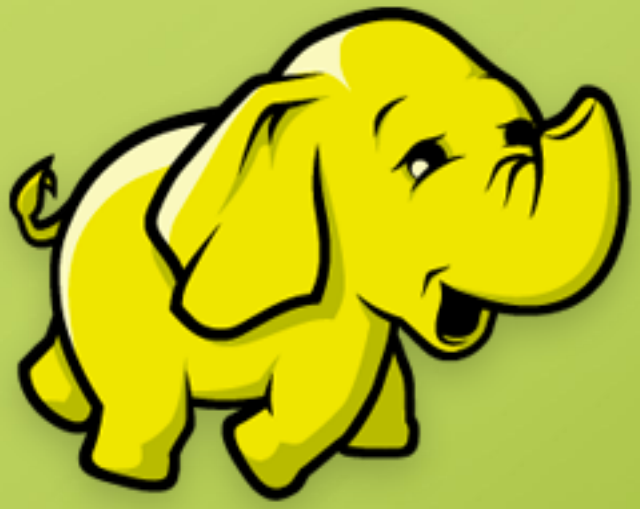
BestBuy QUERY

Machine Learning and Big Data

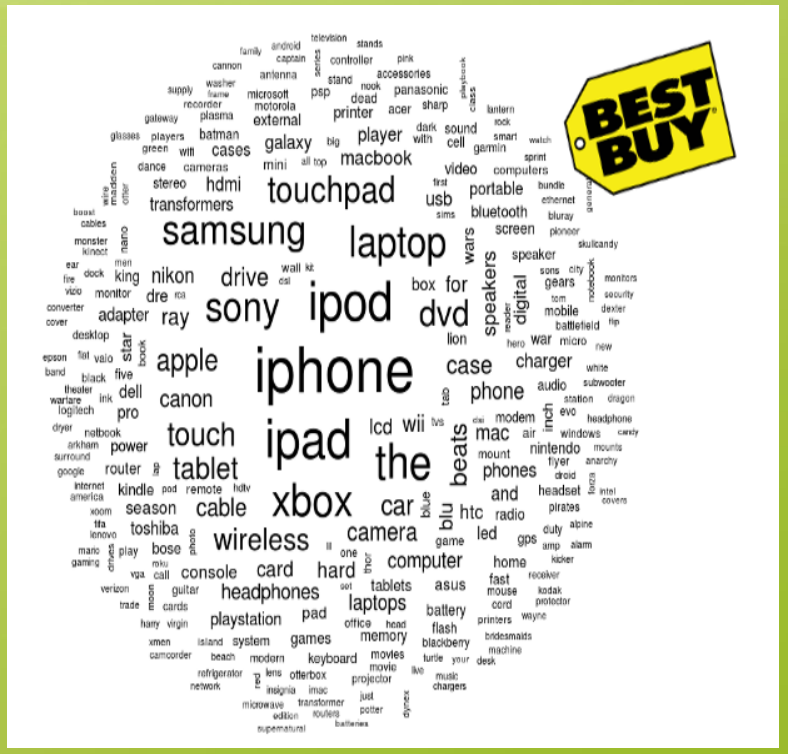


Greg, Lei, Lexie, Liang & Masha

Project Description

Our project is to complete the Best Buy Data Mining Hackathon on the 7GB data set.

Here is the link for the Kaggle Bestbuy Contest: http://www.kaggle.com/c/acm-sf-chapter-hackathon-big



Project Description

This is how data look like:

1	user	sku	category	query	click_time	query_time
2	0001cd0d10bbc585c9ba287c963e00873	2032076	abcat0701002	gears of war	22:56.1	21:42.9
3	00033dbced6acd3626c4b56ff5c55b8d6	9854804	abcat0701002	Gears of war	35:42.2	35:33.2
4	00033dbced6acd3626c4b56ff5c55b8d6	2670133	abcat0701002	Gears of war	36:08.7	35:33.2
5	00033dbced6acd3626c4b56ff5c55b8d6	9984142	abcat0701002	Assassin creed	37:23.7	37:00.0
6	0007756f015345450f7be1df3369542146	2541184	abcat0701002	dead island	15:34.3	15:26.2

Our goal is to predict the sku#.

