Predicting The Best Starting Price for Ebay Auctions

Course Project Milestone 2: Prediction on Item sale

Jiacheng Liao[1], Yi Wan[1], Shuang Zhou[1], Zhaoyin Zhu[2]

1 Department of Computer Science, New York University, New York, NY 10012, USA

2 Division of Biostatistics, NYU School of Medicine, New York, NY 10016, USA

Abstract: Online auctions are one of the most popular methods to buy and sell items on the internet. With more than 100 million active users globally, eBay is the worlds largest online marketplace, where anyone can buy and sell anything. In order to successfully selling products on ebay, a reasonable starting price does not only determine whether the product will be sold or not but also affects the profit you can make from the transaction. In this project, we use the historical auction data collected from eBay from April 2013 to the first week of May 2013 which contains information about 296,048 successful and unsuccessful auctions. Different statistical models and machine learning algorithms will be utilized to study online auction patterns and predict the starting price that maximizes profits. Furthermore, we will compare the performance of different methods and summarize the pros and cons in different situations.

Keywords: Ebay Auction • Predictive Analytics • Data Mining

1. Introduction (Data and Business Understanding)

EBay is the worlds largest marketplace for sports autographs, the vast majority of the sites membership uses it to buy and/or sell items via auction format. The ability to provide a method to estimate auction sale prices is desirable to this community. Members of most communities related to collectibles have reported they most often try to predict how much an auction would sell for by performing a search for item and manually calculating the average sales price. In this project, our first objective is to determine whether an auction listing will result in a sale. In addtion, we aim to predict the final sales price as well as the best starting price using data mining techniques.

2. Data Preparation

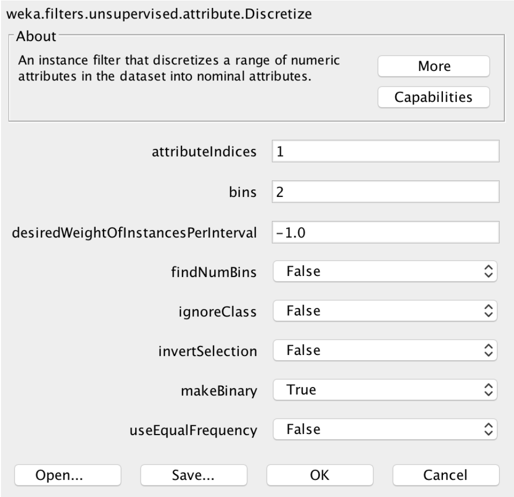
Data Preparation is an crucial and time-consuming part of our data mining project. It involves selecting data to include, cleaning data to improve data quality, constructing new data that may be required, integrating multiple data sets, and formatting data. We first preprocessed the downloaded data sets using shell scripts. To gain more experience on feature selection, we have performed the selection procedure on all three tools used in the class, RapidMiner, Weka and R. With different algorithms and parameters, we get slightly different but generally consistent results.

Data Preprocessing

Before any data can be used for later feature reduction and selection, we preprocessed our data sets. Here the tool we use is shell scripts. Initially, the raw data consists of training sets and test sets. Since we intend to do cross-validation on the whole data set, we merge all the separate data sets as one complete sets. Then we carefully go through all the data attributes and deleted all that obviously contain meaningless or irrelevant information to our analysis, such as ebayID, sellerName, etc. In addition, some attributes with ambiguous meanings are also discarded.

2.1. Feature Selection Using Weka

First, we perform feature selection in Weka. To enable Weka to better handle the data, the first step is establishing a nominal class label. Below is a snapshot of the method we use here to discretize the class field here, which is originally a numeric type with value 0 and 1.



weka.filters.unsupervised.attribute.Discretize

About

An instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes.

attributeIndices: 1

bins: 2

desiredWeightOfInstancesPerInterval: -1.0

findNumBins: False

ignoreClass: False

invertSelection: False

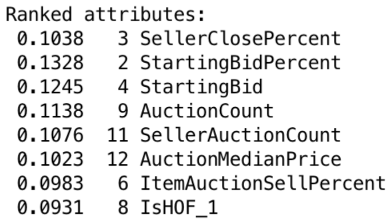
makeBinary: True

useEqualFrequency: False

Two algorithms are used in Weka: Information Gain and Feature Subset Selection. We first used Information Gain Algortihm and the result is as below. According to the information gain results we have, if we set the entropy threshold to 0.95, we will need the top 8 features.

```
Ranked attributes:
 0.318735    3 SellerClosePercent
 0.25068     2 StartingBidPercent
 0.20478    11 SellerAuctionCount
 0.119311    4 StartingBid
 0.07023     7 SellerItemAvg
 0.041762   12 AuctionMedianPrice
 0.040966    5 AvgPrice
 0.031772    9 AuctionCount
 0.028263   10 AuctionSaleCount
 0.005359    6 ItemAuctionSellPercent
 0.000207    8 IsHOF_1
```

