# Predicting eBay Auction Prices with Regression

James Dumas
Luddy School of Informatics,
Computing, and Engineering
Indiana University
Bloomington, USA
jamdumas@iu.edu

Robert Kellems
Luddy School of Informatics,
Computing, and Engineering
Indiana University
Bloomington, USA
rkellems@iu.edu

Enora Marrec
College of Arts and Sciences,
Department of Biology
Indiana University
Bloomington, USA
emarrec@iu.edu

***Abstract*—Various details relating to finished eBay auctions (such as number of bidders, duration, and details regarding individual bids) are available on the website for some time after the end of the auction. For this paper we employ web scraping techniques in order to gather and organize these data for the purpose of using the various attributes of eBay auctions to predict their final sale prices. Through usage of principal component analysis and various forms of regression (e.g. simple linear, polynomial, weighted), we ultimately find that most attributes related to eBay auctions do not significantly correlate with final price and are thus only somewhat helpful in prediction tasks. We conclude that more fruitful results may be found by observing wider trends (e.g. popularity trends relating to certain items) on eBay and similar sites.**

***Keywords—data mining, web scraping, regression, e-commerce, prediction***

## NOMENCLATURE

| | |
|---|---|
| $I_p$ | Inertia |
| $n$ | Total number of data points |
| $p$ | Number of dimensions |
| $G$ | Barycenter of data distribution |
| $C$ | Correlation matrix |
| $X_{sc}$ | Normalized data matrix |
| $P$ | Eigenvectors matrix |
| $D$ | Eigenvalues matrix |

## I. INTRODUCTION

eBay is an e-commerce company known for its website, on which users can purchase a wide array of different products both new and used. The website is perhaps best known for its auction feature, which allows users to purchase/sell items in a manner similar to in-person auctions. To begin an auction, one user creates a listing for an item they would like to sell, which requires them to set an initial price, report the condition of the item (e.g. New, Used), and determine the duration of time for which they would like the auction to be open (e.g. 3 days). After the auction is opened, other users can place bids on the item, increasing its price with each successive bid; the user who has placed the most recent bid at the end of the auction period gets to buy the item.

When an auction ends on eBay, data such as starting price and duration, as well as information pertaining to each individual bid, remains available on the website for up to 30 days after the end of the auction. Various web scraping techniques can be used to gather and organize this information. With this project, we have taken data from some eBay auctions occurring in November 2021 in order to perform various regression tasks on the data; our main goal was to use various attributes relating to each individual auction item in order to predict its final selling price.

## II. DATA COLLECTION

### *A. Web Scraping*

For this project, we collected all the data from eBay's website directly, through the use of a web scraper. We limited the data collected by the scraper to only include auctions for smartphones with a condition of “good” or higher, to limit extraneous variables. The data scraped from the website for each item consisted of: auction duration, number of unique bidders, starting price, starting time, and a list of all bids on the item, with the price and time for each.

The scraper was written in Python, using the built-in requests library to send HTTP requests to the website, and the BeautifulSoup library to parse the HTML content of the web pages retrieved. The process it uses is as follows: 1) Retrieve the page listing auctions for smartphones with a condition of “good” or higher which have already ended. 2) Find a listing’s item ID, which is a unique number that eBay generates for each item listed on the website, and use it to generate the URL to the bid history page for the item. 3) Retrieve the bid history page and parse the relevant data out of the page. 4) Repeat from step 2 for each item on the page. 5) Once all items have been scraped and parsed, save the raw data into a JSON file.

There were numerous problems during the development of the scraper. Firstly, you must be logged in with an eBay account in order to view bid history information for auction listings. Secondly, you cannot allow the eBay website to suspect that you are making requests from a scraper, or it will utilize a CAPTCHA, like the “I’m not a robot” check you see on many websites. In this case, the scraper *is* in fact a robot, and so this is a serious concern. Lastly, some auctions’ bid history information contains data that doesn’t conform with the rest of the data—currencies other than USD, auctions ending before the duration has expired, and any other non-conforming data. The first two problems were easily solved by first logging into eBay manually in a web browser, then saving the HTTP cookies and the HTTP request headers used for requesting pages on the website, and allowing the scraper to use them. This way, it is indistinguishable from a human accessing the site through a browser, from the perspective of the website. The third problem was solved simply by rejecting the data from any auction listing that had inconsistencies, which did decrease the size of the data set, but not significantly so.

