It’s all about the rank!

(On estimating sales rank—an attempt)

By Con Healy, Felipe Perez, and Hari Ravindran

To rank or not to rank? That is the question—

Dataset: Product reviews, metadata, and related Q/A data from a category

Source: Amazon[1]

Data collection duration: 1996-2013

Initial idea: Engineer features out of available information to predict the Amazon sales ranking of a certain product within its category

[1] http://jmcauley.ucsd.edu/data/amazon/

Defining the problem.

Initial idea not viable… So, **improvise**!

Rephrased problem: Estimate the probability of a product falling within a certain range of sales ranks.

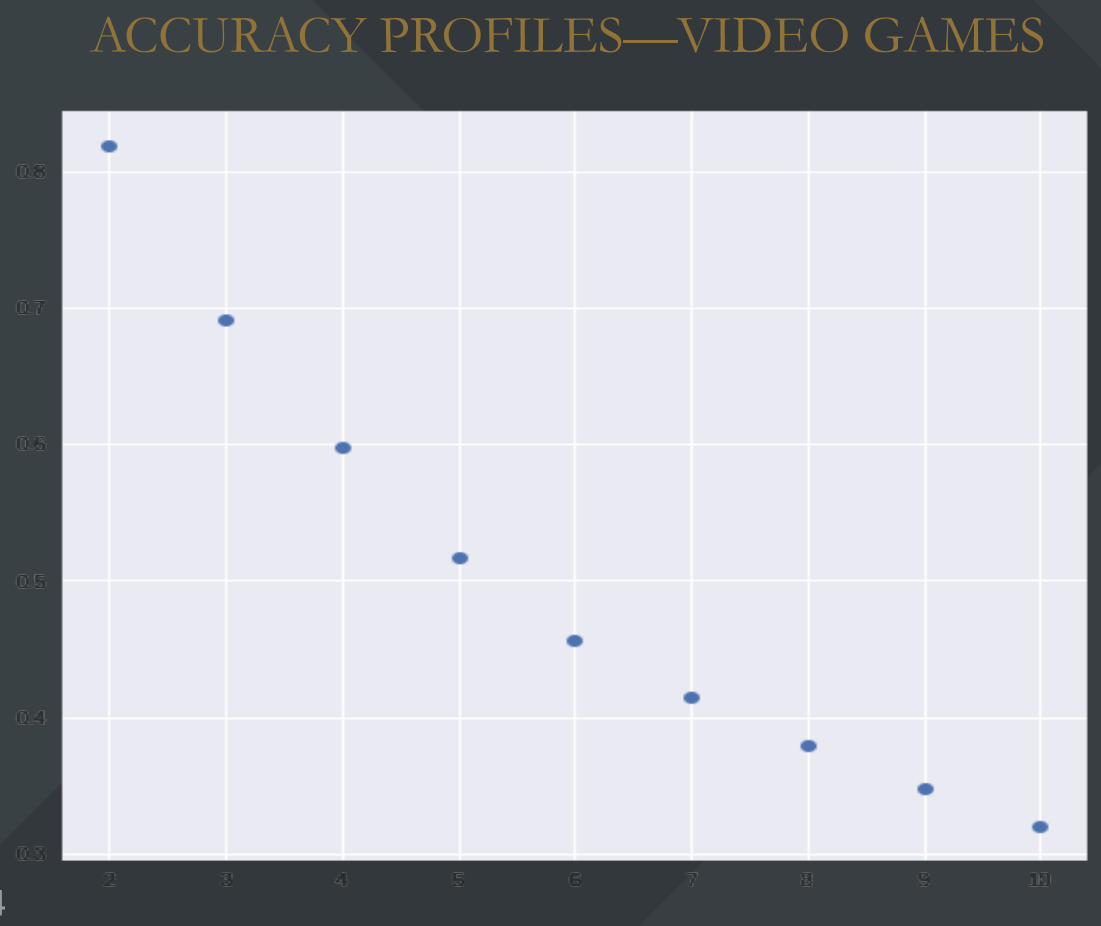
Our solution.

