5/12/2024

Laptop Specs and Market Prices

Data Analysis and Statistical Modelling

***University of Technology Sydney***

***School of Transdisciplinary Innovation***

**Statistical Thinking for Data Science: 36103**

*Group 2*

Benedict Brunker

Britney Odria

Praise Meli

Qazi Fahim

Cameron Burdass

Faith Jepkemboi Chepkwony

..

# Executive Summary

* Global revenue in the laptop market is projected to reach US$147.4bn by 2028.
* There is potential for huge revenues, but new companies face steep barriers to entry:
  + High input costs
  + Supply chain risk
  + Dominance of the Big 10 large firms
* Statistical modelling can give new companies a better idea of a laptop’s market price given its specs.
* Our project:
  + Analyzes relationships between laptop specs and price, revealing significant correlations.
  + Builds a statistical model to predict a likely market price-point for a laptop based on its specs.
* Our model:
  + Predicts price at €268 mean error.
  + Performs well for low-mid market products.
  + Should remain robust into the future, given a regular annual retraining schedule to account for new technological advances and market conditions.

# Table of Contents

Contents

[Executive Summary 1](#_Toc166447877)

[Table of Contents 2](#_Toc166447878)

[1. Introduction 3](#_Toc166447879)

[2. Methods 4](#_Toc166447880)

[2.1 Sourcing Data and Dataset Description 4](#_Toc166447881)

[2.2 Exploratory Data Analysis 5](#_Toc166447882)

[2.3 Preprocessing and further analysis 10](#_Toc166447883)

[2.4 Modelling 12](#_Toc166447884)

[2.5 Data Splitting 12](#_Toc166447885)

[2.6 Feature Engineering **Error! Bookmark not defined.**](#_Toc166447886)

[3. Results 15](#_Toc166447887)

[3.1 Parametric Models 15](#_Toc166447888)

[3.2 Non-Parametric Models 18](#_Toc166447889)

[4. Discussion 20](#_Toc166447890)

[5. Conclusion 22](#_Toc166447891)

[References 23](#_Toc166447892)

[Appendix A 24](#_Toc166447893)

## Introduction

A green squares with black text

Description automatically generatedGlobal revenue in the laptop-computer market is projected to reach US$147.4bn by 2028 (Statista, 2024); huge profits can be realized in this sector. However, would-be market entrants face steep barriers: high input costs, supply chain risk (Sweeney, 2021) and a market dominated by a handful of large firms (see *Figure 1*).

Figure 1

This project aims to assist new market entrants by designing regression models to predict an average price-point for a laptop based on its specifications. This will allow new producers to assess the expected profitability of their products by predicting a price that can be assessed against the input costs of some given specifications. Our hope is that such a model can remain robust into the future, since computational advances and their relationship with price can be captured by the specification data we use as modelling features.

## Methods

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table 1  *Variable Description* | | | | |
| *Columns* | Description | Mode (median) value | No. unique values | Data-Type |
| *Company* | Producer | Dell | 19 | object |
| *Product* | Name of laptop | XPS 13 | 618 | object |
| *TypeName* | Laptop Type | Notebook | 6 | object |
| *Inches* | Screen size | 15.6 | 18 | float64 |
| *ScreenResolution* | Screen resolution description | Full HD 1920x1080 | 40 | object |
| *Cpu* | Model of Central Processing Unit | Intel Core i5 7200U 2.5GHz | 118 | object |
| *Ram* | GigaBytes available for Random Access Memory | 8GB | 9 | object |
| *Memory* | Drive(s) storage description and amount | 256GB SSD | 39 | object |
| *Gpu* | Model of Graphics Processing Unit | Intel HD Graphics 620 | 110 | object |
| *OpSys* | Operating System | Windows 10 | 9 | object |
| *Weight* | Weight of laptop in kilograms | 2.2kg | 179 | object |
| *Price\_euros* | Recommended Retail Price in Euros | (977.0) | 791 | float64 |

### 2.1 Sourcing Data and Dataset Description

Our dataset is sourced by Kaggle and comprises 1,303 observations across 12 variables. Our response variable is ‘Price\_euros’ and the remaining 11 are potential predictors. There were no missing values or duplicate entries. *Table 1* summarizes these variables.

After preparing the dataset for analysis, our focused exploration aims to determine how key laptop specifications might influence price.

