Yelp Recommendation System

Abstract:
The model built in this project will recommend restaurants to the users based upon the current ratings of the user. Unlike the similarity based recommendation, this model in this project recommends unique restaurants to the users which are worth exploring. The data for this model is scraped from the yelp website using python scripting. The data collected is cleaned and preprocessed and then the model is applied to the preprocessed data for recommending surprisingly new restaurants which user may like

Motivation:
Natural Language Processing & Sentiment Analysis
What's in a review? Is it positive or negative? Yelp's reviews contain a lot of metadata that can be mined and used to infer meaning, business attributes, and sentiment.

Dataset Description:
In total, there are -
520,000 user reviews
Information on 160,000 businesses
The data spans 11 metropolitan areas.

Workflow:
- First, we want to find a way to represent reviews using a bag-of-words representation. After doing so, we will also represent categories using a one-hot encoding representation.
- Then, we can manipulate those representations to find similarities and differences while balancing the weights of the two.
- The core idea assumes that you are more likely to love a restaurant if its reviews are similar to the reviews of the restaurants you already love.

Data Acquisition:

This is an important step in every Machine learning project. The output and accuracy for the project depends on volume, Variety and Velocity of Data Collection .The 3 V's of the data is important to build an absolute model with less errors and better accuracy. Business data which is the data of the users for each restaurants and restaurant data is collected individually in common interest.I have built a scraper using python and HTML requests ,that extracts all the html contents of Yelppage. We used XPATHS to extract the data of particular contents in a page.

Data Cleaning:

The scraped data is cleaned removing the unwanted data and pulling out the missing values.As we collected raw data from th website,there are many noisy records in the data.We followed the following techniques to clean the data.

Removing Null Values: Null value indicates the data is missing so all the missing data in the reviews and cuisine are removed completely as the mean and median imputation will not give correct values to the categorical data.

Removing titles with short description: In text based recommendation this is an important step the titles which have short description will form very sparse vectors since,to avoid that the data with short description wich is less than 3 words are removed.

Data Preprocessing:

- Because reviews are too big, we will read them in chunks
- We only get businesses that already show up in our review list and delete the rest.

Text preprocessing:

- Punctuations in the review text are removed
- The word of the text are converted into lowercase and are splitted
- Stop words are identified and removed using python library, nltk(natural language processing toolkit.
    - Tokenization: The word of the text are converted into lowercase and are splitted to identify each unique word.

    - Removal of Stopwords: Stopwords are words with no specific semantics/meaning Examples of stopwords are is, am, are, a, as etc.

Building model:

vectorization:

Now the review and categories are vectorized using count vectorizer method in python which is simple Bag Of Words.

```
vectorizer_reviews = CountVectorizer(min_df = .01,max_df = .99, tokenizer = WordPunctTokenizer
().tokenize)
vectorized_reviews = vectorizer_reviews.fit_transform(df_yelp_review['text'])
```

TF-IDF :

Term frequency is calculated by counting the number of times each word is repeated in the given document divided to the number of words in the given document. Inverse document Frequency is called as log of the number of documents present divided to the documents containing the word

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

- Sparse representations are used to process the dot products easier that also saves memory.

```
%%time
from scipy import sparse
businessxreview = sparse.csr_matrix(pd.get_dummies(df_yelp_review['business_id']).values)
```

Now, we will select a seed restaurant and find other restaurants that are similar and good as the seed restaurant . Best results could be obtained by choosing a restaurant with good number of reviews and ratings.

	business_id	categories	name	stars
89849	aUrOyWFKxKeVXiFzwbTXSA	Vegetarian, Cafes, Vegan, Restaurants	The Green Owl Cafe	4.0

Recommendations are made between reviews and categories

Distance computation:
Two sets of distances are computed ,The correlation distance of the average vectorized reviews to all the reviews, and compute the correlation distance between this category and all other categories .

```
from scipy.spatial.distance import cdist
# find most similar reviews
dists1 = cdist(vectorizer_reviews.transform(new_reviews).todense().mean(axis=0),
              vectorized_reviews.T.dot(businessxreview).T.todense(),
               metric='correlation')
# find most similar categories
dists2 = cdist(vectorizer_categories.transform(new_categories).todense().mean(axis=0),
              vectorized_categories.todense(),
               metric='correlation')
```

The average of the two sets of the distance is considered through this distinct recommendations are obtained unlike the regular recommendations.

Results:

Top matches are recommended by selecting the closest restaurants to our seed restaurant.
and result obtained is shown as:

```
df_yelp_business.loc[df_yelp_business['business_id'].isin(df_yelp_business['business_id'].iloc
[closest]), ['business_id', 'categories', 'name', 'stars']]
```

	business_id	categories	name	stars
36563	yGCrsq0AYI8WN7goMVLHJA	Mexican, Vegan, Vegetarian, Restaurants	Yayo Taco	3.5
47957	AaWIhVc96VvdwNGMiW7HeA	Vegan, Vegetarian, Restaurants	Ashley's	4.0
58566	hM9g6_VR8sdzfvcWNsnUaQ	Vegan, Coffee & Tea, Cafes, Food, Restaurants,...	Greenbridge Teahouse and Cafe	4.5

Conclusion:

- Although many of those seem to come from the same category (Vegetarian and Vegan), there is a good variation in these categories (Mexican, coffee & tea, Asian Fusion, .. etc). Therefore we can obtain a different cuisine recommendation based on our previous interests and reviews .
- Also, the recommended product has high ratings and could be preferred to give a try.