Clasificación

En el presente Módulo se abordarán más técnicas de clasificación avanzadas, entre las cuales están:

- Regresión logística
- Naive Bayes
- Árboles de decisión
- Reglas de clasificación

Regresión logística

La regresión lineal estima los valores de variables continuas, aunque no es adecuada para los valores de variables categóricas compuestas por solo dos categorías, tales como los resultados binarios; por ejemplo, si lloverá o no. En dichas situaciones, se puede utilizar la regresión logística. En la regresión logística, la variable de respuesta generalmente consiste en una variable binaria.

En lo que respecta a los valores categóricos, la estimación tiene que ver más con la probabilidad de que una instancia pertenezca a una de las dos categorías representadas por la variable categórica de respuesta (considerada una tarea de clasificación) y no con hallar el valor real de la variable de respuesta.

La regresión logística también se considera una estimación probable de clases, debido a que se encarga de estimar la probabilidad de que una instancia pertenezca a una clase particular o la ocurrencia de la clase objetivo. Generalmente, la regresión logística es utilizada en tareas de clasificación binarias, en las cuales la instancia pertenece o no a la clase objetivo.

A pesar de que es una forma de regresión ya que usa la regresión lineal, la regresión logística generalmente se emplea para efectos de clasificación, y hace uso de la regresión a fin de calcular una estimación de la probabilidad de clases. Los valores previstos por la regresión lineal están basados en una línea recta. Como esta línea puede extenderse infinitamente en dirección positiva o negativa, no existen valores máximos ni mínimos de la variable de respuesta. Por esto, no es práctico usar una línea recta para estimar probabilidades, ya que los valores de probabilidad oscilan entre un mínimo de 0 y un máximo de 1.

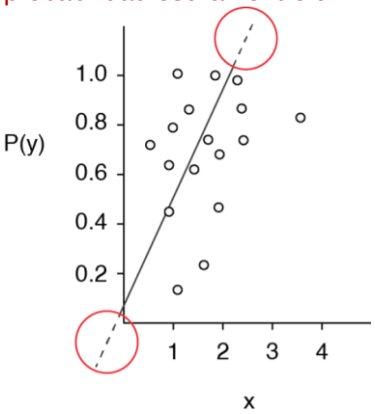

Figura 5.5 - Si estimamos probabilidades usando la regresión lineal, los valores de probabilidad superarían 1 y 0, lo cual no es válido.

Logaritmo de la probabilidad (log - odds)

Para evitar que los valores de probabilidad estén fuera de los rangos de 1 o 0 es necesaria una función matemática que transforme una línea recta en una curva. A este tipo de función se le llama función logística y está basada en el principio que toma un logaritmo natural de las probabilidades de un evento, o el logaritmo de las probabilidades

de una instancia que pertenece a una clase. Las probabilidades son la relación entre las probabilidades de que ocurra un evento y las probabilidades de que no, y oscilan entre los valores de 0 y 1.

Con respecto a la regresión logística, el cálculo de la regresión lineal proporciona el logaritmo de la probabilidad (log - odds) de una instancia que pertenece a una clase objetivo, lo cual puede ser usado posteriormente para calcular la probabilidad de la instancia que pertenece a la clase objetivo. La relación entre la probabilidad y el logaritmo de la probabilidad (log - odds) se resume en la Tabla 5.5.

Probability	Log-Odds
0.5	0.0
> 0.5	> 0.0
< 0.5	< 0.0

Tabla 5.5 - Tabla de la relación entre la probabilidad y el logaritmo de la probabilidad (log - odds).

La Figura 5.6 muestra el uso de la regresión logística frente a la regresión lineal. El uso del logaritmo de la probabilidad (log - odds) produce una curva de probabilidad cuyos valores siguen estando entre 0 y 1. La curva logística se obtiene reconvirtiendo el logaritmo de la probabilidad (log - odds) a los respectivos valores de probabilidad entre 0 y 1, y no usando los valores reales del logaritmo de la probabilidad (log - odds) que pueden adquirir cualquier valor positivo o negativo.

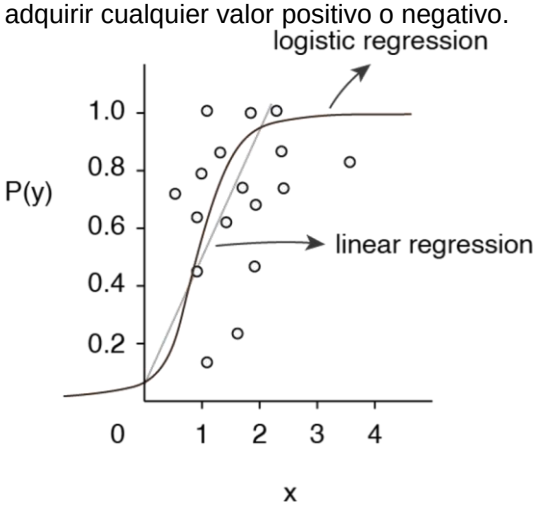

Figura 5.6 – El uso de la regresión logística frente a la regresión lineal.

