06/12/2020

PROYECTO DE COMPUTACIÓN I

CATEGORIZACIÓN DE NOTICIAS RAPIDMINER

ACTIVIDAD 3

Índice

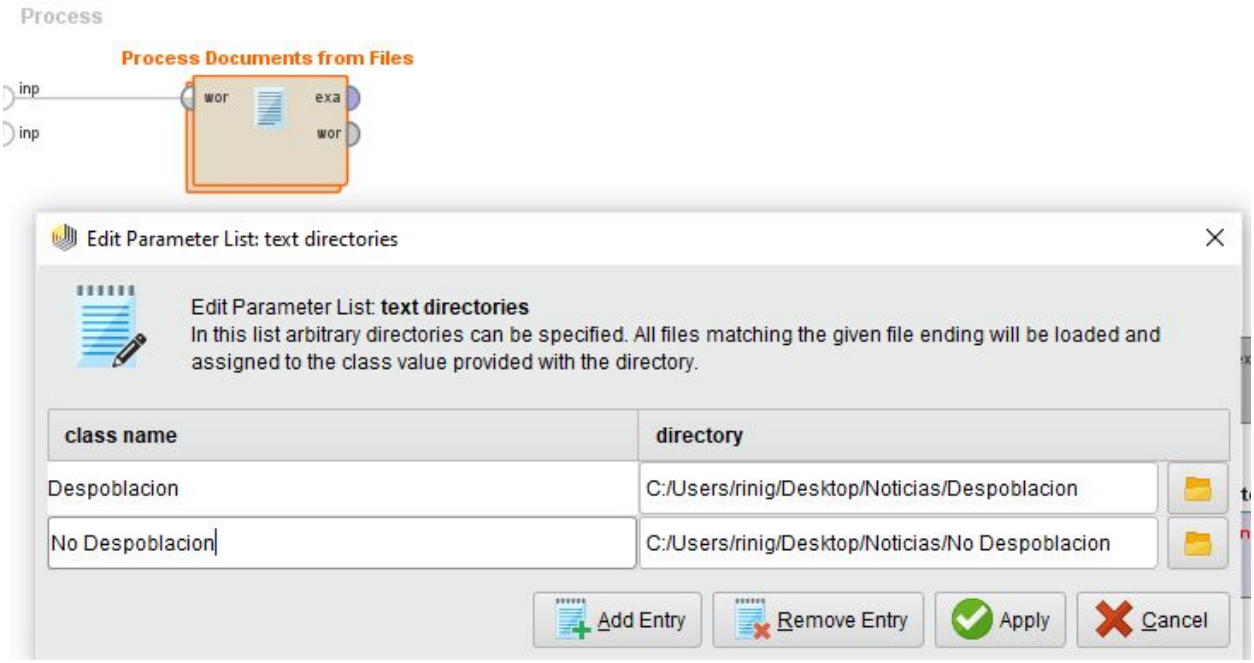
Contextualización

La idea de usar una plataforma de data science para nuestro proyecto como RapidMiner es, partiendo de la anterior etapa del proyecto en donde aplicando una estrategia ETL, donde escogimos noticias tanto generales como en torno al fenómeno de la “España vaciada” y tras “limpiarlas” en donde escogimos las partes que nos interesaban, aplicamos distintos algoritmos de aprendizaje automático para construir y entrenar un clasificador de noticias, automatizando todo el proceso de escoger una noticia para poder decidir si trata del tema de la “España vaciada” o no.