### *B. Organization/Normalization*

After a JSON file containing information for each auction was created, relevant attributes were extracted from the file using Python’s json library. These included the number of bidders who participated in the auction, the duration of the auction (in days), the initial price of the item (in US dollars), the date and time at which the auction began, and the price the item was at with each bid, along with the date and time of each of those bids. Since the dates and times included in the JSON file were text strings, they were converted to Python datetime objects for easier manipulation.

In order to store all of this information in an easily accessible manner, the data were put in a DataFrame object (part of Python’s pandas library), with each row pertaining to an auction which was auctioned. The columns contained the following attributes for each auction: final price (the response variable), number of bidders, duration, price difference mean (mean of the differences in price between each bid, including the initial price), price difference standard deviation, time difference mean (mean of the differences in datetime between each bid, including the date and time at which the auction started), and time difference standard deviation.

## III. PREDICTION TASKS

### *A. Principal Component Analysis*

A Principal Component Analysis is an exploratory statistical method used to reduce the dimensionality of a dataset and describe it.

*1) Information redundancy:* A PCA consists in transforming the $d$ initial correlated variables into two decorrelated Principal Components (PC) that represent as much of the initial variance as possible. It is a strong model that does not rely on strong assumptions such as variable normality. While high variance is often regarded as a sign of a poor statistical model, in PCA the larger the variance the larger the amount of information the variable contains [1]. Moreover, since the initial variables are correlated, less variables can be used to describe an equivalent quantity of information. With only two or three axes and a basis transformation, redundancy can be eliminated and the great majority of the total information can be captured.

*2) PCA limitations:* PCA is limited to quantitative variables. If our variables were categorical (qualitative), we could have performed a Multiple Correspondence Analysis (MCA) [2]. PCA also requires the data to be centered and optionally reduced. We performed a normalized PCA.

*3) Inertia:* Inertia can be defined as a generalization of variance to a multivariate space of $p$ dimensions. It is the sum of the distance between each data point $i$ and the barycenter $G$ of the cloud of $n$ data points.

$$I_p = \frac{1}{n}\sum_{i=1}^{n} d^2(i,G)$$

*Fig 1: Generalized variance formula using Euclidean distance*

The data needs to be centered for the barycenter $G$ to be placed at the origin $O$.

*3) Correlation matrix:* Centered but unreduced PCA uses the variance-covariance matrix, $S$. Normalized PCA uses the correlation matrix $C$. Both are symmetric matrices: $C = C^{T}$. Let $X_{sc}$ be the normalized data matrix of $n$ data points. The correlation matrix $C$ is:

$$C = \frac{1}{n} \cdot X_{sc}^{T} \cdot X_{sc}$$

*Fig 2: Correlation matrix formula*

The correlation matrix must be diagonalized to obtain the eigenvalues matrix $D$ and the eigenvectors matrix $P$. The eigenvectors give the direction of the two axes of the new basis, and the eigenvalues give the inertia associated with each axis. We choose two axes so that the first axis maximizes inertia, and the second axis is orthogonal to the first and maximizes inertia as well. A graphic method can be used to determine the number of PCs, the "elbow method":

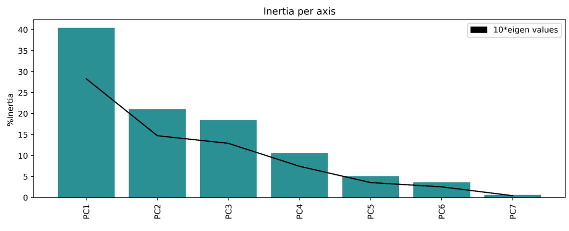

*Fig 3: Scree diagram: a steep curve (black), ideally followed by a bend and a line. We choose the PCs before the tight bend. Here, PC1 and PC2.Eigenvalues have been multiplied tenfold to increase lisibility.*

*4) New basis and Variable Factor Map:* The points are projected into the new basis by matrix product with the eigenvector matrix *P:*

$$T = X_{sc}.P$$

The $T$ matrix can be used for subsequent statistical analysis that require orthogonal variables or a reduced number of variables. The linear relationship between variables can be explored graphically with a Variable Factor Map, also known as a circle of correlations. The variables are projected on a unit circle: the length of each vector corresponds to its weight in the PCA model. The angle between variables shows their correlation, and the angle between a variable and the horizontal (PC1) and vertical (PC2) axis corresponds to their correlation coefficient as well [3]. PCA is not suitable for feature selection, but provides a good insight into the linear relationships between variables.

