Trabajo 12
Clasificación de SPAM

Facultad De Ingeniería, Universidad De Cuenca
TEXT MINING
Freddy L. Abad L.
{freddy.abadl}@ucuenca.edu.ec

Este taller tiene como meta la clasificación de texto. Con este objetivo se aplicará el proceso de clasificación que aprenda la diferencia entre los mensajes spam y los mensajes que realmente desea leer un usuario. Una vez generado el modelo, aplicaremos el mismo a los mensajes nuevos para decidir si son o no spam. El spam es un tema familiar para muchos, por lo que es un medio natural para trabajar. Las mismas técnicas que serán descritas en esta práctica las cuales pueden ser utilizadas para clasificar los mensajes de correo no deseado se pueden utilizar en muchos otros dominios de minería de textos

Para el taller se utilizará en conjunto de datos de 5574 mensajes de texto SMS (teléfono móvil). La colección está compuesta por un solo archivo de texto, donde cada línea tiene la clase correcta seguida del mensaje sin formato.

Algunos ejemplos se muestran a continuación:

> ham What you doing?how are you?
> ham Ok lar... Joking with u oni...
> ham dun say so early hor... U c already then say...
> ham MY NO. IN LUTON 0125698789 RING ME IF UR AROUND! H*
> ham Siva is in hostel aha:-.
> ham Cos i was out shopping wif darren jus now n i called him 2 ask wat present he wan lor. Then he started guessing who i was wif n he finally guessed darren lor.
> spam FreeMsg: Txt: CALL to No: 86888 & claim your reward of 3 hours talk time to use from your phone now! subscribe 6GBP/ month inc 3hrs 16 stop?txtStop
> spam Sunshine Quiz! Win a super Sony DVD recorder if you canname the capital of Australia? Text MQUIZ to 82277. B
> spam URGENT! Your Mobile No 07808726822 was awarded a L2,000 Bonus Caller Prize on 02/09/03! This is our 2nd attempt to contact YOU! Call 0871-872-9758 BOX95QU

Se recomienda usar los siguientes procesos

Crear repositorio

Cree un nuevo repositorio y usando el import wizard importe los datos dentro del repositorio, recordando que los datos están codificados usando UTF-8 y los valores están separados por un Tab y no usa comillas dobles. La otra opción es usar el operador Read CSV y aplicar el mismo procedimiento

Recuerde cambiar la función "att1" de atributo a etiqueta. Esto le dice a RapidMiner que nos gustaría hacer predicciones sobre este atributo. Además, cambie el tipo "att2" de polinomio a texto. Esto le dice a RapidMiner que el atributo contiene texto que nos gustaría manipular.

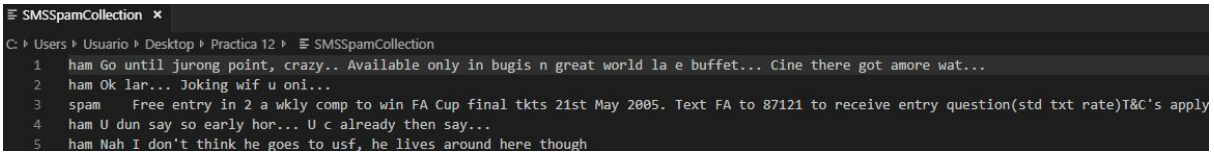

Figura 1 Operador lectura archivo

Configuración del proceso Read CSV, en primer lugar se configura el archivo a leer, y el separador (una tabulación), el formato encoding (UTF-8), nombres de los atributos (att1, att2)

