Recommender Systems.

Technical Specifications Document

Prepared by Ajinth Christudas.

Version 1.0

December 2017

Table of Contents

1. Introduction

Recommender systems are applications or a subclass of information filtering system that help predict user ratings or preferences that the user would give to an item. Recommender systems have become increasingly popular in the recent years and are utilized in a variety of areas including movies, music, news, books, research articles, social queriesm and search tags in general

This project focusses on a dataset of Yelp reviews to recommend potential restaurants to users based on users prior ratings and restaurant attributes. Two different approaches are used to make the restaurant recommendations to users.

Purpose of the Technical Report Document

The purpose of the Technical Report document is to highlight the different phases of the Data Science lifecycle

- Problem Formulation and Overview
- Data Acquisition
- Data Preprocessing
- Model Building
- Model Evaluation

The document also highlights on the potential next steps to be taken to further fine tune this project (E.g. Setting up a web app to host the model as a service)

2. Problem Formulation and Overview

The primary goal of this capstone is to use the Yelp review data to make restaurant recommendations. The two models used for this approach are the collaborative filtering model and the content based model.

Collaboarative Filtering Model: Collaborative filtering approaches, build a model from a user's past behaviour (items previously purchased or selected and/or numerical ratings given to those items) as well as similar decisions made by other users. This model is then used to predict items (or ratings for items) that the user may have an interest in.

In this project, the data points are going to be the ratings that the user has provided to the restaurants, and using that information along with the ratings that the other users have given to make an estimation on the predicted rating, to make the top recommendations. The two approaches to collaborative filtering model include

- Item – Item Filtering Model: This approach involves making predictions based on similar items.
- User – User filteting Model: This approach involves making predictions based on similar users.

Content Based Filteting Model: Content-based filtering approaches utilize a series of discrete characteristics of an item in order to recommend additional items with similar properties.

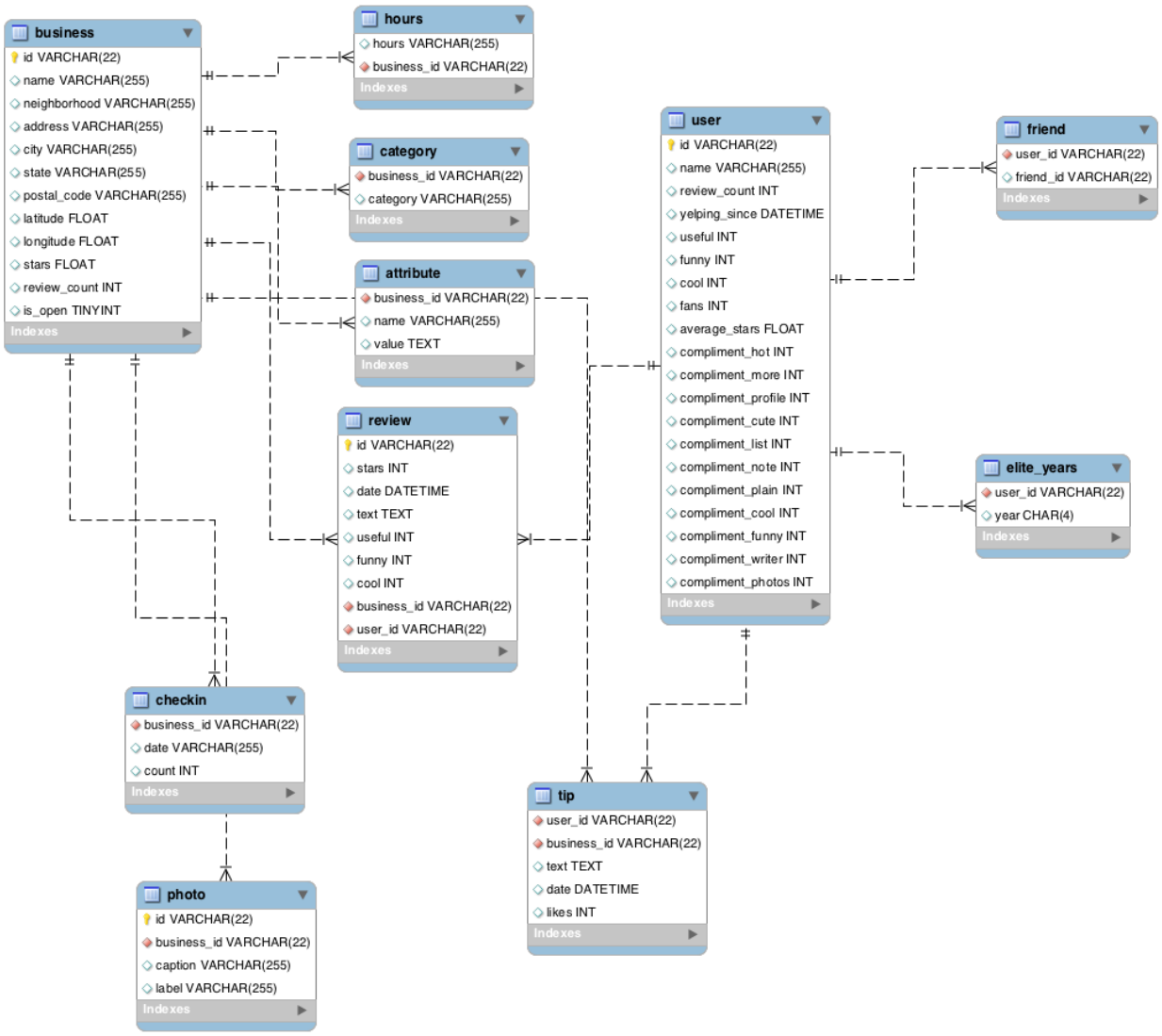
In this project, the data points are going to be the attributes of the restaurants like parking, lighting, music, food, service to name a few. A user profile is then created based on the attributes, and the recommendation engine recommends restaurants based on the similarity to the user profile

