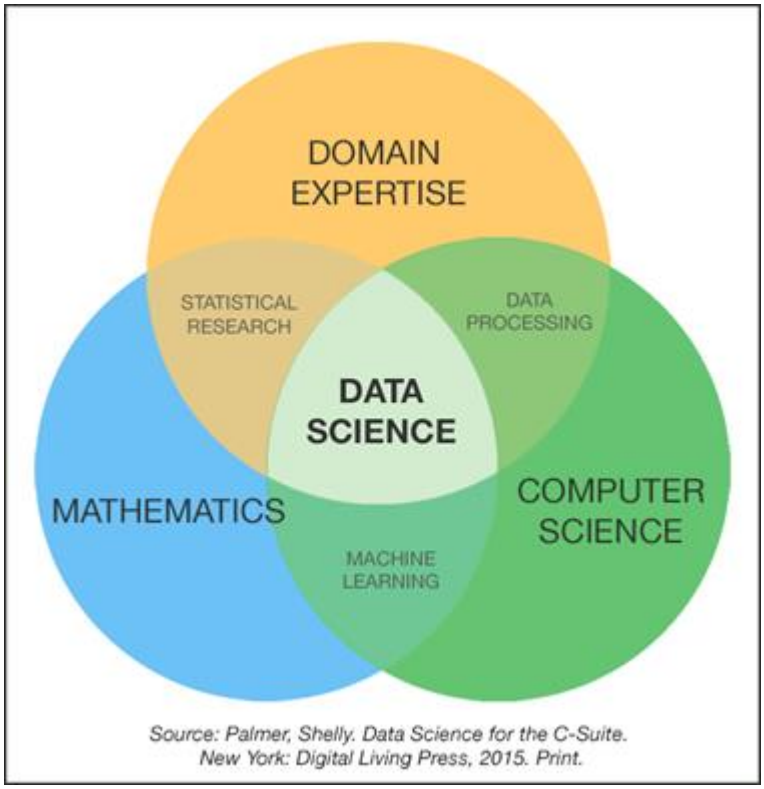
CTIC UNI
CENTRO DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIONES
UNIVERSIDAD NACIONAL DE INGENIERÍA

CIENCIA DE DATOS:

APRENDE LOS FUNDAMENTOS DE MANERA PRÁCTICA



SESION 05

APRENDIZAJE SUPERVISADO TECNICAS DE MACHINE LEARNING-PCA

Juan Antonio Chipoco Vidal
jchipoco@gmail.com

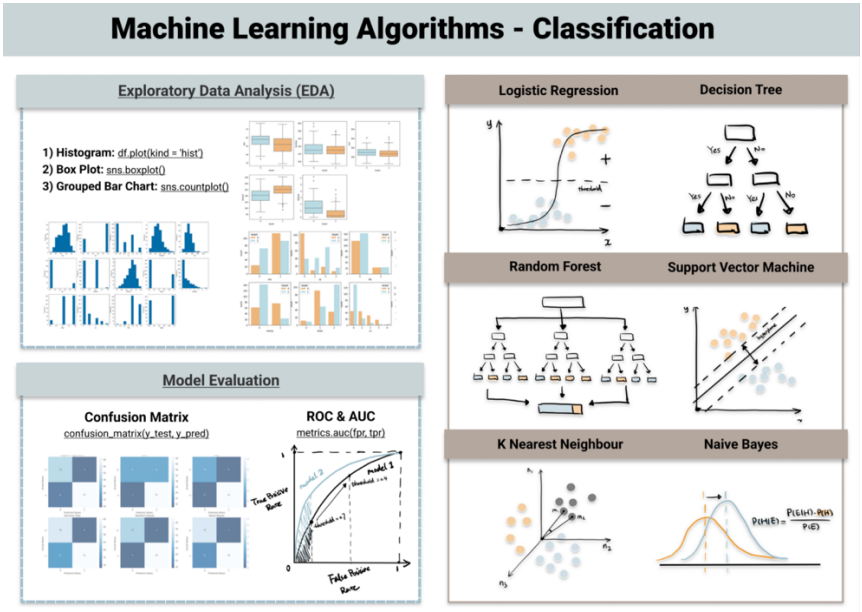
ÍNDICE

OBJETIVO 4
REGRESION LOGISTICA 5
SUPPORT VECTOR MACHINES (SVM) 6
K NEAREST NEIGBHOR 7
DECISION TREE 8
NAIVE BAYES 9
RANDOM FOREST CLASSIFIER 10
RANDOM FOREST CLASSIFIER 11
MÉTRICAS DE PERFORMANCE PARA CLASIFICACION 12
MATRIZ DE CONFUSION 13
MATRIZ DE CONFUSION 14
MATRIZ DE CONFUSION 16
MATRIZ DE CONFUSION 17
MATRIZ DE CONFUSION 18
CURVA AUC-ROC 19
PCA 20