Figura 2 Configuración archivo csv

Figura 3 Configuración archivo csv

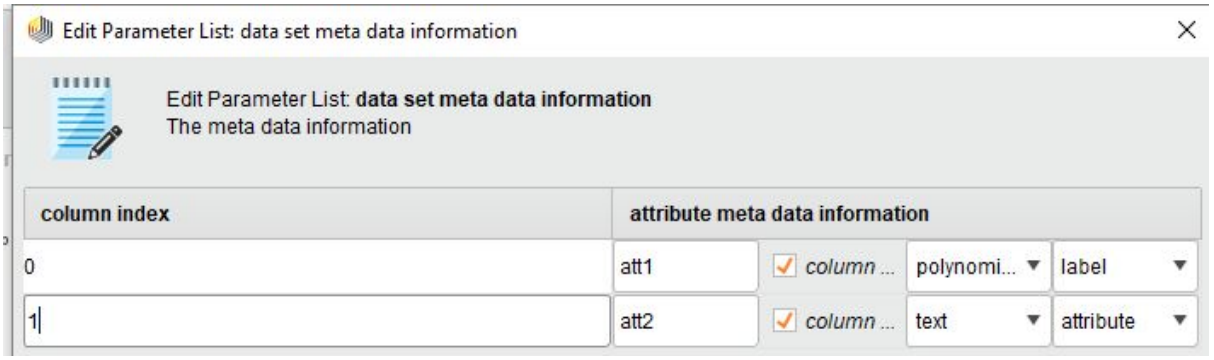

Figura 4 Configuración lista de parámetros

Retrieve

El cual puede ser usado para recuperar datos desde el repositorio.

Process Documents from Data

Para obtener una mejor comprensión del texto, puede ser útil dividir los documentos en palabras individuales y examinar la frecuencia de las palabras. Para ello, utilice los operadores adecuados. Una opción adecuada en este caso es dividir el documento cada vez que el operador encuentre un símbolo, como un carácter de espacio o guión, esto dividirá el documento en un nuevo token. (Figura 7-15)

Una vez dividido el documento tokens se puede analizar el número de ocurrencias de términos, lo que significa que un valor en una celda representa el número de veces que esa palabra aparece en el documento. También puede usar las ocurrencias del término binario, lo que significa que el valor en la celda será cero si la palabra no aparece en el documento, y

uno si la palabra aparece una o más veces en el documento. (Figura 5-6) Siempre es una buena idea examinar los datos procesados para buscar anomalías extrañas. (Figura 7-15)

Figura 5 Operador Process Documents Data

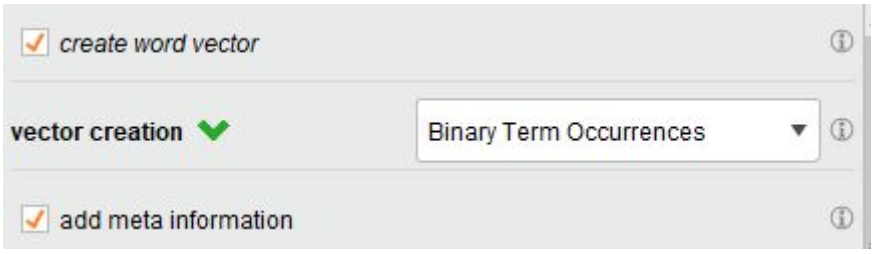

Figura 6 Configuración de Creación de Vector en Operador Process Documents

Figura 7 Diseño de Proceso de procesado de frases, mediante tokenization, filtrado de stopwords, reducir a la raíz de la palabra, tokens y transformar en un solo formato las palabras que entrenan un modelo de clasificación.

Figura 8 Operador Tokenize

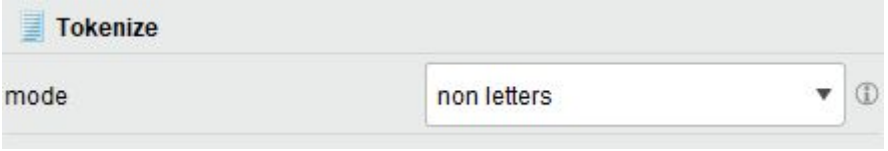

Figura 9 Configuración Operador Tokenize

Figura 10 Operador Filter Stopwords

Figura 11 Operador Stem-Porter

Figura 12 Operador Filter Tokens by Length

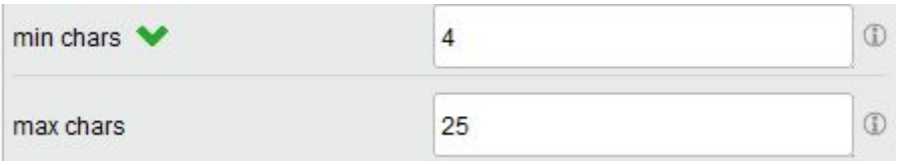

Figura 13 Configuración de Token por tamaño

Figura 14 Operador Transform Cases

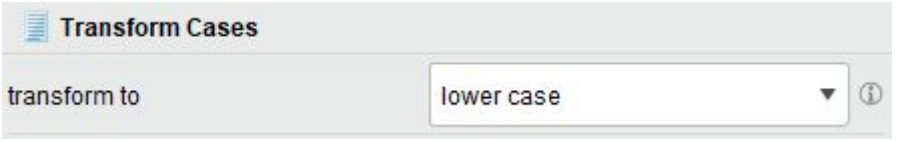

Figura 15 Configuración Transform Cases

Procesando el texto para la clasificación

Si un conjunto de datos tuviera un mensaje de “spam" y 999 “no spam", ¿qué tan preciso sería adivinar que un mensaje es “no spam "?

