Santiago Ramírez 201125182

Carlos Daniel López 201126513

**Tarea 3 Machine Learning**

**Punto 1**

Para realizar la separación en las clases dadas por medio de una “bolsa de palabras”, se utilizaron las herramientas brindadas por el toolbox de scikitlearn en Python. Primero fue necesario filtrar las categorías que no pertenecieran a alguna de las clases, dejando solo ‘entertainment’, ‘world’ y ‘sci\_tech’. Luego se asignaron a su clase correspondiente. La bolsa se generó utilizando todas las noticias de las clases y pasándolas por la función “CountVectorizer()”. Esta función genera una biblioteca con todas las palabras que están en las noticias y realiza un conteo de estas en cada noticia. Por ejemplo, para la primera noticia se obtiene un resultado así:

(0, 10950) 1

(0, 7247) 1

(0, 18788) 1

(0, 12548) 1

(0, 14851) 1

(0, 1006) 1

(0, 14047) 1

(0, 1854) 1

(0, 20595) 1

(0, 13214) 1

(0, 8072) 1

(0, 16957) 1

(0, 9946) 1

(0, 10124) 1

Donde el primer número representa la noticia, el segundo numero representa el índice de la palabra dentro de la bolsa y el tercer numero muestra cuantas veces se encuentra la palabra en la noticia.

Una vez creada la bolsa con sus valores asociados se procedió a separar lo datos de prueba y de entrenamiento, usando la función “train\_test\_split” del toolbox mencionado. Esta función separa un porcentaje de datos de prueba (en este caso 10%) y el resto lo asigna para el entrenamiento del modelo. Cabe resaltar que aunque siempre se escoge el mismo porcentaje de datos, los datos cambian de manera aleatoria cada vez que se ejecuta la función, de manera que las diferentes pruebas realizadas se hicieron con distintos datos de prueba y de entrenamiento.

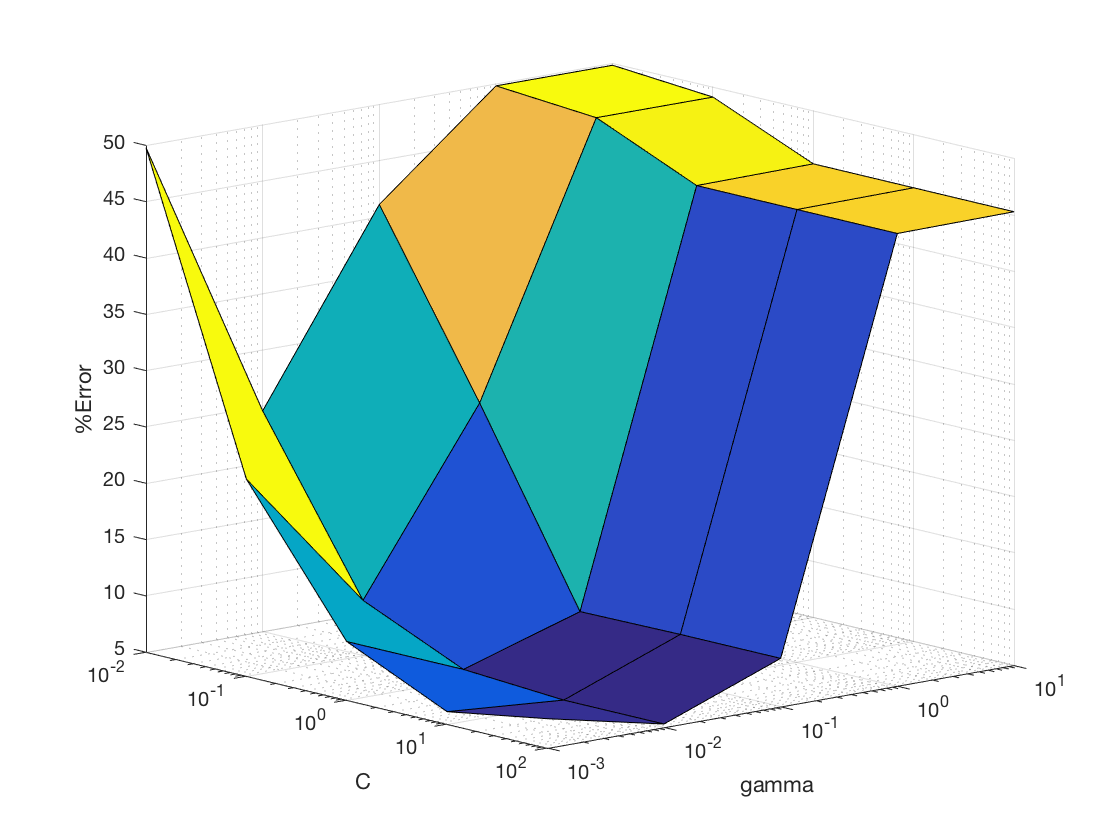
Dado que el enunciado pedía usar un kernel gaussiano se utilizó la función “SVR” con kernel de tipo “RBF”. Este tipo se support vector machine utiliza kernels definidos por la función:

De manera que para calcular el separador optimo , el modelo debe solucionar:

Donde es la función de transformación al nuevo espacio, son los coeficientes del lagrangiano, SV es el set de vectores de soporte y es la clase de cada dato . El valor del bias optimo del clasificador estará dado por:

El toolbox utilizado permite modificar las variables ‘C’ y ‘gamma’ dentro del modelo. Según la documentación, el parámetro ‘gamma’ es el inverso del radio de influencia que se le asigna a los vectores de soporte[[1]](#footnote-1), o en otras palabras es el inverso de la desviación estándar de la función gaussiana . De esta manera, un valor alto de ‘gamma’ significará un radio de influencia pequeño. Por otra parte se tiene que el parámetro ‘C’ permite clasificar mal algunos datos con el propósito de hacer la superficie de decisión más sencilla. Un valor pequeño de ‘C’ resulta en superficies más simples con la posibilidad de clasificar erróneamente más datos de entrenamiento. Las predicciones realizadas por el modelo, se pasaron por una función de activación escalar centrada en 0.5 ya que los resultados se encontraban todos entre 0 y 1.

Intuitivamente se asignó un valor pequeño de ‘gamma’ y un valor moderado en ‘C’. En consecuencia con un gamma=0.1 y C=10 se obtuvo un 92.402% de datos de prueba bien clasificados. A pesar de ser un resultado satisfactorio se realizaron varias pruebas variando estos parámetros, los resultados se ven en la gráfica a continuación.



Grafica 1. Relación C vs gamma vs %error

Con la información obtenida se volvió a intentar entrenar el modelo con C=10 y gamma=0.01 y se obtuvo el 95.8098% de datos bien clasificados. Lo que representó una mejora evidente.

1. http://scikit-learn.org/stable/auto\_examples/svm/plot\_rbf\_parameters.html [↑](#footnote-ref-1)