AULA ESTADÍSTICA

Serie Monográfica: introducción a técnicas de clasificación y aprendizaje automático

Autor: Jorge Lorenzo

Julio 2024

Aula Estadística Serie Monográfica

Introducción

En esta monografía presentaremos una técnica simple de clasificación basada en la visualización de datos: la curva ROC. Dicha técnica es ampliamente utilizada para la reducción de información en sistemas clasificatorios, particularmente en los modelos de regresión. Asimismo, proporciona una versatilidad que la hace apta para ser empleada en modelos complejos de aprendizaje automático, especialmente en la visualización de resultados de árboles de clasificación. En la última sección, se repasarán brevemente dos de estas técnicas conocidas como *Bagging* y *Boosting*.

Técnicas de clasificación: Curvas ROC

Las curvas ROC es el acrónimo de *Receiver Operating Characteristic*, traducido en español como Característica Operativa del Receptor o Característica de Funcionamiento del Receptor, según las fuentes consultadas. Se trata de una herramienta fundamental en la estadística para evaluar el rendimiento de modelos de clasificación binaria, esto es, cuando la variable dependiente asume solo dos valores. La mayoría de los ejemplos que se proponen para explicar el uso de estas curvas provienen de la medicina o de funciones de producción. En el primer caso, se trata de diagnósticos de enfermedades donde, dada una serie de variables predictoras, los pacientes se clasifican según tengan o no una determinada condición médica. En el segundo caso se analizan variables de producción, cuyo resultado final puede ser que un producto presente o no un defecto al final de su montaje. En la investigación educativa estas técnicas se han utilizado como modelo para estudiar el rendimiento académico cuando este se sintetiza en una variable binaria, v.g. aprobado – rechazado. Dado que además se basan en la combinación de distintas variables predictoras, son de mucha utilidad para clarificar modelos de regresión tanto logística como bayesiana. En este artículo veremos cómo se construyen y se interpretan las curvas ROC mediante un ejemplo práctico. En una segunda parte discutiremos la aplicación de estos procedimientos para técnicas clasificatorias más sofisticadas.

Un ejemplo con datos

Analizando algunas variables relevantes a partir de la historia académica de los estudiantes, puede ser posible proponer un modelo que estime si un estudiante aprobará o no un examen, o bien, finalizará o no un curso académico. El propósito del modelo es acercar una predicción de los resultados que se observarán posteriormente. En caso que dicho modelo resulta apropiado, servirá para estimar el comportamiento de otros estudiantes con el conjunto de variables seleccionadas.

Para este ejemplo, supondremos que los valores probabilísticos que se presentan en la tabla que sigue, fueron obtenidos luego de aplicar un modelo de regresión. Los datos se resumen en columnas con las probabilidades predichas y el resultado real observado, para una muestra de diez estudiantes.

Estudiante	Probabilidad Predicha	Resultado Real (Aprobó: 1, No Aprobó: 0)
1	0.95	1
2	0.85	1
3	0.80	0
4	0.70	1
5	0.65	0
6	0.55	1
7	0.45	0
8	0.30	0
9	0.25	1
10	0.15	0

Los datos de la tabla se hallan ordenados de mayor a menor probabilidad asignada de aprobar. Allí pude leerse que el estudiante 1 tiene una probabilidad predicha de aprobar p= 0,95 y el resultado observado es que efectivamente aprobó. El estudiante 10 tiene una probabilidad predicha de aprobar p= 0,15, y el resultado observado es que no aprobó. ¿Qué ocurre con los estudiantes 3 y 9? Para estos alumnos tenemos que la probabilidad predicha no se contrasta con el resultado real observado; en otras palabras, se espera que el estudiante 3 apruebe (y en realidad no aprobó), mientras que el estudiante 9 se espera que no apruebe (y en realidad si aprobó). En este caso, el problema que se desea resolver es crear un sistema clasificatorio que minimice los errores de predicción.

Una curva ROC resulta en este caso una herramienta útil para derivar del modelo de regresión, las probabilidades asignadas a priori y el resultado final observado. En el siguiente apartado se mostrarán los pasos para calcular y graficar esta curva.

