REACHING OUT TO OLD FRIENDS: HOW DO WE RECOMMEND WINES TO OUR CUSTOMERS?

BY JESSICA WILLIAMS

Problem

- X-wines has been running into trouble getting retaining their customer base. The percentage rate for repeat customers is at the lowest it's been in the last five years.
- The company is trying to generate ideas to keep current customers making more purchases.
- X wines would like to enhance their customer experiences by recommending wines to customers that they might be interested in.
- Looking to build a recommender system based on the properties of their wine and the customer ratings.

Data

To investigate this question we I will use an X-wines dataset consisting of the following two dataframes:

- X-Wines_Full_100K_wines (csv)-This file contains a list of 100,000 wines and some of their attributes. The attributes include Type, Elaborate, Grapes, Harmonize, ABV, Body, Acidity, Region, Winery and Vintages.PP_recipes (coded/tokenized recipe data)
- XWines_Full_21M_ratings(csv file)- This file contains a set of 21 million user ratings of wines on a scale of 1-5.PP_users (information about users)

Data Wrangling

This was a fairly tidy dataset so I spent most of my data wrangling examining the data:

General cleaning and examination for the wine list data:

- Dropped the website column from the dataframe.
- Examined the value counts of most of the attribute to view the available responses for each variable.
- Made sure no specific wine ID was repeated in the dataframe.
- Found the value range for the ratings:1-5.
- Counted the number of unique raters in the wine ratings dataframe.
- Created a mean rating column that represented the mean of the ratings given for each unique wine.

EDA: Organization

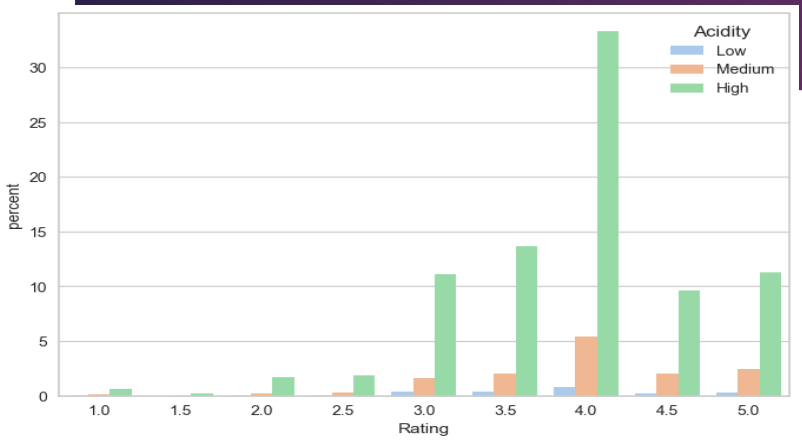
New Features:

- Rater count column: sums the number of rating rows for each unique rater.
- Rater mean column: average rating for each individual user.
- Dummy variable: for categorical wine attributes variables.

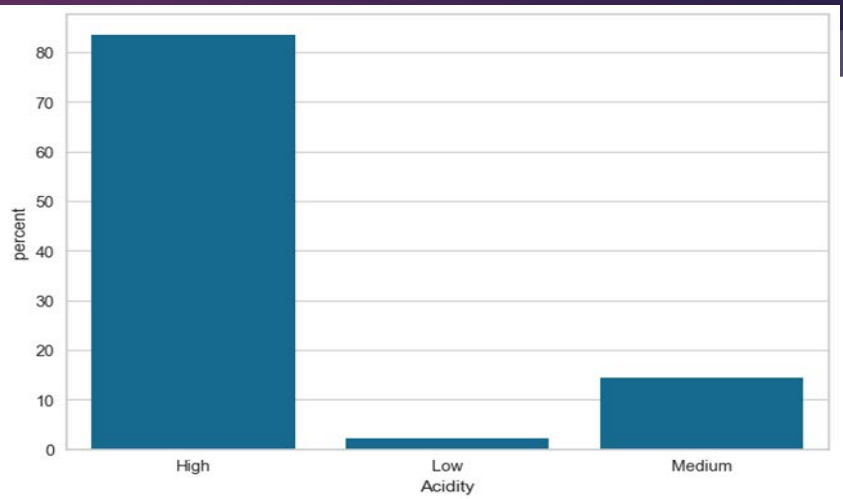
Other:

