Recommendation for Online Retail Data

Sumit Patel

Springboard Capstone #2

Abstract

Recommendation Systems are widely used to recommend products and services to customers and clients. This system allows for the prediction of the rating or preference a user would give to an item. The dataset is for a UK-based and registered non-store online retail. The company mainly sells unique all-occasion gifts. The dataset is comprised of 8 attributes consisting of InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID and Country. The dataset is composed of 541,909 rows. Through exploratory data analysis and using graphlab we will identify how to make a suitable recommender systems for the dataset. Appropriately matching customers to items that they may be more inclined to purchase would increase the likelihood of another purchase using recommender systems approach.

UCI Machine Learning Repository Dataset can be found at:

https://archive.ics.uci.edu/ml/datasets/Online+Retail

Recommendation for Online Retail

Recommender systems provide recommendations through collaborative filtering, content-based filtering or hybrid approaches. Collaborative filtering approaches build a model from a user's past behavior as well as similar decisions made by other users which is then used to predict items that the user may also have interests in. Content-based filtering approaches utilize use a series of discrete characteristics of an item in order to recommend additional items with similar properties. Content-based filtering is based on a description of the item and a profile of the user's preference. These approaches are often combined in hybrid systems. The dataset features would suggest that collaborative filtering would be ideal for our purposes. We will train and test various recommender systems to find one that is most suitable for this dataset application suing graphlab.

Method

Data

Dataset is a UCI Learning Repository dataset that consists of 532,618 rows of online retail data and 8 columns consisting of attributes InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID and Country. The dataset can be found at: https://archive.ics.uci.edu/ml/datasets/Online+Retail. Our recommender system will be based on the features of CustomerID, StockCode, Description and Quantity to provide our recommendations.

Exploratory Data Analysis (EDA)

The dataset exists in xslx format which we used to create a dataframe using pandas. Examining the data we can see that it is in fact consisting of 541,909 rows. Using pandas command we first analyze the data using the describe function. From this we determined that there are only three features comprised of integers being Quantity, UnitPrice and CustomerID. CustomerID will need its datatype changed. The mean quantity is 9.552250 and the mean UnitPrice is 4.611114. Exploring the other features we determine that Country counts reflect the order counts of each country. We have generated the plot using tableau as the computational processing power of my laptop was not able to handle the load on memory for the plots from the dataframe.

Initial observations on dataset were to determine what variables we would select for our recommendation parameters which require triplets composed of user_id, item_id and target data. User_id will be our CustomerID, item_id will be composed of StockCode and target data should be an aspect of our data that we can use to rank or measure for recommendations. Exploring the data we can observe the feature of Quantity which would be suitable for target variable. Quantity is a good measure as it reflects how often or how much of an item is purchased. We can use this to determine the recommender suggestive metric for determining a recommendation as the behavior is comparable to other customer purchase behaviors.

After selecting the features of interest we then convert the dataframe into an SFrame to use in graphlab. We will be using the features of CustomerID, Items (which we derived by concatenating StockCode with Description) and Quantity. Grouping CustomerID and Items we reduce our dataset to 274,339 rows of unique CustomerID instances. As we observed the

Quantity feature has a max value of 80,995 , mean of 9.55225 and a min value of -80,995. Using the normalization in this process we can account for users that did not make a purchase, returned items or otherwise by providing a standard normalized weight to each value.

Figure #1 DataFrame Description

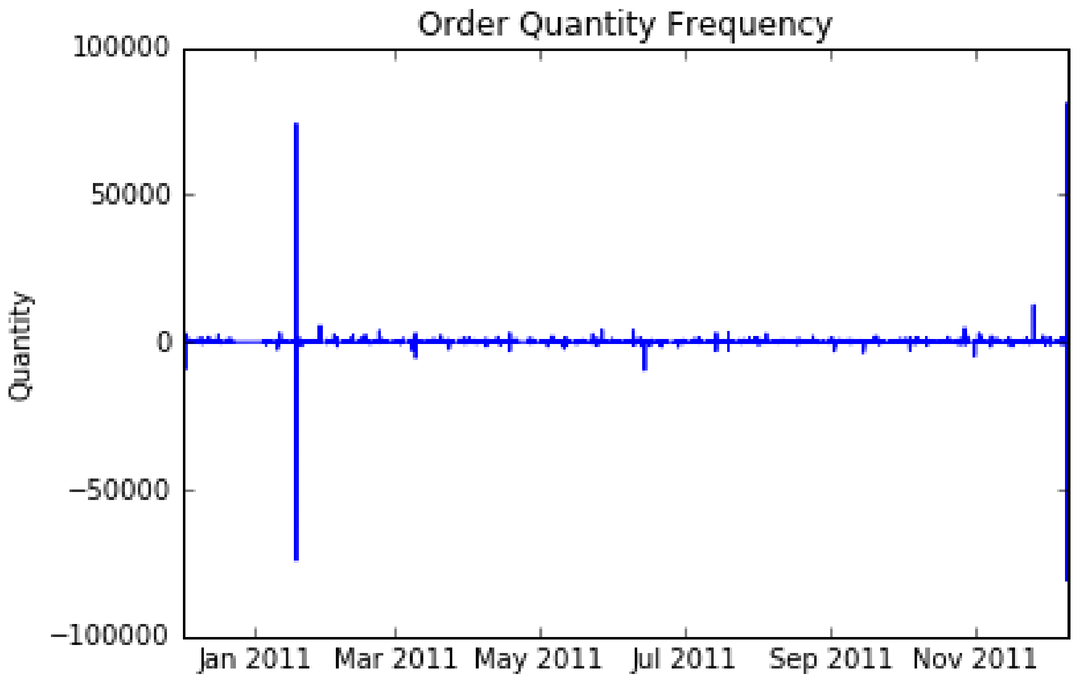
	Quantity	**Unit Price**
count	**541909**	**541909**
mean	**9.552250**	**4.611114**
std	**218.081158**	**96.759853**
min	**-80995**	**-11062.06**
25%	**1**	**1.25**
50%	**3**	**2.08**
75%	**10**	**4.13**
max	**80995**	**38970**