Regresión logística

El resultado de un modelo de regresión logística puede ser malinterpretado fácilmente. Por ejemplo, un resultado de 0,75 solo significa que existe una mayor posibilidad de que la instancia pertenezca a la variable de respuesta, la clase objetivo. Esto no indica que este sea el valor de la variable de respuesta, tal y como es el caso de la regresión lineal.

Los valores de la variable de respuesta (los valores binarios; por ejemplo, si lloverá o no) ya se conocen. Lo que se desconoce es la probabilidad de que una instancia pertenezca o no a la clase representada por la variable de respuesta.

Naive Bayes

Naive Bayes es una técnica de clasificación basada en probabilidades que predice la asociación de clases de acuerdo con la probabilidad previamente observada de todas las características potenciales. Esta técnica se utiliza

cuando una combinación de varias características (llamadas evidencia) afecta el proceso de determinación de la clase objetivo. Debido a esta característica, Naive Bayes es capaz de tomar en cuenta las características que parecen insignificantes si son tomadas por separado, pero que si son consideradas en conjunto, pueden tener un impacto significativo sobre la probabilidad de que una instancia pertenezca a una clase en particular.
Se asume que todas las características son igualmente significativas y que el valor de una no depende del valor de ninguna otra. En otras palabras, las características son independientes entre sí. La probabilidad de asociación de clases de una instancia no mostrada, que consiste en una combinación de diferentes valores de características, podría ser calculada hallando la probabilidad de asociación de clases de una instancia de ejemplo que tenga exactamente los mismos valores de características.

Es inusual hallar una instancia que tenga exactamente el mismo conjunto de valores de características. Incluso si se encuentran coincidencias, el número de instancias debe ser significativo, con el fin de calcular de manera precisa la probabilidad de asociación de clases de una instancia no mostrada. Para superar estas dificultades, Naive Bayes combina la probabilidad observada de múltiples características de todas las instancias con ciertas etiquetas de clase en los datos de ejemplo, a fin de calcular la probabilidad de que una instancia no mostrada pertenezca a dicha clase.

En resumen, el algoritmo de Naive Bayes supone de forma simplista que todas las características tienen igual importancia y que los valores de características son independientes entre sí. Más allá predecir la asociación de clases, el algoritmo también brinda la probabilidad de asociación de clases de una clase prevista.

En términos prácticos, esta suposición simplista es a menudo errónea, ya que los valores de características frecuentemente son interdependientes. Por ejemplo, el aumento en la temperatura está relacionado con el clima soleado. Sin embargo, el algoritmo sigue mostrando un buen rendimiento y actúa con bastante rapidez. Se ocupa de manejar los datos faltantes y atípicos (outliers), ya que cada característica es considerada de manera independiente y los valores faltantes simplemente pueden ser eliminados.

La información obtenida del algoritmo de Bayes puede ser utilizada para clasificar instancias; por ejemplo, cuál cliente se cambiará a la competencia, o para seleccionar las mejores instancias entre las clases objetivo. Sin embargo, dichas estimaciones de probabilidad deberían ser usadas con prudencia a la hora de tomar decisiones, puesto que, por sí solas, proporcionan poca información acerca de los errores de falsos positivos y falsos negativos.

A menudo, el algoritmo de Bayes es utilizado para detectar el correo no deseado (spam) y para clasificar documentos. También sirve como clasificador de línea base para comparar algoritmos más complejos. Asimismo, puede ser utilizado en el aprendizaje progresivo, donde el modelo es actualizado con base en nuevos datos de ejemplo sin necesidad de regenerar todo el modelo desde el principio.

Suavizado de Laplace

El suavizado de Laplace, o suavizado aditivo, es usado para evitar obtener valores de cero o uno como la probabilidad de asociación de clases. Una probabilidad de asociación de clases de cero o uno demuestra una confianza del 100%, algo que generalmente no es posible debido a la naturaleza aleatoria de los procesos de generación de datos.

Se puede añadir un número arbitrario, por lo general uno, a los conteos de ocurrencias del dataset de ejemplo, a fin de que las probabilidades de característica de clases sean diferentes a cero. Esto garantiza numeradores y denominadores diferentes a cero en Naive Bayes. Naive Bayes generalmente utiliza características de categorías. Para trabajar con características numéricas, se pueden usar técnicas de discretización de datos, tales como el encajonamiento (binning), para crear los valores de categoría.

