capillary

Clickstream based Recommendation

Collaborative Filtering of Clickstream Data

Internship Report

By Sankalp P. Apharande

7th May, 2018 to 16th July, 2018

Capillary Technologies Pvt. Ltd.

31/9, Krimson Square, 1st Floor,
Hosur Main Road,
Bengaluru, Karnataka 560068

OVERVIEW

This project is about predicting the items a user will buy in a given stream of clicks performed by him/her in a session. Here, a solution to the 2015 RecSys challenge is presented. The work is based a large scaled dataset of over 9.2 million user-item click sessions. Items recommendations is also done for a given session based on collaborative filtering.

GOALS

1. The main goal of this project is to develop a machine learning model to determine which visits will end up with a purchase and if there is a purchase in a given clickstream session then the second goal is to predict the items that will be purchased.

Here a two-stage approach to item classification from user sessions is used:

1. To predict whether a session will contain a purchase.
2. To predict the items which will be purchased given a session contains a purchase.

Another goal is to use collaborative filtering to develop a primary recommendation system model to recommend items to a user based on his/her clickstream of clicks.

Contents

1. Problem Statement Overview - Clickstream based Recommendation

Many e-commerce businesses use recommendation systems to suggest items to their users. Recommendation systems have become a crucial tool for turning a casual browsers in potential customers and ultimately for the success of e-commerce website in today's highly competitive e-commerce environment. In this report, the solution to RecSys challenge 2015 is provided. Along with the classification problem, a recommendation system model based on collaborative filtering is also developed.

RecSys Challenge 2015 : A Classification Problem

In this challenge, YOOCHOOSE (Germany based Recommendation System provider company) provided a collection of sequences of click events; click sessions. For some of the sessions, there are also buying events.
The data comprises a large collection of user visits to a large European retailer's website. Each user visit is also referred to as a session, and each session is comprised of a sequence of item clicks the user performed during the visit.

The goal is hence to predict whether the user (a session) is going to buy something or not, and if he is buying, what would be the items he is going to buy.

Recommendation System:

The features calculated for classification model can be used for giving recommendations. Given a target session's record of activities, collaborative filtering based techniques such as *k*-Nearest-Neighbor approach, the given session can be compared with other sessions in order to find top-*k* sessions which have similar interests.

The mapping of a given session to its *neighbourhood* is based on the features of a given session.

2. Literature Survey

To provide a starting point for my work, the following two papers were studied as a part of my literature survey:

1. Improving the Effectiveness of Collaborative Filtering on Anonymous Web Usage Data By (Mobasher, Dai, Luo, & Nakagawa)

2. Two-Stage Approach to Item Recommendation from User Sessions By Maksims Volkovs

In *Improving the Effectiveness of Collaborative Filtering on Anonymous Web Usage Data, Mobasher et. al* study the impact of various preprocessing techniques applied to clickstream data, such as clustering, normalization, and significance filtering, on collaborative filtering. Their experimental results, performed on real usage data, indicate that with proper data preparation, the clustering-based approach to collaborative filtering can achieve dramatic improvements in terms of recommendation effectiveness, while maintaining the computational advantage over the direct approaches such as the k-Nearest-Neighbor technique. They conclude that with proper data transformation at the preprocessing stage we can significantly improve the effectiveness of collaborative filtering based on clustering while retaining the computational advantage this approach provides over the *k*NN-based techniques.

In *Two-Stage Approach to Item Recommendation from User Sessions, Maksims Volkovs* provides a solution to the 2015 RecSys Challenge. The principal framework of my project is based on the methodology outlined in this paper. The feature calculation in my project is identical to the one used in this paper. A two stage approach for item recommendations is used in this paper.

1. Session Model: To predict whether a session will contain a purchase or not.
2. Item Mode: If yes, then what are the items that are going to be purchased.

In this framework, the probability that a session s_n contains at least one buy event is modelled with sigmoid function. And the probability that a given item d_{nm} will be bought in Session s_n is also modelled by sigmoid function.

For all experiments, a single 75% / 25% training / validation split is used. Primarily concentration on Neural Network and Gradient Boosting Classifier for both session and item models.

For GBM classifiers they used the excellent XGBoost library. They concentrated on logistic regression GBMs with tree boosters and used validation AUC for early stopping.

