CRAYON DATA WHISKEY RECOMMENDATION ENGINE PROJECT
FINAL REPORT

May 6th, 2019

Group Members: Tian Xia, Di Zhu, Muskan Jain, Huy Nguyen

1. COMPANY OVERVIEW

Crayon Data Pte Ltd., located in Singapore and India, engages in the process of building a business and technology platform that democratizes the use of big data for the average business and consumer. Their website is https://www.crayondata.com/.

2. PROBLEM STATEMENT

Crayon Data engages our team to create a recommendation system for whiskey products using collaborative and content-based filtering and build a prototype user interface for the system.

3. APPROACH

Our project workflow is as follows: (3a) Data acquisition → (3b) Collecting consumer surveys → (3c) Preprocessing data → (3d) Modelling → (3e) Developing user Interface

3a. Data Acquisition

We used Python script to scrape the data from a website called Distiller (www.distiller.com), which features data on whiskeys such as its price, flavor profile and so on. It also had user data such user ratings and comments.

We eventually consolidated everything into two primary datasets:

1. **Whiskey attributes dataset**: contains whiskey attributes such as: whiskey names, type, origin, average rating from all users, whiskey age, alcohol by volume (abv) score, whiskey style, cask type, flavor headlines. This dataset also include a "flavor profile" for each whiskey such as: smoky, peaty, spicy, herbal, oily, full bodied, rich, sweet, briny, salty, vanilla, tart, fruity, floral (on a scale of 0 - 100). Finally, the dataset contains text attributes such as tastes note and description.
2. **Ratings dataset**: contains user comments and ratings.

A few key takeaways from our data acquisition process:

1. There are 61154 distinct users and 2330 distinct whiskeys in our dataset.
2. The ratings are heavily skewed. Most ratings fall in the 4-5 star ranges. This is intuitively true as most whiskey drinkers only rate when they are very satisfied with the products.

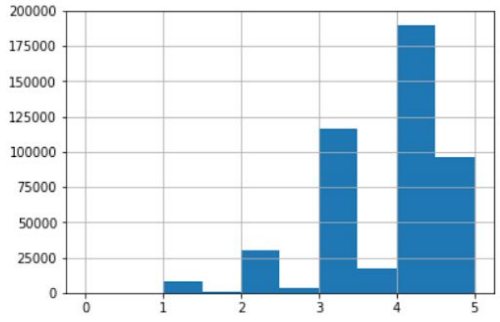


Figure 1. Distribution of whiskey ratings

3b. Collecting and Analyzing consumer surveys

We surveyed people in bars and whiskey shops to understand which factors are weighed more by drinkers and gauge information about consumer preferences for whiskey. We had also put a survey on social media sites targeting whiskey drinkers (such as Reddit) to get collect more data. We managed to collect 68 responses.
After analyzing survey data, we divided the drinkers into two segments: the people drinking whiskey for 0-10 years (the inexperienced) and the people drinking whiskey for more than 10+ years (the experienced). We believe the inexperienced segment has more data. We sought to get answers to some questions through the survey:

- Drinking pattern: **Inexperienced** - once a week (39%), a few times a week (30%) versus **Experienced** - few times a week (54%) and everyday (32%)
- Price when buying whiskey for themselves: **Inexperienced** drinkers said 80% of them will pay $0-100 for the whiskey whereas only 59% of the **experienced** users said that they will buy whiskey priced in the range of $0-100. We believe that the reason behind this is that experienced drinkers look for better quality while buying whiskey whereas inexperienced ones don't have much info about whiskeys and hence, are not willing to spend much.
- Price when buying whiskeys for others: 83% of **inexperienced** drinkers said they will purchase a whiskey product in the range of $0-100 for others. The pattern for buying whiskey for others is the same as when buying for themselves for experienced drinkers.
- Most important factors when buying whiskeys: For **inexperienced** drinkers, the most important factor is flavor, followed by whiskey style and average rating (this makes sense because the inexperienced drinkers depend more on recommendations). For **experienced** drinkers, flavor is the most important factor, style is the second most important factor and country of origin is the third most important factor affecting their choice of whiskey.
- Preferred type: **Inexperienced** drinkers mainly prefer what's popular in the market (Jack Daniels, Johnny Walker) whereas **experienced** drinkers have more specific, refined tastes.