Used a combination of tools such as:

- XGBoost for classification
- FB Research's fastText/Google's word2vec
- Feature engineering

For the graph on the right:

- X-axis: Number of bins
- Y-axis: % accuracy of classification



Our solution.

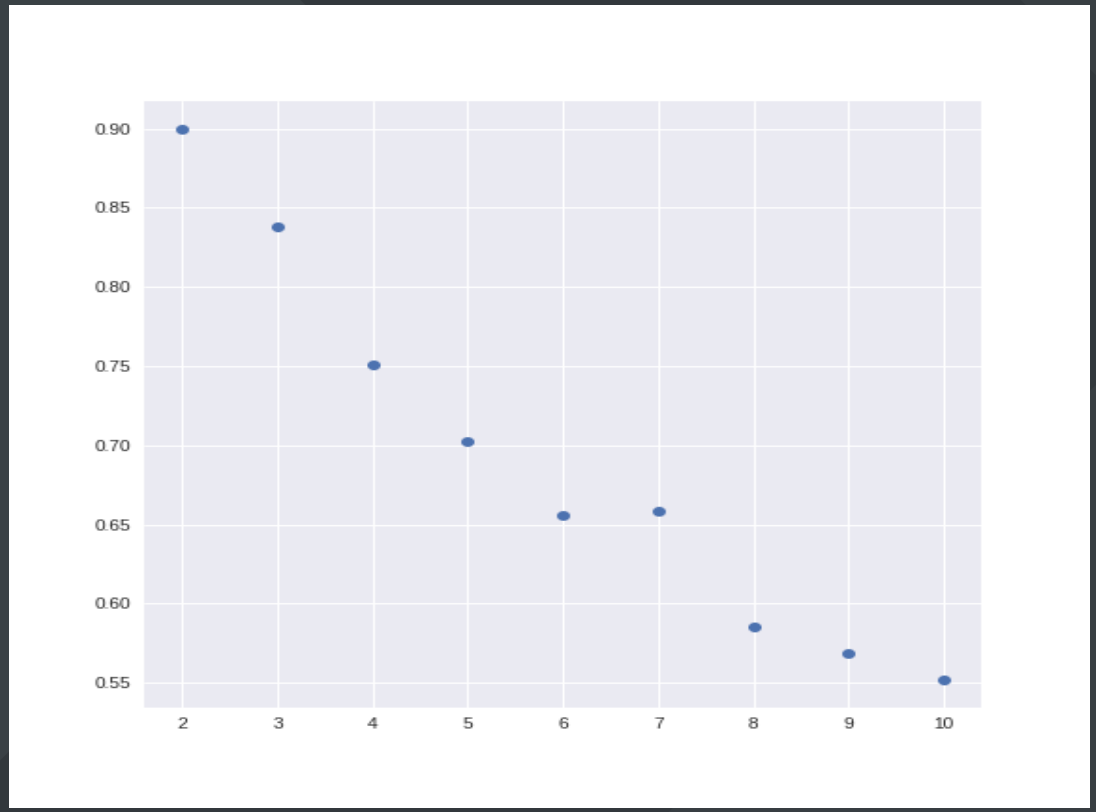
Used a combination of tools such as:

- XGBoost for classification
- FB Research's fastText/Google's word2vec
- Feature engineering

For the graph on the right:

- X-axis: Number of bins
- Y-axis: % accuracy of classification

ACCURACY PROFILES—VIDEO GAMES*



Why oh why? And who is it for?

Imagine a seller on Amazon's seller central.

How does one find profitable inventory for Amazon FBA sourcing?

ROI, legal restrictions, competition, and…

sales rank[2]

[2] http://www.fulltimefba.com/category/sales-rank/

Though this be madness, yet there is method in't.

