eBay Auctions
MATH-564 Final Report

Megha Lokanadham
mlokanadham@hawk.iit.edu
A20403188

Raphaël Cohen
rcohen6@hawk.iit.edu
A20432518

Pranav Behari Lal
plal2@hawk.iit.edu
A20417025

1. Overview

Online auctions are one the most popular methods to buy and sell items on the internet. With more than 100 million active users globally, eBay is the world's largest online marketplace, where practically anyone can buy and sell practically anything. The total value of goods sold in Q4 of 2011 on eBay was $68.6 billion, more than $2,100 every second [1]. This kind of volume produces huge amounts of data that can be utilized to provide services to the buyers and sellers, market research, and product development. In this analysis, we use the historical eBay Auctions dataset to predict the likelihood of an item being sold, and the final selling price for these items. The dataset used contains ~300,000 auction items in the train and test datasets from the sports autograph category on eBay.[data link]

In this paper we are trying to answer two broad questions: (a) Can we predict the likelihood of an auction item getting sold, and (b) Can we predict the final selling price of an auction item, given information on its category, starting bid value and so on.

For the first part, we have implemented and optimized three popular classification algorithms, namely Logistic Regression, Random Forest and Linear Discriminant Analysis. We aim to optimize all three algorithms, and complete a comparative study of the performance improvements within each algorithm. To predict the final selling price, we have implemented Multiple Linear Regression, and the final approach involves identifying clusters within our dataset, and fitting a separate regression model on each cluster. Several tests have been performed to test the suitability of each method and to reduce the overall error rate.

2. Classification

The original dataset consists of 28 variables, including values for the final selling price of an item and its derived attributes. Since we are trying to build a model to predict if an item will be sold at all, the variables associated with the final selling price have been

discarded. We have also removed variables that have no explanatory power such as the eBay auction ID and the seller name. The final dataset used to train our classification models included 18 predictor variables and 1 target variable - *QuantitySold* - which is set to 1 in the case of a sale and 0 otherwise.

2.1. Logistic Regression.

The first model implemented was a Logistic Regression model. Before getting into feature selection and model optimization, we built a baseline model that gave us an overall prediction accuracy of 89.61% on the test dataset. The reason for such a high accuracy value is the distribution of the target variables in our dataset, with approximately 70% of the entries depicting items that did not lead to a successful sale. This was further seen with the specificity of our baseline model, with a low specificity score of 67.72%. The optimizations performed further were an attempt to maintain or slightly improve the overall accuracy while drastically improving the specificity of our model on the test dataset.

2.1.1 Feature Selection

We used stepwise feature selection, with direction set to 'forward', 'backward' and 'both', and also the best subsets algorithm to define the best suited subset of predictor variables. The results of all 4 feature selection algorithms were unimpressive, with just 1 variable - *ReturnsAccepted* - being removed from the original predictor variable. The *ReturnsAccepted* variable had no explanatory power since its value was set to 0 for all the entries in the training dataset. So, we had to adopt different techniques to improve the predictive power of our model.

2.1.2 Weighted Logistic Regression

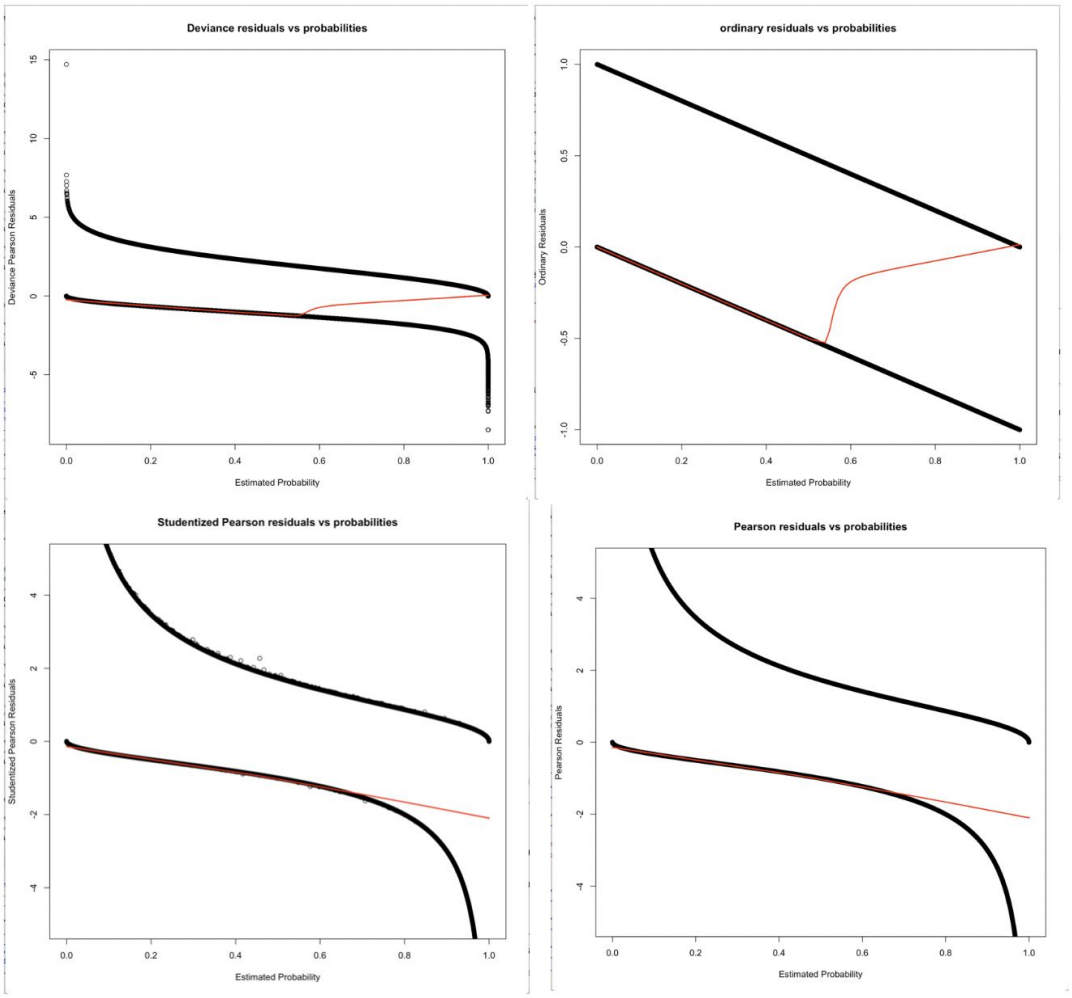
Considering the large skew in the distribution of our target variable, weighted models appeared a good fit. We attempted to fit two weighted models, assigning weights of 2x and 3x to the target variable *QuantitySold* with a positive value. The results showed an immediate improvement to our baseline and feature selected models:

	Accuracy	**Sensitivity**	**Specificity**
Baseline	89.61	96.95	67.72
2x weights	89.19	93.54	76.21
3x weights	88.11	90.49	81.01

While we do see a drop in the overall accuracy, the significant increase in the specificity is a justifiable compromise since we wish to predict the the probability of a sale. Having a high specificity is essential for our use case.

2.1.3 Model Diagnostics

After finalizing on a 3x weighted model, we proceeded with the model diagnostics, starting with the residual plots.



The approximately horizontal lines for the lowess smooth on the above residual plots points to a good fit for the model, that is, **E{Yi} = Πi.**

2.1.4 Goodness of Fit Tests

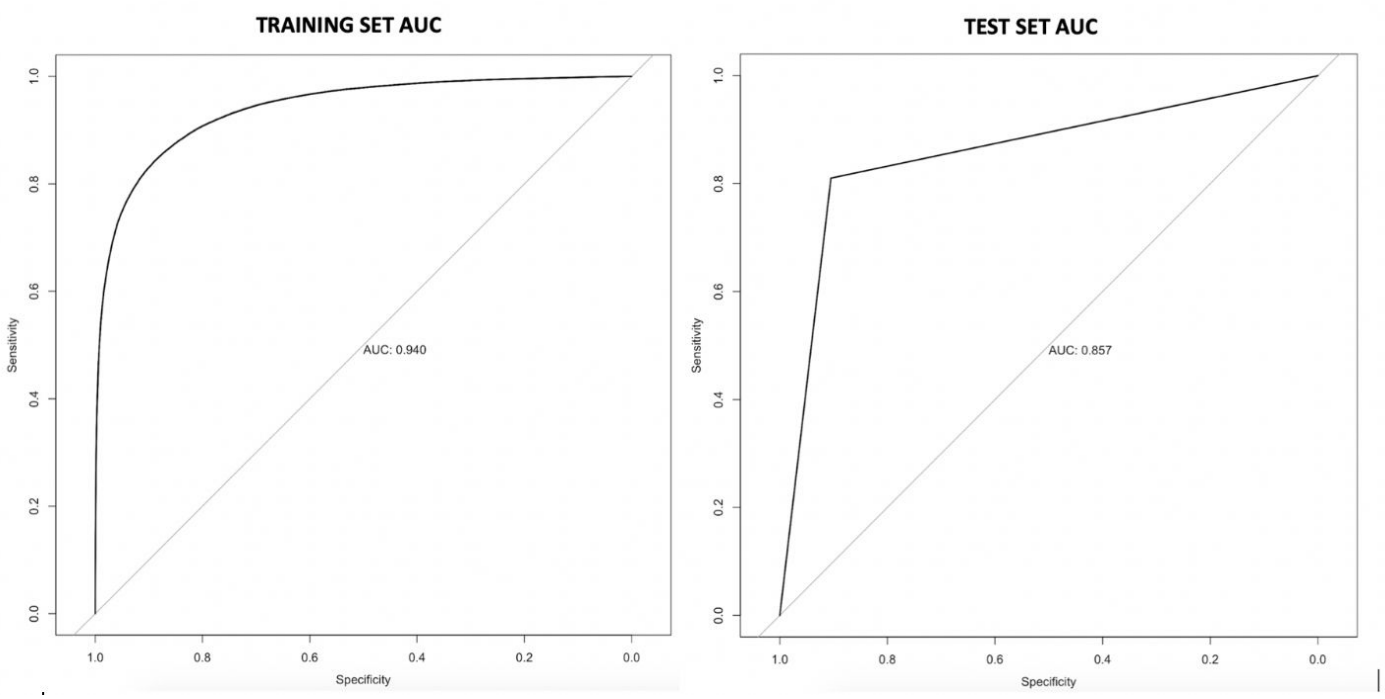
While the residual plots do hint at a good fit in the model, we covered 2 goodness of fit tests to validate our results.

2.1.4.1 Wald Test

A Wald Test was performed on all the individual predictor variables to measure if the variables can be removed from the model. Since we obtained these features after our initial feature selection, we have selected a high value of α = 0.01 to draw a meaningful result from the test. With a high mean p-value < 2.22e-16 across all 18 tests for the predictor variables, we have concluded that all 18 predictor variables should remain in the model with no loss in predictive power.