Outline

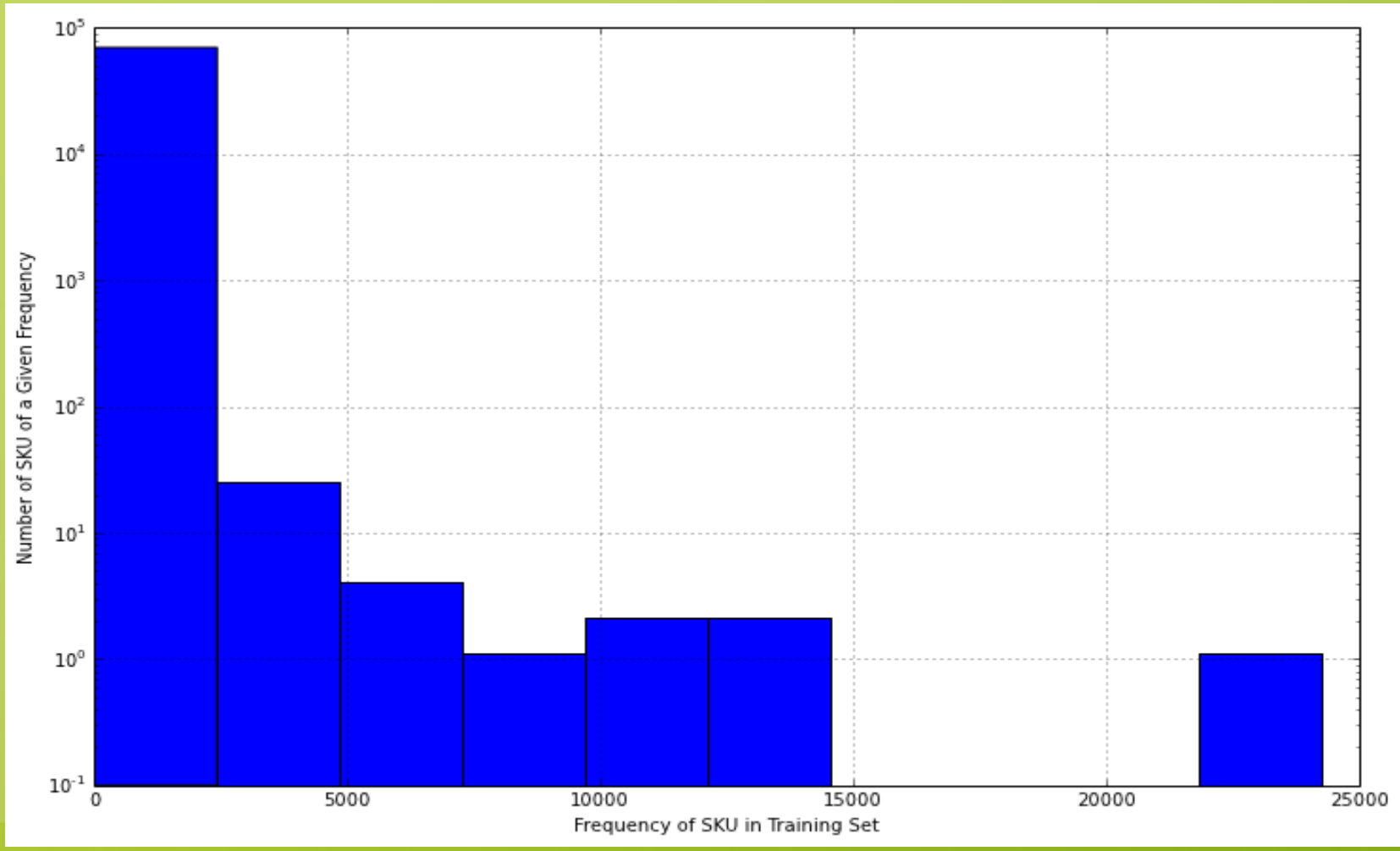
- Data Description
- Machine Learning Techniques
  - Data Cleansing
  - Feature Selection
  - Naïve Bayes Classification
  - Solr Search
- Handle Big Data
  - Hadoop, MongoDB, Solr
- Future Expectation

Data Description

- Rows: 1865269
- Unique Users: 1268702
- Unique SKU: 69858
- Unique Categories: 1540

