Supervised Learning Project 3 – Recell

Presenter: Joshua Willis, PGP-DSBA

Background

The used and refurbished device market has grown considerably over the past decade, and a 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. This growth can be attributed to an uptick in demand for used phones and tablets that offer considerable savings compared with new models.

Objective

The rising potential of this comparatively under-the-radar market fuels the need for an ML-based solution to develop a dynamic pricing strategy for used and refurbished devices. ReCell, a startup aiming to tap the potential in this market. The purpose of this exercise is to analyze given data and build a linear regression model which predicts the price of a used phone/tablet and identify factors that significantly influences it.

Data Dictionary

brand_name	Name of manufacturing brand
os:	OS on which the device runs
Screen_size	Size of the screen in cm
4g	Whether 4G is available or not
5g	Whether 5G is available or not
main_camera_mp	Resolution of the rear camera in megapixels
selfie_camera_mp	Resolution of the front camera in megapixels
int_memory	Amount of internal memory (ROM) in GB
ram	Amount of RAM in GB
battery	Energy capacity of the device battery in mAh
weight	weight
release_year	release_year
days_used	days_used
normalized_new_price	normalized_new_price
normalized_used_price	normalized_used_price

Data Overview

Data Shape: *3,454 Rows & 15 Columns*

Data Type By Column:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3454 entries, 0 to 3453
Data columns (total 15 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   brand_name             3454 non-null   object
 1   os                     3454 non-null   object
 2   screen_size            3454 non-null   float64
 3   4g                     3454 non-null   object
 4   5g                     3454 non-null   object
 5   main_camera_mp         3275 non-null   float64
 6   selfie_camera_mp       3452 non-null   float64
 7   int_memory             3450 non-null   float64
 8   ram                    3450 non-null   float64
 9   battery                3448 non-null   float64
 10  weight                 3447 non-null   float64
 11  release_year           3454 non-null   int64
 12  days_used              3454 non-null   int64
 13  normalized_used_price  3454 non-null   float64
 14  normalized_new_price   3454 non-null   float64
dtypes: float64(9), int64(2), object(4)
```

Columns with Missing Values:

```
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
```

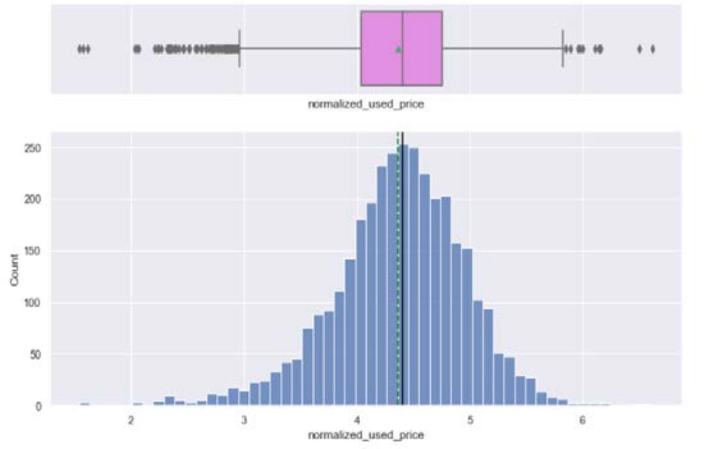
Data Overview (cont.)

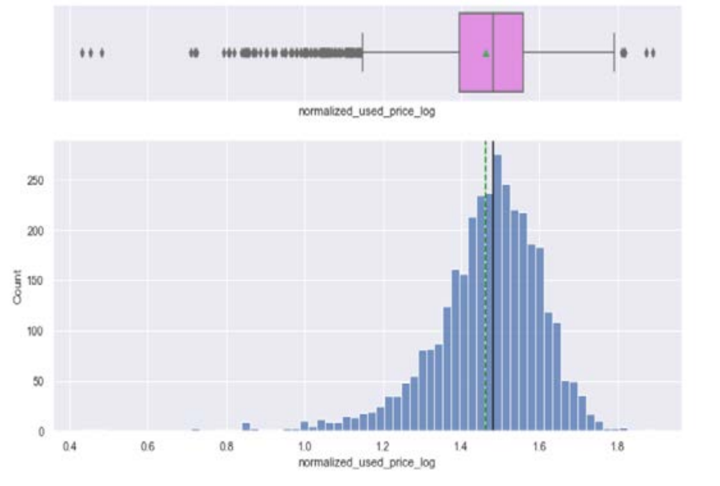
Statistical Info:

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
brand_name	3454	34	Others	502	NaN	NaN	NaN	NaN	NaN	NaN	NaN
os	3454	4	Android	3214	NaN	NaN	NaN	NaN	NaN	NaN	NaN
screen_size	3454.0	NaN	NaN	NaN	13.713115	3.80528	5.08	12.7	12.83	15.34	30.71
4g	3454	2	yes	2335	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5g	3454	2	no	3302	NaN	NaN	NaN	NaN	NaN	NaN	NaN
main_camera_mp	3275.0	NaN	NaN	NaN	9.460208	4.815461	0.08	5.0	8.0	13.0	48.0
selfie_camera_mp	3452.0	NaN	NaN	NaN	6.554229	6.970372	0.0	2.0	5.0	8.0	32.0
int_memory	3450.0	NaN	NaN	NaN	54.573099	84.972371	0.01	16.0	32.0	64.0	1024.0
ram	3450.0	NaN	NaN	NaN	4.036122	1.365105	0.02	4.0	4.0	4.0	12.0
battery	3448.0	NaN	NaN	NaN	3133.402697	1299.682844	500.0	2100.0	3000.0	4000.0	9720.0
weight	3447.0	NaN	NaN	NaN	182.751871	88.413228	69.0	142.0	160.0	185.0	855.0
release_year	3454.0	NaN	NaN	NaN	2015.965258	2.298455	2013.0	2014.0	2015.5	2018.0	2020.0
days_used	3454.0	NaN	NaN	NaN	674.869716	248.580166	91.0	533.5	690.5	868.75	1094.0
normalized_used_price	3454.0	NaN	NaN	NaN	4.364712	0.588914	1.536867	4.033931	4.405133	4.7557	6.619433
normalized_new_price	3454.0	NaN	NaN	NaN	5.233107	0.683637	2.901422	4.790342	5.245892	5.673718	7.847841

Exploratory Data Analysis

Univariate
Data Analysis

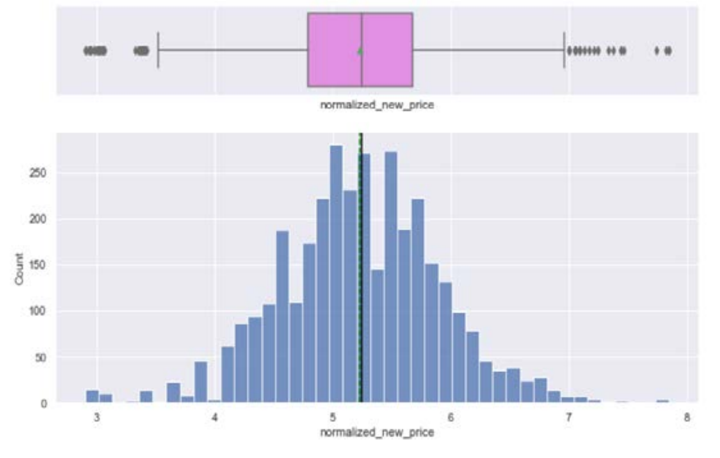


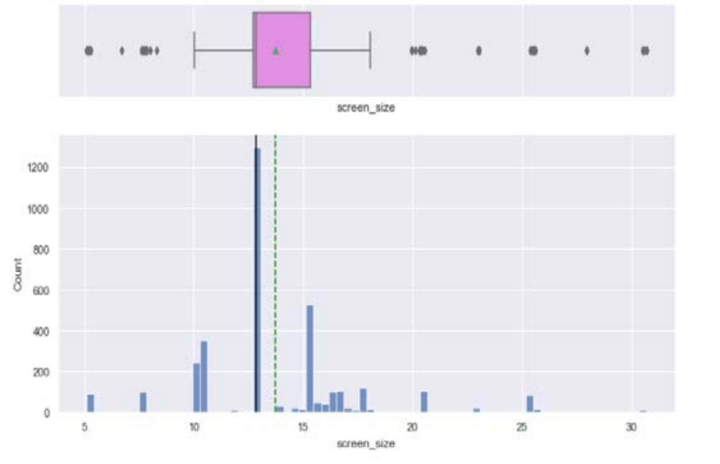


