AirBnB challenge

Where will a new guest book its first destination ?

A Network Tour of Data Science
EE-558

Grisard Malo
Pecoraro Cyril
Jaume Guillaume

Motivations

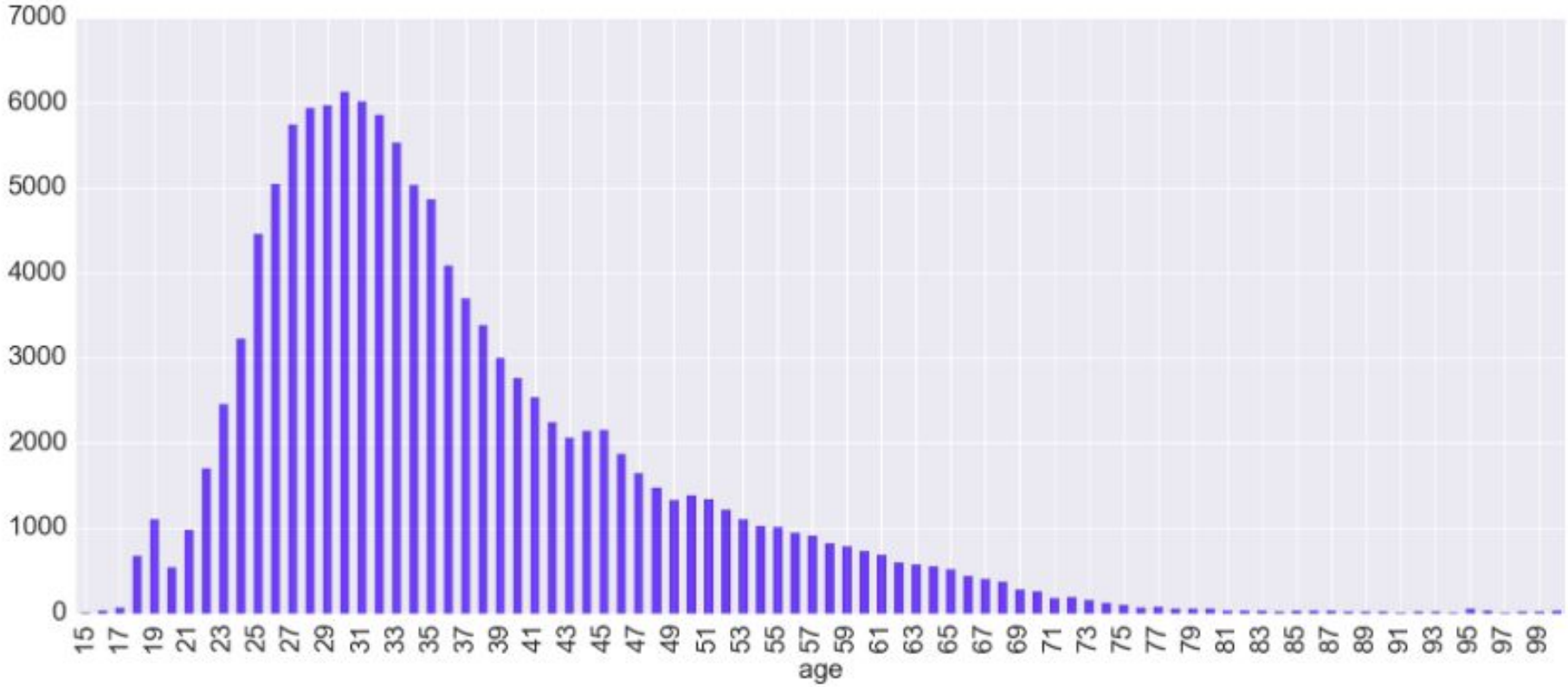
- **Goal : Predict the country of a new user's booking**
  - Give 5 most likely countries
  - Can include "not booking"

