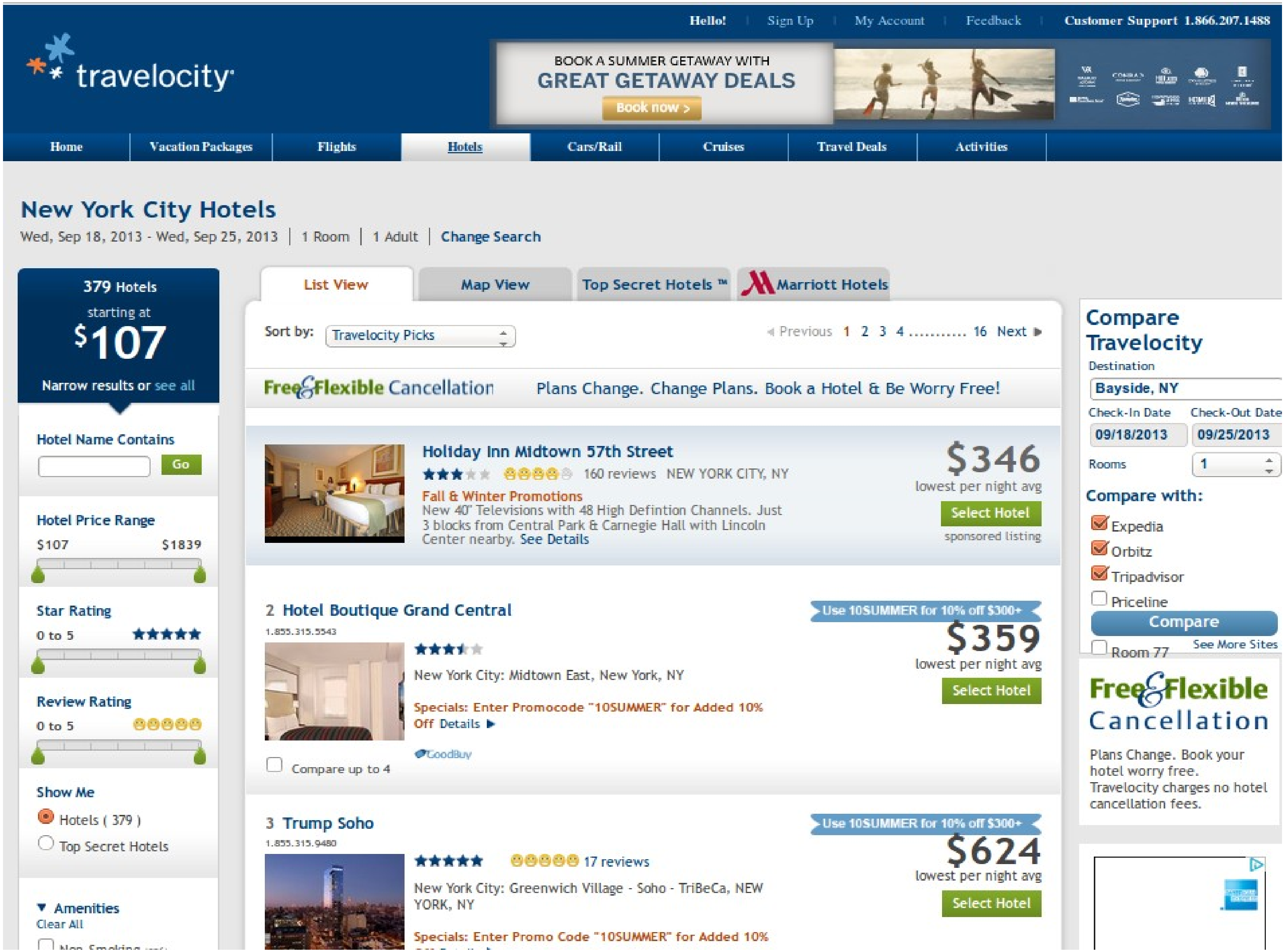
Improving Hotel Ad Performance

General Assembly – Data Science sec. 4
Marc Kirchner – August 2013

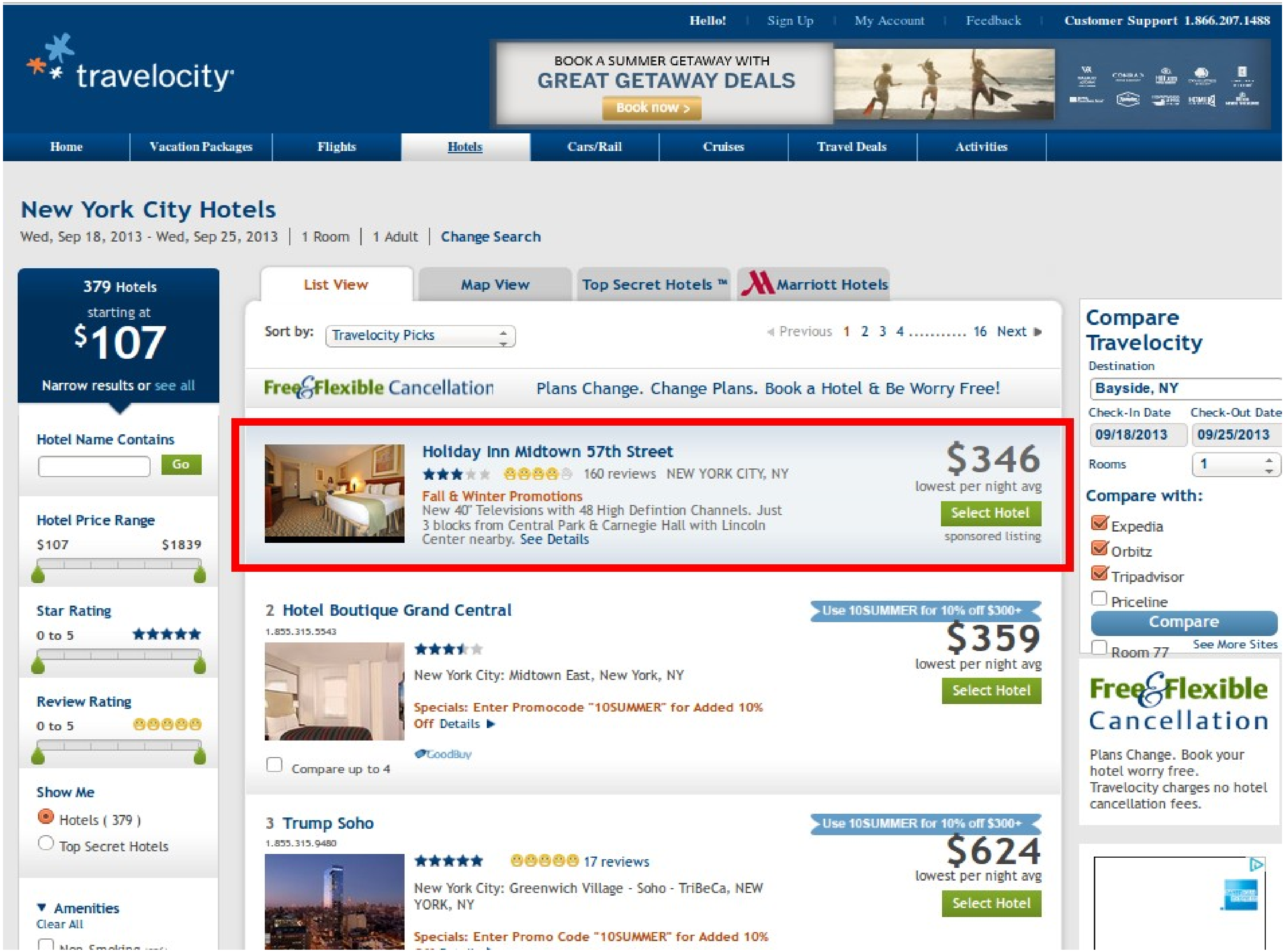
Business Context

- In the company I work we provide an ad-serving service for different sites
- We are focused on the traveling industry
- One of our products is to allow hotel advertisers bid for showing on a top position in a search results page of an OTA site
- Example: travelocity.com