3. Dataset Description and Challenges Involved

The Dataset

This dataset was constructed by YOOCHOOSE[1]. The YOOCHOOSE dataset contain a collection of sessions from a retailer, where each session is encapsulating the click events that the user performed in the session. For some of the sessions, there are also buy events; means that the session ended with the user bought something from the web shop.The data was collected during several months in the year of 2014, reflecting the clicks and purchases performed by the users of an online retailer in Europe.The work is based on a large scaled dataset of over 9.2 million user-item click sessions.

There are two files in training dataset: Dataset is logs of e-commerce website

1. Clicks Data
2. Buy Data

Clicks Dataset File Description :

The clicks dataset file comprising the clicks of the user over the items.

Each record/line in the file has following fields: *Session ID, Timestamp, Item ID, Category.*

E.g : 1743227, 2014-04-22T17:29:16.095Z, 214629210, 0

- → *Session ID* : The id of the session, represented as an integer number. In one session there are one or many clicks
- → *Timestamp* : The time when a click occurred. Format : YYYY-MM-DDThh:mm:ss.SSSZ
- → *Item ID* : The unique identifier of the item that has been clicked, an integer number.
- → *Category* : The context of the click.
  - ✓ The value "S" indicates a special offer.
  - ✓ "0" indicates missing value.
  - ✓ A number between 1 to 12 indicates a real category identifier. i.e sport
  - ✓ Any other number indicates a brand. i.e BOSCH

Buys Dataset File Description :

The buys dataset comprising the buy events of the user over the items.

Each record/line in the file has following fields:*Session ID, Timestamp, Item ID, Price, Quantity.*

E.g. 420374, 2014-04-06T18:44:58.314Z, 214537888, 12462, 1

- → *Session ID* : The id of the session, represented as an integer number. In one session there are one or many buying events.
- → *Timestamp* : The time when a click occurred. Format : YYYY-MM-DDThh:mm:ss.SSSZ
- → *Item ID* : The unique identifier of the item that has been bought, an integer number.
- → *Price* : The price of the item, represented as an integer number.
- → *Quantity* : The quantity in this buying.

Dataset Statistics :

Full dataset statistics are shown in Table 1. The number of sessions containing at least one buying event are shown in the brackets. The training split and cross-validation split were generated by applying 90% / 10% split to the original dataset. These dataset were used for training and validation of the classification models.

Table 1: RecSys Challenge 2015 Dataset Characteristics

Dataset	No of Sessions	No of Item clicks	No of Buy event
Train	9,249,729	33,003,944	11,50,753 (509,696)
Test	2,312,432	8,251,791	
Train_Split (90%)	8,324,756	29,703,549	1,035,677
Validation split (10%)	924,973	3,300,395	115,076

From the table, we can see that the dataset is large extremely imbalanced.

Only 5.5% of training sessions contains buy events and only 3.5% of the clicked items were bought. Dealing with the imbalanced classes and optimizing the model performance is one of the primary challenge in the problem.

The Challenges Involved :

- Primary Challenges:

  The primary challenge was to predict which visits from the dataset will end up with purchase.And if there is a buy event, then second challenge is to determine which items will be bought.

- Secondary Challenges:

  Secondary challenge was to deal with imbalance classes. Only 3.5% of the clicked items were purchased. Data is significantly imbalanced and optimizing custom evaluation metric is another secondary challenging. Also a very little information was provided for sessions and items beyond their IDs.

4. Mathematical Framework

Notation and Evaluation Framework:
In the dataset, we have N sessions in a set of sessions S, where $N = 9,249,729$.

$$S = \{s_1, s_2, s_3, ..., s_N\}$$

Each session s_n contains M_n clicked items, set of those clicked items

$$D_n = \{d_{n_1}, d_{n_2}, d_{n_3}, ..., d_{n_{Mn}}\} .$$

Note : D_n is a set and it contains the unique clicked items only.

Goal is to develop a model over items $f \ : \ d \rightarrow \{0, 1\}$

$$\text{i.e } f(d_{nm}) = y_{nm}$$

$$\text{Where} \quad f(d_{nm}) = \ y_{nm} = \{ \ 1 \ \ if\ item\ d_{nm}\ was\ bought\ in\ session\ s_n \ ; \ 0\ otherwise$$

NOTE: The defined function $f(d_{nm})$ *is binary*

Also, $y_n = 1$ is used to indicate that session s_n has at least one buy event & $y_n = 0$ otherwise

So, goal is to develop a classification function over given items dataset, which will classify each item in a given session with the aim of maximizing the accuracy of classification.

