Predicting purchases based on user clicks (RecSys Challenge)

Vladimir Cristea, Minna Bergendal, Philipp Mapara, Petros Polychronis

1 Introduction

Digital marketing recommendation systems utilize collaborative and content-based filtering to provide personalized suggestions for users. The growth in popularity of such algorithms in e-commerce over the last years has a twofold justification: users are able to find the desired products faster and more easily, while e-commerce businesses can boost their sales and reduce the risk of churn. According to the Boston Consulting Group, companies that integrate advanced digital technologies and proprietary data to create personalized experiences tend to see a 6% to 10% revenue increase, typically two to three times faster than their competitors[1]. Hence, companies adapting digital recommendation systems stand to capture customers, market share, and improve their bottom line.

The objective of this report is to predict whether a user (session) will make a purchase or not based on their clicks, and, if the user is predicted to buy at least one product, predict the product(s) bought. In response to these two subgoals, instead of maximizing a common evaluation metric for classification problems such as misclassification error, this challenge maximizes a scoring function (Equation 1) to account for: (1) the accuracy for predicting buy or not; and (2) the accuracy of predicting the items bought.

$$Score(Sl) = \sum_{s \in Sl} \begin{cases} \frac{|S_b|}{|S|} + \frac{|A_s \cup B_s|}{|A_s \cap B_s|} & , if s \in S_b \\ -\frac{|S_b|}{|S|} & , \text{otherwise} \end{cases} \quad (1)$$

where Sl are the sessions in the solution file we submit, S are all the sessions in the test set, s is a session in the test set, S_b are the sessions in test set which end with a buy event, A_s are the predicted bought items in session s, and B_s are the actual bought items in session s.

To achieve a high evaluation score, we have constructed a feature space that is correlated to the users' purchase behaviour and built a machine learning model to predict item purchases. In this paper, we present a content filtering approach due to a lack of user ID and variables, which are necessary for collaborative filtering. In other words, not having any information about the demographics, interests or other personal attributes of the users, but knowing their browsing history, we use the latter to predict their behaviour. The algorithm finds patterns in session- and item-related features, and produces a binary classifier to determine whether an item is to be purchased in a session (i.e., if an item-session combination is likely to be a buy).

2 Data Analysis and Feature Engineering

2.1 Data Description

The data comprises a large collection of digital user visits to a retailing website, which consists of two training files: **yoochoose-clicks.dat** contains session data describing user clicks (session ID, timestamp, item ID, category), and **yoochoose-buy.dat** contains purchasing history data (session ID, timestamp, item ID, price, quantity). Each user visit (session) is comprised of a sequence of item clicks and purchases. The size of the **buys** and **clicks** data sets is 1,150,753 and 33,003,944 entries respectively. To conduct our analysis, we divide **clicks** into a training set (roughly two-thirds) and a test set based on timestamp, and ensure that no session's clicks are divided across the two sets.

In total, the data set contains 19,949 unique items in the webstore and 9,249,729 distinct sessions. The large number of unique items enhances the difficulty of predicting users' purchase list as the target function is to predict the set of items bought in a session s, which is a subset of all items.

2.2 Feature Engineering

As per the two sub goals of the algorithm: (1) predicting whether a user is going to buy (2) if yes, then predict which items are to be bought, we extract session related features and session-item related features. The former refer to a user (session) and their behaviour, and are naturally the same for all session-item combinations of a given session. The latter refer to the behaviour of a user (session) with regard to a specific item, meaning they will be different for every observation. There are also two item-related features, popularity and category, which do not depend on the session. Raw identifiers and sophisticated aggregations of items and categories are added to the feature space. Based on purchase data analysis and digital marketing domain knowledge, the following features (Table 1) are identified.

