UNIVERSIDAD
ANDRES BELLO

Universidad Andrés Bello
Facultad de Ingeniería

Ingeniería Civil Informática

TAREA2

APRENDIZAJE SUPERVISADO

SISTEMAS INTELIGENTES

DAVID ANDRÉS MOLINA GARRIDO

Profesor: Alejandro Figueroa
Ayudante: Jean Contreras

Santiago - Chile.

Septiembre, 2017.

Introducción

El siguiente informe consta de aprender un modelo de predicción para posteriormente analizarlo basándonos en diferentes métricas. El modelo se realiza sobre un conjunto de perfiles de usuarios provenientes de plataformas (cQA), las cuales tienen como propósito responder a preguntas particulares a los usuarios que la componen. Estos perfiles cuentan con un apartado para la descripción del usuario, y estas descripciones fueron etiquetadas según las siguientes clases, a partir de la ubicación geográfica que nos provee el usuario:

-**USA only** (Si está relacionado con sólo lugares dentro de USA).
-**Non-USA** (Si está relacionado con sólo lugares fuera de USA).
-**World** (Si está relacionado con lugares dentro y fuera de USA).
-**Undetermined** (Si no provee ninguna pista de algún lugar geográfico relacionado al usuario).

Posteriormente, la información que nos provee el usuario en su descripción es representada por un vector, utilizando las palabras contenidas en todo el set de perfiles como los atributos de este vector.

Para este trabajo se utilizó la herramienta de soporte vectorial SVM Multiclass, esta herramienta utiliza los vectores para representar las descripciones de los perfiles como puntos en el espacio separando las clases lo más posible. En otras palabras, la SVM o Maquina de Soporte Vectorial construye un hiperplano que es utilizado para problemas de clasificación en donde una buena separación entre las clases implica una clasificación correcta. Para comparar distintos resultados se hizo un Cross-Validation entre diez grupos de datos distintos, para así obtener los errores, compararlos y analizar sus diferencias o similitudes.

Descripción del Problema

Primeramente, para generar los vectores mediante las descripciones etiquetadas en el trabajo anterior, generaremos una bolsa de palabras de todas nuestras descripciones en los perfiles de usuarios, ésta contará con lo siguiente:

<target> <feature>:<value> <feature>:<value> ... <feature>:<value> # <info>

Imagen1: Formato de la bolsa de palabras.

Donde:

1. Target: indica la clase de la pregunta correspondiente al vector
2. Features: indica el id de la palabra
3. Value: indica la frecuencia de la palabra con ese id, en la descripción correspondiente al vector.

Entonces, para cada descripción (2000), tendremos ese formato en un solo archivo (modelo.dat).

Para que nuestra Maquina de Soporte Vectorial SVM aprenda, es necesario que se generen un entrenamiento previo.

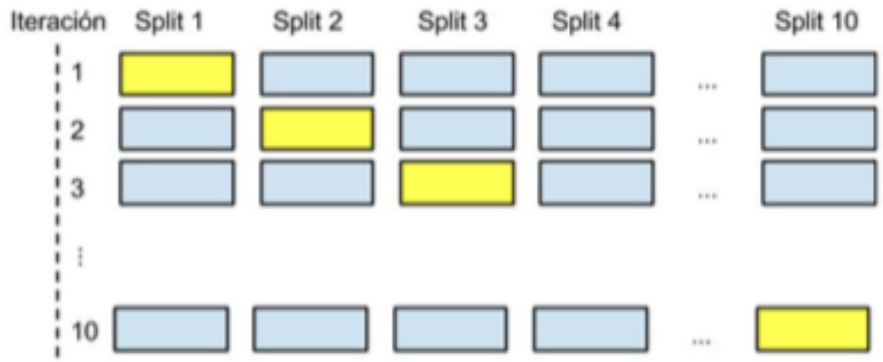


Imagen2: imagen referencial de un ejemplo de Cross-validation.

Entonces, con el propósito de que nuestra SVM aprenda se segmentará la información en diez partes, generando diez sets de entrenamiento diferentes, en los cuales 1/10 partes (diferentes) faltarán en cada set de entrenamiento (Imagen 2).

Para generar los modelos utilizaremos SVM Multiclass, la cual acompañada de una validación cruzada (Cross-validation) de los datos generaremos nuestro modelo para cada parte del set que no fue incluida (Split).