- Graded using Normalized Discounted Cumulative Gain

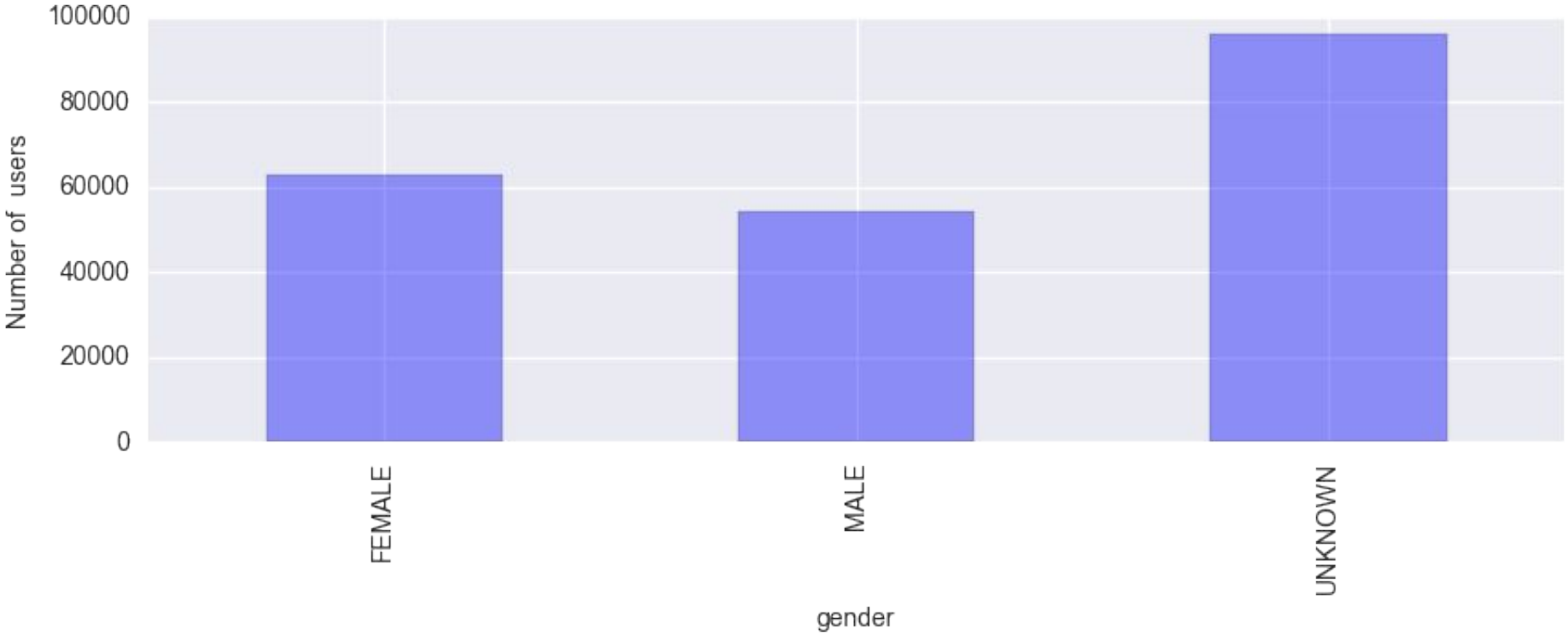
Dataset

- Real data sample from Airbnb from 2010 to 2014
- User information
  - Train set : data includes destination of the user
  - Test set : data does not include destination
- Browser sessions information per user
- Demographic information about countries