Session related features	Item related features
Time spent on website	Time spent on item
Number of clicks per session	Number of clicks per item
Timestamp-related features	Item appearing ≥ once
(month, day, hour)	Item being clicked first
Max. no. of clicks on a item	Item being clicked last
Popularity of all clicked items	Popularity of item
Max. time spent on a item	Category of item
No. of distinct clicked items	
No. of clicks on each category	

Table 1: *Session and session-item related features*

[1] https://www.bcg.com/publications/2017/retail-marketing-sales-profiting-personalization

3 Methodology

3.1 Outline

To solve the two-step problem, the intuitive method would be to train two models, one for predicting whether a session will end in a buy or not with session related data, and another model to predict the purchased items for sessions that are predicted to buy from the previous model with session-item related data. However, this approach did not have the desired results because of limited features in each model. For example, in the second model, predicting purchased items, a user's general behaviour in the session may provide informative information to whether they will purchase an item or not; therefore, excluding these features diminishes the explanatory power of the model and does not capture the interaction between session related features and session-item related features as subtly.

Hence, we have reconstructed the data points as entries of session-item (Table 2) where the target of the classifiers is **buy or not buy an item in a session**.

session ID (s)	**item ID** (i)	**session-features**	**item-features**	**buy**
1	214536500	...	...	True
1	214536502	...	...	False
...	...	...	...	...
7448401	214602489	...	...	?

Table 2: *Reconstructed data points*

With the results of the classifier, the prediction of whether a session made a purchase or not (goal one) is:

$$Purchase_s = \begin{cases} True & \text{, if at least one } buy_{si} \text{ is True} \\ False & \text{, if all } buy_{si} \text{ are False} \end{cases}$$

Furthermore, the prediction of which items are to be bought in a session (goal two) are the rows where buy_{si} is True. As a result, we can construct a table of sessions that are predicted to make a purchase with a set of items that are to be bought (Table 3).

session ID	**items**
7448401	{214602489}
7448402	{214556563, 214835064}
...	...

Table 3: *Predicted items to be bought in sessions*

3.2 Classification Algorithm

Random Forest (RF) is chosen as the classifier due to its flexibility and robustness. RF can handle mixed data types and unnormalized data easily. As we are given a large data set, it can produce a higher accuracy by reducing variance due to its ensemble nature. Moreover, RF can detect non-linearity, whereas for many other algorithms such as logistic regression and linear discriminant analysis, the decision boundaries are linear. While it is possible for a logistic regression model to capture non-linear patterns, that requires additional feature engineering and, perhaps, domain-specific knowledge in order to manually create new features. Conversely, tree-based methods result in non-linear decision boundaries by their nature. Without knowledge of the linearity of the features and target, we choose a flexible algorithm to capture underlying relationships. Lastly, RF is more robust to outliers in the high dimensional data created with feature engineering by constructing multiple trees to reduce variance in our scoring function.

Imbalanced data: With all the advantages the random forest presents, trees generally do not perform well with unbalanced data sets. The constructed data set has far more data points where buy equals False than those with True. Therefore, we implemented stratified sampling for the training data set to create a balanced training set.

Cross-validation with evaluation score: The evaluation score represents the asymmetrical costs of misclassification. In digital marketing, there is a limited budget and e-commerce companies would rather not advertise than spending advertisement budget on customers that will not convert to sales. To account for this, when performing cross-validation, we used our scoring function defined earlier rather than simply try to find the best-performing algorithm in terms of classification.

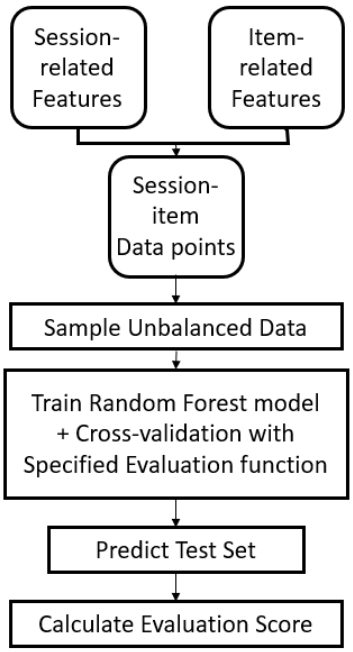

Figure 1: Methodology of classification algorithm

Given superior computational resources such as those to be found in a business environment of an e-commerce organisation, we would tune the hyperparameters of the forest via k-fold cross-validation. To do so, we would perform a search on the following parameter grid: the number of estimators/trees (1, 10, 50, 100, 200, 500, 1000), maximum number of features considered at each split ($\sqrt{p}$, $log_2(p)$), the impurity criterion (Gini Index, Cross-Entropy), and the maximum tree depth (full tree, 5, 20, 40, 75, 100).