Luego del entrenamiento a nuestra SVM, ésta generará una predicción a partir de cada Split, ósea, de cada segmentación que hicimos. A partir de esta predicción mediremos el desempeño mediante una métrica llamada MRR que corresponde al ranking de media recíproco, que evaluará los modelos mediante la probabilidad de error en cada uno de los etiquetados. Su ecuación es:

$$\mathrm{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{\mathrm{rank}_i}.$$

Imagen3: Ecuación de la métrica MRR

Donde:

- Q es el número total de vectores (2000).

-Rank, será la posición de un atributo del vector en el archivo de predicción.

Posteriormente basándonos en diferentes métricas analizaremos los errores del clasificador y del etiquetado del conjunto de datos, cómo poder resolverlos y/o mejorarlos y qué patrones encontramos en ello.

Se utilizarán curvas para graficar los datos y estimadores para poder analizar mejor la información.

Análisis del Resultado

Para que en las distintas clases de las preguntas sea viable su modelamiento, se le entregó un valor numérico que las represente, estos fueron:

Clase	Número
Non-USA	1
USA only	2
Undetermined	3
World	4

Imagen 4: Tabla de clase y su número asignado.

Desempeño del Clasificador

Mediante el Clasificador obtuvimos el siguiente desempeño para los diez Split mediante su propio entrenamiento:

Error del Clasificador			
Split (Subset)	% Error	Cantidad Correctas	Cantidad Incorrectas
S1	30	140	60
S2	33,5	133	67
S3	28	144	56
S4	30,5	139	61
S5	33	134	66
S6	35	130	70
S7	28,5	143	57
S8	29	142	58
S9	34,5	131	69
S10	35	130	70

Imagen 5: Tabla Error del Clasificador (promedio splits).

Con estos datos, el clasificador nos dice que porcentaje tiene de error cada Split que le pasamos, este error viene de la clase que nosotros etiquetamos vs la que él clasificó. Comparándolo las clases que nos entrega en cada archivo de predicción por Split con nuestras clases, nos damos cuenta de que ese porcentaje es equivalente, y que a su vez es proporcional por cada subset de datos.

Entonces, podemos decir que el clasificador tiene un bajo porcentaje al principio, y que éste no va aumentando considerablemente, sino que va fluctuando en todo el set entre

[28-35] porciento de error, por lo cual las etiquetas fueron consistentes. Además, se denota la aleatoriedad en la manera en que iban apareciendo cada una de ellas para su posterior etiquetado.

Por otro lado, vemos que el clasificador tuvo buenos resultados, por lo tanto, aprendió de manera correcta, esto ya que "el núcleo" de su aprendizaje se debía a ubicaciones geográficas, ósea, a palabras específicas que como vemos se repetirían más adelante en otros perfiles. Es por esto, por lo que más adelante veremos que de ciertas clases aprendió de mejor manera y que en ciertas ubicaciones pasó lo mismo.

Los patrones que se detectan a través de los errores son que hay una variabilidad bastante uniforme, es decir, cada subset tuvo datos de diferentes clases, y por ello la cantidad de correctas e incorrectas no varían mucho en cada subset.

Una manera de poder mejorar los errores sería incrementando el set de datos, de esta manera el clasificador podría aprender de más instancias y "pulir" sus predicciones, además se podrían entregar datos de cada clase con la misma distribución de ocurrencias.

Matriz de Confusión

La Matriz de Confusión es una herramienta que permite la visualización del desempeño de un algoritmo que se emplea en aprendizaje supervisado. En este caso analizaremos cada una de las 4 clases, según nuestro etiquetado vs el etiquetado que ha generado la Maquina de Soporte Vectorial SVM.

	Predicted	
	Positive	Negative
Actual True	**TP**	**FN**
Actual False	**FP**	**TN**

Imagen 6 : Matriz de Confusión.

Donde:

TP: True Positive: Era verdadera y fue positiva (por el clasificador).
FN: False Negative: Era verdadera, pero no fue positiva (por el clasificador).
FP: False Positive: Era Falsa y fue positiva (por el clasificador).
TN: True Negative: Era Falsa y no fue positiva (por el clasificador).

Las 4 Matrices son:

True: Mis datos del etiquetado.
Predicción: Los datos del clasificador.