2.1.4.2 Receiver Operating Characteristic (ROC) Curves

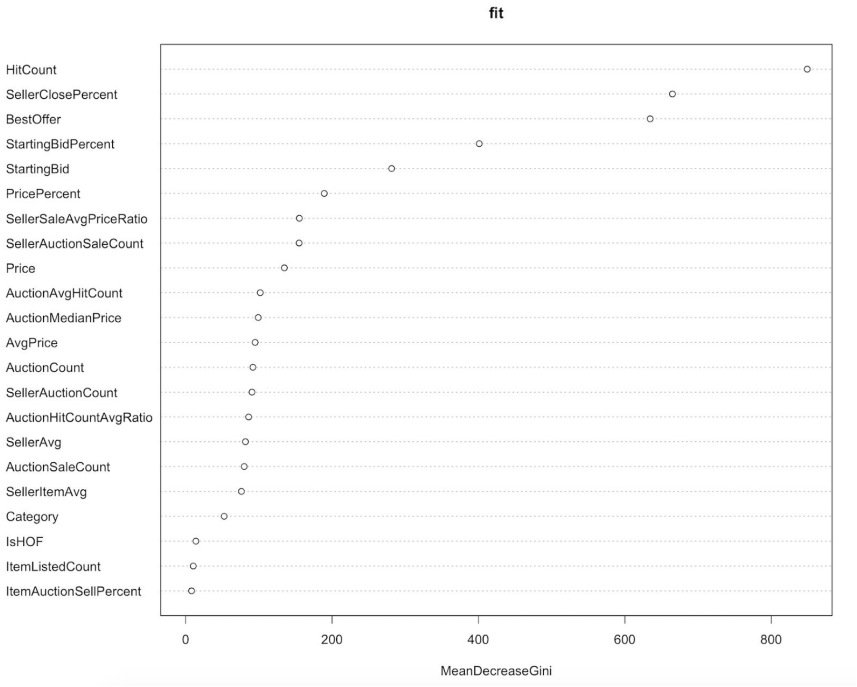
ROC curves were generated for the train and test datasets to measure the Area Under Curve (AUC).



Using the 3x weighted model leads to an AUC of 0.94 on the training dataset and an AUC of 0.857 on the test dataset, which both point to a significantly good fit.

2.1.5 Higher Order Predictors

So far, we have arrived at a performant model with statistically backed features. We now had to test if including higher order terms in our model would result in a better fit. A variable importance plot for our dataset yielded *HitCount*, *SellerClosePercent*, *BestOffer* and *StartingBidPercent* as the top 4 most important variables.



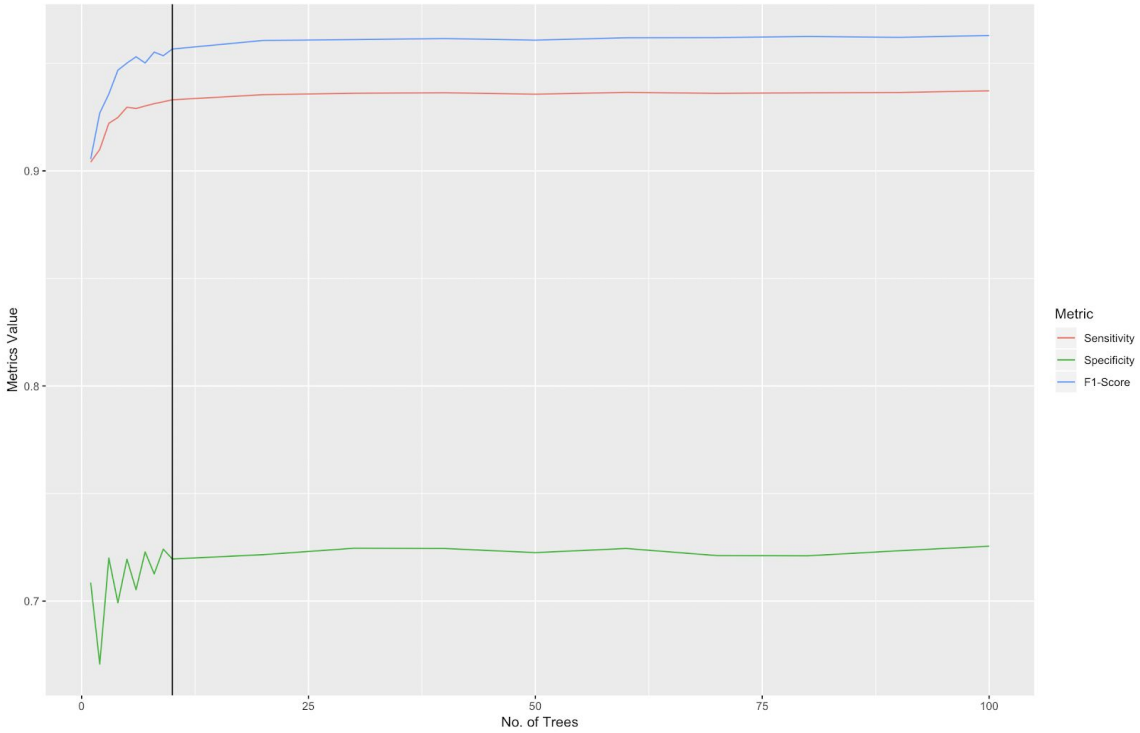
These are the 4 variables that were chosen as viable candidates for inclusion in their second order. A higher order model term with all predictor variables, including the higher order and interaction terms for the above 4 variables, was fit and a Likelihood Ratio Test was performed to justify their inclusion. The conclusion of the test was to accept the null hypothesis that the higher order terms can be removed from our model.