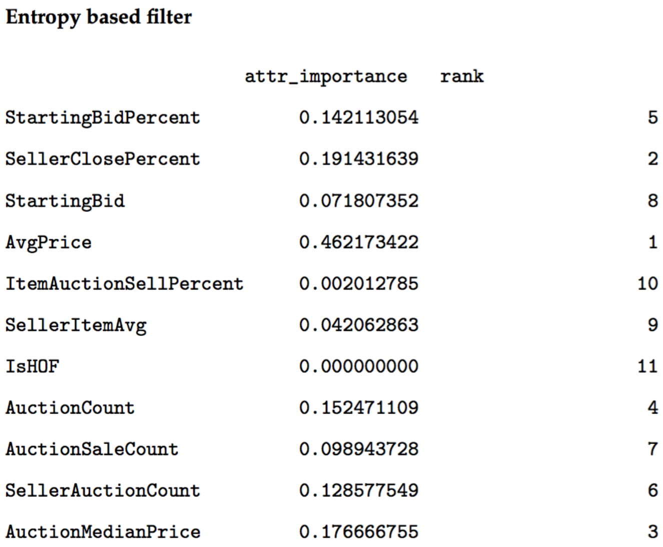
Using the number of features obtained above, we set the number of features to 8 and try feature subset selection. We start with empty subset, and we use CfsSubsetEval as the evaluation of each subset, and we use GreedyStepWise search method, which basically select the best next feature based on current subset. Here is the result we get.



```
Ranked attributes:
 0.1038   3 SellerClosePercent
 0.1328   2 StartingBidPercent
 0.1245   4 StartingBid
 0.1138   9 AuctionCount
 0.1076  11 SellerAuctionCount
 0.1023  12 AuctionMedianPrice
 0.0983   6 ItemAuctionSellPercent
 0.0931   8 IsHOF_1
```

2.2. Feature Selection Using R

The caret R package provides tools automatically report on the relevance and importance of attributes in your data and even select the most important features for you. Here we perform three different feature selection method on our dataset, namely entropy based filter, Chi-square based filter and Correlation based filter. We apply all three filters to get a better sense of what attributes are more important, and we have the results below.



Entropy based filter

	attr_importance	rank
StartingBidPercent	0.142113054	5
SellerClosePercent	0.191431639	2
StartingBid	0.071807352	8
AvgPrice	0.462173422	1
ItemAuctionSellPercent	0.002012785	10
SellerItemAvg	0.042062863	9
IsHOF	0.000000000	11
AuctionCount	0.152471109	4
AuctionSaleCount	0.098943728	7
SellerAuctionCount	0.128577549	6
AuctionMedianPrice	0.176666755	3

Chi-square based filter

```
              attr_importance rank
StartingBidPercent          0.26196726                 5
SellerClosePercent          0.29670612                 2
StartingBid                 0.18784361                 8
AvgPrice                    0.47460475                 1
ItemAuctionSellPercent      0.06509426                10
SellerItemAvg               0.14541187                 9
IsHOF                       0.00000000                11
AuctionCount                0.26282938                 4
AuctionSaleCount            0.20983745                 7
SellerAuctionCount          0.24177050                 6
AuctionMedianPrice          0.29014051                 3
```

Correlation based filter

```
                  attr_importance  rank
StartingBidPercent          0.05002078                10
SellerClosePercent          0.62856687                 1
StartingBid                 0.16766258                 3
AvgPrice                    0.10793493                 6
ItemAuctionSellPercent      0.08816751                 7
SellerItemAvg               0.07423485                 8
IsHOF                       0.01689967                11
AuctionCount                0.11040403                 5
AuctionSaleCount            0.16330465                 4
SellerAuctionCount          0.07086579                 9
AuctionMedianPrice          0.18235711                 2
```

As a result, we summarize all the three different methods and compute average rank for each attribute. The result is presented in table 1.

2.3. Feature Selection Using RapidMiner

First we choose a feature selection method from RapidMiner. Here we use Forward Selection. Forward selection operator selects the most relevant attributes of the given ExampleSet through a highly efficient implementation of the forward selection scheme.