Our main weapon: Feature engineering

Feature categorization

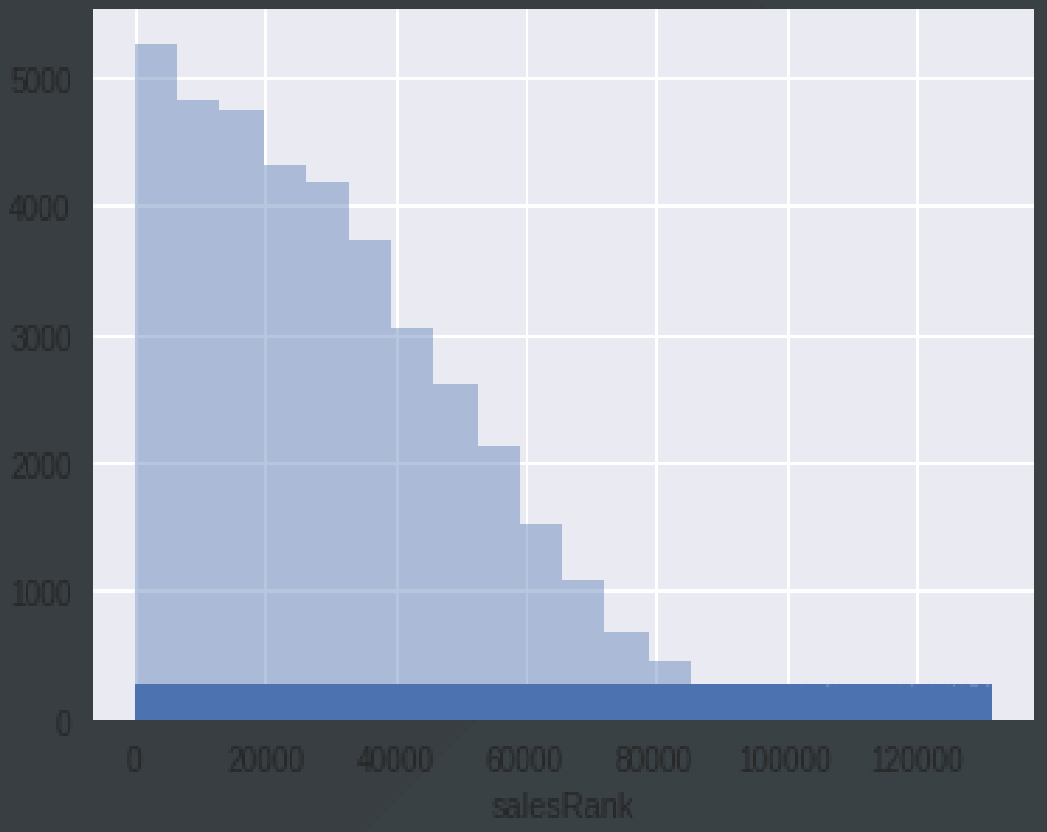
Product-based (Price, product 'hotness'/review density…)

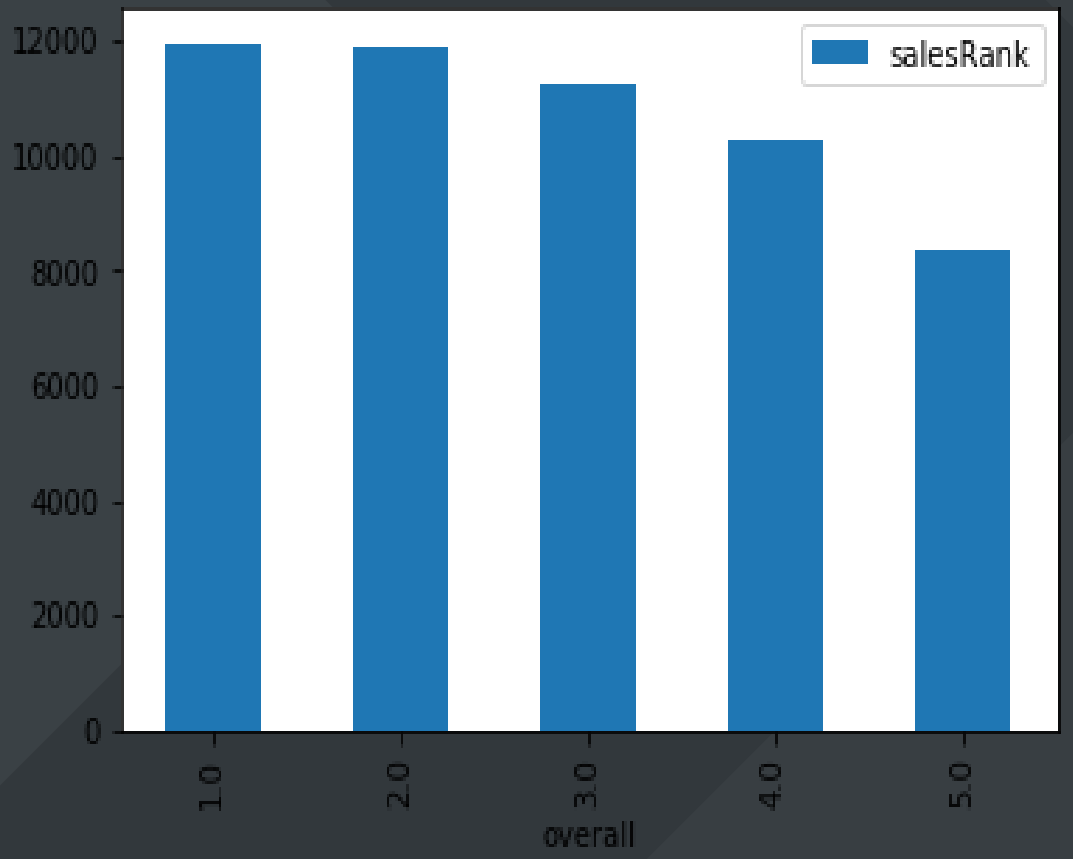
Reviewer-based (Product ratings, review helpfulness…)