2.1.6 Conclusion

To conclude with Logistic Regression, we have fit a 3x weighted logistic regression model on the 18 predictor variables and 1 target variable in the first order to achieve a sensitivity and specificity of 90.49 and 81.01 on the test data set. The overall accuracy of the model is 88.11%.

2.2. Random Forest

To compare the performance of the Logistic Regression Model, Random Forest and Linear Discriminant Analysis were performed on the dataset. A baseline line Decision Tree (Random Forest with 1 tree) yielded an overall accuracy of 85.68%, with a sensitivity and specificity of 90.60% and 70.98%. To determine the ideal forest size, random forests number of tree sizes ranging from 1 to 100 were fit and compared on their specificity, sensitivity and F1-score.

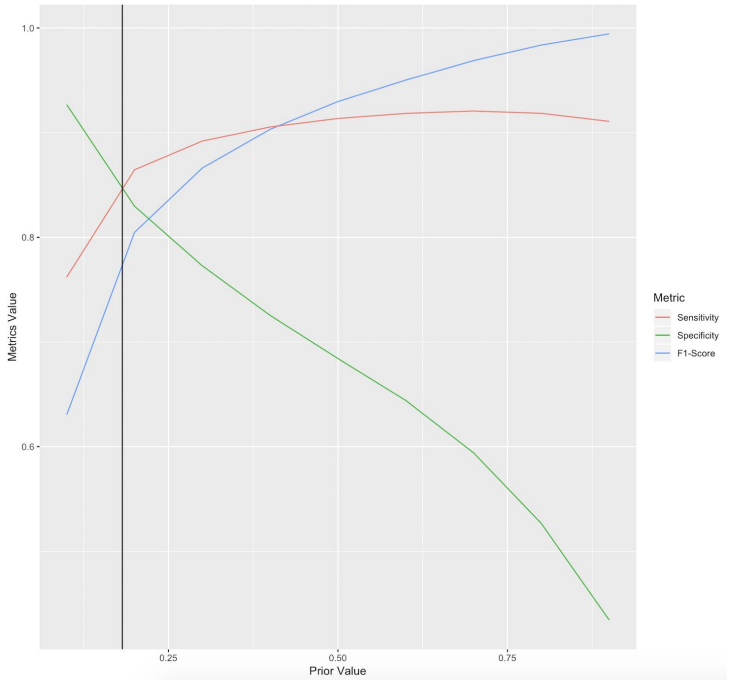


The above plot shows a monotonic improvement in performance for tree size 10, beyond which the model performance essentially remains constant. As such, 10 trees was chosen as an optimal size of our Random Forest, which yielded an overall accuracy of 90.34%, with sensitivity and specificity of 96.30% and 72.55%.

2.3. Linear Discriminant Analysis

The last classification model tested was the Linear Discriminant Analysis. Optimizations on LDA involved adjusting the values of the prior.

Based on the below plot, a prior probability of 0.22 was selected for a negative target variable *QuantitySold*, the model for which yields an overall test accuracy of 87.47%, with a sensitivity and specificity of 96.72% and 59.84%.



2.4 Classification Conclusion

Three models - Logistic Regression, Random Forests and Linear Discriminant Analysis - were implemented and optimized. The results for the models are summarized below:

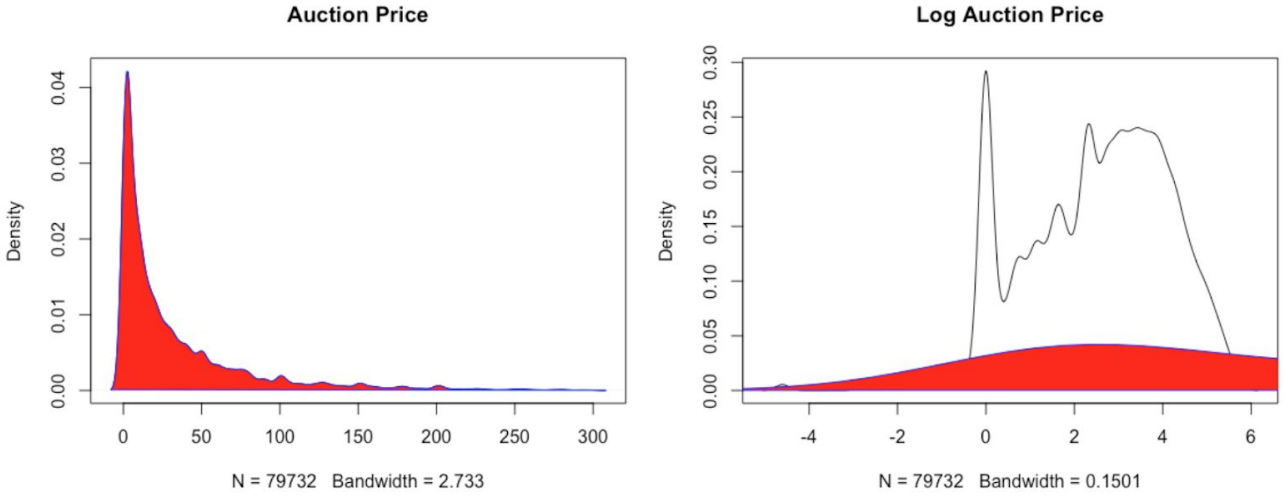
	Accuracy	Sensitivity	Specificity	Balanced Accuracy
3x Logistic Regression	88.11	90.49	81.01	85.75
Random Forest	90.34	96.3	72.55	84.43
Linear Discriminant Analysis	87.47	96.72	59.84	78.28

Linear Discriminant Analysis, while it does have the highest sensitivity score of the three models, is the poorest performer. The high sensitivity value is attributed to the skew in the distribution of our target variable. The balanced accuracy of Logistic Regression and Random Forest have them performing almost equally well, but we have decided with Logistic Regression as our final model solely on its high specificity score.