	Predicción	
True	**Non-USA**	Otras
Non-USA	434	145
Otras	172	1249

	Predicción	
True	**USA only**	Otras
USA only	675	142
Otras	299	884

	Predicción	
True	**Undetermined**	Otras
Undetermined	255	272
Otras	147	1326

	Predicción	
True	**World**	Otras
World	2	75
Otras	16	1907

Las 2 matrices que tuvieron mejor desempeño fue **Non-USA** y **USA only**, de estas podemos extraer fácilmente que los True Positive y los True Negative el clasificador les acertó con una alta tasa en contraste con los False Positive y los False Negative . En las otras dos clases esto no sucede, vemos que el clasificador erró con las falsas y esto tendrá consecuencias en las métricas que analizaremos más adelante.

La matriz que más tuvo equivalencias entre nuestros datos y los del clasificador fue **USA only**, y la que menos, fue **World**, por otro lado, esto no nos dice mucho a simple vista, ya que también depende de cuantos datos tiene cada categoría, ósea su distribución, por lo tanto, en la siguiente tabla se muestra la distribución de cada clase y la probabilidad de la predicción del SVM.

	Clase	Cantidad total Clase	Cantidad Acertada por SVM	Porcentaje de Acierto
1	Non-USA	579	434	0,749568221
2	USA only	817	675	0,82619339
3	Undetermined	527	255	0,483870968
4	World	77	2	0,025974026
		2000		

Imagen 7: Distribución y Acierto por Clase

Observando ahora con probabilidades, tenemos que la probabilidad más baja efectivamente la tiene **World** ya que en poquísimas ocasiones fueron iguales las etiquetadas por nosotros vs la del Clasificador (2 / 77 = 0.026). Por otra parte, la de mayor probabilidad ahora se ve claramente que fue **USA only** (675/817 = 0,83).

Esto resulta en dos hipótesis (que podrían ser las dos):

La primera es que se puede notar la diferencia entre estas dos clases (**World y USA only)** en que su distribución en el set de datos es muy grande, una es la mayor y la otra la menor. Es por esto, que el clasificador aprende de mejor manera en una clase determinada cuando ésta posee mayores ocurrencias.

La segunda es que como se vio en el trabajo anterior la clase que tenía más lugares repetidos fue **USA only** y la clase que tenía más diversidad era **World**. Así, la maquina aprendió con mayor facilidad de datos más frecuentes.

Por otro lado, puede que ambas hipótesis sean correctas o se complementen dependiendo de cómo es la forma científica en la que calcula el clasificador.

MRR

La métrica MRR corresponde al ranking de media recíproco, donde evalúa las predicciones de nuestra SVM con nuestro previo etiquetado para detectar el grado de error en esa Clase, como vemos en el siguiente ejemplo:

Query	Results	Correct response	Rank	Reciprocal rank
cat	catten, cati, **cats**	cats	3	1/3
torus	torii, **tori**, toruses	tori	2	1/2
virus	**viruses**, virii, viri	viruses	1	1

Imagen 8: Ejemplo MRR

Entonces entre más alejada esté nuestra Clase de la que predijo nuestro Clasificador (que siempre estará en la primera posición), el porcentaje será menor y nuestro error mayor.

De los resultados obtenidos una porción de estos fue:

INPUT Cat por mí	Ordenado				Resultados del Clasificador			
2	23,5365	18,337027	-14,786514	-27,087014	18,337027	23,5365	-14,786514	-27,087014
2	43,706488	-5,084911	-17,621454	-21,000123	-21,000123	43,706488	-5,084911	-17,621454

Imagen 9: Ejemplo MRR 2

Entonces sacado este ranking reciproco por cada pregunta, sacamos el promedio de ésta y nos da como resultado:

	Donde quedó	MRR	MRR TOTAL
23,536500	1	1	0,81491667
43,706488	1	1	

Imagen 10: Ejemplo MRR 3

Nos dio como resultado 0.815 de MRR total, lo que significa que tenemos una variabilidad bastante buena. Los datos totales acertados por la SVM fueron 1366 (ósea, la suma de todos los TP), esto da un porcentaje de un 1366/2000 = 0.683, lo que ya era bueno, pero ahora tomando además un cierto porcentaje de las clasificadas como "cercanas" el porcentaje sube y nos muestra de mejor manera cuan cerca estuvo el clasificador. Además, nos sirve para comparar entre otros clasificadores y contrarrestar con mayor exactitud.

