Business recommendations to Data Science Science Lead

Project Recell: Paul-Yvann Djamen

10-07-2022

Contents / Agenda

- Executive Summary
- Business Problem Overview and Solution Approach
- EDA Results
- Data Preprocessing
- Model Performance Summary
- Appendix

Executive Summary and recommendation

1. The model is able to explain ~84% of the variation in the data and within 4.6% of the used-price on the test data, which implies our model does well for prediction and inference
2. If the weight of a device increases by one unit, then its used price increases by 0.0017 units, all other variables held constant
3. If the ram increases by one unit, then its used price increases by 0.0207 units, all other variables held constant
4. If the number of years since the device was model was made increases by one unit, then its used price decreases by 0.0292 units, all other variables held constant
5. The rating for anime released for TV will be 0.5598 units less than those released as DVD specials
6. As the used price increase with an increase in weight of the device for customers who like to use their devices for entertainment, Recell can improve its marketing activities to generate more profit by improving its prediction
7. Since most devices main camera have higher mp than selfie cameras, unlike what the data set suggests, the company can target customers who with age groups and geographical locations who use their devices to take pictures of the world around them as oppose to focusing on occupation.

Business Problem Overview and Solution Approach

- **Problem Overview**

The new IDC (International Data Corporation) forecast predicts that the used phone market would be worth $52.7bn by 2023 with a compound annual growth rate (CAGR) of 13.6% from 2018 to 2023. The rising potential of this under-the-radar market fuels the need for Machine Learning solutions to develop a dynamic pricing strategy for used and refurbished devices. ReCell, a startup aiming to be competitive in the market of used/refurbished cell phones and tablets is looking to hire a team of data scientists to predict prices of used devices

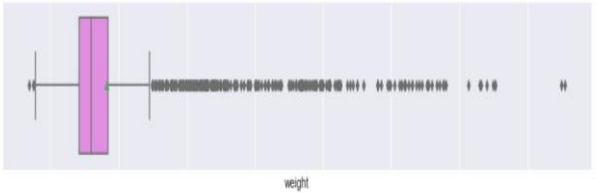
- **Solution approach**

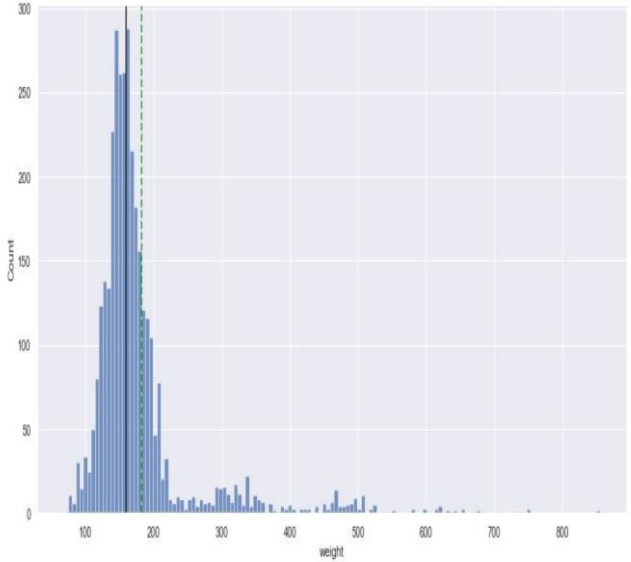
The data science team at Recell will analyze market data provided with several attributes and build a linear regression model to predict the price of a used phone/tablet and identify factors that significantly influence it by performing a number of tests and exploratory data analysis.

Results will be interpreted with key findings and business recommendations presented to the data science lead on which variables the company should focus on for prediction

EDA Results

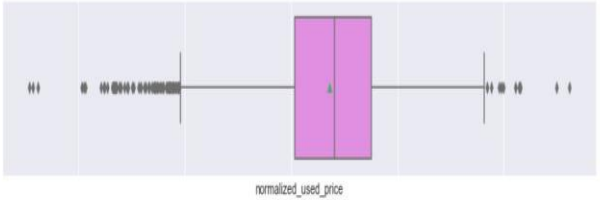
- Univariate Analysis

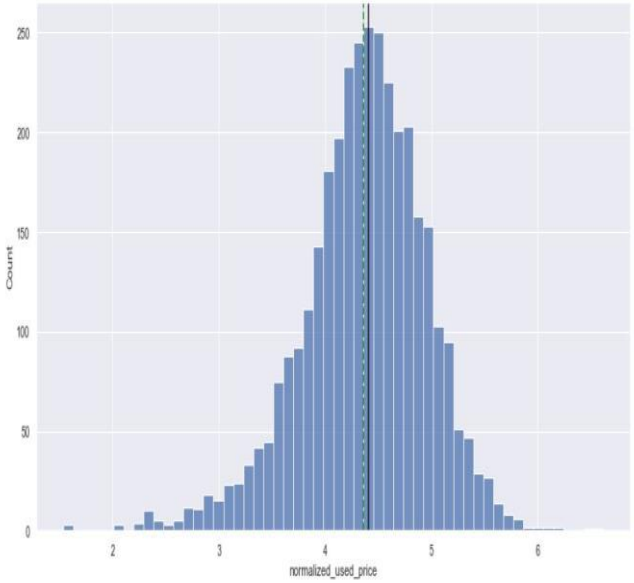




The response variable normalized_used_price is close to normally distributed with the mean and median price close to each other. Its range isn't high with a few outliers on both tails