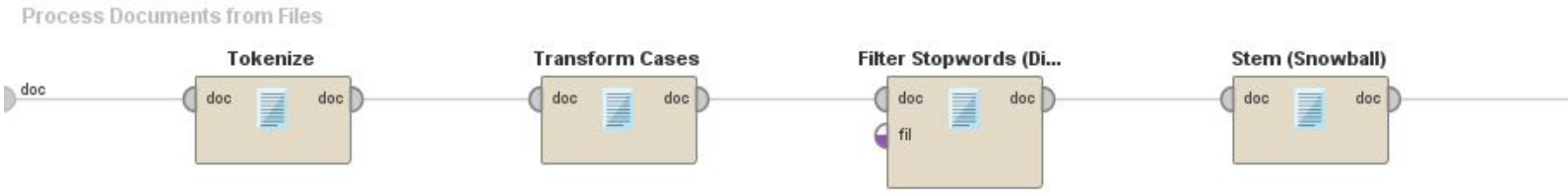
Descripción del proceso

La primera parte del proceso a seguir consiste en preparar los datos de forma previa, ya que si no lo hacemos no podremos aplicar los algoritmos de aprendizaje automático puesto que a pesar del caso en el que dos noticias hablen del mismo tema el algoritmo puede dar resultados totalmente incorrectos.
Por lo tanto, a cada documento se le aplica los mismos cuatro pasos: **tokenización** (para que la máquina entienda texto, necesitamos descomponer esa palabra de forma en que lo entienda), **cambiamos las palabras a minúsculas** (permite hacer las comparaciones más fácilmente), **eliminamos todas las "stopwords**" (palabras que no aplican ningún significado al texto) y **aplicamos stemming (radicalización) a las palabras** (cambiamos todas las palabras diferentes que representan lo mismo a un palabro común).

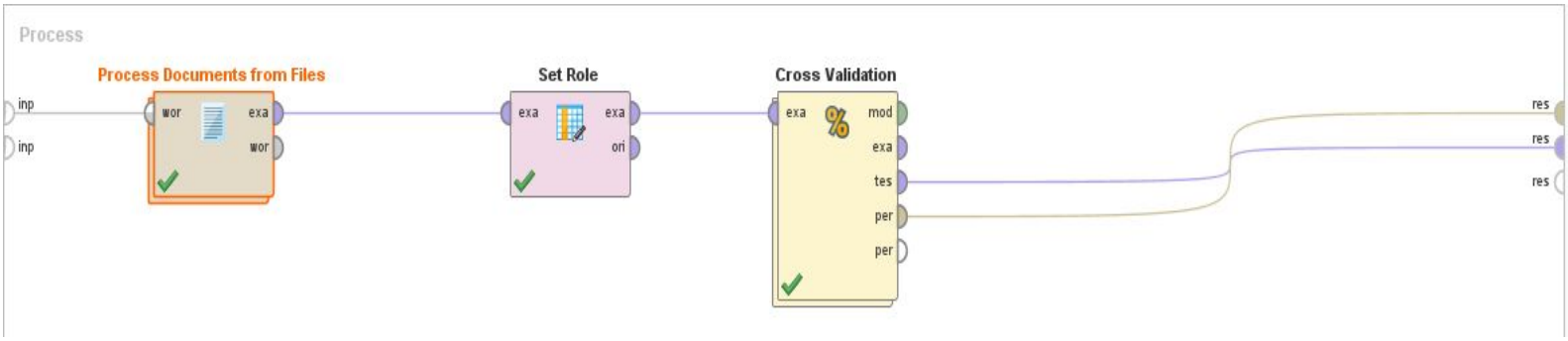
Esto se puede apreciar en las imágenes posteriores, donde al operador "Process documents from Files" le pasamos el corpus de noticias:



Y dentro podemos ver el proceso descrito anteriormente:

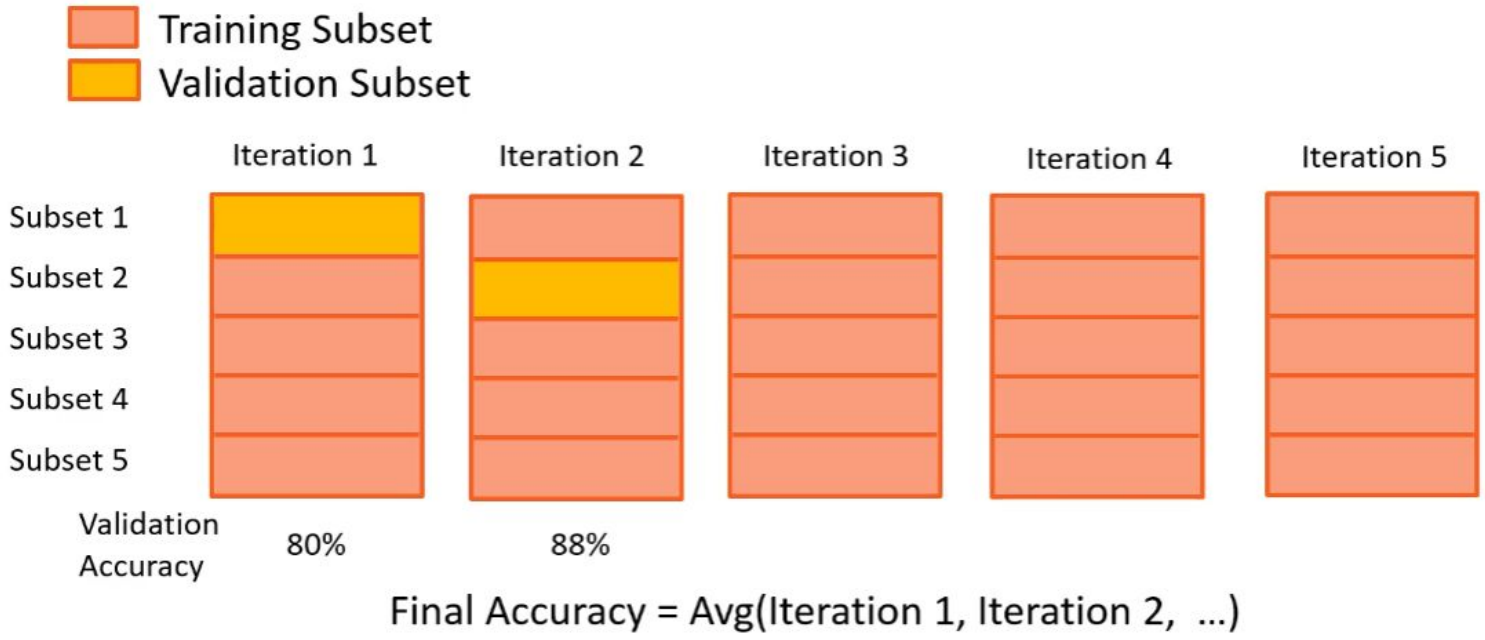


A continuación al procesar los documentos y obtener el “Example set” usando el operador “Set Role”, establecemos un “target” para que los algoritmos de aprendizaje automático puedan realizar sus predicciones y funcionar correctamente en donde (según nuestra label) nos diga en nuestro caso si lo predice cómo “Despoblación” o “No Despoblación”.
A este último operador lo conectamos con el operador final: “Cross Validation” el cual es utilizado para estimar cómo de preciso el algoritmo utilizado es en la práctica, y se divide en dos secciones: la sección de entrenamiento y la sección de testeo. Nuestra salida sería el resultado del modelo y el “Example set” como información adicional para comprobar cómo cada archivo se ha clasificado y las probabilidades de que sea “Despoblación” o “No Despoblación”.

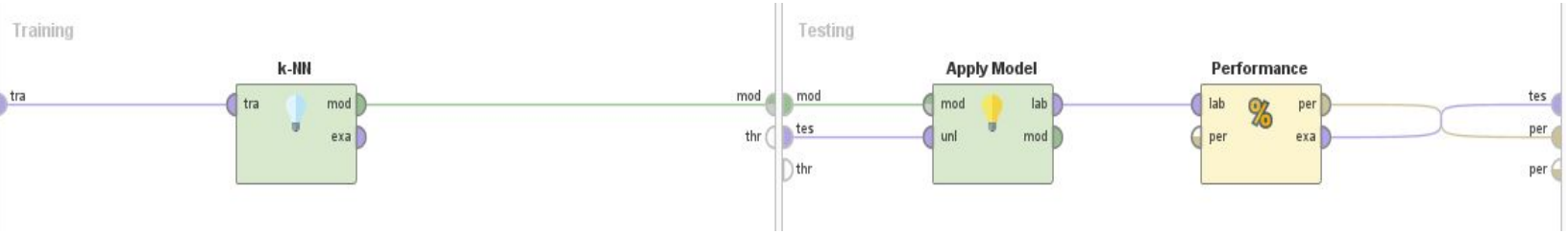


Antes de proceder con el resultado hay que tener en cuenta que Cross Validation contiene dos subprocesos los cuales han sido mencionados anteriormentes, por lo que para elegir cuánto porcentaje del set de datos utilizamos para entrenar y testear establecemos un número de “folds”, en nuestro caso puesto que queremos un 80/20 (80 para entrenar y 20 para testear) nos sirve con 5 “folds”.