Objetivo

El objetivo de esta sesion es conocer los algoritmos de machine learning mas utilizados para el aprendizaje supervisado.

Revisaremos como funcionan los algoritmos de clasificacion y luego procederemos a aplicarlos en nuestrar practica semanal. Tambien revisaremos una de las tecnicas de reduccion de dimensionalidad conocida como PCA:



Los principales algoritmos de clasificación en el aprendizaje automático son los siguientes:

Modelos lineales:

Regresión logística
Máquinas de vectores de soporte (SVM)

Modelos no lineales:

K-vecinos más cercanos
Arboles de Decision (Clasificacion)
Bosques Aleatorios (Clasificacion)

Algunos de los problemas en los que se pueden utilizar algoritmos de clasificación son:

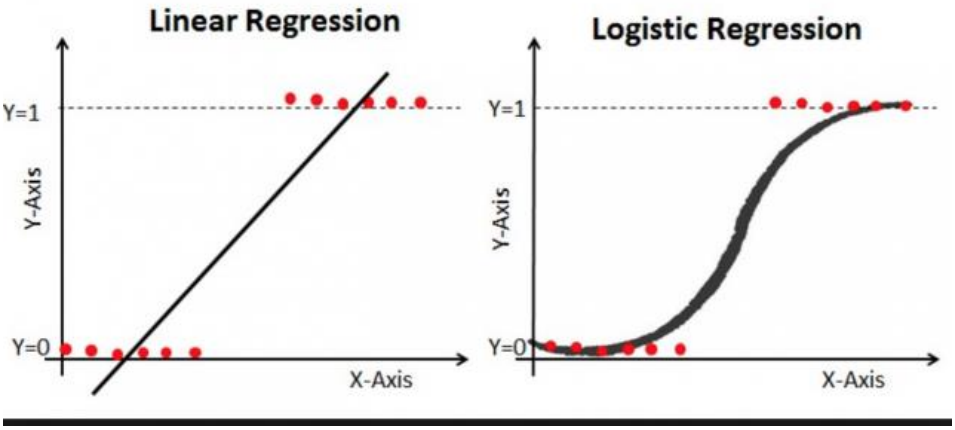
Text Categorization
Fraud Detection
Optical Character Recognition
Face Detection
Language Detection
Customer Segmentation

Regresion Logistica

La regresión logística es un algoritmo de clasificación muy útil en el aprendizaje automático. La regresión logística se utiliza para encontrar una relación entre las entidades de entrada y las etiquetas de salida. Pero eso es lo que hacemos en los problemas de regresión, entonces, ¿cómo es la regresión logística un algoritmo de clasificación?

La respuesta es que en la regresión encontramos el valor de la etiqueta pero se usa la regresión logística para encontrar la categoría de la etiqueta.

Por ejemplo, si queremos predecir las calificaciones de un estudiante, entonces este es el problema de regresión, pero si queremos predecir si el estudiante aprobará o reprobará el examen, entonces es el problema de clasificación donde se puede usar la regresión logística.

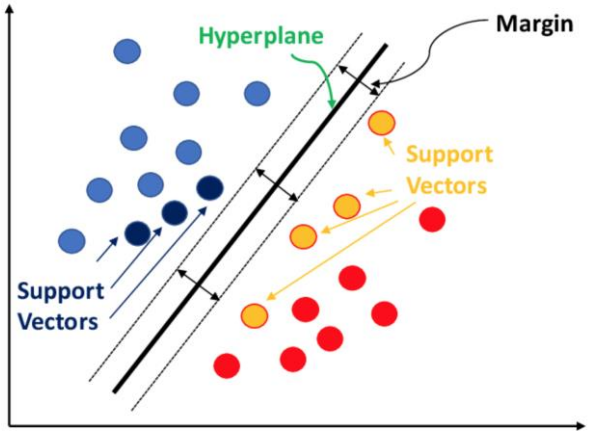


Support Vector Machines (SVM)

Support Vector Machine es un algoritmo de aprendizaje automático muy potente que se puede utilizar tanto para tareas de clasificación como de regresión. Pero dado que SVM se usa más en la clasificación, podemos decir que también es un algoritmo de clasificación.