La respuesta es que sería 99.9% exacto (suponiendo que el modelo es correcto 999 de 1000 veces). Ahora bien este predictor tiene una alta precisión, pero es inútil ya que no tiene un poder predictivo real (pues siempre la respuesta será “no spam”). Para resolver este problema, debemos equilibrar el conjunto de datos, utilizando un número igual de mensajes “no spam "y “spam". Por lo tanto, se puede esperar que un modelo predictor tenga un 50% de precisión, y un número mayor que ese valor indique el poder predictivo real del modelo. ¿Cuál es la desventaja? Tenemos que ignorar una gran parte de nuestros mensajes “no spam".

Para balancear los datos una opción es usar el operador **Sample** *(Figura 16)*. En este operador establezca el parámetro de muestra en absoluto. Esto nos permite elegir el tamaño de muestra. Marque la opción balance data, lo cual nos permite elegir el tamaño de muestra para cada clase de la etiqueta. Aquí elija el botón Editar lista, entonces agregue una clase “spam " con el tamaño 747 por ejemplo (haga clic en el botón agregar entrada para agregar otra fila). Entonces agregue una clase “no spam" con tamaño 747.

Conceptos de procesamiento de texto

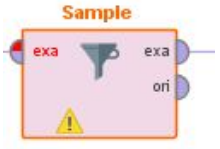

Figura 16 Proceso Sample

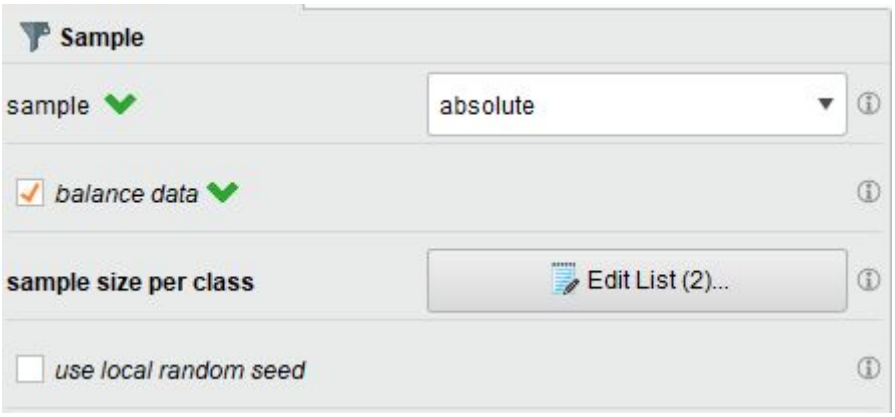

Figura 17 Configuración Proceso Sample

class	size
spam	747
ham	747

Figura 18 Configuración Sample size per class

Verifique si estas condiciones aplican o no al escenario de uso. Deberían "cat", "Cat" y "CAT " contarse como la misma palabra?

Normalmente es una buena idea. Sin embargo, en este caso, el uso distintivo minúscula-mayúscula de letras es un buen predictor de un mensaje de correo no deseado, ya que a los remitentes de spam les gusta llegar a su audiencia con palabras en mayúsculas. En otros procesos, puede usar el operador Transformar casos para forzar todas las palabras a minúsculas.

Caso sin Transform Case	Caso con Transform Case
PerformanceVector PerformanceVector: accuracy: 90.76% ConfusionMatrix: True: ham spam ham: 609 0 spam: 138 747	**PerformanceVector** PerformanceVector: accuracy: 90.76% ConfusionMatrix: True: ham spam ham: 609 0 spam: 138 747

Dada la Matriz de Confusión, no se muestra una cambio de clasificación, ni un cambio de accuracy, en los casos que se varíen todas las palabras a low - upper case. Sin embargo, se puede intuir que variar todo a lowercase si puede disminuir la precisión de predicción.

¿Deberían las palabras "organize", "organizes" y "organized" contarse como la misma palabra?

Esta suele ser una buena idea en la mayoría de los procesos de minería de textos. Sin embargo, en este ejemplo puedes ser que este proceso no mejore la predicción. Verificar este particular ¿Se deben contar los fragmentos de oraciones cortas como elementos distintos? Por ejemplo, junto con el uso individual de “quick", ”brown" y “fox", podríamos incluir el fragmento “quick brown fox" y contar.

