Tema 2: Aprendizaje supervisado y no supervisado

En el campo del aprendizaje automático, el entendimiento de los enfoques supervisados y no supervisados juega un papel crucial. Estos dos enfoques nos permiten abordar gran cantidad de problemas del mundo real, desde la clasificación de observaciones hasta el agrupamiento en diferentes categorías. Hay múltiples aplicaciones del Machine Learning, y cada una de ellas precisa de un modelo que puede clasificarse en una de estas dos categorías.

El aprendizaje supervisado, implica el entrenamiento de algoritmos utilizando datos previamente etiquetados, es decir datos de los que conocemos el valor de la variable dependiente cuando entrenamos el modelo. Los problemas que podemos resolver con estos modelos pueden ser muy diversos, desde clasificar imágenes en categorías específicas hasta predecir el precio de una casa en función de sus características. Dentro del aprendizaje supervisado, existen dos tareas principales: la regresión, donde el objetivo es predecir valores continuos, como el precio de una acción o la temperatura, y la clasificación, que busca asignar una etiqueta o clase a cada instancia de datos, como identificar si un correo electrónico es spam o no.

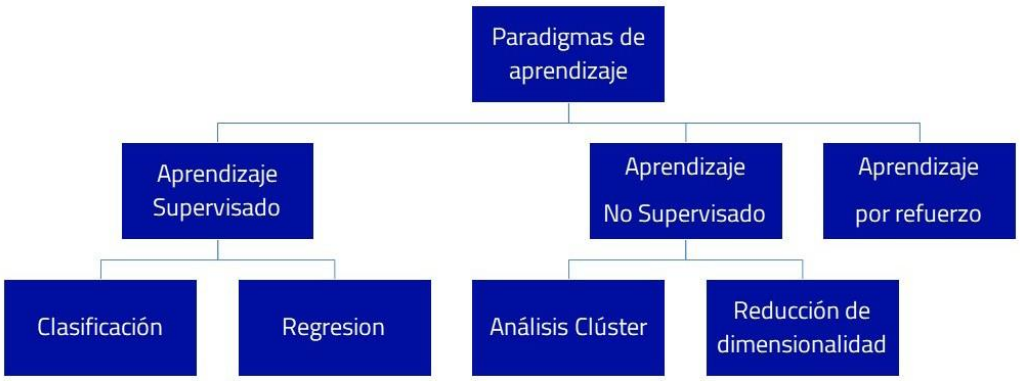
Por otro lado, el aprendizaje no supervisado se centra en el análisis de datos sin etiquetar. Aquí, los algoritmos exploran la estructura de los datos en busca de patrones y relaciones ocultas, sin el uso de etiquetas externas. Uno de los enfoques más destacados dentro del aprendizaje no supervisado es el clustering, donde los datos se agrupan en conjuntos basados en similitudes, lo que permite descubrir agrupaciones naturales dentro de los datos. Otro enfoque importante es la reducción de dimensionalidad, que busca representar los datos en un espacio de menor dimensión, conservando la información relevante y facilitando su interpretación.

En este tema, exploraremos en detalle tanto el aprendizaje supervisado como el no supervisado, examinando sus conceptos fundamentales, algoritmos clave y aplicaciones prácticas en una variedad de campos. A lo largo de este capítulo, descubriremos cómo estos enfoques pueden ser utilizados de manera efectiva para abordar desafíos del mundo real, brindando nuevas perspectivas y posibilidades en el análisis y comprensión de los datos.

Conceptos y diferencias

En este apartado, exploraremos algunos de los paradigmas clásicos de aprendizaje ML, destacando sus principios fundamentales, características distintivas y ejemplos representativos de algoritmos asociados. Desde el aprendizaje supervisado hasta el aprendizaje por refuerzo, cada paradigma ofrece un enfoque único para abordar diferentes tipos de problemas y desafíos en el campo del aprendizaje automático.

Profundizaremos en la naturaleza de estos paradigmas, su aplicación en diversos contextos y las consideraciones clave que influyen en la elección del enfoque más adecuado para un problema específico. Al comprender estos fundamentos, estarás preparado para poder aprovechar al máximo las herramientas y técnicas disponibles para desarrollar modelos eficaces y generar insights valiosos a partir de los datos.



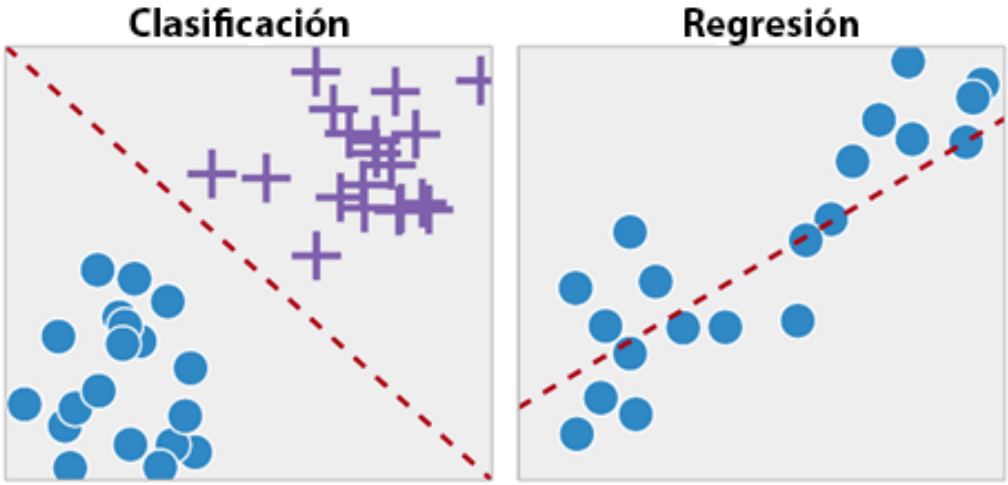
Aprendizaje supervisado

El aprendizaje supervisado representa uno de los pilares fundamentales dentro del campo del aprendizaje automático, donde se utilizan conjuntos de datos etiquetados para entrenar algoritmos con el fin de que aprendan a realizar predicciones o clasificaciones precisas.

Este enfoque se distingue por la presencia de datos etiquetados, lo que significa que cada instancia en el conjunto de datos de entrenamiento está acompañada por una etiqueta que indica la salida deseada o correcta. Dentro del aprendizaje supervisado, dos tipos principales de problemas destacan: clasificación y regresión, cada uno con sus propias características y aplicaciones distintivas.

En el caso de la clasificación, el objetivo es asignar una etiqueta o categoría a cada instancia de entrada. Por ejemplo, consideremos un conjunto de imágenes de diferentes animales. Un algoritmo de clasificación entrenado podría identificar y etiquetar cada imagen como "perro", "gato", "pájaro", etc., basándose en los patrones que encuentra en los datos.

Por otro lado, en la regresión, el objetivo consiste en predecir un valor numérico como salida. Por ejemplo, si tenemos datos que relacionan la temperatura con la cantidad de ventas de helados, un algoritmo de regresión podría predecir cuántas unidades de helado se venderán a una temperatura específica, utilizando modelos matemáticos que se ajustan a los datos observados.



Aquí tienes algunos ejemplos de aplicaciones de modelos de aprendizaje supervisado:

1. Sistemas de recomendación: Los modelos de aprendizaje supervisado se utilizan ampliamente en sistemas de recomendación, como los que se encuentran en plataformas de streaming de música, películas y series. Estos modelos predicen qué contenido es más probable que le guste a un usuario en función de su historial de preferencias y comportamiento de interacción.

2. Clasificación de imágenes médicas: En medicina, los modelos de aprendizaje supervisado se emplean para clasificar imágenes médicas, como radiografías, tomografías computarizadas (TC) y resonancias magnéticas (RM), para ayudar en el diagnóstico de enfermedades y condiciones médicas, como la detección temprana de cáncer.

3. Automatización de procesos de atención al cliente: En el servicio al cliente, los modelos de aprendizaje supervisado pueden clasificar automáticamente las consultas y preguntas de los clientes en diferentes categorías, dirigiendo eficientemente las solicitudes a los departamentos correspondientes o proporcionando respuestas automatizadas.

4. Predicción de precios de mercado: En el ámbito financiero, los modelos de aprendizaje supervisado pueden predecir los precios de mercado de acciones, materias primas, criptomonedas y otros activos financieros utilizando datos históricos y otros factores relevantes.

5. Clasificación de fraudes financieros*: Los modelos de aprendizaje supervisado se utilizan para detectar fraudes financieros, como transacciones fraudulentas con tarjetas de crédito, lavado de dinero y actividades ilegales en el mercado de valores, al identificar patrones anómalos en los datos financieros.

6. Diagnóstico de enfermedades: Además de las imágenes médicas, los modelos de aprendizaje supervisado se utilizan en el diagnóstico de enfermedades a partir de datos clínicos, como síntomas, historial médico y resultados de pruebas de laboratorio, ayudando a los médicos a tomar decisiones informadas sobre el tratamiento.

7. Clasificación de spam y filtrado de correo electrónico: Los modelos de aprendizaje supervisado se aplican para identificar y filtrar automáticamente correos electrónicos no deseados o spam, mejorando la eficiencia y la seguridad de las comunicaciones por correo electrónico.

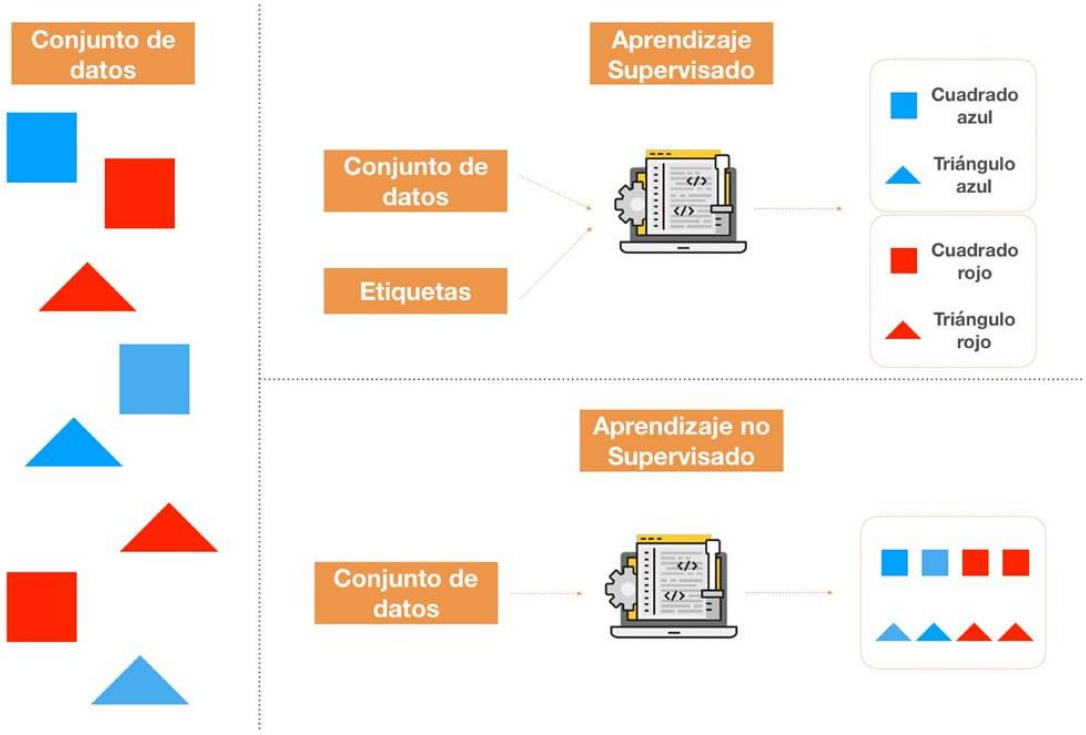
8. Identificación de objetos en imágenes: En sistemas de visión por computadora, los modelos de aprendizaje supervisado se utilizan para identificar y clasificar objetos en imágenes, como personas, animales, vehículos y edificios, en aplicaciones de seguridad, vigilancia y reconocimiento de imágenes.

