Predicting The Best Starting Price for Ebay Auctions

Jiacheng Liao

M.S. in Computer Science.
Interested in machine learning, predictive analytics

Yi Wan

M.S. in Computer Science.
Interested in cloud computing and mobile applications.

Shuang Zhou

M.S. in Computer Science.
Interested in NLP, searching technologies and recommending systems.

Zhaoyin Zhu

Ph.D. in Biostatistics.
Interested in data science and public health.

Contents

Motivation

- Ebay is the world’s largest marketplace
- To study online auction patterns
- To predict the starting price that maximizes profit
- To predict final auction price

Data Source

- Historical auction data was collected from eBay from April 2013 to the first week of May 2013.
- Dataset contains 296,048 observations with 79 columns including starting bidding price, final price and number of bids, etc.

Predictive Analytics Lifecycle

Data Understanding
- Business meaning for each column

Data Preparation
- Clean up data
- Remove unnecessary fields

Feature Selection
- Filter (on Weka, R)
- Wrapper (on RapidMiner)

Modeling
- logistic regression
- linear regression
- decision tree
- random forest
- KNN
- naive bayse

Evaluation
- results comparison
- cross validation

Deployment
- web Application for users

Members Responsibilities

- Jiacheng Liao: data processing, report write-up, modeling in Weka, PPT
- Yi Wan: data processing, feature selection, modeling in Rapidminer & Mahout
- Shuang Zhou: data processing, feature selection in Weka, deployment, report
- Zhaoyin Zhu: data processing, feature selection modeling in R, evaluation

Data Understanding

- The raw datasets contains 79 features
- Auction features: Collected directed from Ebay
- Derived features: Derived from auction features
- A lot of redundant “Category-feature” column

Original Features

Auction Features

Price
StartingBid
BidCount
HitCount
Title
QuantitySold
SellerRating
SellerAboutMePage
StartDate
EndDate
PositiveFeedbackPercent
HasPicture
MemberSince
HasStore

Derived Features

IsHOF
IsAuthenticated
HasInscription
AvgPrice
MedianPrice
AuctionCount
SellerSaleToAveragePriceRatio
SellerAuctionSaleCount
SellerItemSellPercent
StartDayOfWeek
EndDayOfWeek
AuctionDuration
StartingBidPercent

Data Preprocessing

- Remove obviously redundant columns and clean data format
- Remove rows with invalid values
- Preprocess the data sets using shell scripts (cut, sort, unique)

Feature Selection

- Using Filters and Wrappers to select features
- Tools include Weka, R and Rapidminer
- Preserving 95 percent of information
- Narrowed down to 8 features for modeling

Tools and Methods

Filters (Relatively faster)

- Using Weka
  - Information Gain
- Using R
  - Entropy base
  - Chi-Square
  - Correlation

Wrappers (Longer running time)

- Using Rapidminer
  - Naive Bayes
  - Decision Tree
  - Random Forest

Combined Result

Table 2. **Overall Feature Attributing Scores**

Attribute Name	Weka Attributing Score	R Attributing Score	RapidMiner Attributing Score	**Overall Attributing Score**
SellerClosePercent	18	22	16	56
AuctionMedianPrice	12	20	8	40
StartingBid	18	12	8	38
AvgPrice	7	20	8	35
StartingBidPercent	21	10	0	31
AuctionCount	13	16	0	29
SellerAuctionCount	17	8	0	25
AuctionSaleCount	6	14	0	20
ItemAuctionSellPercent	7	4	8	19
SellerItemAvg	8	6	0	14
IsHOF	5	2	0	7

- Narrowed down to 8 features for modeling

Modeling

- Modeling for sale / no sale
- Modeling for final price
- Modeling for best starting price

Modeling for sale / no sale : Trials

- The label of interest is sold (denoted by 1) and not sold (denoted by 0).
- Weka
  - Logistic Regression. advantage: robust to noise, nice probabilistic interpretation, easily update your model to take in new data. Disadvantages: hard to handle categorical (binary) features
  - Naive Bayes classifiers. advantage: simple
- Rapidminer
  - KNN. disadvantage: good for continuous value input.
  - Decision Tree. advantage : handle categorical (binary) features, handle very well high dimensional spaces as well as large number of training examples. Disadvantage: hard to interpret, easily overfit
- Mahout
  - Random Forest. advantages: fast and scalable

Modeling for sale / no sale : Mahout on HDFS

1. Apache Mahout is an open source scalable machine learning library.
2. We choose the random forest algorithm to predict whether a given auction item can be sold or not.
3. random forest: a random forest is a set of decision trees which will product a prediction value. Each decision tree is built using a random subset of the training data. The final prediction value comes from the combination of the output of each tree. Advantage: easy and fast.
4. Disadvantage: hard to interpret.
5. mahout parameter setting: 100 trees, Partition size is is 187423

```
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances      :    25867      87.1119%
Incorrectly Classified Instances    :    3827   12.8881%
Total Classified Instances          :    29694

=======================================================
Confusion Matrix
-------------------------------------------------------
a       b       <--Classified as
6116    2808     |  8924      a       = 1
1019    19751    |  20770   b         = 0

=======================================================
Statistics
-------------------------------------------------------
Kappa                             0.6747
Accuracy                          87.1119%
Reliability                       54.5427%
Reliability (standard deviation)        0.4907
```

