Algoritmos básicos de Data Mining
(Parte I)

Jhon J. Padilla A., PhD.

Introducción

- Una vez visto cómo se representan las entradas y las salidas, es tiempo de estudiar los algoritmos de aprendizaje.
- Ahora estudiaremos las ideas básicas detrás de las técnicas que se utilizan en la práctica del Data Mining.
- Por ahora se estudiarán las bases, y luego se profundizará en ciertos detalles de estas técnicas.

Introducción

- Se recomienda usar una metodología de primero lo más simple, para el análisis de los datos.
- Hay muchas diferentes clases de estructuras que los datos pueden exhibir.
- En la infinita variedad de conjuntos de datos posibles, hay muchas clases de estructuras que pueden ocurrir, y una sola herramienta de Data Mining, no importa cuán potente sea para un tipo de estructura, puede perder regularidades de una clase diferente.
- El resultado puede ser complejo y enredado para una clase pero una estructura simple y elegante para otra clase.

Casos típicos de los conjuntos de datos

1) Puede haber un único atributo que hace todo el trabajo y los otros pueden ser irrelevantes o redundantes.
2) Los atributos podrían contribuir independientemente e igualmente a la salida final.
3) Podrían tener una estructura lógica y simple que envuelve solo unos pocos atributos que pueden ser capturados por un árbol de decisión.

Casos típicos de los conjuntos de datos

4) Podría haber unas pocas reglas independientes que gobiernan la asignación de instancias de diferentes clases

5) Podrían exhibir dependencias entre diferentes subconjuntos de atributos

6) Podrían tener una dependencia lineal entre atributos numéricos, donde lo que importa es la suma de los valores de los atributos con los pesos elegidos apropiadamente.

7) La clasificación puede ser gobernada por las distancias entre las instancias mismas.

8) No se proveen valores de clases, el aprendizaje es no supervisado.

1. Inferencia de reglas rudimentarias

- Hay una forma fácil de encontrar reglas de clasificación de un conjunto de datos.
- Se conoce como 1R ó 1-regla:
  - Genera un árbol de decisión expresado en la forma de un conjunto de reglas que prueban un atributo en particular
- 1R es un método simple y barato que a menudo produce buenas reglas para caracterizar la estructura en los datos.

1R

- En muchos conjuntos de datos reales, la estructura es muy rudimentaria, y solo un atributo es suficiente para determinar la clase de una instancia de manera precisa.
- Es la primera manera al intentar la filosofía primero lo más simple.

1R

- La idea es esta: Hacemos reglas que prueben un atributo único y genere ramificaciones de acuerdo al resultado.
- Cada rama corresponde a un valor diferente del atributo
- En este caso es obvio cuál es la mejor clasificación dada a una rama: es la clase que más ocurre en los datos de entrenamiento.
- La tasa de error de las reglas puede determinarse fácilmente. Sólo cuente los errores que ocurrieron en los datos de entrenamiento, es decir, el número de instancias que no tuvieron la clase principal.

1R: Algoritmo

- Cada atributo genera un conjunto diferente de reglas, una regla para cada valor del atributo.
- El algoritmo para 1R es el siguiente:

```
For each attribute,
  For each value of that attribute, make a rule as follows:
    count how often each class appears
    find the most frequent class
    make the rule assign that class to this attribute-value.
  Calculate the error rate of the rules.
Choose the rules with the smallest error rate.
```

Ejemplo: Datos del Clima

Table 1.2 The weather data.

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Evaluación Datos del clima con 1R

Table 4.1 Evaluating the attributes in the weather data.

	Attribute	Rules	Errors	Total errors
1	outlook	sunny → no	2/5	4/14
		overcast → yes	0/4	
		rainy → yes	2/5	
2	temperature	hot → no*	2/4	5/14
		mild → yes	2/6	
		cool → yes	1/4	
3	humidity	high → no	3/7	4/14
		normal → yes	1/7	
4	windy	false → yes	2/8	5/14
		true → no*	3/6	

*A random choice was made between two equally likely outcomes.