Caso sin Stem Porter	Caso con Stem Porter
PerformanceVector PerformanceVector: accuracy: 92.77% ConfusionMatrix: True: ham spam ham: 640 1 spam: 107 746	**PerformanceVector** PerformanceVector: accuracy: 90.76% ConfusionMatrix: True: ham spam ham: 609 0 spam: 138 747

En el caso de no reducir las palabras a su “raíz”, dado la matriz de confusión, si se nota una reducción de clasificaciones erróneas (spam siendo predecida como ham). Se puede intuir que reducir a la palabra raíz no ayuda a mejorar la predicción.

Verifique si el uso de fragmentos es un mejor pronosticador que el uso de palabras individuales.

El uso de fragmentos si tiene mayor validez en la predicción de spam, ya que por ejemplo podría salir una frase: “ATENCION, AQUI VIAJA GRATIS”, se percibe mucho más como spam que palabras sueltas como : ATENCION, AQUI, VIAJA, GRATIS.

Verifique si la eliminación de stopwords mejora la precisión del modelo clasificando los datos como spam o no.

Caso sin Stopwords	Caso con Stopwords
PerformanceVector PerformanceVector: accuracy: 91.70% ConfusionMatrix: True: ham spam ham: 623 0 spam: 124 747	**PerformanceVector** PerformanceVector: accuracy: 90.76% ConfusionMatrix: True: ham spam ham: 609 0 spam: 138 747

*No incluir Stopwords a nuestro algoritmo de predicción **sí muestra** mejoras a incluirlo, se nota como hubo mayores clasificaciones correctas de hm, y menores clasificaciones erróneas de spam como ham*

Use el método Naive Bayes para calcular la probabilidad que un mensaje sea spam o no. Este método funciona de la misma manera en la minería de textos.

Se calcula la clase más probable (“spam "o “no spam") basada en la multiplicación de las probabilidades de los valores de los atributos.

Para construir el modelo de Naive Bayes, agregue el operador de Naive Bayes después del operador Process Documents from Data. Para encontrar la precisión predictiva del modelo, debemos aplicar el modelo a los datos, y luego contar con qué frecuencia sus predicciones son correctas. La precisión de un modelo es la cantidad de predicciones correctas del número total de predicciones.

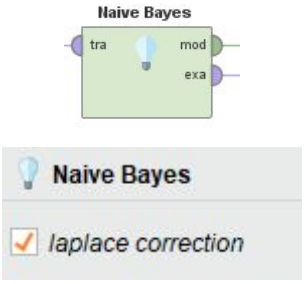

Figura 19 : Operador Naive Bayes y su configuración

Agregue el operador **Apply Model** después del operador de Naive Bayes y conecte los dos nodos.

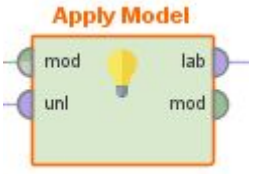

Figura 20 Operación Apply Model

Agregue un operador de Performance después del operador Apply Model y conéctelo a un nodo de res.

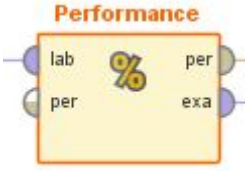

Figura 21 Operation Performance

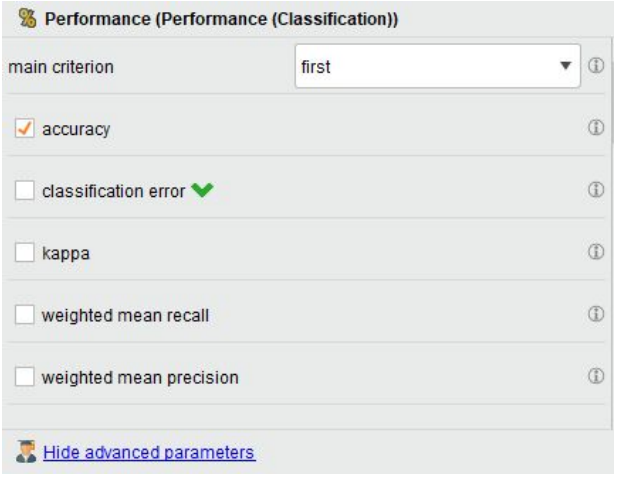

Figura 22 Configuración Operación Performance

Ejecute todo el proceso y verifique cuál es la precisión. Responda porqué la precisión es tan alta?

accuracy: 90.76%

	true ham	true spam	class precision
pred. ham	609	0	100.00%
pred. spam	138	747	84.41%
class recall	81.53%	100.00%	

Figura 23 Matriz de Confusión del Modelo Entrenado

PerformanceVector

```
PerformanceVector:
accuracy: 90.76%
ConfusionMatrix:
True:   ham     spam
ham:    609     0
spam:   138     747
```

Figura 24 Resumen de Accuracy y Matriz de Confusión.