Travelocity sponsored ads



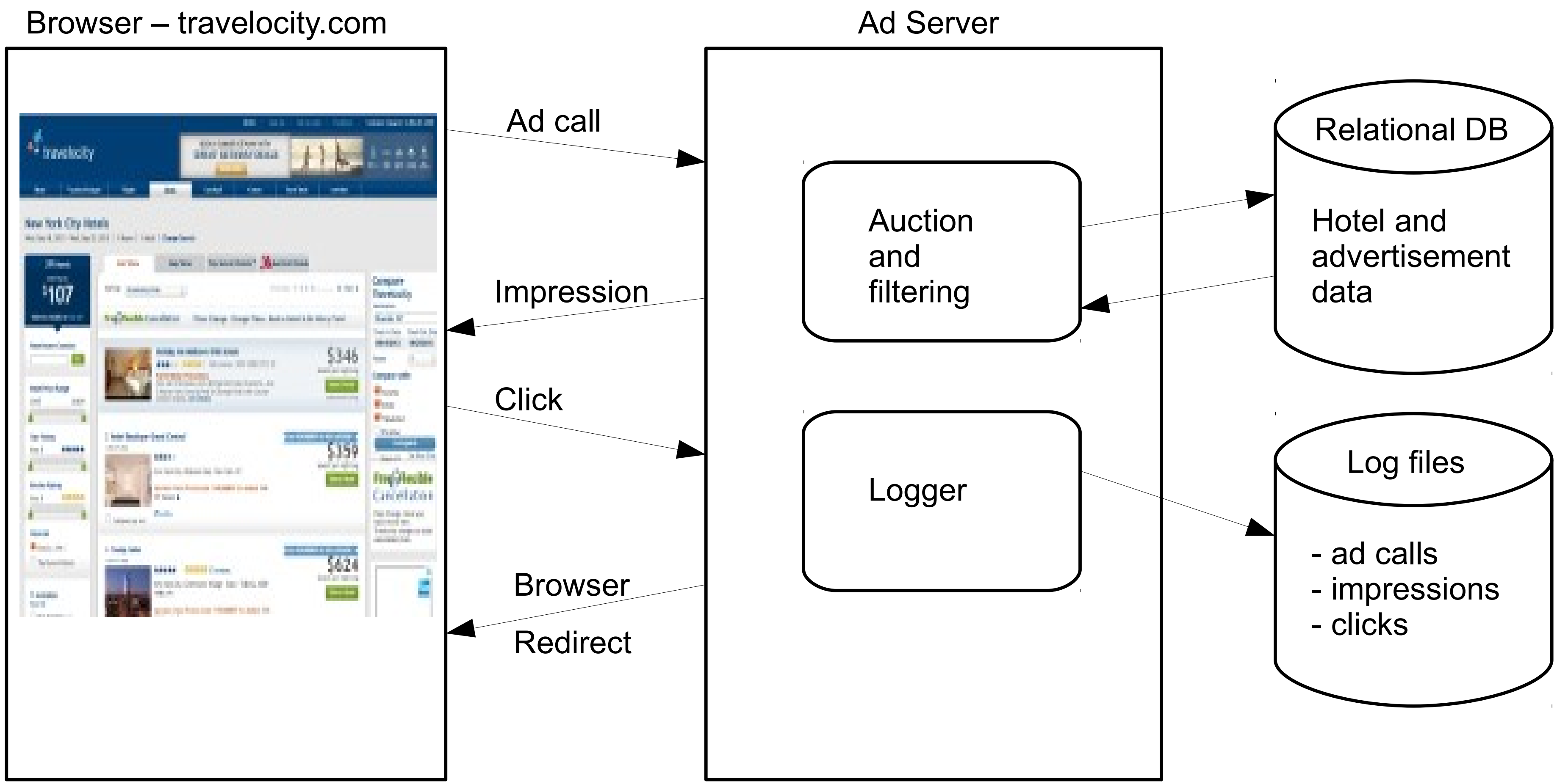
Travelocity sponsored ads



Business Model

- Hotel advertisers decide how much are they willing to pay (bid) for placing an ad
- They can target specific destinations
- When a user performs a search for a hotel, all the advertisers compete for the sponsored placement in an auction
- If a user clicks on an sponsored ad, the advertiser pays an amount to the site.

The system


Browser – travelocity.com
Ad Server
Ad call
Impression
Click
Browser
Redirect
Auction
and
filtering
Logger
Relational DB
Hotel and
advertisement
data
Log files
- ad calls
- impressions
- clicks

Objective

- Try to maximize the performance of a hotel ad by showing to the user a hotel relevant to the intent
- Types of trips: Leisure vs Business
- If we can guess which hotel fits better to the user, we can give priority to that hotel in the auction

How?

- We extract ad calls, impressions and clicks data from the logs and join it
- We choose specific features from that data
- Pairs of user features + hotel features combinations
- We train a classifier model using the features and the labels: click / non-click

