Predecir próxima compra de un usuario

Autores: Gabriel Crisnejo, Gastón Valvassori
Mentor: Bruno Marengo
Diplomatura en Ciencia de Datos (FaMAF-UNC)

Data set

Dataset: historial de navegación


```
{"user_history":
[
{"event_info": "RADIOBOSS"   , "event_timestamp": "2019-10-08T16:23:51.630-0400", "event_type": "search"},
 {"event_info": 505541       , "event_timestamp": "2019-10-08T16:24:06.220-0400", "event_type": "view"},
{"event_info": 505541        , "event_timestamp": "2019-10-08T16:25:56.141-0400", "event_type": "view"},
{"event_info": 505541        , "event_timestamp": "2019-10-08T16:26:15.505-0400", "event_type": "view"},
{"event_info": 505541        , "event_timestamp": "2019-10-08T16:26:46.149-0400", "event_type": "view"},
{"event_info": 505541        , "event_timestamp": "2019-10-08T16:36:11.769-0400", "event_type": "view"},
{"event_info": "RADIOBOSS"   , "event_timestamp": "2019-10-08T22:32:32.256-0400", "event_type": "search"},
 {"event_info": "SOUND FORGE" , "event_timestamp": "2019-10-10T14:32:56.589-0400", "event_type": "search"},
 {"event_info": 1230082       , "event_timestamp": "2019-10-12T09:56:36.964-0400", "event_type": "view"},
{"event_info": 937557        , "event_timestamp": "2019-10-12T10:02:51.871-0400", "event_type": "view"}
],
"item_bought": 1197370}
```

Dataset: metadatos

	item_id	title	domain_id	product_id	price	category_id	condition
0	111260	Casa Sola En Venta Con Gran Patio Solo Pago De...	MLM-INDIVIDUAL_HOUSES_FOR_SALE	NaN	1150000.00	MLM170527	new
1	871377	Resident Evil Origins Collection Nintendo Swit...	MLM-VIDEO_GAMES	15270800.0	1392.83	MLM151595	new
2	490232	Falda De Imitación Piel Negra	MLM-SKIRTS	NaN	350.00	MLM7697	new
3	1150706	Powercolor Red Devil Radeon Rx 580 8gb Gddr5	MLM-GRAPHICS_CARDS	NaN	3200.00	MLM9761	used
4	934912	Laptop Hp Nx6320 Core Duo Con Puerto Db9 Windo...	MLM-NOTEBOOKS	NaN	1599.00	MLM1652	used
5	534737	Transmisor Fm Sin Hilos Bluetooth Reproductor ...	MLM-VEHICLE_ACCESSORIES	NaN	470.00	MLM92472	new
6	369182	Funda Cartera Caseme 007 2 En 1 Huawei P30 Pro	MLM-CELLPHONE_COVERS	NaN	589.00	MLM167442	new
7	690585	Lampara De Techo Tubo Slim Led Base G5 9w Blan...	MLM-WALL_AND_CEILING_LIGHTS	NaN	184.00	MLM1588	new
8	890593	Repisa Organizador Para Cocina, Especieros, T...	MLM-NAPKIN_HOLDERS	NaN	879.00	MLM167544	new
9	1786901	Trío, Balerina, Cosmo, Leopardo/negro/rojo, Ve...	MLM-FLATS	NaN	729.00	MLM193197	new

Sistema de recomendación

Capacity **1 Seater**

Highlights
- Material: Leatherette
- 3 Reclining Positions
- Knock Down
- Filling Material: Foam

Similar Products



Baseline

```python
def baseline_row(row,df,n_items=10):
    viewed = list(dict.fromkeys(df["user_view"][row]))

    if len(viewed) == n_items:
        return viewed
    elif len(viewed) > n_items:
        return viewed[:n_items]
    else:
        random.seed(123)
        return viewed + random.choices(list(train_data["item_bought"]),k=n_items-len(viewed))
```

