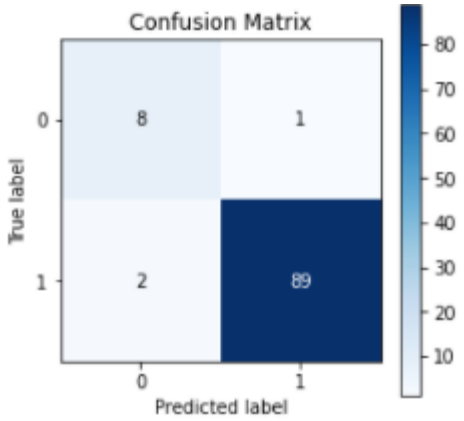
Métricas de la Matriz de Confusión – F1 score

F1 score (Puntuación F1)

F1 score es una métrica muy empleada porque nos resume la **Precision** y **Recall** en una sola métrica. Por ello, es de gran utilidad cuando la distribución de clases es desigual (desequilibrada).

Por ejemplo: en un dataset, tenemos 85% de datos positivos y un 15% de datos negativos, lo que en el campo de salud es bastante común.

$$F1\ score = \frac{2 * (precision * recall)}{precision + recall} = \frac{2TP}{2TP + FP + FN}$$



F1 Score = (2*89)/((2*89) + 1 + 2) = (178/181)
= 0,983425
= 98.3%

Verdaderos Negativos	Falsos Positivos
Falsos Negativos	Verdaderos Positivos

TP = True Positive (Verdadero Positivo)
TN = True Negative (Verdadero Negativo)
FP = False Positive (Falso Positivo)
FN = Falso Negative (Falso Negativo)

```
#Método 1

from sklearn.metrics import f1_score
f1_score = f1_score(val_y, y_pred1)
f1_score
```

```
0.9834254143646408
```

Métricas de la Matriz de Confusión – F1 score

F1 score (Puntuación F1)

```
# Method 1: sklearn
from sklearn.metrics import f1_score
f1_score(y_true, y_pred, average=None)
# Method 2: Manual Calculation
F1 = 2 * (precision * recall) / (precision + recall)
# Method 3: Classification report [BONUS]
from sklearn.metrics import classification_report
print(classification_report(y_true, y_pred, target_names=target_names))
```

Conforme a estas nuevas métricas podemos obtener **cuatro casos posibles para cada clase:**

- **Alta precisión y alto recall:** el modelo de Machine Learning escogido maneja perfectamente esa clase.
- **Alta precisión y bajo recall:** el modelo de Machine Learning escogido no detecta la clase muy bien, pero cuando lo hace es altamente confiable.
- **Baja precisión y alto recall:** El modelo de Machine Learning escogido detecta bien la clase, pero también incluye muestras de la otra clase.
- **Baja precisión y bajo recall:** El modelo de Machine Learning escogido no logra clasificar la clase correctamente.

```
#Método 3

from sklearn.metrics import classification_report
rf_report = classification_report(val_y, y_pred1, target_names=['No es Covid' , 'Es Covid']) #Obtenemos las mátricas de la matriz
print(rf_report)
```

```
              precision    recall  f1-score   support

  No es Covid       0.80      0.89      0.84         9
     Es Covid       0.99      0.98      0.98        91

     accuracy                           0.97       100
    macro avg       0.89      0.93      0.91       100
 weighted avg       0.97      0.97      0.97       100
```

Cuando tenemos un "dataset" con desequilibrio, suele ocurrir que obtenemos un **alto valor de precisión en la clase Mayoritaria y un bajo recall en la clase Minoritaria. En el campo de la salud ésta circunstancia es particularmente frecuente y por ello tenemos que recurrir al balance de clases.**