Interpreting Machine Learning Models for Predicting PC Prices

Tenzin Sim
Applied AI & Analytics, Singapore Polytechnic

June 10, 2022

Abstract

We explore different machine learning methods for predicting the prices of personal computers (PC) from their given specifications. We use linear regression models to understand how the PCs are priced based on their brands and configurations. Among the regression methods, LASSO has the best prediction performance, and it also provides interesting relations of how different brands and configurations could impact the price of PCs. We also explore the more flexible methods including decision tree, random forest, and k-Nearest Neighbors. Although these models have better predicting accuracy than the regression models, they are also harder to interpret. We also explore a simple decision tree, which provides some interesting insights on predicting PC prices.

1 Introduction

In this paper, we explore different machine learning methods for predicting the prices of personal computers (PC) based on their given specifications. Machine learning models that are high in flexibility would perform better in prediction, but their interpretability becomes low (James et al., 2013). The lower the interpretability of the model, the harder it is for someone to comprehend why certain decisions or predictions have been made (Miller, 2019). High interpretability models are useful for inference, which provides understanding of the relationship between the response and the explanatory variables. Figure 1 shows the different machine learning methods on the interpretability-flexibility chart. In general, as the flexibility of a method increases, its interpretability decreases.

For interpretability, we explore linear regression methods, including OLS, Ridge and LASSO, which score high on interpretability. We also explore a simple decision tree, which provides some interesting insights on predicting PC prices.

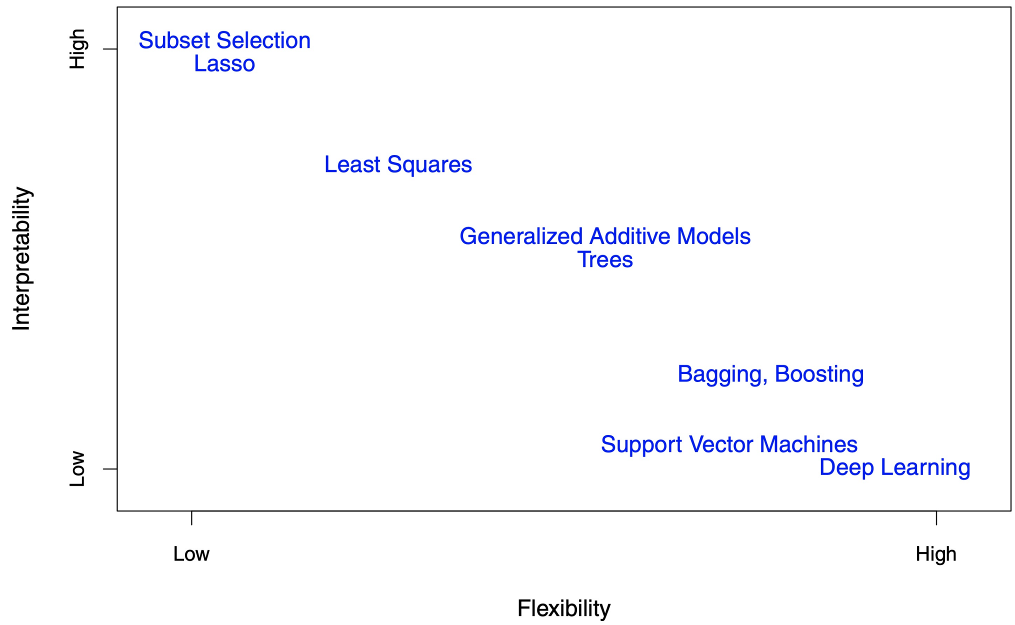

Figure 1: A representation of the tradeoff between flexibility and interpretability, using different statistical learning methods (James et al., 2013).

2 Data and Preprocessing

We preform preprocessing of the data before using them for as inputs to the machine learning models. Figure 2 shows a snapshot of the data, which contain over 15,000 prices of different PC models with respective configurations.