- *normalized_used_price appears to be normally distributed with more outliers outside of the left whisker versus the right whisker.*

- *normalized_used_price log was created to slightly reduce slight skewness on the left.*

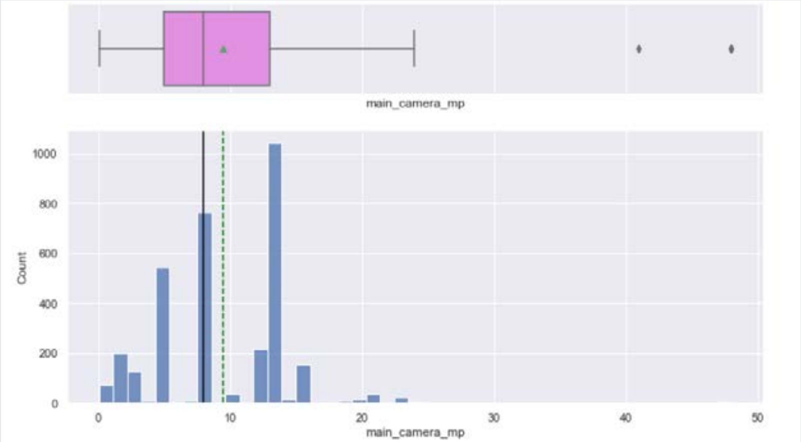
Univariate
Data Analysis

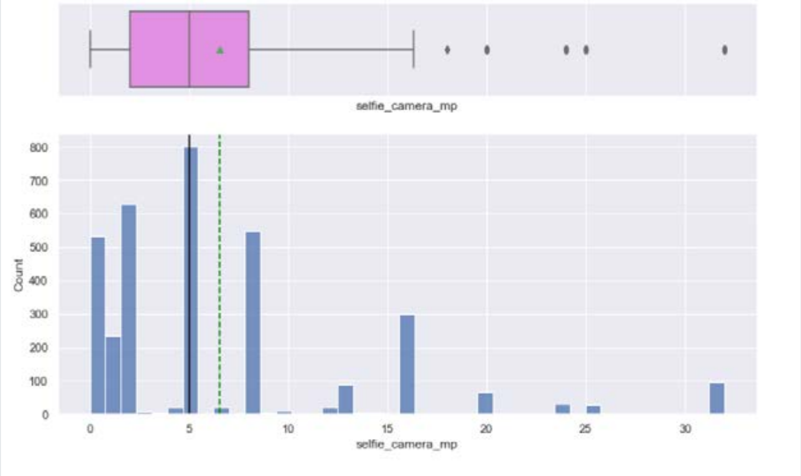




- *normalized_new_price resembles normal distribution with outliers*
- *screen_size has data that is skewed right with outliers*

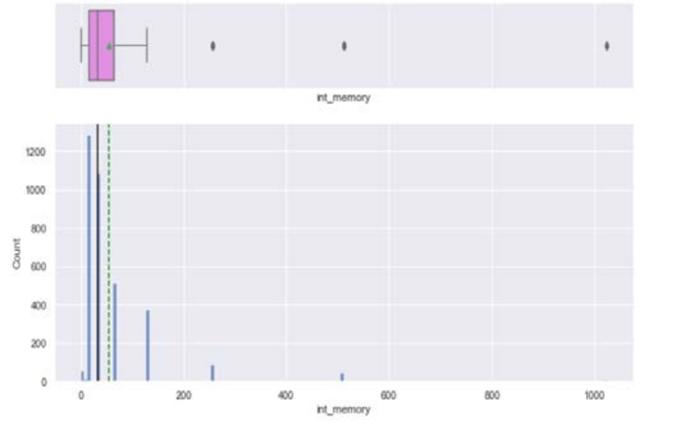
Univariate
Data Analysis

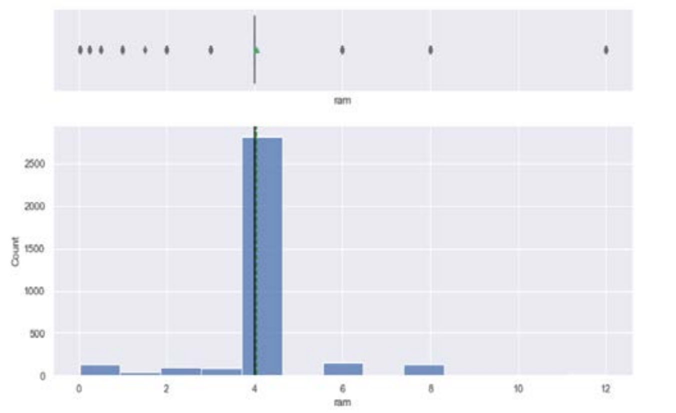




- *main_camera_mp is slightly skewed right with a few outliers*
- *selfie_camera_mp is normally distributed with a few outliers*

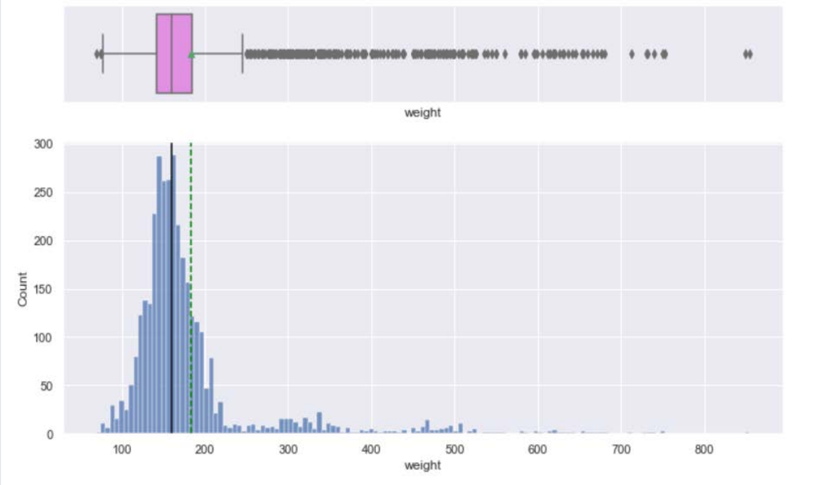
Univariate Data Analysis

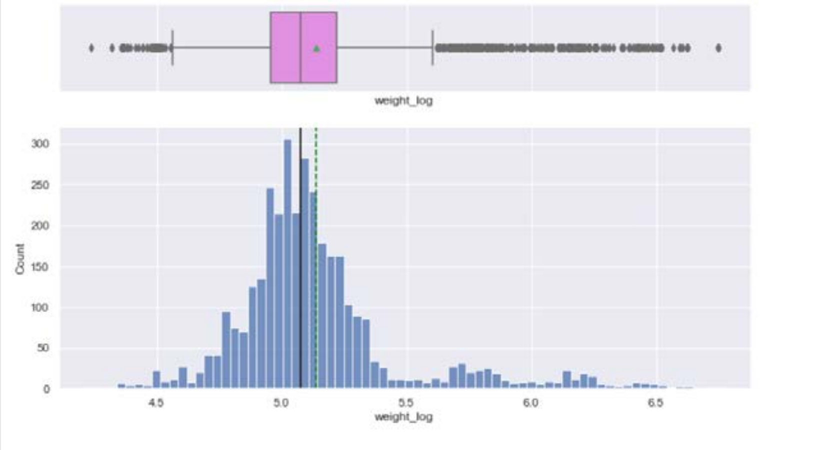




- *int_memory is right skewed with outliers beyond upper limit*
- *ram data is mostly centered around median with outliers beyond lower and upper limits*

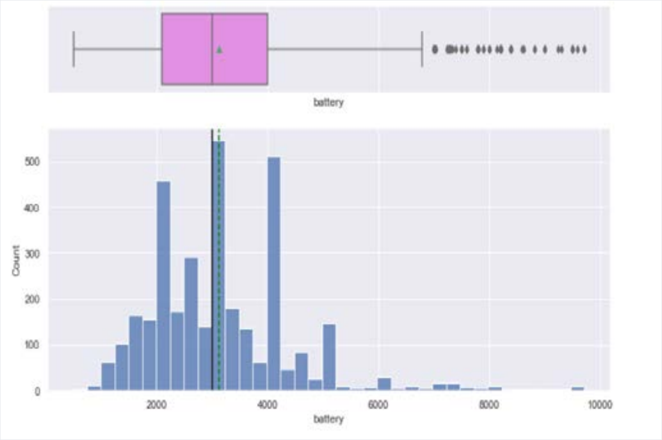
Univariate
Data Analysis

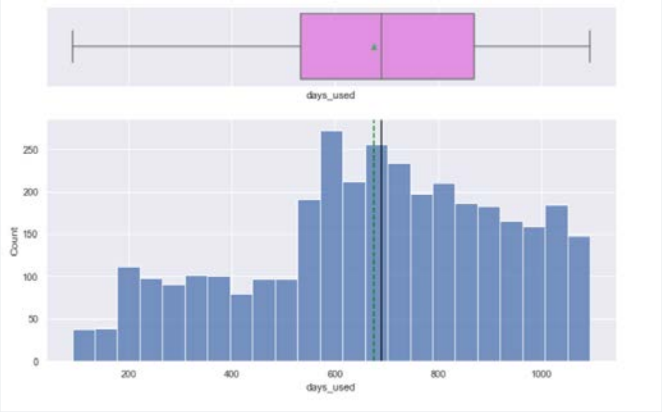




- *weight is close to normal distribution but has a lot of outliers beyond upper limit*
- *weight_log variable is created to make distribution of weight closer to normal*

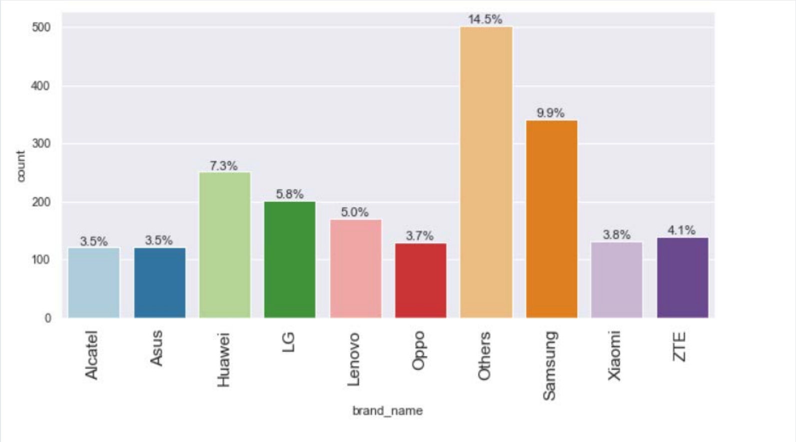
Univariate
Data Analysis

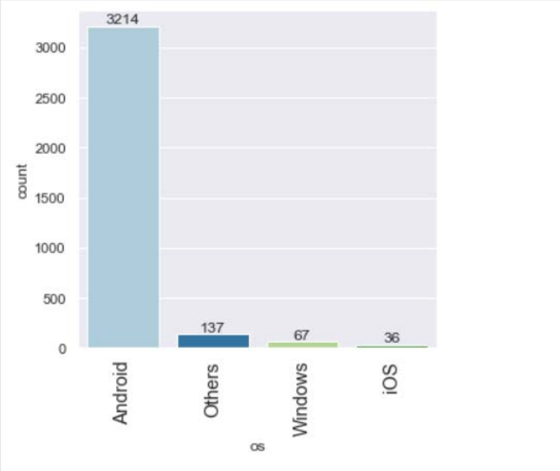




- *battery is normally distributed with outliers beyond upper limit*
- *days_used somewhat normally distributed with no outliers*

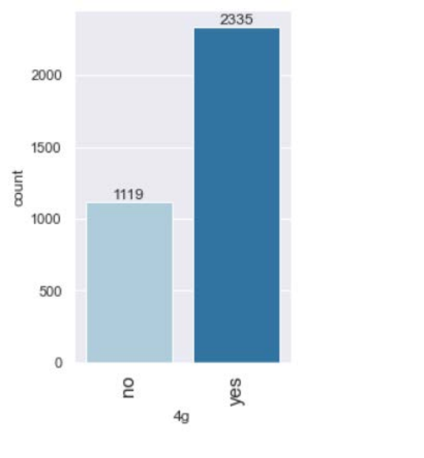
Univariate Data Analysis

