

Predicting PC Prices

Lim Jun Jie



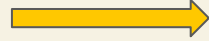


p2100788

DAAA/FT/2A/02





Table of Contents

#1 Feature Engineering (1)		Manipulate and transform the raw data to simplify it so EDA can be done
#2 EDA		Find out the characteristics of the dataset as well as spot patterns and anomalies
#3 Feature Engineering (2)		Further preprocess the data for modeling
#4 ML Modeling		Train, select and evaluate models based on the objective of the task
#5 Model Interpretation		Interpret and explain the model's decision making



Information

- This dataset contains prices of different PCs and its configurations such as its brand, type and weight among many other features
- This task is a regression task to predict the price of a PC, given the information about its configurations

Objectives

- Build a favourable model to estimate prices of PC, by minimising Mean Squared Error



Feature Engineering (1)

Drop columns

Does not have any relationship with target variable: ProductID

Extracted features using regex

Extracted from existing features: CPU Brand, GPU Brand, CPU Clock Speed, Touchscreen, Screen Specs, Screen Resolution

Initial Summary

No missing data, a lot of object data type columns,

Numeric features

Made features numeric by removing the units: RAM (GB), Weight (kg), CPU Clock Speed (GHz)

Cleaned up Hard Disk feature

Split up all different types of Hard Disks and made the columns numeric

New feature: Pixels Per Inch

$$\text{PPI} = \frac{\text{diagonal in pixels}}{\text{diagonal in inches}}$$

PPI measures pixel density of a computer screen



Feature Engineering (1)

Before:

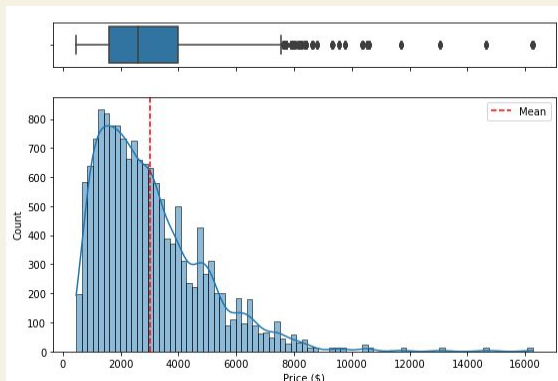
Product ID	Brand	Type	Screen Size	Screen Specs		CPU	RAM	Hard Disk		GPU	Operating System	Weight	Price (\$)
0	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600		Intel Core i5 2.3GHz	8GB	128GB SSD		Intel Iris Plus Graphics 640	macOS	1.37kg	3568.93416
1	Apple	Ultrabook	13.3	1440x900		Intel Core i5 1.8GHz	8GB	128GB Flash Storage		Intel HD Graphics 6000	macOS	1.34kg	2394.77616
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	1531.80000
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	6759.76680
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	4804.79040

After:

Brand	Type	Screen Size	Screen Specs	CPU	RAM (GB)	GPU	Operating System	Weight (kg)	Price (\$)	Flash Storage	HDD	Hybrid	SSD	Touchscreen	CPU Clock Speed (GHz)	CPU Brand	GPU Brand	ppi
Apple	Ultrabook	13.3	IPS Panel Retina Display	Intel Core i5	8	Intel Iris Plus Graphics 640	macOS	1.37	3568.93416	0.0	0.0	0.0	128.0	No	2.3	Intel	Intel	226.983005
Apple	Ultrabook	13.3	None	Intel Core i5	8	Intel HD Graphics 6000	macOS	1.34	2394.77616	128.0	0.0	0.0	0.0	No	1.8	Intel	Intel	127.677940
HP	Notebook	15.6	None	Intel Core i5 7200U	8	Intel HD Graphics 620	No OS	1.86	1531.80000	0.0	0.0	0.0	256.0	No	2.5	Intel	Intel	141.211998
Apple	Ultrabook	15.4	IPS Panel Retina Display	Intel Core i7	16	AMD Radeon Pro 455	macOS	1.83	6759.76680	0.0	0.0	0.0	512.0	No	2.7	Intel	AMD	220.534624
Apple	Ultrabook	13.3	IPS Panel Retina Display	Intel Core i5	8	Intel Iris Plus Graphics 650	macOS	1.37	4804.79040	0.0	0.0	0.0	256.0	No	3.1	Intel	Intel	226.983005