Esto se debe a que divide el set de datos en 5 sub-set de datos más pequeños, y construirá el modelo (entrenamiento) en 4 de ellos dejando 1 para la parte de testeo, después itera seleccionando 4 diferentes sub-set de datos y igual que antes dejando 1 repitiendo hasta que haya hecho las 5 iteraciones o “folds” y midiendo la precisión del modelo usando una media de la precisión de cada iteración, mejor visto en la imagen posterior:



Observando cada subproceso podemos comprobar como en la parte de entrenamiento aplicamos el algoritmo de aprendizaje automático deseado (KNN, Naive Bayes, SVM etc.) insertando el “Training set” proporcionado por los pasos anteriores (Process Documents from Files y Set Role) y pasamos ese modelo a la zona de entrenamiento, donde aplicamos el modelo con los datos de testeo(“Test set”) y medimos cómo de bien rinde el modelo tras haberlo entrenado y testeado:



Resultados

Antes de comparar los diferentes resultados usando las matrices de confusión generadas por el operador “Cross Validation”, por fines demostrativos para el modelo k-NN muestro un ejemplo de un ExampleSet tras haber entrenado y testeado el set en donde se puede ver la predicción de cada noticia y su respectiva confianza:

Row No.	label	prediction(label)	confidence(Despoblacion)	confidence(No Despoblacion)	metadata_file	metadata_d...	metadata_p...
1	Despoblacion	Despoblacion	1	0	cope_despo5.txt	Nov 12, 2020 ...	C:\Users\rini...
2	Despoblacion	Despoblacion	1	0	lasexta_despo2.txt	Nov 12, 2020 ...	C:\Users\rini...
3	No Despobla...	Despoblacion	1	0	antena3_no_despo.txt	Nov 12, 2020 ...	C:\Users\rini...
4	No Despobla...	No Despoblacion	0.400	0.600	cope_no_despo.txt	Nov 12, 2020 ...	C:\Users\rini...
5	No Despobla...	Despoblacion	0.800	0.200	cope_no_despo5.txt	Nov 12, 2020 ...	C:\Users\rini...
6	No Despobla...	Despoblacion	0.800	0.200	lasexta_no_despo.txt	Nov 12, 2020 ...	C:\Users\rini...
7	Despoblacion	Despoblacion	1	0	cope_despo2.txt	Nov 12, 2020 ...	C:\Users\rini...
8	Despoblacion	Despoblacion	1	0	lasexta_despo.txt	Nov 12, 2020 ...	C:\Users\rini...
9	No Despobla...	No Despoblacion	0.400	0.600	antena3_no_despo2.txt	Nov 12, 2020 ...	C:\Users\rini...
10	No Despobla...	No Despoblacion	0.200	0.800	antena3_no_despo5.txt	Nov 12, 2020 ...	C:\Users\rini...
11	No Despobla...	No Despoblacion	0.400	0.600	cope_no_despo3.txt	Nov 12, 2020 ...	C:\Users\rini...
12	No Despobla...	Despoblacion	0.800	0.200	lasexta_no_despo3.txt	Nov 12, 2020 ...	C:\Users\rini...
13	Despoblacion	Despoblacion	1	0	antena3_despo3.txt	Nov 12, 2020 ...	C:\Users\rini...
14	Despoblacion	Despoblacion	1	0	cope_despo.txt	Nov 12, 2020 ...	C:\Users\rini...
15	Despoblacion	Despoblacion	0.800	0.200	cope_despo3.txt	Nov 12, 2020 ...	C:\Users\rini...