Features

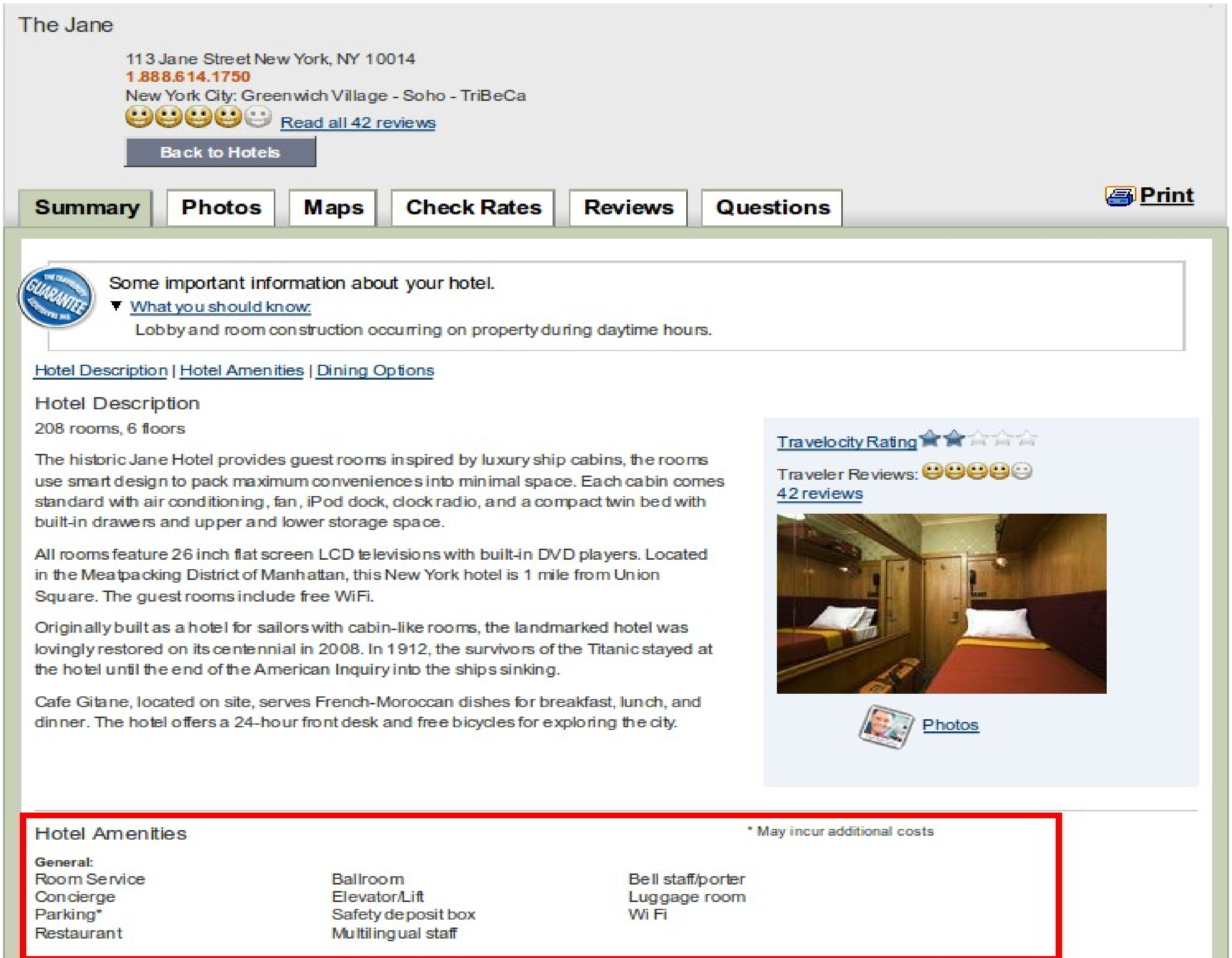
User - Intent
Number of travelers
Number of rooms
Is weekday?
Advance purchase range
OS (Windows, Mac)
Browser (IE, FF, Ch)

Hotel - advertisement
Hotel star rating
Ad copy: promotion
Ad copy: discount
Ad copy: free

We need hotel features

- The hotel data in our system is pretty limited
- We decided to scrape travelocity.com hotel details pages
- We used a Vectorizer to extract hotel amenities
- For each training example, we added the hotel amenities as yes/no features
- We generated 190 extra features this way