La precisión en clasificación se debe a que las muestras primero son etiquetadas correctamentes, adicionalmente son mensajes que evidentemente muestran distancia entre los mensajes que son spam y los que no. Se muestra sin embargo que los mensajes que no son spam no son clasificados correctamentes, esto se puede deber a que algunas palabras pueden dar peso a que la predicción se incline erróneamente a una clasificación spam.

Validar el modelo, para poder predecir la precisión de un modelo en datos nuevos, debemos ocultar algunos de los datos del modelo y luego probar el modelo con los datos nuevos. Una forma de hacerlo es usar **K-fold Cross-Validation**. Al usar, una validación cruzada de 10 veces, ocultaremos una décima parte de los datos del modelo, construiremos el modelo en los 9/10 de los datos restantes y luego probaremos el modelo en todo el conjunto de datos, calculando su exactitud. Entonces nuevamente se oculta una diferencia de 1/10 de los datos del modelo, y se prueba de nuevo. Se ejecuta este proceso 10 veces en total, y entonces tomamos el promedio de las precisiones. Esto proporciona una mejor idea de cómo el modelo funcionará en datos que no se ha visto antes.

Para ejecutar este proceso de validación ejecute los siguientes pasos:

1. Elimine los operadores Naive Bayes, Apply Model y Performance de la ventana Main Process.
2. Conecte un operador X-Validation al operador Process Documents from Data y conecte su nodo av (para rendimiento promedio) a un nodo de res.
3. Haga doble clic en el operador X-Validation. Ponga un operador de Naive Bayes en el lado izquierdo de este proceso interno, y un operador de Apply Model y un operador de rendimiento en el lado derecho del proceso. Conecte todos los nodos requeridos.

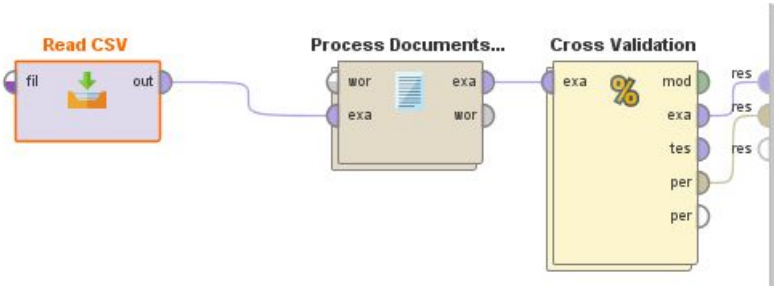

Figura 25 Diseno de Procesos

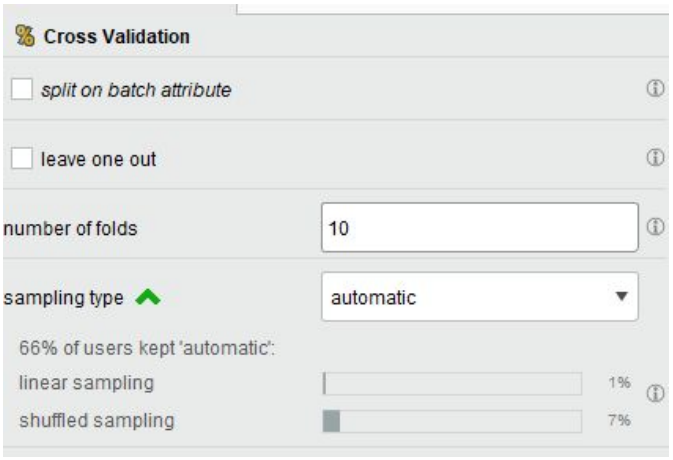

Figura 26 Configuración de Cross Validation.