Other approaches also include a Hybrid approach of combining the Collaborative and the Content based model to make

3. Data Acquistion

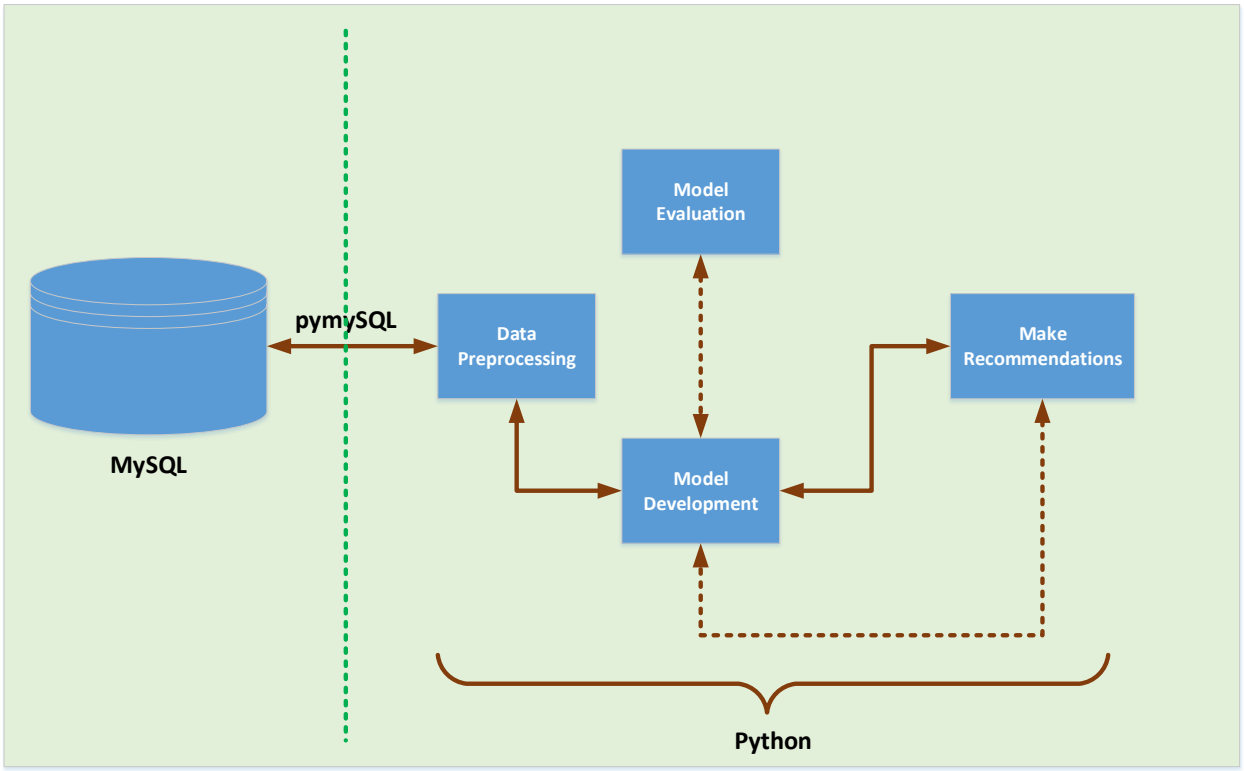
Raw Data Structure

The raw data on the different reviews, business, users, categories is all contained in a SQL file. The SQL file is loaded into a locally hosted MYSQL database. The data model for the raw data looks as in the diagram below.



Data Loading

The following graphic provides an illustration of the Logical Architecture implemented in the capstone



At a high level, the following steps were involved in the build

1. A connection is made to the the local MYSQL database using pymysql library in python
2. Data from the locally hosted MYSQL database is loaded into Pandas using MYSQL. Each table is loaded into its own dataframe.
    a. Reviews – This table contains details of the reviews and ratings that the users have provided. There are also foreign keys (user_id, business_id) to join back to the users and the business tables
    b. Business – This table contains details of the Business. E.g. Business Name, Business City, Business State, Business Zipcode.
    c. Users – This table contains the user details. E.g User Name, Review Count, Average rating. There are also other columns in the table which don't have much significance for this project.
    d. Category – This table contains the business category. E.g. Salon, Bar, Coffee Shop, Restaurants. This table joins with the business table with the foreign key business_id. The category field helps us segment out a specific business for the purpose of our analysis.