Forward selections uses wrapper method to select attributes. Basically, it starts with an empty attribute set. It them add one attribute to run the model and measures the performance. The process keep adding attributes to the model to see if there is performance gain. Depending on the parameter set, it will terminate until there is

Table 1. **Feature Selection Using R Result**

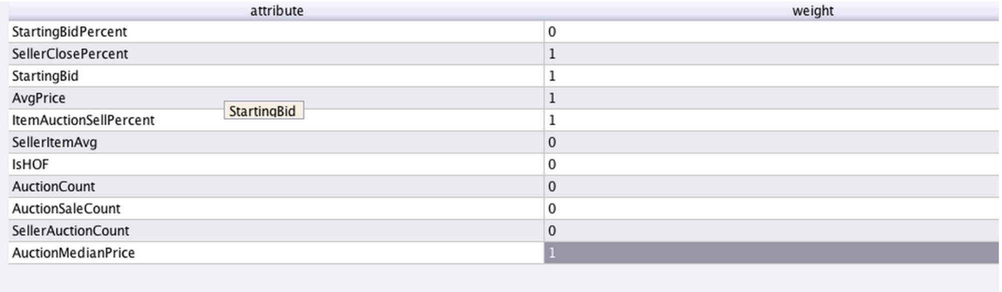
Attribute Name	Entropy based filter	Chi-square based filter	Correlation based filter	Average Rank
SellerClosePercent	2	2	1	1.7
AvgPrice	1	1	6	2.7
AuctionMedianPrice	3	3	2	2.7
AuctionCount	4	4	5	4.3
AuctionSaleCount	7	7	4	6.0
StartingBid	8	8	3	6.3
StartingBidPercent	5	5	10	6.7
SellerAuctionCount	6	6	9	7.0
SellerItemAvg	9	9	8	8.7
ItemAuctionSellPercent	10	10	7	9.0
IsHOF	11	11	11	11.0

no performance improvement or no significant performance improvement. Here we also use cross-validation to measure the performance of the model as well as the current attributes set the operator.

For predicting the whether the given item can be sold or not, we use several classification algorithms, including: Naive Bayes, Decision Tree, Random Forest, Rule Induction, Neural net, Logistic Regression, Support Vector Machine. We will consider efficiency and prediction accuracy for each model to decide which one to adopt.

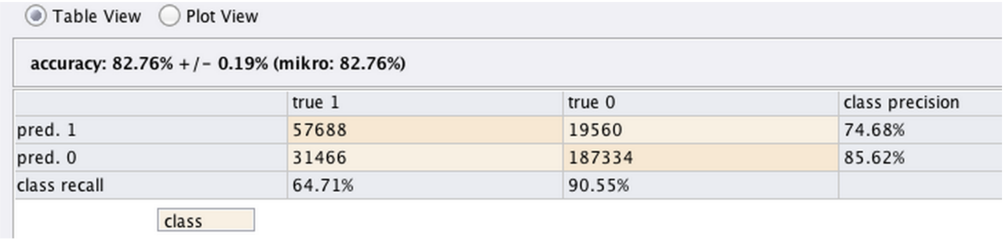
2.3.1. Naive Bayes modeling

A. The attribute weight



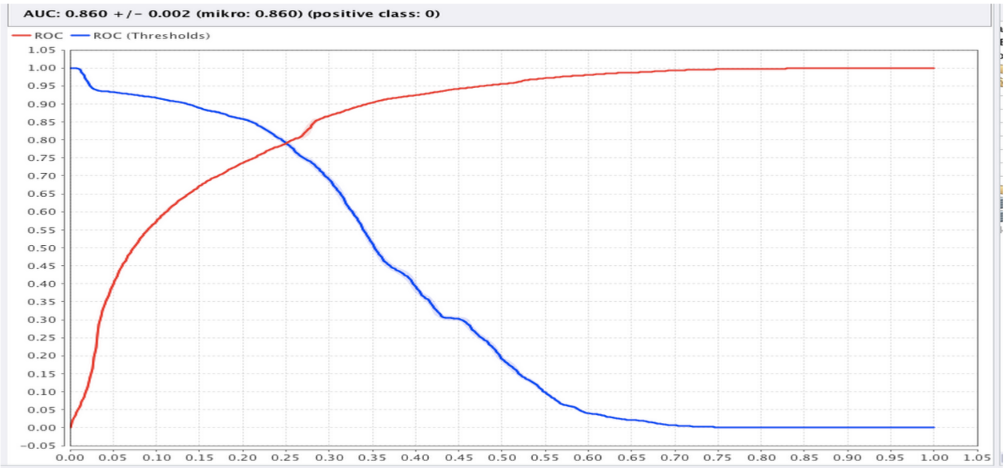
attribute	weight
StartingBidPercent	0
SellerClosePercent	1
StartingBid	1
AvgPrice	1
ItemAuctionSellPercent	1
SellerItemAvg	0
IsHOF	0
AuctionCount	0
AuctionSaleCount	0
SellerAuctionCount	0
AuctionMedianPrice	1