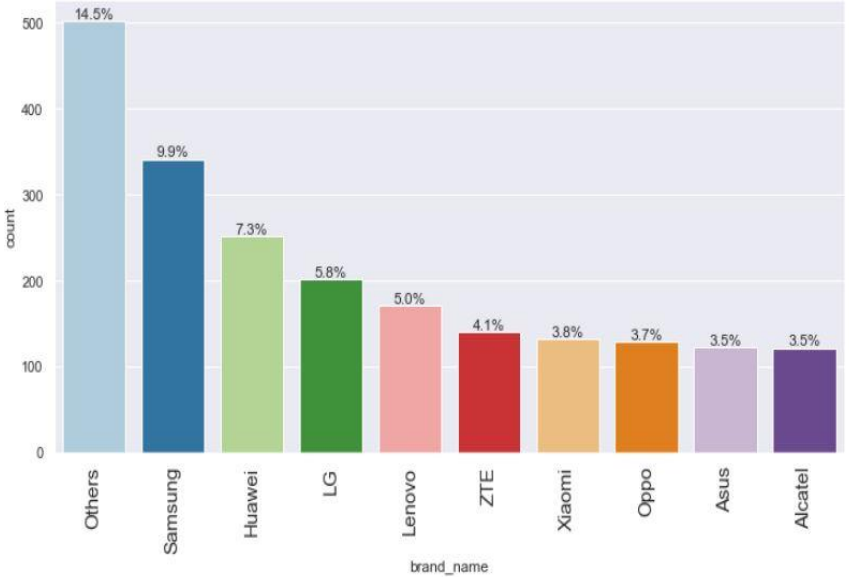
Weight variable has a mean of 182g with a maximum of about 885g. The distribution of the weight variable is slightly right skewed since the mean is greater than the median