    e. Attributes – This table contains the attributes of the business. The foreign key business_id helps join the attrbutes back to the specific business.

3. There other tables that were a part of the data model, but that did not get utilized in the model are listed below.
    a. Hours
    b. Checkin
    c. Photo
    d. Tip
    e. Friend
    f. Elite Years

4. The dataframes associated with Reviews, Business, Users and Categories is merged to get all the data into one common dataframe for easier data processing.
5. The collaborative and content based filtering is applied to this unified dataframe to make the desired predictions for users.
6. Root Mean Square Error & Root Absolute error is used to validate the model by comparing the actual ratings and the predicted ratings.

4. Data Preprocessing

The data preprocessing step involves merging the data from the following four tables into a common dataframe. The join process involves an 4 step iterative process to get the final dataframe

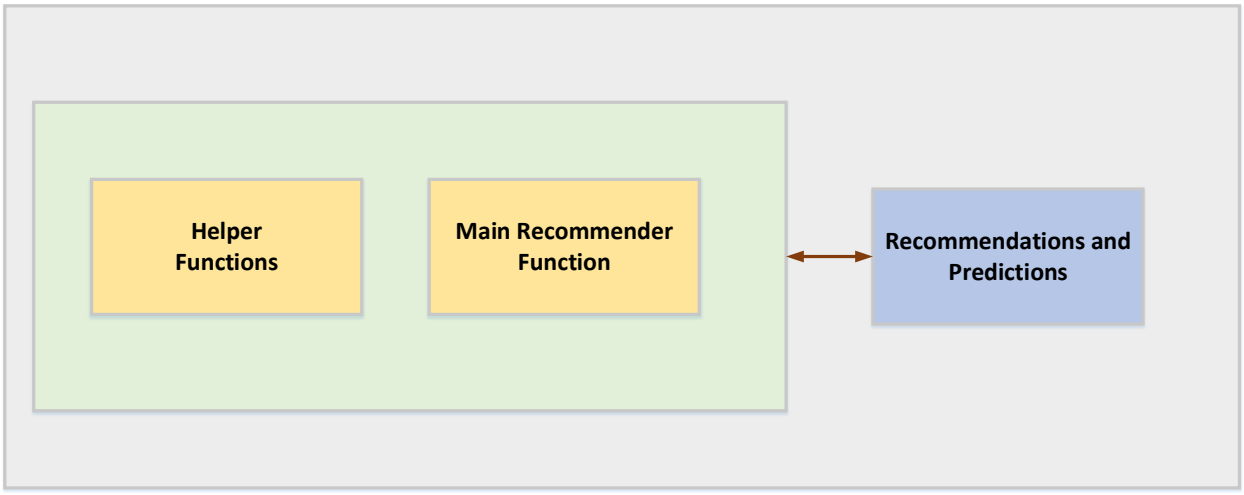
1. Merging the reviews and business dataframes on the common business_id. This step allows us to extract the business details
2. Merging the combined dataframe in step#1 with the users dataframe. This step allows us to extract the user details
3. Merging the combined dataframe in step#2 with the the categories dataframe. This step allows us to extract the category name and segment out the specific categories that are considered in the scope of this project.