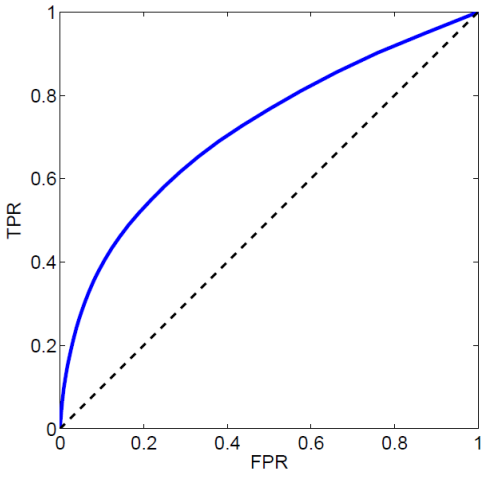
Curvas ROC

La curva ROC (acrónimo de Receiver Operating Characteristic, o Característica Operativa del Receptor) es una representación gráfica de la sensibilidad (TPR) frente a la especificidad (FPR) para un sistema clasificador binario según se varía el umbral de discriminación. En este caso variará en función de si nuestras clases van tomando los siguientes parámetros:

$$TPR = \frac{True\ positives}{True\ positives + False\ negatives}$$

$$FPR = \frac{False\ positives}{False\ positives + True\ negatives}.$$

Imagen 11: Parámetros Curva ROC.



Asumiendo cada variable (clase) como positiva, y generando un Ranking, el cual se obtiene ordenando los casos de prueba en orden decreciente de acuerdo a lo que asignó el clasificador, el desempeño de nuestro clasificador fue el siguiente:

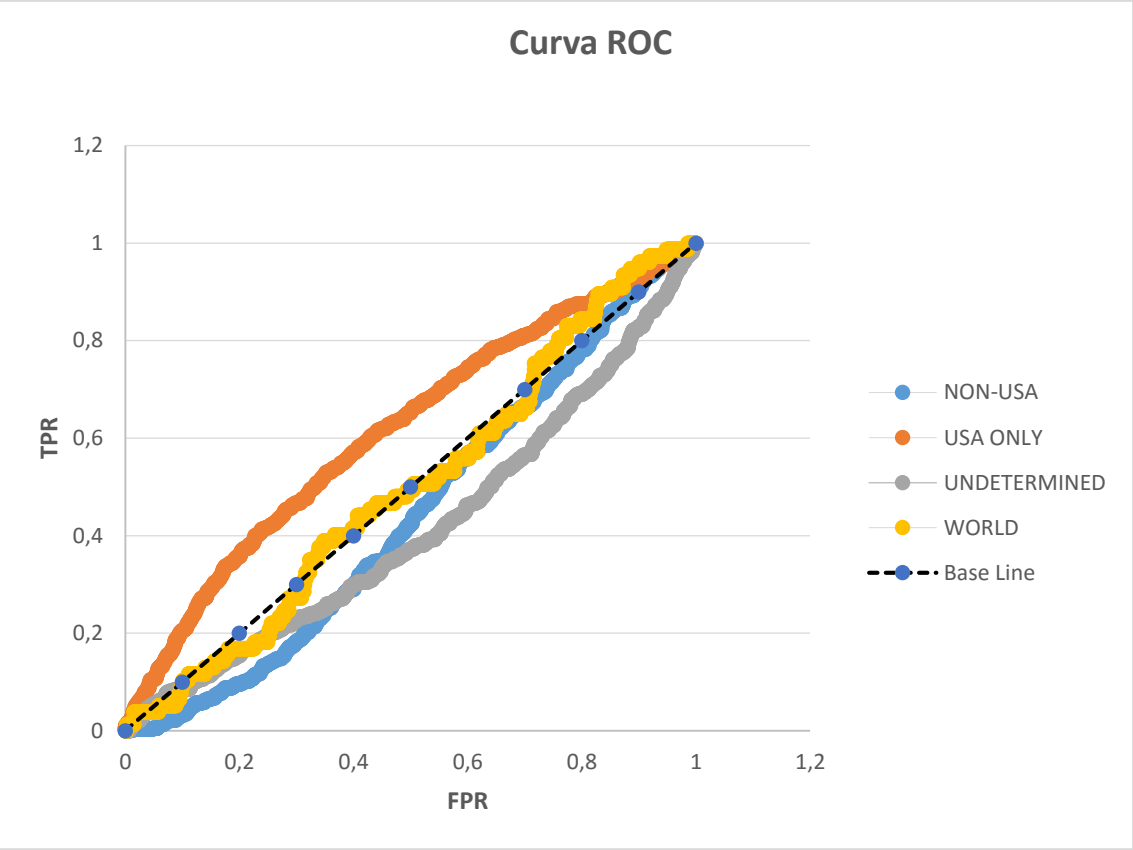


Imagen 12: Curva ROC por Clase.