The purpose of the consumer survey is for us to figure out important factors which can affect a user's preference for a whiskey. We then selected these factors for modelling.

3c. Data preprocessing

Our data processing steps are as follows:

1. Removed the rows which didn't have average user ratings.
2. Removed rows which didn't have flavor headlines.
3. For the age column, we changed all NA's to the mean ratings of all other users.
4. Dropped rows which had NaN for style column
5. Dropped rows which had NaN for origin column.

For the user dataset, we:

1. Dropped timestamp and comments.
2. Added whiskey ID and user ID for each row.

3d. Modelling

We used two classes of models to give recommendations to users: content-based filtering methods, which recommend based on the similarities among whisky products, and collaborative filtering methods, which look at the similarities of a user with other users in our dataset.

Content-based filtering: we decided to use a *combined approach* for our content-based recommendation system. First, we obtained an attribute vector as the weighted average of the attributes in our user input (weighted by ratings). Next, we used five different similarity metrics to find the whiskey products that are most similar to this attribute vector. We obtained 10 whiskeys for each approach. Finally, we put all of these whiskey recommendations into a list, and output 10 whiskeys that appear the most in this list. The five similarity metrics we used are as follows:

1. KNN with ball tree algorithm using Euclidean distance.
2. Jaccard similarity: measures the difference between two sets of attributes. The formula is given by $J(A, B) = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$ where A and B are 2 attribute sets.
3. Cosine similarity: measures the cosine of 2 n-dimensional attribute vectors. The formula is given by $Cosine\ Similarity\ (A, B) = \frac{A \bullet B}{|A| \times |B|}$ where again, A and B are 2 attribute sets.
4. Pearson similarity: is defined as the covariance of 2 n-dimensional vectors divided by the the product of their standard deviations. $Pearson\ similarity = \frac{Cov(A,B)}{\sigma A \times \sigma B}$.
5. TF-IDF (term frequency, inverse document frequency): measures the similarity among bags of words. The importance of a term is positively correlated with the frequency of its appearance in a document, but inversely correlated to how often it appears across documents in a corpus. $tf - idf(t) = tf(t, d) \times idf(d)$ where t is the term and d is the document.

For similar metrics 1 - 4, we used the numerical features (such as the age of the whiskeys and its flavor profile features). For similarity metrics 5, we used the taste notes, descriptions, origins and flavor headlines. By combining these similarity approaches, we hope to highlight the whiskeys that are closest to our users' preferences in terms of not only by quantifiable attributes but also styles, perceptions and subtleties in tastes that can only expressed through texts.

We did not perform train-validate-test on our content-based model as we lack a comprehensive user history to train our model on. We could have used other users for our training set, but we recognize the difference from person to person, which can greatly affect perception of tastes and as a result, rating outcomes.

Collaborative filtering models: using the library **surprise,** we initially chose six candidate models: KNN user-based, KNN item-based, SVD, SVD++, NMF and ALS. The overview for six candidate models is as follows:

1. **KNN (user-user similarity) with means**: collaborative filtering based on the similarities computed between users. We also take into account the mean rating of each user. This choice makes sense in our case because as we observed, the rating distribution is heavily skewed.
2. **KNN (user-item similarity) with means**: similar to the above, based on the similarities computed between items.
3. **SVD**: is a form of matrix factorization where we reduced the dimension of a matrix from N to K < N (in other word, SVD is a method to compute low rank approximation of a sparse matrix). User ratings can be decomposed as $\hat{r}_{ui} = q_i^T p_u$. In summary, we learned the whiskey characteristic vector q(i) and user characteristic vector p(u) by minimizing the regularized squared errors as follows:

$$\min_{q^*, p^*, b^*} \sum_{(u,i) \in s} (r_{ui} - \mu - b_u - b_i - q_i^T p_u)^2 + \lambda(||q_i||^2 + ||p_u||^2 + b_u^2 + b_i^2)$$