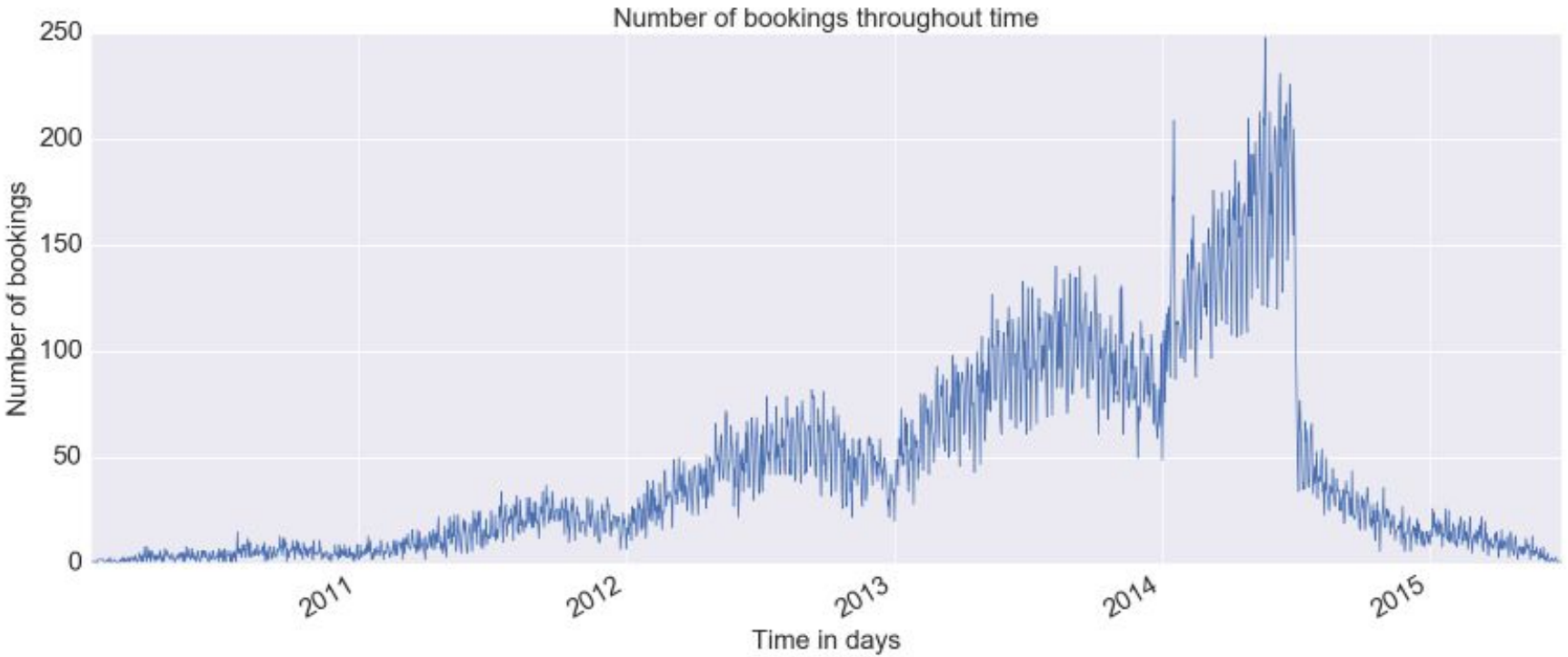
Age distribution



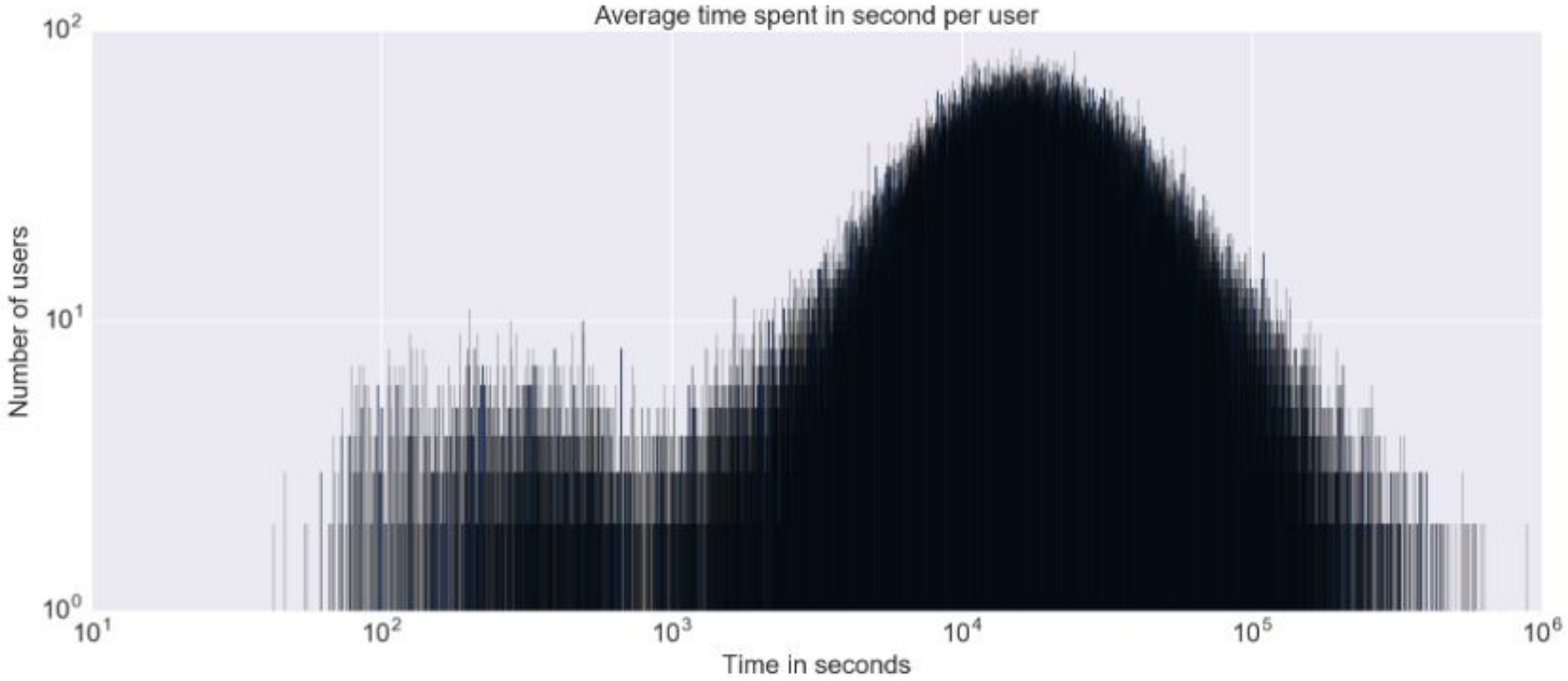
Gender


100000
80000
60000
40000
20000
0
Number of users
FEMALE
MALE
UNKNOWN
gender

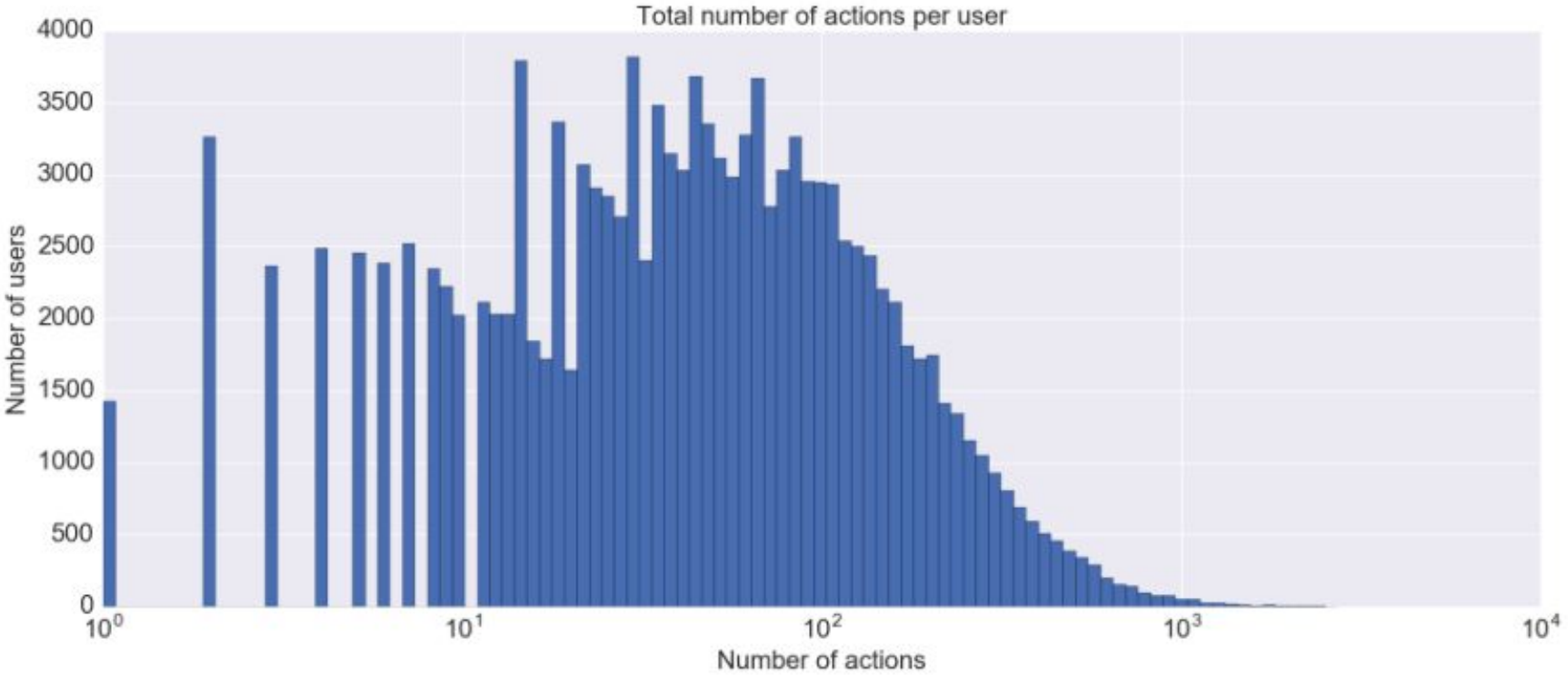
Bookings throughout time



Average time spent per user



Total number of actions per user



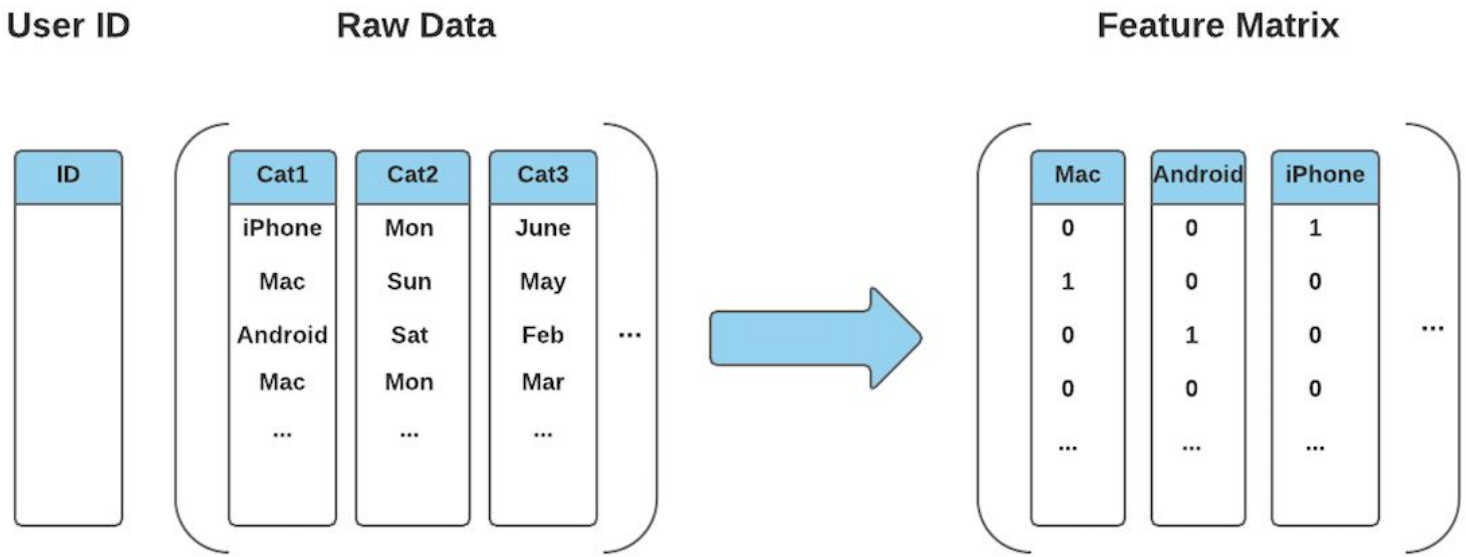