Score

Discounted Cumulative Gain at k,

$$DCG@k = \sum_{i=1}^{k} G(rel(i)) \cdot D(i),$$

donde

$$rel(\hat{y}, y) = \begin{cases} 12 & \text{si} \quad \hat{y} = y \\ 1 & \text{si} \quad \text{dom}(\hat{y}) = \text{dom}(y) \\ 0 & \text{si} \quad \text{si no} \end{cases}$$

$$G(x) = x, \qquad D(i) = \frac{1}{\log(i+1)}$$

Normalized Discounted Cumulative Gain at k,

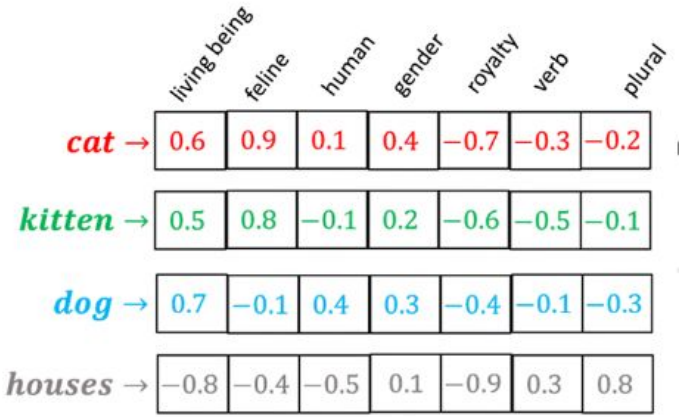
$$nDCG@k = \frac{DCG@k}{IDCG@k}$$

Word2Vec

Word2Vec - (vector representation of words)

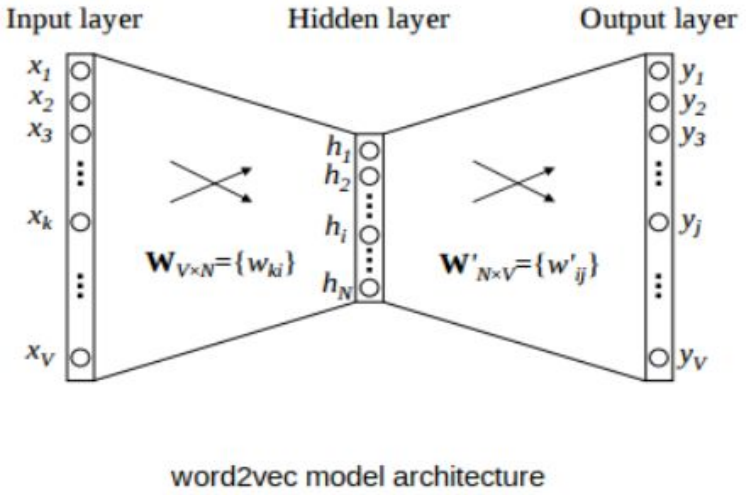
1. Word2Vec es una red neuronal con una sola capa oculta extensamente usado en NLP.
2. Objetivo: predecir las palabras más cercanas para cada palabra en una oración.

Quantum computing researchers teleport data inside a diamond

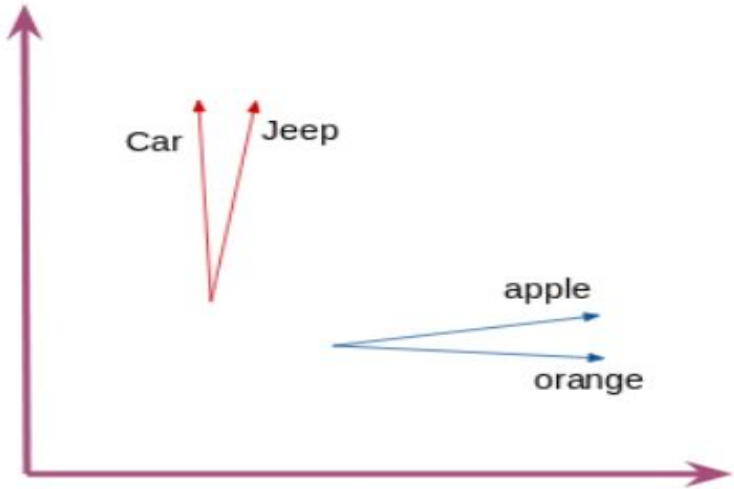


Preparando los datos para procesar

Quantum computing researchers teleport data inside a diamond



word2vec model architecture



¿Cómo utilizamos Word2Vec para armar un sistema de recomendación?

