Disclaimer: Notebooks translated to Spanish are distributed as an optional aid to assist in your learning and comprehension. We make no guarantees that the translations are completely accurate nor that the translated code blocks will run properly.

Homework must be submitted in *English*, Spanish submissions will not be graded.

Generando características útiles para un análisis más profundo de las calificaciones en Amazon

Introducción

Contexto Empresarial. Usted es un consultor de negocios y han llegado unos clientes nuevos que están interesados en analizar las reseñas de sus productos en Amazon (no en Yelp). Quieren responder a preguntas de negocios como: "¿Cuáles son los factores más importantes que están detrás de las críticas negativas?", "¿Ha habido grandes cambios en la satisfacción de los clientes/reseñas de los clientes a lo largo del tiempo?", etc.

Problema de negocios. Su principal tarea es **explorar los datos y utilizar los resultados de su investigación para diseñar características relevantes que puedan facilitar el análisis posterior y la construcción de modelos**.

Contexto analítico. El conjunto de datos proporcionado es un gran cuerpo de reseñas relacionadas con películas y programas de televisión dejadas en Amazon entre 1996 y 2014. Al explorar nuestro conjunto de datos, nos encontraremos rápidamente con un problema familiar que discutimos en el caso anterior: la palabra "bueno" es una de las palabras más importantes en las reseñas tanto positivas *como* negativas. Por lo tanto, debemos desarrollar métodos para poner "bueno" en el contexto apropiado.

Cargando los datos

Utilizamos un conjunto de datos de alrededor de 37.000 reseñas de vídeo de Amazon Instant Video y 1.700.000 reseñas de cine y televisión, todas obtenidas del sitio web: http://jmcauley.ucsd.edu/data/amazon/. Tenga en cuenta que hay conjuntos de datos mucho más grandes disponibles en el mismo sitio. Podemos esperar resultados mejores y más consistentes en conjuntos de datos más grandes (como reseñas de libros). Note que estos conjuntos de datos están comprimidos (gzipped), y están en formato JSON, con cada línea representando una reseña y cada línea siendo un objeto JSON.

Empezamos cargando el conjunto de datos:

[ ]:
```
%matplotlib inline
```

[ ]:
```
%%time
import gzip
import json
import string

import nltk # importa la caja de herramientas de lenguaje natural
import pandas as pd
import plotly

nltk.download('punkt')


# podemos decirle a Pandas que nuestro archivo está en formato gzip y Pandas lo puede␣
 ↪descomprimir por nosotros
```

```
# usamos `lines=True` para indicar que cada línea del archivo es un objeto JSON
instant_video = pd.read_json("reviews_Amazon_Instant_Video_5.json.gz", lines=True,␣
↪compression='gzip')


# ----------
# El archivo de películas y televisión es muy grande. Si tiene problemas para␣
↪cargarlo, puede cargar solo las primeras
# 100,000 reseñas usando 'chunksize' (descomente la línea con 'chunksize' y comente la␣
↪línea
# siguiente, la cual carga el archivo entero en la variable peliculas_tv. Todo el␣
↪análisis puede ser
# realizado de la misma manera usando solo el subconjunto de las reseñas pero algunos␣
↪resultados podrían ser diferentes a los de los ejemplos.
# ----------
# peliculas_tv = next(pd.read_json("reviews_Movies_and_TV_5.json.gz", lines=True,␣
↪compression='gzip', chunksize=100000))
peliculas_tv = pd.read_json("reviews_Movies_and_TV_5.json.gz", lines=True,␣
↪compression='gzip')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data…
[nltk_data]   Unzipping tokenizers/punkt.zip.
CPU times: user 43.9 s, sys: 7.84 s, total: 51.7 s
Wall time: 53.2 s
```

Examinando los datos

Echamos un vistazo a las primeras 5 filas de cada conjunto de datos para ver qué atributos están disponibles. Estos son

- **reviewerID:** Una identificación única para identificar al autor de la reseña.
- **asin:** El "Número de Identificación Estándar de Amazon" que proporciona más información sobre el producto y la versión exacta.
- **reviewerName:** El nombre de usuario elegido por el revisor.
- **helpful:** Un registro de cuantos usuarios indicaron que la revisión fue útil/no útil.
- **reviewText:** El texto completo de la revisión.
- **overall:** La calificación general (1-5) dejada por el revisor.
- **summary:** Una versión corta de la revisión, usada como título.
- **unixReviewTime:** La fecha de creación de la revisión, en formato Unix Epoch.
- **reviewTime:** Una fecha legible por el ser humano que contiene el día, mes y año.