Ejemplo: 1R

- 1R elige el atributo que produce reglas con el número de errores menor. Es decir, el 1er y 3er conjunto de reglas.
- Si se rompe el empate arbitrariamente, se elige:

```
outlook: sunny    → no
         overcast → yes
         rainy    → yes
```

Valores perdidos y atributos numéricos

- Aunque rudimentario, 1R acomoda valores perdidos y atributos numéricos de forma simple pero efectiva
- Los valores perdidos son tratados como otro valor de atributo.
- P.ej. Si los datos del tiempo han contenido valores perdidos para el atributo *Outlook*, un conjunto de reglas formado para *outlook* podría especificar 4 posibles valores de clase: *sunny, overcast, rainy, missing (perdido)*.

Discretización

- Podemos convertir atributos numéricos en nominales usando un método simple de discretización:
  - Primero, ordene los ejemplos de entrenamiento de acuerdo a los valores del atributo numérico. Esto produce una secuencia de valores de clase.
  - Ejemplo: Para la versión numérica de la tabla del clima, al ordenar los datos de acuerdo a los valores de temperatura, se produce la secuencia:

64	65	68	69	70	71	72	72	75	75	80	81	83	85
yes	no	yes	yes	yes	no	no	yes	yes	yes	no	yes	yes	no

Discretización

- La discretización involucra la partición de esta secuencia ubicando puntos de ruptura en ella.
- Una posibilidad es ubicar puntos de ruptura donde la clase cambia, produciendo ocho categorías:

yes | no | yes yes yes | no no | yes yes yes | no | yes yes | no

- Y se eligen los puntos de ruptura a media distancia entre los ejemplos que limitan con el punto de ruptura en: 64.5, 66.5, 70.5, 72, 77.5, 80.5 y 84.

Desventajas de 1R

- En este ejemplo, dos instancias con valor 72 causan problema porque tienen el mismo valor de temperatura pero caen en diferentes clases.
- La solución simple es remover el punto de ruptura en 72 en uno de los lugares, en 73.5, produciendo una partición mezclada en la cual no hay una clase principal.

Desventajas de 1R

- Un problema más serio es que este procedimiento tiende a formar un gran número de categorías.
- El método 1R gravitará normalmente hacia elegir un atributo que se divide en muchas categorías, lo que partirá el conjunto de datos en muchas clases, haciendo que las instancias tengan las mismas clases que la mayoría en su partición.

Desventajas de 1R

- El caso límite es un atributo que tiene diferente valor para cada instancia.
- Esto lleva a una tasa de error cero en el entrenamiento porque cada partición contiene sólo una instancia.
- Por tanto, atributos con muchas ramificaciones no trabajaran bien en los ejemplos de prueba; por tanto, nunca predecirá correctamente un ejemplo fuera del conjunto de entrenamiento. Este fenómeno se conoce como *overfitting*.

Overfitting

- El overffitting ocurrirá cuando un atributo tenga un gran número de valores posibles.
- Por tanto, cuando se discretiza un atributo numérico, se adopta una regla que debe tener un número mínimo de ejemplos de la clase principal en cada partición.

Overfitting

- Suponga que se fija el mínimo en 3. Esto elimina todas menos dos de las particiones precedentes.
- Entonces el proceso de partición empieza:

```
yes no yes yes | yes . . .
```

- Asegurando que hay tres ocurrencias de *yes* en la clase principal en la primera partición.

- Sin embargo, ya que el siguiente ejemplo es también *yes*, no perdemos nada incluyendo eso en la primera partición también. Esto lleva a una nueva división:

yes no yes yes yes | no no yes yes yes | no yes yes no

- Donde cada partición contiene al menos 3 instancias de la clase principal, excepto la última, que usualmente tiene menos.
- Los límites de la partición siempre caen entre ejemplos de diferentes clases.

- Cuando las particiones adyascentes tienen la misma clase principal, tal como se hizo en las dos particiones anteriores, ellas pueden ser fusionadas sin afectar el significado del conjunto de reglas. Así, la discretización final es:

  yes no yes yes yes no no yes yes yes | no yes yes no

- Lo que lleva al conjunto de reglas:

  temperature: ≤ 77.5 → yes
  > 77.5 → no

- La segunda regla fue una elección arbitraria y se eligió el No. Si hubiésemos elegido Si (yes), no hubiésemos necesitado de un punto de ruptura, y hubiese sido mejor usar categorías adyascentes para ayudar a romper los empates.

- Esta regla genera 5 errores en el conjunto de entrenamiento y por tanto es menos efectiva que la regla precedente para Outloook.
- El mismo procedimiento lleva a esta regla para humedad:

```
humidity: ≤ 82.5 → yes
          > 82.5 and ≤ 95.5 → no
          > 95.5 → yes
```

- Esta genera solo 3 errores en el conjunto de entrenamiento y es la mejor regla 1R para los datos del clima.