Árboles de decisión

Un árbol de decisión es un algoritmo de clasificación que representa un concepto en forma de un conjunto jerárquico de decisiones lógicas mediante una estructura en forma de árbol, y es utilizado para determinar el valor objetivo de una instancia. Las decisiones lógicas se toman realizando pruebas sobre los valores de características

de las instancias, de tal manera que cada prueba filtre cada vez más la instancia hasta conocer su valor objetivo o asociación de clase.

Un árbol de decisión se parece a un diagrama de flujo compuesto por nodos de decisión, que realizan pruebas sobre el valor de las características de una instancia, y nodos de hoja, también conocidos como nodos terminales, donde se determinan los valores objetivo de la instancia como resultado de un cruce de información de los nodos de decisión.

La Figura 5.7 muestra un ejemplo de un árbol de decisión. Los nodos de hoja se etiquetan según la clase de las instancias mayoritarias, cuyo valor objetivo es el mismo, junto con la probabilidad de que una instancia pertenezca a dicha clase.

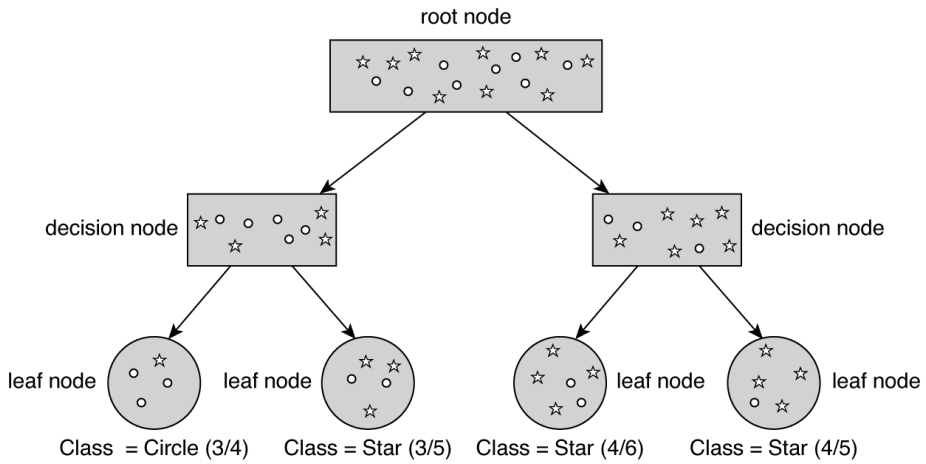

Figura 5.7 – Ejemplo de un árbol de decisión.

La combinación de decisiones o las reglas representadas por todos los nodos de decisión compone el modelo actual. Para el aprendizaje, los árboles de decisión siguen una estrategia heurística que consiste en dividir un problema difícil en tantas partes como sea necesario, donde cada nodo de decisión sucesivo divide las instancias en subgrupos cada vez más pequeños de valores objetivo relativamente semejantes.

En el nodo raíz del árbol, se selecciona la característica que mejor logra dividir los ejemplos de acuerdo con el valor objetivo, antes de dividir los ejemplos verificando distintos valores de característica. Cada prueba divide los datos de ejemplo en dos ramas. Este paso se repite de forma recurrente utilizando las mejores características posibles hasta cumplir los criterios de parada determinado, según se define en la siguiente sección.

Criterios de parada

Los criterios de parada generalmente corresponden a una de las siguientes condiciones:

- Un grupo contiene la mayoría de las instancias que cuentan con el mismo valor objetivo.
- Ya han sido buscadas todas las características de manera exhaustiva.
- El árbol ha alcanzado una profundidad específica en términos de cantidad de ramas.

Un árbol de decisión puede emplear valores de categoría o numéricos para los nodos de decisión.

Árboles de decisión

Los árboles de decisión son un algoritmo de clasificación generalizado dentro del aprendizaje supervisado debido a su transparencia y a que son fáciles de comprender.

Puesto que las reglas en las que se basa la funcionalidad del algoritmo pueden ser examinadas visualmente mediante un diagrama de flujo, los árboles de decisión pueden ser usados en situaciones en las cuales la decisión tomada (asociación de clases prevista) debe ser explicada a miembros de la empresa que no cuentan con conocimientos técnicos.

Poda (pruning) del árbol de decisión

Los árboles de decisión tienden a crecer demasiado, lo cual puede deberse a que hay una gran cantidad de posibles valores de características o a que hay una gran cantidad de características en sí.