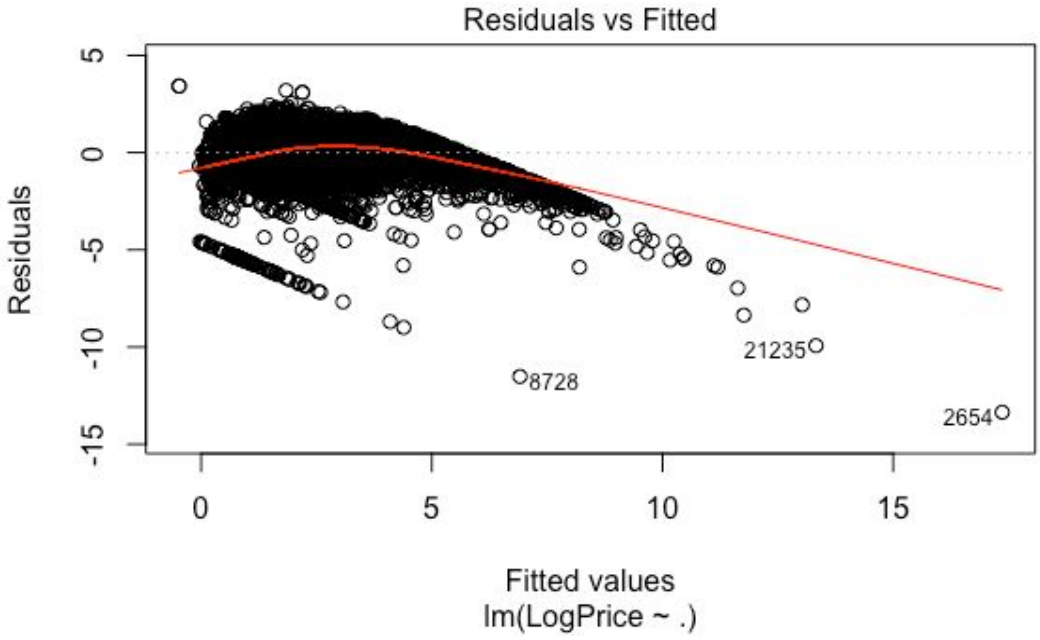
3. Multiple Linear Regression

The aim to build the multiple linear regression model was to predict the final selling price the item would be auctioned for. Before modeling, we wanted to test the response variable for skewness. We found the data to be highly skewed and hence performed a

Log transformation on the feature values before using it to build linear regression models. The baseline model after the transformation containing all the 23 feature variables resulted in a 82% correlation accuracy on the test dataset, an R^2 value of 0.7421 and a RMSE of 0.6261358.



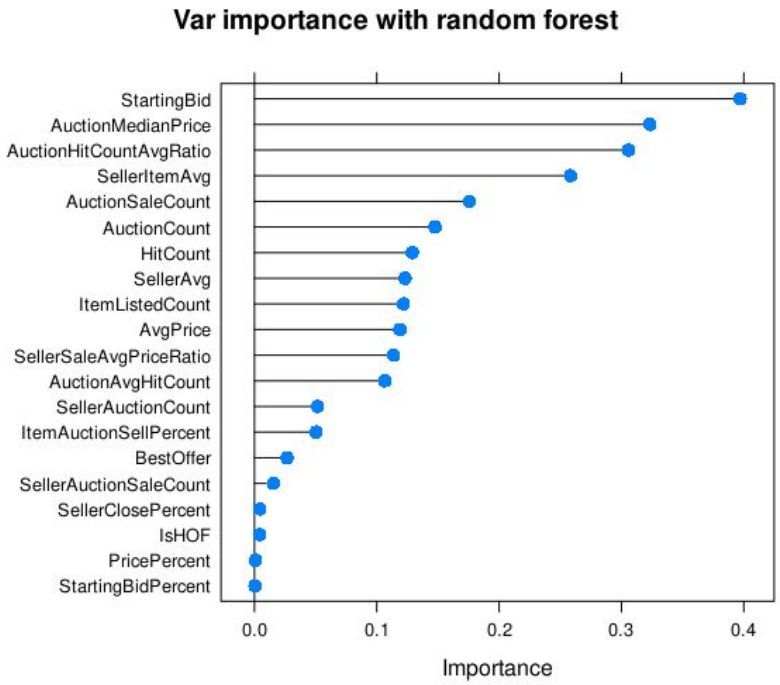
The residual plot for this model is as follows:



This plot clearly shows that the full model fit is not a good one and the value of R^2 and RMSE for this model are not reliable enough to assess the model fit.

3.1 Feature Selection

We Performed forward and backward stepwise regression to perform feature selection to identify the significant variables for our models. Both the tests resulted in no feature elimination, i.e, the resulting model is the same as the baseline multiple linear regression model. We next chose to evaluate variable importance in the dataset by using random forest and the following was the result:



The following features were chosen from the above plot and log transforms were performed to each price feature to get it to the same scale as the response variable and compare its performance with the baseline model:

- StartingBid
- AuctionMedianPrice
- AvgPrice
- ItemListedCount
- AuctionHitCountAvgRatio
- SellerSaleAvgPriceRatio
- SellerItemAvg
- AuctionCount
- AuctionSaleCount
- SellerAvg

Modeling with the selected features with a log transformation on the price resulted in a R^2 value of 0.621 and a RMSE value of 0.9640. This indicates that feature selection results in a bad model fit compared to when we model with the entire feature set. This is probably because of high correlations existing in the dataset.