Exploratory Data Analysis



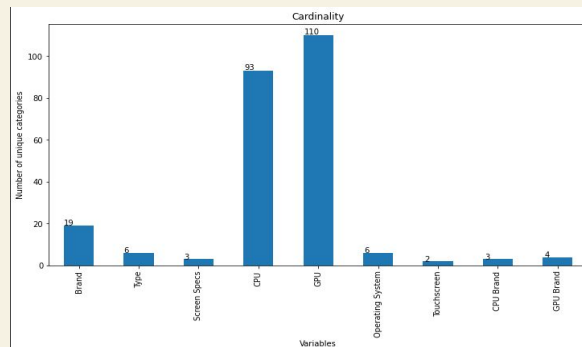
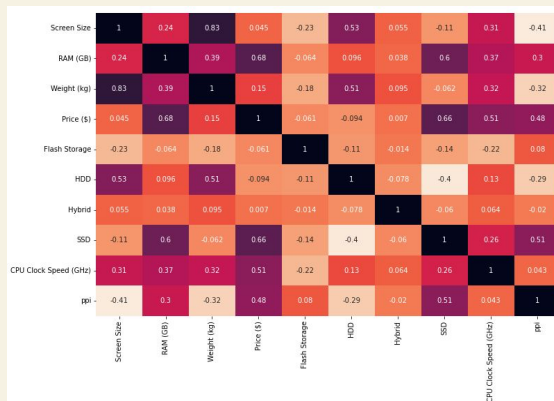
Skewness of Target Feature

Distribution of target variable
Price is positively skewed, this may affect the predictions as the outliers of target variable may affect how the model fits the dataset

Multicollinearity

Bivariate linear correlation can be observed between some numerical features.

This may be a issue in a regression task as features are assumed independence



High Cardinality

CPU and GPU features have very high cardinality (93 and 110 unique values respectively). Feature selection may need to be done to reduce the number of features in order to avoid the 'Curse of Dimensionality'

Feature Engineering (2)

#1

Operating System Feature

#2

Log Transformation

#3

Multicollinearity in nominal variables
(Cramer's V)

#4

Multicollinearity in continuous variables
(Variance Inflation Factor + Pearson
Correlation)

#5

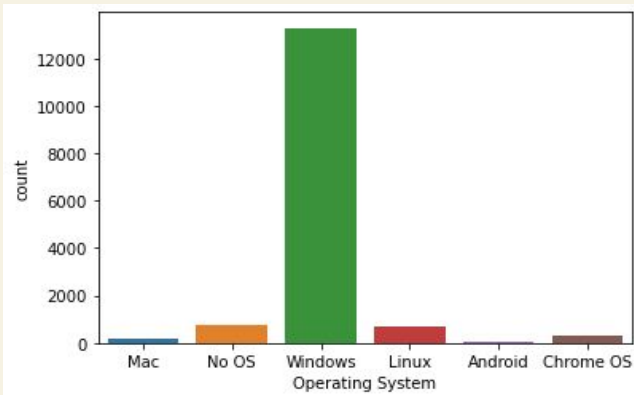
One Hot Encoding

#6

Feature Selection (Recursive Feature
Elimination)

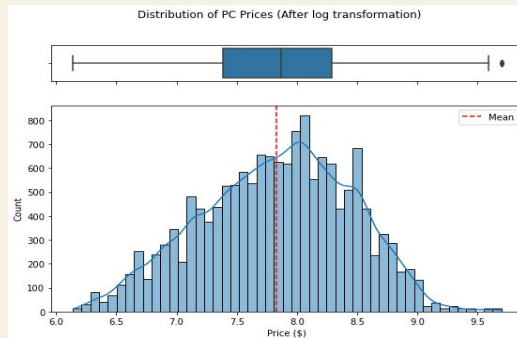


Feature Engineering (2)



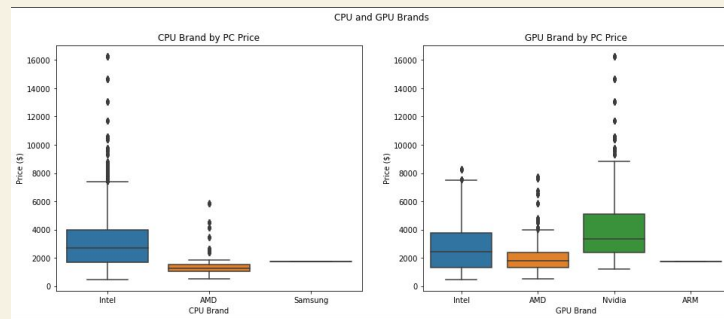
#1 Operating System Feature

Grouped different versions of Windows Operating Systems under 1 category 'Windows OS' and did the same process on all Mac OSes



#2 Log Transformation

Natural log transformation is done on price to set it to near-normal distribution so the outliers would not have a huge impact on the model during training





Feature Engineering (2)

About Multicollinearity

- When building a regression model, it is assumed that all explanatory variables are independent of each other
- if independent features are highly correlated, change in 1 feature may affect the other features
- This can lead to the model making bad predictions and can also affect the interpretability of the model
- For nominal features, Cramer's V test will be used to determine the strength of association
- For continuous features, Pearson's correlation (bivariate) and Variance Inflation Factor (multivariate) will be used