We extract from the data in total of five categorical input variables as follows:

1. **Brand:** 19 different choices including Apple, HP, Acer, Dell, Asus, and so forth.
2. **PC Type:** 6 different types including Netbook, Ultrabook, Notebook, Gaming and so forth.
3. **CPU model:** 93 different types including Intel Core i5, Intel Core i7, and so forth.
4. **GPU model:** 110 different types including Nvidia Geforce GTX 1050, AMD Radeon Pro 560, Intel HD Graphics 620, and so forth.
5. **Operating System:** 9 different types including Android, macOS, Windows 10, Linux and so forth.

pc_data (regression)

1	Product ID	Brand	Type	Screen Size	Screen Specs	CPU	RAM	Hard Disk	GPU	Operating System	Weight	Price ($)
2	0	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8GB	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37kg	3568.93416
3	1	Apple	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8GB	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34kg	2394.77616
4	2	HP	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8GB	256GB SSD	Intel HD Graphics 620	No OS	1.86kg	1531.8
5	3	Apple	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16GB	512GB SSD	AMD Radeon Pro 455	macOS	1.83kg	6759.7668
6	4	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8GB	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37kg	4804.7904
7	5	Acer	Notebook	15.6	1366x768	AMD A9-Series 9420 3GHz	4GB	500GB HDD	AMD Radeon R5	Windows 10	2.1kg	1065.6
8	6	Apple	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.2GHz	16GB	256GB Flash Storage	Intel Iris Pro Graphics	Mac OS X	2.04kg	5700.88008
9	7	Apple	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8GB	256GB Flash Storage	Intel HD Graphics 6000	macOS	1.34kg	3086.7768
10	8	Asus	Ultrabook	14	Full HD 1920x1080	Intel Core i7 8550U 1.8GHz	16GB	512GB SSD	Nvidia GeForce MX150	Windows 10	1.3kg	3982.68
11	9	Acer	Ultrabook	14	IPS Panel Full HD 1920x1080	Intel Core i5 8250U 1.6GHz	8GB	256GB SSD	Intel UHD Graphics 620	Windows 10	1.6kg	2051.28
12	10	HP	Notebook	15.6	1366x768	Intel Core i5 7200U 2.5GHz	4GB	500GB HDD	Intel HD Graphics 620	No OS	1.86kg	1049.3496
13	11	HP	Notebook	15.6	Full HD 1920x1080	Intel Core i3 6006U 2GHz	4GB	500GB HDD	Intel HD Graphics 520	No OS	1.86kg	919.05336
14	12	Apple	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.8GHz	16GB	256GB SSD	AMD Radeon Pro 555	macOS	1.83kg	6500.08008
15	13	Dell	Notebook	15.6	Full HD 1920x1080	Intel Core i3 6006U 2GHz	4GB	256GB SSD	AMD Radeon R5 M430	Windows 10	2.2kg	1329.0696
16	14	Apple	Ultrabook	12	IPS Panel Retina Display 2304x1440	Intel Core M m3 1.2GHz	8GB	256GB SSD	Intel HD Graphics 615	macOS	0.92kg	3363.0336
17	15	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8GB	256GB SSD	Intel Iris Plus Graphics 640	macOS	1.37kg	4045.4172
18	16	Dell	Notebook	15.6	Full HD 1920x1080	Intel Core i7 7500U 2.7GHz	8GB	256GB SSD	AMD Radeon R5 M430	Windows 10	2.2kg	1984.68
19	17	Apple	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.9GHz	16GB	512GB SSD	AMD Radeon Pro 560	macOS	1.83kg	7613.712
20	18	Lenovo	Notebook	15.6	Full HD 1920x1080	Intel Core i3 7100U 2.4GHz	8GB	1TB HDD	Nvidia GeForce 940MX	No OS	2.2kg	1329.336
21	19	Dell	Ultrabook	13.3	IPS Panel Full HD / Touchscreen 1920x1080	Intel Core i5 8250U 1.6GHz	8GB	128GB SSD	Intel UHD Graphics 620	Windows 10	1.22kg	2608.056
22	20	Asus	Netbook	11.6	1366x768	Intel Atom x5-Z8350 1.44GHz	2GB	32GB Flash Storage	Intel HD Graphics 400	Windows 10	0.98kg	511.2216
23	21	Lenovo	Gaming	15.6	IPS Panel Full HD 1920x1080	Intel Core i5 7300HQ 2.5GHz	8GB	128GB SSD + 1TB HDD	Nvidia GeForce GTX 1050	Windows 10	2.5kg	2661.336
24	22	HP	Notebook	15.6	1366x768	AMD E-Series E2-9000e 1.5GHz	4GB	500GB HDD	AMD Radeon R2	No OS	1.86kg	687.312
25	23	Dell	2 in 1 Convertible	13.3	Full HD / Touchscreen 1920x1080	Intel Core i5 8250U 1.6GHz	8GB	256GB SSD	Intel UHD Graphics 620	Windows 10	1.62kg	2181.816
26	24	HP	Ultrabook	15.6	Full HD 1920x1080	Intel Core i7 8550U 1.8GHz	8GB	256GB SSD	Intel HD Graphics 620	Windows 10	1.91kg	1755.576
27	25	Dell	Notebook	15.6	1366x768	Intel Core i3 6006U 2GHz	4GB	1TB HDD	Intel HD Graphics 520	Windows 10	2.3kg	1115.25696
28	26	Apple	Ultrabook	13.3	1440x900	Intel Core i5 1.6GHz	8GB	128GB Flash Storage	Intel HD Graphics 6000	Mac OS X	1.35kg	2927.736

Figure 2: Snapshot of data on Apple Numbers.