Sistema de recomendación

Entrenamiento del modelo de Gensim Word2Vec

```
nworkers = 2
#train word2vec model
model = Word2Vec(window = 5, sg = 0, hs = 0, min_count=1,
                 negative = 10, # for negative sampling
                 alpha=0.03, min_alpha=0.0007,
                 seed = 14, workers=nworkers,
                 cbow_mean = 1)


#model.build_vocab(purchases_train, progress_per=200)
model.build_vocab(train_words, progress_per=200)

model.train(train_words, total_examples = model.corpus_count, epochs=20, report_delay=1)
```

train_words cuenta con 2.5 millones de filas compuestas por los títulos de los items vistos y las búsquedas realizadas por el usuario

1. Cada fila del *train_data.jl* corresponde a un usuario, el mismo tiene asociado una lista de IDs de items vistos y una lista de búsquedas
2. Para cada ID de item visto buscamos el título del artículo y preprocesamos (tokenizar, stop-words, etc.), los concatenamos en una única lista que contiene todas las palabras de los artículos vistos, luego:
   2.1. Buscamos 10 palabras similares a la lista de strings con *Word2Vec.most_similar* y armamos una nueva lista de 10 strings
   2.2. Contabilizamos la cantidad de palabras similares de esta lista contra todos los items disponibles en el dataset de testeo y guardamos los 5 artículos con mayor cantidad de palabras similares
3. Repetimos el mismo mecanismo pero con la lista de búsquedas realizadas por el usuario, obteniendo otros 5 items que corresponden a los que más palabras similares tienen
4. Armamos una lista de 10 items para recomendarle al usuario

```
39      def _preprocess(self):
40          """
41          This method preprocess the user view item title and bought title
42          """
43          # Read user view item titles
44          user_view_title = [self.item_title[str(item_id)] for item_id in self.user_view]
45          user_view_title_nvoid = [user_view_title[i] for i in range(len(user_view_title)) if user_view_title[i] != []]
46          self.view = []
47          for title in user_view_title_nvoid:
48              titl = [word for word in title if word in self.model.wv]
49              self.view.append(titl)
50
51          self.view = [self.view[i] for i in range(len(self.view)) if self.view[i] != []]
52
53          # Preprocess user search
54          search = [simple_preprocess(self.user_search[i]) for i in range(len(self.user_search))]
55          self.search = [search[i] for i in range(len(search)) if search[i] != []]
```

1. Cada fila del *train_data.jl* corresponde a un usuario, el mismo tiene asociado una lista de IDs de items vistos y una lista de búsquedas
2. Para cada ID de item visto buscamos el título del artículo y preprocesamos (tokenizar, stop-words, etc.), los concatenamos en una única lista que contiene todas las palabras de los artículos vistos, luego:
   2.1. Buscamos 10 palabras similares a la lista de strings con *Word2Vec.most_similar* y armamos una nueva lista de 10 strings
   2.2. Contabilizamos la cantidad de palabras similares de esta lista contra todos los items disponibles en el dataset de testeo y guardamos los 5 artículos con mayor cantidad de palabras similares
3. Repetimos el mismo mecanismo pero con la lista de búsquedas realizadas por el usuario, obteniendo otros 5 items que corresponden a los que más palabras similares tienen
4. Armamos una lista de 10 items para recomendarle al usuario

```
57      def model_ms(self):
58          """
59          This method will use wv.most_similar to find the top 10 words by search and view titles
60          for the given user_id.
61          """
62          self._preprocess()
63
64          self.recom_by_views = [self.model.wv.most_similar(view, topn=10)[i][0] for i in range(10) for view in self.view]
65          self.recom_by_search = [self.model.wv.most_similar(search, topn=10)[i][0] for i in range(10) for search in self.search]
```