B. The prediction model result and auc curve



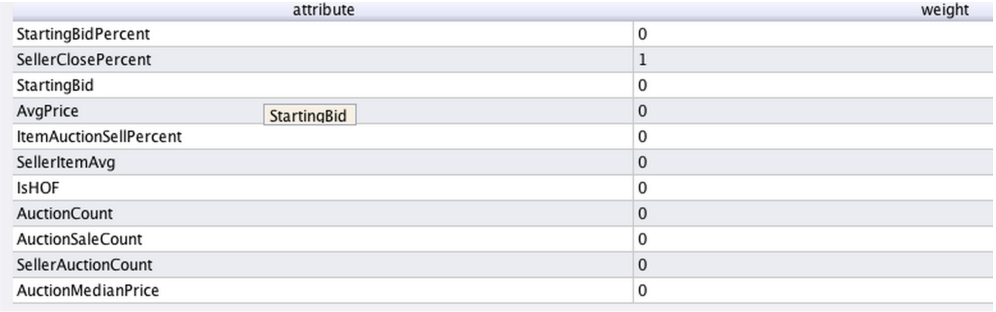
accuracy: 82.76% +/- 0.19% (mikro: 82.76%)

	true 1	true 0	class precision
pred. 1	57688	19560	74.68%
pred. 0	31466	187334	85.62%
class recall	64.71%	90.55%	



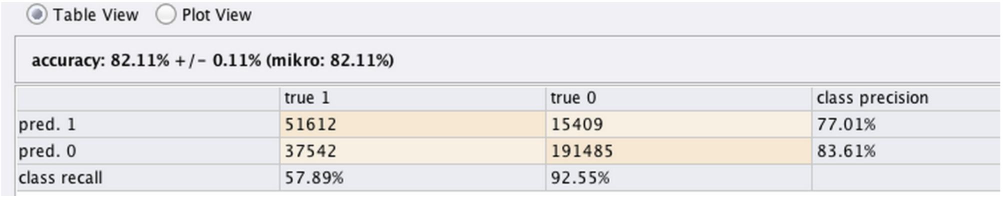
2.3.2. Decision Tree modeling

A. The attribute weight



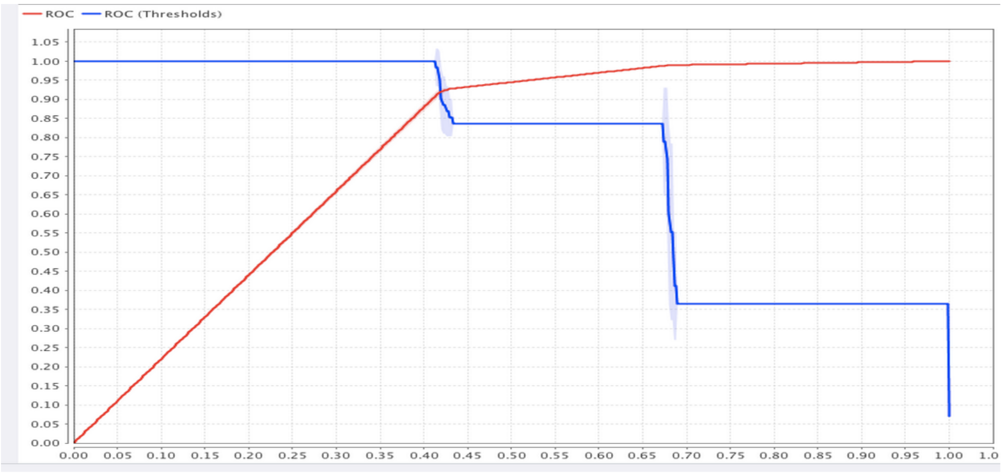
attribute	weight
StartingBidPercent	0
SellerClosePercent	1
StartingBid	0
AvgPrice	0
ItemAuctionSellPercent	0
SellerItemAvg	0
IsHOF	0
AuctionCount	0
AuctionSaleCount	0
SellerAuctionCount	0
AuctionMedianPrice	0

B. The prediction model result and auc curve



accuracy: 82.11% +/- 0.11% (mikro: 82.11%)

	true 1	true 0	class precision
pred. 1	51612	15409	77.01%
pred. 0	37542	191485	83.61%
class recall	57.89%	92.55%	



2.4. Other Trials

The other methods including Random Forest, Rule Induction, Neural net, Logistic Regression, Support Vector Machine takes a very long time to finish (more than one hour). Since the dataset we are using is not very large, we consider them not suitable for our problem.

2.5. Final Feature Selection

After integrating all results from 3 different tools, we narrowed the features down to these list of 8 features.

- StartingBidPercent
- SellerClosePercent
- StartingBid
- AvgPrice
- AuctionCount
- AuctionSaleCount
- SellerAuctionCount
- AuctionMedianPrice

And from here on, unless specified otherwise, we are using the reduced dataset with only these features.