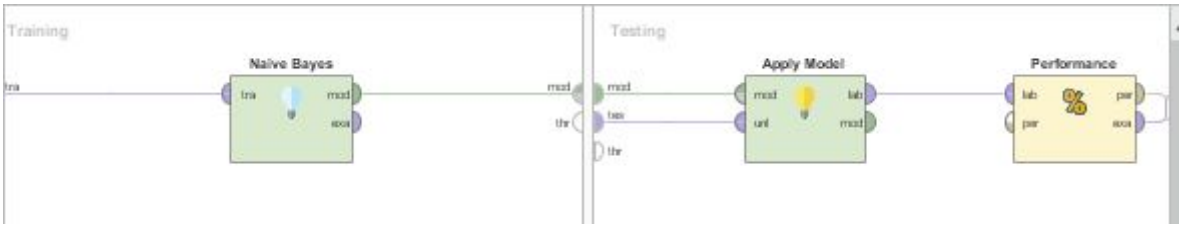

Figura 27 Subproceso Cross Validation Training-Testing

Por favor verifique ¿cuál es la precisión ahora?

accuracy: 75.49% +/- 1.52% (micro average: 75.49%)

	true ham	true spam	class precision
pred. ham	3534	73	97.98%
pred. spam	1293	674	34.27%
class recall	73.21%	90.23%	

Figura 28 Matriz de Confusión.

PerformanceVector

```
PerformanceVector:
accuracy: 75.49% +/- 1.52% (micro average: 75.49%)
ConfusionMatrix:
True:   ham     spam
ham:    3534    73
spam:   1293    674
```

Figura 29 Resumen de Accuracy y Matriz de Confusión.

Cuál es la validación promedio luego de los 10 validaciones?

Intento	accuracy (%)	Pred Ham (%)	Pred Spam (%)
1	75,49% +/- 1,52%	97,98	34,27
2	75,49	97,98	34,27
3	75,49	97,98	34,27
4	75,49	97,98	34,27
5	75,49	97,98	34,27
6	75,49	97,98	34,27
7	75,49	97,98	34,27
8	75,49	97,98	34,27
9	75,49	97,98	34,27
10	75,49	97,98	34,27

Figura 30 Resultados de Accuracy y Matriz de Confusión.

El accuracy promedio es de 75.49% ***(Figura 30)***

Otras tareas

Aplicando el modelo a nuevos datos.

Para aplicar el modelo aprendido a datos nuevos, primero debemos guardar el modelo, para poder usarlo nuevamente en nuevos datos. También tenemos que guardar la lista de palabras. La razón por la que necesitamos guardar la lista de palabras es porque tenemos que comparar manzanas con manzanas. Cuando estamos estimando la probabilidad de que un nuevo mensaje sea “spam" o “no spam", tenemos que usar los mismos atributos (palabras) que usamos en el proceso original. Para aplicar el modelo a nuevo datos, necesita la misma lista de palabras, el mismo modelo y la necesidad de procesar los datos nuevos exactamente de la misma manera en que procesó los datos de aprendizaje. Lo único diferente es la nueva información.

1. Conecte un operador Store al puerto wor en el operador Process Documents from Data.

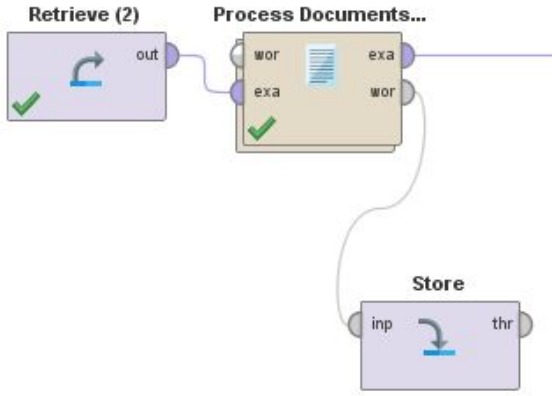

Figura 31 Retrieve - Store datos.

Establezca el parámetro de entrada del repositorio en algo memorable. Esto guardará el modelo para ser usado más tarde.

Figura 32 Entrada del repositorio.

2. Conecte un operador Store al puerto de mod (para el modelo) del operador de Cross Validation.

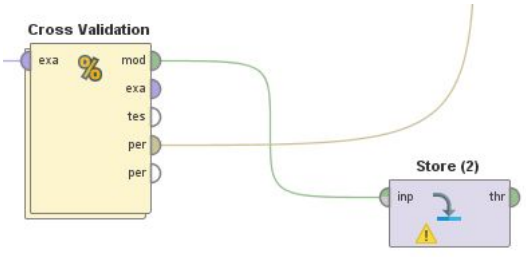

Figura 33 Almacenamiento de datos procesados por X-Validation en un repositorio

Establezca el parámetro de entrada del repositorio en algo memorable. Esto guardará el modelo para ser usado más tarde.