Travelocity hotel details page



Data challenges

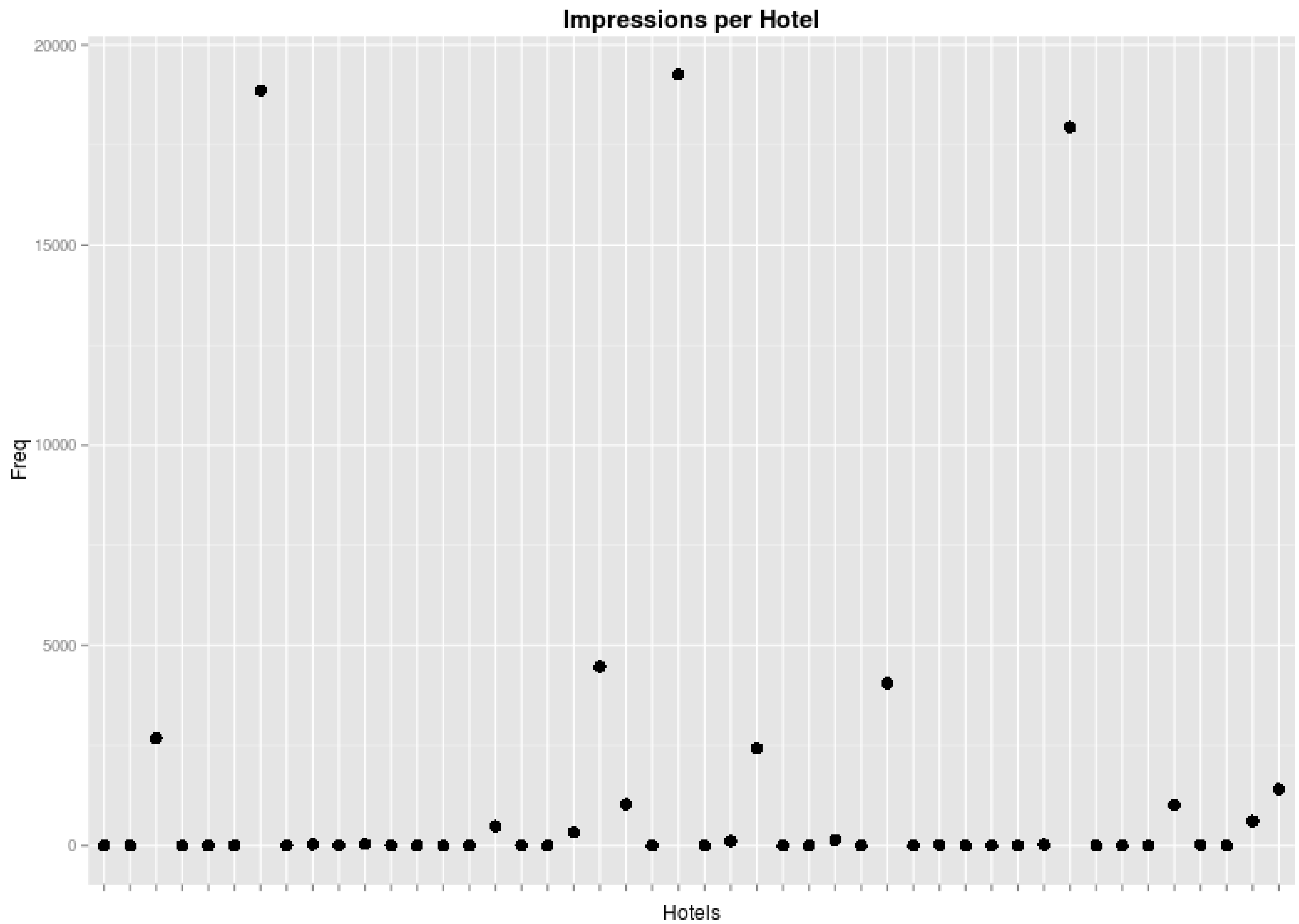
- Our systems contain lots of data, most of it is not relevant to our problem
- One single day of travelocity data is a few Gb of data
- We decided to limit the experiment to users searching for hotels in New York
- It reduced the data size to approx 25 Mb of relevant data

How does the data look?