3. Team Members Responsibilities and Plan for the Next Phase

Right now, all group members are actively participate in the project. An approximate list of each team members' responsiblity is:

- Jiacheng Liao: data preprocessing, report write-up
- Yi Wan: data preprocessing, feature selection in Rapidminer
- Shuang Zhou: data preprocessing, feature selection in Weka
- Zhaoyin Zhu: data preprocessing, feature selection in R

For the next phase,we intend to build different models to make predictions. We plan to apply various statistical machine learning model and use cross-validation to test each model. Also, we would study some economic aspects of the auction theory to help us build the model.

4. Modeling

Modeling involves selecting suitable modeling techniques, generating test designs to validate the model, building predictive models and assessing these models.

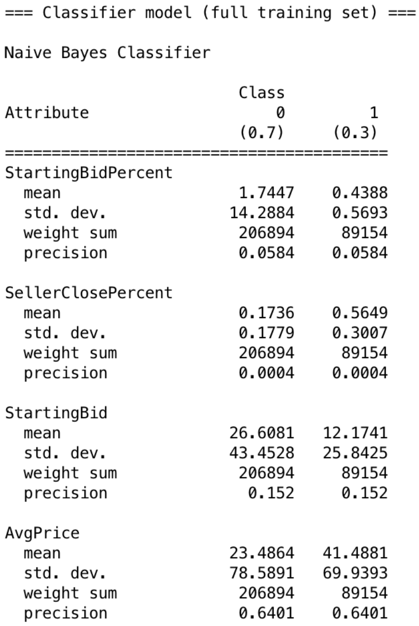
A predictive model is a mathematical function that predicts the value of some output variables based on the mapping between input variables. Historical data is used to train the model to arrive at the most suitable modeling technique. For example, a predictive model might predict the risk of developing a certain disease based on patient details. Some commonly used modeling techniques are as follows: Regression analysis that analyzes the relationship between the response or dependent variable and a set of independent or predictor variables. Decision trees that help explore possible outcomes for various options. Cluster analysis that groups objects into clusters to look for patterns. Association techniques that discover relationships between variables in large databases.

4.1. Modeling for Sale / No Sale

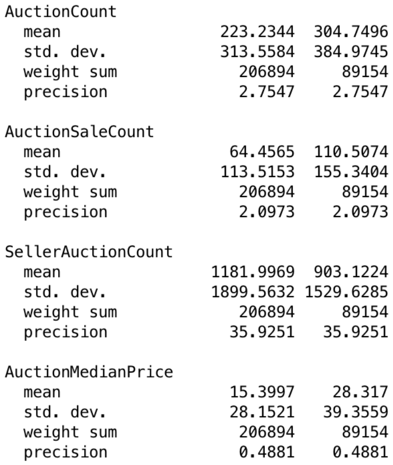
In all three tools here, we tried to use different type of classification algorithms to decide whether items can be sold or not based on some information of item across Ebay.

4.1.1. Modeling in Weka

In Weka, we tried to models, Naive-bayes model and Logistic Regression model. For NB, the model computed is as follow:

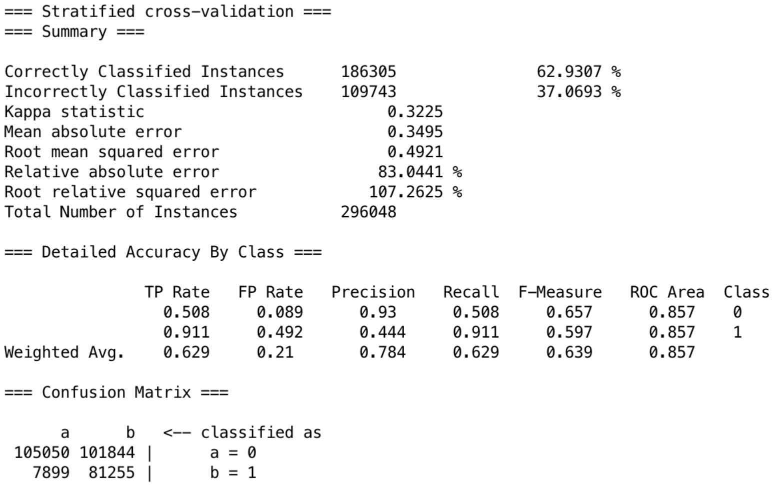


```
=== Classifier model (full training set) ===

Naive Bayes Classifier

                         Class
Attribute                    0         1
                         (0.7)     (0.3)
=========================================
StartingBidPercent
  mean                  1.7447    0.4388
  std. dev.            14.2884    0.5693
  weight sum            206894     89154
  precision             0.0584    0.0584

SellerClosePercent
  mean                  0.1736    0.5649
  std. dev.             0.1779    0.3007
  weight sum            206894     89154
  precision             0.0004    0.0004

StartingBid
  mean                 26.6081   12.1741
  std. dev.            43.4528   25.8425
  weight sum            206894     89154
  precision              0.152     0.152

AvgPrice
  mean                 23.4864   41.4881
  std. dev.            78.5891   69.9393
  weight sum            206894     89154
  precision             0.6401    0.6401
```



```
AuctionCount
  mean                223.2344  304.7496
  std. dev.           313.5584  384.9745
  weight sum            206894     89154
  precision             2.7547    2.7547

AuctionSaleCount
  mean                 64.4565  110.5074
  std. dev.           113.5153  155.3404
  weight sum            206894     89154
  precision             2.0973    2.0973

SellerAuctionCount
  mean               1181.9969  903.1224
  std. dev.          1899.5632 1529.6285
  weight sum            206894     89154
  precision            35.9251   35.9251

AuctionMedianPrice
  mean                 15.3997    28.317
  std. dev.            28.1521   39.3559
  weight sum            206894     89154
  precision             0.4881    0.4881
```