En un árbol de decisión que haya crecido demasiado, los grupos generados se vuelven cada vez más pequeños con cada división, hasta el punto que cada grupo contiene la totalidad de las mismas instancias de valores objetivo, lo cual se traduce en un sobreajuste del modelo, ya que en este punto el modelo es específico de los datos de ejemplo y podría no funcionar con precisión con datos no mostrados. Por otro lado, es posible que un árbol de decisión que ha crecido demasiado sea difícil de entender y no proporcione un valor real. En tal situación, los árboles de decisión requieren ser podados (pruning), ya sea mediante técnicas de prepoda o postpoda.

Poda (pruning) del árbol de decisión: prepoda

Este proceso, también llamado poda (pruning) hacia adelante, consiste en un enfoque proactivo en el cual el crecimiento del árbol se detiene después de tomar cierto número de decisiones, o en caso de que exista un número limitado de ejemplos que se deban aprender. Si bien dicho enfoque es más eficiente y requiere menos tiempo en términos de evitar que el algoritmo lleve a cabo una categorización innecesaria, puede ocasionar que se ignoren las pruebas de características de aprendizaje que pueden contribuir a aumentar la precisión del modelo.

Poda (pruning) del árbol de decisión: postpoda

Este proceso, también llamado poda (pruning) hacia atrás, consiste en un enfoque pasivo en el cual el árbol crece demasiado, permitiendo que el algoritmo categorice de forma perfecta y correcta todos los ejemplos antes de eliminar los nodos de decisión innecesarios. Esto se lleva a cabo ejecutando el modelo en función del dataset de validación luego de la poda (pruning) o usando una medida estadística; por ejemplo, estimaciones de errores o prueba de significancia. El proceso actual de postpoda implica reemplazar los árboles secundarios con nodos de hoja, a fin de que el árbol se vuelva más pequeño.

En comparación con la prepoda, la postpoda es de uso más generalizado, ya que ayuda a crear un modelo más preciso sin el riesgo de ignorar ninguna prueba importante de valor de característica. Puesto que un requisito para la postpoda es permitir que el árbol crezca hasta alcanzar un gran tamaño, este proceso generalmente requiere más recursos. Un árbol también puede aumentar de tamaño de manera innecesaria, debido a la presencia de datos atípicos (outliers). La poda (pruning) de los árboles de decisión mejora la precisión del modelo, eliminando los nodos que realizan pruebas sobre los datos atípicos (outliers).

Separación por características (feature splitting)

Los árboles de decisión también pueden ser usados para identificar variables con la mayor capacidad de predicción, debido a que cada nodo de hoja muestra la asociación de clases prevista y generalmente la probabilidad de la asociación de clases de una instancia. La separación por características (feature splitting) consiste en la selección de atributos que proporcionan el mejor valor, separando las instancias de tal manera que cada nodo de decisión cree más subgrupos puros de las instancias. Esta es una de las actividades más importantes al desarrollar un modelo de árboles de decisión.

Un grupo es considerado 100% puro en caso de que todas las instancias pertenezcan a la misma clase. La pureza disminuye a medida que las instancias pertenecientes a diferentes clases aparecen en el grupo. El algoritmo utiliza un criterio de separación para seleccionar la mejor característica del valor y la prueba que será realizada sobre el valor de características para separar las instancias, usando una medida para determinar la mejor característica del valor.

En las características discretas, por cada valor del árbol de decisión crece una rama, tal como se ilustra en la Figura 5.8. Con respecto a los árboles binarios que solo consisten en ramas verdaderas o falsas, dos ramas crecen en el lugar donde cada rama cubre un subconjunto del conjunto completo de valores discretos, según lo especifica el criterio de separación, como se ilustra en la Figura 5.9.

En las características continuas, el valor continuo es dividido en un punto, según lo especifica el criterio de separación, antes de que las ramas crezcan en menor, igual y mayor medida que los valores, según se ilustra en la Figura 5.10.

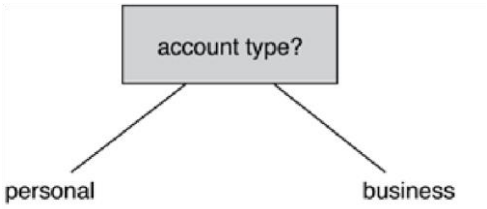

Figura 5.8 - Ejemplo de separación por características (feature splitting), donde el valor de características es discreto.

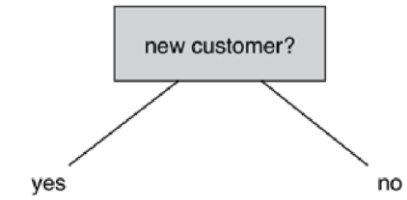

Figura 5.9 - Ejemplo de un árbol binario compuesto únicamente por ramas verdaderas o falsas.