From raw data to feature matrix (1)

- Replace all missing data with -1 (age, sessions)

- Created new features from the browser session
  - Number of actions
  - Average time per session
  - Total time per user

- Created new features from user information
  - Date account created
  - Date account first active

From raw data to feature matrix (2)



- 165 features in total

Evaluation Metric

NDCG score: Normalized Discounted Cumulative Gain

$$DCG_k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2 (i + 1)}$$

$$nDCG_k = \frac{DCG_k}{IDCG_k}$$

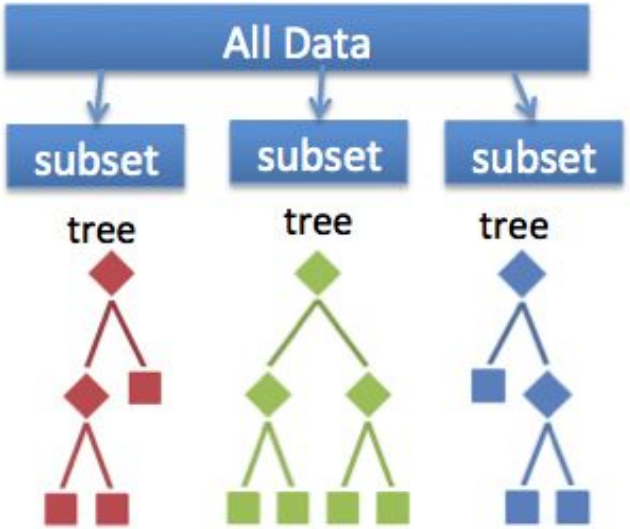
Example : [US,FR] $DCG = \frac{2^0-1}{log_2(1+1)} + \frac{2^1-1}{log_2(2+1)} = \frac{1}{1.58496} = 0.6309$

