Recommendation Engine for Investors/Traders:
Exploring Options on Algorithmic Broker Recommendations

Capstone Project Summary - MS Finance
2021/2022 Spring
Central European University
Balint Bojko

Contents

Introduction... 1

Establishing user profiles... 2

Offline evaluation of recommender models... 2

Model building, results and conclusion 3

Introduction

The sponsor of this MS Finance Capstone Project was BrokerChooser, a broker recommendation and comparison platform. BrokerChooser has extensive data on 100+ brokers across 9 dimensions: fees, security, deposit and withdrawal, trading platforms, account opening, product portfolio, customer service, education and research. The platform provides users information in the form of reviews and comparisons on advantages and disadvantages of broker services and helps them choose a broker that best fits their needs. The business of BrokerChooser builds strongly on affiliate partnerships with brokers – they place links on their website that redirect users to broker pages where they can start a sign-up process. BrokerChooser receives revenue from these redirections.

In addition to being a broker intelligence platform, BrokerChooser can actively recommend brokers to its users through its "Find My Broker" feature. The purpose of this feature is the identification of user preferences and the personalized recommendation of brokers based on these preferences. When interacting with this feature, answers to 4 simple questions are used to identify the top 5 most appropriate brokers for a given user. The list of these questions can be seen in *Table 1* below.

Question	Options to choose from
Where do you live?	List of countries
What is your trading experience?	first-timer / know basics / can handle complex transactions / professional
How active will you be?	daily / weekly / monthly / yearly trading
What product will you primarily trade?	Stocks and ETFs / Forex / Options and futures / Funds / CFDs / Cryptos

Table 1. List of questions BrokerChooser asks users to determine their preferences. Source: www.brokerchooser.com

This recommendation method requires users to answer multiple questions, and derives recommendations based on the answers. A main challenge of this method is that the user must explicitly input a set of parameters into the recommendation model and is therefore required to spend time getting his/her own recommendations.

An alternative way of recommending brokers was proposed, that may build on implicit user preferences. Implicit user preferences can be inferred from behavioral data such as page visits and clicks on links to external (broker) websites. Broker recommendations then, could be made according to those inferred preferences automatically. The purpose and goal of this Capstone project was to establish whether such recommendations would be possible on the platform.

Establishing user profiles

The first challenge in creating a recommendation methodology without answers to the BrokerChooser questionnaire is the inference of user preferences from behavioral data. User preferences must be accurately estimated to be able to create good recommendations. Available data to infer these preferences include website visits and clicks on links to external (broker) websites. Preferences were established by mapping each user-broker pair using 4 possible values: a rating of 1-3, or null. The description of what each value meant can be found in *Table 2* below.

Rating	**1**	**2**	**3**	**null**
Description	User reads article about broker with other brokers mentioned in it	User reads article about broker with no other brokers mentioned in it	User clicks on link that leads user to broker website	No user action registered for a given broker
Level of inferred user preference for broker	Low	Medium	High	NA

Table 2. Levels of user ratings as inferred from behavioral data collected from website.

User profiles of broker ratings were inferred algorithmically using website usage data and the preference assignment rules mentioned above. Recommender models were built using the resulting user profiles.

Offline evaluation of recommender models

Generally speaking, the proper way to evaluate the performance of a recommender model is through live, A/B testing on the website/platform it is built for. In our case however, this was not a possibility, as we could only use historical data. We needed to test the performance of the recommender models using an “offline” evaluation. Offline evaluation meant splitting a chunk of historical data to train and test data sets, and then, building the recommender models on data from the train set, and evaluating their performance on the “unseen” test set.

Model building, results and conclusion

Four types of recommender models were built: content-based filtering, user-based collaborative filtering, item-based collaborative filtering and matrix factorization based recommender models. Each type of model was trained using a different algorithm. Implementation of the first three was completely manual (i.e. almost all required code written by me). Implementation of the matrix factorization algorithm was done using the *Implicit* python library.

Performances of the models were evaluated using the precision evaluation metric, which means that for each user and model, the % of correct recommendations was established for the test data set. Then, for each model, the averages of these user-level precision metrics were taken. The average precision across users was the final performance evaluation criterion for each model.

The performances of the models were compared against a baseline model that worked by recommending the most visited brokers to each user. In the results, none of the four listed recommender models could outperform the baseline model. This result was attributed to some key challenges of building a recommender system in the business domain of BrokerChooser. Details of these challenges can unfortunately not be shared here due to a confidentiality agreement.

It was concluded that building a well-functioning broker recommender system is a realistic possibility for BrokerChooser but is not a simple task and requires resources. The timeline to build a prototype engine may reach the scale of a couple of months, if not done on a full-time basis. Considering the growing user base of BrokerChooser it can be a worthy long-time investment as it may increase conversion rates significantly. In time, the larger the user base of BrokerChooser grows, the more relevant this topic may still become.