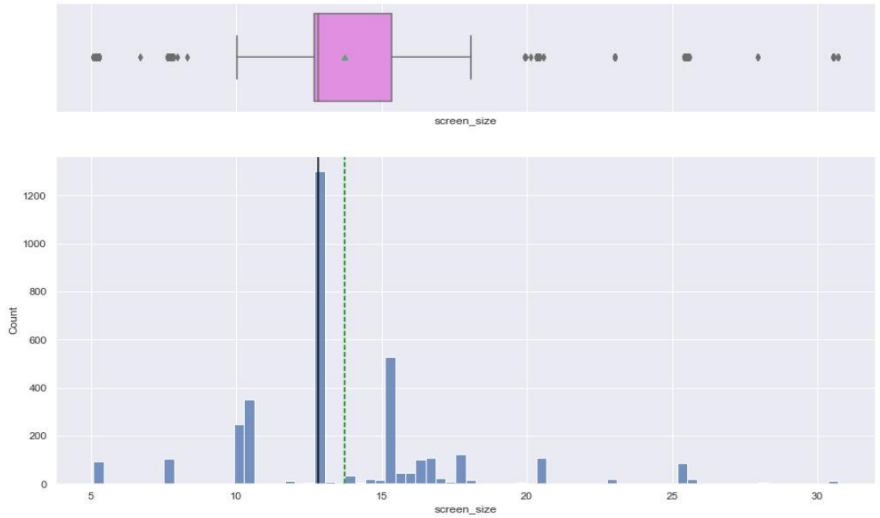




EDA Results

- Univariate Analysis: brand_name on the left and screen_size on the right

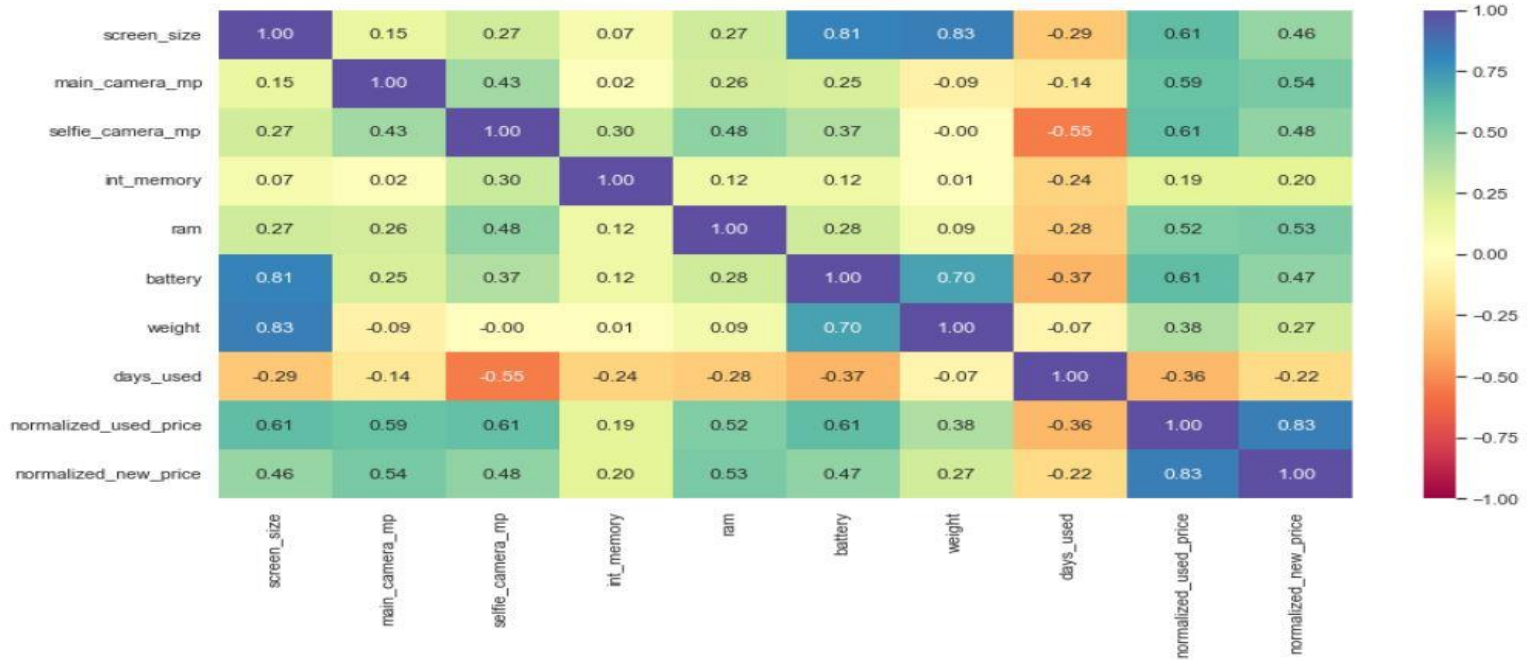




- We notice outliers and missing values in the screen-size variable pushing the mean to the right of the median
- The others category in brand_name has the highest count followed by Samsung while Alcaltel has the least count

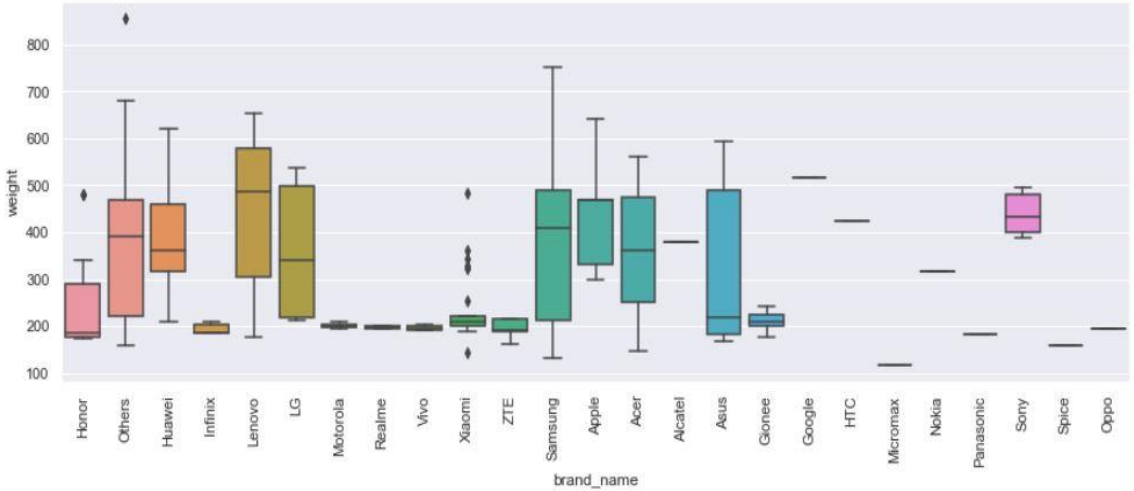
EDA Results

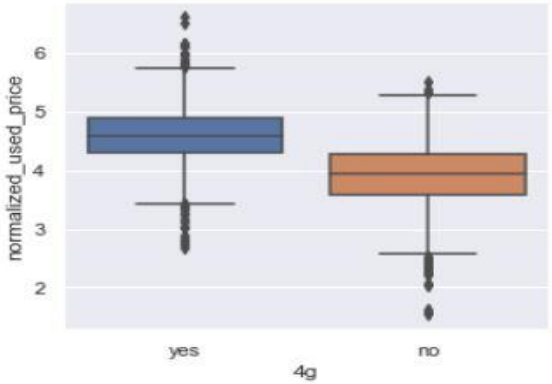
- Bivariate Analysis: Correlation map

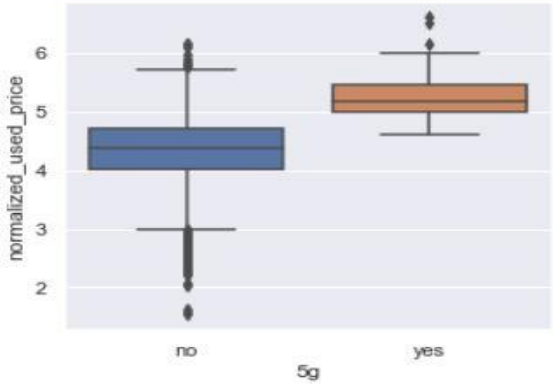


- Weight and screen_size are the most highly correlated with a value of 0.83.
- Battery and screen_size are relatively highly correlated with a value of 0.81
- Battery and weight are moderately to highly correlated with a value of 0.70