- Finalmente, si un atributo numérico tiene valores perdidos, se crea una categoría adicional para ellos, y el procedimiento de discretización se aplica solo a las instancias para las cuales el valor de atributo está definido.

Rendimiento de 1R

- En un estudio de un artículo titulado "*Very simple classification rules perform well on most commonly used datasheets*"(Holte 1993), se realizó un estudio de rendimiento del método 1R con 16 conjuntos de datos frecuentemente usados por investigadores de Machine Learning para evaluar sus algoritmos.
- El método1R se comportó asombrosamente bien en comparación con los demás métodos del estado del arte.
- Las reglas producidas sólo estuvieron unos puntos porcentuales por debajo de la precisión de los árboles de decisión obtenidos por otros métodos.
- Estos árboles fueron más grandes que las reglas 1R.

Rendimiento de 1R

- Las reglas que prueban un atributo único a menudo son una alternativa viable para estructuras más complejas.
- Por tanto, es recomendable usar una estrategia de “lo más simple primero”, antes de continuar con técnicas más complejas que inevitablemente generan salidas que son duras de interpretar para las personas.

Árboles de decisión a partir de 1R

- 1R aprende un árbol de decisión de un nivel cuyas hojas representan varias clases diferentes.
- Una manera de complementar el resultado es usar una regla diferente para cada clase. Cada clase es un conjunto de pruebas, una para cada atributo.
- Para atributos numéricos, la prueba verifica si el atributo se encuentra dentro de un intervalo.
- Para atributos nominales, la prueba verifica si está en un subconjunto de valores del atributo.
- Estos dos tipos de pruebas se aprenden del conjunto de datos de entrenamiento para cada clase.

Árboles de decisión a partir de 1R

- Para atributos numéricos, los extremos del intervalo son los valores mínimo y máximo que ocurren en los datos de entrenamiento para esa clase.
- Para un atributo nominal, los subconjuntos contienen solo los valores que ocurren para el atributo en los datos de entrenamiento para esa clase.
- Las reglas que representan diferentes clases suelen traslaparse, y al momento de hacer la predicción, suele escogerse la clase con más coincidencias. Esta técnica da a menudo una útil primera impresión del conjunto de datos, es extremadamente rápida y puede aplicarse a grandes cantidades de datos.

2. Modelado Estadístico

- El método 1R usa un atributo único para la base de sus decisiones y elige el que trabaja mejor.
- Otra técnica simple es usar todos los atributos y permitirles hacer contribuciones a la decisión.
- Los atributos son igualmente importantes e independientes de otros. Esto por supuesto es poco realista.
- En los datos reales, los atributos no son igualmente importantes o independientes. Esto lleva a un esquema simple que trabaja muy bien en la práctica.

Ejemplo: Los datos del Clima

- La tabla 4.2 muestra un resumen de los datos del clima obtenidos contando cuantas veces cada par valor-atributo ocurre con cada valor para el atributo *play (yes-no)*.

-

Ejemplo: Datos del Clima

Table 1.2 The weather data.

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

- Ud. puede ver en la **tabla 1.2** que *outlook=sunny* para 5 ejemplos, dos de los cuales tienen *play=yes*, y tres de los cuales tienen *play=no*. Las celdas en la primera fila de la **tabla 4.2** simplemente cuentan estas ocurrencias para todos los posibles valores de cada atributo, y la salida *play* en la columna final cuenta el número de ocurrencias totales de *yes* y *no*.

Ejemplo: Los datos del clima

Table 4.2 **The weather data with counts and probabilities.**

Outlook			Temperature			Humidity			Windy		Play	
	yes	*no*		*yes*	*no*		*yes*	*no*		*yes*	*no*	*yes*	*no*
sunny	2	3	hot	2	2	high	3	4	false	6	2	9	5
overcast	4	0	mild	4	2	normal	6	1	true	3	3		
rainy	3	2	cool	3	1								
sunny	2/9	3/5	hot	2/9	2/5	high	3/9	4/5	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	mild	4/9	2/5	normal	6/9	1/5	true	3/9	3/5		
rainy	3/9	2/5	cool	3/9	1/5								

- En la parte baja de la tabla, escribimos la misma información en la forma de fracciones, o probabilidades observadas. P.ej. De los 9 días que *play=yes, outlook=sunny* en 2 ocasiones, dando una fracción de 2/9. Para *play* las fracciones son diferentes: ellas son la proporción de días que *play=yes* y *play=no* respectivamente, con respecto al total de filas de la **tabla 1.2** (que son 14).