Questions-based (Number of questions in the Q/A product forum)

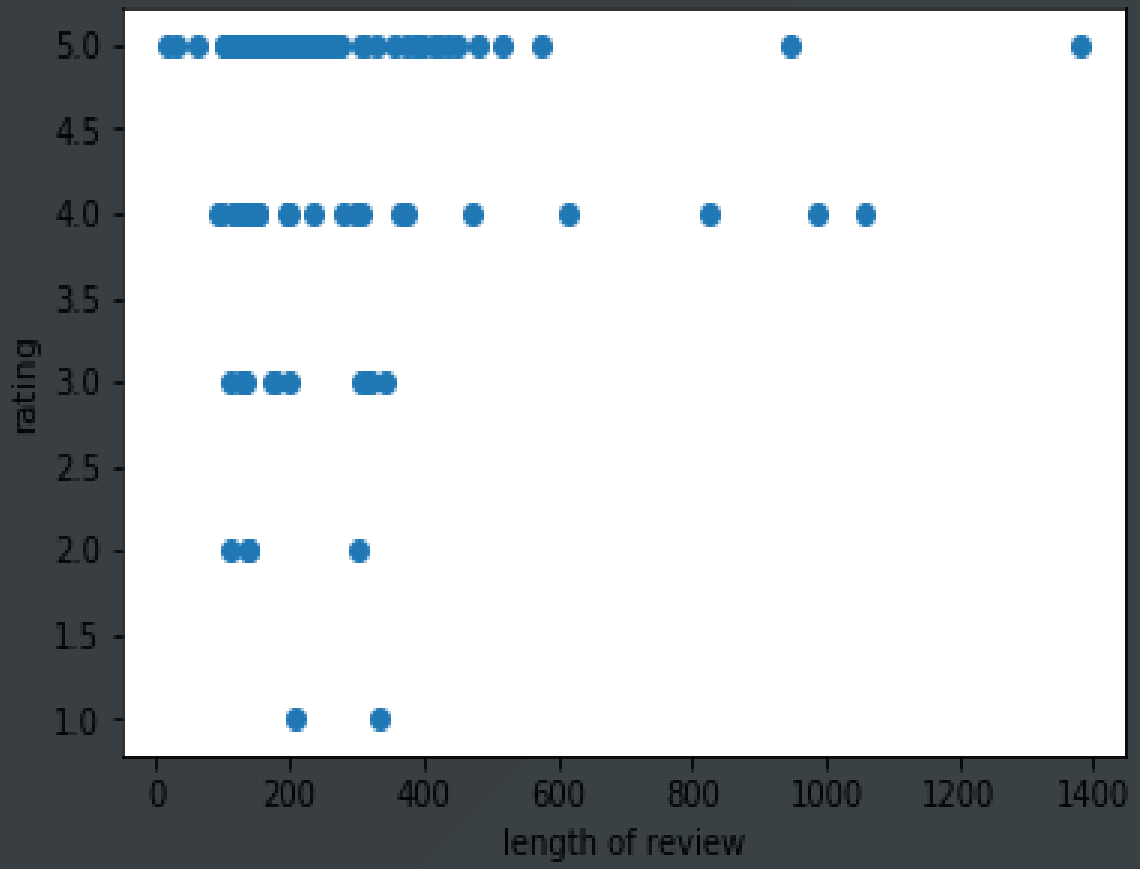
Review text-based (Sentiment score correlation with review ratings…)