1. Cada fila del *train_data.jl* corresponde a un usuario, el mismo tiene asociado una lista de IDs de items vistos y una lista de búsquedas
2. Para cada ID de item visto buscamos el título del artículo y preprocesamos (tokenizar, stop-words, etc.), los concatenamos en una única lista que contiene todas las palabras de los artículos vistos, luego:
   2.1. Buscamos 10 palabras similares a la lista de strings con *Word2Vec.most_similar* y armamos una nueva lista de 10 strings
   2.2. Contabilizamos la cantidad de palabras similares de esta lista contra todos los items disponibles en el dataset de testeo y guardamos los 5 artículos con mayor cantidad de palabras similares
3. Repetimos el mismo mecanismo pero con la lista de búsquedas realizadas por el usuario, obteniendo otros 5 items que corresponden a los que más palabras similares tienen
4. Armamos una lista de 10 items para recomendarle al usuario

```
67      def find_matches(self, min_match):
68          """
69          This method will find string matches in every item title and bought title
70
71          min_match          Minimum string matches
72          """
73          def allopt(recom, itemID):
74              """
75              This function finds matches of strings between two lists.
76
77              recom       A 10 strings list
78              itemID      ID of an item
79              """
80              return [elem in self.item_title[itemID] for elem in recom]
81
82          self.model_ms()
83          max_len = max(map(len, self.item_title))
84          aux_view = {itemID : sum(allopt(self.recom_by_views, itemID))/max_len for itemID in self.item_title.keys() if sum(allopt(self.recom_by_views, itemID)) > min_match}
85          aux_search = {itemID : sum(allopt(self.recom_by_search, itemID))/max_len for itemID in self.item_title.keys() if sum(allopt(self.recom_by_views, itemID)) > min_match}
86
87          # Store top ten matches
88          self._view_match = sorted(aux_view, key=aux_view.get, reverse=True)[:5]
89          self._search_match = sorted(aux_search, key=aux_search.get, reverse=True)[:5]
90
91          # The final 10-items-recommendation will be a list with the top 5 items of each match
92          self.reco_list = self._view_match + self._search_match
```

Ejemplo cualitativo

Historial del usuario:

```
Usuario 5475 vio
-------------------------------------------------
Celular Xiaomi Redmi Note 8 64gb 4gb Branco Versão Global+nf
 Xiaomi Redmi Note 8 64gb Versão Global
Celular Xiaomi Redmi Note 8 64gb 4gb Branco Versão Global+nf
Celular Xiaomi Redmi Note 8 64gb 4gb Versão Global + Nf
-------------------------------------------------
Usuario 5475 buscó ['XIAOMU NOTE 7', 'XIAOMU NOTE 8']
```

Ejemplo cualitativo

Compra del usuario y recomendación del sistema:

```
El usuario compró el item 396062, Celular Xiaomi Redmi Note 8 64gb 4g- Branc +capa+nota Fiscal
-----------------------------------------------------------------------------------------------
Recomendamos
-----------------------------------------------------------------------------------------------
<Xiaomu Redimir 7a 32g Xiomim Xiomi Xiaiome Celular Vermelho>
-----------------------------------------------------------------------------------------------
<Xiaomu Redimir 7 64g Xiomim Xiomi Xiaiome Celular Top Novo>
-----------------------------------------------------------------------------------------------
<Xiaomu Redimir 7 32g Xiomim Xiomi Xiaiome Celular Top Novo>
-----------------------------------------------------------------------------------------------
<Xiaomu Redimir A2 Lite 64g Xiomim Xiomi Xiaiome Celular Top Novo Envio De Imediato>
-----------------------------------------------------------------------------------------------
<Xiaomu Redimir Note 7 64g Xiomim Xiomi Xiaiome Celular Top Novo Original >
-----------------------------------------------------------------------------------------------
<Xiaomi Redimir 8 64g Xaomi Xiomim Xiomi Xiaiome Xiaomu Black>
-----------------------------------------------------------------------------------------------
<Audifonos Redmi Airdots Xiaomi Inalambricos Bluetooth>
-----------------------------------------------------------------------------------------------
<Auricular Bluetooth Xiaomi Redmi Airdots Wireless>
-----------------------------------------------------------------------------------------------
<Xiaomi Redmi Airdots Auriculares Bluetooth Envío En 1 Día>
-----------------------------------------------------------------------------------------------
<Audifonos Xiaomi Redmi Airdots Bluetooth V5.0 Inalambricos>
-----------------------------------------------------------------------------------------------
```