3.2 Standardized Linear Regression

The cause for the poor behaviour of the above model fits could be associated with the differing points in our dataset, essentially skewing our results, such as entries with very high or low selling prices. Hence, model performance with standardized feature values was tested for a better fit.

Standardized the model using the following:

$$Y_i = \frac{1}{\sqrt{n-1}}\left(\frac{Y_i - \overline{Y}}{s_Y}\right)$$
$$X_{ik} = \frac{1}{\sqrt{n-1}}\left(\frac{X_{ik} - \overline{X}}{s_X}\right)$$

The resulting model fit had a R^2 value of 0.6614 and a RMSE value of 5.565879. This is clearly not an improvement on the above models.

3.3 Multiple Linear Regression Conclusion

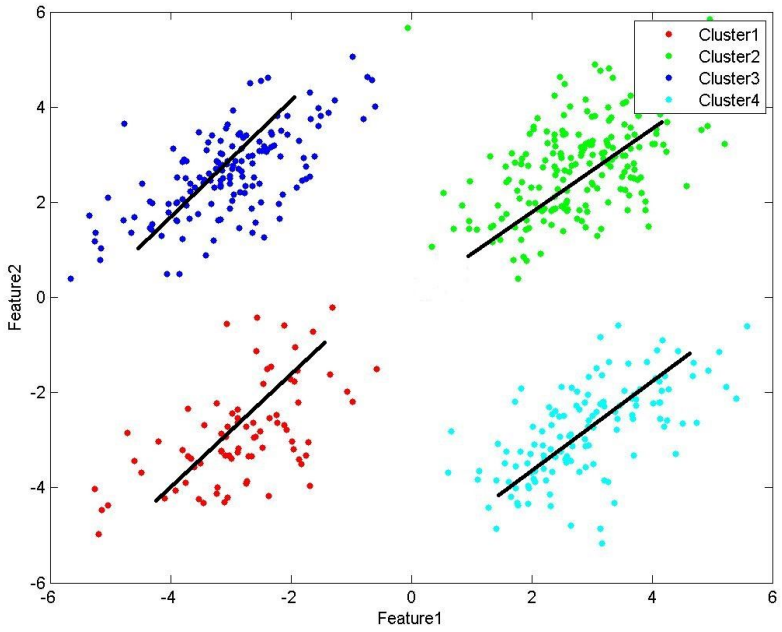
The following are the results form the various model tests:

Model	R-Square Value	RMSE Value of Log(Price)	RMSE Value of Original Price
Full Model	0.746	0.8569795	2.356034
Feature Selected Model	0.621	0.9640126	2.622197
Standardized Model	0.6614	5.565879	261.3548

We also could not perform a lack of fit F-test on the fitted models as our dataset does not include replicated groups of same levels of the predictor variables, so we must rely on the above values on the test set to determine the model fit. Since none of the above model fits performed satisfactorily, we tried to perform regression on clusters to see if that resulted in a better model fit to predict the selling price of an item on auction.

4. Regression on Clusters

The motivations for this part is that we wanted to see if we could improve our previous results by trying to find different clusters on our dataset that had predictable behaviours when considered separately.
Our goal is to implement a model similar to the one shown below, where, within the same dataset, there are different groups that, when taken as a whole do not display any linear characteristics, but have predictable behavior when considered separately.

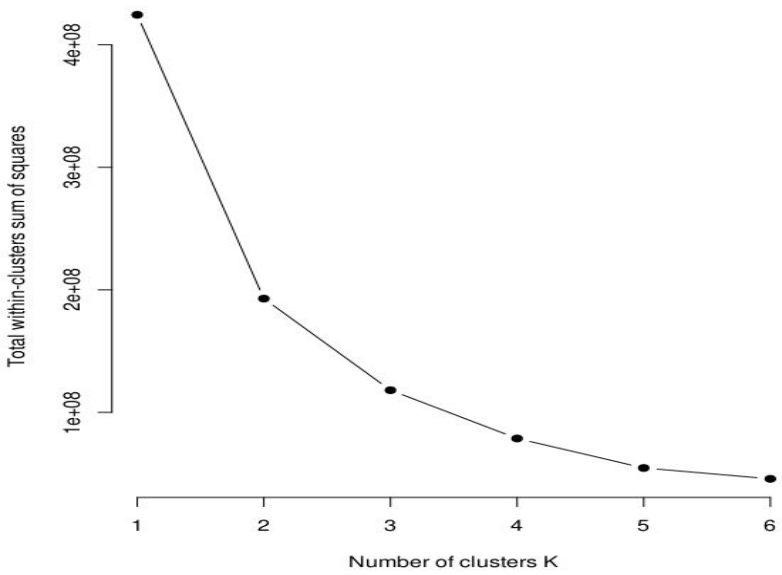


To do so, we first need to create our clusters. We want to have clusters based on the seller features. We decided to use *'AvgPrice'* and *'AuctionHitCountAvgRatio'* as features to build our clusters, because these are the two first features appearing in the "variable importance" plot that are directly related to the seller and not the item itself. Average Price is the average price for each items sold by the seller, so it would be beneficial to separate those who are selling expensive items from those who sell relatively less expensive items.). *'AuctionHitCountAvgRatio'* is a ratio that represents the number of hit counts (per auction) for one seller compared to hit counts from other sellers with similar items.