Machine Learning Approach

4 models investigated:

- **Random Forest**
- **Extreme gradient boosting**
- **2 Layers Stacking**
- **Voting**: combination of the above

Model 1 : Random forest



Best parameters found with **Cross Validation** :

- Depth : 16
- Trees : 600

Results :

- Kaggle NDCG = **0.86686**
- Ranking 1000th

Model 2 : Extreme gradient boosting

Combines **weak "learners"** into a single **strong learner**, in an iterative fashion. The goal is to learn a model F to predict values in the form y = F(x).

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma)$$

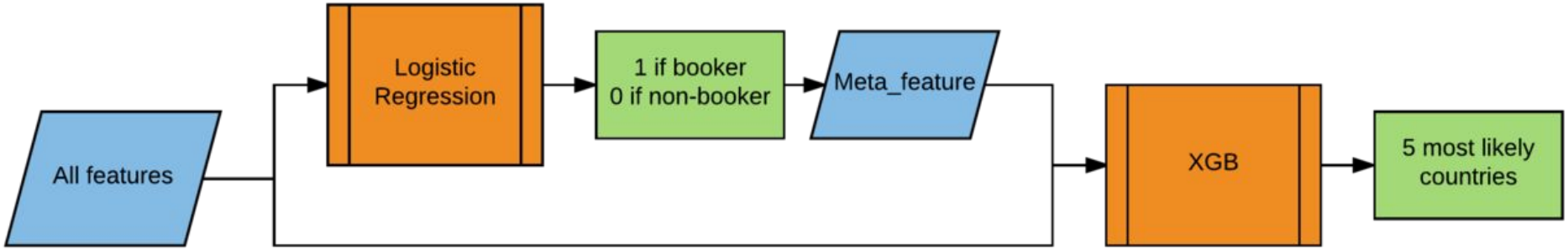
$$F_{m+1}(x) = F_m(x) + h(x) = y$$

Best parameters found with **Cross Validation** :

- Depth : 7
- learning rate : 0.1
- gamma : 0.7
- estimators : 75
- Loss Function : multisoftProb

Model 3 : 2 Layers Stacking

- **Main idea :** counter the unbalanced classes
- Non bookers (50%) / bookers (50%) -> balanced