Como vemos en la *Imagen 12* a simple vista la mejor curva (la de mayor área), es la clase **USA only** , esto se debe a que el resultado del clasificador acertó de mejor manera, por lo tanto, hay más verdaderos positivos en promedio que el resto. Por otro lado, la clase de peor aprendizaje fue **World**, ya que está más cercana a Base Line, es decir, el clasificador no discierne entre los verdaderos positivos y los falsos positivos, no predice bien.

En el caso de **Undetermined** y **Non-USA** vemos que éstas están bajo la Base Line. Esto se debe cuando el clasificador clasifica de manera inversa, pero si posee la capacidad de predecir datos, es por esto que la peor es **World**.

Además, el AUC entre estas clases claramente es **USA only**, lo cual se verificó además en la matriz de confusión al ver los números de los verdaderos positivos del eje Y.

Curva LIFT

Método para la etiquetación de dos clases (cada clase como positiva, frente al resto), nuestros casos se dividen en 10 paquetes (eje x), y nuestros casos positivos en relación al paquete correspondiente corresponde al otro eje (eje y).

Al igual que la Curva ROC la línea perfectamente diagonal nos indica que no hubo aprendizaje, es decir, que los datos positivos están distribuidos de la misma manera para cada paquete. Además se ocupó el mismo Ranking de ordenamiento que en la curva ROC.

Las Curvas Lift correspondiente a las clases son:

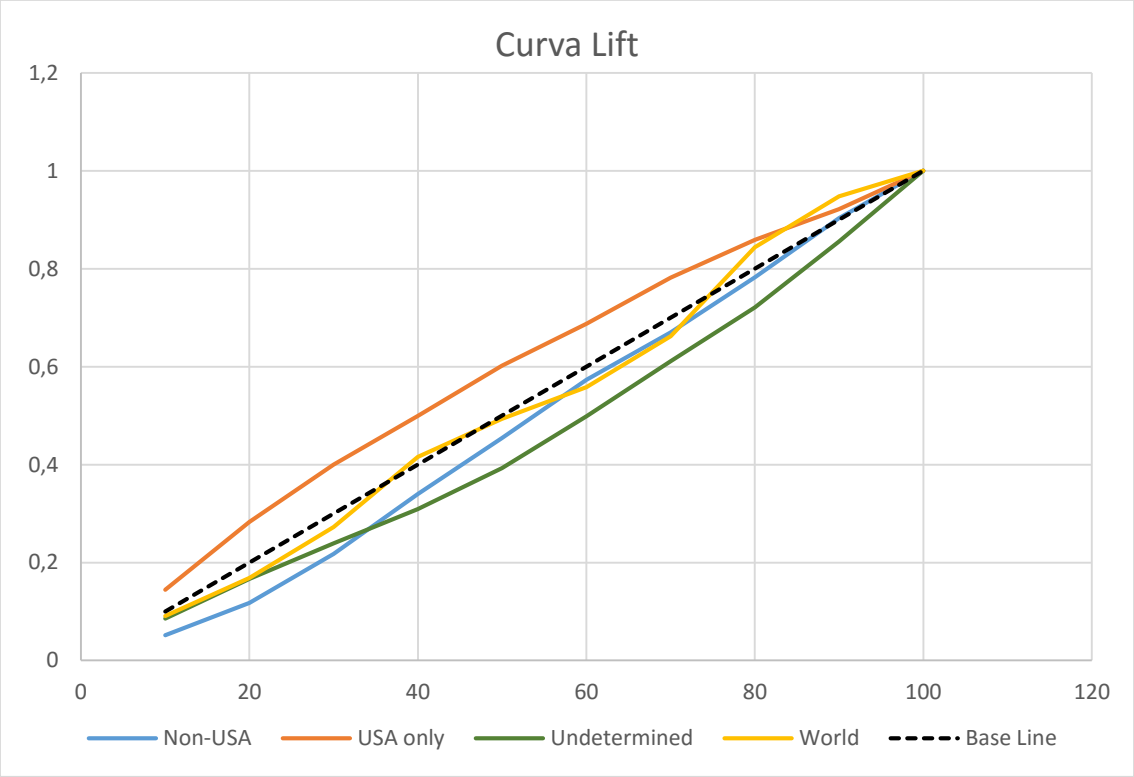


Imagen 13: Curva Lift por Clase.