### 2.2 Exploratory Data Analysis

A graph of a distribution of computer prices

Description automatically generatedFigure 2

Figure 1 shows that the distribution of laptop prices is right-skewed, with a higher concentration in the lower to mid-range segments. This indicates that most consumers prefer more affordable laptops, while premium, higher-priced models are less common, suggesting a niche market for these segments. The distribution is abnormal.

Figure 3

A comparison of a graph

Description automatically generated with medium confidence

Intel Central Processing Units (CPUs) span a broad price range, particularly in mid-range to premium segments, with the highest median price. AMD targets more affordable options with lower median prices and a tighter price distribution, focusing on budget to mid-range consumers. The dataset contains only one CPU produced by Samsung.

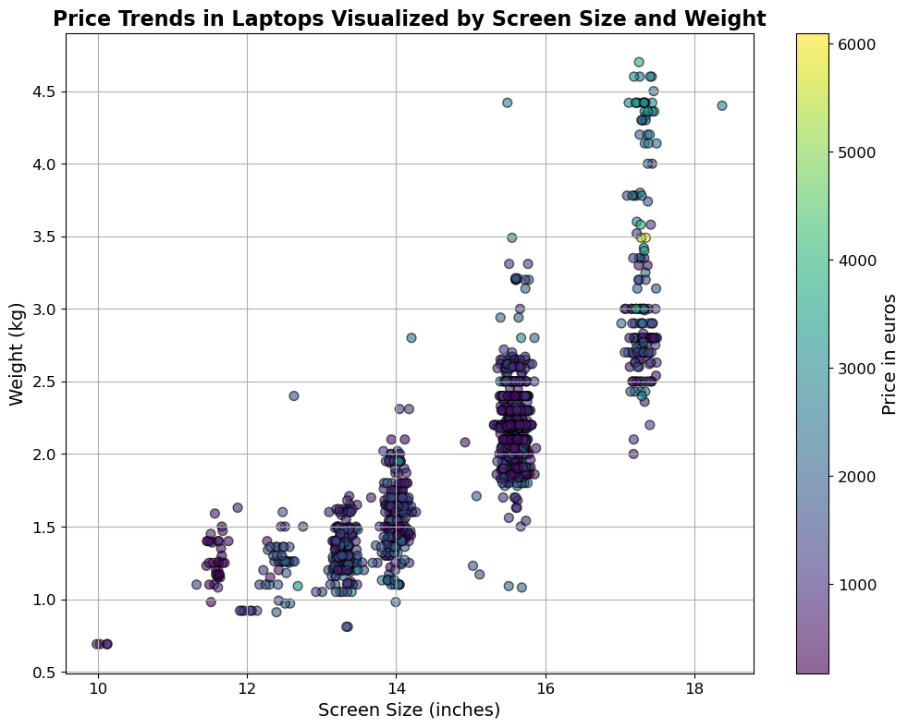
*Figure 3* shows most laptops weigh between 1.5 to 2.5 kg with screen sizes of 14 to 16 inches. Notably, the most expensive laptops typically weigh around 2 kg and have 15-inch screens, suggesting consumers value a balance of portability and screen size.

Figure 4

A chart of different colored squares

Description automatically generatedWorkstations, designed for professional use with powerful specifications, have the highest mean price. Gaming laptops also command high prices due to advanced graphics and performance features. Ultrabooks occupy the mid-range, balancing performance and portability, while notebooks and netbooks are the most affordable, targeting budget-conscious buyers. Convertibles show a broad price range, emphasising their versatility.

Figure 4

A graph of different colored bars

Description automatically generated  
  
*Figure 5* shows a highly linear relationship between RAM and price, suggesting the utility of this variable as a feature in predictive modelling.

Figure 5

*A diagram of a number of different colored squares

Description automatically generated with medium confidenceFigure 6* shows that RAM and CPU frequency positively correlate with price. Memory storage anti-correlates slightly, suggesting more storage doesn't necessarily mean higher prices. Screen resolution (Pixels) has a moderate positive correlation, while screen size shows a very weak correlation with price.

Figure 6

### 2.3 Preprocessing and further analysis

Ten of the eleven potential predictor variables consisted of text-data which had to be processed into numbers interpretable by regression models; *Appendix A* summarizes this processing.