The results of the three steps above is a combined dataframe with every row containing all details about the review, users, business and the categories. The goal of this capstone was to isolate the data to a specific state while building the model. This allows us to work and handle a sizeable amount of data while building out the model

5. Model Development

The model development uses the following design pattern

- Helper functions are defined to do repetitive tasks
- Main function is defined which uses the helper function along with some additional logic to predict ratings for the user



Collaborative (Item -Item Filtering Model):

Item based collaborative filtering is a model-based algorithm for recommender engines. In item based collaborative filtering similarities between items are calculated from rating-matrix (sparse matrix). And based upon these similarities, user's preference for an item not rated by the user is calculated.

E.g. A user dined at Restaurant A, and also given it a great rating. Restaurant B is very similar to Restaurant A based on rating given by other users, but the user has never been to Restaurant B. Based on the fact that the user has never been to restaurant B, but has visited a similar restaurant in Restaurant A, the recommender system would recommend Restaurant B to me.

Item-Item filtering Model:

The implementation of the Item-Item filtering model involves

- Creation of a sparse matrix with the users as rows, and columns as restaurants with the values of the matrix being the ratings as users
    - The sparse matrix is then normalized to handle hard raters and normal raters
- Definition of multiple Helper functions to
    - Return the list of restaurants rated by a specific user, by taking the user id as an argument
    - Return the list of restaurants yet to be rated by a specific user, by taking the user id as an argument
    - Return the cosine similarity of two vectors
    - Return the co-occurrence matrix (dataframe) between restaurants rated by the user (in rows) and all other restaurants (in columns)

- Definition of the main function that executes the helper function in addition to some more additional logic to provide the top 5 restaurant recommendations. The function not only provides the restaurant recommendations but also provides the ratings that the user would have given to the restaurants that the user is yet to visit.

Code Objects:

Helper Function Names	Main Function Names
• user_rated_list • user_not_rate_list • cosim • cooccurence_matrix_creator	• rating_predictor_colab_ii_all

Collaborative (User – User Filtering Model):

User based collaborative filtering is a model-based algorithm for recommender engines. In user based collaborative filtering similarities between users are calculated from rating-matrix (sparse matrix). And based upon these similarities, user's preference for an item not rated by the user is calculated.

E.g. A user A dined at Restaurant A, and also given it a great rating. Another user B dined at Restaurant B. Now based on the rating behavior user A is very similar to user B. Since user A has not been to Restaurant B, the user-user filteting model would recommend Restaurant B to user A.

User-User filtering Model Implementation:

The implementation of the User-User filtering model involves

- The sparse matrix that was utilized for the item-item filtering model can be utilized for this model. As before one the steps to standardize the hard raters from the soft raters we have to standardize the rating matrix.
- Definition of multiple Helper functions to
    - Return the co-occurrence matrix (dataframe) of user similarities
    - The helper functions defined as a part of the item-item filtering model also gets utilized in this model
- Definition of the main function that executes the helper function in addition to some more additional logic to provide the top 5 restaurant recommendations. The function not only provides the restaurant recommendations but also provides the ratings that the user would have given to the restaurants that the user is yet to visit.