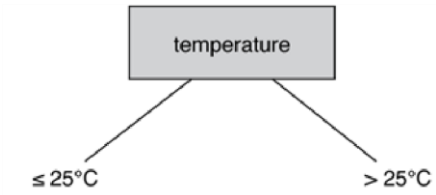

Figura 5.10 - Ejemplo de separación por características (feature splitting), donde el valor de características es continuo.

Criterio de división: entropía y ganancia de información

Los diferentes tipos de medidas —como la ganancia de información, el índice de ganancia y el índice de Gini— pueden ser usados como criterios de separación. La ganancia de información es el criterio más generalizado que se basa en la medida de entropía. La entropía es la medida de impureza dentro de un grupo, y aumenta proporcionalmente con el nivel de impureza dentro de un grupo.

La ganancia de información mide el cambio en la entropía después de la operación de separación. Existe una relación inversa entre la entropía y la ganancia de información, en la cual la ganancia de información aumenta a medida que la entropía disminuye, y viceversa. Asimismo, mide cuánta más información es proporcionada por una característica con el fin de determinar el valor objetivo de una instancia.

Reglas de clasificación

Las reglas de clasificación son una variación de las reglas de decisión, donde el modelo se expresa en forma de enunciados "si - entonces", de la forma "si *condición* entonces *resultado*". La parte de la regla que contiene la condición se conoce como el antecedente. El resultado, conocido como consecuente, especifica la clase prevista. Por ejemplo, si la presión del aire es baja, entonces es posible que llueva.

El antecedente está compuesto por pruebas lógicas llevadas a cabo en uno o más valores de características. Se considera que una regla cubre una instancia si se cumplen los antecedentes de la regla, o en otras palabras, si una regla es capaz de clasificar una instancia.

Medidas de evaluación: cobertura y precisión

Las reglas son evaluadas mediante sus medidas de cubrimiento y precisión. El cubrimiento consiste en la proporción de instancias cubiertas por la regla, entre todas las instancias; mientras que la precisión consiste en la proporción de instancias correctamente clasificadas, entre las instancias cubiertas. Para el aprendizaje, las reglas de clasificación también siguen una estrategia heurística que consiste en dividir un problema difícil en tantas partes como sea necesario, en la cual se divide un subgrupo de ejemplos incluido en una regla. El proceso continúa hasta que todos los ejemplos hayan sido cubiertos, lo que da como resultado múltiples subgrupos y reglas.

Reglas de clasificación

Un subgrupo puede disminuir de tamaño a medida que se añaden más condiciones al antecedente de la regla, según se muestra en la Figura 5.11. Esto se debe a que la precisión aumenta en relación con la predicción correcta de la etiqueta de una clase.

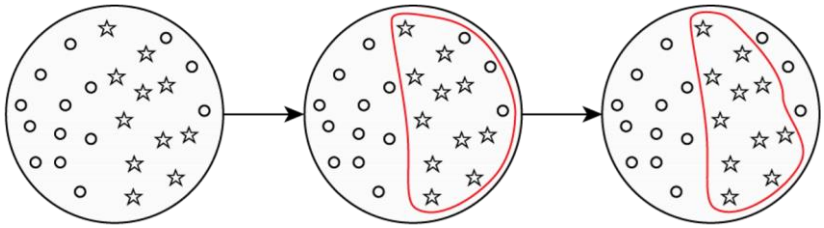

Figura 5.11 – Ejemplo de un subgrupo disminuyendo de tamaño.

Reglas de clasificación

En comparación con el árbol de decisión, un modelo basado en reglas es más fácil de comprender y puede llevar a cabo múltiples pruebas simultáneamente. A diferencia de los árboles de decisión, las reglas no se ven afectadas por la jerarquía de los resultados previos, debido a que cada regla es una prueba independiente que divide los ejemplos sin tener en cuenta ninguna otra regla. Además, las reglas se concentran únicamente en una clase a la vez.

En los árboles de decisión, la característica seleccionada como criterio de separación busca la ganancia de información. En contraste, el objetivo de las reglas de clasificación es la máxima precisión en la asociación de clases de una sola clase. Las reglas de clasificación requieren generalmente de un conjunto de características que comprenda valores nominales y sea eficiente al clasificar instancias con poca posibilidad de ocurrencia.

Algoritmo de una regla (1R)

Este algoritmo consiste en una sola regla que verifica una sola característica. A pesar de ser bastante simplista, este algoritmo es capaz de predecir la asociación de clases con gran precisión. Primero, se consideran múltiples características. Para cada característica, los ejemplos se dividen en subgrupos, donde cada ejemplo dentro del subgrupo tiene el mismo valor de la característica. Posteriormente, se compara el índice de error o precisión de cada regla, con base en una sola característica, y se selecciona la regla con el menor índice de error o mayor

precisión. Por ejemplo, se seleccionan las dos características de color y olor para predecir si determinados hongos son comestibles o no. Las instancias con fondo de color rojo en la Figura 5.12 han sido clasificadas incorrectamente.