In the following parts, the approach and methodology is discussed.

5. Approach Towards Model Design

Probabilistic Approach:

As described earlier, here a two-stage approach to item classification from user sessions is used:

1. To predict whether a session will contain a purchase.
2. To predict the items which will be purchased given a session contains a purchase.

We develop a session model for binary classification of sessions and item model to predict whether a given item was bought in that session.

A probabilistic approach is used for predictions:

$$f(d_{mn}) = \{1 \; if \; P(y_n = 1) \; > \varepsilon \; and \; P(y_{nm} = 1) \; > \beta; \; 0 \; otherwise$$

Here, ε and β are thresholds applied to probability predictions of session model $P(y_n = 1)$ and item model $P(y_{nm} = 1)$ respectively. These thresholds are determined by tuning the parameters of the classification algorithms.

In the following section, both model architectures and procedure used to train those models are describe.

Session Model:

Very little information was provided beyond item and session ids, hence a supervised feature-based classification method is used instead of collaborative filtering directly.

As described earlier, the probability that s_n contains a buy event, is modelled with classification algorithms:

1. Logistic Regression
2. Random Forest
3. Gradient Boosted Trees
4. Neural Networks

All the models used characteristic features extracted from each session s_n which is being classified.

Most of the efforts were spent on feature engineering. A total of 30 features were used for training these models.

Session Model Features:

There are 3 types of features used in model building :

1. Session Statistics.
2. Global Features.
3. Time Features.

An information about the helper functions used in features:

$click(\ s_n, d_{nm})$: *No of times* d_{nm} *was clicked in session* s_n.

$buy(\ s_n, d_{nm})$: *Quantity of* d_{nm} *that was bought in* s_n.

$clicked(\ ., d_{nm})$: *Number of training sessions where* d_{nm} *that was clicked.*

$buy(\ ., d_{nm})$: *Number of training sessions where* d_{nm} *that was bought.*

d_{nMn} : *Last clicked item in* s_n

Session Statistics: Features describing general characteristics of session s_n

1. s_n duration in milliseconds
2. Number of clicks in s_n
3. Number of unique items clicked in s_n
4. max time spent on any item in s_n
5. Number of items with 2 clicks in s_n
6. No of items with ≥ 3 clicks in s_n
7. Number of items with category 0
8. Number of items with category "S" (special category)
9. Number of items with category in between (0, 12]

Global Characteristics:

The goal here was to aggregate buy and click information for items in s_n from all training sessions where these items appear. This “global” information is then combined with “local” click patterns in s_n to estimate the likelihood of s_n containing a buy event.

Note: Here $\beta = 10$ is used as a smoothing factor

10. $\sum_{m=1}^{M_n} click(\ .,d_{nm})$

11. $\sum_{m=1}^{M_n} buy(\ .,d_{nm})$

12. $\sum_{m=1}^{M_n} buy(\ .,d_{nm}) \times click(\ s_n\ ,d_{nm})$

13. $\sum_{m=1}^{M_n} \frac{buy(\ .,d_{nm})}{click(\ .\ ,d_{nm})+\ \beta}$

14. $max\ d_{nm}\ \frac{buy(\ .,d_{nm})}{click(\ .\ ,d_{nm})+\ \beta}$

15. $\sum_{m=1}^{M_n} \frac{buy(\ .,d_{nm})}{click(\ .\ ,d_{nm})+\ \beta} \times clicks(\ s_n,\ \ d_{nm})$

16. $clicked(\ .,d_{nMn})$

17. $\frac{buy(.,d_{Mn})}{clic(.,\ d_{Mn})+\beta} \times click(\ s_n, d_{Mn})$

Time Features:

From initial data inspection we found evidence of seasonality where number clicks/buys changed depending on day of the week and hour of the day 3 . Time features were aimed at capturing these seasonality effects. These are categorical features with class [0,1]

18. - 24. Day of the week indicator

25. - 30. Hour of day indicator grouped into six four-hour intervals 0-4, 4-8, 8-12 etc.

These features were calculated for every session in the training dataset and used for training the session model with c4 classification algorithms mentioned earlier.

Item Model:

Item model is used to determine whether each clicked item in a session is bought or not.

Here also a similar supervised feature-based classification method is used.