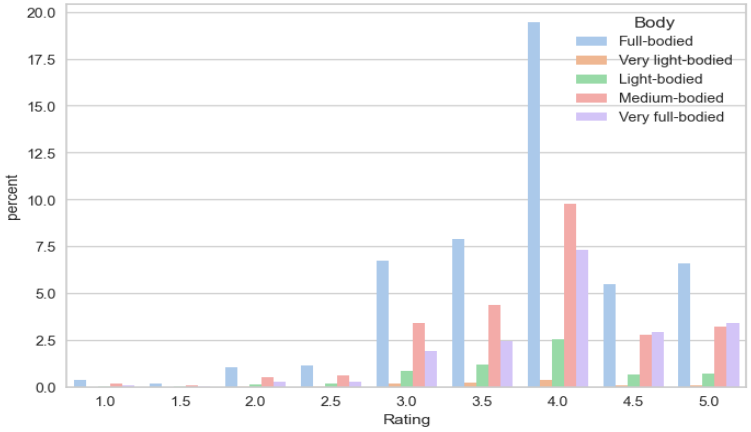
- Merged wine list to wine ratings dataframe on WineID.
- Added dummy variables post merge.

EDA: Distributions

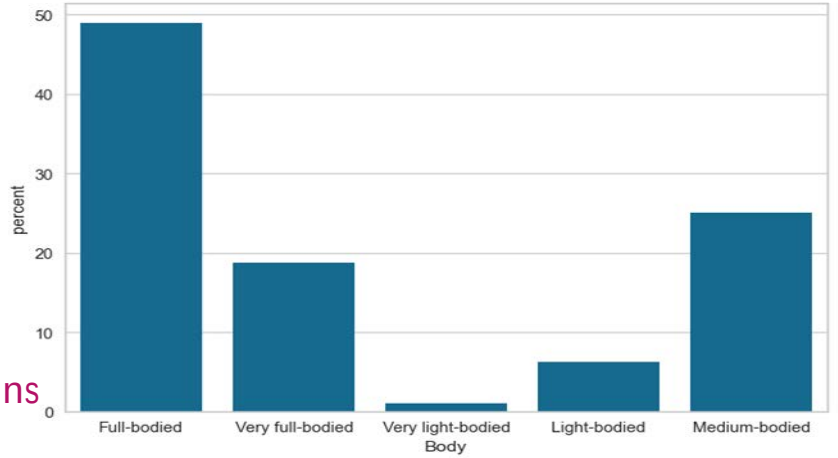


Acidity Variable distribution





Body Variable Distribution



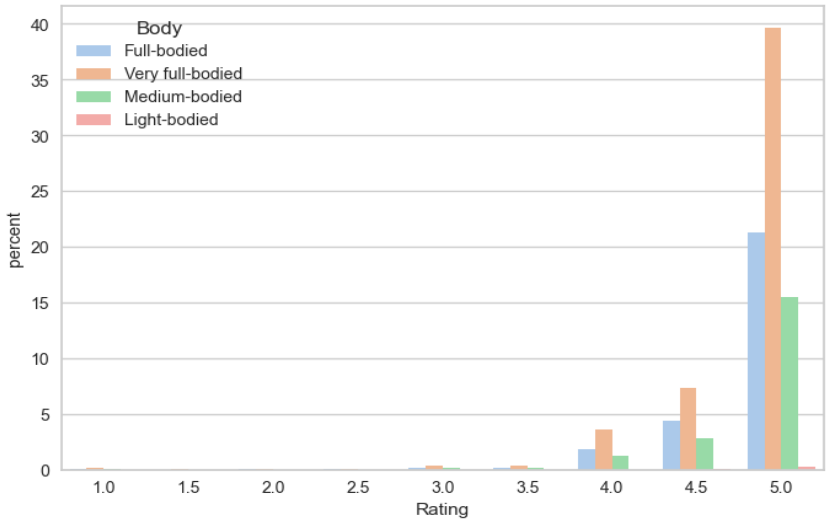
Both variables have similar distributions over the ratings a they do in the overall dataset.