Según el bajo índice de error, se seleccionan las siguientes reglas basadas en el olor:

- *"SI el olor es penetrante, ENTONCES el hongo no es comestible".*
- *"SI el olor se asemeja al del pescado, ENTONCES el hongo es comestible".*

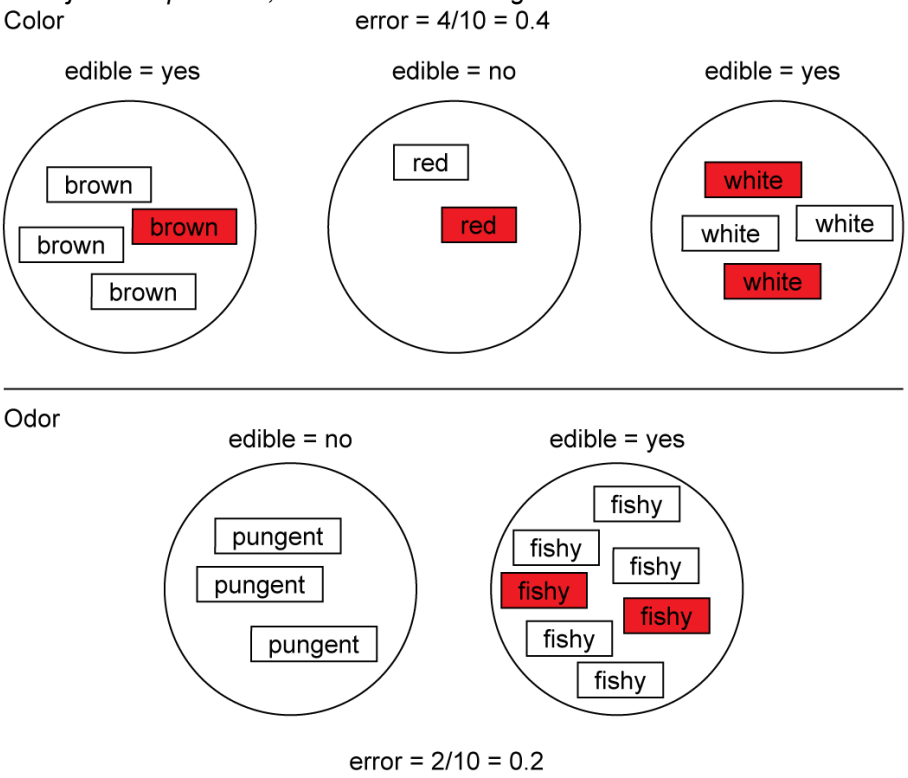

Figura 5.12 – Ejemplo de un algoritmo de una regla (1R)

Clasificación y otras técnicas

Los árboles de decisión pueden ser usados para seleccionar las características de mayor capacidad predictiva como variables explicativas en la regresión. Debido a que pueden surgir datos atípicos (outliers) en los árboles de decisión de gran tamaño, es recomendable eliminar dichos datos mediante técnicas de detección de datos atípicos (outliers) para crear árboles de tamaño y clasificación razonable.

De igual manera, los datos atípicos (outliers) también afectan la operación del algoritmo k-NN si se selecciona un valor de k pequeño, debido a que esto aumenta las posibilidades de que una instancia sea erróneamente clasificada como un dato atípico (outlier). Las técnicas de detección de datos atípicos (outliers) deben ser utilizadas para conseguir una clasificación precisa.

Las técnicas de clasificación, tales como las reglas de clasificación también pueden ser usadas para localizar e identificar los datos atípicos (outliers) ya que las reglas de clasificación son eficientes al momento de clasificar instancias con poca probabilidad de ocurrencia. Cuando se usa en conjunto con el agrupamiento (clustering), la clasificación puede utilizarse para el aprendizaje semisupervisado en situaciones en las cuales se dispone de datos limitados de entrenamiento. Esto es especialmente cierto para aquellos datos obtenidos dentro de los entornos Big Data donde la revisión y el etiquetado de un amplio número de instancias puede requerir muchos recursos y puede ser potencialmente imposible.

En situaciones en las cuales solo se dispone de una cantidad limitada de datos de ejemplo, la clasificación se usa principalmente para crear un modelo de clasificación. En la Figura 5.13, las figuras sombreadas representan los datos etiquetados.