The extracted categorical values go through Python sklearn OneHotEncoder where each unique type of a given categorical input variable is assigned to a separate dummy numerical variable, with value of either zero or one, where one represents the selection of the type associated with the categorical input value. The are also nine numerical input variables as follows,

1. **Pixel count:** Screen resolution data is extracted from the Screen Specs column where the horizontal and vertical pixel counts are multiplied to derive the total number of pixels.

2. **CPU Clock speed:** The base clock speed (in GHz) of the CPU is extracted from the CPU column.

3. **Memory (RAM) size :** The memory size is (in GB).

4. **SSD storage:** The storage size of the SSD (in GB) is extracted from the Hard Disk column.

5. **Hard Drive (HDD) storage:** The storage size of the Hard Disk (in GB) is extracted from the Hard Disk column.

6. **Flash Drive storage:** The storage size of the Flash Drive (in GB) is extracted from the Hard Disk column.

7. **Hybrid Drive storage:** The storage size of the Hybrid Drive (in GB) is extracted from the Hard Disk column.

8. **Weight:** The weight of the laptop (in kg)

After preprocessing, we represent the data as a Python numpy 2D array. The array then goes through the Python sklearn Train Test Split function to generate the Testing data by randomly selecting 30% of the data, while the rest of the data is used for training and validation of the machine learning models as follows:

```
# Python code for splitting data
X_train,X_test,y_train,y_test=
    train_test_split(X,y,test_size=0.3,random_state=21)
```

We use the Testing data for to evaluating and comparing the performance across different machine learning models. We use Mean square error (MSE) as an indicator for model performance, which evaluates the average squared deviations between the predicted and true values. Hence, MSE more severely punishes models for producing outlier results. We explore different machine learning methods including Linear Regression, Ridge, LASSO, Random Forest, Gradient Boosting, k-Nearest Neighbour and Support Vector Regression, and Decision Tree (see Figure 3). Among these approaches, Decision Tree has the best performance with the lowest MSE. In terms of computational speed, Linear Regression methods performs significantly faster than the tree based methods. However, the tree Based models could be sped up using hardware acceleration such as Intel® Deep Learning Boost.

3 Linear regression models

Linear regression models can provide a better understanding on how the price of PCs are related to their brands and specification. We explore three linear regression methods

	r2_Score	Mean Squared Error	RMS Error	Mean Absolute Error	% Mean Absolute Error
Model					
RandomForest	0.998175	6100.852661	78.107955	21.510221	0.008017
GB	0.985094	49836.285459	223.240421	154.138586	0.065327
knn	0.998562	4809.264072	69.348858	18.306743	0.007101
linreg	0.885331	383373.340272	619.171495	432.141416	0.168568
tree	0.998578	4753.941328	68.948831	18.446395	0.007126
Ridge	0.885347	383320.891664	619.12914	431.923648	0.168382
Lasso	0.885364	383264.485911	619.083586	432.300473	0.168313
svr	0.97373	87829.610213	296.360608	106.426866	0.036854
Dummy	-0.03097	3446849.192673	3446849.192673	1373.209377	0.627274

Figure 3: Table comparing the testing results across the various models

including LASSO, Ridge and Ordinary Least Square (OLS) regression. OLS minimizes the square sum of errors squared as the loss function,

$$\sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2.$$

LASSO (Least Absolute Shrinkage and Selection Operator) works similarly by adding the "absolute value of magnitude" of coefficient as penalty term to the loss function as follows,

$$\sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 + \alpha \sum_{j=1}^{p} |\beta_j|.$$

The key difference between these techniques is that LASSO shrinks the less important feature's coefficient to zero thus, removing some features altogether. This is useful for feature selection from a huge number of features where the zero coefficients could be removed and be regarded as insignificant input variables.