Definición de métricas y umbrales

Para construir la curva ROC, necesitamos calcular: a) TPR (*True Positive Rate*, o tasa de verdaderos positivos), lo cual se denomina sensibilidad. Es la proporción de verdaderos positivos entre todos los casos reales positivos; b) FPR (*False Positive Rate*, o tasa de falsos positivos) que se denomina especificidad. Es la proporción de falsos positivos entre todos los casos reales negativos. Las ecuaciones para estas dos métricas son las siguientes:

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

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

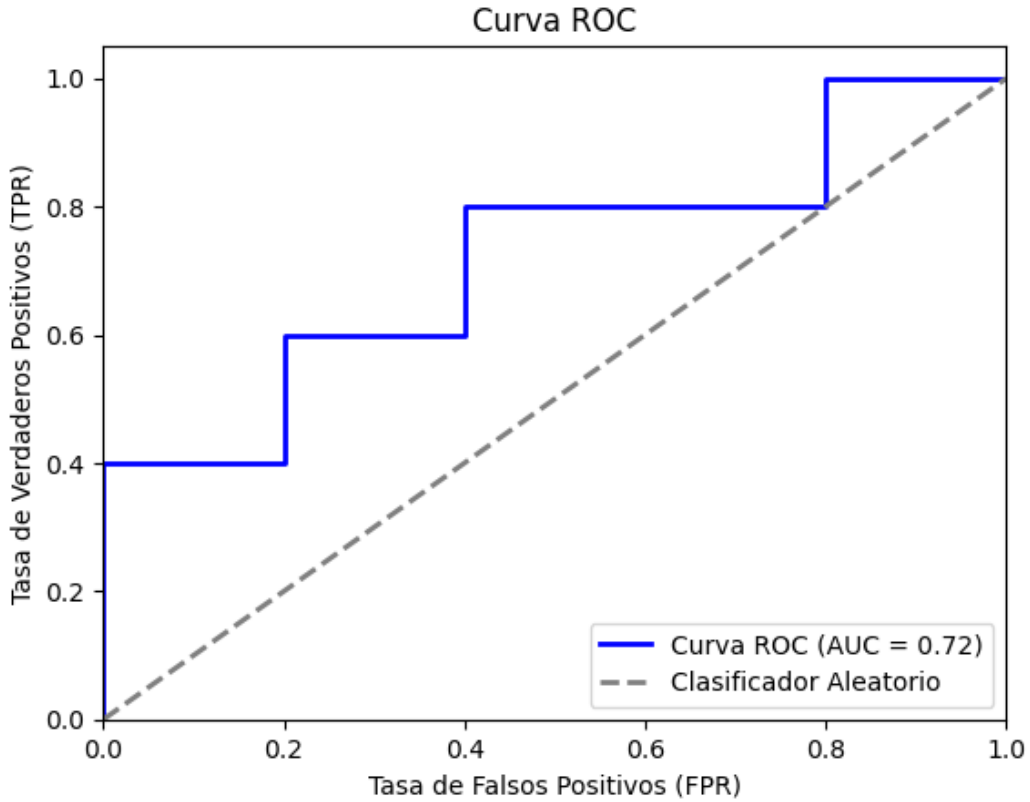
Dónde: TP (true positive)= positivos verdaderos; FN (false negative)= falsos negativos; FP (false positive)= falsos positivos y TN (true negative)= verdadero negativo.

La curva ROC se grafica sobre ejes cartesianos, donde el eje *x* se ocupa con los valores asignados a la tasa de falsos positivos (FPR); en el eje *y* se asignan los valores de la tasa de verdaderos positivos (TPR). Ambos ejes asumen valores entre 0 y 1. A partir de aquí se busca el umbral de clasificación que minimice el error de clasificación. Con la curva ROC podemos comparar distintos métodos de clasificación, así el mejor modelo de clasificación es el que encuentra el punto más alto en la curva. En otras palabras, cuanto mayor sea el área bajo la curva, mejor será el clasificador. Esta área se refleja en el valor AUC (*Area Under the Curve*, área bajo la curva). El valor AUC varía entre 0 y 1, cuanto mayor sea dicho valor, mejor será el clasificador.