The complexity in processing the ‘Cpu’ column resulted in a large number of new covariant variables (see *Figure 7*). We dealt with this by using Principal Components Analysis (PCA) on these new variables (see *Table 3).*

|  |  |  |
| --- | --- | --- |
| Table 3  *Results of Principal Components Analysis of CPU specifications* | | |
| *Principal Component* | **Explained Variance (%)** | **Cumulative Explained Variance (%)** |
| *PC1* | 41.55 | 41.55 |
| *PC2* | 33.95 | 75.5 |
| *PC3* | 9.95 | 85.45 |
| *PC4* | 6.48 | 91.93 |
| *PC5* | 4.17 | 96.10 |
| *PC6* | 3.25 | 99.35 |
| *PC7* | 0.52 | 99.87 |
| *PC8* | 0.13 | 100.0 |

A colorful squares with text

Description automatically generated with medium confidence

**Figure 7**

Analyzing these Prinicpal Components (PCs) led us to drop PCs 6-8, which appeared less distinctive and explained less than 4% of the total variance together. By analyzing these Principal Components we also made an informed guess as to what they might represent, and renamed the columns accordingly (see *Table 4*).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ­Table 4  *Feature Variance Captured by Principal Components* | | | | | |
| *Features* | **PC1** | **PC2** | **PC3** | **PC4** | **PC5** |
| *Cpu\_Intel* | -0.1754 | -0.6472 | -0.1659 | -0.117 | 0.0203 |
| *Cpu\_AMD* | 0.1729 | 0.6427 | 0.1654 | 0.1172 | -0.0143 |
| *Cpu\_frequency* | -0.3905 | 0.2433 | -0.3232 | 0.3032 | 0.5177 |
| *Cpu\_model\_no.* | -0.2594 | 0.2559 | 0.0945 | -0.8252 | -0.1593 |
| *Cpu\_performance* | -0.2743 | 0.1217 | -0.4327 | 0.1188 | -0.2183 |
| *Cpu\_efficiency* | 0.3455 | -0.1649 | 0.5045 | 0.1212 | 0.2366 |
| *Cpu\_family* | -0.6178 | -0.031 | 0.5298 | -0.0172 | 0.3753 |
| *Price\_euros* | -0.3785 | -0.021 | 0.3305 | 0.4131 | -0.6793 |
| *Renamed as* Cpu\_ | **efficient\_low\_end** | **AMD\_mid\_range** | **efficient\_high\_end** | **fast\_high\_end** | **fast\_low\_end** |

A graph with a red line

Description automatically generated

Logarithmic transformation of the target variable *Price\_euros,* along with dropping extreme outliers,failed to yield a normal distribution, as demonstrated in *Figure 8;* the p-value for the normality test was 8x10-10 (Appendix B [3.1]).

Figure 8

### 2.4 Modelling

Our modelling approaches can be grouped into:

* + - 1. Non-parametric tree-based models, including:
         1. Simple Decision Trees
         2. Random Forest ensembles
      2. Parametric Linear models, including:
         1. Ordinary Least Squares (OLS)
         2. Lasso
         3. Ridge
         4. ElasticNet

### Data Splitting and Feature Engineering

We split our dataset between training, validation and test data by an 80:20:20 ratio.

We took two complementary approaches to feature selection.

#### 2.5.1 The “levelling up” approach

Based on our EDA (cf. §2), we selected a narrow range of features (labelled *X\_basic)* to fit to OLS models, then analyzed coefficient weights and t-statistics for those features. Our first OLS model was fitted with ‘Inches’, ‘Ram’ and ‘Weight’. *Table 5* displays some summary statistics for those features, along with a ‘Constant’ column which predicts the intercept term. We can see for instance that ‘Ram’ and ‘Inches’ are almost certainly predictive of price given their t-statistics, while the predictive power of ‘Weight’ is less certain.