El algoritmo SVM se utiliza para aprender half-spaces con algún conocimiento de la preferencia de margen grande. Hay dos tipos de SVM; Hard-SVM, Soft-SVM. El Hard-SVM encuentra el half-space que divide perfectamente los datos con el mayor margen. Soft-SVM no encuentra ningún punto de la división, permite violar las restricciones hasta cierto punto.

La gran fortaleza de SVM es que el entrenamiento es muy simple. No requiere ningún óptimo, a diferencia de las redes neuronales. También ajusta muy bien los datos a datos dimensionales muy altos.

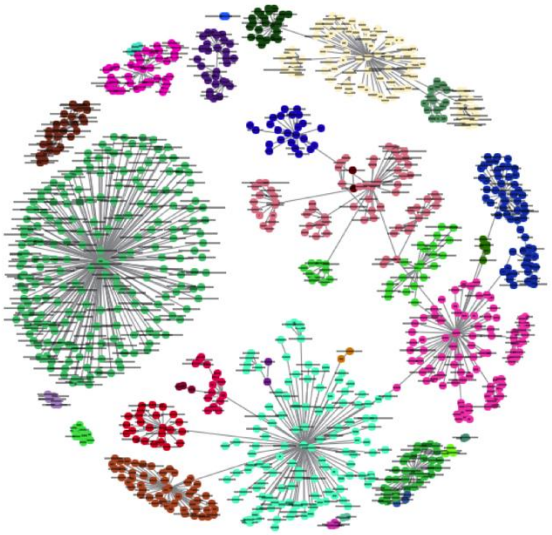


K Nearest Neigbhor

K Nearest Neighbor es uno de los algoritmos de clasificación más simples en el aprendizaje automático. Funciona aprendiendo de los datos de entrenamiento y luego predice la etiqueta de cualquier categoría en función de las etiquetas de sus vecinos más cercanos en los datos de entrenamiento.

De ello se deduce que las características utilizadas para describir la estructura de los puntos de datos son más relevantes para sus etiquetas de una manera que los acerca a los puntos para tener la misma etiqueta.

KNN es un algoritmo de clasificación de aprendizaje automático muy simple que se basa en la suposición de que los elementos que se parecen deben ser iguales. Necesitamos que todo el conjunto de entrenamiento se almacene durante el entrenamiento, y mientras probamos el algoritmo, necesitamos probar todo el conjunto de datos para encontrar el vecino más cercano.



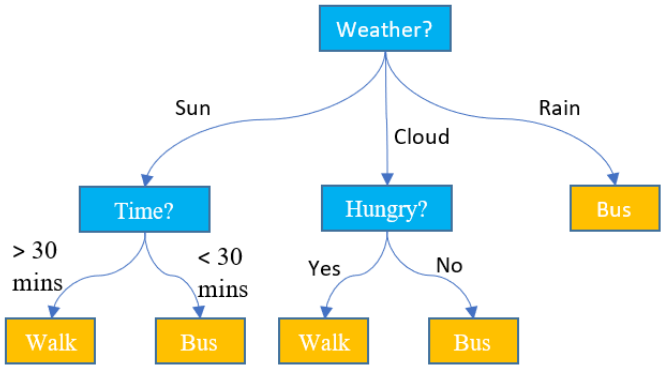
Decision Tree

Un árbol de decisión es un algoritmo que predice la etiqueta asociada con una instancia viajando desde un nodo raíz de un árbol hasta una hoja. Por ejemplo, necesitamos clasificar si la papaya es sabrosa o no, veamos cómo se expresará el árbol de decisión en este problema.

Para clasificar si la papaya es sabrosa o no, el algoritmo del árbol de decisión verificará primero el color de la papaya. Si el color no se encuentra entre el verde pálido y el amarillo pálido, entonces el algoritmo predecirá que la papaya no es sabrosa sin observar más características.