Code Objects:

Helper Function Names	Main Function Names
• user_rated_list • user_not_rate_list • cosim	• rating_predictor_colab_uu_all

• user_cooccurence_matrix_creator • dataframe_tranposer • drop_self_user • joint_matrix_creator	

Content Based Filtering Model

Content based filtering model is a recommender model where recommendations are based on the attributes of the restaurant the user likes and the user profile. At a high level, key words or attributes are used to describe the restaurant and a user profile is created based on the attributes. Based on the users profile, restaurants that closely match the user profile are recommended to the user. To abstract the features of the items in the system, an item presentation algorithm is applied. A widely used algorithm is the **TF-IDF**. In this project we did not have to go ahead and abstract the features using TF-IDF methodology as the features were readily available to the user

Content Based Model Implementation:

The implementation of the Content based filtering model involves

- An intermediate dataframe to store a matrix of restaurant attributes and the user ratings. This matrix is used to create the user profile, which in turn is used to make recommendations.
- Definition of multiple Helper functions to
    - Return the co-occurrence matrix (dataframe) of user similarities
    - A separate helper function is used to create the co-occurrence matrix to calculate the restaurant similarity
- Definition of the main function that executes the helper function in addition to some more additional logic to provide the top 5 restaurant recommendations. The function not only provides the restaurant recommendations but also provides the ratings that the user would have given to the restaurants that the user is yet to visit.

Code Objects:

Helper Function Names	Main Function Names
• user_rated_list • user_not_rate_list • content_based_cooccurence_matrix_creator	• content_based_rating_predictor

6. Model Evaluation

The accuracy metric used for evaluating each of the recommender models was a combination of

- Root Mean Square Error (RMSE)
- Mean Absolute Error (MAE)

Each of the evaluation metric was carried out on the actual ratings and the predicted ratings. The average RMSE was in the range between 0.6 and 0.9. To interpret this it means that on an average our models was predicting within 0.7 points from the actual prediction. RMSE, is also very sensitive to outlier, hence a better metric to use was the Mean absolute error (MAE). The MAE absolute error is less sensitive to outliers and is a better metric in scenarios where the outliers is not a Huge concern

Model	Evaluator Functions
Collaborative filtering model	colab_ii_model_evaluator
Content based filtering model	content_based_evaluator

7. Next Steps

- Fine tune the content based model by parsing out the actual reviews from users (using TF-IDF) and re-evaluating the model
- Build out a web ui using flask to deploy the model for the users to be able to interact with using a UI

8. Github

https://github.com/Ajinth/Recommender-System

Appendix A: Record of Changes

Version Number	Date	Author/Owner	Description of Change
1.0	*12/17/2017*	*Ajinth Christudas*	*Intial Documentation*

References

Books:

- Recommender Systems by Charu C Agarwal
- Machine Learning in Action by Peter Harrington

Blogs and Websites:

- https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf
- https://ashokharnal.wordpress.com/2014/12/18/worked-out-example-item-based-collaborative-filtering-for-recommenmder-engine/
- https://github.com/kdais/RecommenderEngine/wiki/Content-based-filtering
- https://github.com/Ajinth/RecommenderSystems_PyData_2016
- https://cambridgespark.com/content/tutorials/implementing-your-own-recommender-systems-in-Python/index.html
- https://blog.dominodatalab.com/recommender-systems-collaborative-filtering/
- https://medium.com/@m_n_malaeb/the-easy-guide-for-building-python-collaborative-filtering-recommendation-system-in-2017-d2736d2e92a8
- https://beckernick.github.io/matrix-factorization-recommender/
- https://www.youtube.com/watch?v=-8BrRnFzq_Y
- https://www.youtube.com/watch?v=KeqVL-0vSQg&list=PLseNcwx1RJ4WdgtrMTXndw4B4nlf4-pgS
- http://aimotion.blogspot.com/2009/11/collaborative-filtering-implementation_13.html
- https://www.themarketingtechnologist.co/a-recommendation-system-for-blogs-content-based-similarity-part-2/