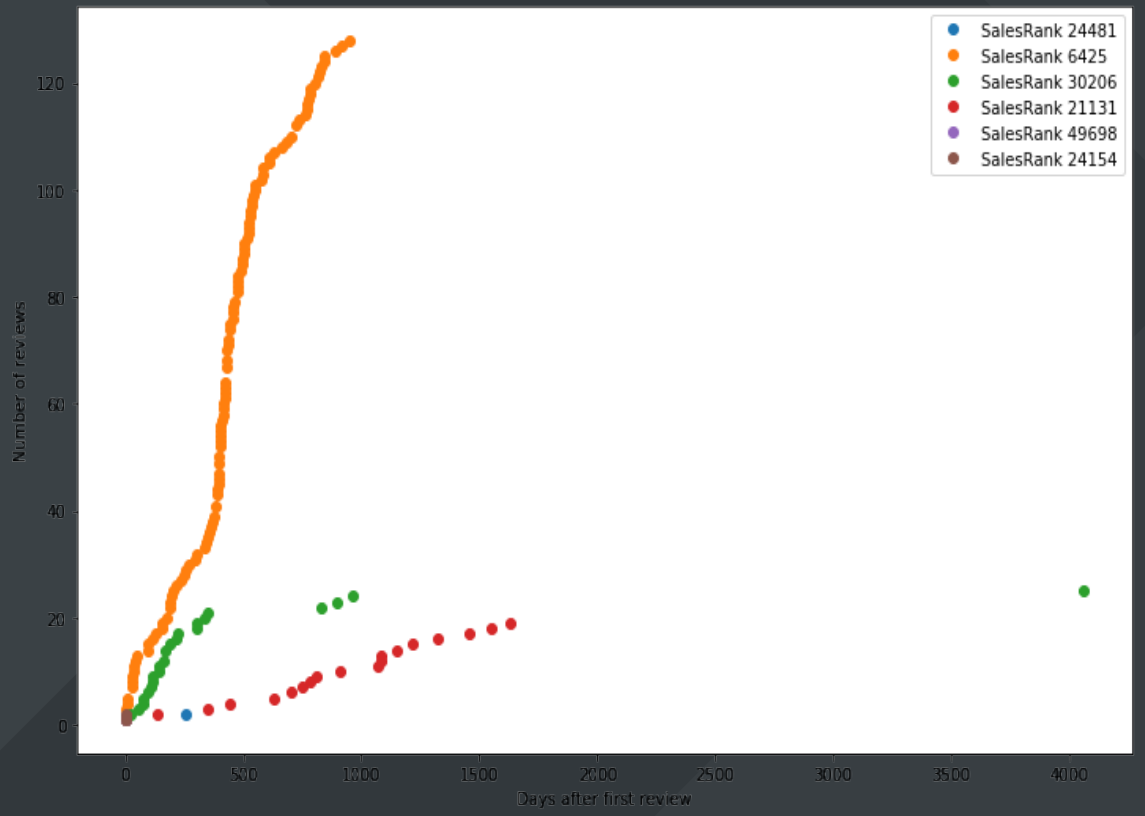
Data exploration—a sample





Data exploration—a sample





There's always room for improvement.

Product **segmentation** via 'Image2vec'

Dimensionality reduction to filter out 'noise'

Use of bona fide **ranking algorithms**

More **raw data manipulation** to deal with skewness

Thank you!