Ejemplo cualitativo

Historial del usuario:

```
Usuario 27770 vio
--------------------------------------------------
Tarjetero Metalica Acero Inoxidable Credito Anti Rfid
Redlemon Smartwatch Reloj Inteligente Bluetooth Chip Sim Z60
Tapete Pokemon Tcg 2010 Hawai
Cartera Hombre Caballero  Swdvogan Original Con Envio
Pokemon Pikachu Tapete De Juego
125 Top Loaders Ultra Pro 3x4 Regular Para Tarjetas. Nuevos
Carteras Caballero Piel / Billetera De Piel Para Hombre
Frokie Shiny Pokemon Tcg Hidden Fates
Colgate Natural Extracts Carbón Charcoal Menta Pasta Dental
Pokemon Tcg Elite Trainer Box Hidden Fates Español
Tapete Pokemon Tcg Tapu Guardians
Sexy Lencería Bralett Panty Encaje Coordinado Body Conjunto
Pokémon Tcg: Dragon Majesty Elite Trainer Box
Cartera Victorinox Tres Dobleces Tri-fold Wallet (31172401)
Mini Tanga Corte V Resalta La Pompi Sexy Disponible S Y M
Tapetes Varios De Pokemon
Tarjetero Metalica Acero Inoxidable Credito Anti Rfid
Dragon Majesty Pin Collectionslatias And Latios
--------------------------------------------------
Usuario 27770 buscó ['TAPETE POKEMON']
```

Ejemplo cualitativo

Compra del usuario y recomendación del sistema:

```
El usuario compró el item 1674073, Tarjetero Con Bloqueo Rfid Slim Práctico Y Elegante
--------------------------------------------------------------------------------------------------
Recomendamos
--------------------------------------------------------------------------------------------------
<Peluche Pokemon Pikachu Raichu Cyndaquil Mewtwo Chikorita>
--------------------------------------------------------------------------------------------------
<Pins Pichu Pikachu Raichu>
--------------------------------------------------------------------------------------------------
<Camiseta Berserk Guts Anime Nerd Geek Akira Dbz 4190>
--------------------------------------------------------------------------------------------------
<Pokemon Mew Para Lets Go Pikachu Eevee>
--------------------------------------------------------------------------------------------------
<Mfc Pokemon Gba Memories Pikachu Charizard Mew Frete Grátis>
--------------------------------------------------------------------------------------------------
<Set 2 Pulseras Cierra Tus Ojos Y Pide Un Deseo Amor, Amistad Mpb-094>
--------------------------------------------------------------------------------------------------
<Fiesta Unicornio Centro De Mesa Foamy Fomi Fomy Unicornios>
--------------------------------------------------------------------------------------------------
<Thermic Plasma Conduits Warhammer 40k Sector Mechanicus Rpg>
--------------------------------------------------------------------------------------------------
<Peluche Pokemon Pikachu Raichu Cyndaquil Mewtwo Chikorita>
--------------------------------------------------------------------------------------------------
<Cartera Unicornios Moda Rainbow Colores Porta Dinero Tarjetas Unicorns Moda Oferta Remate Liquidación Mayoreo Billetera>
--------------------------------------------------------------------------------------------------
```

Conclusiones

Conclusiones

1. El baseline da un score nDCG de 0.16
2. Nuestro sistema de recomendación dio un score nDCG de 0.058 para un testeo con 5 usuarios, es muy chico pero el análisis cualitativo nos da la pauta que con algunas mejoras y con un set de testeo más grande, se puede mejorar. Cabe aclarar que es poco eficiente, demandando mucho tiempo de cómputo para muy pocos usuarios, por ejemplo para 50 usuarios tardó aproximadamente 2 horas y el dataset de testeo tiene 32 mil usuarios
3. El dataset original tiene alrededor de 600 mil filas pero por cuestiones de infraestructura solo utilizamos 200 mil, tal vez el sistema mejore si se entrenara con todo el dataset
4. Word2Vec tiene una falencia que recae en la imposibilidad de relacionar palabras que no estén en el vocabulario entrenado, esto se puede mejorar utilizando FastText que posee un sistema de relacionar palabras desconocidas con aquellas disponibles en el vocabulario
5. Se podría replantear el mecanismo de recomendación, en lugar de buscar similitudes entre palabras, Word2Vec permite hacer un entrenamiento por sentencias (oraciones) e idear un sistema de similitud entre conjuntos de oraciones

Fin