(New York hotel destinations in 2013)

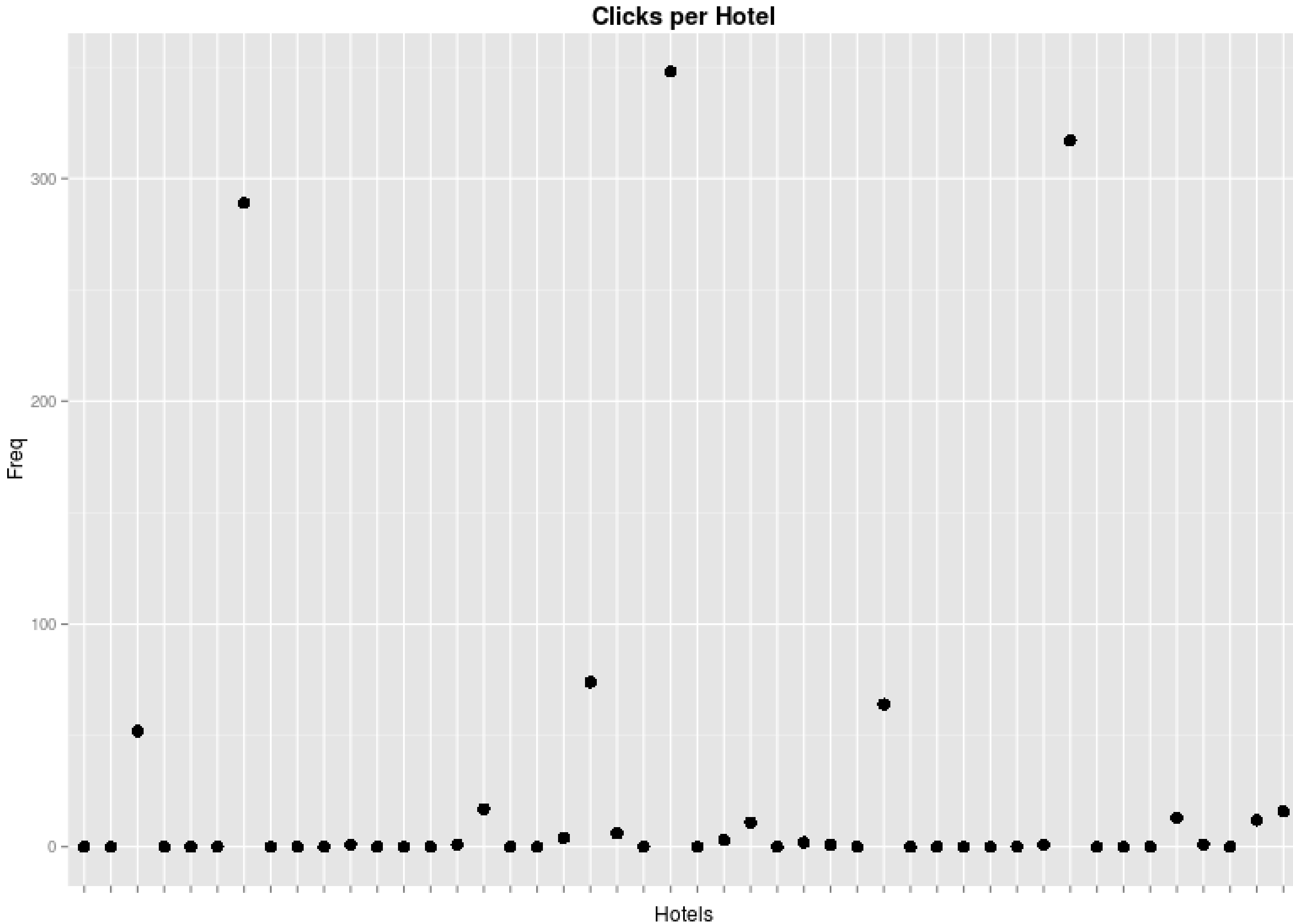
- Number of impressions: 73,800
- Number of clicks: 1,200
- There are only 1.67% clicks relative to the number of impressions (CTR)
- There are approx 50 different NYC hotels that have been shown in sponsored ads
- CPC summary:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.2500	0.2500	0.3700	0.4027	0.4800	1.5000