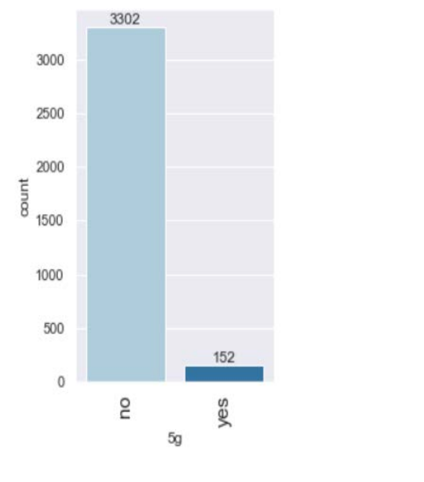


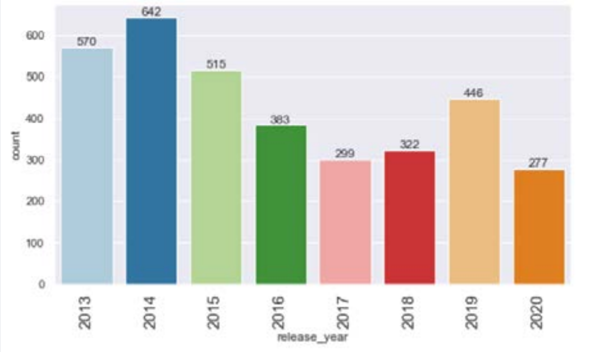


- *The most purchased brands are “Others”, “Samsung”, and “Huawei”*
- *The most purchased os system is “Android”*

Univariate
Data Analysis







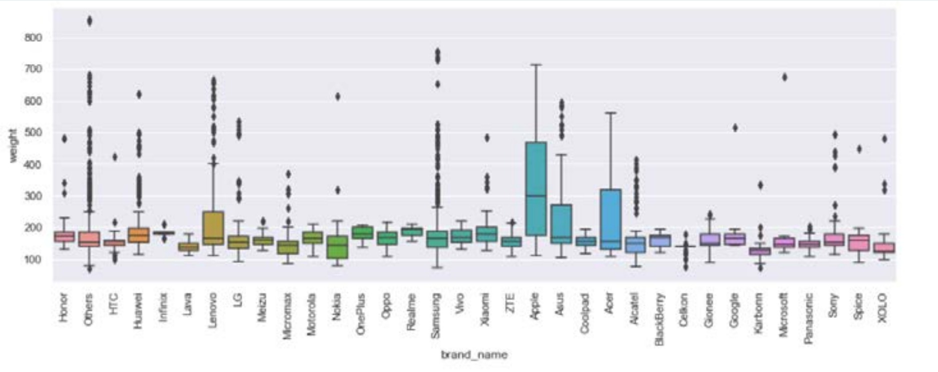
- *Most phones are 4g*
- *The release year with the most phones are 2014, 2015, and 2013 respectfully*

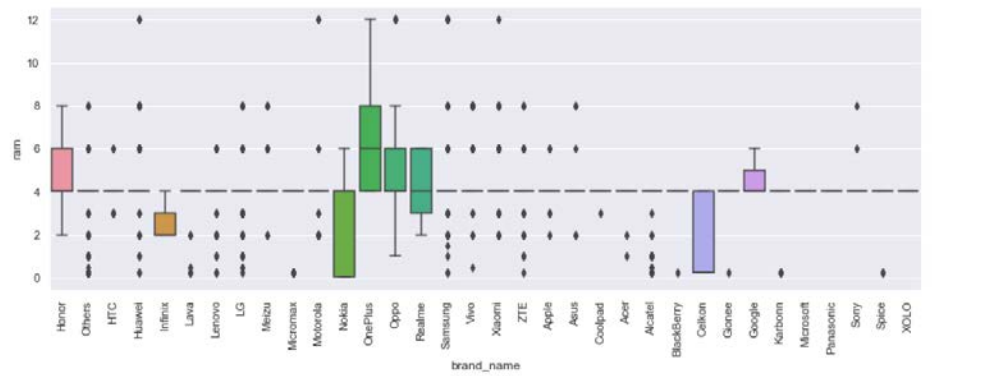
Bivariate Analysis
Data Analysis

Variables with high correlation:

- *screen_size vs. battery, weight/weight_log*

Bivariate Analysis
Data Analysis

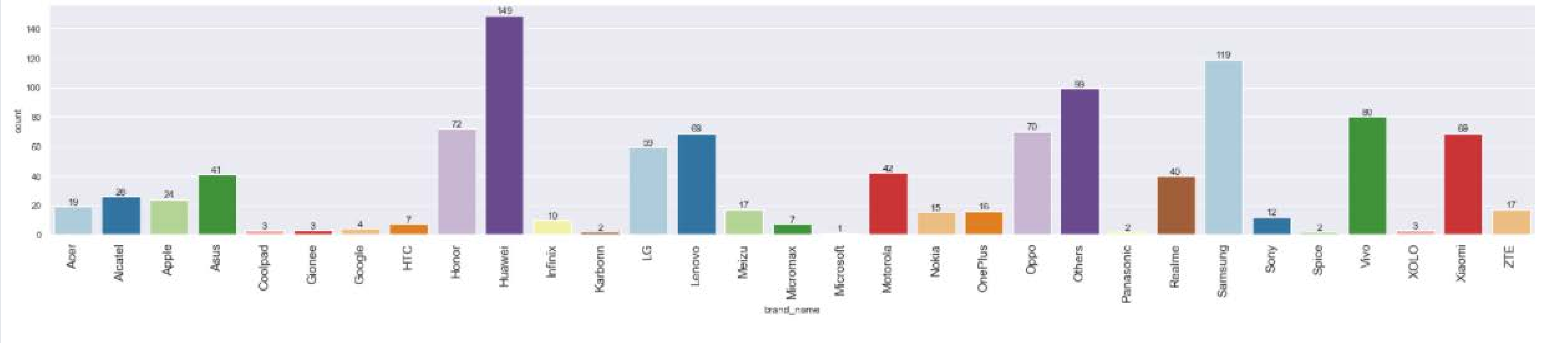




- *The brands with the heavier phones were Apple, Acer, and Lenovo*
- *The brands with highest ram were OnePlus, Honor, and Oppo*

Bivariate Analysis

Data Analysis

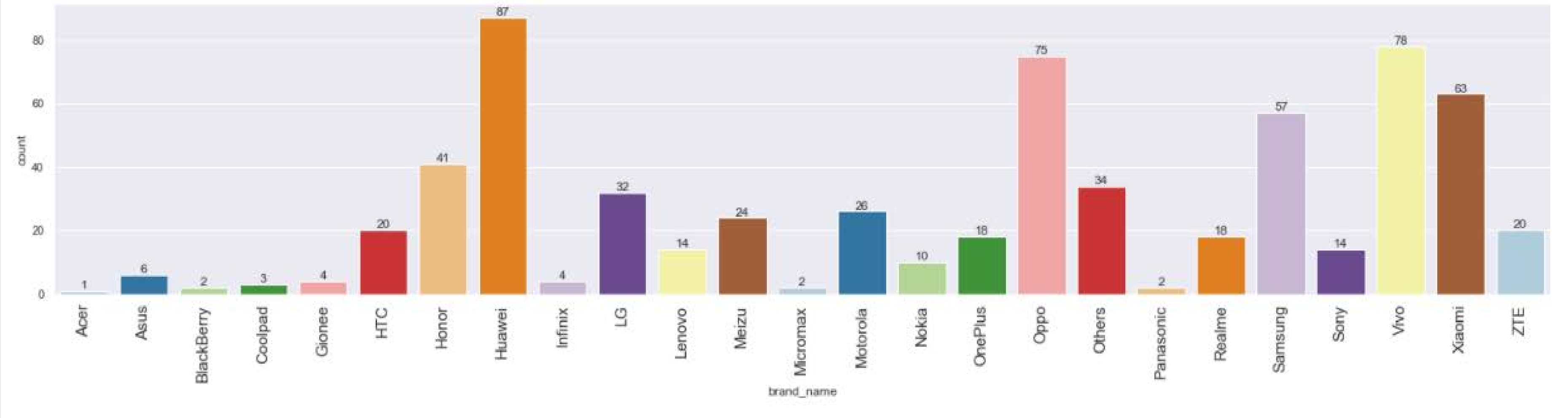


The brands with screen sizes larger than 4500 cm were:

- *Huawei*
- *Samsung*
- *Others*

Bivariate Analysis

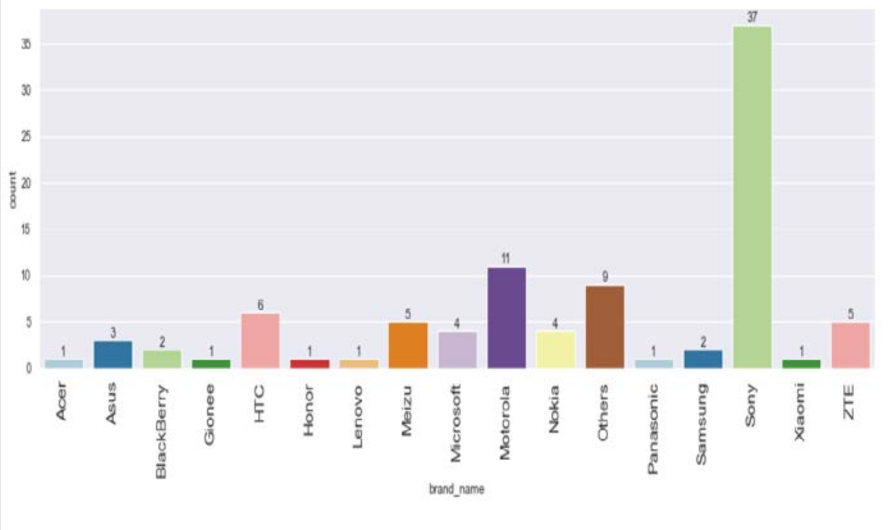
Data Analysis

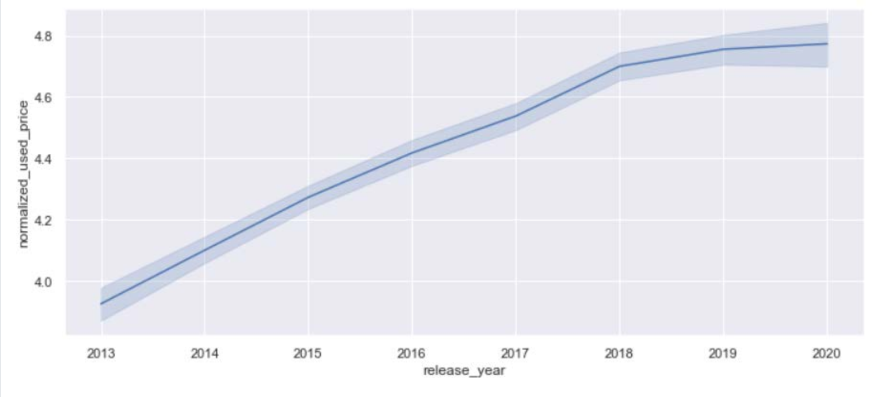


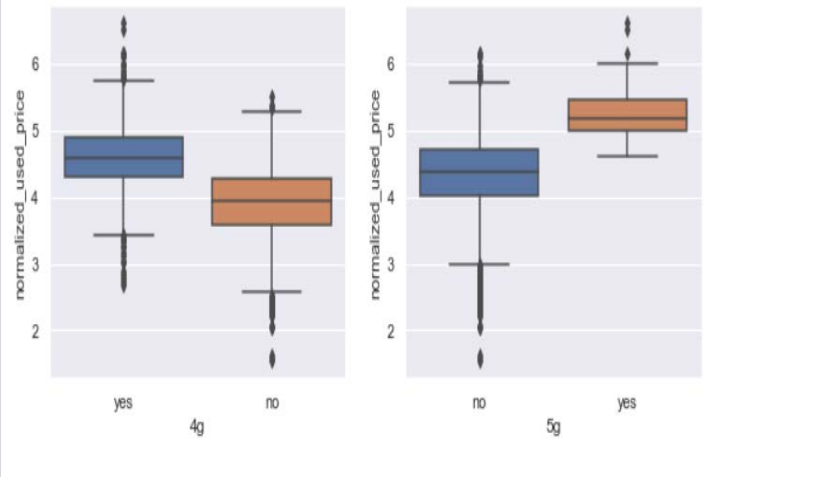
The cell phones with selfie camera that has more than 8 mp:

- *Huawei*
- *Vivo*
- *Oppo*

Bivariate Analysis
Data Analysis







- *Sony, Motorola, and Others were brands that had the most phones with main camera larger than 16 mp*
- *Normalized_used_price has consistently increased over the last 7 years*
- *The normalized_used_price for 5g is higher than 4g*

Data Preprocessing

Data Preprocessing

Missing Value Imputation / Feature Engineering

*Data **Before** Imputation*