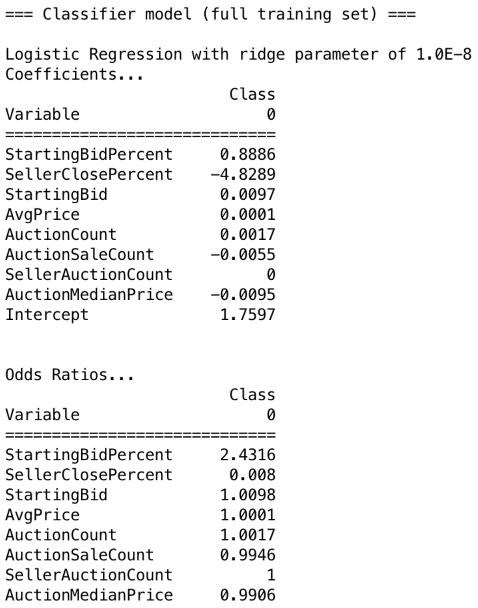
And also we evaluated the performance of this model:



```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      186305               62.9307 %
Incorrectly Classified Instances    109743               37.0693 %
Kappa statistic                          0.3225
Mean absolute error                      0.3495
Root mean squared error                  0.4921
Relative absolute error                 83.0441 %
Root relative squared error            107.2625 %
Total Number of Instances           296048

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall   F-Measure   ROC Area   Class
                 0.508     0.089      0.93      0.508     0.657      0.857      0
                 0.911     0.492      0.444     0.911     0.597      0.857      1
Weighted Avg.    0.629     0.21       0.784     0.629     0.639      0.857

=== Confusion Matrix ===

      a      b   <-- classified as
 105050 101844 |      a = 0
   7899  81255 |      b = 1
```

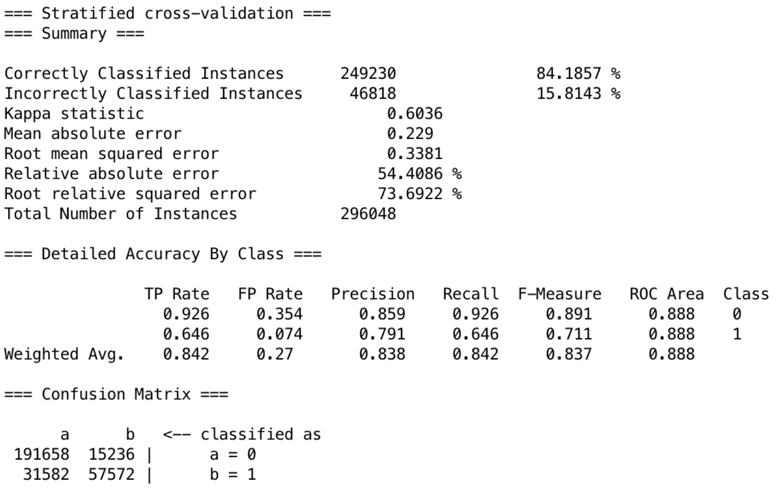
We decided to use F measure as the criteria to measure performances of each model. For NB model, the F measure is 0.639.