With these two features we should be able to capture sellers that have similar items, and hence, similar price behaviours.

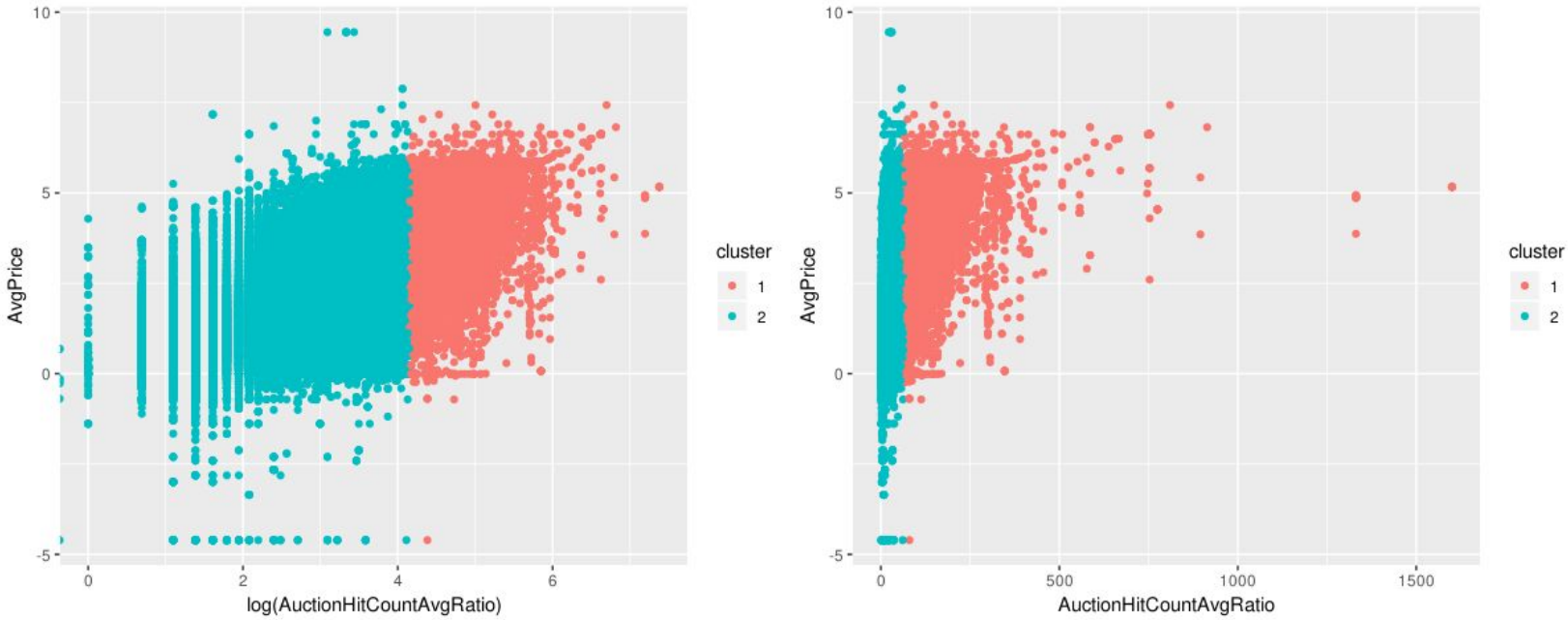
For the clustering algorithm we are choosing K-Means as we want to discover a pattern inside our dataset. However, we need to determine the number of clusters, k. We

created an elbow plot with the number of cluster against the total within-cluster sum of squares to determine the *'optimal'* K-value.



It would appear that K = 2 is a great value in our case (We are taking the K at the beginning of the elbow).

In this plot we can clearly see how our dataset has been split into two distinct clusters. (each color is a cluster).



This resulted in two clusters with sizes 39422 and 219166. This is a great result because we are trying to capture the auctions that are *'classic'* versus the ones that are more extreme.

We need to look at the average final price for each of those clusters to confirm that we have an accurate split of our dataset.

	Cluster 1	Cluster 2
Cluster size	39422	219166
Average observed price	78.62$	20.02$

From this table we can see that we have successfully separated *'extreme'* auctions from the *'normal'* ones. Overall we can say that there are two types of sellers: those who sell expensive items and those who sell inexpensive items. We believe that the behaviour for those two types of items are different but predictable when considered individually.

The goal now is to build two linear models within each clusters. We decided to use a forward stepwise feature selection to build them.

These are the results we have for the first cluster. We can see that the forward selection kept all the features for the model but no intercept.

```
Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
AuctionMedianPrice       6.985e-01  3.574e-03 195.448  < 2e-16 ***
StartingBid              6.231e-03  5.986e-05 104.106  < 2e-16 ***
SellerAvg                5.362e-01  8.598e-03  62.370  < 2e-16 ***
SellerSaleAvgPriceRatio  1.038e+00  2.687e-02  38.636  < 2e-16 ***
ItemListedCount          5.743e-02  7.724e-03   7.435 1.06e-13 ***
AuctionHitCountAvgRatio  1.767e-03  8.896e-05  19.860  < 2e-16 ***
SellerItemAvg           -1.386e-03  7.265e-05 -19.080  < 2e-16 ***
AuctionCount             5.353e-04  2.465e-05  21.710  < 2e-16 ***
AuctionSaleCount        -1.269e-03  6.129e-05 -20.700  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7006 on 39413 degrees of freedom
Multiple R-squared:  0.9706,    Adjusted R-squared:  0.9706
F-statistic: 1.447e+05 on 9 and 39413 DF,  p-value: < 2.2e-16
```