```
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
normalized_used_price_log    0
weight_log                   7
dtype: int64
```

*Data **After** Imputation*

```
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
release_year                 0
days_used                    0
normalized_new_price         0
normalized_used_price_log    0
weight_log                   0
dtype: int64
```

years_since_release (new column)

```
count    3454.000000
mean        5.034742
std         2.298455
min         1.000000
25%         3.000000
50%         5.500000
75%         7.000000
max         8.000000
Name: years_since_release, dtype: float64
```

The following columns need to be filled with data to address missing values

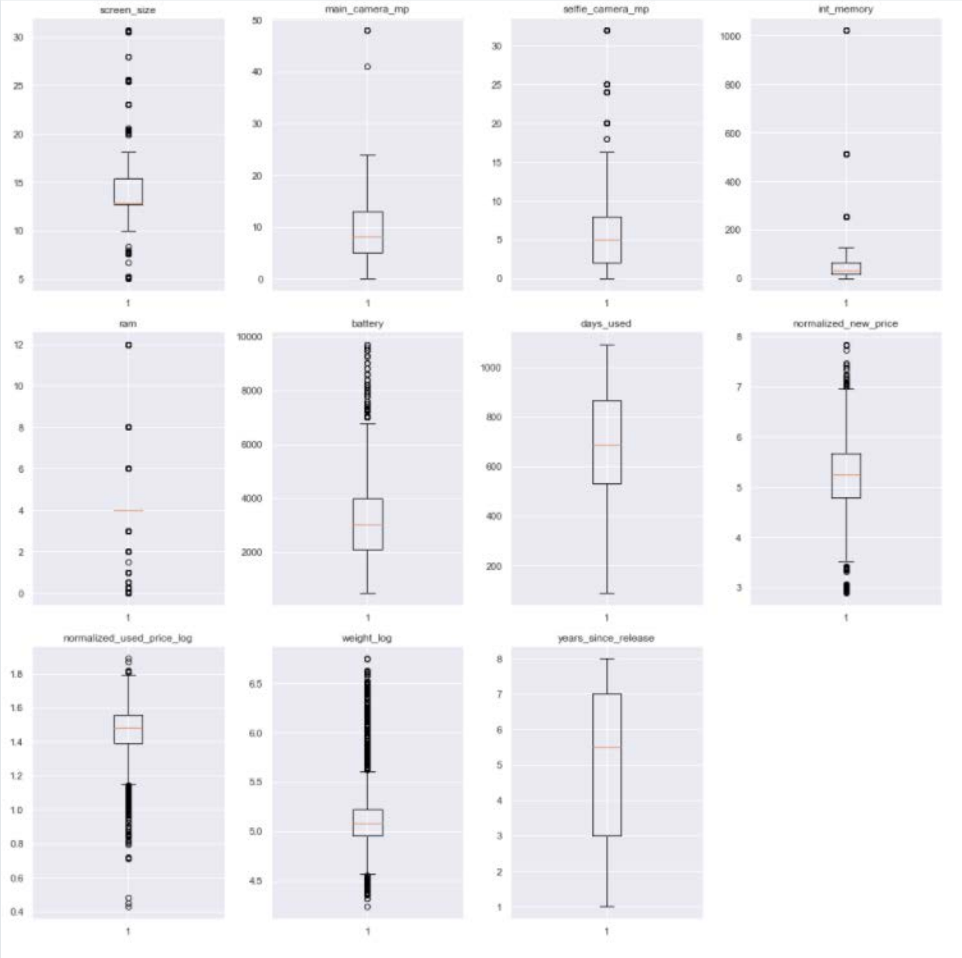
- main_camera_mp
- selfie_camera_mp
- int_memory
- ram
- battery
- weight
- weight_log

✓ ***The above were filled with the median and checked again to ensure no missing values remained***

✓ ***A column called years_since_release has been created and release_year has been dropped.***

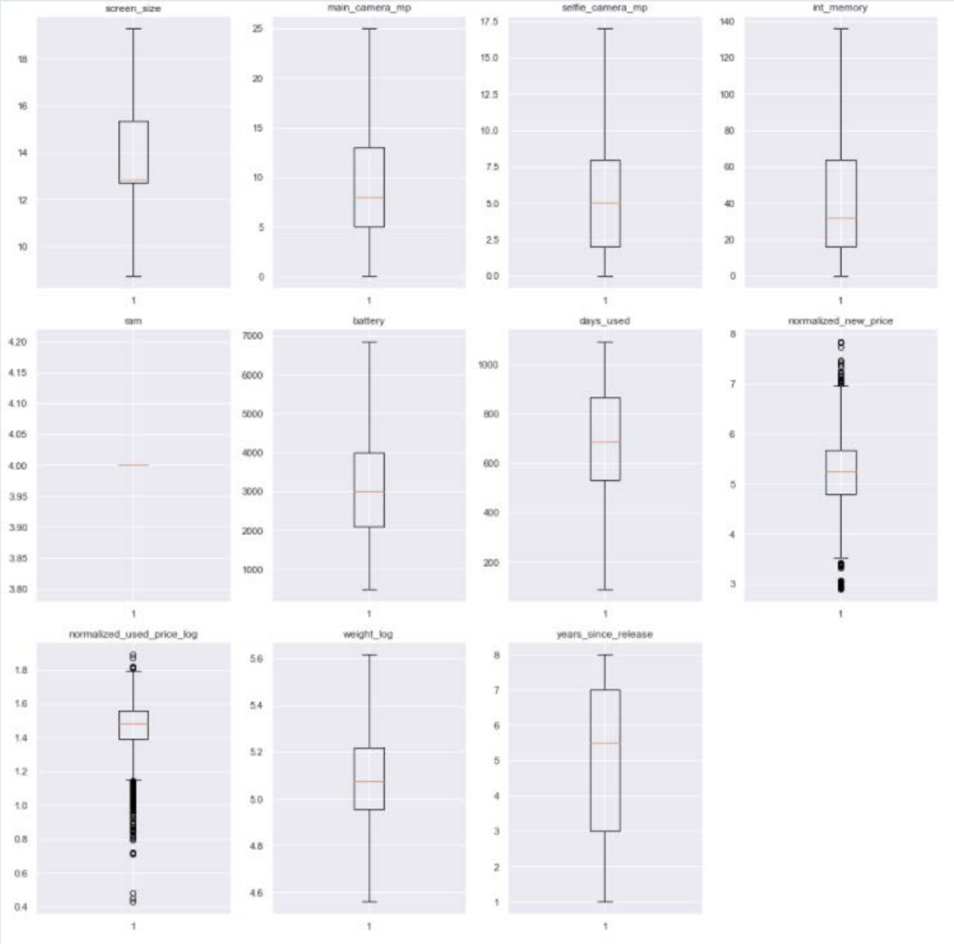
✓ ***The years_since_release column was calculated using year 2021 minus the release_year.***

Data Preprocessing
Outlier Check



- *All variables except days_used and years_since_release have outliers*
- *ram has min, quartile ranges, and max that is very similar*
- *normalized_new_price and normalized_used_price_log are discrete variables and will not be treated for outliers in order to maintain range of data*

Data Preprocessing
Outlier Treatment



- *All outliers have been treated except normalized_new_price and normalized_used_price_log*

Data Preprocessing

Data Prep & Modeling

OLS Regression Results

Dep. Variable:	normalized_used_price_log	R-squared:	0.822
Model:	OLS	Adj. R-squared:	0.819
Method:	Least Squares	F-statistic:	233.5
Date:	Wed, 30 Mar 2022	Prob (F-statistic):	0.00
Time:	02:19:34	Log-Likelihood:	3353.7
No. Observations:	2417	AIC:	-6611.
Df Residuals:	2369	BIC:	-6333.
Df Model:	47		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
screen_size	0.0097	0.001	9.533	0.000	0.008	0.012
main_camera_mp	0.0053	0.000	12.347	0.000	0.004	0.006
selfie_camera_mp	0.0039	0.000	8.440	0.000	0.003	0.005
int_memory	-1.824e-05	5.03e-05	-0.363	0.717	-0.000	8.04e-05
ram	0.0930	0.012	8.049	0.000	0.070	0.116
battery	-2.159e-06	1.95e-06	-1.109	0.267	-5.97e-06	1.66e-06
days_used	1.43e-05	8.11e-06	1.764	0.078	-1.6e-06	3.02e-05
normalized_new_price	0.1032	0.003	31.164	0.000	0.097	0.110
weight_log	0.0701	0.011	6.568	0.000	0.049	0.091
years_since_release	-0.0025	0.001	-1.998	0.046	-0.005	-4.7e-05
brand_name_Alcatel	-9.711e-05	0.013	-0.008	0.994	-0.025	0.024
brand_name_Apple	0.0863	0.038	2.246	0.025	0.011	0.162
brand_name_Asus	-0.0033	0.013	-0.266	0.791	-0.028	0.021
brand_name_BlackBerry	0.0169	0.019	0.914	0.361	-0.019	0.053
brand_name_Celkon	-0.0384	0.017	-2.201	0.028	-0.073	-0.004
brand_name_Coolpad	0.0032	0.019	0.168	0.866	-0.034	0.041
brand_name_Gionee	0.0003	0.015	0.017	0.986	-0.029	0.030
brand_name_Google	-0.0111	0.022	-0.501	0.616	-0.055	0.032
brand_name_HTC	-0.0060	0.013	-0.476	0.634	-0.031	0.019
brand_name_Honor	0.0043	0.013	0.330	0.742	-0.021	0.030
brand_name_Huawei	-0.0055	0.012	-0.472	0.637	-0.028	0.017
brand_name_Infinix	0.0240	0.024	0.983	0.326	-0.024	0.072
brand_name_Karbonn	0.0179	0.018	1.014	0.311	-0.017	0.053
brand_name_LG	-0.0044	0.012	-0.367	0.714	-0.028	0.019
brand_name_Lava	0.0048	0.016	0.295	0.768	-0.027	0.037
brand_name_Lenovo	0.0075	0.012	0.629	0.529	-0.016	0.031
brand_name_Meizu	-0.0067	0.015	-0.454	0.650	-0.036	0.022
brand_name_Micromax	-0.0239	0.013	-1.902	0.057	-0.049	0.001
brand_name_Microsoft	0.0193	0.023	0.834	0.405	-0.026	0.065
brand_name_Motorola	-0.0039	0.013	-0.303	0.762	-0.029	0.022
brand_name_Nokia	0.0210	0.014	1.550	0.121	-0.006	0.048
brand_name_OnePlus	-0.0056	0.020	-0.274	0.784	-0.046	0.034
brand_name_Oppo	0.0022	0.013	0.172	0.863	-0.022	0.027
brand_name_Others	-0.0065	0.011	-0.592	0.554	-0.028	0.015
brand_name_Panasonic	0.0145	0.015	0.989	0.323	-0.014	0.043
brand_name_Realme	-0.0101	0.016	-0.623	0.533	-0.042	0.022
brand_name_Samsung	-0.0069	0.011	-0.610	0.542	-0.029	0.015
brand_name_Sony	-0.0204	0.013	-1.538	0.124	-0.047	0.006
brand_name_Spice	-0.0135	0.017	-0.812	0.417	-0.046	0.019
brand_name_Vivo	-0.0132	0.013	-1.037	0.300	-0.038	0.012
brand_name_XOLO	0.0067	0.014	0.462	0.644	-0.022	0.035
brand_name_Xiaomi	0.0097	0.013	0.769	0.442	-0.015	0.035
brand_name_ZTE	-0.0027	0.012	-0.215	0.829	-0.027	0.022
os_Others	-0.0591	0.008	-7.296	0.000	-0.075	-0.043
os_Windows	0.0021	0.012	0.174	0.862	-0.021	0.025
os_iOS	-0.1062	0.038	-2.790	0.005	-0.181	-0.032
4g_yes	0.0091	0.004	2.175	0.030	0.001	0.017
5g_yes	-0.0101	0.007	-1.372	0.170	-0.024	0.004

Omnibus:	833.868	Durbin-Watson:	1.948
Prob(Omnibus):	0.000	Jarque-Bera (JB):	8774.862
Skew:	-1.320	Prob(JB):	0.00
Kurtosis:	11.953	Cond. No.	1.73e+05

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.73e+05. This might indicate that there are strong multicollinearity or other numerical problems.

Training Performance

	RMSE	MAE	R-squared	Adj. R-squared	MAPE
0	0.060417	0.044445	0.822435	0.818835	3.234354

Test Performance

	RMSE	MAE	R-squared	Adj. R-squared	MAPE
0	0.065029	0.046708	0.814156	0.805127	3.518508

- *Dependent and independent variables established for model*
- *Dummy variables assigned for independent variables*
- *Data split into 70:30 ratio for train to test data*
- *Train data rows = 2,417*
- *Test data rows = 1,037*

Next step will be to test for linear assumptions.

Checking Linear Assumptions

- ✓ No Multicollinearity
- ✓ Linearity of variables
- ✓ Independence of error terms
- ✓ Normality of error terms
- ✓ No Heteroscedasticity

No Multicollinearity

- *If VIF is 1 then there is no correlation between the kth predictor and the remaining predictor variables.*
- ***If VIF exceeds 5 or is close to exceeding 5, we say there is moderate multicollinearity.***
- *If VIF is 10 or exceeding 10, it shows signs of high multicollinearity.*

Training Performance

	RMSE	MAE	R-squared	Adj. R-squared	MAPE
0	0.060417	0.044445	0.822435	0.818835	3.234354

Test Performance

	RMSE	MAE	R-squared	Adj. R-squared	MAPE
0	0.065029	0.046708	0.814156	0.805127	3.518508

	feature	VIF
0	screen_size	5.358249
1	main_camera_mp	2.473792
2	selfie_camera_mp	3.847589
3	int_memory	2.493624
4	ram	1386.915875
5	battery	3.696971
6	days_used	2.658056
7	normalized_new_price	3.293840
8	weight_log	4.576046
9	years_since_release	5.416757
10	brand_name_Alcatel	3.407253
11	brand_name_Apple	12.890960
12	brand_name_Asus	3.335329
13	brand_name_BlackBerry	1.649100
14	brand_name_Celkon	1.782415
15	brand_name_Coolpad	1.468772
16	brand_name_Gionee	1.950169
17	brand_name_Google	1.322409
18	brand_name_HTC	3.413504
19	brand_name_Honor	3.348326
20	brand_name_Huawei	5.988217
21	brand_name_Infinix	1.278526
22	brand_name_Karbonn	1.576535
23	brand_name_LG	4.848224
24	brand_name_Lava	1.708548
25	brand_name_Lenovo	4.555392
26	brand_name_Meizu	2.185740
27	brand_name_Micromax	3.361677
28	brand_name_Microsoft	1.867365
29	brand_name_Motorola	3.269240
30	brand_name_Nokia	3.449315
31	brand_name_OnePlus	1.440405
32	brand_name_Oppo	3.958808
33	brand_name_Others	9.714260
34	brand_name_Panasonic	2.107060
35	brand_name_Realme	1.944994
36	brand_name_Samsung	7.550549
37	brand_name_Sony	2.956516
38	brand_name_Spice	1.693160
39	brand_name_Vivo	3.665894
40	brand_name_XOLO	2.144148
41	brand_name_Xiaomi	3.730691
42	brand_name_ZTE	3.797729
43	os_Others	1.639247
44	os_Windows	1.594513
45	os_iOS	11.538803
46	4g_yes	2.498034
47	5g_yes	1.424002

- *Dropped every column one by one that had a VIF score greater than 5.*
- *Looked at the adjusted R-squared and RMSE of all the models.*
- *Dropped the variable that made the least change in adjusted R-squared.*
- *Checked the VIF scores again and got all but one of the scores under 5.*

Variables with VIF over 5 were:

- screen_size
- ram
- years_since_release
- brand_name_Apple
- brand_name_Huawei
- brand_name_Others
- brand_name_Samsung
- os_IOS

Variables dropped in order to reduce VIF of all variables below 5:

- os_iOS
- brand_name_Huawei
- years_since_release
- screen_size

Ram variable remained with a VIF over 5 despite several variable drops.

No Multicollinearity (P-values)