Now we go on and try another model, Logistic Regression. Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution. It fits very well in our situation, and it's a fast model to train and use. Below is the model we get:



```
=== Classifier model (full training set) ===

Logistic Regression with ridge parameter of 1.0E-8
Coefficients...
                        Class
Variable                    0
=============================
StartingBidPercent     0.8886
SellerClosePercent    -4.8289
StartingBid            0.0097
AvgPrice               0.0001
AuctionCount           0.0017
AuctionSaleCount      -0.0055
SellerAuctionCount          0
AuctionMedianPrice    -0.0095
Intercept              1.7597


Odds Ratios...
                        Class
Variable                    0
=============================
StartingBidPercent     2.4316
SellerClosePercent      0.008
StartingBid            1.0098
AvgPrice               1.0001
AuctionCount           1.0017
AuctionSaleCount       0.9946
SellerAuctionCount          1
AuctionMedianPrice     0.9906
```

Again, we run 10 fold cross-validation on this model, and here is what we get:

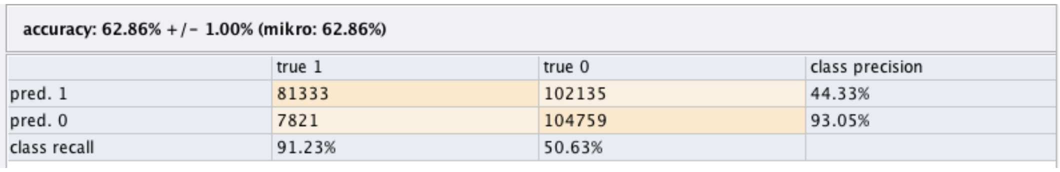


```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      249230               84.1857 %
Incorrectly Classified Instances     46818               15.8143 %
Kappa statistic                          0.6036
Mean absolute error                      0.229
Root mean squared error                  0.3381
Relative absolute error                 54.4086 %
Root relative squared error             73.6922 %
Total Number of Instances           296048

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.926     0.354     0.859     0.926     0.891      0.888    0
                 0.646     0.074     0.791     0.646     0.711      0.888    1
Weighted Avg.    0.842     0.27      0.838     0.842     0.837      0.888

=== Confusion Matrix ===

      a      b   <-- classified as
 191658  15236 |      a = 0
  31582  57572 |      b = 1
```

Obviously Logistic Regression model works much better than NB model, with a F measure of 0.837. Thus Logistic Regression will be our model of choice in Weka for this project.

4.1.2. Modeling in RapidMiner

In RapidMiner, models we tried include Naive Bayes, Random Forest, KNN, Decision Tree, Logistic Regression, Neural Network. We present Naive Bayse, Random Forest, and Decision Tree results here. We use cross validation for testing the performance. The figure below shows the performance for Naive Bayes.



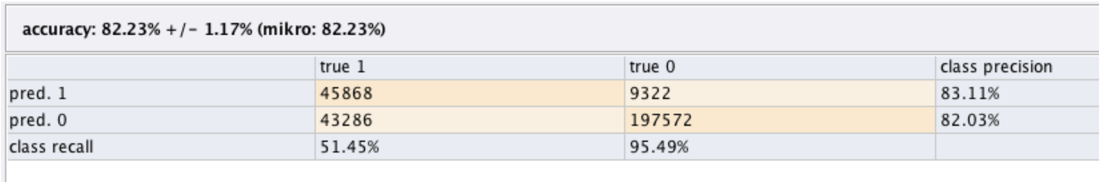
accuracy: 62.86% +/- 1.00% (mikro: 62.86%)

	true 1	true 0	class precision
pred. 1	81333	102135	44.33%
pred. 0	7821	104759	93.05%
class recall	91.23%	50.63%	

As we can see, too many instances that should belong to class "0" are assigned to class "1", actually 70 percent of all instances are assigned to class "1" based on this model. Therefore, we can say that NB model is a little biased in this case and we shouldn't rely on the result.

The second model we tried is Decision Tree classifier. A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. The decision tree algorithm comes with a "depth" parameter and a evaluation algorithm. For the evaluation algorithm, we used Gain Ratio, and to obtain the best result, we tried multiple depths, and here is what we get.

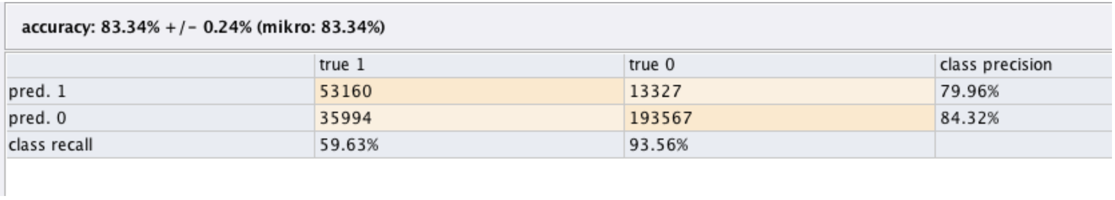
Decision Tree performance with depth=5



accuracy: 82.23% +/- 1.17% (mikro: 82.23%)

	true 1	true 0	class precision
pred. 1	45868	9322	83.11%
pred. 0	43286	197572	82.03%
class recall	51.45%	95.49%	

