Ahora ustedes van a estudiar otro método de aprendizaje de máquina que se llama Regresión logística. Por favor estudian desde Lecture 6.1 hasta Lecture 6.7.

También lo que vamos a hacer es programar la Regresión logística y para ello vamos a usar un corpus que se llama "SMS_Spam_Corpus_big", está en la carpeta "Corpus" en Drive, es un archivo de texto.

El corpus contiene mensajes que pertenecen a dos categorías: spam y no-spam. El no-spam se llama "ham". Entonces todos los mensajes están etiquetados como "spam" o "ham".

Los mensajes están en inglés. Un mensaje es una línea del archivo. La última palabra de la línea es la etiqueta de la categoría. La etiqueta está separada del mensaje por una coma. Entonces la coma y la etiqueta no son partes del mensaje.

Como para el programa de la Regresión lineal, hay que generar dos estructuras: la matriz X (de mensajes) y el vector y (de etiquetas).

Primero vamos a hablar sobre la matriz X.

Antes de pasar los textos a una representación numérica, los vamos a normalizar:

1. Pasar lo todo a minúsculas, tokenizar.
2. Etiquetar con POS. Como el texto está en inglés, vamos a usar el etiquetador POS de NLTK. Véanlo en la pág. 201 del PDF del libro "Natural Language Processing with Python".
3. El texto etiquetado hay que lematizar con el lematizador de NLTK, véanlo en la pág. 147-148 del PDF del libro "Text Analytics with Python".
4. No vamos a limpiar los tokens, no vamos a quitar los símbolos especiales y los de puntuación porque si ustedes observen los textos, estos símbolos son bastante frecuentes en mensajes de spam en comparación con los textos de ham, entonces ellos servirán como características para distinguir spam de no-spam.

Después de la normalización, vamos a vectorizar los textos utilizando frecuencia normalizada (probabilidad). Lo vamos que hacer como antes: primero hay que obtener el vocabulario de TODOS los textos y luego generar un vector para cada texto, así lo hacíamos en las clases anteriores.

Los vectores vamos a meter como columnas en la matriz X, también vamos a hacer el vector y, representando "spam" como el número 1 (ojo: 1 como número, no como cadena o caracter) y "ham" como el número 0.

Luego vamos a mezclar aleatoriamente los datos como en la Regresión lineal y dividirlos en conjunto de entrenamiento (70%) y en conjunto de prueba (30%).

Entrenamos el modelo como en la Regresión lineal, imprimiendo cada 50ª iteración el valor de la función de costo. Después, aplicarán el modelo entrenado al conjunto de prueba e imprimirán el valor de la función de costo para este conjunto.

El programa y los resultados (captura de pantalla con los valores de la función de costo) me van a entregar hasta el martes 7 de abril.