Estos ejemplos ilustran la diversidad de aplicaciones y el impacto significativo que los modelos de aprendizaje supervisado tienen en una variedad de industrias y campos de aplicación.

En resumen, el aprendizaje supervisado es una herramienta poderosa que se utiliza en una amplia gama de campos para tomar decisiones, predecir resultados y descubrir patrones en los datos. La capacidad de utilizar datos etiquetados para entrenar algoritmos y aprender a partir de ellos es esencial para numerosas aplicaciones del mundo real, impulsando avances significativos en áreas como la medicina, la seguridad cibernética, la detección de fraudes y más.

Aprendizaje no supervisado

El aprendizaje no supervisado se distingue de los modelos de aprendizaje supervisado por su enfoque en la exploración y descubrimiento de patrones intrínsecos en los datos, sin el uso explícito de respuestas o etiquetas. Aquí, el algoritmo se enfrenta únicamente a las características de los datos, sin información sobre las categorías a las que pertenecen. En lugar de buscar predicciones específicas, el objetivo principal es comprender la estructura subyacente de los datos y agruparlos en sus respectivos conjuntos.



Una herramienta fundamental en el aprendizaje no supervisado es el algoritmo de análisis de clúster, también conocido como análisis de conglomerados. Este método estadístico busca agrupar elementos o variables con características similares, buscando maximizar la homogeneidad dentro de cada grupo y la diferenciación entre ellos. Por ejemplo, en el campo del marketing, este enfoque podría ayudar a identificar segmentos de clientes con comportamientos de compra similares.

Otro enfoque común son los algoritmos de reducción de dimensionalidad, que buscan crear abstracciones al reducir el número de variables en un conjunto de datos. Esto se hace por varias razones, como identificar y eliminar variables irrelevantes, mejorar el rendimiento computacional y simplificar la comprensión del modelo y sus resultados. Por ejemplo, en biología, este enfoque podría ayudar a simplificar la descripción de datos genéticos complejos.

Además, existen los algoritmos de asociación, que se enfocan en descubrir patrones entre las características de los datos. Este enfoque es crucial en aplicaciones como el análisis de transacciones en seguros, donde se pueden encontrar asociaciones entre diferentes tipos de riesgos o reclamaciones.

Algunas de las aplicaciones de los modelos de aprendizaje no supervisado incluyen:

1. Segmentación de mercado: Identificación de grupos homogéneos de clientes basados en comportamientos de compra, preferencias o características demográficas, sin la necesidad de etiquetas predefinidas.

2. Análisis de redes sociales: Descubrimiento de comunidades o grupos de usuarios con intereses similares en redes sociales, lo que puede ayudar en la personalización de contenido y publicidad dirigida.

3. Detección de anomalías: Identificación de patrones inusuales o anómalos en datos, como transacciones financieras fraudulentas, comportamientos de usuarios sospechosos o fallos en sistemas industriales.

4. Compresión de datos: Reducción del tamaño de conjuntos de datos manteniendo la mayor cantidad posible de información relevante, lo que puede mejorar la eficiencia del almacenamiento y la transmisión de datos.

5. Análisis de imágenes médicas: Agrupación de imágenes médicas, como radiografías, tomografías computarizadas (TC) o resonancias magnéticas (RM), para identificar patrones comunes entre diferentes tipos de enfermedades o condiciones.

6. Clasificación automática de documentos: Agrupación de documentos similares en función de su contenido, lo que puede facilitar la organización y búsqueda eficiente de información en grandes repositorios de datos.

7. Reconocimiento de patrones en datos genéticos: Identificación de secuencias genéticas similares o grupos de genes que tienden a co-ocurrir juntos, lo que puede ayudar en la comprensión de la genética y la biología molecular.

8. Análisis de datos de tráfico: Identificación de patrones de tráfico en redes de transporte, como congestiones, rutas frecuentes o comportamientos de conducción anormales, para mejorar la planificación urbana y la seguridad vial.

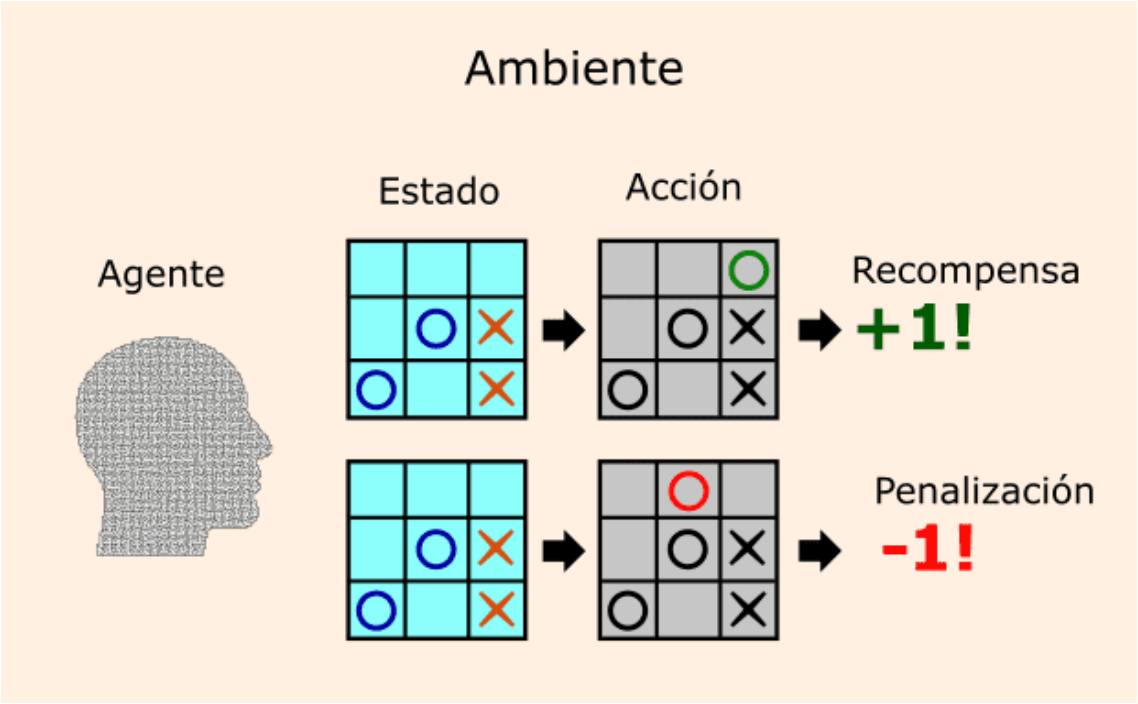
9. Segmentación de audiencia en marketing digital: Agrupación de usuarios en función de su comportamiento en línea, como patrones de navegación, interacciones con contenido o preferencias de productos, para campañas de marketing más efectivas y personalizadas.

Estas aplicaciones ilustran cómo los modelos de aprendizaje no supervisado pueden utilizarse en una variedad de campos para descubrir patrones, estructuras y conocimientos útiles en conjuntos de datos sin etiquetas previas.

Aprendizaje por refuerzo

El aprendizaje por refuerzo es una rama del aprendizaje automático que se inspira en la psicología conductual y se centra en el aprendizaje a través de la interacción de un agente con un entorno. A diferencia de los modelos de aprendizaje supervisado y no supervisado que hemos comentado previamente, donde el modelo recibe datos etiquetados o no etiquetados, en el aprendizaje por refuerzo, el modelo aprende a través de la retroalimentación en forma de recompensas o castigos.

En el aprendizaje por refuerzo, el agente toma decisiones secuenciales en un entorno dinámico con el objetivo de maximizar una recompensa acumulada a largo plazo. Cada acción que el agente toma tiene una consecuencia en el estado futuro del entorno, y el agente aprende a través de la experiencia, ensayando y cometiendo errores.



El proceso típico de aprendizaje por refuerzo incluye los siguientes elementos:

- **Agente**: Es la entidad que aprende y toma decisiones en el entorno.
- **Entorno**: Es el mundo en el que el agente interactúa y toma decisiones.
- **Acciones**: Son las decisiones que el agente puede tomar en cada paso de tiempo.
- **Estado**: Es la representación del entorno en un momento dado, que incluye toda la información relevante para tomar decisiones.
- **Recompensas**: Son señales de retroalimentación que indican al agente qué tan bien está actuando en el entorno. El objetivo del agente es maximizar la recompensa acumulada a lo largo del tiempo.