   (r(ui) is the rating of user u for whiskey i, b(u) as well as b(i) are the biases, and lambda is the regularization term.)
4. **SVD++:** SVD++ is a matrix factorization method which takes into account implicit interactions as opposed to only using p(u), which only accounts for the impact of the explicit interactions.
5. **NMF (non-negative matrix factorization)**: matrix factorization technique where matrix p and q are non-negative. Non-negative factorization automatically extracts information for non-negative set of vector.
6. **ALS:** is a collaborative filtering method in which users and products are described by a small set of latent factors that can be used to predict missing entries. This model uses Alternating Least squares algorithm to learn these latent factors.

Our procedure is as follows: we only used 5% of our dataset due to computational limitations. We then used this data as the training set with the intention of choosing an optimal set of hyperparameter and a best-performing model. Our performance metrics are RMSE and MAE. First, we ran a grid search with cross validation for the candidate models on the entire training set to choose an optimal set of hyperparameter for each model. Then, we ran 10-fold cross validations on the training set with the chosen set of hyperparameter and the result is presented in Section 4. We then

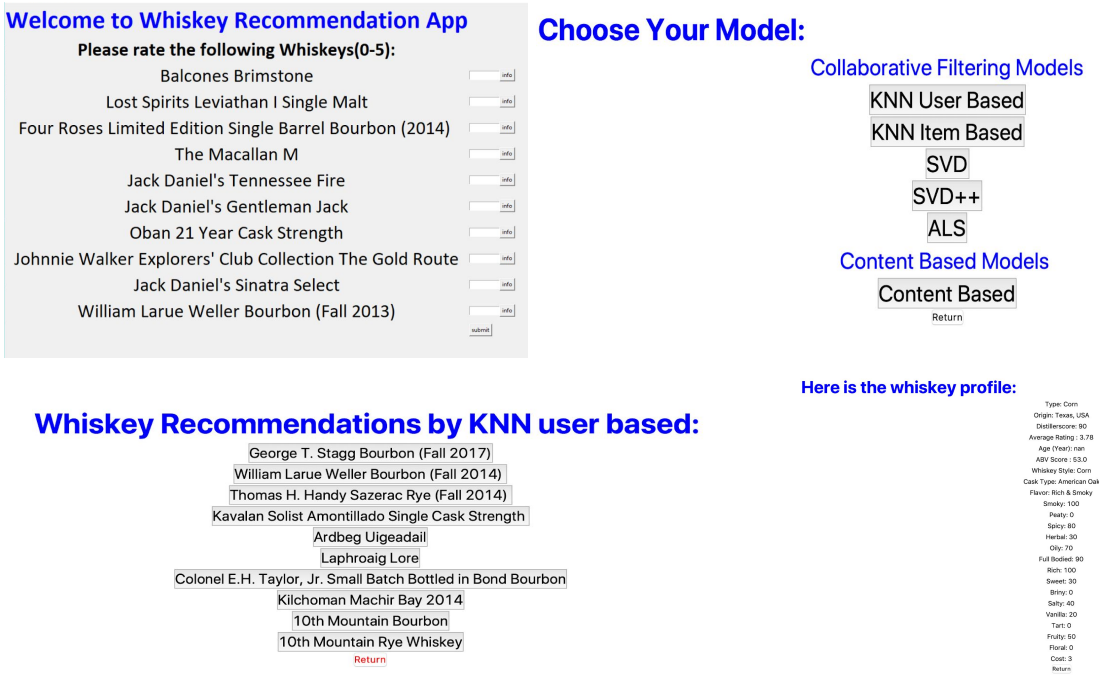
proceed to select the model and tested on a customized test set. Finally, we used the chosen model (SVD) for predictions using the **get_top_recommendations**() function.

3e. Developing user interface

For our prototype app, we used **tkinter** to build our user interface. On the interface, we listed the top ten whiskeys, three of which are the most popular ones and two of which was added in to represent a complete set of flavor profile. We, then, ask the new user to rate them. We concatenated these ratings to our existing training set to produce a new training set, based on which we built a test set and predicted results.