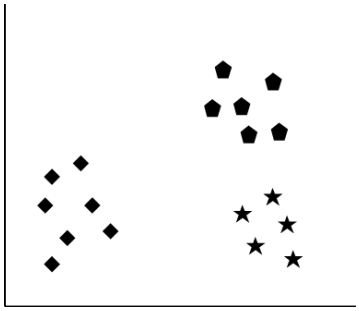

Figura 5.13 - Ejemplo de un modelo de clasificación.

Posteriormente, el agrupamiento (clustering) es usado para crear clusters de datos sin etiquetas, tal y como se muestra en la Figura 5.14. Las instancias dentro de estos clústeres son etiquetadas de acuerdo con las semejanzas que comparten con los ejemplos etiquetados. Los datos recientemente etiquetados aparecen como estrellas, diamantes o pentágonos de color gris.

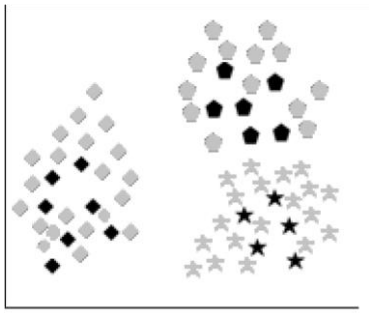

Figura 5.14 – Ejemplo de agrupamiento (Clustering).

Esto se traduce en una gran cantidad de datos etiquetados que pueden ser utilizados para crear un clasificador más preciso, tal y como se muestra en la Figura 5.15.

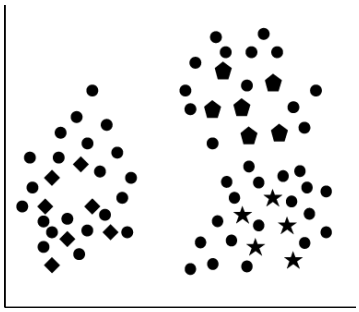

Figura 5.15 – Una gran cantidad de datos etiquetados.

Clasificación y datasets de gran volumen

Los grandes volúmenes de datos contribuyen al desarrollo de modelos de clasificación más precisos, debido a que hay más ejemplos de los cuales aprender y con los cuales comparar (en el caso de k-NN). Por ejemplo, cuando

se utiliza k-NN con un dataset de poco volumen, los ejemplos que pertenecen a clases que rara vez ocurren pueden estar clasificados de manera equivocada, puesto que una instancia puede estar rodeada por ejemplos mayoritarios distantes a los cuales no pertenece realmente, tal y como se muestra en la Figura 5.16. Con los datasets de gran volumen, aumentan las posibilidades de disponer de más ejemplos que pertenecen a una clase poco común, tal y como se muestra en la Figura 5.17. Por ello, al usar k-NN, es probable que se encuentren más cerca cada vez más instancias, lo que se traduce en una clasificación correcta.

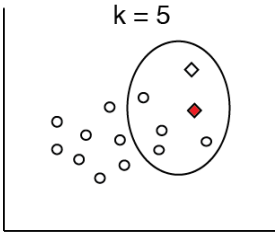

Figura 5.16 - Un dataset de poco volumen.

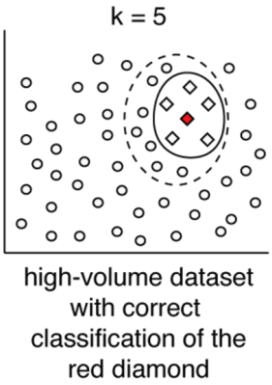

Figura 5.17 - Un dataset de gran volumen.

En el caso de la clasificación basada en la evidencia —por ejemplo, las reglas de clasificación y Naive Bayes—, una mayor cantidad de ejemplos genera más evidencia, a fin de crear antecedentes o calcular la probabilidad de que un ejemplo pertenezca a una clase correcta.

Con los datasets de gran volumen, para seleccionar el mejor clasificador generalmente es necesario aplicar diversos algoritmos de clasificación. Se puede utilizar Naive Bayes como un clasificador de línea base que, a pesar de no ser tan preciso como otros algoritmos, proporciona un modelo de clasificación eficiente y rápido, el cual puede ser actualizado gradualmente.

Mediante los datasets de gran volumen, las técnicas de postpoda producen árboles de gran tamaño que requieren más recursos. Todas las implementaciones disponibles del algoritmo del árbol de decisión que trabajan de manera paralela y distribuida deberían ser usadas, con el fin de utilizar un mecanismo de motor de procesamiento subyacente.

Los datasets de gran volumen también plantean dificultades de rendimiento para aquellos algoritmos que deben actuar en datasets completos cada vez que una nueva instancia requiera ser clasificada, tanto en términos de memoria como de recursos de procesamiento. Al emplear algoritmos de clasificación como k-NN, la implementación algorítmica debería poder ejecutarse un entorno distribuido y paralelo.