Algunos casos de uso reales del aprendizaje por refuerzo incluyen:

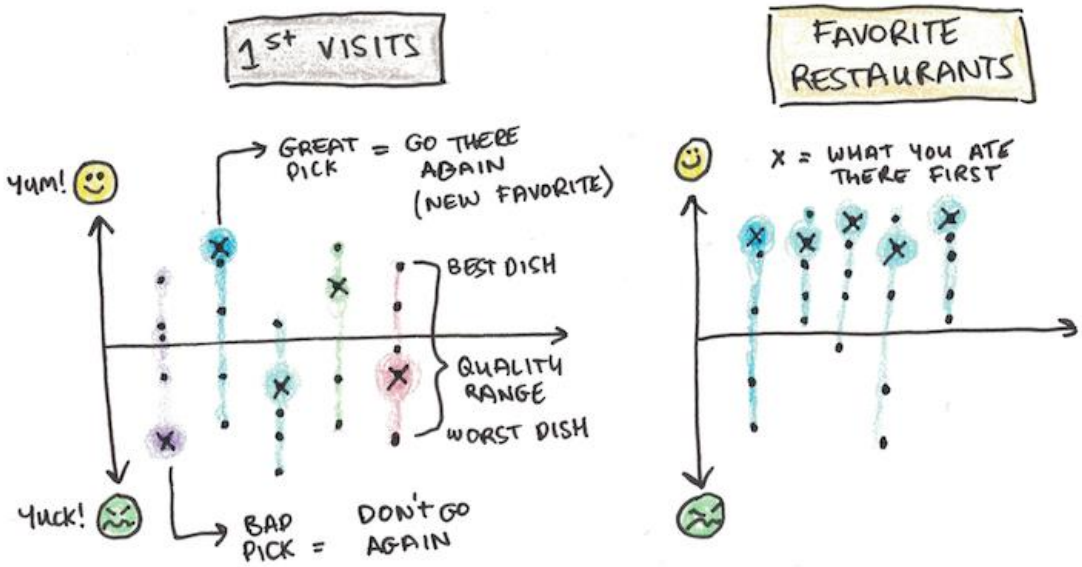
1. Juegos: El aprendizaje por refuerzo ha sido aplicado con éxito en juegos como el ajedrez, Go y videojuegos, donde el agente aprende a través de la experiencia y la retroalimentación del juego para mejorar su rendimiento y alcanzar un nivel de maestría.
2. Robótica: En robótica, el aprendizaje por refuerzo se utiliza para entrenar robots para realizar tareas complejas, como la manipulación de objetos, la navegación autónoma o el aprendizaje de habilidades motoras.
3. Control de procesos: En entornos industriales, el aprendizaje por refuerzo se emplea para optimizar el control de procesos, como la producción de energía, la gestión de inventarios o la optimización de la cadena de suministro.
4. Publicidad en línea: En marketing digital, el aprendizaje por refuerzo se utiliza para personalizar y optimizar las estrategias de publicidad en línea, como la selección de anuncios y el ajuste de ofertas, para maximizar la tasa de clics o las conversiones.

Estos ejemplos muestran cómo el aprendizaje por refuerzo se aplica en una amplia variedad de campos para tomar decisiones óptimas en entornos dinámicos y complejos. A medida que avanza la investigación en este campo, se esperan aún más aplicaciones y avances en el futuro.

Estos agentes se plantean en muchas ocasiones cuestiones sobre cual es la mejor decisión que pueden tomar, la exploración de nuevas acciones, o la explotación de las ya conocidas. El equilibrio entre exploración y explotación es un concepto fundamental en el aprendizaje por refuerzo y se refiere al delicado balance entre dos estrategias opuestas pero complementarias que un agente debe emplear para maximizar sus recompensas a largo plazo en un entorno dinámico.

La exploración implica probar nuevas acciones o estrategias con el fin de descubrir información sobre el entorno que aún no se ha aprendido. Es fundamental en las etapas iniciales del aprendizaje, cuando el agente tiene poco conocimiento sobre el entorno o cuando se enfrenta a situaciones nuevas o desconocidas. La exploración permite al agente recopilar datos y aprender sobre las consecuencias de diferentes acciones, lo que puede llevar a descubrir estrategias más efectivas y a evitar quedarse atrapado en un óptimo local subóptimo.

La explotación implica aprovechar el conocimiento existente y seleccionar las acciones que se sabe que tienen una alta probabilidad de conducir a recompensas positivas en función de la experiencia previa. Es crucial en etapas avanzadas del aprendizaje, cuando el agente ha acumulado suficiente conocimiento sobre el entorno y las acciones que maximizan las recompensas. La explotación permite al agente aprovechar eficientemente su experiencia pasada y tomar decisiones óptimas basadas en lo que ya se sabe.



El desafío radica en encontrar el equilibrio adecuado entre exploración y explotación. Si el agente se centra demasiado en la exploración, puede desperdiciar oportunidades para obtener recompensas inmediatas, mientras que, si se centra demasiado en la explotación, puede perder la oportunidad de descubrir estrategias aún mejores o adaptarse a cambios en el entorno.

Algunas estrategias comunes para lograr este equilibrio incluyen:

- **Epsilon-greedy**: Esta estrategia consiste en seleccionar la mejor acción conocida con una probabilidad de 1-epsilon y seleccionar una acción aleatoria (explorar) con una probabilidad de epsilon.
- **Softmax**: En esta estrategia, las acciones se seleccionan de acuerdo con su valor de utilidad estimado, pero con una probabilidad ponderada que permite cierta exploración.
- **Upper Confidence Bound** (UCB): Esta estrategia asigna valores de confianza a cada acción y selecciona acciones con una alta incertidumbre (exploración) o un alto valor de utilidad (explotación).

El equilibrio entre exploración y explotación es esencial para el éxito del aprendizaje por refuerzo, ya que permite al agente adaptarse eficazmente a entornos dinámicos y maximizar sus recompensas a largo plazo. En la práctica, encontrar el equilibrio óptimo puede requerir un ajuste cuidadoso de los parámetros y el diseño de estrategias de exploración y explotación que se adapten a las características específicas del entorno y del problema que se está abordando.

Aprendizaje Semi-Supervisado

El aprendizaje semisupervisado es un enfoque en el aprendizaje automático que combina elementos del aprendizaje supervisado y no supervisado. En este paradigma, el modelo se entrena utilizando un conjunto de datos que contiene tanto ejemplos etiquetados como no etiquetados. La idea es utilizar la información proporcionada por los datos no etiquetados para mejorar el rendimiento del modelo, que de otra manera se limitaría al conjunto de datos etiquetados más pequeño.

La razón principal para utilizar el aprendizaje semisupervisado es que, en muchos casos, es costoso o difícil obtener grandes cantidades de datos etiquetados. Los datos no etiquetados, sin embargo, son a menudo más fáciles de adquirir o generar en grandes cantidades. Al aprovechar esta información adicional, el aprendizaje semisupervisado puede mejorar la generalización y la capacidad predictiva del modelo.

Existen varios enfoques y técnicas para el aprendizaje semisupervisado:

1. Co-entrenamiento: Este enfoque consiste en entrenar múltiples modelos independientes, cada uno utilizando una porción diferente de los datos no etiquetados. Los modelos luego se combinan para hacer predicciones en nuevos datos, aprovechando la información que cada modelo ha aprendido de manera independiente.

2. Propagación de etiquetas: En este enfoque, las etiquetas conocidas se propagan a los datos no etiquetados en función de la similitud entre ellos. Por ejemplo, si dos puntos de datos tienen características similares, pero uno está etiquetado y el otro no, se puede suponer que el segundo punto de datos también debería tener la misma etiqueta.

3. Modelos generativos: Los modelos generativos, como las redes generativas adversariales (GAN) o los modelos de mezcla de Gaussianas, se pueden utilizar en el aprendizaje semisupervisado para generar datos sintéticos que se asemejen a los datos no etiquetados. Estos datos sintéticos pueden ayudar al modelo a aprender representaciones más robustas de los datos y mejorar su capacidad de generalización.

Las aplicaciones del aprendizaje semisupervisado son diversas y se encuentran en una amplia gama de campos, incluyendo el procesamiento del lenguaje natural, la visión por computadora, la biología computacional y más.

Por ejemplo, en el procesamiento del lenguaje natural, el aprendizaje semisupervisado se utiliza para mejorar la clasificación de texto mediante el uso de grandes cantidades de datos no etiquetados, como corpus de texto sin procesar. En la visión por computadora, puede ayudar en la clasificación de imágenes utilizando datos no etiquetados para aprender características genéricas que son útiles para una variedad de tareas de clasificación.

Algoritmos supervisados (regresión, clasificación)

Los algoritmos supervisados son una categoría fundamental en el campo del aprendizaje automático, donde el modelo se entrena utilizando datos etiquetados, es decir, datos que tienen una respuesta conocida o salida deseada. Dentro de los algoritmos supervisados, dos de las técnicas más comunes son la regresión y la clasificación.

La regresión se utiliza cuando la variable de salida es continua. El objetivo es predecir un valor numérico en función de una o más variables de entrada.

Algunos ejemplos de algoritmos de regresión incluyen:

- Regresión lineal: Se ajusta una línea recta a los datos para modelar la relación entre las variables de entrada y la variable de salida.

- Regresión polinómica: Se ajusta un polinomio a los datos para modelar relaciones más complejas que una línea recta.

- Regresión de vecinos más cercanos (KNN): Predice el valor de salida basándose en los valores de salida de los k ejemplos más cercanos en el espacio de características.

- Regresión de árboles de decisión: Utiliza árboles de decisión para dividir el espacio de características en regiones y predecir un valor numérico para cada región.

En esta sección vamos a centrarnos en el algoritmo de la regresión lineal

Por otro lado, la clasificación se utiliza cuando la variable de salida es discreta o categórica. El objetivo es asignar una etiqueta de clase a una instancia de entrada en función de sus características.

Algunos ejemplos de algoritmos de clasificación incluyen:

- Regresión logística: Se utiliza para problemas de clasificación binaria y asigna una probabilidad a cada clase.

- Máquinas de vectores de soporte (SVM): Busca un hiperplano que separe las clases en el espacio de características.

- Árboles de decisión: Divide el espacio de características en regiones utilizando decisiones en forma de árbol.

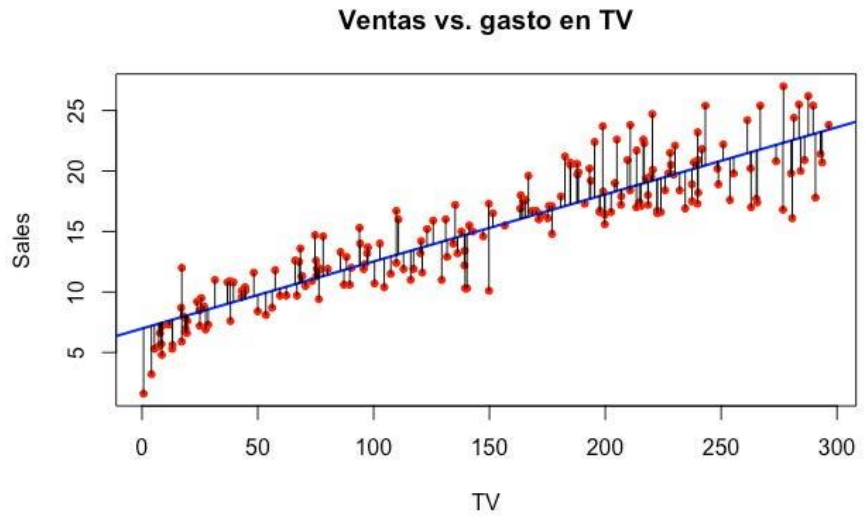
- Bosques aleatorios: Consiste en un conjunto de árboles de decisión que votan por la clase más popular para una instancia dada.

- Redes neuronales: Modelan relaciones complejas entre las características utilizando capas de neuronas interconectadas.