EDA

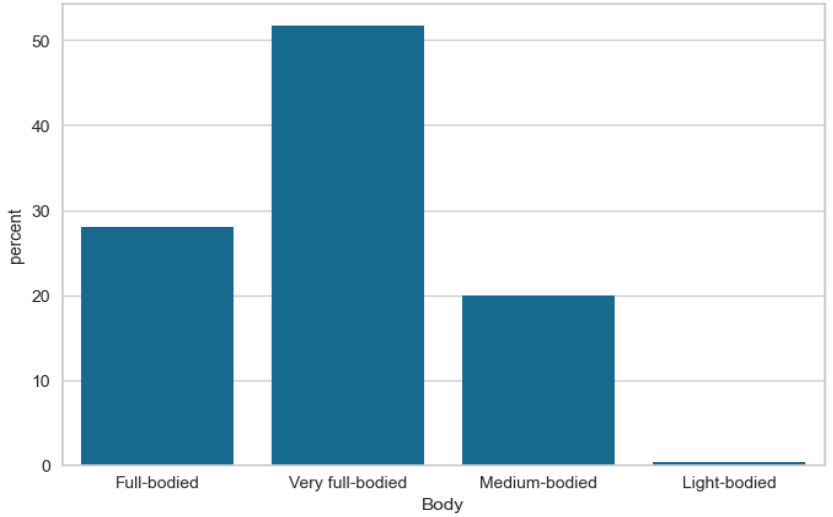
- What If you used only the highest mean ratings in the dataset? Would the distribution change?

EDA: Distributions(Top Subset)

Body Ratings Distribution



Body subset Distribution



The Body variable did have a shift in distribution when examining the top rating subset. Very full bodied Wine now has the highest representation in the distribution.

EDA: Distributions(Subset)

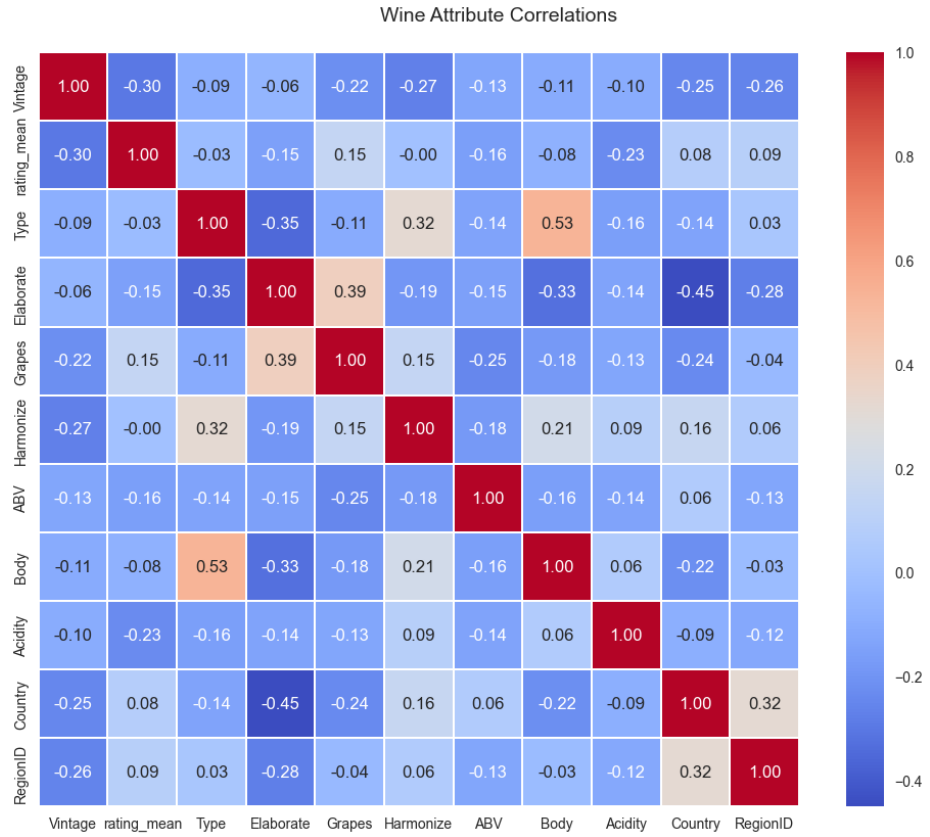
Other Variables that had a distribution shift:

- Elaborate(from Varietal/100% to Assemblage/Blend)
- Winery(from Vega Sicilia to Krug)
- Winery distribution also changed for frequent rater subset(top changed to M. Chapoutier)

EDA: Correlations?

Wine Attributes:

- Highest correlation is between ‘Body’ and ‘Type’ at 0.53.
- Quite a few weaker negative correlations.
- No strong/significant correlations.



Modeling:Testing Models

To preprocess for modeling, I started by parsing out the user id, wine id and rating from the data. From there I created the surprise data set to cross-validate our models on. The models I tested on the data were as follows:

- Normal Predictor model: Mean RMSE =0.906
- SVD Model: Mean RMSE =0.489
- KNNWithMeans: Mean RMSE =0.524
- KNNBaseline: Mean RMSE =0.487

Modeling:Testing Models

- Normal Predictor model: Mean RMSE =0.906
- SVD Model: Mean RMSE =0.489
- KNNWithMeans: Mean RMSE =0.524
- KNNBaseline: Mean RMSE =0.487

Best Model: KNNBaseline:

Possible Option: SVD Model: Mean RMSE

- SVD Model has similar(lhigher) mean RMSE and much faster fit and test times.

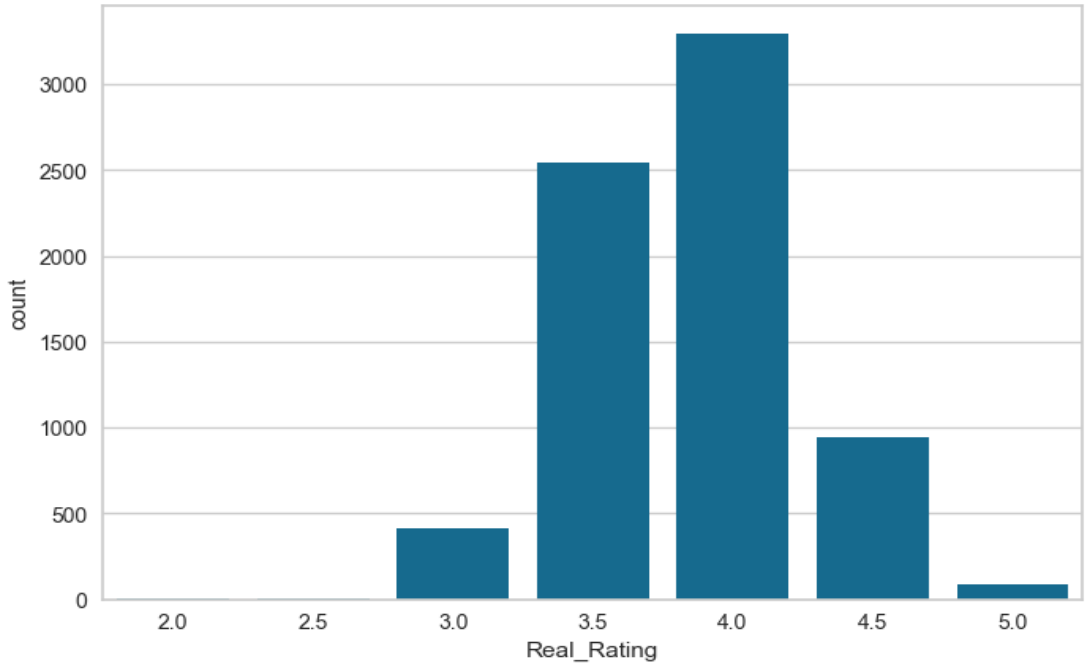
To apply the model and get predictions, I created a dataframe that displays the userID, wineID, the actual rating and the predicted rating from our model. I also included an error column that displays the difference between the actual rating and the predicted rating.