Con los datos de este ejemplo es posible implementar un script usando Python y `sklearn` para visualizar la curva ROC. A continuación, se transcribe el código que se puede correr en *Jupyter*.

```
# Importar las bibliotecas necesarias
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve, roc_auc_score


# Datos de predicciones y etiquetas verdaderas
y_true = np.array([1, 1, 0, 1, 0, 1, 0, 0, 1, 0])
y_scores = np.array([0.95, 0.85, 0.80, 0.70, 0.65, 0.55, 0.45, 0.30, 0.25, 0.15])


# Calcular las tasas de verdaderos positivos y falsos positivos
fpr, tpr, thresholds = roc_curve(y_true, y_scores)


# Calcular el AUC
roc_auc = roc_auc_score(y_true, y_scores)


# Graficar la curva ROC
plt.figure()
plt.plot(fpr, tpr, color='blue', lw=2, label='Curva ROC (AUC = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='grey', lw=2, linestyle='--', label='Clasificador
Aleatorio')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('Tasa de Falsos Positivos (FPR)')
plt.ylabel('Tasa de Verdaderos Positivos (TPR)')
plt.title('Curva ROC - Predicción de Aprobación de Estudiantes')
plt.legend(loc="lower right")
plt.show()
```

Como resultado obtendremos la siguiente curva:



La interpretación de la Curva ROC es relativamente sencilla, deben tenerse en cuenta los valores de los pares ordenados de los TPR y FPR, los cuales muestran cómo en la curva varían la tasa de verdaderos positivos y la tasa de falsos positivos a medida que se cambia el umbral de decisión. La diagonal, denominada clasificador aleatorio, representa un modelo aleatorio cuya probabilidad de acertar en una clasificación tiene una probabilidad p=0,5 (50%). El área bajo la curva (AUC), mide la capacidad del modelo para distinguir entre las clases. Como valores de referencia se toma un AUC de 0.5 que indica un modelo aleatorio, mientras que un AUC de 1 indica un modelo perfecto.

El valor obtenido con los datos del ejemplo, muestra un AUC de 0,72. Esto indica que el modelo utilizado para obtener las probabilidades predichas, tiene una capacidad razonablemente buena para distinguir entre estudiantes que aprobarán y los que no aprobarán el examen, pero no es perfecto. Para un AUC de 0.72 se tiene que la capacidad de discriminación indicaría que en el 72% de los casos, el modelo asignará una puntuación más alta a un estudiante que aprueba el examen en comparación con uno que no lo aprueba. Esto significa que hay un 72% de probabilidad de que el modelo clasifique correctamente a un estudiante positivo (que aprueba) como más probable que un estudiante negativo (que no aprueba). El modelo evidencia un buen rendimiento, aunque hay margen de mejora. Un AUC entre 0.7 y 0.8 se considera aceptable y muestra que el modelo tiene una capacidad discriminativa sustancial, no obstante, el modelo puede cometer errores en aproximadamente el 28% de los casos.

En conclusión, el uso práctico de las curvas ROC en educación, puede ayudar a evaluar modelos de predicción que utilicen diferentes variables para un mismo fin, v.g. predecir el éxito de los estudiantes. Al aplicarse más de un modelo, es posible tomar decisiones sobre cuáles son los umbrales apropiados para tomar decisiones basadas en predicciones (por ejemplo, intervención temprana para estudiantes en riesgo). Con esta

herramienta, los educadores y administradores pueden tomar decisiones informadas para mejorar el rendimiento y el apoyo a los estudiantes.

Modelo para obtener las probabilidades predichas

La curva ROC puede obtenerse fácilmente de la regresión logística, y usualmente se toma el 50% como umbral. En otras palabras, con una regresión logística se puede crear una curva ROC y determinar si las variables del modelo logístico mejoran la capacidad de predicción. Para no saturar un modelo con muchas variables predictoras, se puede especificar dos o más modelos con distintas combinaciones de variables, en tal caso la curva ROC será la herramienta que permita determinar cuál es el mejor modelo.