Como vemos en la gráfica, las clases tuvieron un porcentaje regular en cada paquete de datos, generando una curva casi lineal en la mayoría. Además podemos apreciar que la clase **World** sigue (como en la curva ROC) más cercana a Base Line, por lo cual no tuvo un buen aprendizaje.

En caso de **USA only**, vemos que al comienzo tiene mayor porcentaje que las demás clases por lo que la curva comienza con mayor ventaja que el resto, y por lo tanto hace que al final tenga una mayor área.

F(1)-score y otras Métricas

F(1)-Score es una métrica, la cual vendría siendo la media armónica entre la Precisión y el Recall.

$$precision = \frac{TP}{TP + FP} \qquad recall = \frac{TP}{TP + FN}$$

$$F_1 = 2 \cdot \frac{1}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

Imagen 14: Métricas .

Como vemos en la imagen anterior Precisión y Recall calculan los verdaderos positivos en base a conjuntos diferentes, es por esto que F1-Score viene a ser una media armónica. Para que esta métrica dé un buen resultado, Precisión y Recall deben de ser altos.

Non-USA	
Precision	0,71617162
Recall	0,74956822
F1-Score	0,73248945

USA only	
Precision	0,69301848
Recall	0,82619339
F1-Score	0,75376884

Undetermined	
Precision	0,63432836
Recall	0,48387097
F1-Score	0,5489774

World	
Precision	0,11111111
Recall	0,02597403
F1-Score	0,04210526

Nuevamente se da la misma tónica, el mejor desempeño en la métrica F1-score la obtuvo **USA only**, y el peor **World**. Aquí podemos ver la gran diferencia entre estas dos clases, basándonos en los datos anteriores de nuestra matriz de confusión podíamos prever estos resultados, ya que los TP (verdaderos positivos) de cada clase con respecto a su totalidad eran muy diferentes, y en la clase **World** solo acertó a 2 de 77.

Conclusión

Como conclusión después de haber presentado el análisis de nuestros datos, podemos decir que a grandes rasgos el clasificador tuvo un buen aprendizaje de las descripciones en los perfiles de los usuarios. Esto se denota claramente ya que, tomando todas las clases, nuestro clasificador tuvo un buen conjunto de acierto en los TP (verdaderos positivos). Pero de manera más específica, vemos que cada clase tuvo predicciones diferentes de parte de nuestra SVM, y algunas de ellas contrastan bastante.

Podemos ver que la clase con mayor desempeño en distintas métricas y mediciones fue **USA only** y la peor fue **World**. Esto se debe a que de la clase **USA only** obtuvo un mayor aprendizaje. Primeramente, porque tuvo más datos de los cuales aprender, y por otro lado las ubicaciones en esta clase eran más recurrentes. En contraste vemos que la clase **World** carecía de altas instancias y además eran bastante variables sus ubicaciones

Además, la matriz de confusión nos muestra de fácil manera como fue la predicción de nuestra SVM y como mitigar los errores de manera focalizada.

Si bien es cierto la clasificación no fue exacta, hay algunos puntos con los cuales se podría mejorar la clasificación de la SVM a partir de los datos de entrada:

-Incrementar nuestro Set de datos que entregamos y entrenamos a la SVM.

-Incluir una distribución equitativa de Clases en el set de datos.

-Incrementar los Stop-Words para nuestra clasificación y eliminarlos.

Por último, podemos decir que nuestra clasificación fue buena y que, si nos concentráramos en depurar ciertos errores en los datos de entrada, la SMV podría predecir con mayor alcance nuestros datos. Entonces, como la SVM está enfocada ciertamente a detectar frecuencias y a encontrar patrones, por ende, los sets de entrenamiento que le entreguemos deben ser lo más parecidos, obviamente no iguales en sus datos, sino que parecido en el sentido de la ocurrencia de las clases.