**** All inference is made only with training set to minimize data leakage***



Feature Engineering (2)

#3 Multicollinearity in nominal variables (Cramer's V)

- Cramer's V is a statistical test to determine the strength of association between 2 variables which has more than 2x2 rows and columns.
- It is only used when chi-square has determined that there is association between the 2 variables.
- For ALL combinations 2 nominal variables, p-values are 0 which is smaller than $\alpha = 0.05$, thus chi-square test determined that there is association between all combinations of variables

	variable_1	variable_2	p-value	cramer_v
24	CPU	CPU Brand	0.0	1.0
29	GPU	GPU Brand	0.0	1.0
28	GPU	CPU Brand	0.0	0.9348
35	CPU Brand	GPU Brand	0.0	0.81
25	CPU	GPU Brand	0.0	0.807
12	Type	Touchscreen	0.0	0.7777
10	Type	GPU	0.0	0.6863
15	Screen Specs	CPU	0.0	0.6861
22	CPU	Operating System	0.0	0.6511
1	Brand	Screen Specs	0.0	0.641
16	Screen Specs	GPU	0.0	0.6301
17	Screen Specs	Operating System	0.0	0.594
21	CPU	GPU	0.0	0.593
9	Type	CPU	0.0	0.5926
4	Brand	Operating System	0.0	0.5238
2	Brand	CPU	0.0	0.5148
26	GPU	Operating System	0.0	0.5094
27	GPU	Touchscreen	0.0	0.451
23	CPU	Touchscreen	0.0	0.4502
3	Brand	GPU	0.0	0.428
14	Type	GPU Brand	0.0	0.4178
7	Brand	GPU Brand	0.0	0.3392
0	Brand	Type	0.0	0.3279
5	Brand	Touchscreen	0.0	0.2963



Feature Engineering (2)

- A heatmap is plotted using Cramer V's test scores. There is a strong correlation between CPU-CPU Brand and GPU-GPU Brand
- Although there might be some hidden relationship between the CPU and GPU type and the price, I decide to drop CPU and GPU features from the dataset due to its very high cardinality
- As the score for GPU brand to CPU brand is 0.801, I printed out the crosstab to analyse it further
- From the crosstab, it is seen that the CPU brand can be deduced just from looking at the GPU brands. All Nvidia GPUs are only paired with Intel CPUs, Intel GPUs are only paired with Intel CPUs, ARM GPUs are only paired with Samsung CPUs, and AMD GPUs are only paired with either AMD or Intel CPUs.
- Hence I decided to drop CPU brand column due to its high association with GPU brand column

GPU Brand						
GPU Brand	AMD	ARM	Intel	Nvidia	All	
CPU Brand						
AMD	576	0	0	0	576	
Intel	1096	0	6816	3757	11669	
Samsung	0	11	0	0	11	
All	1672	11	6816	3757	12256	





Feature Engineering (2)

#4 Multicollinearity in continuous variables (VIF + Pearson's Correlation)

Variance Inflation Factor (VIF) is used to detect for collinearity exists in a regression model. It measures how much the correlation coefficient is inflated due to collinearity

Screen Size has a very large VIF of 66, thus it should be dropped. Dropping Screen Size would not affect the model's predictive power much as its linear correlation with target variable price is only 0.038.

After dropping Screen Size. VIF of other features dropped significantly. I decided to not drop anymore numerical features as VIF is < 20 for all features and there are no bivariate correlations of above 0.8 for all numerical features

	variables	VIF
0	Screen Size	66.410563
1	RAM (GB)	9.555751
2	Weight (kg)	30.650716
3	Flash Storage	1.168355
4	HDD	3.115334
5	Hybrid	1.075008
6	SSD	5.749792
7	CPU Clock Speed (GHz)	27.612514
8	ppi	16.366771

	variables	VIF
0	RAM (GB)	8.890992
1	Weight (kg)	15.187743
2	Flash Storage	1.156830
3	HDD	3.115079
4	Hybrid	1.070046
5	SSD	5.733217
6	CPU Clock Speed (GHz)	19.488171
7	ppi	10.959962