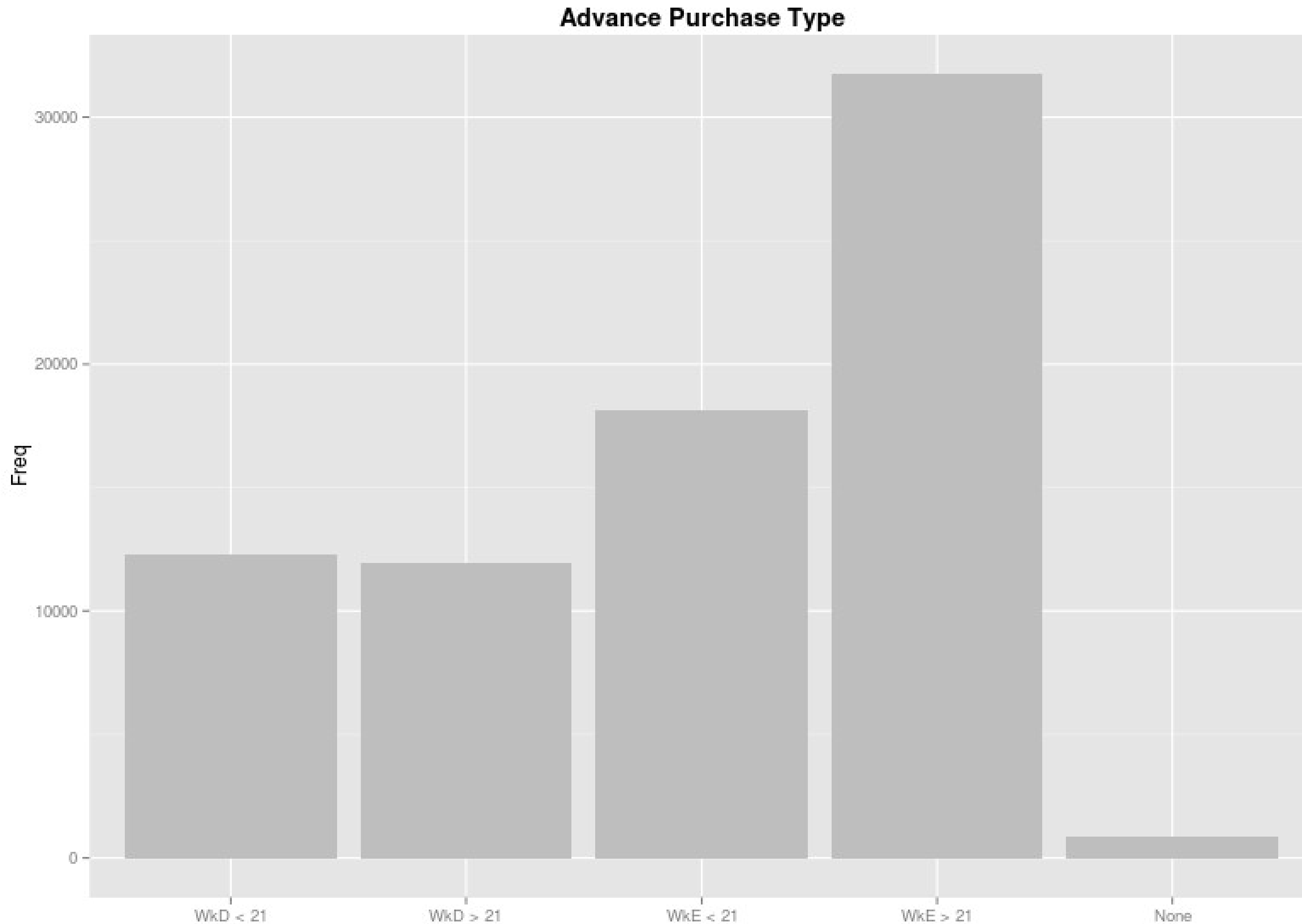
How does the data look?