Cómo clasificar un nuevo ejemplo?

- Supongamos que ahora tenemos un nuevo ejemplo dado en la tabla:

Table 4.3 A new day.

Outlook	Temperature	Humidity	Windy	Play
sunny	cool	high	true	?

- Cuál deberá ser la salida para Play, (yes, no)?
- Trataremos las 5 características: *outlook, temperature, humidity, windy y play*, como igualmente importantes e independientes y multplicaremos las fracciones correspondientes.

Pasos para clasificar el nuevo ejemplo:

- Calcularemos la posibilidad para la salida *play=yes*:

  posibilidad de yes = 2/9 x 3/9 x 3/9 x 3/9 x 9/14 = 0.0053

- Las fracciones son tomadas de las entradas yes en la tabla 4.2 de acuerdo a los valores de los atributos para el nuevo día, y el 9/14 final es la fracción total que representa la proporción de días en los cuales play=yes.
- Un cálculo similar se puede hacer para la posibilidad de *play=no*:

  posibilidad de no = 3/5 x 1/5 x 4/5 x 3/5 x 5/14 = 0.0206

Pasos para clasificar el nuevo ejemplo:

- Lo anterior indica que la posibilidad para el No es mayor que para el YES. Es cuatro veces más posible.
- El número puede ser convertido a probabilidades, normalizando los resultados para que ellos sumen 1:

$$\text{Probability of } yes = \frac{0.0053}{0.0053 + 0.0206} = 20.5\%,$$

$$\text{Probability of } no = \frac{0.0206}{0.0053 + 0.0206} = 79.5\%.$$

Regla de Bayes

- Este método simple e intuitivo está basado en la regla de Bayes para la probabilidad condicional.
- La regla de Bayes dice que si se tiene una hipótesis H y una evidencia E, que se basa en esa hipótesis, entonces:

$$\Pr[H|E] = \frac{\Pr[E|H]\Pr[H]}{\Pr[E]}.$$

Regla de Bayes

- Pr[A] denota la probabilidad del evento A
- Pr[A|B] denota la probabilidad de A dado el evento B.
- La hipótesis H es que *Play=yes*
- Pr[H|E] será 20.5%, como fue explicado anteriormente
- La evidencia E es la combinación de atributos para el nuevo día: *outlook=sunny, temperature=cool, humidity=high, y windy=true*. A estas 4 componentes de la evidencia les llamaremos E1, E2, E3 y E4 respectivamente.
- Asumimos que estos componentes son independientes y que su probabilidad combinada se obtiene multiplicando las probabilidades:

$$\Pr[yes|E] = \frac{\Pr[E_1|yes] \times \Pr[E_2|yes] \times \Pr[E_3|yes] \times \Pr[E_4|yes] \times \Pr[yes]}{\Pr[E]}$$

Regla de Bayes

- La probabilidad Pr[yes] al final del numerador es la probabilidad de que *play=yes* sin importar la evidencia E, es decir, sin tomar en cuenta nada sobre el día particular referenciado. Esta se llama la probabilidad principal de la hipótesis H. En este caso es 9/14, porque de los 14 datos de entrenamiento, 9 tienen *play=yes*.
- Sustituyendo las fracciones obtenidas en la tabla 4.2, la probabilidad da:

$$\Pr[yes|E] = \frac{2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14}{\Pr[E]}$$

Método Naive Bayes

- Pr[E] desaparecerá al normalizar.
- Este método lleva el nombre de Naive Bayes (Bayes ingenuo), porque está basado en la regla de Bayes asumiendo independencia ingenuamente.
- Las probabilidades se pueden multiplicar cuando los eventos son independientes.
- Aunque la independencia no es siempre cierta en la realidad, este método simplifica las cosas y trabaja bastante bien cuando es probado en conjuntos de datos reales, particularmente cuando es combinado con alguno de los procedimientos de selección introducidos en el capítulo 7, que eliminan redundancia, y por tanto, atributos no independientes.