We have also addressed the issue where the user is completely new to the world of whiskey. In this case, no additional ratings are added to our training set, and the recommendations given are based entirely on historical data. Finally, we output the recommendation results from all the models, per the requests of our client, and let the user choose for himself as to which model he wants to get recommendations from.

Figure 2. Demonstration of user interface



4. RESULT

Content-based Filtering: below we present the recommendations using a combined approach.
Figure 3. Results for content-based filtering

```
['Glenfarclas 21 Year',
 'A Drop of the Irish Sherry Cask Finish (Blackadder)',
 'Fettercairn Fior',
 'Nikka Taketsuru Pure Malt 21 Year',
 'Nikka Yoichi 15 Year Single Malt',
 'Jura Superstition',
 'Kurayoshi 12 Year Pure Malt Whisky',
 'Barrell Bourbon Batch 004',
 'Jim Beam Single Barrel Bourbon',
 'Bunnahabhain 18 Year']
```

Collaborative Filtering: The optimal set of hyperparameter, RMSE and MAE for each model is presented as follows:

Model	Optimal Hyperparameter Set	RMSE	MAE
KNN User-based	k = 40, sim_options={'name': 'msd', 'min_support': 1, 'user_based': True})	0.9778	0.7547
KNN Item-based	k = 10, sim_options={'name': 'pearson_baseline', 'min_support': 4, 'user_based': False})	0.8564	0.6689
SVD	**n_factors = 110, n_epochs = 90, lr_all = 0.005, reg_all = 0.15**	**0.8053**	**0.6228**
SVD++	n_factors = 110, n_epochs = 90, lr_all = 0.005, reg_all = 0.15	0.8113	0.6295
NMF *	n_factors = 160, n_epochs = 90	0.9392	0.7355
ALS	MaxIter=20, RegParam=1.0	1.6050	NA

* We do not include NMF in our user interface due to problems with zero division

We can see that SVD performs the best. As a final step, we tested our SVD model on our test set. The final RMSE on the test set is 0.7993 and the MAE is 0.6193. We see that these errors are not much different from the cross-validation errors, showing that our model did not overfit the training set. The errors is relatively high (when the 0-5 scale of ratings is put into context), which is because we can only train on 5% of our dataset. We proceeded to use SVD with the chosen hyperparameter for recommendations.

Figure 4: Recommendations of SVD

```
input_file_name = 'user_input_set.csv'
svd = SVD(n_factors = 110, n_epochs = 90, lr_all = 0.005, reg_all = 0.15)
get_top_recommendations(svd, input_file_name)
```

```
['George T. Stagg Bourbon (Fall 2017)',
 'Kavalan Solist Vinho Barrique Single Cask Strength ',
 "Parker's Heritage Promise of Hope",
 'Colonel E.H. Taylor, Jr. Four Grain Bottled-in-Bond',
 'Laphroaig 32 Year ',
 'Pappy Van Winkle 20 Year',
 'William Larue Weller Bourbon (Fall 2015)',
 'The Macallan Rare Cask ',
 'Lagavulin Distillers Edition',
 'Four Roses Limited Edition Small Batch Bourbon (2015)']
```

5. CONCLUSION

SVD is the chosen model because it yields the lowest RMSE and MAE after cross validation. Surprisingly SVD++ doesn't perform as well although this algorithm incorporates implicit interaction information.

Generally, this recommendation system does capture the flavor preference of the users. We observe that our sample user prefer full-bodied, sweet and rich whiskeys as well as whiskeys that are not peaty or briny, our SVD recommender successfully delivered those recommendations.

Future Work: We hope to leverage deep learning and other bias correction methods to produce a more accurate recommendation system. We would also would like to use cloud computing to be able to process more data, which we believe, will give us better results. Finally, we would like to use the comments for text analysis and topic modelling for recommendations.

REFERENCE

1. Surprise Library Documentation: https://surprise.readthedocs.io/en/stable/
2. R. J. Mooney, P.N. Bennett, and L. Roy. Book Recommending Using Text Categorization with Extracted Information. Proc. Recommender Systems Papers from 1998 Workshop, Technical Report WS-98-08, 1998