#5 One Hot Encoding

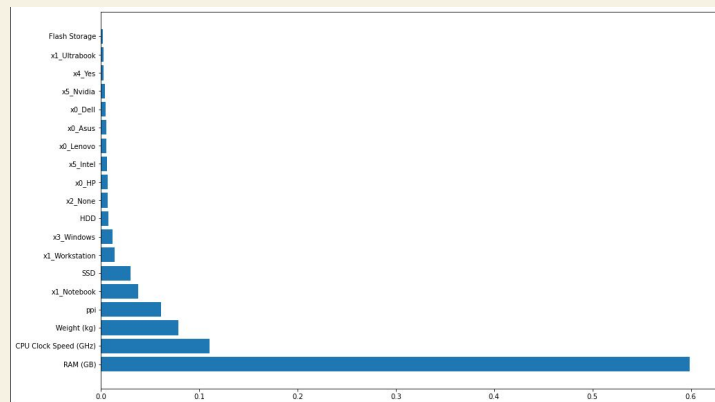
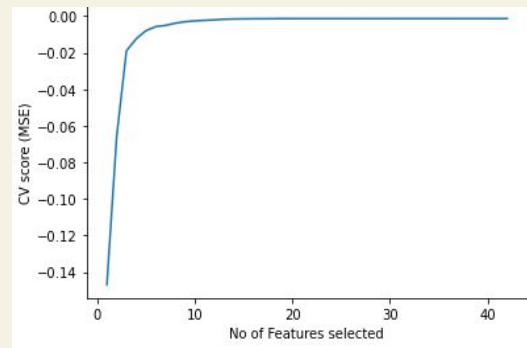
#6 Feature Selection (Recursive Feature Elimination)

All nominal features are One-Hot Encoded.
After One Hot Encoding there are 43 columns.

In order to reduce the dimensionality of the dataset and achieve a more interpretable model, recursive feature elimination is used with randomforestregressor as the estimator and mean squared error as the scoring metric

RFE recommended 19 features as the optimal number of features.

By fitting another RandomForestRegressor separately, the feature importance can be shown.



ML MODELING



ML Modeling

```
dummy_regr = DummyRegressor()
dummy_regr.fit(X_train, y_train)
y_pred = dummy_regr.predict(X_test)
✓ 0.3s

r2 = r2_score(y_test, y_pred)
print(f"Dummy Regressor R^2: {r2}")
✓ 0.3s

Dummy Regressor R^2: -0.0016682325811376852
```

Dummy Regressor

Serves as a reference point for model selection

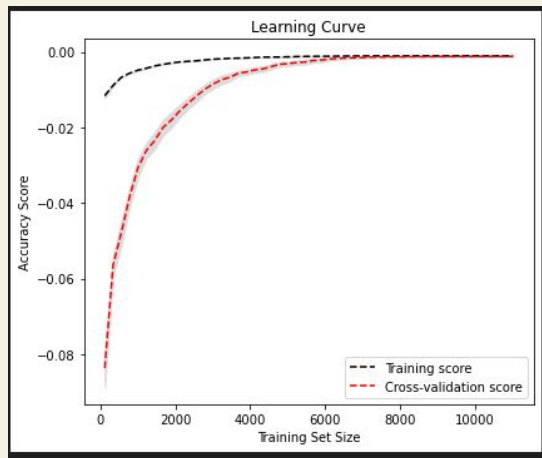
	Model	5-fold cv Negative MSE	test_mse	5_fold cv R^2 (%)	test_r2
0	Linear Regression	-0.094927	0.094683	75.226097	0.758440
1	K-Nearest Neighbors	-0.001873	0.001702	99.511219	0.995658
2	Elastic Net	-0.175617	0.175613	54.175750	0.551968
3	Decision Tree	-0.001229	0.001589	99.679399	0.995947
4	Random Forest	-0.001243	0.001584	99.675600	0.995959
5	ExtraTreesRegressor	-0.001229	0.001589	99.679399	0.995947
6	AdaBoost	-0.072906	0.072704	80.981059	0.814515
7	Gradient Boosting	-0.038590	0.039369	89.932129	0.899559
8	Bagging Regressor	-0.001281	0.001602	99.665730	0.995914

Baseline Model Selection

The tree-based models and Bagging Regressor both performed really well, having R^2 of over 99% and low mean squared errors of below 0.002



Interpretations



The learning curve for DecisionTreeRegressor shows that the model did not show any signs of overfitting, as the training score and cross validation score curves converged at one point to form a horizontal asymptote.

From the feature importances:

	feature	importance
18	x5_Nvidia	0.212989
6	ppi	0.182471
4	SSD	0.148652
1	Weight (kg)	0.096040
5	CPU Clock Speed (GHz)	0.084903
0	RAM (GB)	0.060364
11	x1_Notebook	0.038130
15	x3_Windows	0.031286
14	x2_None	0.028476
12	x1_Ultrabook	0.028344
3	HDD	0.017975
9	x0_HP	0.010597
10	x0_Lenovo	0.010459
13	x1_Workstation	0.010154
7	x0_Asus	0.010151
8	x0_Dell	0.010120
16	x4_Yes	0.009412
17	x5_Intel	0.007568
2	Flash Storage	0.001909

The model finds whether the PC contains Nvidia GPU or not as the most important, followed by ppi and SSD size.