Desventajas de Naive Bayes

- Una cosa que puede salir mal con Naive Bayes es que si un valor de un atributo particular no ocurre en los datos de entrenamiento en conjunción con cada valor de clase, las cosas se pueden desviar.
- Suponga que en el ejemplo los datos de entrenamiento son diferentes y que el valor de atributo *outlook=sunny* siempre ha sido asociado con la salida *No*. Entonces la probabilidad de *outlook=sunny* da un *Yes*. Esto es, *Pr[outlook=sunny|yes]* podría ser cero, y debido a que otras probabilidades son multiplicadas por esta, la probabilidad final de *yes* podría ser cero sin importar cuántos datos hubiese.
- Las probabilidades que son cero ponen un veto sobre las otras. Esto no es una buena idea.

Correcciones a Naive Bayes

- Una solución a este problema es hacer ajustes al método de calcular las probabilidades a partir de las frecuencias.
- P.ej., la parte alta de la tabla 4.2 muestra que para *play=yes, outlook=sunny* para dos ejemplos, *outlook=overcast* para 4 ejemplos y *outlook=rainy* para 3 ejemplos. Así, la parte de abajo de la tabla da las probabilidades 2/9, 4/9 y 3/9 respectivamente.
- Si en lugar de eso adicionamos 1 a cada numerador y compensamos adicionando 3 al denominador, tendremos las probabilidades 3/12, 4/12 y 4/12, respectivamente.
- Esto asegura que un valor de atributo que ocurre cero veces, recibe una probabilidad diferente de cero, aunque muy pequeña.
- La estrategia de sumar 1 a cada cuenta es una técnica estándar llamada el **estimador de Laplace**.

Correcciones a Naive Bayes

- Aunque esto trabaja bien en la práctica, no hay una razón particular para sumar 1 a la cuenta: podríamos en lugar de esto seleccionar una constante μ y hacer:

$$\frac{2+\mu/3}{9+\mu}, \frac{4+\mu/3}{9+\mu}, \text{and } \frac{3+\mu/3}{9+\mu}$$

- El valor de μ, que en este caso es 3, provee efectivamente de un peso que determina cuan influyentes son cada uno de los tres posibles atributos. Para este caso, cada uno pesa 1/3.

Correcciones a Naive Bayes

- Un valor grande de μ dice que estos precedentes son más importantes comparados con la nueva evidencia que viene del conjunto de entrenamiento, mientras que un valor pequeño da menos influencia.
- Finalmente, no hay una razón particular para dividir μ entre 3 partes iguales en los numeradores, por lo que podemos hacer:

$$\frac{2+\mu p_1}{9+\mu}, \frac{4+\mu p_2}{9+\mu}, \text{and } \frac{3+\mu p_3}{9+\mu}$$

- Donde p1, p2 y p3 suman 1. Estos números son probabilidades a priori de los valores del atributo *outlook* siendo *sunny*, *overcast* y *rainy*, respectivamente

Correcciones a Naive Bayes

- Esto es una formulación completamente Bayesiana donde las probabilidades de precedencia han sido asignadas a todo a ojo.
- Esto tiene la ventaja de ser completamente riguroso, pero la desventaja de que no es totalmente claro cómo deben asignarse estas probabilidades.
- En la práctica, las probabilidades de precedencia tienen poca diferencia dado que hay un número razonable de instancias de entrenamiento y la gente generalmente solo estima las frecuencias usando el Estimador de Laplace, inicializando las cuentas a uno en lugar de cero.

Valores y atributos perdidos

- Si por ejemplo se pierde el valor para outlook en la tabla 4.3, para el cálculo simplemente se omite este atributo, dando como resultado:

$$\text{likelihood of } yes = 3/9 \times 3/9 \times 3/9 \times 9/14 = 0.0238$$
$$\text{likelihood of } no = 1/5 \times 4/5 \times 3/5 \times 5/14 = 0.0343.$$

- Estos dos números son más altos que los calculados antes debido a que no se multiplican por la primera fracción (falta la fracción del outlook). Sin embargo, esto no es un problema debido a que se hace un proceso de normalización.

Valores y atributos perdidos

- Si un valor se pierde en una instancia de entrenamiento, sencillamente no se incluye en las cuentas de frecuencias, y las tasas de probabilidad se basan en el número de valores que ocurren realmente, en lugar del número total de instancias.