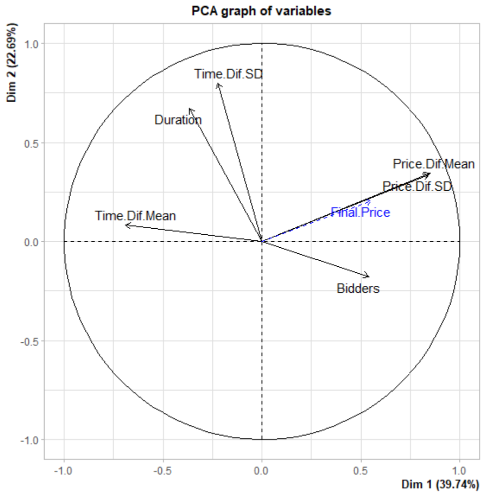

*Fig 4: Variable Factor Map.*

The first PC represents approximately 40% of the total inertia, while the second PC represents approximately 20%. This graph shows that *Duration* and *Time Dif SD* are strongly correlated. *Price Dif Mean* and *Price Dif SD* are also strongly correlated, as expected. The variables that represent the most inertia are *Time Dif SD* and *Price Dif Mean*. This graph can be coupled with an Individual Map, which is the projection of data points in the new basis.

### *B. Linear Regression*

We performed simple linear regression using each of the attributes in the dataset to predict final price, calculating the training mean square error and R-squared value for each. The results are as follows:

**Number of Bidders**

Coefficient: 2.22528495, Intercept: 432.10216649
Mean squared error: 166579.3703287211
R-squared: 0.00116382899746037

**Duration**

Coefficient: -31.86584903, Intercept: 639.74472028
Mean squared error: 161436.53202753922
R-squared: 0.032001098382624105

**Price Difference Mean**

Coefficient: 33.32666873, Intercept: 183.02027777
Mean squared error: 63720.54132328699
R-squared: 0.6179215866648803

**Price Difference Standard Deviation**

Coefficient: 22.72533748, Intercept: 265.92289971
Mean squared error: 100013.12219683222
R-squared: 0.40030554907272853

**Time Difference Mean**

Coefficient:-5.96954802e-05,
Intercept: 455.99891075
Mean squared error: 166680.7992671138
R-squared: 0.0005556450893743348

**Time Difference Standard Deviation**

Coefficient: -0.00043503, Intercept: 484.94792292
Mean squared error: 165211.18816499715
R-squared: 0.00936766498838304

Number of bidders, duration, and time difference mean/standard deviation all seem to perform very poorly when used as predictor variables, with large mean squared error values and very small R-squared values. Price difference mean/standard deviation fare somewhat better, producing mean squared error values that are noticeably smaller than those of the other attributes, along with R-squared values which are noticeably larger. With an R-squared value of roughly 0.62, price difference mean appears to be the most helpful attribute for predicting final price, although it is still far from perfect. Intuitively, this result is to be expected; if the prices for an item are being raised by large increments on average, then it makes sense that the final price would be high as a result.

### *C. Polynomial Regression*

We performed a polynomial regression using all the attributes from the dataset and a Mean Square Error metric. The Adjusted R-Squared is 0.007575. Polynomial regression performs poorly because of the nature of our data: auction prices can only increase over time, thus polynomial functions are not suitable for our task.

### *D. Multiple Regression*

Multiple Linear Regression relies on several assumptions:

1. Existing linear relationship between the predictor variables and the predicted variable
2. Independence of all observations
3. Homoscedasticity
4. Multivariate Normality

To test hypothesis (1), can be verified by plotting a simple *y* against *x* plot. If the distribution follows a normal distribution, the assumption can be considered as met. To test for assumption (2), independence , we performed a Durbin-Watson test, which is used to detect autocorrelation in a linear regression [4]. Assumption (3) can be tested by plotting the fitted regression against its residual values. Inconsistent variance means that patterns will be visible in the distribution. Multivariate normality, assumption (4), can be verified by plotting a QQplot. Some variables did not meet these assumptions: multiple regression with the variables *Bidders*, *Duration*, *Price.Dif.Mean* results in a R-squared of 0.6259. However, the variables are auto-correlated with a p-value of 0.018 (< 0.05), homoscedasticity is not met, and the data is not normally distributed.