Un modelo de predicción puede estimarse a partir de una regresión bayesiana. La regresión bayesiana es una extensión del modelo de regresión lineal que incorpora la probabilidad para estimar los parámetros del modelo. A diferencia de la regresión lineal clásica, que utiliza estimaciones puntuales para los coeficientes, la regresión bayesiana utiliza distribuciones de probabilidad para capturar la incertidumbre en las estimaciones.

El ejemplo ofrecido tiene una finalidad didáctica, en la actualidad los modelos clasificatorios pueden utilizar una gran cantidad de variables y distribuciones conjuntas de probabilidades. Existen distintas técnicas clasificatorias, de las cuales comentaremos dos, que son muy utilizadas en aprendizaje automático.

Técnicas de *Bagging* y *Boosting* en aprendizaje automático

Tanto el bagging como el boosting son técnicas de aprendizaje diseñadas para mejorar el rendimiento de los modelos de aprendizaje automático al combinar predicciones de múltiples bases de aprendizajes. Estos enfoques difieren en sus metodologías, aplican diferentes estrategias y persiguen distintos objetivos. En esta monografía, se expondrán algunas características de las técnicas, atendiendo especialmente a su uso en el tratamiento masivo de datos de estudiantes y la predicción de rendimiento académico.

El bagging es una técnica de aprendizaje por conjuntos que se centra en reducir la varianza y mejorar la estabilidad de los modelos de aprendizaje automático. El término se deriva de la idea de crear múltiples subconjuntos de datos de entrenamiento mediante un proceso de bootstrapping. A su vez, el bootstrapping consiste en muestrear aleatoriamente un conjunto de datos para generar subconjuntos del mismo tamaño a partir de los datos originales. Luego, cada uno de estos subconjuntos se utiliza para entrenar a una base de aprendizaje independiente. Uno de los principales objetivos del bagging es reducir el sobreajuste, exponiendo a cada base de aprendizaje a variaciones ligeramente diferentes de los datos de entrenamiento. El algoritmo más utilizado en este proceso se denomina *Random Forest*, el cual se construye a partir de una colección de árboles de decisión, cada uno entrenado con un subconjunto diferente de los datos originales. Durante el entrenamiento, cada árbol se forma seleccionando un subconjunto aleatorio para cada rama o partición, añadiendo una capa extra de aleatoriedad y diversidad al conjunto. La predicción final se obtiene promediando o votando entre las predicciones de los árboles individuales.

Una ventaja del bagging es su capacidad para manejar de forma eficaz, conjuntos de datos con lotes heterogéneos y valores atípicos, dado que el modelo agrega predicciones de múltiples bases de aprendizajes. En el bagging cada base de aprendizaje puede entrenarse de forma independiente, lo que permite implementaciones eficientes. En síntesis, la técnica de bagging resulta útil cuando se trata de modelos complejos con una varianza elevada. Cuando se trata de unidades de

análisis que operan con variables en diferentes niveles, el solo agregado de variables con alta colinealidad y errores no aleatorios produce efectos que limitan las técnicas tradicionales de regresión y clustering, especialmente el análisis de correspondencias múltiple, siendo estas técnicas frecuentemente utilizadas en educación.

El boosting, al igual que el bagging, es una técnica de aprendizaje por conjuntos, pero su objetivo es mejorar el rendimiento de los aprendizajes débiles combinándolos de forma secuencial. La idea central es otorgar mayor peso a las instancias mal clasificadas durante un proceso de entrenamiento, lo que permite a los aprendizajes posteriores, centrarse en los errores cometidos por sus predecesores. Esto es un equivalente de la retropropagación del error en los modelos de redes neurales. El boosting no se basa en subconjuntos de datos obtenidos por bootstrapping. En su lugar, asigna pesos a cada instancia del conjunto de entrenamiento y ajusta estos pesos a lo largo de las iteraciones. En cada iteración, se entrena un nuevo proceso de aprendizaje débil con los datos anteriores y se aumentan las ponderaciones de las instancias mal clasificadas. Esto permite que la etapa de aprendizaje siguiente preste más atención a los ejemplos clasificados erróneamente con anterioridad.