Cálculo de probabilidad usando la distribución Gaussiana

- Los valores numéricos se manejan usualmente asumiendo que tienen una distribución Gaussiana.
- La tabla 4.4 muestra un resumen de los datos del clima con características numéricas.
- Para atributos nominales, se hacen los cálculos igual que antes. Mientras que para los atributos numéricos simplemente se listan los valores que ocurren.
- Como se normalizan las cuentas para los atributos nominales en probabilidades, se calcula la media y la desviación estándar para cada clase y cada atributo numérico.
- Así, el valor medio de la temperatura para las instancias yes es 73, y la desviación estándar es 6.2. La media es el promedio de los valores precedentes, es decir, la suma dividida por el número de valores. La desviación estandar es la raiz cuadrada de la varianza muestral.

Cálculo de parámetros básicos de los datos

Table 4.4 The numeric weather data with summary statistics.

Outlook			Temperature		Humidity		Windy			Play	
	yes	*no*	*yes*	*no*	*yes*	*no*		*yes*	*no*	*yes*	*no*
sunny	2	3	83	85	86	85	false	6	2	9	5
overcast	4	0	70	80	96	90	true	3	3		
rainy	3	2	68	65	80	70					
			64	72	65	95					
			69	71	70	91					
			75		80						
			75		70						
			72		90						
			81		75						
sunny	2/9	3/5	*mean*	73	74.6	*mean*	79.1	86.2	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	*std. dev.*	6.2	7.9	*std. dev.*	10.2	9.7	true	3/9	3/5		
rainy	3/9	2/5											

Cálculo de la probabilidad

- Cálculo de la probabilidad:
- Para el caso de una temperatura de 66 y una humedad de 90 se hace el reemplazo en la fórmula de la distrib. Gaussiana así:

$$f(temperature = 66 \mid yes) = \frac{1}{\sqrt{2\pi} \cdot 6.2} e^{\frac{(66-73)^2}{2 \cdot 6.2^2}} = 0.0340.$$

$$f(humidity = 90 \mid yes) = 0.0221.$$

Cálculo de probabilidades

- Usando las probabilidades calculadas para calcular las posiblidades del sí y el no:

$$\text{likelihood of } yes = 2/9 \times 0.0340 \times 0.0221 \times 3/9 \times 9/14 = 0.000036,$$
$$\text{likelihood of } no = 3/5 \times 0.0221 \times 0.0381 \times 3/5 \times 5/14 = 0.000108;$$

- Entonces las probabilidades se calculan como:

$$\text{Probability of } yes = \frac{0.000036}{0.000036 + 0.000108} = 25.0\%,$$

$$\text{Probability of } no = \frac{0.000108}{0.000036 + 0.000108} = 75.0\%.$$

Conclusiones

- Los valores obtenidos no son idénticos, pero son cercanos a los obtenidos para el día de la tabla 4.3 porque la temperatura y la humedad son bastante aproximadas para los valores cool y high usados antes.
- El supuesto de distribución normal hace fácil extender el método de Naive Bayes para que coincida con atributos numéricos.
- Si los valores de los atributos numéricos se pierden, la media y la desviación estándar se calculan sólo con base en los valores que están presentes.

Modelos Bayesianos para clasificación de documentos

Discusión

- Naive Bayes es una aproximación simple con semántica clara, para representar, usar y aprender conocimiento probabilisticamente.
- Se pueden obtener resultados impresionantes usándolo.
- Los métodos rivales de Naive Bayes argumentan mayor precisión, pero son más complejos.
- Muchas veces en Machine Learning ocurre que se llevan años usando métodos más complejos y luego se descubre que métodos tan simples como 1R y Naive Bayes funcionan tan bien o aún mejor.

Discusión

- Hay muchos conjuntos para los cuales Naive Bayes no funciona bien. Esto se debe a que los atributos son tratados como si fuesen independientes. Por tanto, la adición de atributos dependientes sesga el proceso de aprendizaje
- Una manera de disminuir este efecto es seleccionar cuidadosamente los ejemplos para disminuir la dependencia (se verá después).
- Otra restricción usada aquí es el supuesto de normalidad para los datos. Si usted conoce la distribución de probabilidad para un conjunto de datos, puede usar esa distribución y el cálculo de su media y desviación estándar.
- Si usted sospecha que no es una distribución normal pero no conoce la distribución, hay procedimientos para estimar la densidad del núcleo que no asumen una distribución particular para los valores de atributo. Otra posibilidad es simplemente discretizar los datos primero.