As described earlier, the probability that each item d_{nm} in s_n is bought or not is modelled with classification algorithms:

1. Logistic Regression
2. Random Forest
3. Gradient Boosted Trees
4. Neural Networks

All the models used characteristic features extracted from each item d_{nm} each session s_n which is being classified.

Most of the efforts were spent on feature engineering. A total of 11 features were used for training these models. In addition to item-specific features, corresponding session information as input to the item model is also included. It is done by passing corresponding session features as an additional input.]

Item Model Features:

There are 2 types of features used in model building :

1. Session Statistics.
2. Global Statistics.

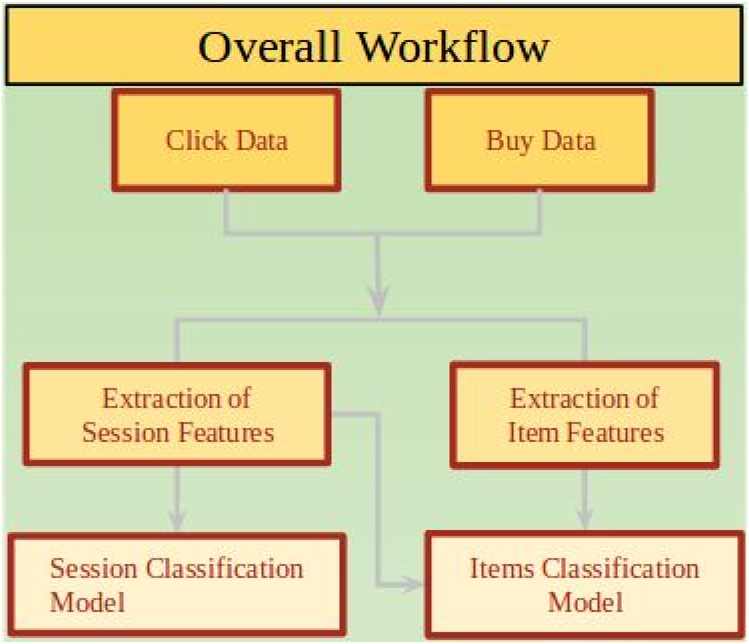
Session Statistics:

1. $click(s_n, d_{nm})$
2. Time spent on d_{nm} in seconds if
3. Last item indicator: 1 if d_{nm} was clicked last in s_n ; 0 otherwise
4. Item category indicator for category 0
5. Item category indicator for category “S”
6. Item category indicator for categories in [0,12]

Global Features:

7. $click(\ .,d_{nm})$
8. $buy(\ .,d_{nm})$
9. $buy(\ .,d_{nm}) \times click(\ s_n\ ,d_{nm})$
10. $\frac{buy(\ .,d_{nm})}{click(\ .\ ,d_{nm})+\ \beta}$
11. $\frac{buy(\ .,d_{nm})}{click(\ .\ ,d_{nm})+\ \beta} \times clicks(\ s_n,\ d_{nm})$

Fig 1. Overall Workflow Diagram


Overall Workflow
Click Data
Buy Data
Extraction of Session Features
Extraction of Item Features
Session Classification Model
Items Classification Model

Note: Corresponding session features are passed as a raw input along with item features in item model

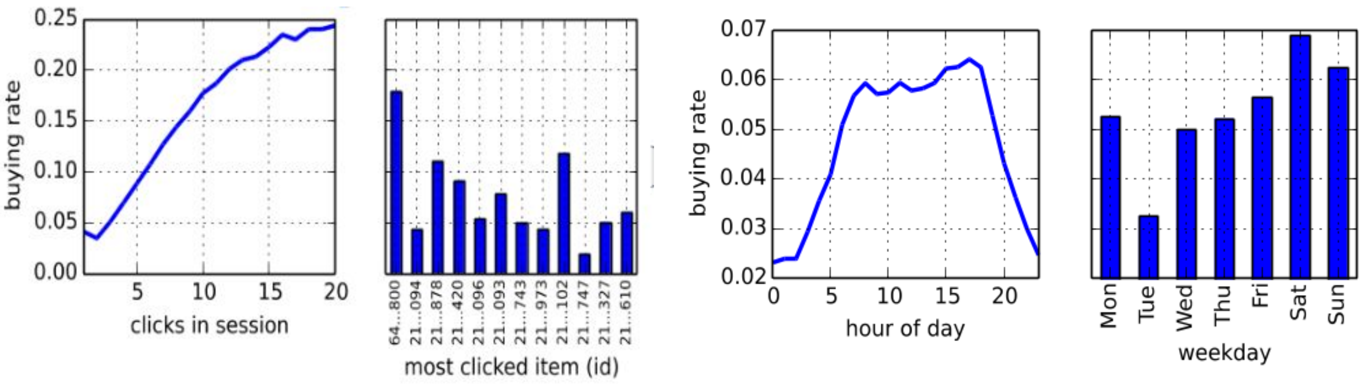


Fig. 2 variation in buying rate as a function of time features

6. Classification Model Development

Classification model development consist of following parts:

1. Calculations of features
2. Feature scaling
3. Model Training
    a. Logistic Regression
    b. Random Forest
    c. Gradient Boosted Trees
    d. Neural Networks

1. Feature Calculation:

    As mentioned in previous section, all the features were calculated for training dataset:

    a. 30 features for Session classification model.
    b. 11 features for item classification model along with corresponding 30 session features.

    In the case of time durations, it should be noted that the time spent by a user on the last item visited in the session is not available. We set the duration for the last clicked item to be the mean time duration for the item taken across all sessions in which the item does not occur as the last clicked item.

2. Feature Scaling:

    Since the range of values of features vary widely, objective functions will not work properly without normalization in machine learning algorithms used.Feature scaling is used to standardize the range of features data, also known as normalization.

Standardization

Standardization method of feature normalization is used here.Feature standardization makes the values of each feature in the data have zero-mean (when subtracting the mean in the numerator) and unit-variance.

The general method of calculation:

1. Determine the distribution mean and standard deviation for each feature.
2. Subtract the mean from each feature. Then we divide the values (mean is already subtracted) of each feature by its standard deviation.

$$x' = \frac{x - \bar{x}}{\sigma}$$

Where x is the original feature vector, $\bar{x}$ is the mean of that feature vector, and σ is its standard deviation.

Feature scaling improved the performance of Logistic Regression dramatically and also improved the performance speed of the algorithm.

3. Model Training :

With these scaled session and item features, model training was done by using following machine learning algorithms:

01. Logistic Regression
02. Random Forest
03. Gradient Boosted Trees
04. Neutral Networks

Model training with neural networks and GBT took a longer time than Random forest and Logistic Regression.

Results of these models are discussed in Results section.

7. Primary Recommendation System Model

Recommendation system:

Another task was to develop a collaborative filtering based recommendation system model in order to recommend items based on similar behaviour in the clickstream.

Collaborative Filtering With Clickstream Data - *k*-Nearest Neighbors approach:

Each session can be viewed as 30-dimensional vector with vector entries as 30 feature values calculated above.

In the case of kNN, the similarity or correlation between the active session s and each remaining session s measured. The top k most similar sessions to s are considered to be the neighborhood for the current active session s.

Here, Euclidean distance is used for similarity measurement. Features extracted from session records after feature normalization are used for Euclidean distance measurement.

Recommendations:

For each session, item recommendations are given based on following steps:

1. Distance Measurement:
   Features extracted feature vector (after feature normalization) is used for Euclidean distance of given active session from remaining session feature vectors.

2. Finding Similar Sessions:
   For a given session, *k*-nearest neighbors are determined based on Euclidean distance

3. Recommendations:
   Based on number of times an item is clicked and bought in neighboring sessions, a weighted recommendation score is calculated for each item in every neighboring session And *n* items with the highest recommendation score are given as recommendations for a given active session

$$Reco\ Score = \frac{1\times n_1+19\times n_2}{20}$$

$n_1 = no.\ of\ times\ item\ clicked\ in\ the\ neighborhood\ sessions$

$n_2 = no.\ of\ times\ item\ bought\ in\ the\ neighborhood\ sessions$

Nearest Neighbors Determination:

Training data contains over 9.2 millions sessions. So, finding neighbors by brute-force approach is computationally very expensive. So approach used here is Locality Sensitive Hashing.

Locality-Sensitive Hashing:

Often we want only the most similar pairs or all pairs that are above some lower bound in similarity. If so, then we need to focus our attention only on pairs that are likely to be similar, without investigating every pair. There is a general theory of how to provide such focus, called locality-sensitive hashing (LSH) or near-neighbor search.

One general approach to LSH is to "hash" items several times, in such a way that similar items are more likely to be hashed to the same bucket than dissimilar items are.