[ ]:
```
print(len(instant_video))
print(instant_video.head(5))
```

```
37126
       reviewerID        asin  … unixReviewTime   reviewTime
0  A11N155CW1UV02  B000H00VBQ  …     1399075200   05 3, 2014
1  A3BC8O2KCL29V2  B000H00VBQ  …     1346630400   09 3, 2012
2   A60D5HQFOTSOM  B000H00VBQ  …     1381881600  10 16, 2013
3  A1RJPIGRSNX4PW  B000H00VBQ  …     1383091200  10 30, 2013
4  A16XRPF40679KG  B000H00VBQ  …     1234310400  02 11, 2009
```

```
[5 rows x 9 columns]
```

```
[ ]: print(len(peliculas_tv))
     print(peliculas_tv.head(5))
```

```
1697533
       reviewerID        asin  … unixReviewTime   reviewTime
0   ADZPIG9QOCDG5  0005019281  …     1203984000  02 26, 2008
1  A35947ZP82G7JH  0005019281  …     1388361600  12 30, 2013
2  A3UORV8A9D5L2E  0005019281  …     1388361600  12 30, 2013
3  A1VKW06X1O2X7V  0005019281  …     1202860800  02 13, 2008
4  A3R27T4HADWFFJ  0005019281  …     1387670400  12 22, 2013

[5 rows x 9 columns]
```

Notamos que `peliculas_tv` es extremadamente larga con casi 2 millones de reseñas, y varias columnas parecen poco interesantes o difíciles de trabajar (por ejemplo, `reviewerID`, `asin`, `reviewername`, `reviewtime`). Eliminemos un poco de información para hacer más eficientes algunos de nuestros análisis posteriores. También añadamos una columna de fecha y hora con objetos de fecha y hora en Python para resumir más fácilmente los datos:

```
[ ]: %%time
     peliculas_tv['datetime'] = pd.to_datetime(peliculas_tv['reviewTime'], format="%m %d,␣
      ↪%Y")
     instant_video['datetime'] = pd.to_datetime(instant_video['reviewTime'], format="%m %d,␣
      ↪%Y")
```

```
CPU times: user 5.06 s, sys: 7.13 ms, total: 5.06 s
Wall time: 5.07 s
```

```
[ ]: peliculas_tv = peliculas_tv.drop(columns = ['reviewerID', 'asin', 'reviewerName',␣
      ↪'reviewTime'])
     instant_video = instant_video.drop(columns = ['reviewerID', 'asin', 'reviewerName',␣
      ↪'reviewTime'])

     peliculas_tv.head(5)
```

```
[ ]:   helpful  …   datetime
     0  [0, 0]  … 2008-02-26
     1  [0, 0]  … 2013-12-30
     2  [0, 0]  … 2013-12-30
     3  [0, 0]  … 2008-02-13
     4  [0, 0]  … 2013-12-22

     [5 rows x 6 columns]
```

Ejercicio 1: (45 min)

1.1

Dibuje los histogramas de todas las cantidades numéricas. ¿Nota algo interesante en ellos?

Respuesta.

1.2

¿Cómo cambian las puntuaciones medias a lo largo del tiempo? Dibuje la calificación promedio de cada año y anote las tendencias.

Respuesta.

1.3

Examine la longitud media de las reseñas por año. ¿Nota alguna tendencia?

Respuesta.

Ejercicio 2: (60 min)

2.1

Encuentre las diez palabras más frecuentes que ocurren en: i) todas las críticas, ii) las críticas positivas, iii) las críticas negativas. ¿Le sorprenden los resultados? ¿Por qué sí o por qué no? (solo incluya palabras que no sean *stopwords*).

Respuesta.

2.2

Encuentre palabras que sean indicativas de malas críticas. Es decir, palabras que aparecen a menudo en malas críticas y *no* en buenas críticas. ¿Cuáles son estas palabras? ¿Le sorprenden?

Respuesta.

Ejercicio 3: (25 min)

Inspeccione manualmente las primeras 10 críticas negativas que contengan la palabra "bueno". ¿Qué nota? ¿Cómo sugiere esto que debemos proceder a continuación?

Respuesta.

Ejercicio 4: (45 min)

Revise la lista de críticas malas que contienen la palabra "bueno" que encontramos en la última pregunta. Para cada crítica, extraiga lo siguiente:

1. La primera palabra después de "bueno"
2. La primera palabra después de "bueno" que es un sustantivo o un cardinal
3. La última palabra antes de "bueno" que es un sustantivo o un cardinal

Respuesta.

Ejercicio 5: (30 min)

Hemos visto que las palabras individuales no siempre son muy informativas. Busque los bigramas y trigramas más informativos, tanto en las reseñas positivas como en las negativas. Muestre los bigramas y trigramas más informativos y haga un breve análisis de los n-gramas que ha identificado.

Respuesta.

Ejercicio 6: (15 min)

A lo largo de nuestra búsqueda de palabras informativas, hemos visto que los unigramas no son suficientes, y que las palabras importantes (como "bueno") no siempre están junto a las palabras informativas que describen. Diseñe un método para extraer estas palabras informativas. Proporcione una breve descripción de cómo extraerá las palabras informativas.

Respuesta.

Ejercicio 7: (30 min)

Escriba una función o funciones que transformen una oración en una nueva lista de texto emparejando iterativamente cada adjetivo de la oración con el siguiente sustantivo que le sigue en la oración. Por ejemplo, el texto "That was a good, long movie" ("Esa fue una película buena y larga") debe devolver `["good movie", "long movie"]` ("película buena", "película larga").

Respuesta.