In practice, due to computational reasons, it was not possible to run such an ample cross-validation procedure on data set as large as this one. Instead, we have tried a number of different combinations of parameters manually on different machines, and selected the model that yielded the best score on validation data in the end: a random forest with 35 fully-grown trees, $\sqrt{p}$ features considered and splits judged on Gini index. It is to be noted that, with more computational power, a higher number of trees could perform better.

After selecting the highest performing model with cross-validation, the algorithm we devised proceeds to predict the test set. Note that the test set is unbalanced data that is the last one-third of the data set based on its timestamp.

4 Experiment / Outcomes

Apart from a Random Forest, other classifiers were implemented as benchmarks. Due to the asymmetrical nature of the evaluation metric, the Neural Network appears to perform worse than the Naive Classifier, which always predicts majority class ('False'), leading to a score of 0. This is likely because it overfits the data set and aggressively predicts 'True' for session items, which led to a negative evaluation score. This reflects the costs of advertising in digital marketing, as a company would rather not advertise than spend the budget on advertisements that are not effective. Moreover, logistic regression without regularization, with LASSO (l1 norm regularization) and Ridge (l2 norm regularization) are implemented for comparison. These methods have worse performance, possibly due to the linear decision boundary that cannot capture non-linear relationships between features. The results from all models we ran are displayed in Table 4.

The original score of our random forest was 19,175. A fruitful analysis of its performance is understanding how this score is broken down. It appears that the value is made up of a score of 92,946 arising from correct predictions, and a negative score of 73,772 from the incorrect predictions. We can observe that the main hurdle to a better score is that we have a relatively large negative component, which means that even this model, which clearly outperforms all others we have attempted, is still slightly too sensitive in predicting that a session will end with a buy event, which leads to a large number of false positives. In a digital marketing context, these errors represent an inefficient spend of budget on users unlikely to purchase.

To mitigate this, we changed the prediction threshold from the default 50% to 65%. This threshold restricts the number of false positives, requiring more certainty that a session will end with a buy before predicting so. This raises the score to an impressive 31,121, made up of a positive 69,610 component and a negative component of only 38,490, which shows a tremendous decrease in our false positive rate.

Random Forest	**Naive Classifier**	**Neural Network**
31,121.88	0	-2755.99
Logistic regression	**Ridge**	**LASSO**
121.98	83.47	83.69

Table 4: *Evaluation score of prediction on test set*

5 Concluding remarks

This report showed how effective machine learning is in a digital marketing context to predict purchases based merely on one's clicks. This reinforces the importance of data-driven marketing, and outlines how the constant reinvention of marketing has caught a new flavour with the advent of modern techniques. Despite the success of our experiment, there are a number of further steps one can take in order to improve the performance of such a model.

Besides the further tuning of the models presented above using a better computational apparatus, the first such steps involve the use of Latent Factor methods and collaborative filtering. Firstly, latent factor methods can detect some latent (unknown) underlying factors that explain users' behaviours. Therefore, not only the extracted features are used, but also linear combinations of them, which can reduce noise and explain the strongest determinants of purchasing behaviour. For example, the famous diaper-beer Walmart analysis [2] pointed out that users that beer and diapers are commonly bought together, which probably indicates that the customer is a newborn's dad. Feature combinations like this can capture customer characteristics hard to detect with the human eye, and may lead to high explanatory power while using less features. This is particularly powerful when the feature space is high compared to the sample size of the data set. However, such an approach comes with high computational costs. In our experiment, our limited computational power could not support the matrix factorization process. In addition, the sparse data set did not provide a low number of latent factors that explains a high proportion of the total variance.

Secondly, collaborative filtering can be implemented with the use of information of the users that initiate each session. This information was not provided in the data set, but e-commerce companies commonly encourage users to create an account to save their personalized purchase history and collect data. Lastly, while our best model was tree-based, a more complex process of model selection could uncover even more suitable machine learning models.

Regardless of the specifics of the models used, one insight is clear: content filtering and collaborative filtering are key tools for any business operating on the internet, and the power of big data analysis for both inference and prediction is difficult to overstate.

[2] https://www.forbes.com/forbes/1998/0406/6107128a.html?sh=56cb26260f33