Ridge regression, similar to LASSO adds a penalty function to the loss function, except in this case, the penalty function is squared as follows,

$$\sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \alpha \sum_{j=1}^{p} \beta_j^2.$$

For both the LASSO and Ridge regression, the hyper-parameter α is tuned using 5-fold cross validation with Grid Search on alpha:

```
'Ridge':{'Ridge__alpha':np.logspace(-5,5,20)},
'LASSO':{'LASSO__alpha':np.logspace(-5,5,20)}
```

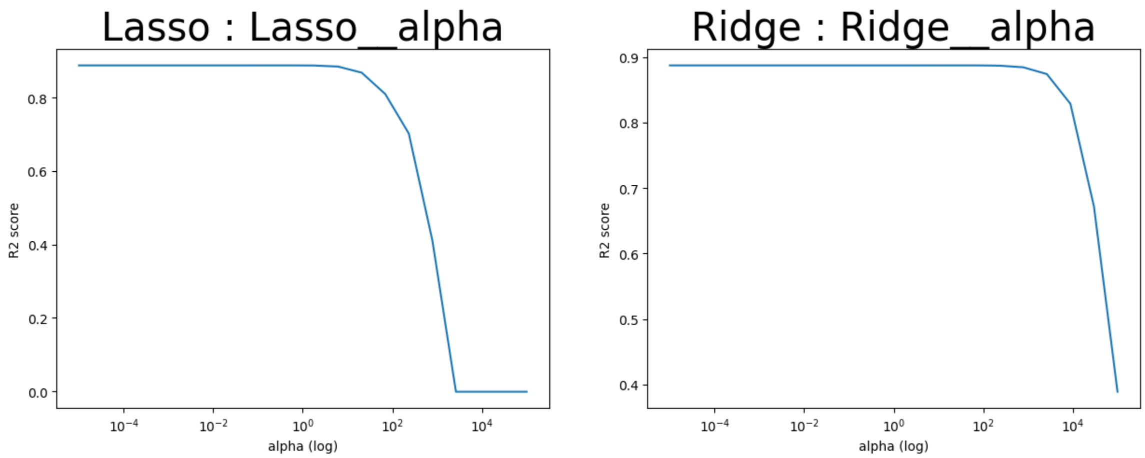

Figure 4: Plot of R^2 cross validation performance against α.

We have found that smaller alpha values tend to yield better model prediction results (see Figure 4). Hence, their prediction performance is quite similar to OLS. Nevertheless, when validated on the test data, LASSO has slightly lower MSE compared to Ridge and OLS (see Figure 3)

Among the three techniques, LASSO has the lowest Mean squared error (MSE) loss when evaluated on the test data (see Figure 3). We note that while regression techniques perform poorly on MSE compared to some of the others, they are better suited for interpretation. The coefficients of the input variables could be obtained and interpreted, which can be used to understand how the computer brands and configurations could have positive/negative impacts on the PC prices.

From the coefficients of the different PC brands in Table 1, we can analyze the impact of the Brand Name on the PC prices. For instance, we infer that Razer and LG tend to have high PC prices given the same product specifications. These brands may have overpriced their products compared to baseline brands such as Apple, Lenovo and MSI.

4 Decision tree model

We provide the tree diagram for predicting the price of PCs in Figure 5. However, such complex decision trees may be hard to interpret. Hence, in Figure 6, we have included a smaller tree of a maximum of eight features and a depth of three. Although the simple de-

Brands	Coefficients
Razer	1368.3
LG	1018.5
Google	477.9
Samsung	386.1
Toshiba	364.6
Microsoft	225.6
Xiaomi	222.7
HP	35.8
Apple	0.0
Lenovo	0.0
MSI	-0.0
Dell	-78.4
Asus	-226.0
Acer	-319.4
Huawei	-385.8
Mediacom	-421.7
Vero	-464.3
Fujitsu	-479.9
Chuwi	-653.8

Table 1: LASSO regression coefficient on the 19 brands.