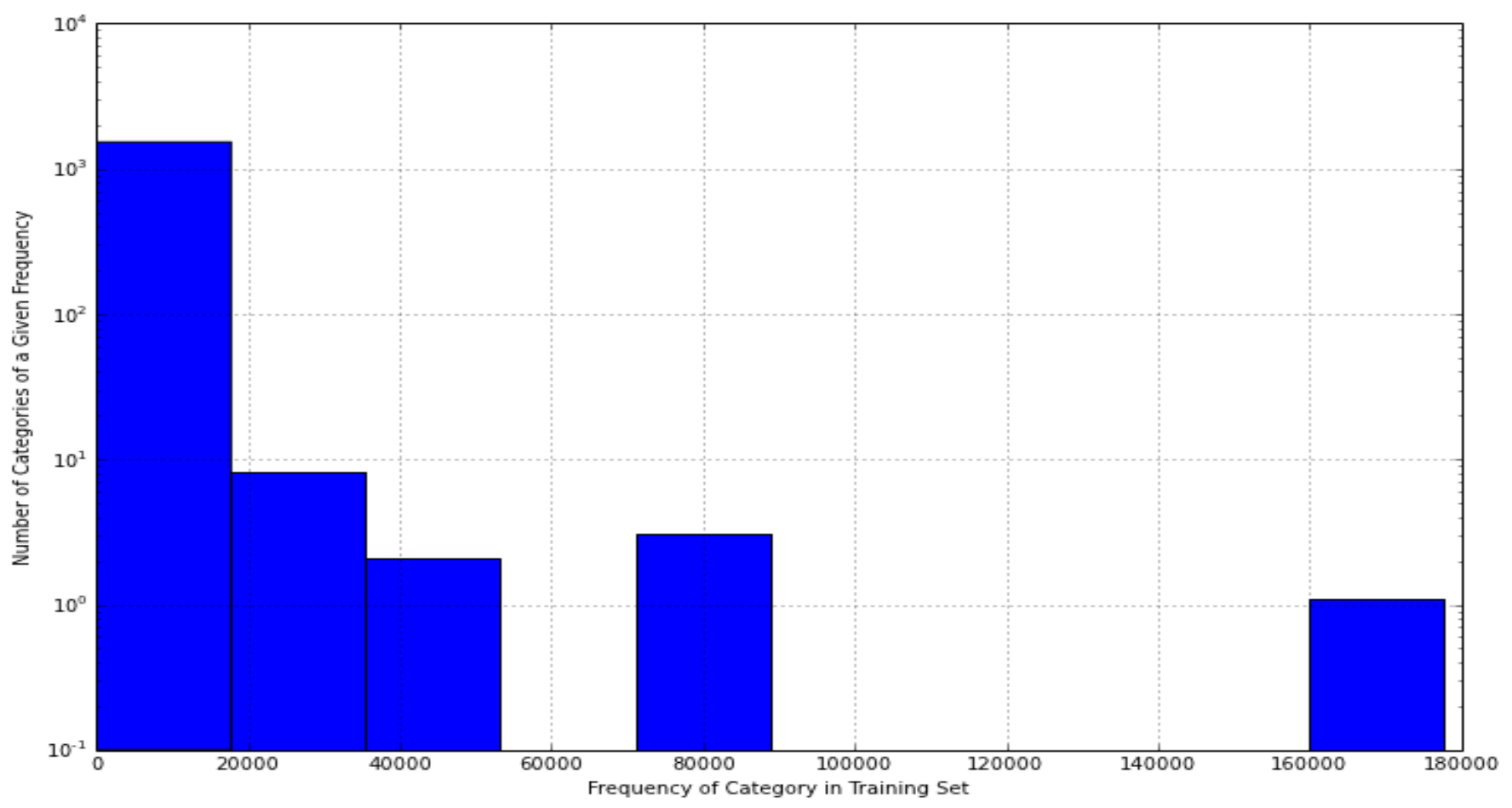
Data Description-SKUs

Frequency of SKUs in Training Set



Data Description-Categories

Frequency of Categories in Training Set



How to predict – Naïve Bayes

Naïve Bayes Classifier

- Well suited to text classification (query)
- Predict popular SKUs given a query term
  - Filter on Category
  - Ignore UserID
  - Query/Click time is an open question
- Efficient: linear complexity

Naïve Bayes

Naïve Bayes is a powerful machine learning algorithm for big data. Here is the idea for the Naïve Bayes:

$$P(C|F_1, F_2, \cdots F_n) = \frac{P(C)P(F_1, F_2, \cdots F_n|C)}{P(F_1, F_2, \cdots F_n)}$$

The goal is to maximize the probability.

In our project, sku# will be our class label. And after preprocessing, the feature we choose is the query term.

Machine Learning Model

- Preprocess data-Data Cleansing
  - Google Refine.
  - Preprocess and Tokenize Query Terms
- Feature Selection
  - Select most frequent unigrams and bigrams (choose 10000 here)
  - Or LSA (poor accuracy!)
- Multinomial NB
  - Feature Set: td-idf terms of words
  - Label Set: SKUs
  - Return 5 most possible SKUs
- Solr Search: reorder the SKUs

How good is our model?