```
                            OLS Regression Results
==============================================================================
Dep. Variable:     normalized_used_price_log   R-squared:                 0.811
Model:                              OLS   Adj. R-squared:                 0.810
Method:                   Least Squares   F-statistic:                    862.1
Date:                  Wed, 30 Mar 2022   Prob (F-statistic):              0.00
Time:                          02:19:38   Log-Likelihood:                3281.1
No. Observations:                  2417   AIC:                           -6536.
Df Residuals:                      2404   BIC:                           -6461.
Df Model:                            12
Covariance Type:              nonrobust
======================================================================================
                         coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------------
main_camera_mp         0.0055      0.000     13.689      0.000       0.005       0.006
selfie_camera_mp       0.0047      0.000     14.394      0.000       0.004       0.005
ram                    0.0410      0.010      4.261      0.000       0.022       0.060
battery              5.54e-06   1.79e-06      3.097      0.002    2.03e-06    9.05e-06
normalized_new_price   0.0989      0.003     37.578      0.000       0.094       0.104
weight_log             0.1333      0.008     16.088      0.000       0.117       0.150
brand_name_Celkon     -0.0431      0.014     -3.141      0.002      -0.070      -0.016
brand_name_Lenovo      0.0114      0.006      1.973      0.049    7.06e-05       0.023
brand_name_Micromax   -0.0230      0.007     -3.165      0.002      -0.037      -0.009
brand_name_Nokia       0.0280      0.008      3.453      0.001       0.012       0.044
brand_name_Sony       -0.0198      0.008     -2.440      0.015      -0.036      -0.004
os_Others             -0.0653      0.007     -8.746      0.000      -0.080      -0.051
4g_yes                 0.0113      0.004      3.026      0.003       0.004       0.019
==============================================================================
Omnibus:                      804.022   Durbin-Watson:                   1.963
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             7551.716
Skew:                          -1.296   Prob(JB):                         0.00
Kurtosis:                      11.263   Cond. No.                     3.65e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.65e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Training Performance

