California State University - East Bay

From Specs to Price Tags: Modeling Laptop Value

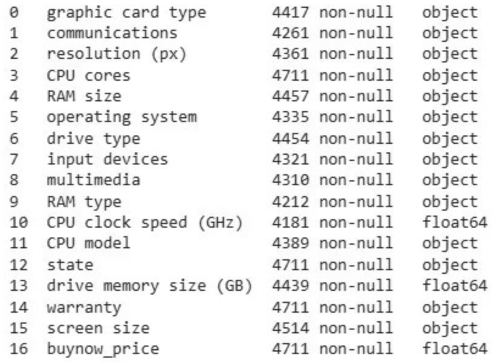
Yaron Lidgi
STAT 632 - Linear and Logistic Regression
Dr. Joshua Kerr
May 14, 2025

Introduction

Laptops vary greatly in specifications and performance. Our research involved utilizing regression analysis to make price predictions based on various laptop characteristics. After exploratory data analysis, which included substantial cleaning and preprocessing, we began exploring the impact of the predictors. Our goal was to develop a model with high predictive accuracy that may provide actionable insights in marketing efforts.

Data Description

The source of this dataset is the website StrataScratch.com, a learning platform geared towards individuals interested in pursuing careers in data science. Our decision to proceed with the project chosen was partly based on its being rated as hard in difficulty level. This particular project was originally obtained from Allegro, a Polish E-commerce company, which used it as a take home interview for potential employees. The dataset includes 16 predictors and a target variable of price (measured in Polish złoty), for 4711 laptop observations. The predictor variables cover a range of hardware and software features, among which are RAM size and type, CPU model and clock speed, screen resolution, graphics card type, storage capacity, operating system, and screen size (see *Figure.1*).



```
 0   graphic card type       4417 non-null   object
 1   communications          4261 non-null   object
 2   resolution (px)         4361 non-null   object
 3   CPU cores               4711 non-null   object
 4   RAM size                4457 non-null   object
 5   operating system        4335 non-null   object
 6   drive type              4454 non-null   object
 7   input devices           4321 non-null   object
 8   multimedia              4310 non-null   object
 9   RAM type                4212 non-null   object
 10  CPU clock speed (GHz)   4181 non-null   float64
 11  CPU model               4389 non-null   object
 12  state                   4711 non-null   object
 13  drive memory size (GB)  4439 non-null   float64
 14  warranty                4711 non-null   object
 15  screen size             4514 non-null   object
 16  buynow_price            4711 non-null   float64
```

Figure.1: Names of all variables

Data Cleaning and Preprocessing

The first course of action was to eliminate the state, warranty, and communication columns, as they seemed non-informative and too complex to incorporate sensibly into our model. The dataset also included multiple columns with cell entries including lists and extra characters such as brackets, commas, or quotes, which were dealt with appropriately. Some examples include drive_size which needed the GB unit removed, to be converted into a numerical variable. Also, screen_resolution was listed as an ‘X’ by ‘Y’ dimension set, and was split into x and y values into two unique columns and converted into numerical. For screen size values were originally input as a range, for example 15”-15.9”. They were subsequently converted to the floor value for simplicity and converted to numerical. The dataset also contained much whitespace and NA cell entries, as well. After the above preprocessing and cleaning, the R code for removing observations with missing values resulted in the removal of over 1000 observations. This called for a concern of whether the remaining data would indeed be an unbiased sample. Inputting CPU clock speed column would save over 500 of these observations and was decided to be an appropriate course of action to sustain a random sample.

Predictors with extremely low variance were removed, reducing total variables from 64 to 45. A stepwise selection method was then applied to the additive model, further narrowing the set to 30 predictors with no loss in adjusted R², as compared to the model with 64 predictors.

Model Development and Selection

We started the modeling process with a simple, full additive model containing all predictors. We proceeded to evaluate model assumptions of constant variance and normality of the residuals based on diagnostic plots. The QQ plot (*see Figure.2)* was severely flared upwards at the right tail with several outlier points trailing even further above the normality reference line, showing a lack of normality. The residuals vs fitted values scatterplot (*see Figure.3*) revealed non-constant variance as well, fanning outwards towards the right of the plot. To correct these issues, we used a Box- Cox transformation to obtain an optimal lambda = 0, and a log transformation on the response variable. The transformed model showed a substantial improvement in meeting assumptions, by examining the plots (*see Figure.4 and Figure.5*), and this was confirmed by the Shapiro-Wilk test with W = 0.95737, exceeding the rule of thumb 0.95 threshold.