EDA Results:Bivariate Analysis



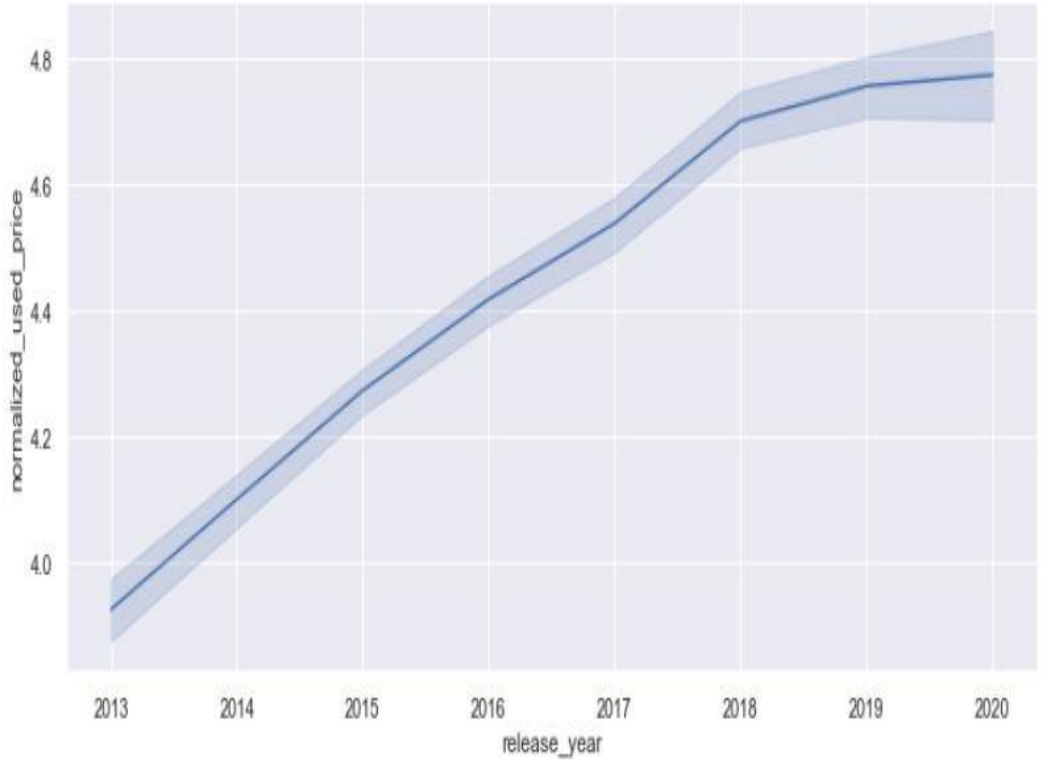




- Lenovo has the highest median of all brands in the category of frequent travelers with devices which need heavier batteries while Micromax have the smallest.

- 4g and 5g network both have higher medians indicating most phone and tablet users have accessibility to use 4g and 5g networks. 5g has users do have the lowest range of all users since the technology was very new in 2021

EDA Results:Bivariate Analysis



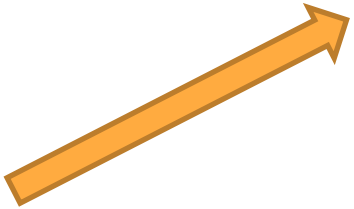
- Pattern of the line plot shows price of used devices increasing from 2013 to 2020
- The growth is rapid and appears to slow down around 2020

Data Preprocessing

- *Duplicate value check*

Out[167]:

brand_name	0
os	0
screen_size	0
4g	0
5g	0
main_camera_mp	179
selfie_camera_mp	2
int_memory	4
ram	4
battery	6
weight	7
release_year	0
days_used	0
normalized_used_price	0
normalized_new_price	0
dtype: int64	



brand_name	0
os	0
screen_size	0
4g	0
5g	0
main_camera_mp	179
selfie_camera_mp	2
int_memory	0
ram	0
battery	6
weight	7
release_year	0
days_used	0
normalized_used_price	0
normalized_new_price	0
dtype: int64	

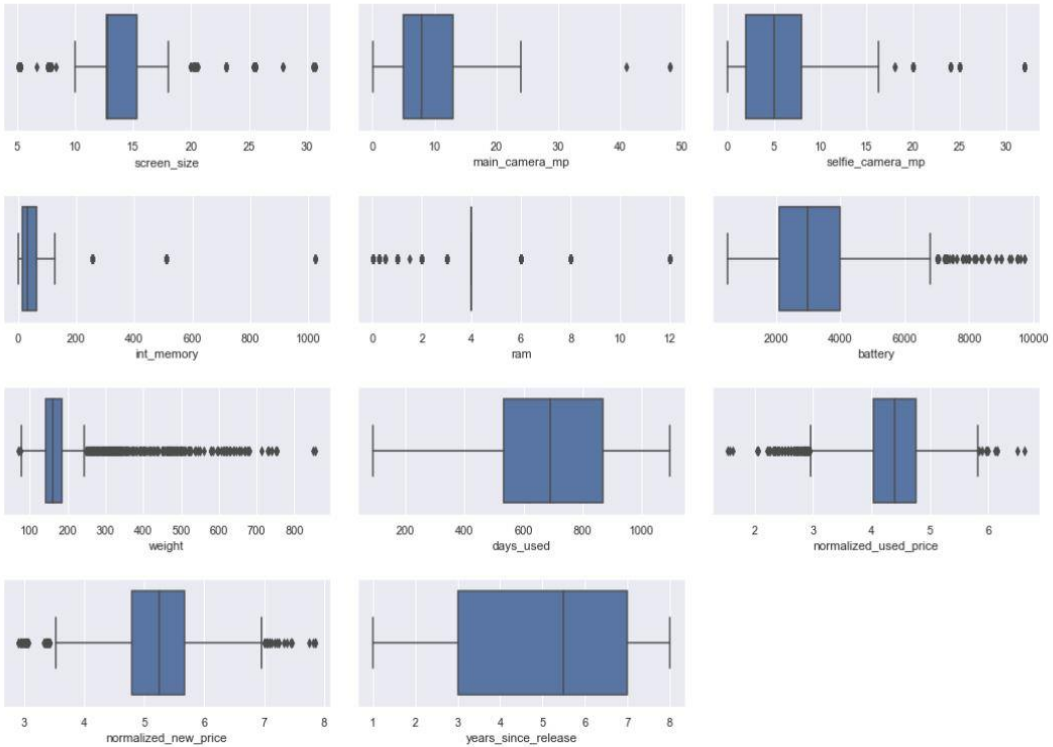
Out[217]:

brand_name	0
os	0
screen_size	0
4g	0
5g	0
main_camera_mp	0
selfie_camera_mp	0
int_memory	0
ram	0
battery	0
weight	0
release_year	0
days_used	0
normalized_used_price	0
normalized_new_price	0
dtype: int64	