How does the data look?



How does the data look?



Testing the model

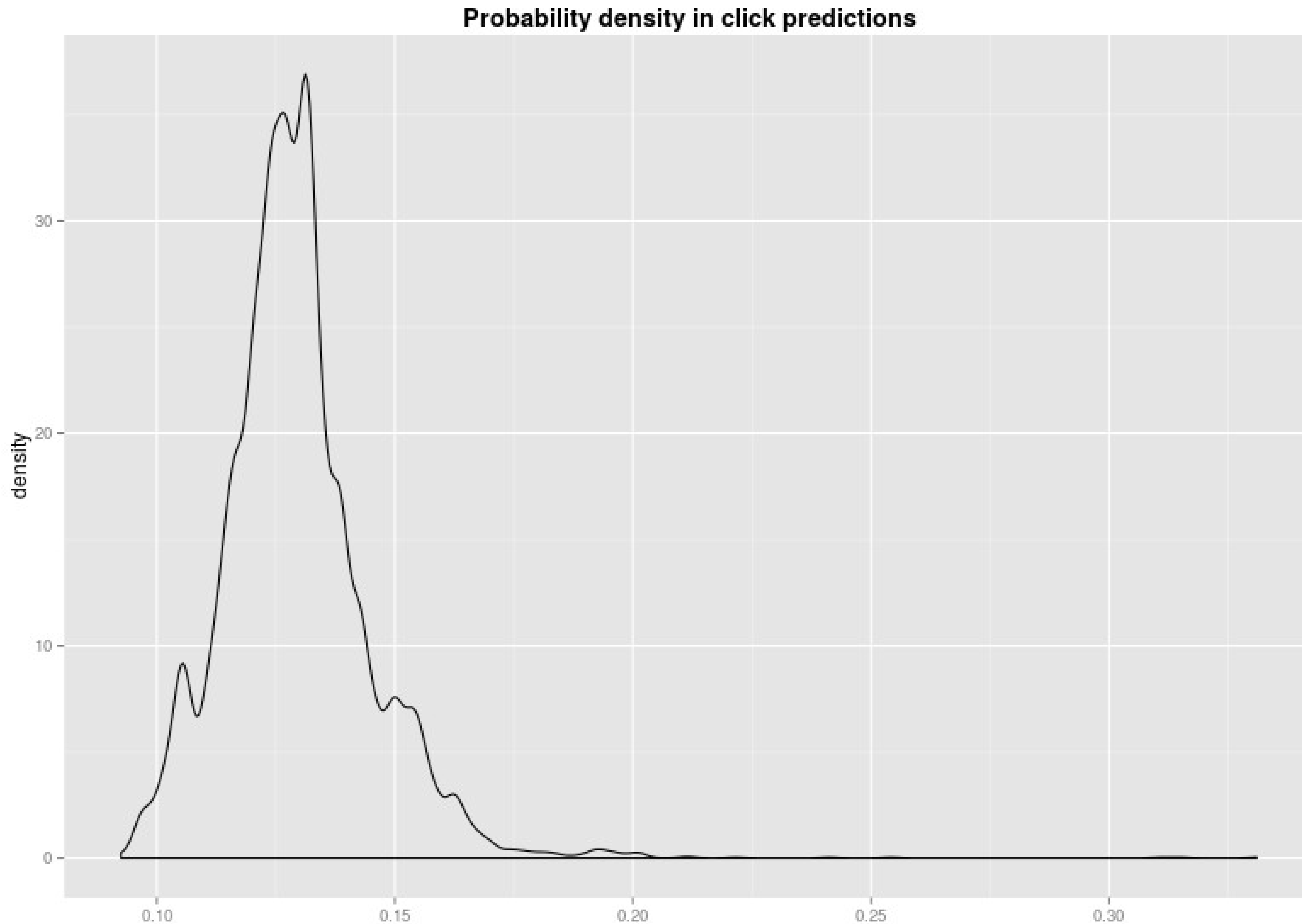
- The vast majority of outcomes in the testing set is negative (no click)
- Average AUC cross validation value: **0.58**
- Better than tossing a coin
- Is it enough to make a difference?