El algoritmo de boosting más conocido es AdaBoost (*Adaptive Boosting*), mediante el cual las primeras etapas de aprendizajes suelen ser modelos simples con escaso poder predictivo: árboles de decisión poco profundos o con una única división. Cada etapa de aprendizaje se entrena de forma secuencial y en cada iteración se incrementan los pesos de las instancias mal clasificadas, lo que obliga al modelo a centrarse en los ejemplos difíciles de clasificar. De este modo, la ventaja del boosting es su capacidad para manejar relaciones complejas en los datos y mejorar significativamente el rendimiento en los distintos pasos de los aprendizajes. El boosting suele superar al bagging a la hora de reducir tanto el sesgo como la varianza. Sin embargo, el boosting es más sensible a los datos ruidosos y a los valores atípicos. De lo dicho hasta aquí, es posible crear una tabla comparativa con ambas técnicas.

Características de las técnicas de Bagging y Boosting

Bagging	Boosting
Las bases de aprendizaje se entrenan independientemente y en paralelo, trabajando con un subconjunto diferente de datos. La predicción final es una media o un voto	Las bases de aprendizaje se entrenan secuencialmente, cada una se centra en corregir los errores de sus predecesores. La predicción final es una suma ponderada de las predicciones de cada proceso
Utiliza bootstrapping para crear múltiples subconjuntos de datos de entrenamiento.	Asigna pesos a las instancias del conjunto de entrenamiento, con pesos más altos a las instancias mal clasificadas para guiar a los aprendices posteriores
Todas las base de aprendizaje tienen el mismo peso a la hora de realizar la predicción final	Asigna diferentes pesos a cada base de aprendizaje en función de su rendimiento
Robusto frente a datos ruidosos y valores atípicos gracias al mecanismo de promediado o votación, que reduce el impacto de los errores individuales.	Más sensible a los datos ruidosos y a los valores atípicos, ya que el hecho de centrarse en los casos mal clasificados puede llevar al sobreajuste.
Principalmente reduce la varianza promediando las predicciones de múltiples modelos, por lo que es eficaz para modelos con alta varianza.	Aborda tanto el sesgo como la varianza, centrándose en reducir el sesgo corrigiendo secuencialmente los errores cometidos por los aprendices débiles.

Las Técnicas de Bagging y Boosting en el Análisis de Datos Educativos

Las técnicas de Bagging y Boosting pueden ser utilizadas para mejorar la precisión y la robustez de los modelos predictivos en el análisis de datos educativos. Ambas son técnicas que combinan múltiples modelos para crear un modelo más potente y preciso. Si se quiere predecir el rendimiento académico de los estudiantes (v.g. calificación final de un examen o un curso), basado en diversas características como horas de estudio, asistencia a clases, participación en actividades extracurriculares, entre otras, la técnica de Bagging, puede entrenar varios árboles de decisión en diferentes subconjuntos de los datos y luego promediar sus predicciones para obtener una estimación más robusta y precisa del rendimiento académico. Por otro lado, con los mismos datos y usando la técnica de Boosting, podemos empezar con un modelo simple (como un árbol de decisión de una o dos bifurcaciones), y luego agregar modelos adicionales que se centren en los errores cometidos por el modelo inicial. Este enfoque secuencial puede llevar a una mejora significativa en la precisión de las predicciones del rendimiento académico.

En síntesis, en el contexto educativo estas técnicas permiten trabajar con grandes volúmenes de datos y son útiles para la predicción del rendimiento estudiantil, al permitir crear modelos que pueda captar las complejas interacciones entre diferentes factores que afectan el rendimiento académico. Por otro lado, es posible identificar los factores más significativos que influyen en el éxito académico, permitiendo a los educadores diseñar mejores estrategias de enseñanza. Ambas técnicas son poderosas herramientas en el análisis de datos educativos, permitiendo a los investigadores y educadores crear modelos predictivos más precisos y robustos.