- Missing and duplicated were imputed in respective columns with median by grouping the data on release_year and brand_name

Feature Engineering

- Outlier check



- As expected, there are many outliers in the dataset.
- Imputing 179 missing values that were imputed in the main camera would certainly inflate corresponding values of those in the selfie camera leading to more outliers.

- Although outliers are observable, they do not appear to be influential data points but rather represent the natural variation in the dataset
- There was no need to treat outliers since data imputation

Data Preparation for modeling

```
  brand_name       os  screen_size   4g   5g  main_camera_mp  \
0      Honor  Android        14.50  yes   no            13.0
1      Honor  Android        17.30  yes  yes            13.0
2      Honor  Android        16.69  yes  yes            13.0
3      Honor  Android        25.50  yes  yes            13.0
4      Honor  Android        15.32  yes   no            13.0

   selfie_camera_mp  int_memory  ram  battery  weight  days_used  \
0               5.0        64.0  3.0   3020.0   146.0        127
1              16.0       128.0  8.0   4300.0   213.0        325
2               8.0       128.0  8.0   4200.0   213.0        162
3               8.0        64.0  6.0   7250.0   480.0        345
4               8.0        64.0  3.0   5000.0   185.0        293

   normalized_new_price  years_since_release
0              4.715100                    1
1              5.519018                    1
2              5.884631                    1
3              5.630961                    1
4              4.947837                    1

0    4.307572
1    5.162097
2    5.111084
3    5.135387
4    4.389995
Name: normalized_used_price, dtype: float64
```

- The table below with Out[222] shows the dummy variables generated using one hot encoding
- The second table shows the list of independent variables with column titles on the top while the only dependent variable is the normalized_used_price with its column title in the bottom

Out[222]:

rice	...	brand_name_Spice	brand_name_Vivo	brand_name_XOLO	brand_name_Xiaomi	brand_name_ZTE	os_Others	os_Windows	os_iOS	4g_yes	5g_yes
100	...	0	0	0	0	0	0	0	0	1	0
018	...	0	0	0	0	0	0	0	0	1	1
631	...	0	0	0	0	0	0	0	0	1	1
961	...	0	0	0	0	0	0	0	0	1	1
837	...	0	0	0	0	0	0	0	0	1	0

Data modeling

OLS Regression Results

Dep. Variable:	normalized_used_price	R-squared:	0.845
Model:	OLS	Adj. R-squared:	0.842
Method:	Least Squares	F-statistic:	268.7
Date:	Fri, 07 Oct 2022	Prob (F-statistic):	0.00
Time:	15:42:24	Log-Likelihood:	123.85
No. Observations:	2417	AIC:	-149.7
Df Residuals:	2368	BIC:	134.0
Df Model:	48		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	1.3156	0.071	18.454	0.000	1.176	1.455
screen_size	0.0244	0.003	7.163	0.000	0.018	0.031
main_camera_mp	0.0208	0.002	13.848	0.000	0.018	0.024
selfie_camera_mp	0.0135	0.001	11.997	0.000	0.011	0.016
int_memory	0.0001	6.97e-05	1.651	0.099	-2.16e-05	0.000
ram	0.0230	0.005	4.451	0.000	0.013	0.033
battery	-1.689e-05	7.27e-06	-2.321	0.020	-3.12e-05	-2.62e-06
weight	0.0010	0.000	7.480	0.000	0.001	0.001
days_used	4.216e-05	3.09e-05	1.366	0.172	-1.84e-05	0.000
normalized_new_price	0.4311	0.012	35.147	0.000	0.407	0.455
years_since_release	-0.0237	0.005	-5.193	0.000	-0.033	-0.015
brand_name_Alcatel	0.0154	0.048	0.323	0.747	-0.078	0.109
brand_name_Apple	-0.0038	0.147	-0.026	0.980	-0.292	0.285
brand_name_Asus	0.0151	0.048	0.314	0.753	-0.079	0.109
brand_name_BlackBerry	-0.0300	0.070	-0.427	0.669	-0.168	0.108
brand_name_Celkon	-0.0468	0.066	-0.707	0.480	-0.177	0.083
brand_name_Coolpad	0.0209	0.073	0.287	0.774	-0.122	0.164
brand_name_Gionee	0.0448	0.058	0.775	0.438	-0.068	0.158
brand_name_Google	-0.0326	0.085	-0.385	0.700	-0.199	0.133
brand_name_HTC	-0.0130	0.048	-0.270	0.787	-0.108	0.081
brand_name_Honor	0.0317	0.049	0.644	0.520	-0.065	0.128
brand_name_Huawei	-0.0020	0.044	-0.046	0.964	-0.089	0.085
brand_name_Infinix	0.1633	0.093	1.752	0.080	-0.019	0.346
brand_name_Karbonn	0.0943	0.067	1.405	0.160	-0.037	0.226
brand_name_LG	-0.0132	0.045	-0.291	0.771	-0.102	0.076
brand_name_Lava	0.0332	0.062	0.533	0.594	-0.089	0.155
brand_name_Lenovo	0.0454	0.045	1.004	0.316	-0.043	0.134
brand_name_Meizu	-0.0129	0.056	-0.230	0.818	-0.123	0.097
brand_name_Micromax	-0.0337	0.048	-0.704	0.481	-0.128	0.060
brand_name_Microsoft	0.0952	0.088	1.078	0.281	-0.078	0.268
brand_name_Motorola	-0.0112	0.050	-0.226	0.821	-0.109	0.086
brand_name_Nokia	0.0719	0.052	1.387	0.166	-0.030	0.174
brand_name_OnePlus	0.0709	0.077	0.916	0.360	-0.081	0.223
brand_name_Oppo	0.0124	0.048	0.261	0.794	-0.081	0.106
brand_name_Others	-0.0080	0.042	-0.190	0.849	-0.091	0.075
brand_name_Panasonic	0.0563	0.056	1.008	0.314	-0.053	0.166
brand_name_Realme	0.0319	0.062	0.518	0.605	-0.089	0.153
brand_name_Samsung	-0.0313	0.043	-0.725	0.469	-0.116	0.053
brand_name_Sony	-0.0616	0.050	-1.220	0.223	-0.161	0.037
brand_name_Spice	-0.0147	0.063	-0.233	0.816	-0.139	0.109
brand_name_Vivo	-0.0154	0.048	-0.318	0.750	-0.110	0.080
brand_name_XOLO	0.0152	0.055	0.277	0.782	-0.092	0.123
brand_name_Xiaomi	0.0869	0.048	1.806	0.071	-0.007	0.181
brand_name_ZTE	-0.0057	0.047	-0.121	0.904	-0.099	0.087
os_Others	-0.0510	0.033	-1.555	0.120	-0.115	0.013
os_Windows	-0.0207	0.045	-0.459	0.646	-0.109	0.068
os_iOS	-0.0663	0.146	-0.453	0.651	-0.354	0.221
4g_yes	0.0528	0.016	3.326	0.001	0.022	0.084
5g_yes	-0.0714	0.031	-2.268	0.023	-0.133	-0.010