**Table 5**

*Feature Statistics for OLS*

***Feature* coef std err t P>|t| 0.025 0.975**

------------------------------------------------------------------------------

*Constant* 803.8037 61.757 13.016 0.000 682.584 925.024

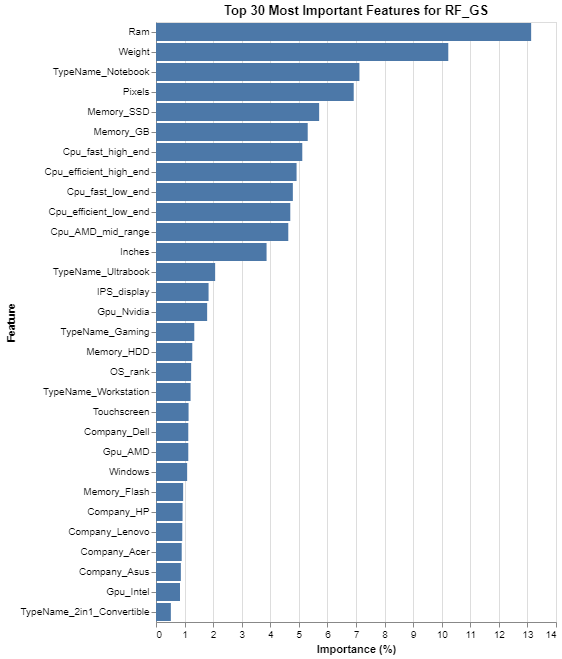
*Inches*  -615.0228 160.404 -3.834 0.000 -929.871 -300.175

*Ram*  6064.6834 212.610 28.525 0.000 5647.362 6482.005

*Weight* 160.6220 174.585 0.920 0.358 -182.062 503.306

#### 2.6.2 The “bulk and shred” approach

Figure 9



The opposite approach is to start with many features and progressively winnow them down. Our RandomForest (RF) models were useful in this approach for two reasons:

1. Each tree in the forest can be limited to observing only a specified number of features. For example, our best performing RF limited each tree to observing only five features.
2. RFs have a *feature\_importances\_* attribute which allows us to learn how important the given features were for the model in making predictions. By way of illustration, see *Figure 9*, which shows feature importances for our best performing RF model. We can see for example that the analysis corroborates both the strong coefficient weight assigned to ‘Ram’ by the OLS model, and the correlation discovered by EDA. On the other hand, the high degree of importance assigned to ‘Weight’ contradicts the OLS’ relative skepticism about this feature.

Our regularizing models – Lasso, Ridge and ElasticNet – also accommodate this approach, since they automatically increase model sparsity by applying penalties to coefficients.

## Results

### Parametric Models

For the OLS model mentioned above, the p-values for Omnibus and Jarque-Bera tests were 0 or near 0, indicating a non-normal distribution of residuals, and thereby a violation of the parametric assumption of linear regression models. We tested the same model on a logarithmically transformed sample of the target data and got the same results (see *Table 6* below), suggesting that even log-transformation was insufficient to normalize the distribution of the target variable. Possibly a larger dataset would conform to a more normal distribution. As a consequence, we achieved lowest overall error and best fit using non-parametric models which don’t assume normality.

**Table 6**

OLS Regression Results (with log-transformation)

==============================================================================

Dep. Variable: Log\_Price\_euros R-squared: 0.458

Model: OLS Adj. R-squared: 0.457

Method: Least Squares F-statistic: 232.6

Date: Fri, 10 May 2024 Prob (F-statistic): 2.72e-109

Time: 10:08:12 Log-Likelihood: -521.95

No. Observations: 828 AIC: 1052.

Df Residuals: 824 BIC: 1071.

Df Model: 3

Covariance Type: nonrobust

==============================================================================

Omnibus: 98.857 Durbin-Watson: 2.059

**Prob(Omnibus): 0.000** Jarque-Bera (JB): 290.833

Skew: -0.592 **Prob(JB): 7.02e-64**

Kurtosis: 5.651 Cond. No. 20.1

Our best performing linear model was a Lasso regressor (L1 Regularization) with α=2.0. L1 Regularization seemed best suited to offsetting the violation of parametric assumptions. The best models used a large feature-set and mitigated overfitting through regularization. *Table 7* below shows results for linear models on the test data.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Table 7  *Test Metrics for Linear Models* | | | | | | | |
| Model Name | **Model Type** | **RMSE** | **MAE** | **R2** | **RMSE Generalization** | **MAE Generalization** | **R2 Generalization** |
| LC2 | Lasso | 367.78 | 268.25 | 72.17 | 34.81 | 19.09 | 1.9 |
| EN\_opt | ElasticNet | 369.09 | 269.58 | 71.97 | 45.45 | 26.89 | 3.5 |
| RC2 | Ridge | 372.77 | 276.85 | 71.40 | 40.01 | 25.79 | 2.7 |
| OLS3 | Ordinary Least Squares | 382.68 | 279.51 | 69.86 | 37.66 | 21.21 | 2.3 |

