CS 512 Final Project Stage 4

Zixuan Zheng, Ningyuan Zhang, Suyu Huang, Hang Miao

December 16, 2018

Specify the language and programming environemnt you used for your implementation. We use python3.7 as the primary program language. The script is written on Jupyter Notebook and runs on Anaconda environment.
The deliverables for this stage include the following items:

Sample Input Data.

	train_id	name	item_condition_id	category_name	brand_name	price	shipping	item_description
0	0	MLB Cincinnati Reds T Shirt Size XL	3	Men/Tops/T-shirts	NaN	10.0	1	No description yet
1	1	Razer BlackWidow Chroma Keyboard	3	Electronics/Computers & Tablets/Components & P...	Razer	52.0	0	This keyboard is in great condition and works ...
2	2	AVA-VIV Blouse	1	Women/Tops & Blouses/Blouse	Target	10.0	1	Adorable top with a hint of lace and a key hol...
3	3	Leather Horse Statues	1	Home/Home Décor/Home Décor Accents	NaN	35.0	1	New with tags. Leather horses. Retail for [rm]...
4	4	24K GOLD plated rose	1	Women/Jewelry/Necklaces	NaN	44.0	0	Complete with certificate of authenticity

UI Input Error Message:

The User's input data should have the same tabular form as described above and placed in the same directory as the app file. Otherwise the error message down below will pop up

```
pandas\_libs\parsers.pyx in pandas._libs.parsers.TextReader.__ci

pandas\_libs\parsers.pyx in pandas._libs.parsers.TextReader._set

FileNotFoundError: File b'../input/train.tsv' does not exist
```

Application Outline

- Divide the raw data set into training, validition and testing set
- Exploratory analyze the data features such as shipping cost, category, description etc

- Tokenize the most important feature-description and apply machine learning techniques to make dimension reduction
- Use validation set, and cross validation techniques to tune the penalized parameters such as the number of cluster of K-means and LDA
- Divide the price into certain interval, using the reduced feature to train the Logistic LASSO model to fit the price interval
- Use test data calculate the MSE avoid overfitting

Tokenization and Word Frequency(tf-idf algorithm)

- tf-idf stands for term frequency-inverse document frequency. It's a numerical statistic intended to reflect how important a word is to a document or a corpus (i.e a collection of documents).
- To relate to this post, words correspond to tokens and documents correspond to descriptions. A corpus is therefore a collection of descriptions.
- The tf-idf a of a term t in a document d is proportional to the number of times the word t appears in the document d but is also offset by the frequency of the term t in the collection of the documents of the corpus. This helps adjusting the fact that some words appear more frequently in general and don't especially carry a meaning.

```
description: No description yet
tokens: ['description', 'yet']

description: This keyboard is in great condition and works like it came out of the box. All of
the ports are tested and work perfectly. The lights are customizable via the Razer Synapse app
on your PC.
tokens: ['keyboard', 'great', 'condition', 'works', 'like', 'came', 'box', 'ports', 'tested',
'work', 'perfectly', 'lights', 'customizable', 'via', 'razer', 'synapse', 'app']

description: Adorable top with a hint of lace and a key hole in the back! The pale pink is a 1
X, and I also have a 3X available in white!
tokens: ['adorable', 'top', 'hint', 'lace', 'key', 'hole', 'back', 'pale', 'pink', 'also', 'av
ailable', 'white']

description: New with tags. Leather horses. Retail for [rm] each. Stand about a foot high. The
y are being sold as a pair. Any questions please ask. Free shipping. Just got out of storage
tokens: ['new', 'tags', 'leather', 'horses', 'retail', 'stand', 'foot', 'high', 'sold', 'pai
r', 'questions', 'please', 'ask', 'free', 'shipping', 'got', 'storage']

description: Complete with certificate of authenticity
tokens: ['complete', 'certificate', 'authenticity']
```

tf-idf acts as a weighting scheme to extract relevant words in a document

$$tfidf(t,d) = tf(t,d).idf(t) \tag{1}$$

tf(t,d) is the term frequency of t in the document d

idf(t) is the inverse document frequency of the term t

$$idf(t) = log(1 + \frac{1 + n_d}{1 + df(d,t)}) \tag{2}$$

Dimensional Reduction and t-SNE

SNE converts the high-dimensional Euclidean distances between data_points into conditional probabilities that represent similarities.The similarity of datapoint x_j to datapoint x_i is the conditional probability, $p_{i|i}$.We denote a similar conditional probability by $q_{j|i}$.In order to find a low-dimensional data representation that minimizes the mismatch between $p_{i|i}$ and $q_{j|i}$.SNE minimizes the sum of Kullback-Leibler divergences over all datapoints using a gradient descent method.

$$p_{i|j} = \frac{exp(-\|x_i - x_j\|^2/(2\sigma_i^2))}{\sum_{k \neq i} exp(-\|x_i - x_k\|^2/(2\sigma_i^2))}$$

$$q_{j|i} = \frac{exp(-\|y_i - y_j\|^2)}{\sum_{k \neq i} exp(-\|y_i - y_k\|^2)}$$

$$C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{i|j} log(\frac{p_{j|i}}{q_{j|i}})$$