We then consider any pair that hashed to the same bucket for any of the hashings to be a candidate pair. And neighbor search is done by considering only these candidate pairs.

Here, Euclidean distance is used for calculating smilary.

Only those sessions are considered for neighbors which have Euclidean distance less than 1.00.

Recommendations are given for only those sessions which had greater than or equal to 7 neighboring sessions

8. Results

Evaluation Measurement:

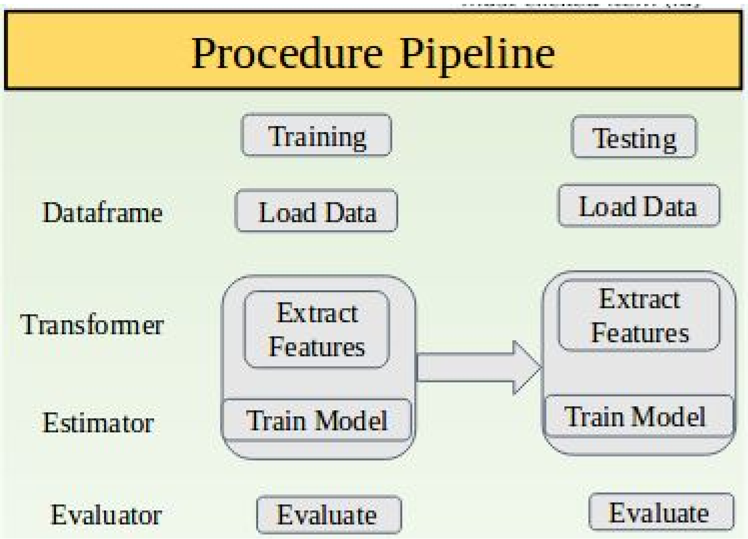


Fig.3: Procedure Pipeline Diagram

Results of Classification Model: Session Model

1. Logistic Regression:

   With 500 no of iteration :