The Lasso model’s residual plot (*Figure 11* overleaf) reveals a fairly regular pattern of errors. Most large errors occur when predicting over €1500; there are more large undervaluations than overvaluations. 97% of predictions fall within the prediction interval represesented by the dotted green line, defined as ± 2 x Standard Error*.*

Figure 10

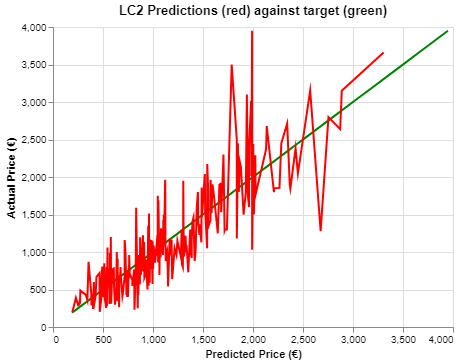
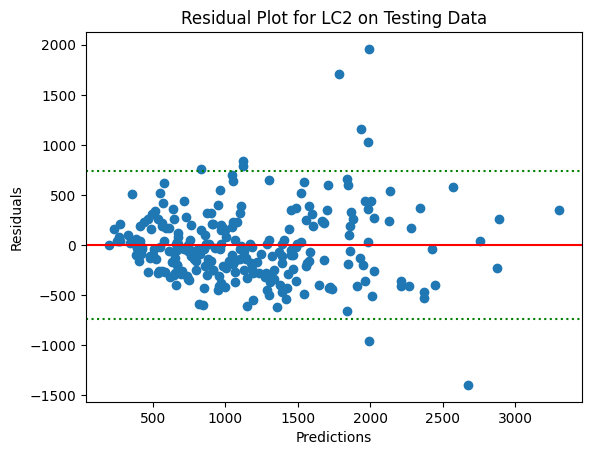


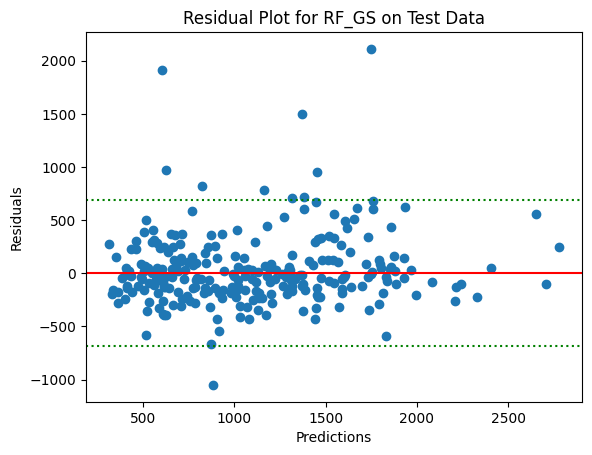
Figure 11



### 3.2 Non-Parametric Models

Decision Tree and Random Forest models had lower error and better fit (R2 Score) on the test data, despite a bigger gulf between training and test error than we achieved with linear models. Our **Random Forest** model had the lowest Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) and highest R2-score even while overfitting much more than the linear models, as shown by the higher error generalization, which subtracts test from training error.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Table 8  *Test Metrics for All Models* | | | | | | | |
| Model Name | **Model Type** | **RMSE** | **MAE** | **R2** | **RMSE Generalization** | **MAE Generalization** | **R2 Generalization** |
| RF\_gs | Random Forest | 343.68 | 226.42 | 75.7 | 210.04 | 133.02 | 20.1 |
| tree3a | Decision Tree | 393.99 | 263.37 | 68.1 | 163.33 | 129.32 | 19.5 |
| tree\_gs\_X4 | Decision Tree | 388.83 | 265.61 | 68.9 | 82.75 | 64.74 | 9.2 |
| LC2 | Lasso | 367.78 | 268.25 | 72.2 | 34.81 | 19.09 | 1.9 |
| EN\_opt | ElasticNet | 369.09 | 269.58 | 72.0 | 45.45 | 26.89 | 3.5 |
| RC2 | Ridge | 372.77 | 276.85 | 71.4 | 40.01 | 25.79 | 2.7 |
| OLS3 | Ordinary Least Squares | 386.80 | 279.51 | 69.9 | 37.66 | 21.21 | 2.2 |

**A graph with a green line and a line

Description automatically generated*Figure 12* shows the RandomForest’s predictions in blue against the target in green. We can see that the model keeps track of the price-trend fairly well up to around Quartile 3 (Q3=€1487.87), after which much larger errors occur. These larger errors in predicting pricier laptops explain the difference between RMSE and MAE, since the former magnifies larger errors through squaring. 96% of predictions fall within the prediction interval.