- Model 1 summary with statistical summary, p_values and adjusted r squared values

Data modeling

Out[242]:

	feature	VIF
0	const	227.744081
1	screen_size	7.677290
2	main_camera_mp	2.285051
3	selfie_camera_mp	2.812473
4	int_memory	1.364152
5	ram	2.282352
6	battery	4.081780
7	weight	6.396749
8	days_used	2.660269
9	normalized_new_price	3.119430
10	years_since_release	4.899007
11	brand_name_Alcatel	3.405693
12	brand_name_Apple	13.057668
13	brand_name_Asus	3.332038
14	brand_name_BlackBerry	1.632378
15	brand_name_Celkon	1.774721
16	brand_name_Coolpad	1.468006
17	brand_name_Gionee	1.951272
18	brand_name_Google	1.321778
19	brand_name_HTC	3.410361
20	brand_name_Honor	3.340687
21	brand_name_Huawei	5.983852
22	brand_name_Infinix	1.283955
23	brand_name_Karbonn	1.573702
24	brand_name_LG	4.849832
25	brand_name_Lava	1.711360
26	brand_name_Lenovo	4.558941
27	brand_name_Meizu	2.179607
28	brand_name_Micromax	3.363521
29	brand_name_Microsoft	1.869751
30	brand_name_Motorola	3.274558
31	brand_name_Nokia	3.479849
32	brand_name_OnePlus	1.437034
32	brand_name_OnePlus	1.437034
33	brand_name_Oppo	3.971194
34	brand_name_Others	9.711034
35	brand_name_Panasonic	2.105703
36	brand_name_Realme	1.946812
37	brand_name_Samsung	7.539866
38	brand_name_Sony	2.943161
39	brand_name_Spice	1.688863
40	brand_name_Vivo	3.651437
41	brand_name_XOLO	2.138070
42	brand_name_Xiaomi	3.719689
43	brand_name_ZTE	3.797581
44	os_Others	1.859863
45	os_Windows	1.596034
46	os_iOS	11.784684
47	4g_yes	2.467681
48	5g_yes	1.813900

- Of all numeric variables, only weight and screen_size had moderate to high vifs
- I decided to remove screen_size since it was fairly collinear with weight and checked p-values after dropping screen_size

Summary of most important factors

- Both variables do not change the adjusted R squared value by a significant amount.
- Since the adjusted R squared of model 1 is 0.8416, I decided to remove screen_size because it is moderately-highly correlated to weight and appear to carry the same information when predicting price

Out[247]:

	col	Adj. R-squared after_dropping col	RMSE after dropping col
0	screen_size	0.838381	0.234703
1	weight	0.838071	0.234928

Link to Appendix slide on model assumptions

Model Performance

- Overview of model performance

The tables below show the result of the model performance on the training and test set after dropping screen_size variable.

```
Training Performance
```

Out[227]:

	RMSE	MAE	R-squared	Adj. R-squared	MAPE
0	0.229884	0.180326	0.844886	0.841675	4.326841

```
Test Performance
```

Out[230]:

	RMSE	MAE	R-squared	Adj. R-squared	MAPE
0	0.238358	0.184749	0.842479	0.834659	4.501651