Entonces, lo que sucede es que comenzamos con un árbol con una sola hoja y lo etiquetamos según la mayoría de las etiquetas que tenemos en el conjunto de aprendizaje. Luego hacemos algunas iteraciones, con cada iteración vemos el efecto de dividir la hoja única.

Luego, entre todo el número posible de divisiones, o seleccionamos la que maximiza la ganancia, o elegimos no seleccionar la hoja en absoluto.

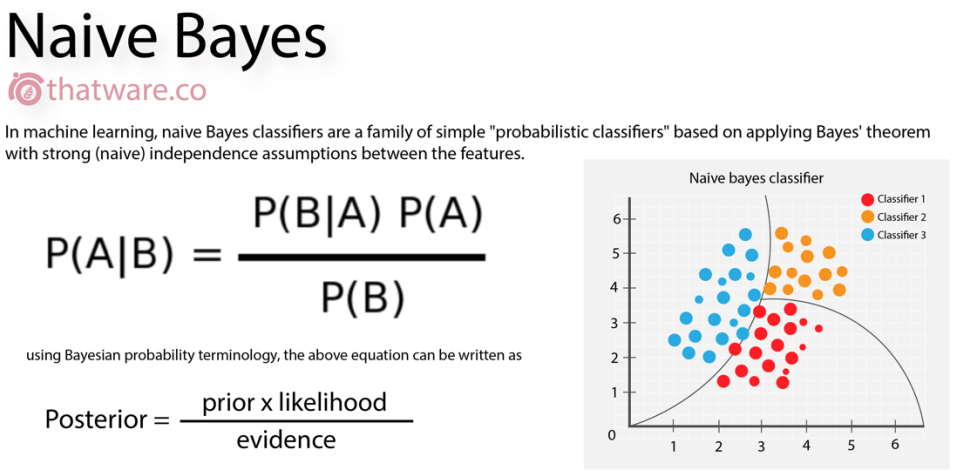


Naive Bayes

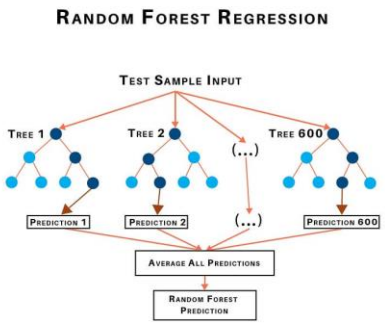
Naive Bayes es un poderoso algoritmo de aprendizaje automático supervisado que se utiliza para problemas de clasificación. Utiliza características para predecir una variable objetivo. La diferencia entre Naive Bayes y otros algoritmos de clasificación es que Naive Bayes asume que las características son independientes entre sí y que no hay correlación entre las características.

Entonces lo que pasa es que esta hipótesis no se evalúa en base a cuestiones de la vida real. Por lo tanto, esta suposición ingenua de que las características no están correlacionadas es la razón por la que este algoritmo se conoce como Naive Bayes.

Este algoritmo es una demostración clásica de cómo las suposiciones generativas y las estimaciones de parámetros pueden simplificar el proceso de aprendizaje. En el algoritmo Naive Bayes, hacemos predicciones asumiendo que las características dadas son independientes entre sí.



Random Forest Classifier



Recordemos que la varianza es la medida de la variabilidad de un conjunto de datos que indica hasta qué punto se distribuyen los diferentes valores. Matemáticamente, se define como la suma de los cuadrados de las diferencias entre una variable y su media, dividido entre el numero de datos.

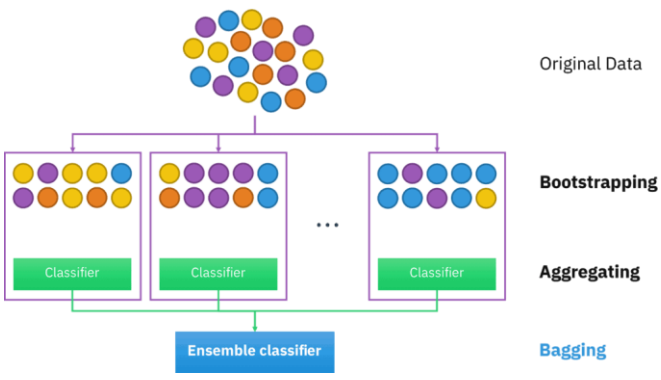
Veamos el caso de los árboles de decisión. Como sabemos, pueden reconstruir patrones muy complejos, pero tienden a tener un rendimiento inferior incluso si se producen cambios menores en los datos. Es por eso que un árbol de decisiones independiente no obtendrá grandes resultados. Aún así, si compone muchos de estos árboles, el rendimiento predictivo mejorará drásticamente. Esto es un método de conjunto llamado Random Forest.