These are the results we have for the second cluster. We can see that the forward selection kept all the features for the model but no intercept, same results as for the first cluster.

```
Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
AuctionMedianPrice       4.968e-01  1.808e-03 274.770  < 2e-16 ***
SellerAvg                6.867e-01  2.903e-03 236.503  < 2e-16 ***
StartingBid              1.453e-02  6.189e-05 234.752  < 2e-16 ***
AuctionHitCountAvgRatio  2.498e-02  1.542e-04 162.007  < 2e-16 ***
SellerItemAvg           -1.163e-02  1.247e-04 -93.253  < 2e-16 ***
ItemListedCount          1.311e-01  4.699e-03  27.904  < 2e-16 ***
AuctionCount             1.311e-04  1.430e-05   9.165  < 2e-16 ***
AuctionSaleCount        -2.120e-04  4.013e-05  -5.283 1.27e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.726 on 219158 degrees of freedom
Multiple R-squared:  0.9179,    Adjusted R-squared:  0.9178
F-statistic: 3.061e+05 on 8 and 219158 DF,  p-value: < 2.2e-16
```

	Cluster 1	**Cluster 2**
Cluster size	39422	219166
R^2	0.9706	0.9179

We can clearly observe that the R^2 values for both clusters are much higher than the one we got from the regression without clusters. These are encouraging results for the upcoming test predictions comparaisons.

Our final step is to compare the RMSE values with the ones we get from the full model without clusters.

	RMSE(log Price)	**exp(RMSE(log Price))**	**RMSE(Price)**
Cluster 1	0.6857	1.9852	$17.29
Cluster 2	0.7628	2.144	$8.94
Weighted average	0.7528	2.123	$10.01
Full dataset	2.678	14.560	$16.00

With this table we can appreciate the results we get. We can very clearly observe that the predictive abilities of the model with the two clusters are much greater than that of the classic model.

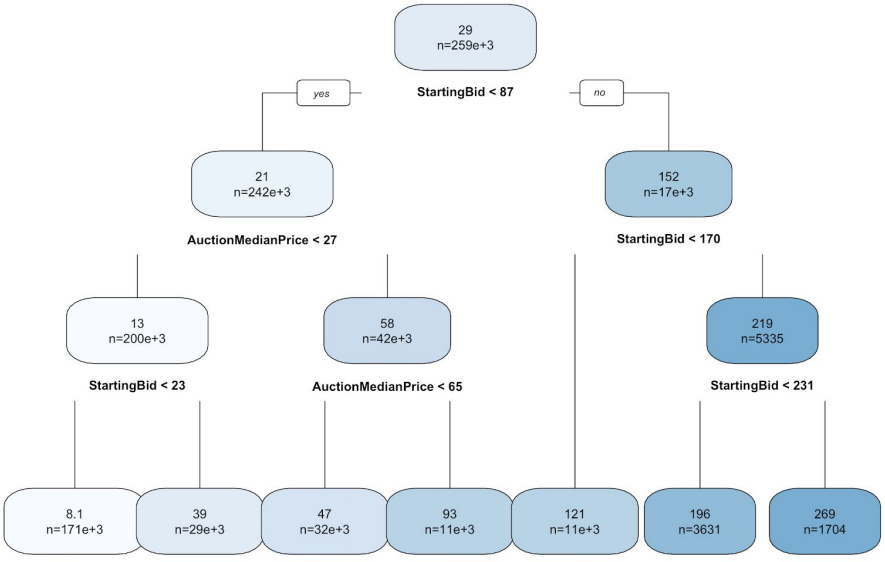
On average we have a difference of $10.00 between the price predicted and the actual price with the clustered regression model, whereas with the traditional model gives an average difference of $16.00.

To conclude, we did find some good results with the clusters compared to the classical regression model. These results were expected because when dealing with sales within a specific category (in our case: baseball sports items) it is common to have a clear distinction in the prices of items based on the popularity of the sports team or player involved, essentially leading to two clusters within our dataset, each with their own trends.

5. Classification and Regression Trees

Classification and regression trees are machine learning methods for constructing prediction models from data. The models are obtained by recursively partitioning the data space and fitting a simple prediction model within each partition. As a result, the partitioning can be represented graphically as a decision tree. Classification trees are designed for dependent variables that take a finite number of unordered values, with prediction error measured in terms of misclassification cost. Regression trees are for dependent variables that take continuous or ordered discrete values, with prediction error typically measured by the squared difference between the observed and predicted values. [2]

We created the following regression trees on our training set:



Predictions were then created on the test set to yield an RMSE of 18.523, albeit with a misclassification error rate of 83.79%. This led us to dismiss regression trees as a viable candidate on our dataset.

6. Conclusion

To conclude, Logistic Regression was found to be the best model for our classification problem, with an overall accuracy of 88.11%. The Logistic regression model outperforms its classification counterparts in Random Forest and LDA. Regression models for predicting the final selling price of an auction item was not as straightforward. Multiple Linear Regression models did not perform satisfactorily, which was found to be because of two distinct clusters within our dataset. An ensemble of clustering and regression was found to have higher predictive capabilities.

7. Data Source and Description

eBay Dataset: https://cims.nyu.edu/~munoz/data/
Data Description: eBay Auctions Dataset Description

8. Source Code

Source code for our project can be found on GitHub:

https://github.com/pranavlal30/EbayAuctions

9. Citations

[1] Jay Grossman, 'Predicting eBay Auction Sales': *http://jaygrossman.com*

[2] Wei-Yin Loh, 'Classification and Regression Trees': *https://doi.org/10.1002/widm.8*