Figure 12

*Figure 13* plots the model’s residuals, revealing:

Figure 8

1. A highly abnormal distribution of residuals.
2. The model undervalues more often than it overvalues: there is only one overvaluation large enough to exceed the prediction interval represented by the dotted green line.

Figure 13

## Discussion

The popular understanding of Moore’s Law is the claim that “computing power at fixed cost is doubling every 18 months” (Tuomi, 2002). Although the validity of the “Law” is a matter of controversy, the relevant point is that the relationship between computing power and price does not remain constant over time.

|  |  |
| --- | --- |
| Table 9  *LC2 Coefficient Weights* | |
| Feature | **Coefficient Weight** |
| Ram | 3383 |
| Pixels | 693 |
| TypeName\_Workstation | 471 |
| Weight | 365 |
| Memory\_SSD | 247 |
| Cpu\_fast\_high\_end | 223 |
| TypeName\_Ultrabook | 172 |
| Windows | 155 |
| Company\_Apple | 77 |
| Company\_Toshiba | 69 |
| IPS\_display | 27 |
| Gpu\_Nvidia | 27 |
| Cpu\_efficient\_low\_end | 9 |
| Touchscreen | -17 |
| Company\_Dell | -17 |
| Memory\_Flash | -68 |
| Company\_Asus | -129 |
| TypeName\_Netbook | -132 |
| Gpu\_AMD | -141 |
| Company\_Acer | -158 |
| TypeName\_Notebook | -249 |

There are two main implications to this. The first is that we should prefer a linear model over a tree-based regressor like RandomForest, even if the latter scores better on test data. This is because tree models are not capable of extrapolating trends forward in time in the way linear models are. For example, if we look at an example Decision-Tree structure (*Figure 14* overleaf), we’ll see that price predictions (“value=y”) are conditioned on fixed values for x, like Ram <= 14 for example. In other words, the model wouldn’t be concerned if every laptop it predicted on met that condition, nor update its valuation if a future laptop had 1400GB RAM.

On the other hand, a linear model like the Lasso will predict by applying coefficient weights to the overall magnitude of the specification. For example, as *Table 9* shows, every additional GB RAM increases price prediction by ~€33.83 *ceteris paribus* (units of the target variable are denominated to the cent).

The essential caveat to be borne in mind is the non-constant relationship between computing power and price. On deployment, a linear model must be updated on a minimum 18-month schedule to halve coefficient weights related to computing power described by Moore’s Law, like ‘Ram’ and ‘Pixels’. Better yet, it should be retrained on new data, ideally every year, or on every major release of new products. More data is also likely to normalize the distribution of our target variable given the Central Limit Theorem, and thereby better meet the parametric assumptions of linear models.

A diagram of a network

Description automatically generated

Figure 14

*Example Decision Tree Structure*

## Conclusion

Our Lasso model, *LC2,* although imperfect, will provide new businesses looking to enter the laptop market with a good sense of an expected market price for a laptop based on its specifications. We found RAM, Screen Resolution, Weight and the inclusion of a Solid State Drive (SSD) to be the most price-predictive specifications. Workstations sold highest and Netbooks lowest. Acer is most represented in the market for cheaper products, and Apple for premium products. Fast, high-end CPUs sold for most.

This tool will aid new companies in assessing potential profit when balancing input costs against expected revenue. The model works best for cheap to mid-range products; companies should exercise caution when using the model to predict prices for higher-end products. Users should also bear in mind a roughly €275 average error even in these ranges.

We recommend finally that the model must be retrained with new data on a minimum 18-month schedule to keep pace with computational advances and market conditions.

# References

Laptops - Worldwide. (n.d.). Retrieved May 12, 2024, from <https://www-statista-com.ezproxy.lib.uts.edu.au/outlook/cmo/consumer-electronics/computing/laptops/worldwide>

Sweney, M. (2021, March 22). “Global shortage in computer chips ‘reaches crisis point’”. *The Guardian.* <https://www.theguardian.com/business/2021/mar/21/global-shortage-in-computer-chips-reaches-crisis-point>.