¿Qué es el esemble learning?

La idea general del *esemble learning* es bastante simple. Debe entrenar varios algoritmos de ML y combinar sus predicciones de alguna manera. Tal enfoque tiende a hacer predicciones más precisas que cualquier modelo individual. Un modelo Ensemble es un modelo que consta de muchos modelos base.

Bagging

Bootstrap Aggregating o Bagging es una técnica bastante simple pero realmente poderosa. Comprender el concepto general de Bagging es realmente crucial, ya que es la base del algoritmo Random Forest (RF).



Random Forest Classifier

En general, el bagging es una buena técnica que ayuda a manejar el sobreajuste y reduce la varianza.

¿Qué es un bosque aleatorio?

Random Forest es un algoritmo de aprendizaje supervisado que se basa en el método de esemble learning y muchos árboles de decisión. Random Forest es una técnica de bagging, por lo que todos los cálculos se ejecutan en paralelo y no hay interacción entre los árboles de decisión al construirlos. RF se puede utilizar para resolver tareas de clasificación y regresión.

El nombre "bosque aleatorio" proviene de la idea de bagging de la aleatorización de datos (aleatorio) y la construcción de múltiples árboles de decisión (bosque). En general, es un poderoso algoritmo de ML que limita las desventajas de un modelo de árbol de decisiones.

Métricas de performance para clasificacion

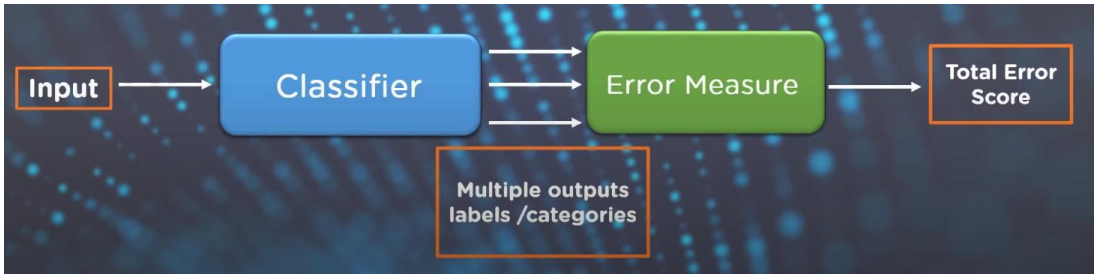
Metric	Formula	Evaluation focus
Accuracy	$ACC = \frac{TP + TN}{TP + TN + FP + FN}$	Overall effectiveness of a classifier
Precision	$PRC = \frac{TP}{TP + FP}$	Class agreement of the data labels with the positive labels given by the classifier
Sensitivity	$SNS = \frac{TP}{TP + FN}$	Effectiveness of a classifier to identify positive labels. Also called true positive rate (TPR)
Specificity	$SPC = \frac{TN}{TN + FP}$	How effectively a classifier identifies negative labels. Also called true negative rate (TNR)
F_1 score	$F_1 = 2\frac{PRC \cdot SNS}{PRC + SNS}$	Combination of precision (PRC) and sensitivity (SNS) in a single metric
Geometric mean	$GM = \sqrt{SNS \cdot SPC}$	Combination of sensitivity (SNS) and specificity (SPC) in a single metric
Area under (ROC) curve	$AUC = \int_0^1 SNS \cdot dSPC$	Combined metric based on the receiver operating characteristic (ROC) space (*Powers, 2011*)

- Los problemas de clasificación son una de las áreas más investigadas del mundo. Los casos de uso están presentes en casi todos los entornos industriales y de producción. Reconocimiento de voz, reconocimiento facial, clasificación de texto: la lista es interminable.
- Los modelos de clasificación tienen una salida discreta, por lo que necesitamos una métrica que compare clases discretas de alguna forma. Las métricas de clasificación evalúan el rendimiento de un modelo y te dicen qué tan buena o mala es la clasificación, pero cada una de ellas lo evalúa de manera diferente.
- Para evaluar los modelos de clasificación, discutiremos estas métricas en detalle:
- Matriz de confusión

  - ✓ Accuracy
  - ✓ Precisión
  - ✓ Recall (Sensitivity)
  - ✓ F1-Score
  - ✓ ROC-AUC

Matriz de confusion

- Los modelos de clasificacion tienen multiples categorias de salida. La mayoria de las metricas de error nos indicaran el error total de un modelo, pero a partir de eso no podremos averiguar errores individuales en nuestro modelo.