   $Precision = 0.45$

   $Recall = 0.07$

   $F1\ score = 0.1211$

2. Random Forest

   With 100 no of trees in random forest,

   feature subset strategy = 0.5

   $Precision = 0.2387$

   $Recall = 0.2983$

   $F1\ score = 0.2652$

3. Gradient Boosted Trees

   $Precision = 0.3517$

   $Recall = 0.101$

   $F1\ score = 0.1530$

4. Neural Networks

   $Precision = 0.44$

   $Recall = 0.074$

   $F1\ score = 0.1267$

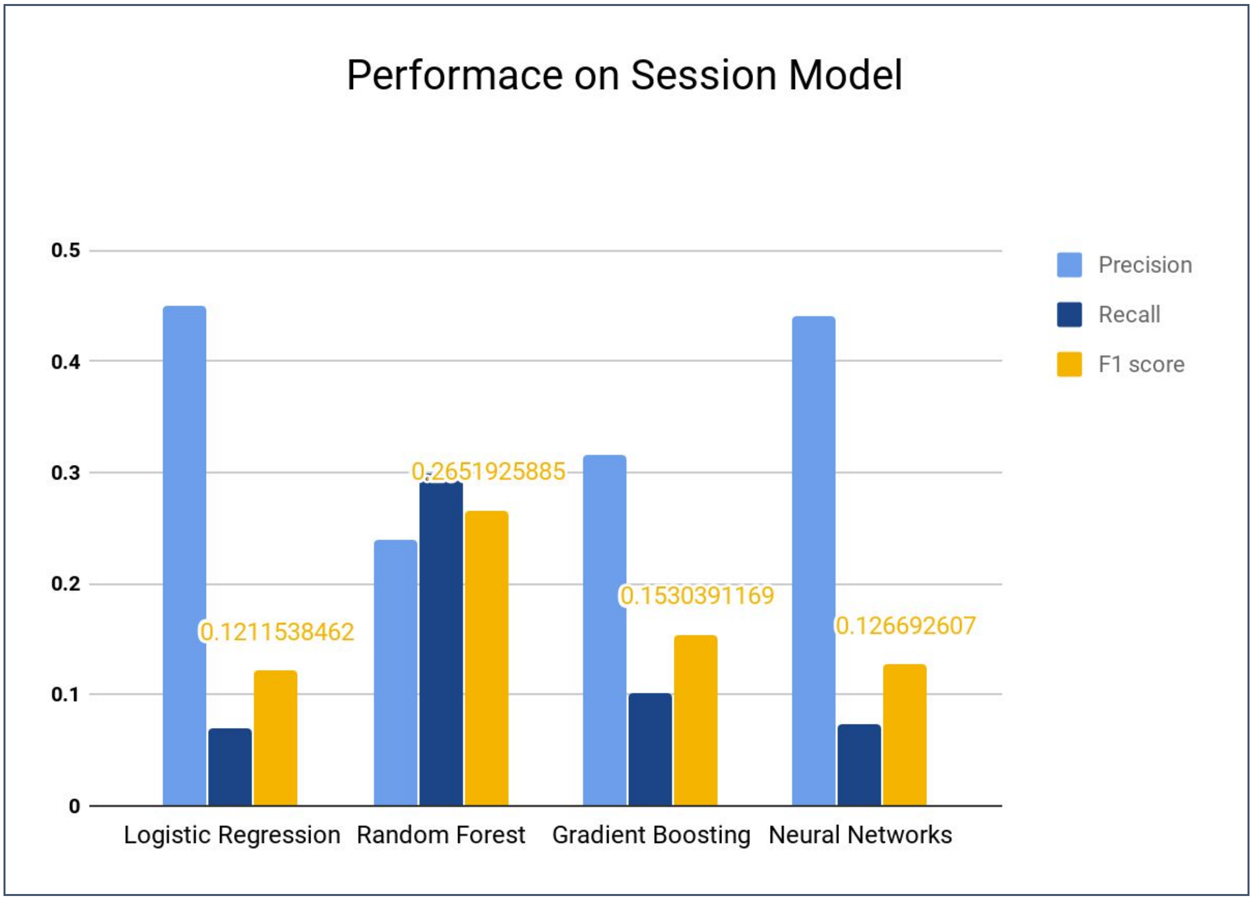


Fig. 4 Session Model Performance

Best Results were given by Random Forest. After proper parameter tuning, teh F1 score for session model was improved to 0.3023.

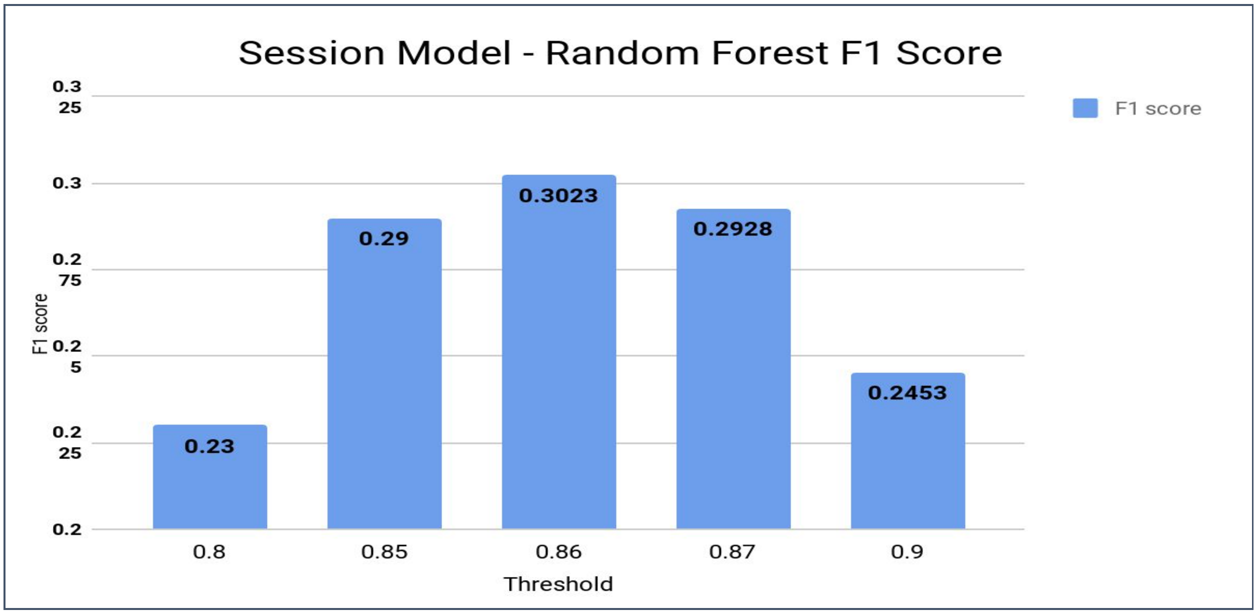


Fig. 5 Random Forest-Session Model Performance

Results of classification Model: Item Model

Best Results for Item classification were also derived from Random Forest.