Modeling: Results(KNN)

- The KNN Baseline Model produced 7278 estimated ratings out of 150,000 with errors less than 0.1.
- From the distribution it seems this model is best at predicting ratings of 4.0. It is also worth noting that this rating distribution is similar to the total data set.

```
low_error.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7275 entries, 5 to 37492
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   UserID           7275 non-null   int64
 1   WineID           7275 non-null   int64
 2   Real_Rating      7275 non-null   float64
 3   Estimated_Rating 7275 non-null   float64
 4   Error            7275 non-null   float64
dtypes: float64(3), int64(2)
memory usage: 341.0 KB
```

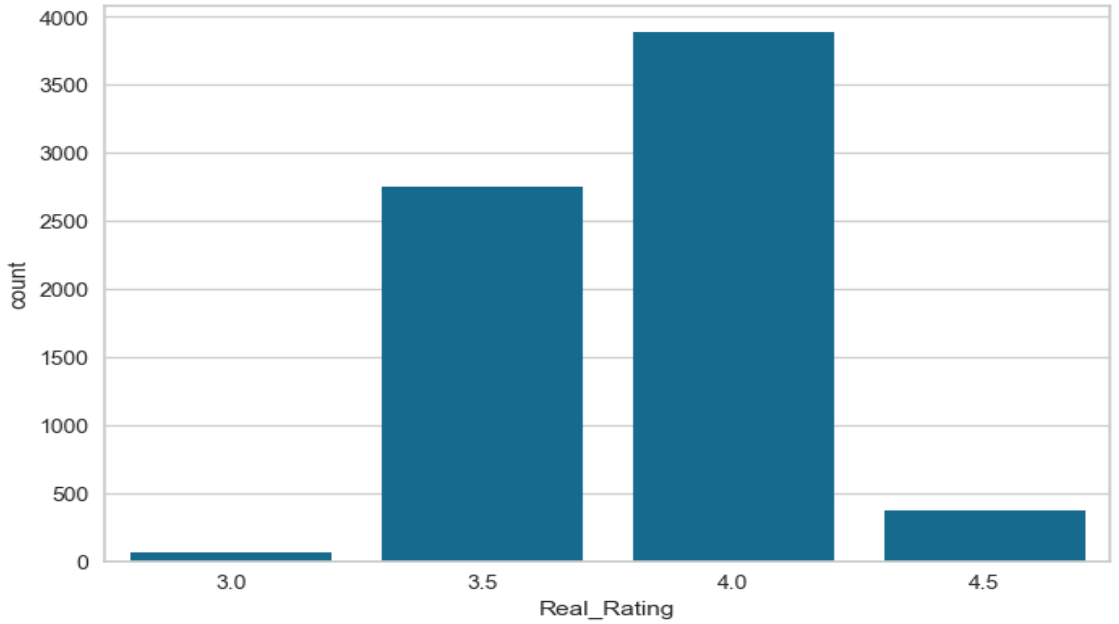


Modeling: Results(SVD)

- The number of predictions in this data subset is 7070. This is similar to but less than the number of low error predictions from the KNN baseline model.
- The rating distribution for this model is also very similar to the KNNBaseline model with the top represented rating being 4.0.From the distribution it seems this model is best at predicting ratings of 4.0.

```
low_error2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7070 entries, 2 to 37496
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   UserID           7070 non-null   int64
 1   WineID           7070 non-null   int64
 2   Real_Rating      7070 non-null   float64
 3   Estimated_Rating  7070 non-null   float64
 4   Error            7070 non-null   float64
dtypes: float64(3), int64(2)
memory usage: 331.4 KB
```



Conclusions

After using each model to create predictions it looks like the KNNBaseline model produces better results than the SVD model.

- The KNN Baseline model has a better RMSE and has more estimated ratings with low errors than the SVD model.
- Could we possibly still use the SVD Model in the interest of saving time and resources?

The End!