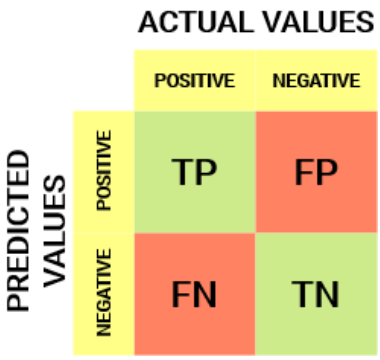


- Una matriz de confusión es una matriz N x N utilizada para evaluar el rendimiento de un modelo de clasificación, donde N es el número de clases objetivo. La matriz compara los valores objetivo reales con los predichos por el modelo de aprendizaje automático. Esto nos da una visión holistica de qué tan bien está funcionando nuestro modelo de clasificación y qué tipo de errores está cometiendo.

		Predicted Class: Positive	Predicted Class: Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	**Negative Predictive Value** $\frac{TN}{(TN+FN)}$	**Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$

Matriz de confusion

- Analizemos para el caso de una clasificacion binaria, supongamos que deseamos predecir cuántas personas están infectadas con un virus peligroso antes de que muestren los síntomas y en base al eso aislarlos de la población sana.



True Positive (TP)

El valor predicho coincide con el valor real.

El valor real fue positivo y el modelo predijo un valor positivo

True Negative (TN)

El valor predicho coincide con el valor real

El valor real fue negativo y el modelo predijo un valor negativo

False Positive (FP): Type I error

El valor predicho fue predicho falsamente

El valor real fue negativo pero el modelo predijo un valor positivo

También conocido como error tipo 1.

False Negative (FN): Type II error

El valor predicho fue predicho falsamente

El valor real fue positivo pero el modelo predijo un valor negativo

También conocido como el error tipo 2

- Prosiguiendo con nuestro ejemplo nos damos cuenta que nuestro conjunto de datos es un ejemplo de un conjunto de datos no balanceados (*imbalanced dataset*). Hay 997 puntos de datos para la clase negativa y 3 puntos de datos para la clase positiva.

ID	Actual Sick?	Predicted Sick?	Outcome
1	1	1	TP
2	0	0	TN
3	0	0	TN
4	1	1	TP
5	0	0	TN
6	0	0	TN
7	1	0	FP
8	0	1	FN
9	0	0	TN
10	1	0	FP
:	:	:	:
1000	0	0	FN

- **Accuracy,** la exactitud se refiere a cuán cerca del valor real se encuentra el valor medido.

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

De la tabla obtenemos TP = 30, TN = 930, FP = 30, FN = 10

$$Accuracy = \frac{30 + 930}{30 + 30 + 930 + 10} = 0.96$$

El 96% nos indica que porcentaje de casos negativos y positivos acerto el modelo. Nada mas. Esto nos conduce a introducir los conceptos de Precision y Recall.

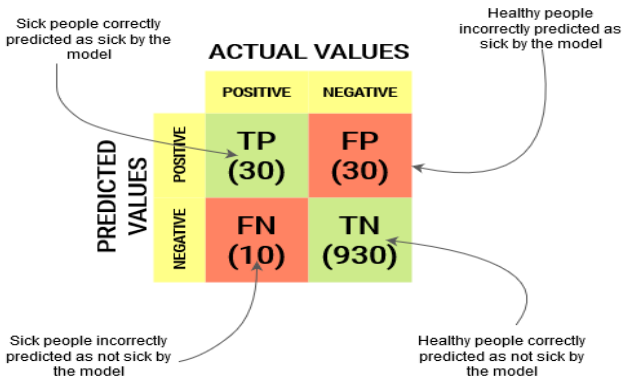
- **Precision**, se refiere a la dispersión del conjunto de valores obtenidos de mediciones repetidas de una magnitud. Cuanto menor es la dispersión mayor la precisión. Nos dice cuántos de los casos predichos como positivos resultaron ser positivos realmente.