Voting Model

Goal: Balance individual weaknesses

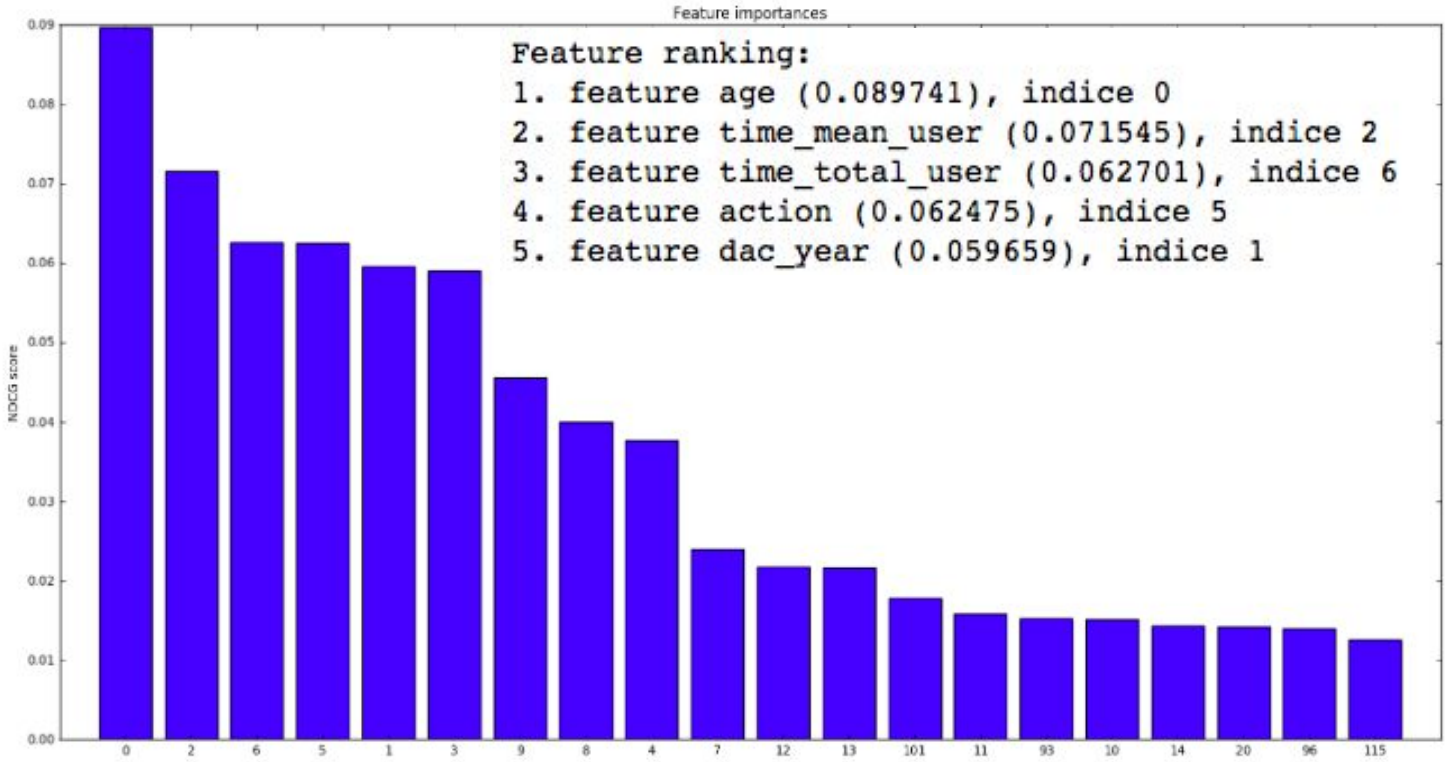
Combine models:

- Random forest
- Xgboost
- 2-Layers stacking

classifier	class 1	class 2	class 3
classifier 1	w1 * 0.2	w1 * 0.5	w1 * 0.3
classifier 2	w2 * 0.6	w2 * 0.3	w2 * 0.1
classifier 3	w3 * 0.3	w3 * 0.4	w3 * 0.3
weighted average	0.37	0.4	0.23

For each user.

Features importance



Conclusion

- Best model : Simple XGBoost, **Kaggle NDCG score of 0.86967**.
- Stacking model is not better than XGBoost.

Why ? First layer logistic regression F1 scores just above 60%.

- Voting model is not better than the XGBoost

Why ? Probably because it is composed of models performing less well, not bringing information

- Future investigations : improve stacking model

695	↓158	Matt	0.86987	7	Thu, 11 Feb 2016 21:05:58	
696	↓72	Shi Fan ‡	0.86987	7	Mon, 18 Jan 2016 05:26:28 (-28.4h)	
697	↓118	Julien	0.86987	6	Fri, 15 Jan 2016 10:44:54 (-0.5h)	
698	↑87	SingleModel	0.86987	12	Sat, 30 Jan 2016 23:38:58	
-		**Malo Grisard**	**0.86987**	-	**Wed, 18 Jan 2017 15:31:50**	**Post-Deadline**

Post-Deadline Entry
If you would have submitted this entry during the competition, you would have been around here on the leaderboard.

699	↑70	MilapShah	0.86986	3	Thu, 24 Dec 2015 12:26:02
700	↓81	Kuber@IITB	0.86986	6	Tue, 15 Dec 2015 05:32:58 (-3.7d)
701	↑20	Swaroop Kallakuri	0.86986	2	Tue, 15 Dec 2015 18:02:54
702	↑169	Hu Qin	0.86986	14	Mon, 21 Dec 2015 16:10:59 (-5.1d)
703	↑169	Harshal	0.86986	1	Thu, 17 Dec 2015 09:00:17

Model 2 : Extreme gradient boosting

- “multi:softprob”

  –same as softmax, but outputs a vector of ndata * nclass, which can be further reshaped to ndata, nclass matrix. The result contains predicted probability of each data point belonging to each class.

- gamma:

  minimum loss reduction required to make a further partition on a leaf node of the tree. The larger, the more conservative the algorithm will be.

- max_depth :
  maximum depth of a tree, increase this value will make the model more complex / likely to be overfitting.