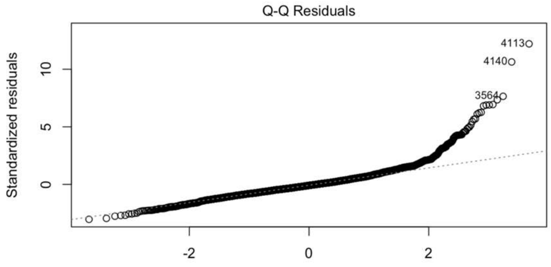


Figure.2

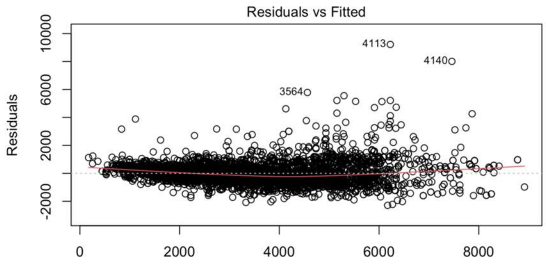


Figure.3

Figure.2: QQ plot for lm0 shows deviation from normality, especially at the upper tail.
Figure.3: Residuals vs. fitted plot reveals non-constant variance and non-linearity.

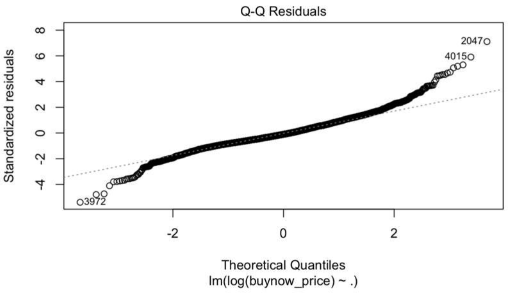


Figure.4

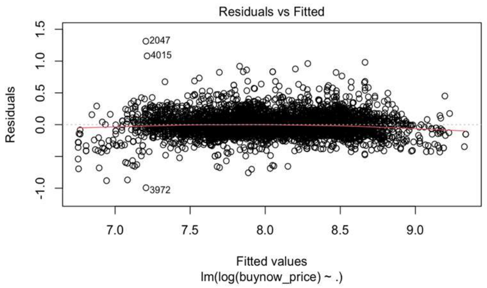


Figure.5

Figure.4: QQ plot after log transformation shows improved normality, with fewer deviations.
Figure.5: Residuals vs. fitted values plot after log transformation displays more consistent variance and linearity.

Dealing with Multicollinearity

Multicollinearity happens when the information the predictor provides is redundant, and its occurrence can distort coefficient estimates and inflate standard errors. Inspection for multicollinearity was through the use of the Variance Inflation Factor values. The two variables showing high VIF were resolution2 and CPU_cores_2, with values of 235.9 and 23.3, respectively. This exceeded the rule of thumb threshold of 5, and the predictors were subsequently removed.