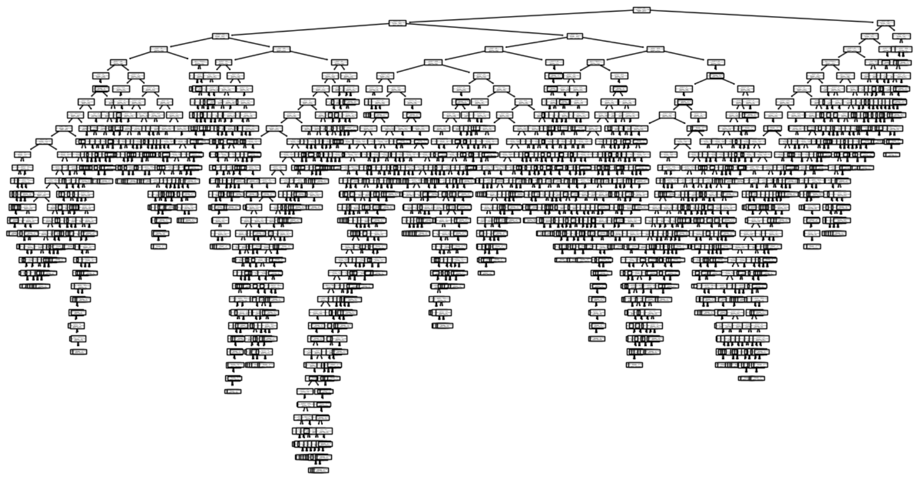

Figure 5: Full Decision Tree for Price Prediction

cision tree has poorer prediction, it provides interesting insights. For instance, in analysing Figure 6, we find that memory size, followed by GPU model, are one of the main factors influencing the price of the PC. We also find that higher end PCs have more than 14GB of RAM and use the Nvidia Geforce GTX 1080 (GTX 1080) Graphics card. In fact, back in 2017, the GTX 1080 was one of the best graphics card options for high performance gaming PCs. On the other hand, cheaper PCs tend to have less than 14GB of RAM and use the Intel HD Graphics 500. With less RAM installed, the manufacturer is able to cut the costs of the PC and lower its selling price. Furthermore, considering that the Intel HD Graphics 500 is integrated graphics, it means that the PC does not have an additional Discreet Graphics Processor which further decreases the price of the PC.

5 Discussions

We identify several interesting applications of using the machine learning models for predicting PC prices, as well as knowing how the different configurations could affect the prices. By analysing the coefficients of the LASSO model, we are able to find the factors which impact the price of the PC. PC manufacturers could use this relationship to produce PCs with configurations that are most profitable, that is, the price over cost ratio. Retailers selling PCs could also use high resolution Decision Tree to identify a collection of profitable products to sell, and how much to sell them for. Procurement departments in companies

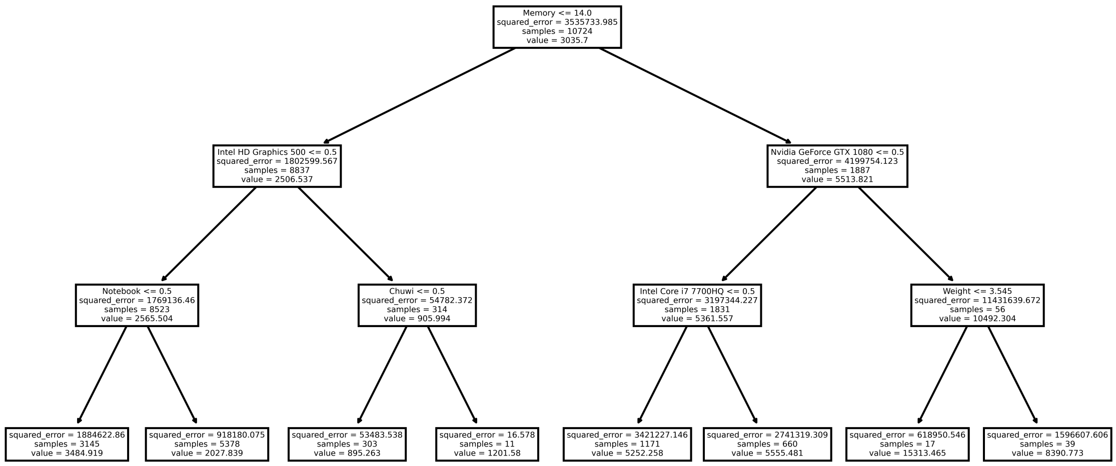

Figure 6: Reduced tree diagram limited 8 features and depth 3

buying PCs in bulk can use the machine learning models to estimate the prices for the range of PCs that suit their needs. Thus, they could better estimate the budget for the purchase.

6 Conclusion

In this paper, we explore the various machine learning techniques to understand the trade-offs between interpretability and prediction performance. Models such as linear regression and simple decision trees are easier to interpret but they are less precise in their predictions. In contrast, models that are highly flexible and accurate, such as Random Forrest and k-Nearest Neighbours are much harder to interpret in relating how the price of PCs are affected by their configurations. We also discuss several applications of predicting PC prices with machine learning models.

References

James, G., D. Witten, T. Hastie, and R. Tibshirani (2013). *An introduction to statistical learning*, Volume 112. Springer.

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence 267*, 1–38.