Figura 34 Almacenamiento de datos procesados.

3. Ejecute el proceso nuevamente. Ejecutando el Modelo en Datos Nuevos

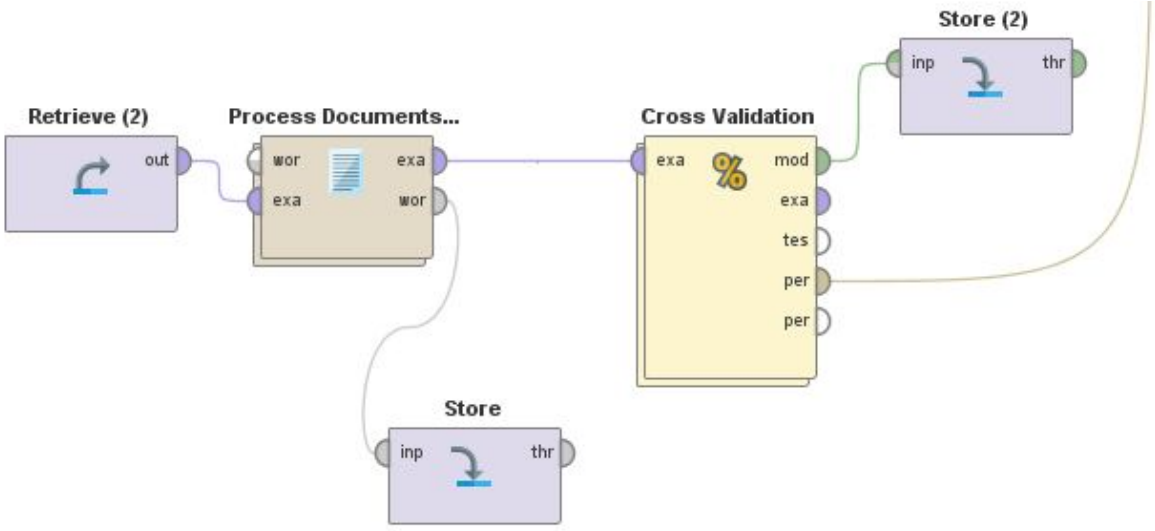

Figura 35 Ejecucion de modelo con datos nuevos.

Word	Attribut...	Total O...	Docum...	ham	spam
aah	aah	1	1	1	0
aathi	aathi	6	6	6	0
ab	ab	1	1	0	1
abbei	abbei	1	1	1	0
abeg	abeg	1	1	1	0
abel	abel	1	1	1	0
aberdeen	aberdeen	1	1	0	1
abi	abi	2	2	2	0

Figura 36 Resultados.

SimpleDistribution

```
Distribution model for label attribute attl


Class ham (0.849)
4134 distributions

Class spam (0.151)
4134 distributions
```

Figura 37 Clasificación con Datos nuevos.

1. Cree y guarde un nuevo proceso. Usaremos este proceso para aplicar el modelo en nuevos datos para predecir si un mensaje nuevo es spam o no.
2. Importe datos nuevos, no usados en el repositorio, como fue descrito al comienzo de esta práctica. Debe estar en el mismo formato que los otros datos. Agregue un operador Retrieve para recuperar los datos.

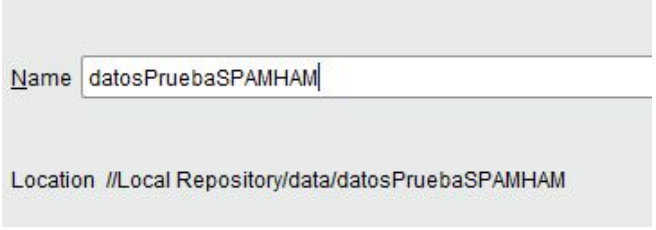

Figura 38 Importe de datos nuevos

Row No.	att1 (polynominal) *label*	att2 (text) *regular*
1	spam	We tried to ca...
2	ham	K...k...when w...
3	spam	This is the 2n...
4	ham	He's just gon...
5	ham	Did you get a...

Figura 39 Resultado importe datos

3. Copie y pegue el operador Process Documents from Data del proceso anterior en este proceso.

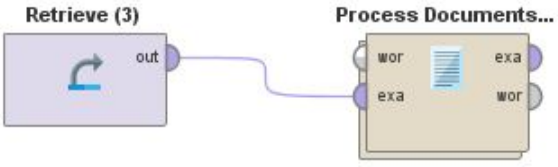

Figura 41 Operadores necesarios para procesar con datos nuevos el modelo.