En esta sección cubriremos muchos de estos algoritmos utilizados en una amplia gama de problemas en diversas áreas, como finanzas, medicina, marketing, visión por computadora, procesamiento del lenguaje natural y más. La elección del algoritmo adecuado depende del tipo de datos, la naturaleza del problema y las características específicas de la aplicación.

Regresión

El modelo de regresión lineal es uno de los métodos más simples y ampliamente utilizados en el análisis estadístico y el aprendizaje automático para modelar la relación entre una o más variables independientes (predictoras) y una variable dependiente (objetivo) continua. La idea fundamental detrás de la regresión lineal es encontrar la mejor línea recta que se ajuste a los datos, de modo que pueda predecir la variable dependiente en función de las variables independientes.



El modelo de regresión lineal se puede expresar matemáticamente de la siguiente manera:

$$y^i = \beta_0 + \beta_1 x_1^i + \beta_2 x_2^i + \ldots + \beta_k x_k^i + \epsilon^i$$

Donde:

- y es la variable dependiente que estamos tratando de predecir.
- $x_1, x_2, \ldots, x_n$ son las variables independientes o características que utilizamos para hacer la predicción.
- $\beta_0, \beta_1, \ldots, \beta_n$ son los coeficientes de regresión que representan la pendiente de la línea para cada variable independiente, y β_0 es el intercepto.
- ϵ es el término de error, que representa la diferencia entre la predicción del modelo y el valor real de y.

El objetivo del modelo de regresión lineal es estimar los valores de los coeficientes que minimicen la suma de los cuadrados de los errores (método de mínimos cuadrados), es decir, que minimicen la diferencia entre las predicciones del modelo y los valores reales en los datos de entrenamiento.

Para ajustar el modelo de regresión lineal, se utilizan técnicas como el método de los mínimos cuadrados ordinarios (OLS) o algoritmos de optimización más avanzados. Una vez que el modelo está ajustado, se pueden utilizar los coeficientes de regresión estimados para hacer predicciones sobre nuevos datos.

Es importante destacar que el modelo de regresión lineal hace algunas suposiciones, como la linealidad y la independencia de los errores. Si estas suposiciones no se cumplen, pueden surgir problemas en la interpretación y precisión de las predicciones del modelo. Además, la regresión lineal es más adecuada para relaciones lineales simples entre las variables. Si la relación es más compleja, pueden ser necesarios modelos más avanzados.

El coeficiente de determinación, comúnmente conocido como R cuadrado (R^2), es una medida estadística que proporciona información sobre la proporción de la variabilidad de una variable dependiente que es explicada por las variables independientes en un modelo de regresión. En otras palabras, indica qué tan bien se ajustan los valores observados a los valores predichos por el modelo.

El R cuadrado se calcula como la proporción de la varianza explicada por el modelo con respecto a la varianza total de la variable dependiente. Matemáticamente, se define como:

$$R^2 = 1 - \frac{SSE}{SST}$$

Donde:

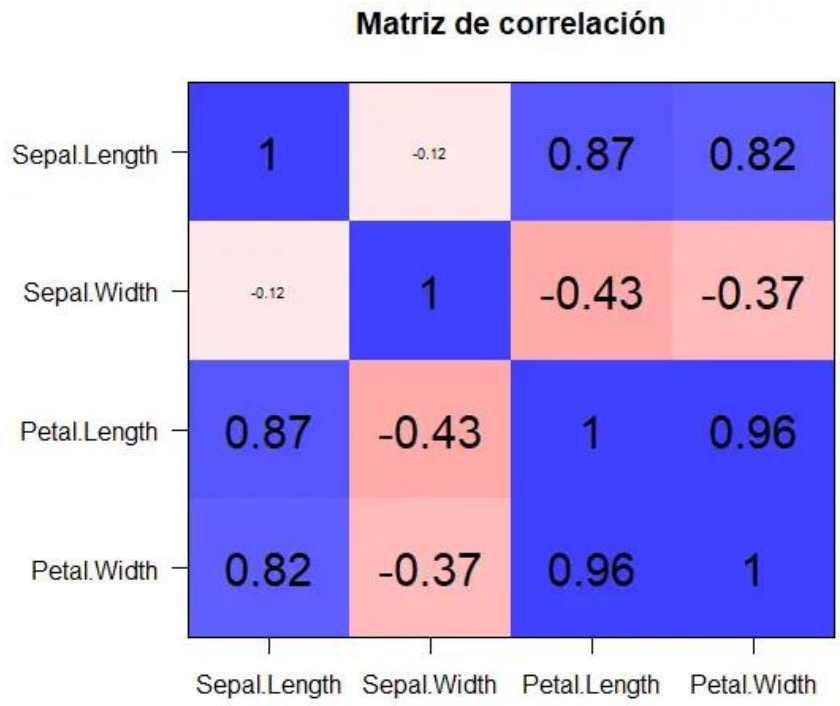
- SSE es la suma de los cuadrados de los residuos o errores del modelo. Es la diferencia entre los valores observados y los valores predichos por el modelo.
- SST es la suma total de los cuadrados, que mide la variabilidad total de la variable dependiente. Se calcula como la suma de los cuadrados de las diferencias entre los valores observados y la media de la variable dependiente.

El valor de R cuadrado puede variar de 0 a 1. Cuanto más cercano a 1 sea el valor de R cuadrado, mejor se ajusta el modelo a los datos, lo que significa que una mayor proporción de la variabilidad en la variable dependiente es explicada por el modelo. Un valor de R cuadrado cercano a 0 indica que el modelo no explica bien la variabilidad de la variable dependiente y que podría haber otras variables importantes que no se están teniendo en cuenta en el modelo.

Es importante tener en cuenta que el R cuadrado no indica la validez del modelo en sí mismo, sino más bien cuánta varianza en la variable dependiente es explicada por las variables independientes incluidas en el modelo. Por lo tanto, es crucial interpretar el R cuadrado en conjunto con otras métricas y considerar el contexto específico del problema y los datos.

Para construir mejores modelos es importante fijarse en la correlación, un estadístico que nos va a describir la relación entre dos variables. Se utiliza para determinar si y cómo cambian juntas dos variables diferentes. En otras palabras, la correlación indica cómo se relacionan la variación de una variable con la variación de otra variable.

Una matriz de correlaciones es una tabla que muestra las correlaciones entre todas las posibles combinaciones de pares de variables en un conjunto de datos. En esta matriz, cada fila y cada columna corresponden a una variable en el conjunto de datos, y cada celda muestra el coeficiente de correlación entre las dos variables correspondientes.



La correlación lineal se utiliza para medir la fuerza y la dirección de la relación lineal entre dos variables continuas. El coeficiente de correlación de Pearson es la medida más común para la correlación lineal. Este coeficiente varía entre -1 y 1, donde:

- 1 indica una correlación positiva perfecta (ambas variables aumentan juntas).
- -1 indica una correlación negativa perfecta (una variable aumenta mientras que la otra disminuye).
- 0 indica ausencia de correlación lineal.

Es importante tener en cuenta que la correlación lineal no implica causalidad. Solo porque dos variables están correlacionadas no significa que una cause la otra.

Seleccionar variables que estén correlacionadas con la variable dependiente pero no entre sí es fundamental en el proceso de construcción de modelos predictivos sólidos.

Si las variables independientes están altamente correlacionadas entre sí (fenómeno conocido como multicolinealidad), puede ser difícil para el modelo distinguir el efecto único de cada variable en la variable dependiente. Esto puede llevar a estimaciones poco fiables de los coeficientes del modelo y a interpretaciones erróneas sobre la importancia de cada variable.

Al incluir variables independientes que están correlacionadas con la variable dependiente pero no entre sí, se maximiza la capacidad del modelo para explicar la variabilidad en la variable dependiente. Cada variable puede contribuir de manera única a la predicción del resultado, lo que resulta en un modelo más robusto y preciso.

Además, al evitar la multicolinealidad, se facilita la interpretación de los coeficientes del modelo. Esto permite comprender mejor cómo cada variable independiente influye en la variable dependiente sin que sus efectos se vean distorsionados por la presencia de otras variables altamente correlacionadas.

Revisa el siguiente notebook para tener una noción completa de como implementar modelos de regresión lineal utilizando Python y Scikit Learn: https://colab.research.google.com/drive/1GpOPUX2m4XdICTIsQaDMLLeYTq7yqUI4?usp=sharing

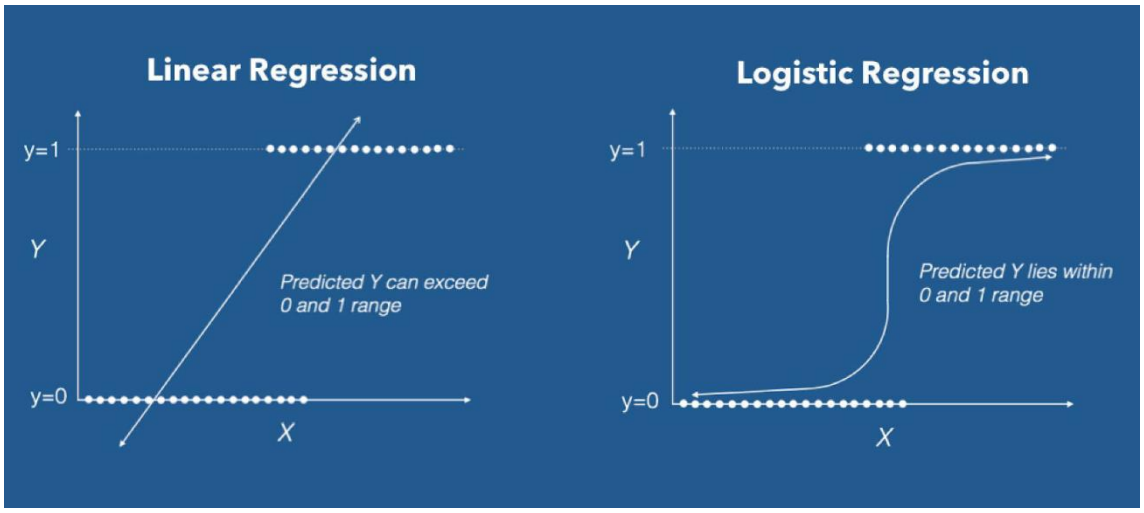
Clasificación

Los modelos de clasificación desempeñan un papel fundamental al permitirnos comprender y predecir patrones en datos para asignar categorías a nuevas instancias. Desde la detección de spam en correos electrónicos hasta la identificación de enfermedades en imágenes médicas, los modelos de clasificación abarcan una amplia gama de aplicaciones que impactan significativamente en nuestra vida cotidiana.

En este apartado, exploraremos los modelos de clasificación en aprendizaje automático. Revisaremos desde los conceptos fundamentales hasta los algoritmos más avanzados, comprendiendo cómo funcionan, cuándo utilizarlos y cómo evaluar su desempeño.