Parameters:

- No of Trees : 400
- Probability Classification Threshold : 0.89
- Feature Subset Strategy : 0.8

Best F1 score achieved : 0.2603

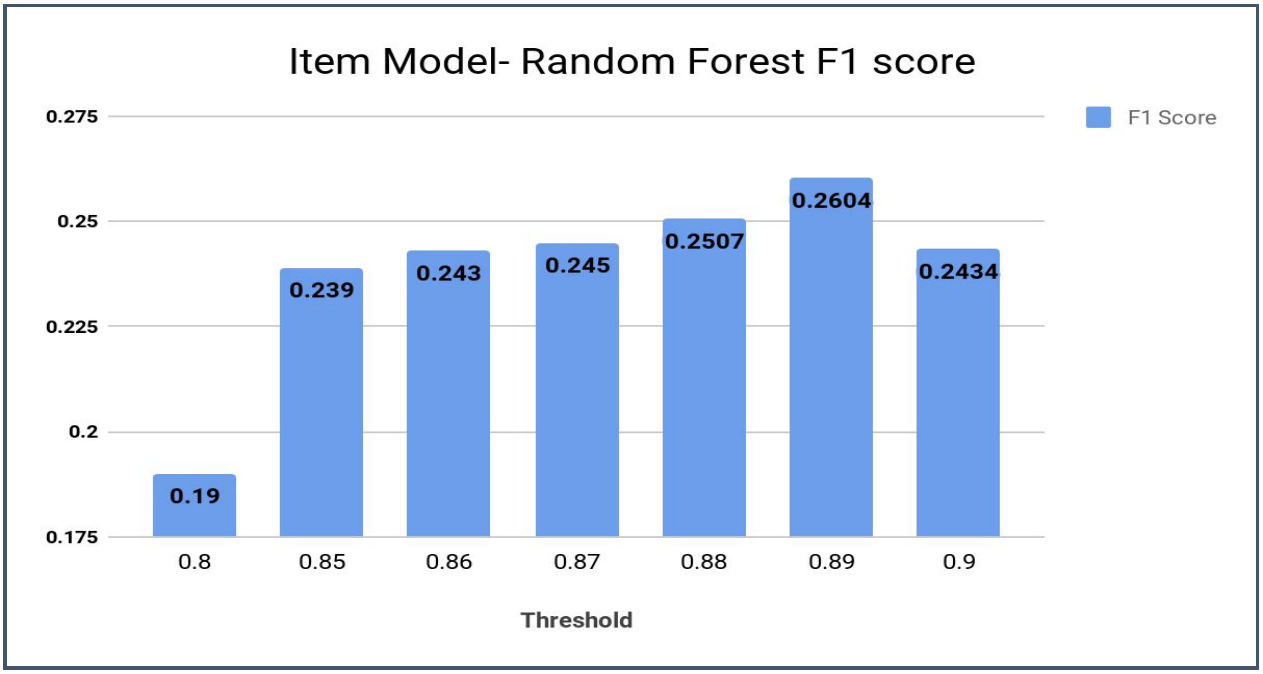


Fig. 6 Random Forest-Item Model Performance

Primary Recommendation System Model Results:

Recommended items were derived based on strategy discussed earlier. A maximum of 10 items are recommended for every session in a validation dataset.

LSH Parameter tuning:

- Bucket Length: 0.8
- Similarity Threshold: 0.1

Results:

%Hit_Rate@10 = 7.31

References

Volkovs, M. , 2015 , *Two-Stage Approach to Item Recommendation from User Sessions*.

Bamshad, M., Dai, H., Luo T. and Nakagawa, M. , 2001, *Improving the Effectiveness of Collaborative Filtering on Anonymous Web Usage Data.*

Leskovec, J., Rajaram, A. and Ullman, J.D. (2014). *Mining of Massive Dataset.* Palo Alto, California: Cambridge University Press.

Romov, P., Sokolov, E., *RecSys Challenge 2015: ensemble learning with categorical features* [PDF File]
Retrieved from:
https://github.com/romovpa/ydf-recsys2015-challenge/blob/master/slides/RecSysPresentation.pdf

List of Figures

List of Tables