4. Conecte un operador de Apply Model al operador de Process Documents.

Figura 42 Operadores necesarios para procesar con datos nuevos el modelo.

5. Conecte un operador Retrieve al puerto wor del lado izquierdo del operador Process Documents, y configure su parámetro de entrada al repositorio con el nombre de la lista de palabras que guardó previamente. Esto cargará la lista de palabras anterior.

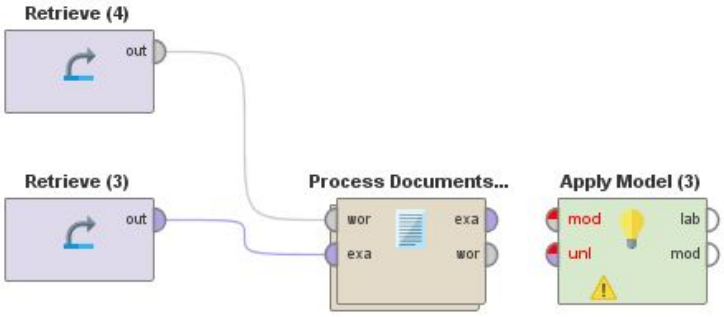

Figura 43 Operadores necesarios para procesar con datos nuevos el modelo.

Figura 44 Repositorio con datos nuevos.

6. Conecte un operador Retrieve al puerto de mod del lado izquierdo del operador Apply Model y establezca el parámetro de entrada del repositorio al nombre del modelo que guardó previamente. Esto cargará el modelo previamente aprendido.

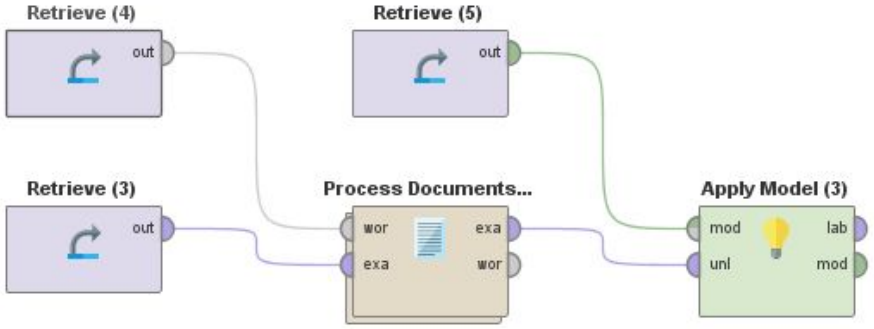

Figura 45 Carga de modelo previamente entrenado.

Figura 46 Retrieve con datos nuevos.

7. Ejecute el nuevo proceso y observará las predicciones del modelo en la salida.

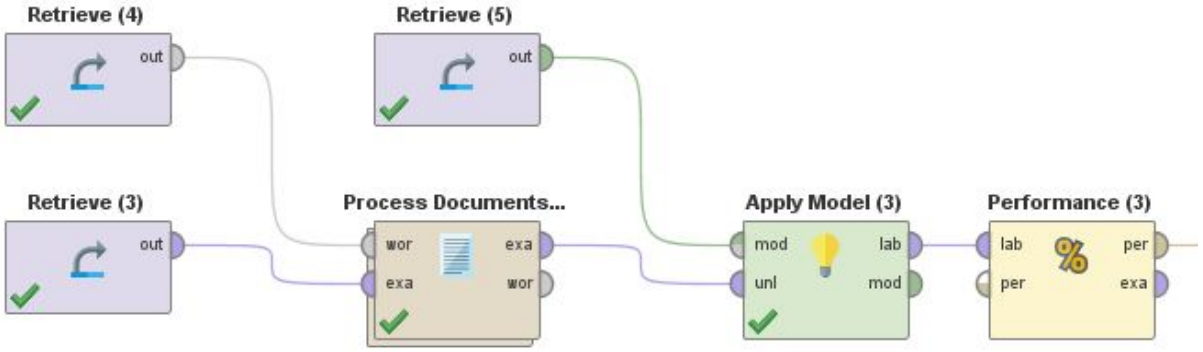

Figura 47 Operadores necesarios para obtener porcentajes de precisión del modelo entrenado con nuevo datos.

accuracy: 85.48%

	true spam	true ham	class precision
pred. spam	189	202	48.34%
pred. ham	23	1136	98.02%
class recall	89.15%	84.90%	

Figura 48 Matriz de Confusión, el accuracy es equivalente a 84.48%.