	RMSE	MAE	R-squared	Adj. R-squared	MAPE
0	0.062259	0.045791	0.811441	0.810421	3.334939

Test Performance

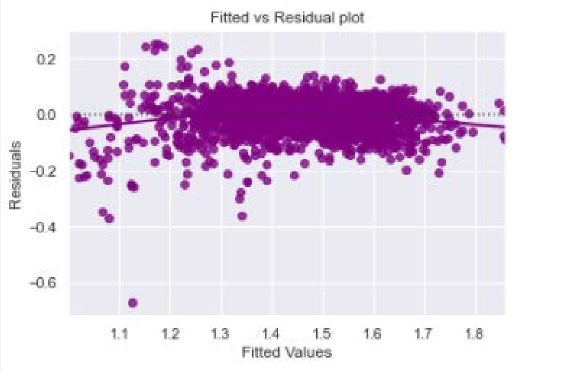
	RMSE	MAE	R-squared	Adj. R-squared	MAPE
0	0.066303	0.047602	0.8068	0.804345	3.588689

- *The predictor variables having a p-value greater than 0.05 were dropped as they do not significantly impact the target variable.*
- *But sometimes p-values change after dropping a variable. So, the variables were not all dropped at once; built model, checked p-value, then dropped the one with highest p-value one at a time.*