This table show the basic analysis of the dataframe from numerical features.

Figure #2 Distribution of Items Ordered

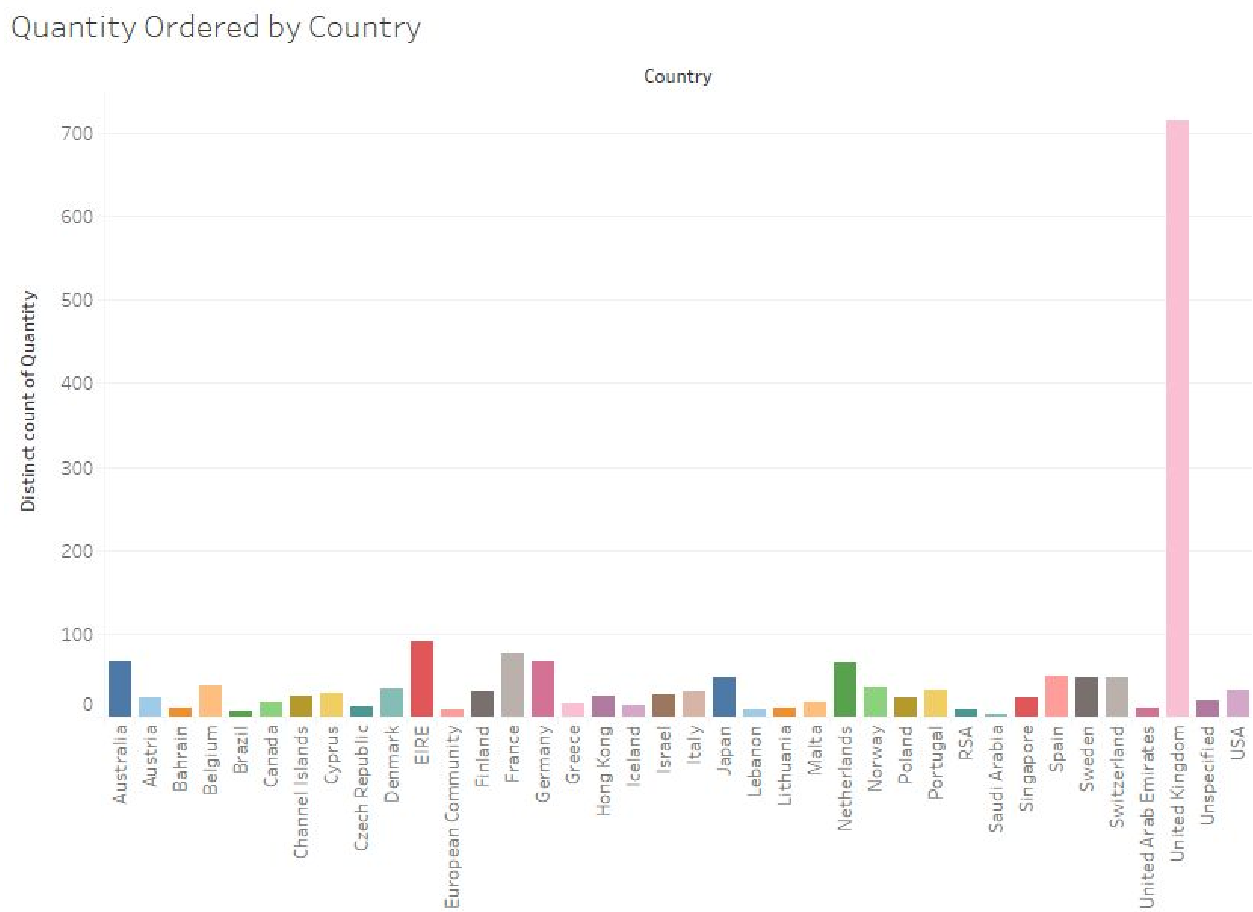
This plot shows the distribution of the items ordered. Clearly we can observe that the top 40% of items are mostly ordered.