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

- **Recall**, la sensitividad (True positive rate) nos dice cuántos de los casos positivos reales pudimos predecir correctamente con nuestro modelo.

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

Matriz de confusion



$$Precision = \frac{30}{30+30} = 0.5$$

$$Recall = \frac{30}{30+10} = 0.75$$

El 50% por ciento de los casos pronosticados correctamente resultaron ser casos positivos. Mientras que el 75% de los positivos fueron predichos con éxito por nuestro modelo.

Precision es una métrica útil en los casos en los que los falsos positivos son más preocupantes que los falsos negativos.

La precisión es importante en los sistemas de recomendación de música o video, sitios web de comercio electrónico, etc. Los resultados incorrectos pueden provocar la pérdida de clientes y ser perjudiciales para el negocio.

Recall es una métrica útil en los casos en que el Falso Negativo supera al Falso Positivo.

Recall es importante en los casos médicos en los que no importa si activamos una falsa alarma, Los casos positivos reales no deben pasar desapercibidos, pues no queremos dar de alta accidentalmente a una persona infectada y dejar que se mezcle con la población sana, propagando así el virus contagioso.

En los casos que no esta claro cual de los dos es mas importante procedemos a combinarlos.

- **F1-Score,** en la práctica, cuando tratamos de aumentar la precisión de nuestro modelo, el recall disminuye y viceversa. La puntuación F1 captura ambas tendencias en un solo valor**:**

$$F1 - score = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}}$$

Matriz de confusion

- **False Negative Rate** (FNR) nos dice qué proporción de la clase positiva fue clasificada incorrectamente por el clasificador.

  Es deseable un TPR más alto y un FNR más bajo, ya que queremos clasificar correctamente la clase positiva.

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

- **Specificity** (**True Negative Rate** , TNR) La especificidad nos dice qué proporción de la clase negativa se clasificó correctamente.

  Tomando el mismo ejemplo que en Sensibilidad, Especificidad significaría determinar la proporción de personas sanas que fueron identificadas correctamente por el modelo.

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

- **False Positive Rate** (FPR) nos dice qué proporción de la clase negativa fue clasificada incorrectamente por el clasificador.

  Es deseable una TNR más alta y una FPR más baja, ya que queremos clasificar correctamente la clase negativa.

$$FPR = \frac{FP}{TN + FP} = 1 - Specificity$$

- De todas estas métricas, la Sensibilidad y la Especificidad son quizás las más importantes y veremos más adelante cómo se utilizan para construir una métrica de evaluación. Pero antes de eso, entendamos por qué la probabilidad de predicción es mejor que predecir la clase objetivo directamente.
- **Probabilidad de predicciones,** se puede usar un modelo de clasificación de aprendizaje automático para predecir la clase real del punto de datos directamente o predecir su probabilidad de pertenecer a diferentes clases. Esto último nos da más control sobre el resultado. Podemos determinar nuestro propio umbral para interpretar el resultado del clasificador. Establecer diferentes umbrales para clasificar la clase positiva para los puntos de datos cambiará inadvertidamente la Sensibilidad y la Especificidad del modelo. Y uno de estos umbrales probablemente dará un mejor resultado que los demás, dependiendo de si nuestro objetivo es reducir el número de falsos negativos o falsos positivos.

Matriz de confusion

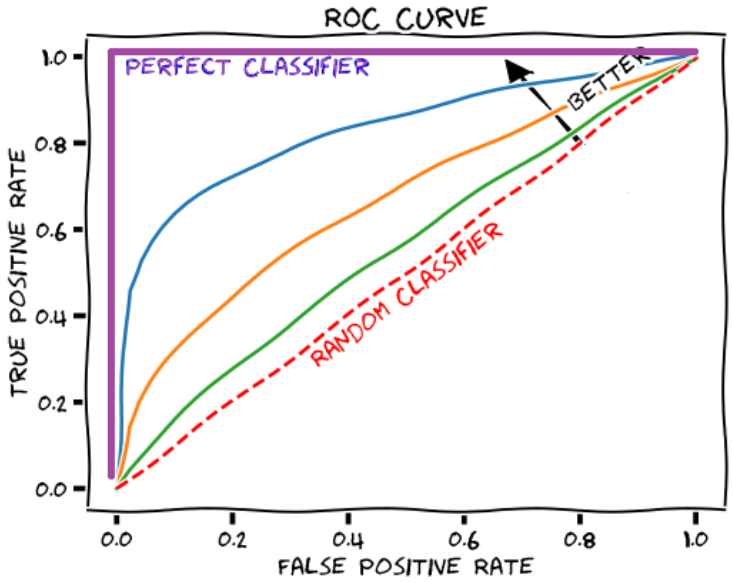
Veamos la siguiente tabla:

ID	Actual	Prediction Probability	>0.6	>0.7	> 0.8	Metric
1	0	0.98	1	1	1	
2	1	0.67	1	0	0	
3	1	0.58	0	0	0	
4	0	0.78	1	1	0	
5	1	0.85	1	1	1	
6	0	0.86	1	1	1	
7	0	0.79	1	1	0	
8	0	0.89	1	1	1	
9	1	0.82	1	1	1	
10	0	0.86	1	1	1	
			0.75	0.5	0.5	TPR
			1	1	0.66	FPR
			0	0	0.33	TNR
			0.25	0.5	0.5	FNR

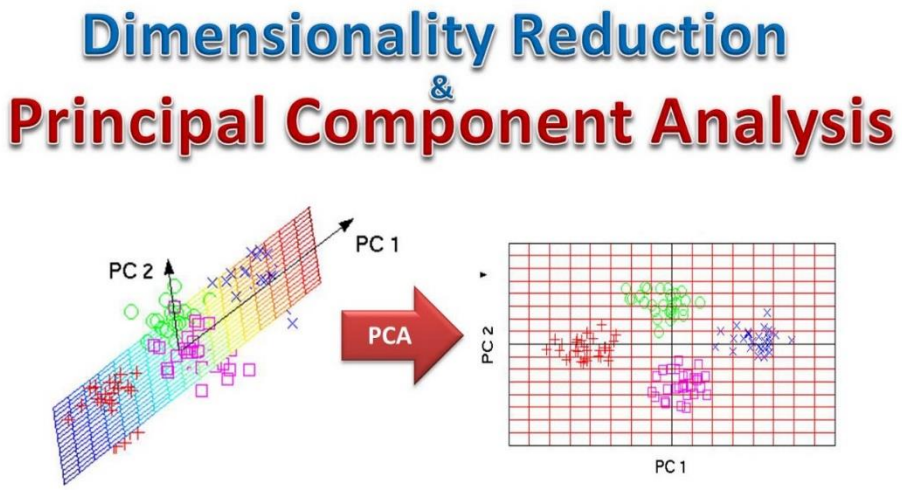
Las métricas cambian con los valores de umbral cambiantes. Podemos generar diferentes matrices de confusión y comparar las diversas métricas que discutimos en la sección anterior. Pero no sería prudente hacerlo. En cambio, lo que podemos hacer es generar un gráfico entre algunas de estas métricas para que podamos visualizar fácilmente qué umbral nos está dando un mejor resultado. Esta es la curva AUC-ROC.

Curva AUC-ROC

La curva característica del operador del receptor (ROC) es una métrica de evaluación para problemas de clasificación binaria. Es una curva de probabilidad que traza la TPR contra la FPR con varios valores de umbral y esencialmente separa la "señal" del "ruido". El área bajo la curva (AUC) es la medida de la capacidad de un clasificador para distinguir entre clases y se utiliza como resumen de la curva ROC.



PCA



PCA es una reducción de dimensionalidad que identifica relaciones importantes en nuestros datos, transforma los datos existentes en función de estas relaciones y luego cuantifica la importancia de estas relaciones para que podamos mantener las relaciones más importantes y descartar las demás. Para recordar esta definición, podemos dividirla en cuatro pasos:

1 Identificamos la relación entre las variables predictoras a través de una Matriz de Covarianza.
2 A través de la transformación lineal o autodescomposición de la Matriz de Covarianza, obtenemos autovectores y autovalores.
3 Luego transformamos nuestros datos usando vectores propios en componentes principales.
4 Por último, cuantificamos la importancia de estas relaciones utilizando valores propios y conservamos los componentes principales importantes.