Clasificación y datasets altamente veloces

Las tareas de clasificación que se encargan de los datasets altamente veloces deben utilizar algoritmos que requieren menos recursos de procesamiento y que puedan trabajar de manera progresiva. Por ejemplo, Naive Bayes puede ser utilizado para actualizar el modelo de clasificación al actualizar las probabilidades basadas únicamente en el procesamiento de datos adicionales.

Los algoritmos que deben examinar todo el dataset con fines de clasificación —por ejemplo, kNN— generalmente primero requieren que los datasets altamente veloces sean guardados en el disco o que todos los datos estén disponibles en la memoria para llevar a cabo una clasificación de datos nuevos en tiempo real, lo cual genera dificultades de rendimiento.

Los datasets altamente veloces que tienden a modificar a largo plazo el comportamiento del proceso de generación de datos podrían requerir que el modelo de clasificación subyacente sea actualizado por medio del reaprendizaje periódico. En dichas circunstancias, se puede utilizar un mecanismo de motor de flujo de trabajo (Workflow),

combinado con un mecanismo de motor de transferencia de datos, con el objetivo de actualizar periódicamente los modelos de clasificación. No obstante, es importante considerar que los modelos regenerados deben ser verificados antes de ser implementados nuevamente en el sistema en vivo.

Clasificación y datasets de gran variedad

Los datasets de gran variedad pueden resultar ventajosos para las tareas de clasificación, debido a que diversos datasets pueden fusionarse para crear datasets de gran variedad que contengan características adicionales que son usadas por los clasificadores para desarrollar modelos más precisos. Sin embargo, la clasificación de datasets extensos plantea dificultades de rendimiento, ya que hay más valores de características que requieren procesamiento. Por ejemplo, en el caso de las reglas de clasificación, se pueden generar más reglas o las reglas se pueden volver más complejas. Asimismo, respecto a k-NN, el cálculo de la distancia euclidiana se hace más complejo porque se añaden más características.

Al combinar datasets de gran variedad, el dataset resultante puede contener una variedad de características nominales y ordinales. Dependiendo del tipo de clasificador utilizado, las características nominales deberán ser convertidas a ordinales o viceversa. Por ejemplo, Naive Bayes requiere valores nominales, lo cual significa que será necesaria una discretización de las características continuas. No obstante, k-NN requiere valores numéricos, por lo cual los valores nominales deberán ser convertidos a valores numéricos.

Clasificación y datasets altamente veraces

Los datasets altamente veraces son importantes para aprender el modelo de clasificación correcto, ya que la presencia de ruido afecta la precisión de dicho modelo y provoca errores en la clasificación de datos. El ruido debe ser eliminado para lograr que la clasificación sea más precisa y reducir los registros generales que deben ser procesados, lo cual contribuye a que los tiempos de procesamiento sean más rápidos. Durante la etapa de adquisición y filtración de datos del ciclo de vida de análisis de Big Data, se pueden utilizar técnicas automatizadas de eliminación de ruido.

La eliminación excesiva de registros causada por la reducción del ruido puede tener efectos adversos en la tendencia de desarrollo de un modelo de clasificación preciso. Algunas técnicas de clasificación, como los árboles de decisión, funcionan de buena manera en caso de datos que contienen ruido. En tal caso, en lugar de eliminar los registros que tienen ruido, se establecen valores predeterminados o nulos para aquellos valores de características que contienen ruido. Esto garantiza datasets más grandes y con menor cantidad de ruido. En el caso de Naive Bayes, las características que contienen valores faltantes simplemente pueden eliminadas a la vez que se conserva el ejemplo real.

Clasificación y datasets de gran valor

Los datasets de gran valor que están diseñados para propósitos de clasificación deben contener una cantidad significativa de ejemplos de entrenamiento. En aquellos casos en los que eliminar ruido para obtener datasets de gran valor produce una pequeña cantidad de datos de ejemplo, Naive Bayes puede ser utilizado como un clasificador de línea base, ya que funciona bien incluso con pequeñas cantidades de datos de ejemplo. Obtener el valor máximo de un dataset requiere la elaboración rápida de un modelo de clasificación, lo que a su vez requiere el soporte de la plataforma subyacente de Big Data.

Podría no ser posible desarrollar modelos precisos de clasificación si la plataforma de Big Data dispone de recursos de procesamiento limitados, o si debe brindarle soporte a otras aplicaciones cotidianas y fundamentales para el funcionamiento de la empresa y que utilizan un gran volumen de recursos. En estos casos, se puede seleccionar un clasificador ligero —como Naive Bayes— para desarrollar un modelo de clasificación sin sobrecargar los recursos subyacentes.