Como se puede observar en algunos casos hay un 100% de confianza en cuanto a la predicción mientras que en otros hay valores distintos, en donde la predicción depende del número mayor de la confianza del label “Despoblación” o “No Despoblacion”.

Tabla de confusión del modelo k-NN en donde tenemos 15 aciertos de “Despoblación” y 9 aciertos y 6 falsos aciertos de “No Despoblación”, además de una precisión del 80% +/- 18.26% en donde la precisión es buena pero la variación deja que desear (cuanto menos más fiable).

accuracy: 80.00% +/- 18.26% (micro average: 80.00%)

	true Despoblacion	true No Despoblacion	class precision
pred. Despoblacion	15	6	71.43%
pred. No Despoblacion	0	9	100.00%
class recall	100.00%	60.00%	

Table de confusión del modelo Naive-Bayes en donde tenemos 15 aciertos de “Despoblación” y 13 aciertos y 2 falsos aciertos de “No Despoblación”, además de una precisión del 93.3% +/- 14.91% en donde la precisión es buena pero la variación al igual que el modelo anterior es bastante mala.

accuracy: 93.33% +/- 14.91% (micro average: 93.33%)

	true Despoblacion	true No Despoblacion	class precision
pred. Despoblacion	15	2	88.24%
pred. No Despoblacion	0	13	100.00%
class recall	100.00%	86.67%	

Table de confusión del modelo Decision Tree en donde tenemos 10 aciertos y 5 falsos aciertos de “Despoblación” y 15 aciertos de “No Despoblación”, además de una precisión del 83.3% +/- 0.00% en donde la precisión es buena y la variación es muy buena (demasiado buena).

accuracy: 83.33% +/- 0.00% (micro average: 83.33%)

	true Despoblacion	true No Despoblacion	class precision
pred. Despoblacion	10	0	100.00%
pred. No Despoblacion	5	15	75.00%
class recall	66.67%	100.00%	

Table de confusión del modelo Random Forest en donde tenemos 12 aciertos y 3 falso aciertos de “Despoblación” y 11 aciertos y 4 falsos aciertos de “No Despoblación”, además de una precisión del 76.67% +/- 19.00% en donde la precisión es relativamente buena pero la variación es bastante mala.

accuracy: 76.67% +/- 19.00% (micro average: 76.67%)

	true Despoblacion	true No Despoblacion	class precision
pred. Despoblacion	12	4	75.00%
pred. No Despoblacion	3	11	78.57%
class recall	80.00%	73.33%	

Table de confusión del modelo SVM en donde tenemos 15 aciertos de “Despoblación” y 9 aciertos y 6 falsos aciertos de “No Despoblación”, además de una precisión del 80% +/- 18.26% en donde la precisión es relativamente buena pero la variación es bastante mala al igual que el modelo anterior.

accuracy: 90.00% +/- 14.91% (micro average: 90.00%)

	true Despoblacion	true No Despoblacion	class precision
pred. Despoblacion	13	1	92.86%
pred. No Despoblacion	2	14	87.50%
class recall	86.67%	93.33%	

Conclusiones

Un criterio crítico para escoger el mejor algoritmo de aprendizaje automático para nuestro caso (a parte de la precisión) es la variación en donde cuanto más pequeño sea el número más estable será el modelo, por lo que mientras la precisión sea lo suficientemente alta y la variación sea lo menor posible ese será el mejor modelo.
Es por esto que el mejor modelo para mi es Naive Bayes seguido cercanamente de Random Forest, ya que a pesar de tener una variación alta al tener una precisión tan alta sigue siendo más efectivo que Random Forest.

A pesar de ello, al tener un data set bastante pequeño (solo 30 noticias) estos números podrían cambiar bastante si usasemos un data set mas grande (lo cual deberíamos) y en cuanto a los “folds” del operador “Cross Validation” es complicado obtener un número específico ya que puede ser que para nuestro caso y nuestro set de datos iria mejor con menos o más “folds”, aunque el último caso sólo se daría si el set de datos es mucho más grande (en torno a los cientos de miles), aun asi con 5 folds nos da para resultados precisos por lo general.