Tuomi, I. (2002). ‘The Lives and Death of Moore’s Law’. *First Monday*, 7: 11. <https://firstmonday.org/ojs/index.php/fm/article/download/1000/921>.

The dataset was sourced at:

<https://www.kaggle.com/datasets/ultimus/laptop-prices-prediction>.

# Appendix A

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Table A  *Summary of Data Preprocessing* | | | | | | | |
| *Original Column Names* | **Original Dtype** | **Data Preparation Technique(s)** | **Appendix §** | **Rationale(s)** | **Cleaned Column Name(s)** | **Original (cleaned) Column(s) Dropped?** | **New Datatype(s)** |
| *Company* | object | One-hot encoding | 3.3 | Machine interpretation | 19 new columns formatted as ‘Company\_{unique\_value}’ e.g. ‘Company\_Razer’­ | Yes | Binary Integers |
| *Product* | object | Dropped column | 3.2 | Too many unique values | NA | Yes | NA |
| *TypeName* | object | One-hot encoding | 3.3 | Machine interpretation | 6 new columns formatted as ‘TypeName\_{unique\_value}’ e.g. ‘TypeName\_Notebook’ | Yes | Binary Integers |
| *ScreenResolution* | object | Dummies for screen type | 2.6, 3.4 | Machine interpretation, retaining information | ‘Touchscreen­’, ‘IPS\_display’ | Yes | Binary Integers |
| Extracting int for overall resolution | 3.4 | ‘Pixels’ |  | Int |
| *Ram* | object | Extracting int for GBs | 2.8 | Machine interpretation | ‘Ram’ | No | Int |
| *Memory* | object | Dummies for drive type(s) | 3.7.1 | Machine interpretation, retaining information | 4 new columns formatted as ‘Memory\_{drivetype}’ | Yes | Binary Integers |
| Extracting int for total GBs storage | 3.7.2 | ‘Memory\_GB’ | Int |
| *Gpu[[1]](#footnote-1)* | object | Dummies for maker | 3.5 | Machine interpretation, retaining information | ‘Gpu\_AMD’, ‘Gpu\_Nvidia’, ‘Gpu\_Intel’ | Yes | Binary |
| *Cpu* | object | Dummies for maker | 3.6.1 | Machine interpretation, retaining information, differentiation | ‘Cpu\_Intel’, ‘Cpu\_AMD’, ‘Cpu\_Samsung’ | (Yes) | Binary |
| Extracting float for GHz frequency | 3.6.2 | ‘Cpu\_frequency’ | (Yes) | Float |
| Extracting Model Number | 3.6.3.1 | ‘Cpu\_model\_no.’ | (Yes) | Int |
| Mapping to performance and efficiency scales | 3.6.3.2 | ‘Cpu\_efficiency’, ‘Cpu\_performance’ | (Yes) | Ordinal integers |
| Ranking processor family | 3.6.4 | ‘Cpu\_family’ | (Yes) | Ordinal integers |
| Principal Components Analysis | 3.11 | Dimensionality reduction | ‘Cpu\_efficient\_low\_end’, ‘Cpu\_AMD\_mid\_range’, ‘Cpu\_efficient\_high\_end’, ‘Cpu\_fast\_high\_end’, ‘Cpu\_fast\_low\_end’ | Yes (No) | Floats |
| *OpSys* | object | Dummies for software company | 3.8.1 | Machine interpretation, retaining information | ‘Mac’, ‘Windows’, ‘Linux’, ‘Google’. ‘No OS’ represented with 0s in all columns. | Yes | Binary |
| Ranking OS for same company | 3.8.2 | Scaling values | ‘OS\_rank’ | Ordinal Integers 0-3 |
| *Weight* | object | Extracting float for KGs | 2.8 | Machine interpretation | ‘Weight’ | No | Float (2 decimal places) |
| *Price\_euros* | float64 | Dropping extreme outliers defined as outside 3\*IQR+Q3 | 3.1 | Normalization, anomaly cleaning | ‘Price\_euros’ | No | Float (2 decimal places) |
| Log transformation | ‘Log\_Price\_euros’ | Float (log notation) |

1. A more skilled preprocessing team might have been able to make more of this column, as we could with the ‘Cpu’ column, but we were ultimately stumped by the sheer variety of naming conventions for GPU models. [↑](#footnote-ref-1)