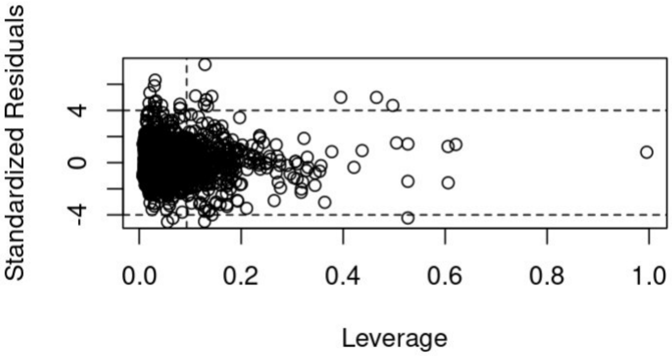
Interaction Terms

In order to explore whether or not the interactions between predictor variables can enhance model performance, we fit a model that includes all pairwise interactions. The interaction terms created resulted in a large quantity of predictors of 435. Due to the quantity of predictors it seemed most appropriate to utilize R's step function to automatically remove insignificant predictors and produce an optimal model. As the amount of compute needed to run this exceeded the capability of our computers and consequently resulted in a crash, an alternative course of action was called for. Examining the summary output for the model, it was discovered many of the pvalues for the coefficients were quite high. An alternative that seemed viable would be to remove predictors with pvalues higher than 0.5 in batches of various sizes. This seemed proper as, although we learned this semester whenever one variable is removed the pvalues of remaining variables change, a pvalue for a coefficient of .5 or higher would likely still remain above 0.05. After dropping 140 predictors in this manner, with a reduced number of predictors, the step function was finally run with success.

We also explored potential non-linear relationships by evaluating polynomial terms. Of the five 2nd degree polynomial terms tested, two presented as statistically significant, screen size and RAM size. The final model included all main effects as well as 178 interactions and 2 squared terms. Although this increased the model complexity and might have decreased interpretability, it significantly improved prediction performance.

Outlier Analysis

We inspected the data for outliers based on standardized residuals as well as leverage values. Being the data set was quite large, a threshold value for standard deviation of residuals of 4, seemed appropriate, and was consequently utilized. Additionally, the rule of thumb of 2(p + 1)/n was the threshold used for leverage. The Standardized Residuals vs. Leverage plot was used to visualize these observations (see *Figure.6*). Although these observations were considered as influential outliers, we chose not to remove them from the train dataset, as we did not have sufficient laptop expertise to determine whether they were errors.

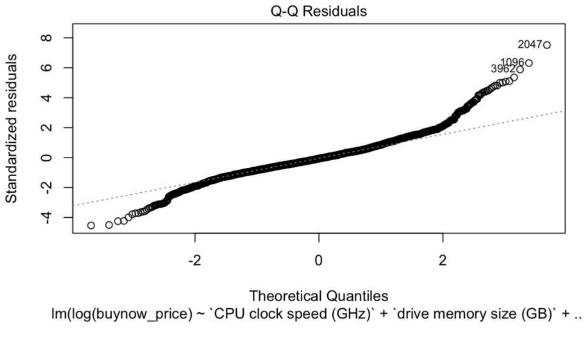


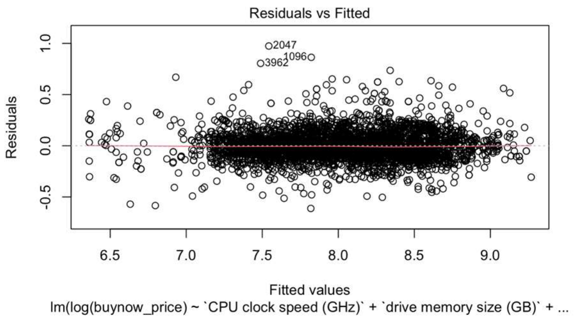
This plot helps identify influential outliers by displaying standardized residuals against leverage. Points beyond the dashed lines in the top and bottom right areas are classified as influential outliers. These are observations: 1344, 1884, 2047, 2222, 2674, 2735, 2774, 3622, 3972, 4015, 4140, and 4158.

Figure.6: Standardized Residuals vs. Leverage plot

Model Evaluation

After we fit the final model, we used diagnostic plots and statistical tests to confirm that model assumptions remained satisfied. The QQ plot (*Figure.7*) and residuals vs. fitted values plot (*Figure.8*) seemed similar to our earlier model, indicating assumptions were indeed satisfied. The Shapiro-Wilk test confirmed returning a W of 0.95671.





Performance was measured using the test dataset, also provided by Allegro. The final model achieved an adjusted R² of 0.849, which was reasonable performance although lower than the 0.92 adjusted R² from the train dataset. The Root Mean Square Error (RMSE) was 799 (roughly $200 in USD). While the large number of interactions increased model complexity and reduced interpretability, they significantly improved model performance, as measured by the adjusted R^2.

Limitations and Future Work

We were limited by insufficient computer power when running code. We also realize, by inspecting described components, that the dataset may be dated. Future work might involve analysis with more current and perhaps more relevant domestic data.