Figure #3 Quantity Ordered by Country



This plot shows frequency of Quantities ordered over time. We can clearly observe a spike early on in the year and another at the end of the year.

Figure #4 Quantity Ordered by Country



This plot shows that a significant amount of orders come from United Kingdom. The rest of the countries do not even come close to the order frequency of United Kingdom.

Using protocols and functions available in graphlab we were able to apply recommender create to our SFrame to find out which recommender system model would be suggested, the result was ranking factorization model. This is a collaborative recommender that learns latent factors for each user and item and uses them to rank recommended items according to the likelihood of observing user-item pairs. This result is what we desire as collaborative filtering would be the ideal recommender system for our dataset. We will now test the model performance against other recommender systems and evaluating them with root mean-square error (RMSE) metric. RMSE is the measure of the differences between values (sample and population values) predicted by a model or an estimator and the values actually observed. We will be comparing factorization, ranking factorization, popularity and item similarity recommender systems for our observations. Recall metric reflects the ratio of items that a user likes that were actually recommended. Precision is the metric of recommendation that the user actually liked. RMSE will be the evaluation metric for measuring prediction errors. Ideally a low RMSE would be desired reflecting a better recommendation.

Figure #5 RMSE of Models on Original Dataset

Recommender System	RMSE Overall
Recommender	107.45061613654305
Factorization	56.280878107633875
Ranking Factorization	90.10179238695382
Popularity	100.02054212968166
Item Similarity	99.81926838136826

This table shows the RMSE overall values of the models using the original value of quantity as target for the recommenders. From the table we can see that the RMSE values a fairly high for our purposes but given the results we see that the factorization recommender is the ideal recommender using this quantity value. We see that the error-rate is of approximately 56 items which is not good enough for our purposes. Let's try normalizing the quantity values using the z-scores of quantity as target.

Normalization of Data

Our observations from the original dataset showed that normalization of data may be required to scale our data to provide a normal distribution. The RMSE observed was very high across all models but least of all in factorization. We can see though that factorization model had the best RMSE granted that the error rate generated is about 56 items. In factoriation the users and items are represented by a fixed length vector, or a latent factor to provide the dot product of the factors to give a predictive rating. We need to derive the mean and standard deviations of Quantity to perform our z-score standardization manually since feature scaling is not available in graphlab. Standardization or Z-score normalization entails rescaling the features to reflect the properties of a standard normal distribution having a mean of zero and a standard deviation of 1. Z-score is calculated by subtracting the mean and dividing by the standard deviation from the feature that is being scaled. This allows us to distribute the model weights equally. In this dataset that means to take the quantity measures to be standardized to distribute the measures equally so that there is no value heavily or lightly weighted to skew our observations. Removing the mean and scaling to unit variance allows us to center our observations independently otherwise it may not work in objective functions for the recommender.

Figure #6 RMSE of Models on Normalized Data using mean

Recommender System	**RMSE Overall**
Recommender	3.025176044046389e+18
Factorization	3.02517610215155e+18
Ranking Factorization	3.02517610215155e+18
Popularity	2.94856491969459e+18
Item Similarity	2.948979229494945e+18

This table shows the RMSE overall values of the models using the normalized data with z-score of quantity as target. We can see that the RMSE improved drastically with an average error rate of 3 items for recommendation. From this we also see that our result changed to popularity recommender which had a slightly better score than item similarity.

Model Selection

After ETL and EDA we continued to scale our data for our observations. We created and implemented training and test sets to test our various models and their performance. The metrics used to measure model performance were RMSE. We then scaled our data to provide a more normal distribution using z-score method. We took the mean of each quantity and the standard deviation to calculate the z-score which we then used as our target variable since it is normalized because it has a mean of zero and standard deviation of one. Our observations reflected improvement from original data observations which means that our data benefited from z-score normalization.

Conclusion

Evaluation of the models utilized the RMSE metric to measure the error rate of the recommendation systems by units of items. Our results and calculations did provide that the original dataset would benefit from normalization to scale our data by a vast margin of improvement. Using mean and standard deviation we were able to calculate the z-score and improve the model RMSE values. We were able to notice that the best RMSE lied with the popularity recommender system with an error-rate of less than 3 items. The popularity recommender would be the choice for recommender in this application due to its performance and minimal error rate.

References

Daqing Chen, Sai Liang Sain, and Kun Guo, Data mining for the online retail industry: A case study of RFM model-based customer segmentation using data mining, Journal of Database Marketing and Customer Strategy Management, Vol. 19, No. 3, pp. 197â€“208, 2012 (Published online before print: 27 August 2012. doi: 10.1057/dbm.2012.17).

http://archive.ics.uci.edu/ml/datasets/online+retail