Result for sale / no sale

Algorithms	Naive Bayes	Logistic Regression	Decision Tree (depth = 10)	Random Forest
F-measure	0.639	0.674	0.633	0.761

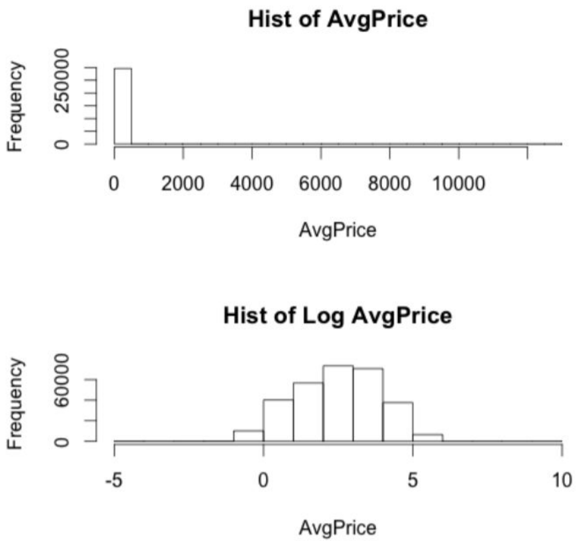
Random Forest achieves the best performance. However, we choose Logistic Regression in our deployment because it provides probabilities of the instance being in each class. We will need that for the computation for the final expectation.

Modeling for final price

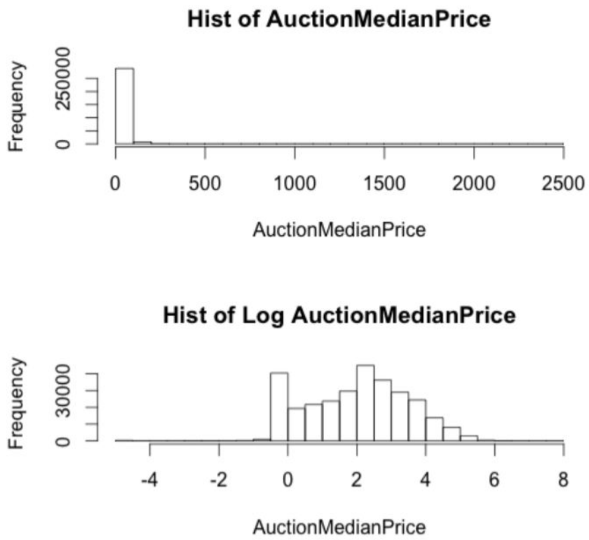
- As the final price is a continuous numerous value, any classification algorithm that deals with discrete values require extra work of partitioning.
- The final price field has a very wide range of values, it will be very hard to distribute instances evenly and maintain similar width of each class at the same time.
- Linear Regression provides a very user-friendly representation of the model that can leverage some insights of the data.
- R
  - Linear Regression in STATS package

Log Transform

AveragePrice and LoggedAveragePrice



MedianPrice and LoggedMedianPrice



After transformation, both originally very skewed fields becomes approximately normal distributed.

Result for final price

	Model Without Log	Model With Log
Bias	0.161	0.159
Standard Deviation	0.0015	0. 0017
Mean Square Error	0.139	0.138

Modeling for best starting price

- The idea is to pick the starting price that maximizes P(sale) * p, where p is the computed final price given this starting price.

$$E_p = Prob(sold|p) \times Price(p)$$

- We set an upper bound and lower bound of starting price percent, as well as an stepsize. Then we go through the range each time incrementing by the stepsize and compute the result for each starting price.

Evaluation

- Use the test dataset which contains 7460 auctions ending in first week of May 2013 to evaluate the models
- Use precision, recall and F-measurement to assess the performance of logistic model
- Use absolute bias, standard deviation and MSE to evaluate linear model
- Primarily done in R

Evaluation Result

Evaluation of logistic model with test dataset

	Without Log	With Log
Precision	0.848	0.858
Recall	0.526	0.528
F-measurement	0.643	0.654

Evaluation of simple linear model with test dataset

	Without Log	With Log
Bias	0.151	0.148
Standard Deviation	0.0017	0.0016
MSE	0.330	0.266

- For logistic regression model, all the precision, recall and f-measurement with log transformation are higher that those without transformation.

- For linear regression model, we can observe a big performance boost when we use log transformation, with a 21% MSE rate drop.

Deployment

- Deploy a Java backed Web Page to provide an interactive service
- The system allows users to query about their items for prediction.
- The backend starts with a model trained with training data, and simply classifies every instance coming in
- Use Weka JAR files. The server hosts on CIMS machines and can be monitored remotely

Demo

Gains

- Experience with various tools : Weka, RapidMiner, R, Mahout
- Deeper understanding of different machine learning & data mining algorithms and techniques
- Practice in predictive analytics lifecycle

Thank You