El primer modelo que vamos a presentar es el de regresión logística. El modelo de regresión logística es un método utilizado para modelar la relación entre una variable dependiente categórica (binaria o multinomial) y un conjunto de variables independientes. Aunque el nombre incluye "regresión", la regresión logística se utiliza principalmente para problemas de clasificación.

La regresión logística utiliza la función logística o sigmoide para modelar la probabilidad de que una observación pertenezca a una determinada categoría. La función sigmoide es una curva en forma de "S" que tiene la siguiente forma:



La función sigmoide tiene la propiedad de transformar cualquier valor de entrada en un valor en el rango de 0 a 1. Esto es útil en el contexto de la regresión logística, ya que nos permite interpretar la salida como una probabilidad.

La función sigmoide se utiliza para modelar la probabilidad de que una observación pertenezca a la categoría de interés (por ejemplo, "sí" o "no", "positivo" o "negativo"). Una vez hechas las predicciones con este o cualquier otro modelo de clasificación lo mas habitual es construir lo que se conoce como una matriz de confusión, con el objetivo de analizar los resultados.

Las matrices de confusión son una herramienta fundamental en la evaluación del rendimiento de modelos de clasificación. Permiten visualizar el desempeño del modelo al comparar las predicciones con los valores reales de las observaciones. En una matriz de confusión, las predicciones del modelo se organizan en función de cuatro posibles resultados:

- Verdaderos positivos (True Positives, TP): Son los casos en los que el modelo predijo correctamente la clase positiva (la clase de interés) y la observación real también pertenece a esa clase.

- Falsos positivos (False Positives, FP): Estos casos ocurren cuando el modelo predice incorrectamente la clase positiva, es decir, clasifica erróneamente una observación negativa como positiva.

- Verdaderos negativos (True Negatives, TN): Son los casos en los que el modelo predice correctamente la clase negativa y la observación real también pertenece a esa clase.

- Falsos negativos (False Negatives, FN): Estos casos suceden cuando el modelo predice incorrectamente la clase negativa, es decir, clasifica erróneamente una observación positiva como negativa.

Con esta información, podemos construir una matriz de confusión como la siguiente:

<table>
<tr><td colspan="2" rowspan="2">Matriz de confusión</td><td colspan="2">Estimado por el modelo</td><td rowspan="2" colspan="2"></td></tr>
<tr><td>Negativo (N)</td><td>Positivo (P)</td></tr>
<tr><td rowspan="2">Real</td><td>Negativo</td><td>a: (TN)</td><td>b: (FP)</td><td></td><td></td></tr>
<tr><td>Positivo</td><td>c: (FN)</td><td>d: (TP)</td><td>Precisión
(“precision”)
Porcentaje predicciones positivas correctas:</td><td>d/(b+d)</td></tr>
<tr><td colspan="2" rowspan="2"></td><td>Sensibilidad, exhaustividad
(“Recall</i”)
Porcentaje casos positivos detectados</td><td>Especifidad
(Specifity)
Porcentaje casos negativos detectados</td><td colspan="2">Exactitud
(“accuracy”)
Porcentaje de predicciones correctas
(No sirve en datasets poco equilibrados)</td></tr>
<tr><td>d/(d+c)</td><td>a/(a+b)</td><td colspan="2">(a+d)/(a+b+c+d)</td></tr>
</table>

A partir de la matriz de confusión, podemos calcular diversas métricas de evaluación del modelo, que incluyen:

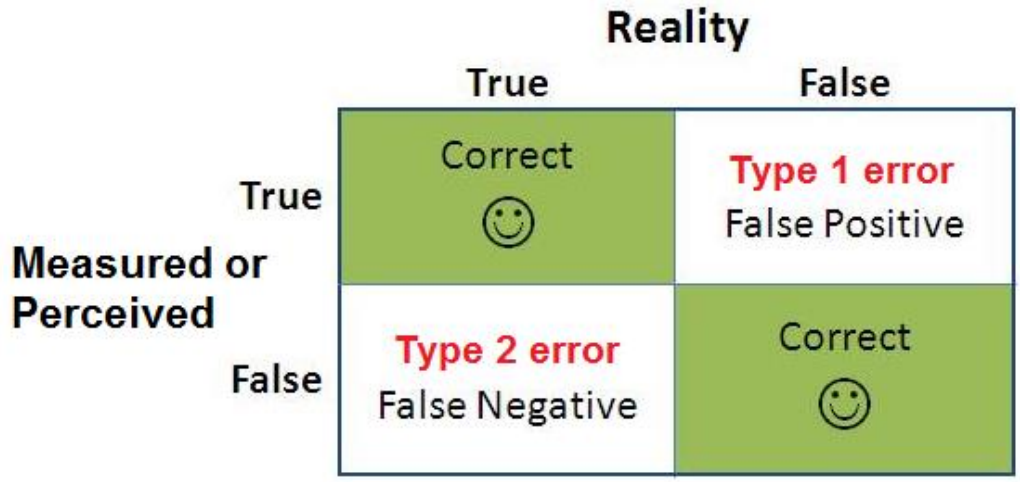
- **Precisión** (Accuracy): Proporción de predicciones correctas. La exactitud proporciona una visión general de cuán bien el modelo clasifica las observaciones, pero puede ser engañosa en situaciones donde las clases están desbalanceadas.

- **Sensibilidad** (Recall o True Positive Rate): Proporción de casos positivos reales que fueron identificados correctamente por el modelo. El recall es importante cuando es crucial identificar todos los casos positivos, como en problemas médicos donde la detección temprana de una enfermedad es fundamental.

- **Especificidad** (Specificity o True Negative Rate): Proporción de casos negativos reales que fueron identificados correctamente por el modelo.

- **Valor predictivo positivo** (Precision o Positive Predictive Value): Proporción de predicciones positivas correctas entre todas las predicciones positivas realizadas por el modelo. La precisión es importante cuando el coste de los falsos positivos es alto, ya que se enfoca en la calidad de las predicciones positivas.

Estas métricas son fundamentales para evaluar el rendimiento de modelos de clasificación y ayudan a comprender su comportamiento en diferentes situaciones. Es importante considerarlas en conjunto y no de forma aislada, ya que cada una proporciona información valiosa sobre aspectos específicos del rendimiento del modelo.



Los diferentes tipos de errores se relacionan con los valores de la matriz de confusión:

- **Error tipo I**: Ocurre cuando se clasifica erróneamente una observación negativa como positiva, es decir, un falso positivo (FP).

- **Error tipo II**: Sucede cuando se clasifica erróneamente una observación positiva como negativa, es decir, un falso negativo (FN).

La comprensión de estos conceptos y métricas es crucial para evaluar adecuadamente el rendimiento de un modelo de clasificación y determinar su idoneidad para una tarea específica.

Uno de los hiperparametros que debemos definir en cualquier modelo de clasificación es el threshold. El threshold (umbral) es un concepto utilizado en modelos de clasificación para tomar decisiones sobre cómo asignar las observaciones a las diferentes clases. En un modelo de clasificación binaria, el threshold se utiliza para determinar el punto de corte en la salida del modelo, que separa las predicciones en las clases positiva y negativa.

La salida de un modelo de clasificación binaria generalmente consiste en una probabilidad o puntaje que representa la confianza del modelo en que una observación pertenece a la clase positiva. El threshold se aplica a esta salida para decidir si una observación se clasifica como positiva o negativa.

Por ejemplo, si el threshold se establece en 0.5, todas las observaciones con una probabilidad mayor o igual a 0.5 se clasificarán como positivas, mientras que aquellas con una probabilidad menor que 0.5 se clasificarán como negativas.

Modificar el threshold puede influir en el equilibrio entre la sensibilidad y la especificidad del modelo. Por ejemplo:

- Si se reduce el threshold, se clasificarán más observaciones como positivas, lo que aumentará la sensibilidad del modelo, pero posiblemente disminuirá su especificidad.

- Si se aumenta el threshold, se clasificarán menos observaciones como positivas, lo que aumentará la especificidad del modelo, pero posiblemente disminuirá su sensibilidad.

La elección del threshold adecuado depende del contexto del problema y de los objetivos específicos del modelado. Por ejemplo, en un problema médico donde es crucial identificar todos los casos positivos (alta sensibilidad), podría ser preferible utilizar un threshold más bajo, aunque esto pueda resultar en más falsos positivos. Por otro lado, en aplicaciones donde es crucial minimizar los falsos positivos (alta especificidad), se podría preferir un threshold más alto, aunque esto pueda resultar en una menor sensibilidad.

En resumen, el threshold es un parámetro importante en modelos de clasificación binaria que determina cómo se traducen las salidas del modelo en decisiones sobre las clases de las observaciones. Ajustar este threshold puede ser necesario para alcanzar un equilibrio óptimo entre sensibilidad y especificidad o para cumplir con los requisitos específicos del problema.

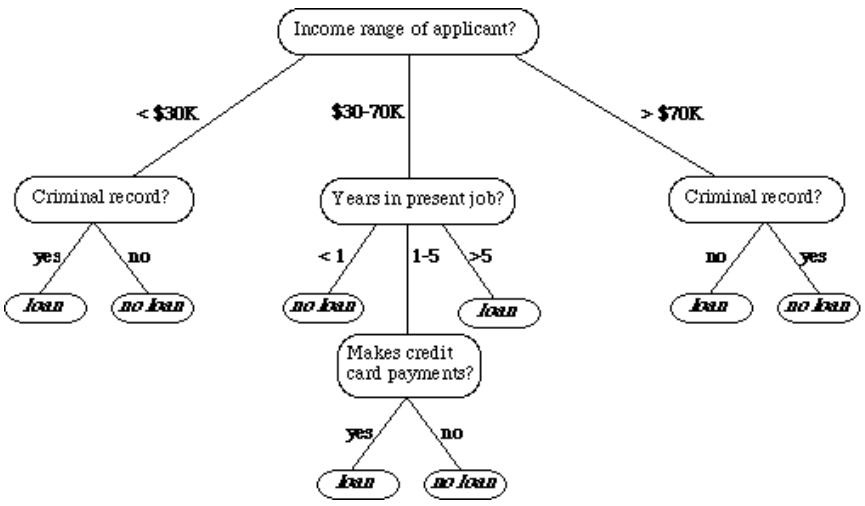
Modificar el threshold en un modelo de clasificación puede tener un impacto significativo en cómo se toman las decisiones de clasificación. Al ajustar el threshold, se pueden cambiar las tasas de falsos positivos y falsos negativos, lo que a su vez afecta la precisión, el recall y otras métricas de evaluación del modelo.

Para variar el threshold, primero necesitas entender cómo se relaciona con las predicciones del modelo. En un modelo de clasificación binaria, el threshold determina el punto de corte en la probabilidad de que una observación pertenezca a la clase positiva. Por lo tanto, si la probabilidad predicha por el modelo es mayor que el threshold, la observación se clasificará como positiva; de lo contrario, se clasificará como negativa.