Link to Appendix slide on model assumptions

Summary of final model

OLS Regression Results

Dep. Variable:	normalized_used_price	R-squared:	0.839
Model:	OLS	Adj. R-squared:	0.838
Method:	Least Squares	F-statistic:	895.7
Date:	Fri, 07 Oct 2022	Prob (F-statistic):	0.00
Time:	18:26:55	Log-Likelihood:	80.645
No. Observations:	2417	AIC:	-131.3
Df Residuals:	2402	BIC:	-44.44
Df Model:	14		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	1.5000	0.048	30.955	0.000	1.405	1.595
main_camera_mp	0.0210	0.001	14.714	0.000	0.018	0.024
selfie_camera_mp	0.0138	0.001	12.858	0.000	0.012	0.016
ram	0.0207	0.005	4.151	0.000	0.011	0.030
weight	0.0017	6e-05	27.672	0.000	0.002	0.002
normalized_new_price	0.4415	0.011	39.337	0.000	0.419	0.463
years_since_release	-0.0292	0.003	-8.589	0.000	-0.036	-0.023
brand_name_Karbonn	0.1156	0.055	2.111	0.035	0.008	0.223
brand_name_Samsung	-0.0374	0.016	-2.270	0.023	-0.070	-0.005
brand_name_Sony	-0.0670	0.030	-2.197	0.028	-0.127	-0.007
brand_name_Xiaomi	0.0801	0.026	3.114	0.002	0.030	0.130
os_Others	-0.1276	0.027	-4.667	0.000	-0.181	-0.074
os_iOS	-0.0900	0.045	-1.994	0.046	-0.179	-0.002
4g_yes	0.0502	0.015	3.326	0.001	0.021	0.080
5g_yes	-0.0673	0.031	-2.194	0.028	-0.127	-0.007

Omnibus:	246.183	Durbin-Watson:	1.902
Prob(Omnibus):	0.000	Jarque-Bera (JB):	483.879
Skew:	-0.658	Prob(JB):	8.45e-106
Kurtosis:	4.753	Cond. No.	2.39e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.39e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

Link to Appendix slide on model assumptions

Summary of final model

Training and Testing performance on the final model

```
Training Performance
```

Out[281]:

	RMSE	MAE	R-squared	Adj. R-squared	MAPE
0	0.23403	0.182751	0.83924	0.838235	4.395407

```
Test Performance
```

Out[283]:

	RMSE	MAE	R-squared	Adj. R-squared	MAPE
0	0.241434	0.186649	0.838387	0.836013	4.556349

final model

OLS Regression Results

Dep. Variable:	normalized_used_price	R-squared:	0.839
Model:	OLS	Adj. R-squared:	0.838
Method:	Least Squares	F-statistic:	895.7
Date:	Fri, 07 Oct 2022	Prob (F-statistic):	0.00
Time:	18:26:55	Log-Likelihood:	80.645
No. Observations:	2417	AIC:	-131.3
Df Residuals:	2402	BIC:	-44.44
Df Model:	14		
Covariance Type:	nonrobust		

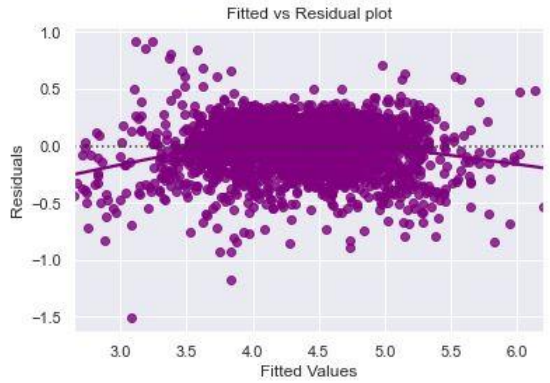
	coef	std err	t	P>|t|	[0.025	0.975]
const	1.5000	0.048	30.955	0.000	1.405	1.595
main_camera_mp	0.0210	0.001	14.714	0.000	0.018	0.024
selfie_camera_mp	0.0138	0.001	12.858	0.000	0.012	0.016
ram	0.0207	0.005	4.151	0.000	0.011	0.030
weight	0.0017	6e-05	27.672	0.000	0.002	0.002
normalized_new_price	0.4415	0.011	39.337	0.000	0.419	0.463
years_since_release	-0.0292	0.003	-8.589	0.000	-0.036	-0.023
brand_name_Karbonn	0.1156	0.055	2.111	0.035	0.008	0.223
brand_name_Samsung	-0.0374	0.016	-2.270	0.023	-0.070	-0.005
brand_name_Sony	-0.0670	0.030	-2.197	0.028	-0.127	-0.007
brand_name_Xiaomi	0.0801	0.026	3.114	0.002	0.030	0.130
os_Others	-0.1276	0.027	-4.667	0.000	-0.181	-0.074
os_iOS	-0.0900	0.045	-1.994	0.046	-0.179	-0.002
4g_yes	0.0502	0.015	3.326	0.001	0.021	0.080
5g_yes	-0.0673	0.031	-2.194	0.028	-0.127	-0.007

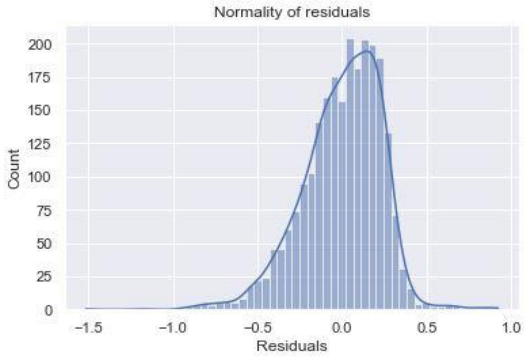
Omnibus:	246.183	Durbin-Watson:	1.902
Prob(Omnibus):	0.000	Jarque-Bera (JB):	483.879
Skew:	-0.658	Prob(JB):	8.45e-106
Kurtosis:	4.753	Cond. No.	2.39e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.39e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