Regression results after p-values over 0.05 have been dropped

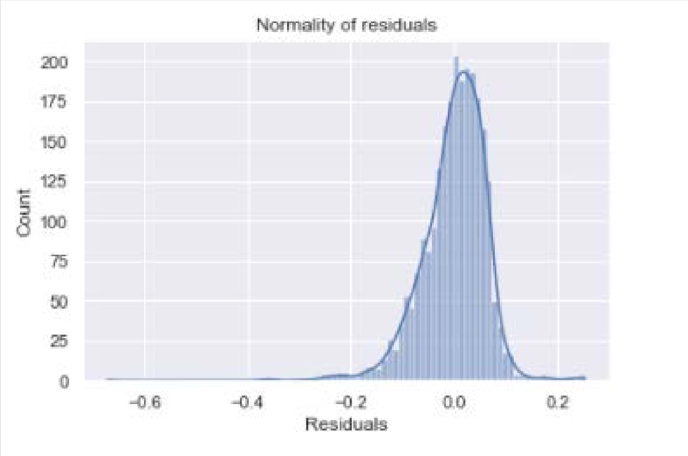
Test for Linearity and Independence

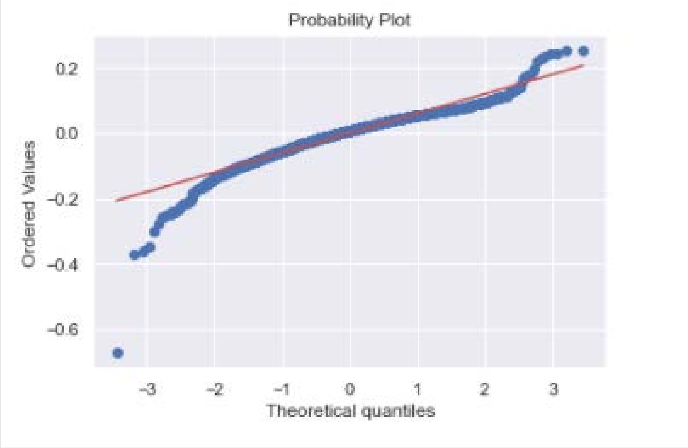
	Actual Values	Fitted Values	Residuals
3026	1.407931	1.332180	0.075750
1525	1.492544	1.563807	-0.071262
1128	1.462179	1.459702	0.002477
3003	1.454436	1.418452	0.035984
2907	1.494350	1.510436	-0.016086



- *Conducted test for linearity and independence by making a plot of fitted values vs residuals and checking for patterns.*
- *Since there is no pattern, then the model is linear, and residuals are independent.*

Test for Normality





- *Test for normality by checking the distribution of residuals, Q-Q plot of residuals, and the Shapiro-Wilk test*
- *The residuals follow a normal distribution*
- *The residuals make a reasonable straight-line plot*
- *The p-value of the Shapiro-Wilk test is greater than 0.05*

```
ShapiroResult(statistic=0.9271098375320435, pvalue=9.287473601987381e-33)
```

Test for Homoscedasticity

```
[('F statistic', 1.0000237544777306), ('p-value', 0.4998329861096695)]
```

- *Assess homoscedasticity by using the goldfeldquandt test.*
- *If p-value is greater than 0.05, then residuals are homoscedastic, otherwise heteroscedastic.*

Test result is homoscedastic

Final Model Summary

Final Model Summary

```
olsmodel_final = sm.OLS(
    y_train, x_train6
).fit()  ## Complete the code to fit the final model
print(olsmodel_final.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:     normalized_used_price_log   R-squared:                       0.811
Model:                                   OLS   Adj. R-squared:                  0.810
Method:                        Least Squares   F-statistic:                     862.1
Date:                       Wed, 30 Mar 2022   Prob (F-statistic):               0.00
Time:                               02:19:41   Log-Likelihood:                 3281.1
No. Observations:                       2417   AIC:                            -6536.
Df Residuals:                           2404   BIC:                            -6461.
Df Model:                                 12
Covariance Type:                   nonrobust
==============================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
main_camera_mp        0.0055      0.000     13.689      0.000       0.005       0.006
selfie_camera_mp      0.0047      0.000     14.394      0.000       0.004       0.005
ram                   0.0410      0.010      4.261      0.000       0.022       0.060
battery             5.54e-06   1.79e-06      3.097      0.002    2.03e-06    9.05e-06
normalized_new_price  0.0989      0.003     37.578      0.000       0.094       0.104
weight_log            0.1333      0.008     16.088      0.000       0.117       0.150
brand_name_Celkon    -0.0431      0.014     -3.141      0.002      -0.070      -0.016
brand_name_Lenovo     0.0114      0.006      1.973      0.049    7.06e-05       0.023
brand_name_Micromax  -0.0230      0.007     -3.165      0.002      -0.037      -0.009
brand_name_Nokia      0.0280      0.008      3.453      0.001       0.012       0.044
brand_name_Sony      -0.0198      0.008     -2.440      0.015      -0.036      -0.004
os_Others            -0.0653      0.007     -8.746      0.000      -0.080      -0.051
4g_yes                0.0113      0.004      3.026      0.003       0.004       0.019
==============================================================================
Omnibus:                      804.022   Durbin-Watson:                   1.963
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             7551.716
Skew:                          -1.296   Prob(JB):                         0.00
Kurtosis:                      11.263   Cond. No.                     3.65e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.65e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Training Performance

	RMSE	MAE	R-squared	Adj. R-squared	MAPE
0	0.062259	0.045791	0.811441	0.810421	3.334939

Test Performance

	RMSE	MAE	R-squared	Adj. R-squared	MAPE
0	0.066303	0.047602	0.8068	0.804345	3.588689

- ***Final model didn't change since p-values over 0.05 were dropped and linear assumptions were conducted***

- *Adj. R-squared for overall fit of model is 0.81 which is good and explains 81% of the variation in data*
- *The train and test RMSE and MAE are low and comparable thus the model is not suffering from overfitting*
- *The MAPE on test performance suggest that we can predict within 3.5% of the used price*
- *Hence, final model is good for prediction and inference purposes.*

Additional Insights

Final Model Summary

```
olsmodel_final = sm.OLS(
    y_train, x_train6
).fit()  ## Complete the code to fit the final model
print(olsmodel_final.summary())
```

```
                            OLS Regression Results
==================================================================================
Dep. Variable:     normalized_used_price_log   R-squared:                  0.811
Model:                                   OLS   Adj. R-squared:             0.810
Method:                        Least Squares   F-statistic:                862.1
Date:                       Wed, 30 Mar 2022   Prob (F-statistic):          0.00
Time:                               02:19:41   Log-Likelihood:            3281.1
No. Observations:                       2417   AIC:                       -6536.
Df Residuals:                           2404   BIC:                       -6461.
Df Model:                                 12
Covariance Type:                   nonrobust
=========================================================================================
                          coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------
main_camera_mp          0.0055      0.000     13.689      0.000       0.005       0.006
selfie_camera_mp        0.0047      0.000     14.394      0.000       0.004       0.005
ram                     0.0410      0.010      4.261      0.000       0.022       0.060
battery               5.54e-06   1.79e-06      3.097      0.002    2.03e-06    9.05e-06
normalized_new_price    0.0989      0.003     37.578      0.000       0.094       0.104
weight_log              0.1333      0.008     16.088      0.000       0.117       0.150
brand_name_Celkon      -0.0431      0.014     -3.141      0.002      -0.070      -0.016
brand_name_Lenovo       0.0114      0.006      1.973      0.049    7.06e-05       0.023
brand_name_Micromax    -0.0230      0.007     -3.165      0.002      -0.037      -0.009
brand_name_Nokia        0.0280      0.008      3.453      0.001       0.012       0.044
brand_name_Sony        -0.0198      0.008     -2.440      0.015      -0.036      -0.004
os_Others              -0.0653      0.007     -8.746      0.000      -0.080      -0.051
4g_yes                  0.0113      0.004      3.026      0.003       0.004       0.019
==============================================================================
Omnibus:                      804.022   Durbin-Watson:                   1.963
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             7551.716
Skew:                          -1.296   Prob(JB):                         0.00
Kurtosis:                      11.263   Cond. No.                     3.65e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.65e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Training Performance

	RMSE	MAE	R-squared	Adj. R-squared	MAPE
0	0.062259	0.045791	0.811441	0.810421	3.334939

Test Performance

	RMSE	MAE	R-squared	Adj. R-squared	MAPE
0	0.066303	0.047602	0.8068	0.804345	3.588689

- *Weight_log and normalized_new_price had the strongest relationship with normalized_used_price*
- *Brand names Celkon, Micomax, and Sony, have inverse relationships with normalized_used_price; as it increases the variables decrease.*
- *Os_Others has an inverse relation with normalized_used_price as well.*
- *The model is showing collinearity with the variable battery and will need to be explored further*