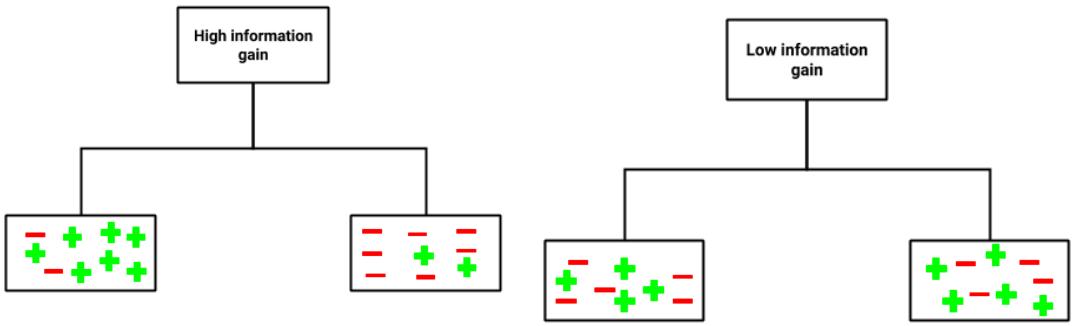
Revisa el siguiente notebook para tener una noción completa de como implementar modelos de regresión logística utilizando Python y Scikit Learn: https://colab.research.google.com/drive/1pm99BZMk0aB1yK_7V8P7ojf6C-JWSRLR?usp=sharing

El segundo modelo que vamos a presentar referente a la clasificación es el de los árboles de decisión. El modelo de árboles de decisión es un algoritmo de aprendizaje supervisado utilizado tanto para problemas de clasificación como de regresión. Es una técnica intuitiva y fácil de entender que se basa en la construcción de un árbol de decisiones mediante la partición recursiva del espacio de características.

El proceso comienza con un nodo raíz que contiene todas las observaciones de entrenamiento. En cada nodo, el algoritmo selecciona una característica y un punto de corte que divide el conjunto de datos en dos subconjuntos más puros. Este proceso se repite recursivamente en cada subconjunto hasta que se cumple algún criterio de parada, como alcanzar una profundidad máxima, no tener suficientes observaciones en un nodo o no mejorar la pureza del subconjunto.



Para decidir qué característica y punto de corte utilizar en cada nodo, el algoritmo de árbol de decisión utiliza un criterio de división, como el índice Gini o la ganancia de información (entropía). El índice Gini mide la impureza de un conjunto de observaciones y se utiliza para minimizar la mezcla de clases en los nodos del árbol. La ganancia de información mide cuánta información proporciona una característica al reducir la incertidumbre sobre la clase de una observación.



En el contexto de árboles de decisión, tanto el índice Gini como la entropía se utilizan para determinar la mejor división de un conjunto de datos en subconjuntos más homogéneos. El algoritmo de árbol de decisión seleccionará la división que minimice la impureza de los subconjuntos resultantes, ya sea utilizando el índice Gini o la entropía como criterio de división. Ambas medidas son efectivas y se pueden usar indistintamente, aunque pueden producir resultados ligeramente diferentes.

Después de construir el árbol completo, es posible que esté sobreajustado a los datos de entrenamiento. Para evitar el sobreajuste, se puede podar el árbol, eliminando nodos internos que no mejoran significativamente la precisión del modelo en un conjunto de datos de validación.

Para hacer predicciones con un árbol de decisión, una nueva observación se pasa a través del árbol de acuerdo con las reglas de división hasta que alcanza una hoja, que corresponde a una clase (en clasificación) o a un valor (en regresión).

Los árboles de decisión tienen varias ventajas, como su capacidad para manejar datos categóricos y numéricos, su facilidad de interpretación y su capacidad para capturar relaciones no lineales en los datos. Sin embargo, también tienen algunas limitaciones, como su tendencia al sobreajuste, especialmente en conjuntos de datos con alta dimensionalidad.

Para mejorar el rendimiento y la generalización de los árboles de decisión, se pueden utilizar técnicas como la poda, la selección de características y el ensamblaje de árboles (por ejemplo, bosques aleatorios o gradient boosting). En general, los árboles de decisión son una herramienta versátil y poderosa en el conjunto de técnicas de aprendizaje automático.

Revisa el siguiente notebook para tener una noción completa de como implementar modelos de regresión logística utilizando Python y Scikit Learn: https://colab.research.google.com/drive/1pm99BZMk0aB1yK_7V8P7ojf6C-JWSRLR?usp=sharing

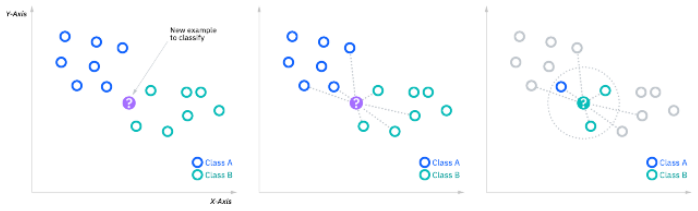
Para continuar, vamos a hablar acerca del modelo de k-Nearest Neighbors (k-Vecinos Más Cercanos o k-NN). Este es un algoritmo de aprendizaje supervisado utilizado tanto para problemas de clasificación como de regresión. Es un método simple pero poderoso que se basa en la idea de que objetos similares tienden a estar en la misma clase o tener valores similares en el espacio de características.

El primer paso del algoritmo consiste en almacenar todos los datos de entrenamiento. El parámetro k es un número entero positivo que representa el número de vecinos más cercanos que se tomarán en cuenta para realizar una predicción. Este valor se selecciona antes de aplicar el algoritmo y puede ajustarse mediante técnicas de validación cruzada.

Para hacer una predicción para un punto de datos desconocido, el algoritmo calcula la distancia entre este punto y todos los puntos de datos en el conjunto de entrenamiento. La distancia puede calcularse utilizando diversas métricas, como la distancia euclidiana o la distancia de Manhattan.

Una vez calculadas las distancias, se seleccionan los k puntos de datos más cercanos al punto de datos desconocido. Estos puntos se denominan "vecinos más cercanos".

Para problemas de clasificación, la predicción se realiza asignando la clase más común entre los k vecinos más cercanos al punto de datos desconocido. Para problemas de regresión, la predicción se realiza calculando el promedio de los valores de la variable objetivo de los k vecinos más cercanos.



Una característica clave del algoritmo k-NN es que no requiere una fase de entrenamiento explícita. En cambio, todas las operaciones de computación se realizan durante la fase de predicción. Esto hace que k-NN sea un algoritmo de aprendizaje flexible y fácil de implementar.

Sin embargo, k-NN también tiene algunas limitaciones, como la sensibilidad al ruido y la necesidad de almacenar todos los datos de entrenamiento, lo que puede hacer que sea computacionalmente costoso en conjuntos de datos grandes.

Revisa el siguiente notebook para tener una noción completa de como implementar modelos de KNN utilizando Python y Scikit Learn: https://colab.research.google.com/drive/1CV8QSvfUotZYONy0azVvOOBDYLNHhBHs?usp=sharing

Algoritmos no supervisados (clustering, reducción de dimensionalidad)

El aprendizaje no supervisado es una rama del aprendizaje automático en la que el algoritmo se enfrenta a datos que no están etiquetados y tiene la tarea de encontrar patrones, estructuras o relaciones intrínsecas en los datos sin la guía de una variable objetivo-específica. A diferencia del aprendizaje supervisado, donde el modelo se entrena utilizando ejemplos etiquetados, en el aprendizaje no supervisado, el algoritmo debe inferir la estructura subyacente de los datos por sí mismo.

Dentro del aprendizaje no supervisado, dos tipos de modelos ampliamente utilizados son el clustering (agrupamiento) y el análisis de componentes principales (PCA). Estas técnicas son fundamentales para explorar y comprender la estructura subyacente de los datos cuando no se dispone de etiquetas o información sobre las salidas deseadas.

El agrupamiento es una de las tareas más comunes en el aprendizaje no supervisado, donde el objetivo es dividir un conjunto de datos en grupos o clústeres basados en la similitud entre las observaciones. Algunos algoritmos de agrupamiento populares incluyen K-Means, Agrupamiento Jerárquico, DBSCAN y Mean Shift.

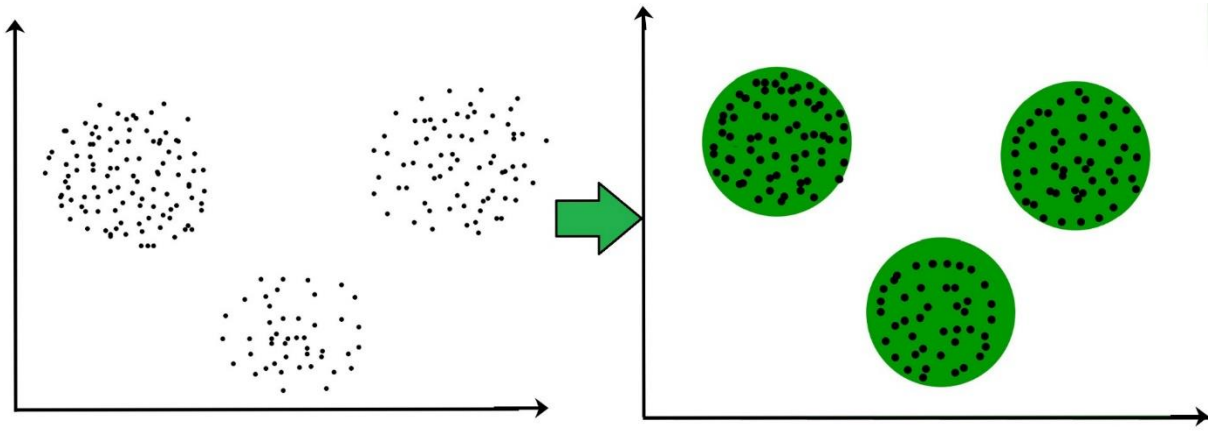
Por otro lado, la reducción de dimensionalidad es otra técnica importante en el aprendizaje no supervisado, que consiste en transformar un conjunto de datos de alta dimensionalidad en un conjunto de datos de menor dimensionalidad mientras se conserva la mayor cantidad posible de información relevante.

Algunos métodos comunes de reducción de dimensionalidad son Análisis de Componentes Principales (PCA), T-distributed Stochastic Neighbor Embedding (t-SNE) y Autoencoders.

El aprendizaje no supervisado es útil en situaciones donde los datos no están etiquetados o cuando se desea explorar y descubrir patrones o estructuras intrínsecas en los datos. Es una herramienta poderosa para la exploración de datos, la segmentación de clientes, la recomendación de productos, la detección de anomalías y más. Sin embargo, la interpretación de los resultados en el aprendizaje no supervisado puede ser más desafiante que en el aprendizaje supervisado debido a la falta de etiquetas explícitas.