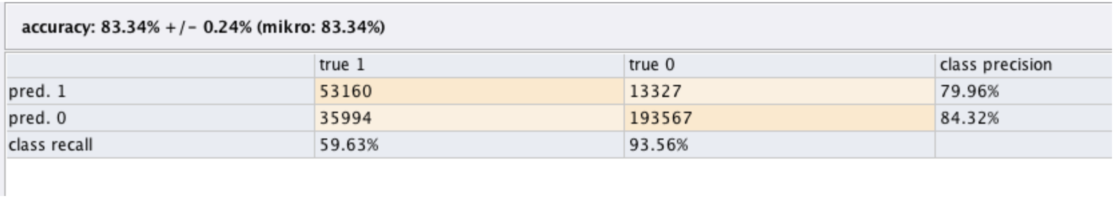
Decision Tree performance with depth=10



accuracy: 83.34% +/- 0.24% (mikro: 83.34%)

	true 1	true 0	class precision
pred. 1	53160	13327	79.96%
pred. 0	35994	193567	84.32%
class recall	59.63%	93.56%	

Decision Tree performance with depth=20



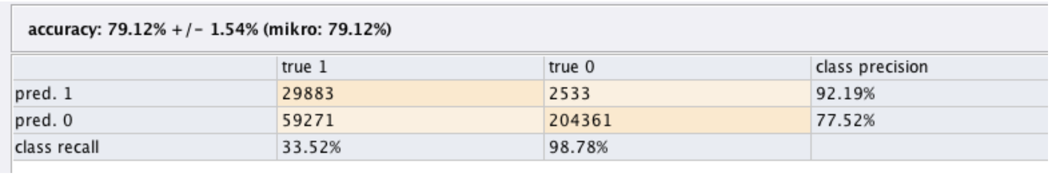
accuracy: 83.34% +/- 0.24% (mikro: 83.34%)

	true 1	true 0	class precision
pred. 1	53160	13327	79.96%
pred. 0	35994	193567	84.32%
class recall	59.63%	93.56%	

As we can see, the overall accuracy goes up a little when the depth increases from 5 to 10, but then the performance stays the same. Intuitively this makes sense because as the depth go up, it's more likely to find relations between each attribute and the label, but given there're only 8 features in total, the depth wouldn't matter too much after it exceeds the number of features.

The third model we tried is Random-Forest classifier. Random forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random forests correct for decision trees' habit of overfitting to their training set.

Random Forest algorithm needs a user specified number of trees to start with, and here we fix it to 10 in all experiments. Still we use Gain Ratio as the evaluation algorithm, here is what we get:



accuracy: 79.12% +/- 1.54% (mikro: 79.12%)

	true 1	true 0	class precision
pred. 1	29883	2533	92.19%
pred. 0	59271	204361	77.52%
class recall	33.52%	98.78%	

We can tell that the overall accuracy is slightly lower than Decision Tree algorithm, but as a property of Random Forest, it tends to get rid of overfitting behaviors of decision trees, so this is acceptable. And we will find out which one is more accurate when we apply them on unknown datasets.

4.1.3. Modeling in R

We ran Logistic Regression algorithm from the famous "Generalized Linear Model" package in R to start with,

5. Evaluation (ToDo)

Evaluation involves evaluating the results against the business success criteria defined at the beginning of the project.

6. Deployment (ToDo)

Deployment involves consolidating the findings, determining what might be deployed and planning the monitoring and maintenance required to keep the model relevant.

7. Conclusion (ToDo)

TODO

Acknowledgements

The author(s) would like to thank some institutions for support and so on.

References

[1] Han, Jiawei, Micheline Kamber, and Jian Pei. *Data mining: concepts and techniques: concepts and techniques*. Elsevier, 2011.

[2] Rajaraman, Anand, and Jeffrey D. Ullman. *Mining of massive datasets*. Vol. 77. Cambridge: Cambridge University Press, 2012.

[3] Bari, Anasse, Mohamed Chaouchi, and Tommy Jung. *Predictive analytics for dummies*. John Wiley & Sons, 2014.

[4] Bari, Anasse. Predictive Analytics Course Lecture Notes. 2015 Fall.

[5] Brownlee, Jason. Feature Selection with the Caret R Package. http://machinelearningmastery.com/feature-selection-with-the-caret-r-package/

[6] Grossman, Jay. Predicting eBay Auction Sales with Machine Learning.
Retrived from http://jaygrossman.com/post/2013/06/10/Predicting-eBay-Auction-Sales-with-Machine-Learning.aspx