- Benchmark Result: 0.3 @ MAP
- Test on 100 elements:

Total MAP score is 0.57
Total MAX score is 0.82
time is 13.5274701118

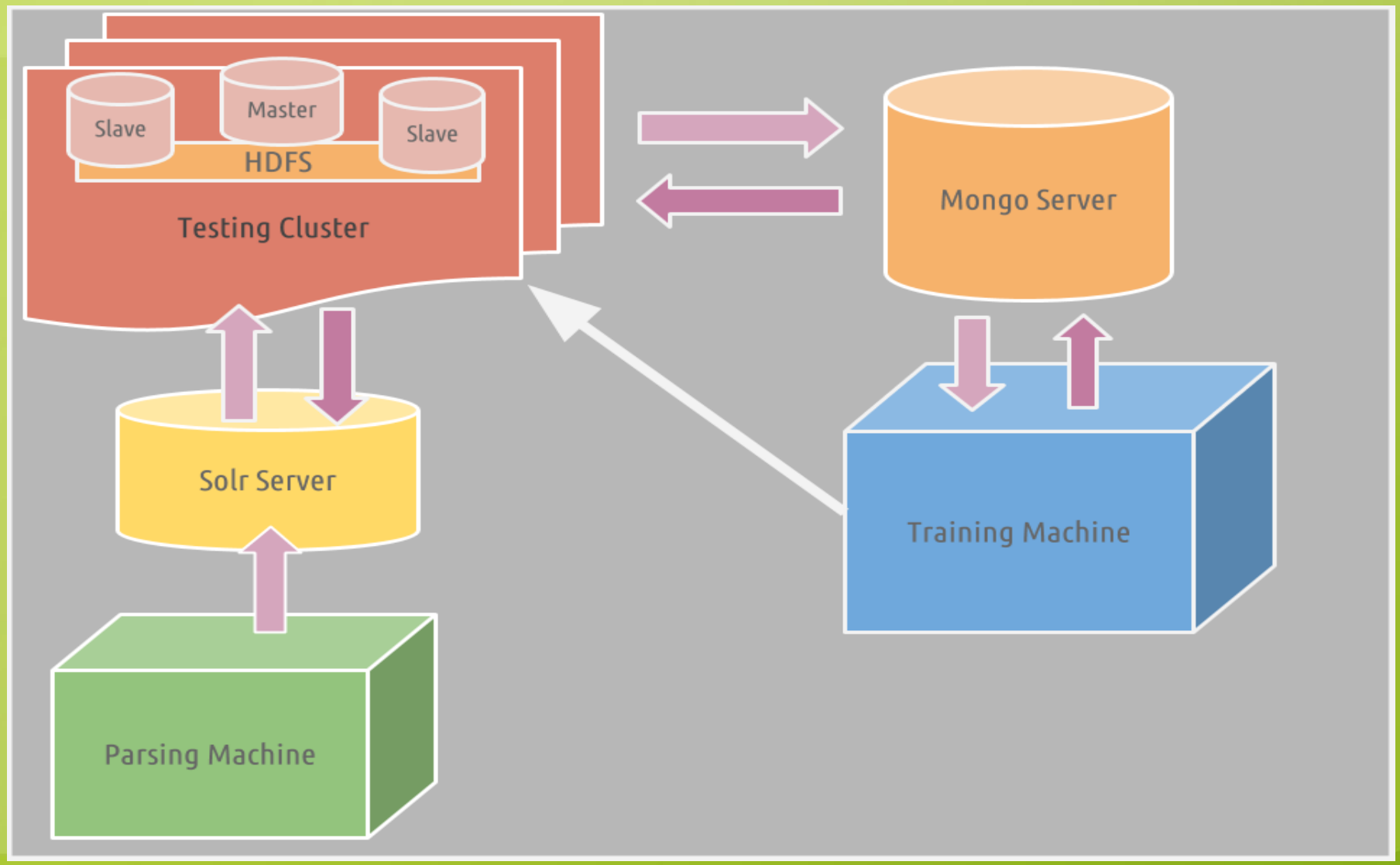
- Test on 1000 elements:

Total MAP score is 0.52
Total MAX score is 0.711
time is 144.144557953

Data is TOOO… BIG

- Export Data into MongoDB
- Naïve Bayes classifiers do not fit into RAM
  - Only load the portion for a given category to RAM
  - Parallelize classification function
  - Training Classifiers in Multiprocessing
- Heavy workload for testing data-read intensive
  - Map-Reduce in data
  - Hadoop
  - Solr search on Cluster

Architecture Diagram



Deliverable - Website

http://lexiemartin.com/query/index.html

Thank you and Questions