El clustering, también conocido como agrupamiento, es una técnica de aprendizaje no supervisado que se utiliza para dividir un conjunto de datos en grupos o clústeres, donde los elementos dentro de cada grupo son más similares entre sí que con los elementos de otros grupos.

El objetivo principal del clustering es identificar estructuras ocultas o patrones intrínsecos en los datos sin la necesidad de etiquetas predefinidas.



Aquí hay algunas aplicaciones comunes del clustering:

- El clustering se utiliza ampliamente en marketing para segmentar clientes en grupos homogéneos con características y comportamientos similares. Esto permite a las empresas personalizar sus estrategias de marketing y ofrecer productos o servicios específicos a cada segmento.

- En la minería de textos y el análisis de texto, el clustering se utiliza para agrupar documentos similares basados en el contenido. Esto es útil en la organización y la búsqueda de grandes colecciones de documentos, así como en la detección de temas emergentes o tendencias en el texto.

- En sistemas de recomendación, el clustering se utiliza para agrupar usuarios con preferencias similares o productos con características similares. Esto permite recomendar contenido relevante a los usuarios basado en los patrones de comportamiento o en las similitudes entre los elementos.

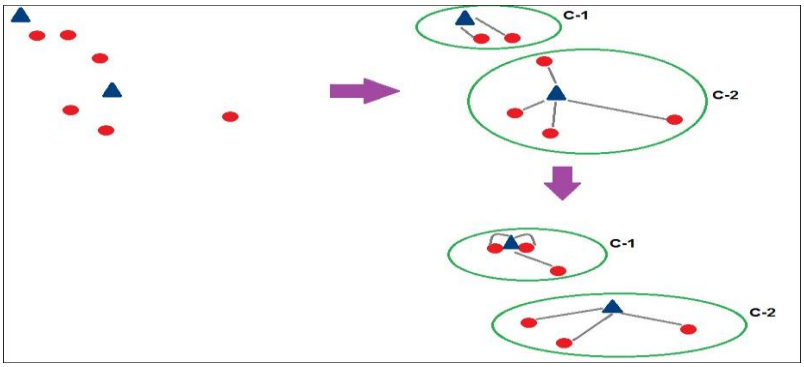
- En biología, el clustering se utiliza para agrupar genes, proteínas o secuencias de ADN similares con el fin de identificar patrones genéticos, relaciones filogenéticas o funciones biológicas comunes.

- En análisis espacial, el clustering se utiliza para dividir áreas geográficas en regiones similares en función de características como la densidad de población, el ingreso per cápita, la infraestructura, etc. Esto es útil en la planificación urbana, el marketing local y la gestión de recursos naturales.

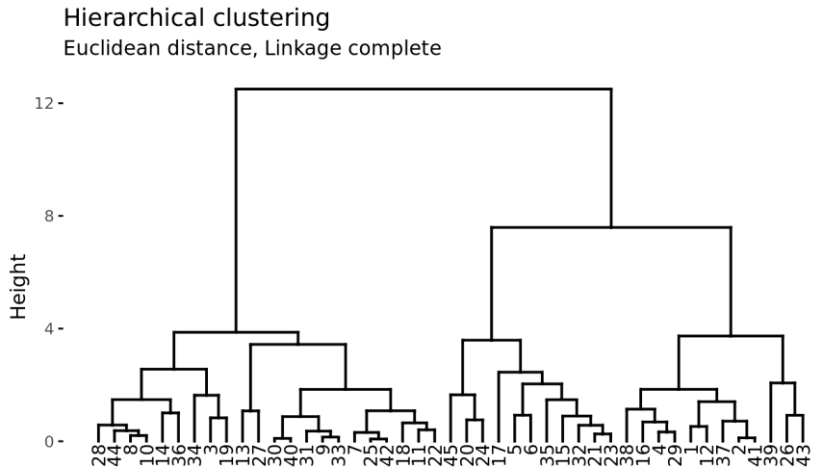
Estas son solo algunas de las muchas aplicaciones del clustering en una amplia variedad de campos. En general, el clustering es una herramienta versátil que permite descubrir patrones interesantes y obtener información valiosa a partir de datos no etiquetados.

Existen varios modelos de clustering, cada uno con sus propias características, suposiciones y aplicaciones. Vamos a presentar tres de ellos para posteriormente revisar uno con más profundidad.

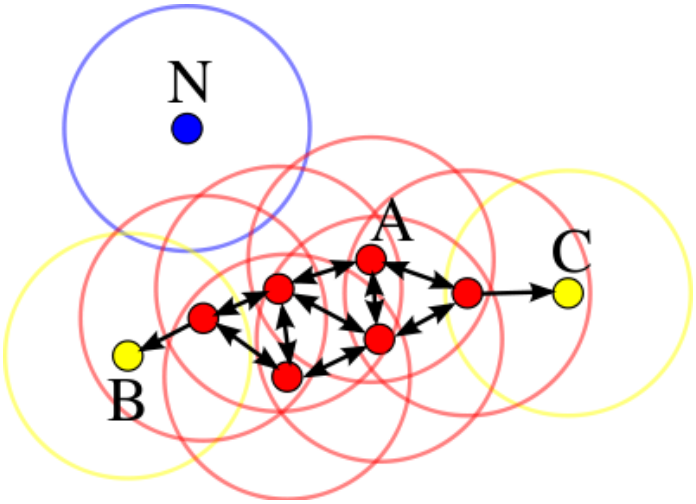
K-Means es uno de los algoritmos de clustering más populares y simples. Divide los datos en k grupos predefinidos basados en la distancia euclidiana entre los puntos de datos y los centroides de los clústeres. Es eficiente computacionalmente y funciona bien en conjuntos de datos grandes, pero requiere especificar el número de clústeres k de antemano.



Por otro lado, el clustering jerárquico construye una jerarquía de clústeres, donde los clústeres se agrupan de manera recursiva hasta formar un único clúster que contiene todos los datos. Puede ser aglomerativo (comenzando con cada punto de datos como un clúster y fusionando clústeres) o divisivo (comenzando con todos los datos en un clúster y dividiendo en subclústeres). No requiere especificar el número de clústeres de antemano y proporciona una representación visual de la estructura de los datos.



Por último, DBSCAN es un algoritmo de clustering basado en densidad que puede identificar clústeres de cualquier forma y tamaño en el espacio de características. Divide los datos en clústeres basados en la densidad de los puntos de datos: regiones densas se consideran parte de un mismo clúster, mientras que regiones menos densas se consideran como ruido o outliers. Es robusto a la presencia de ruido y no requiere especificar el número de clústeres de antemano.



Estos son solo algunos ejemplos de modelos de clustering. Cada modelo tiene sus propias ventajas, desventajas y suposiciones, por lo que la elección del modelo adecuado depende del tipo de datos, la estructura del problema y los objetivos de la aplicación. Es importante explorar y comparar diferentes modelos para encontrar el más adecuado para una tarea de clustering específica. Veamos a continuación el método de K-Means de manera más detallada.

El algoritmo K-Means (k-medias) es uno de los métodos de clustering más utilizados y efectivos. Funciona dividiendo un conjunto de datos en k clústeres, donde cada clúster está representado por su centroide, que es el punto medio de todos los puntos asignados a ese clúster.

El algoritmo comienza seleccionando aleatoriamente k puntos como centroides iniciales. Estos centroides pueden ser puntos de datos seleccionados al azar del conjunto de datos o pueden ser elegidos utilizando algún método heurístico.

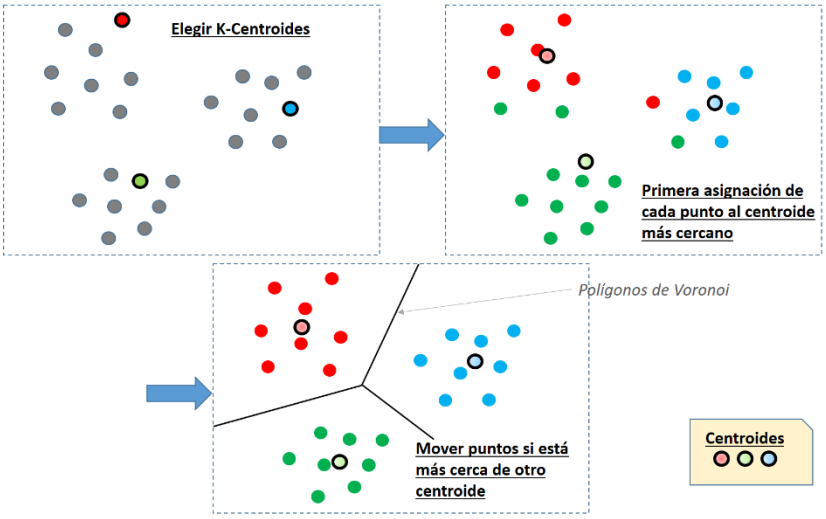
Para cada punto de datos en el conjunto de datos, se calcula la distancia entre el punto y todos los centroides. El punto se asigna al clúster cuyo centroide está más cercano (generalmente utilizando la distancia euclidiana).

Después de asignar todos los puntos a los clústeres, se recalcula el centroide de cada clúster como el promedio de todos los puntos asignados a ese clúster.

Este paso reubica los centroides hacia el centro de los puntos asignados, lo que significa que los centroides se actualizan para reflejar mejor la posición del clúster.

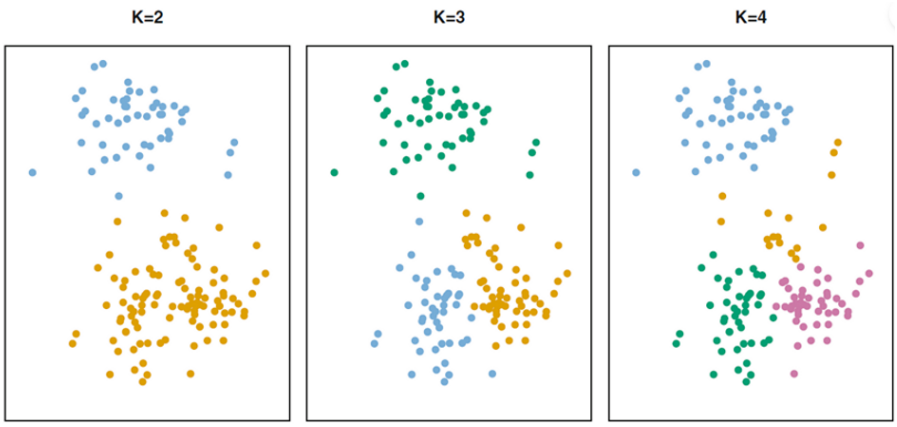
Los dos pasos anteriores se repiten iterativamente hasta que se cumple algún criterio de detención, como un número máximo de iteraciones o cuando los centroides dejan de cambiar significativamente entre iteraciones.