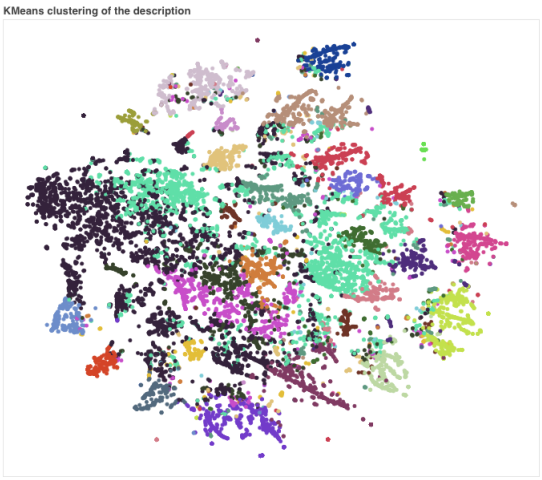
Visualization Feature Clustering

K-means Clustering of the description key words

0. Start with initial guesses for cluster centers (centroids)
1. For each data point, find closest cluster center (partitioning step)
2. Replace each centroid by average of data points in its partition
3. Iterate 1+2 until convergence

Write $x_i = (x_{i1}, ...x_{ip})$:
If centroids are $m_1, m_2, ...m_k$, and partitions are
$c_1, c_2, ...c_k$, then one can show that K-means converges to a *local* minimum of

$$\sum_{k=1}^{K} \sum_{i \in c_k} ||x_i - m_k||^2 \qquad \text{Euclidean distance}$$



LDA Clustering of the description key words

In natural language processing, Latent Dirichlet Allocation (LDA) is a generative model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's creation is attributable to one of the document's topics. LDA assumes the following generative process for each document w in a corpus D:

1. Choose N $\sim Poisson(\zeta)$.

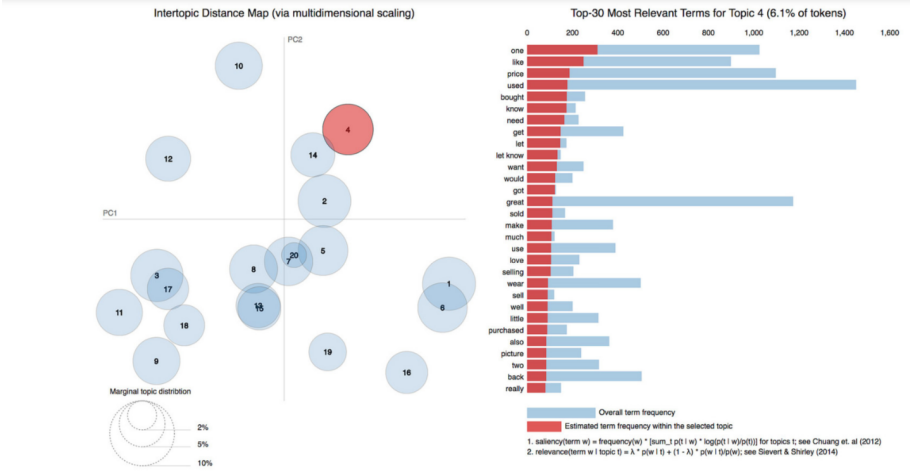
2.$Choose\theta \sim Dir(\alpha)$.

3.$For each of the N words \mathrm{w}_n$:

   (a) Choose a topic $z_n \sim Multinomial(\theta)$.

   (b) Choose a word w_n from $P(w_n|z_n, \beta)$, a multinomial probability conditioned on the topic z_n.

LDA distance map and 30 most relevent Terms



Working code

Attached in the same uploaded folder

Demo and sample findings

The For detailed information please check the demostration slide.

- Data size: 155mb in terms of RAM size; no Disk Resident; no Streaming;
- List the most interestng findings in the data if it is a Data Exploration Project. For other project types consult with your project supervisor what the corresponding outcomes shall be. Concentrate on demonstrating the Usefuness and Novelty of your application.

From our analysis, we find the description catagory is the most important feature for price prediciton. After compute the df-idf matrix, we could find the relevent distance between each key word and using unsupervised machine learning Cluster techneques we find product with those high frequency key world tend to clustered together. As a result, we could use the frequency of the word to forecast the price.

Pirce Prediction Output

1	price	test_id
2	8.473664297539269	0
3	8.91134521733877	1
4	61.47520137654318	2
5	12.308189910660753	3
6	7.827985542190516	4
7	8.965452303482326	5
8	8.585368807219762	6
9	25.574046424654924	7
10	53.717512521159684	8
11	11.378033758412322	9
12	48.54737380725747	10
13	12.294637025138622	11
14	34.527090730080318	12
15	38.816565789037135	13
16	33.023214101821524	14
17	12.387078604347894	15
18	16.3777258	16