Behavior on out of sample data

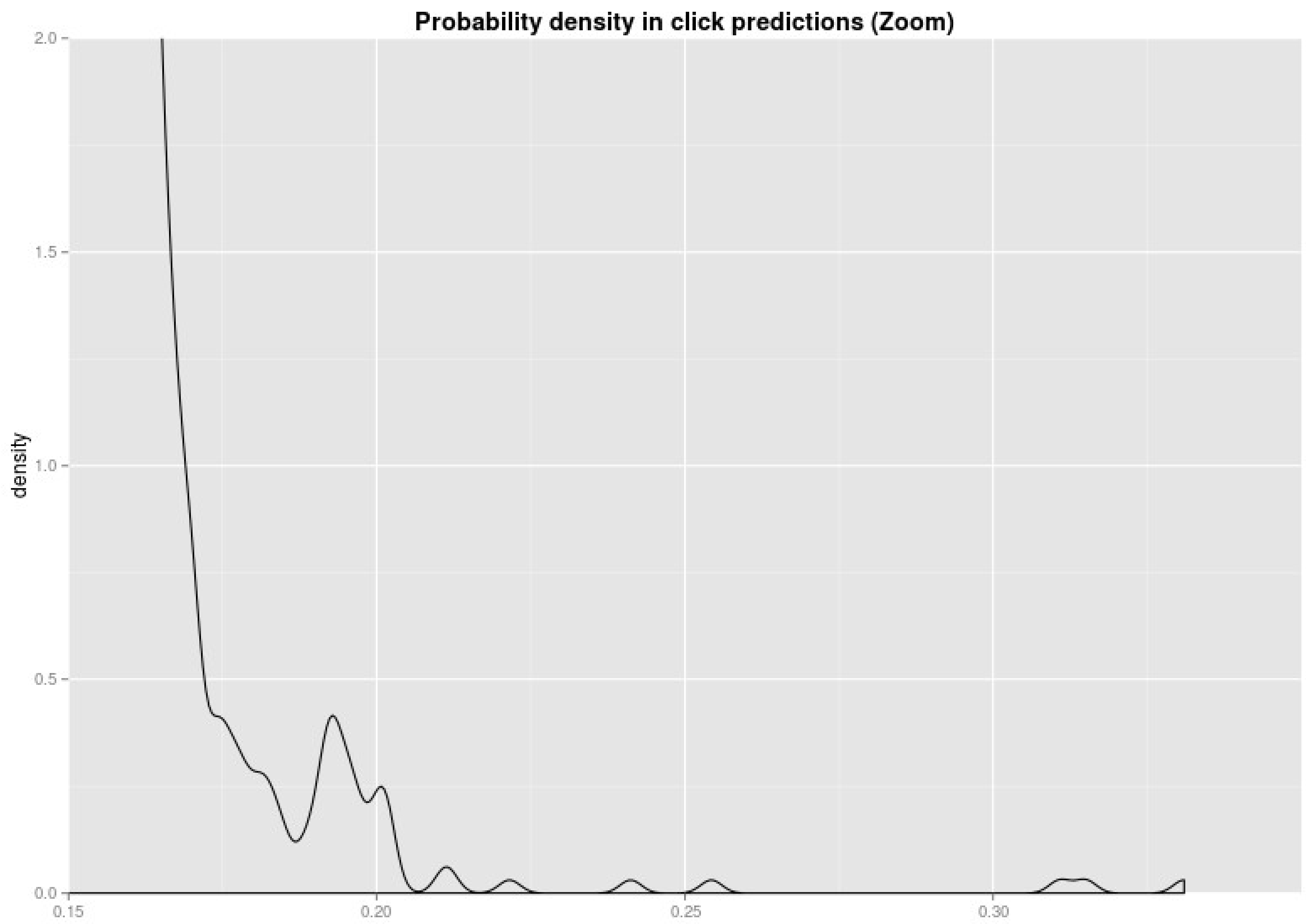
- Predicting the probability of a click on OOS data (7500 predictions)
- The probability is very low across examples
- Some predictions are far higher than others

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.09245	0.12010	0.12740	0.12930	0.13540	0.33100

Behavior on out of sample data



Behavior on out of sample data



Questioning the model

- We don't want to predict clicks in general. We want to predict the probability of a click **given that** we show a specific hotel
- Clicks are sparse, it's difficult to extract patterns in type of intent + hotel ad combinations
- How can we differentiate between users who would click on the ad *anyway*?

We can test the model in real life

- One way to test the impact is to release the model and use it for a share of the traffic
- We can then compare average CTR rates between the two groups
- We need a “clean” group anyway because when retraining the model in the future we don't want to introduce bias in the data (the model effectively influences the user behavior)

Possible improvements

- Do a better job identifying hotel types: Use a clustering algorithm
- Use pricing information
- Introduce features related to seasonality
- Explore the past behavior of users in a more fine-grained level (browsing patterns)
- More data!