Cuando el algoritmo converge, los centroides se estabilizan y ya no cambian entre iteraciones. En este punto, se considera que el algoritmo ha convergido y los clústeres resultantes están formados.



Es importante tener en cuenta que el algoritmo K-Means puede converger a un mínimo local, lo que significa que la calidad de los clústeres resultantes puede depender de la inicialización de los centroides. Por esta razón, a menudo se ejecuta el algoritmo varias veces con diferentes inicializaciones y se selecciona el mejor resultado en términos de la función objetivo (por ejemplo, la suma de las distancias cuadradas de cada punto al centroide de su clúster).

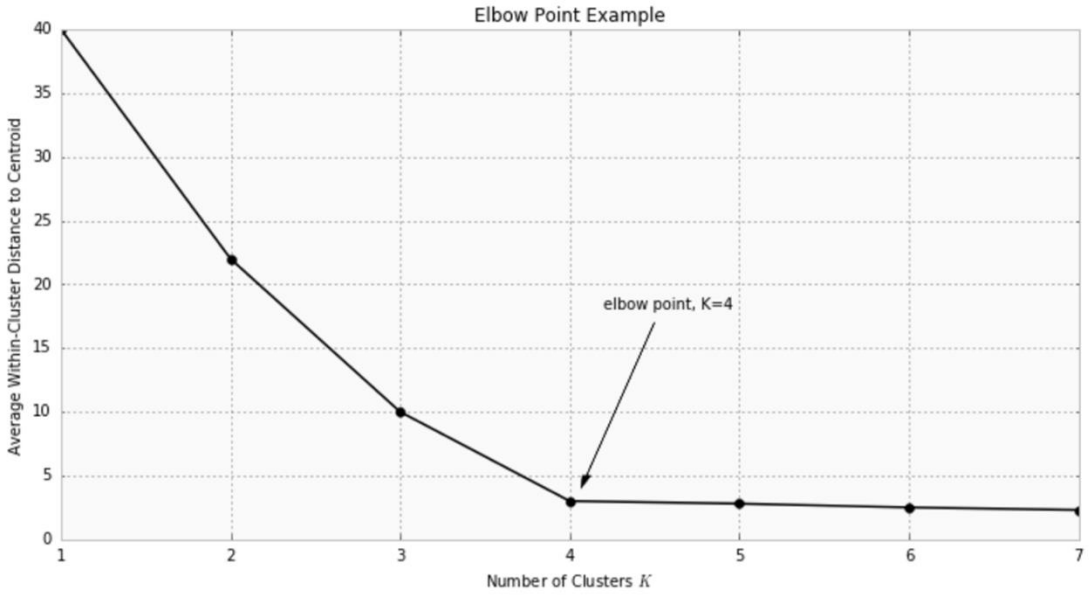
El algoritmo K-Means es rápido y escalable, lo que lo hace adecuado para grandes conjuntos de datos. Es eficaz para encontrar clústeres de forma esférica o globular, pero puede tener dificultades con clústeres de forma irregular o de densidad variable. Además, el número de clústeres k debe especificarse de antemano, lo que puede ser un desafío en algunos casos.



La técnica del codo (elbow method) y el coeficiente de silhouette son dos métricas comunes utilizadas para determinar el número óptimo de clústeres k en el algoritmo K-Means y otros métodos de clustering.

La técnica del codo es un método gráfico que ayuda a seleccionar el número óptimo de clústeres observando la variación de la suma de las distancias cuadradas intra-cluster (la suma de las distancias al cuadrado de cada punto al centroide de su clúster) en función del número de clústeres.

Se realiza el clustering para diferentes valores de k, y se calcula la suma de las distancias cuadradas intra-cluster (inertia) para cada valor de k. Luego, se traza un gráfico de la suma de las distancias cuadradas intra-cluster en función de k, y se busca el punto donde la curva forma un "codo" o una inflexión significativa. Este punto es considerado como el número óptimo de clústeres.

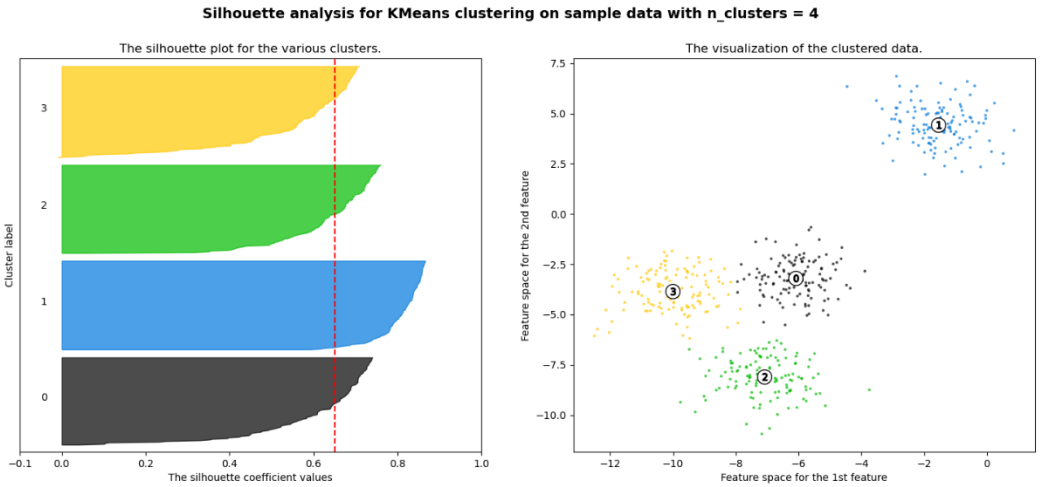


Por otro lado, el coeficiente de silhouette es una medida de la cohesión intra-cluster y la separación inter-cluster.

Para cada punto de datos, se calcula el coeficiente de silhouette, que es la diferencia entre la distancia media al resto de puntos en el mismo clúster (a) y la distancia media al punto más cercano en un clúster diferente (b), dividido por el máximo de a y b.

El coeficiente de silhouette para un conjunto de datos es el promedio de los coeficientes de silhouette de todos los puntos de datos. El valor del coeficiente de silhouette varía entre -1 y 1. Un valor alto indica que el punto está bien clasificado, mientras que un valor bajo indica que el punto podría estar mal clasificado. Un valor cercano a 0 indica que el punto está cerca del límite entre dos clústeres.

El número óptimo de clústeres se puede seleccionar maximizando el coeficiente de silhouette para diferentes valores de k.



Ambas técnicas son útiles para determinar el número adecuado de clústeres, pero es importante tener en cuenta que ninguna de ellas es perfecta y pueden producir resultados diferentes en algunos casos. Por lo tanto, se recomienda utilizar ambas métricas y otros métodos de validación cruzada para tomar una decisión informada sobre el número óptimo de clústeres para un conjunto de datos específico.

Revisa el siguiente notebook para tener una noción completa de como implementar modelos de árboles de decisión y KMeans utilizando Python y Scikit Learn:
https://colab.research.google.com/drive/12TgbWOBo9QQiksNlB7o8GXmhpkZ4wpe9?usp=sharing

Otra de las aplicaciones de los modelos de aprendizaje no supervisado es la reducción de la dimensionalidad. En ese sentido, es importante mencionar el PCA, una técnica muy utilizada y con múltiples aplicaciones.

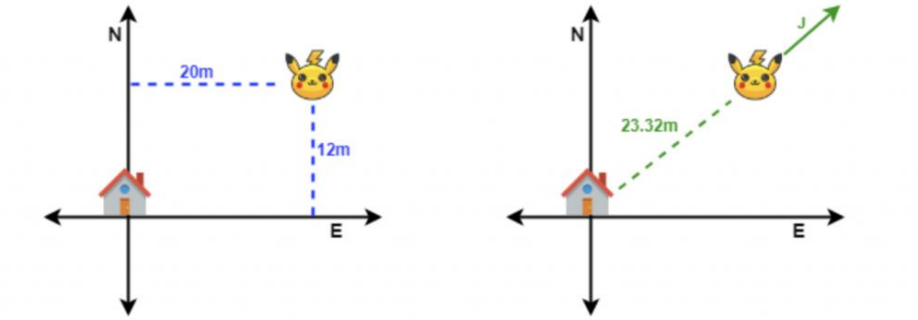
El Análisis de Componentes Principales (PCA por sus siglas en inglés) es una técnica de reducción de dimensionalidad utilizada para simplificar conjuntos de datos complejos, conservando la mayor cantidad posible de información importante. Funciona transformando los datos originales en un nuevo conjunto de variables (llamadas componentes principales) que son combinaciones lineales de las variables originales, de manera que estas nuevas variables capturen la mayor variabilidad posible en los datos.

El funcionamiento es el siguiente. En primer lugar, se centran los datos restando la media de cada variable. Esto es importante para eliminar cualquier sesgo o efecto de escala en los datos. A continuación, se calcula la matriz de covarianza de los datos centrados. Esta matriz describe la relación entre las diferentes variables y su variabilidad conjunta.

Después de calcular la matriz de covarianza, se realiza la descomposición en valores propios (eigendecomposition) de esta matriz. Los valores propios representan la cantidad de varianza explicada por cada componente principal, y los vectores propios correspondientes representan la dirección de máxima variabilidad en los datos.

Se ordenan los valores propios de mayor a menor y se seleccionan los primeros k componentes principales que explican la mayor parte de la varianza en los datos. Por lo general, se eligen los componentes principales que explican una cantidad significativa de varianza, como el 90% o el 95% del total.

Finalmente, se proyectan los datos originales en el espacio de los componentes principales seleccionados. Esto implica calcular las coordenadas de cada observación en términos de los nuevos componentes principales.



Los principales usos del PCA incluyen:

- Reducir la cantidad de variables en conjuntos de datos con muchas características (variables) correlacionadas, manteniendo al mismo tiempo la mayor parte de la variabilidad en los datos.

- Visualizar datos de alta dimensionalidad en un espacio de menor dimensión (generalmente 2D o 3D), lo que facilita la identificación de patrones o estructuras en los datos.

- Preprocesar los datos antes de aplicar otros algoritmos de aprendizaje automático, ya que puede mejorar la eficiencia y el rendimiento de los modelos al reducir la dimensionalidad de los datos.

- Eliminar la multicolinealidad entre las variables, lo que puede mejorar la interpretación de los modelos de regresión y reducir el riesgo de sobreajuste.

PCA es una herramienta poderosa y versátil que se utiliza ampliamente en diferentes campos, incluyendo ciencia de datos, análisis de datos, bioinformática, reconocimiento de patrones, entre otros. Ayuda a simplificar y comprender conjuntos de datos complejos al extraer las características más importantes que explican la variabilidad en los datos.

Revisa el siguiente notebook para tener una noción completa de como implementar PCA utilizando Python y Scikit Learn: https://colab.research.google.com/drive/1YaZ1vDJbY9sh7MkxhkkCX3rx7fbib2hL?usp=sharing