Link to Appendix slide on model assumptions

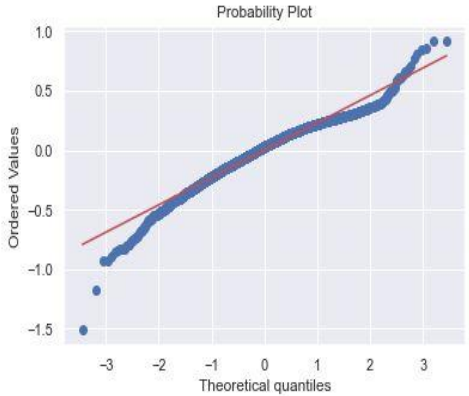
Linear Regression assumptions






```
Out[270]: ShapiroResult(statistic=0.9676972031593323, pvalue=6.995328206686811e-23)
```

```
Out[272]: [('F statistic', 1.0087504199106763), ('p-value', 0.4401970650667071)]
```



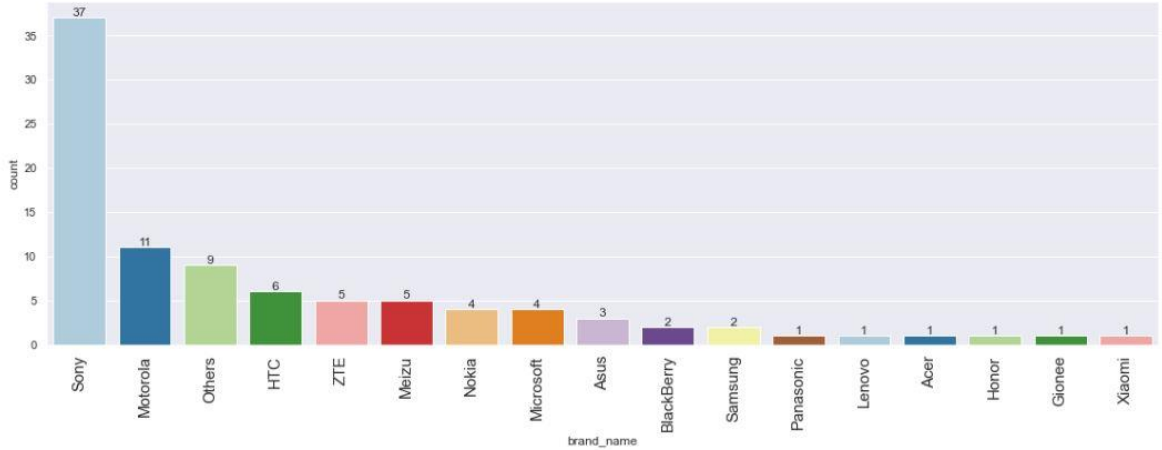
Link to Appendix slide on model assumptions

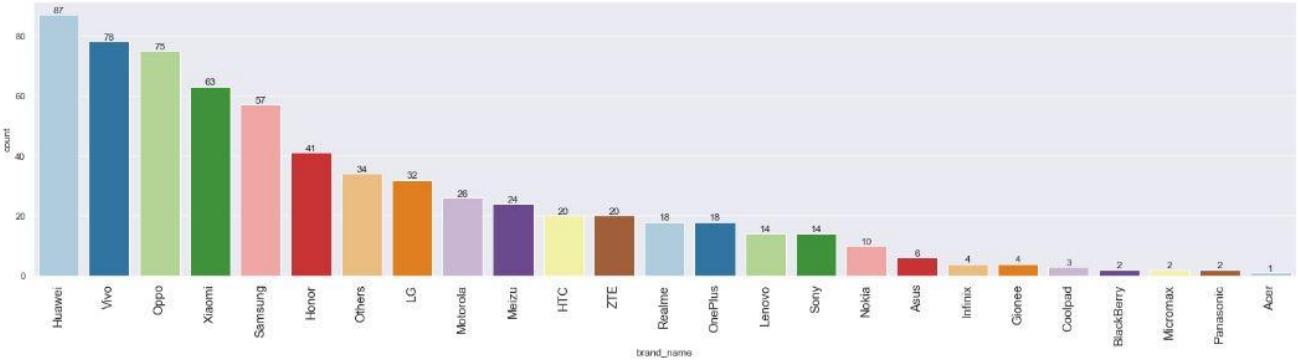
Linear Regression assumptions

- No proper pattern noticeable on the normal plot of residuals vs fitted values so independence can be assumed
- The Residuals plot is bell-shaped although not centered and the q-q plot does not seem to capture the tails which is expected by the number of outliers we noticed in the dataset. The p value of the Shapiro test is less than 0.05 which means strictly speaking residuals are not normal. However, we can approximate near normality since the outliers in the data set inflate the values making selfie camera mp much higher than main camera mp which isn't common. Assumption of normality have been violated, but the outliers in the dataset and imputation allows us to assume normality
- P value of the goldfeldquandt test is greater than 0.05 which means the assumption of constant variance or homoscedasticity is valid

Appendix

- olsmodel_final = sm.OLS(y_train,x_train_final).fit() ## the code to fit the final model : print(olsmodel_final.summary())





Great Learning

Happy Learning !