On the opposite, the variables *Duration, Price.Dif Mean, Price.Dif.SD, Time-Dif.Mean, Time.Dif.SD* have the best R-squared ( $r^2 = 0.7492$). The model is close to a normal distribution:

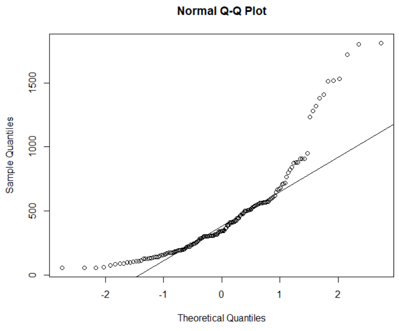

*Fig 5: QQplot of the regression model with the variables Duration, Price.Dif.Mean, Price.Dif.SD, Time-Dif.Mean, Time.Dif.SD. The model does not follow a normal distribution..*

However, homoscedasticity is not met because patterns are visible in the following plot:

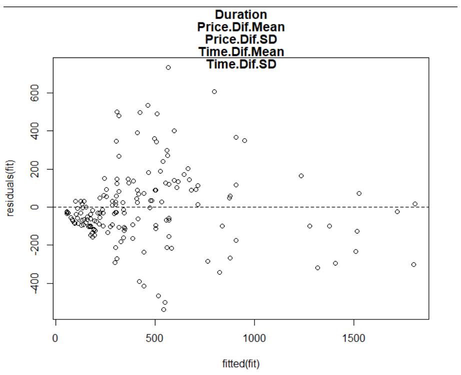

*Fig 6: Standardized values vs. Residual values of the regression model*

### *E. Weighted regression*

To solve the problem of heteroscedasticity, we applied a Weighted regression method. To solve the problem of normal distribution, we applied a *log()* transformation to the data.

With the same variables, a normal distribution is obtained:

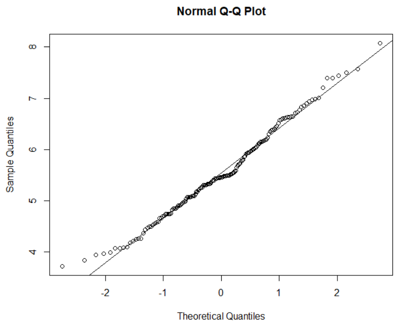

*Fig 7: QQplot of the weighted regression model with variables Duration, Price.Dif.Mean, Price.Dif.SD, Time-Dif.Mean, Time.Dif.SD. The model does not follow a normal distribution..*

However, the model trades assumption verification for a poorer performance The R-squared is slightly lower than simple multiple regression ($r^2$ = 0.6498). Similarly to linear regression, price difference mean (*Price.Dif.Mean)* and price difference standard deviation (*PRice.Dif.SD)* seem to be the most critical attributes for our regression models.

## IV. CONCLUSIONS

After experimenting with simple linear, polynomial, and weighted regression, it seems clear that the attributes which we chose to predict final price are not very helpful for these tasks. Ultimately, weighted regression utilizing all attributes except number of bidders gave us the most success (R-squared: 0.6498), with simple linear regression using mean price difference between individual bids falling closely behind (R-squared: about 0.6179). Although these numbers indicate that certainly some correlation exists, it is unlikely that they would be very reliable in classifying new data. Although we collected most of the possible attributes for individual bids, maybe more interesting results could be found in other areas of online shopping and auctions; for example, one could use regression to predict the popularity of certain items for sale, or predict the popularity of certain sellers using data relating to their profile.

## INDIVIDUAL CONTRIBUTIONS

**James**- wrote the code pertaining to section IIA, general help with writing the paper/preparing the presentation
**Robert**-wrote the code pertaining to section IIB and parts of IIIB, general help with writing the paper/preparing the presentation
**Enora-**wrote the code pertaining to section IIIA and parts of IIB, general help with writing the paper/preparing the presentation
We were supposed to have a 4th group member as well (Harry Kim), but we were unable to contact him and thus he did not contribute to the project.

# REFERENCES

[1] Delgado D. “Understanding Principal Component Analysis Once And For All”. Medium.com. https://medium.com/bluekiri/understanding-principal-component-analysis-once-and-for-all-9f75e7b33635 (accessed December 11, 2021)

[2] Fermin A. “Analyse en Composantes Principales (ACP) et Classification Non supervisée”. http://fermin.perso.math.cnrs.fr/Files/ACP-AFM.html (accessed December 11, 2021)

[3] Unknown Author. “Principal Components Analysis”. Factominer.free.fr. http://factominer.free.fr/factomethods/principal-components-analysis.html (accessed December 13, 2021)

[4] Hateka, N. *Principles of Econometrics: An Introduction (Using R).* India: SAGE Publications, November 10, 2010. pp. 379–82.