Odds/Kid

ReCell

Project 3: Supervised Learning Foundations

02/04/2022

Contents / Agenda

Executive Summary

- We can explain 84% of the variance in the resale price of a device based on the linear regression model built using the available dataset
- We can predict the resale price of a device within $19.25 if we know its features
- It was found that: screen size, main camera resolution, selfie camera resolution, RAM, battery, weight, release year, price when new, brand: Lenovo, brand: Nokia, brand: Xiaomi, Non-Android nor IOs operating systems and 4G capabilities are statistically significant features we can use to predict used price
- 93% of devices studied in this dataset run Android as operating system
- 75% of devices in this data set had a resale price below $116
- We should avoid devices with an OS that is not Android or IOs
- We should prefer devices that have just been released or that were released a year or two ago
- Only 1% of devices in the data set where Apple

Business Problem Overview and Solution Approach

The used and refurbished device market has grown considerably over the past decade, forecasts predict that the market would be worth $52.7bn by 2023 with a compound annual growth rate (CAGR) of 13.6% from 2018 to 2023.

ReCell , a start-up aiming to capture the potential in this market, has gathered data with different attributes of used/refurbished phones and tablets and wants to explore how they affect their resale price.

- There is a need for an ML-based solution to develop a dynamic pricing strategy for used and refurbished devices.

- We will process the data in order to develop a linear regression model that can predict the price of a used device and understand the factors that significantly influence it.

EDA Results: Summary of Data Set

Odds/Kid

- 50% of screens are 12-15 inches
- Average main camera resolution is 9 Mpx, 75% of cameras are below 13 Mpx
- Average selfie camera resolution is 6Mpx, 75% of cameras are below 8Mpx
- Average ROM is 54Gb, 75% of phones have less than 64Gb
- Only 25% of phones have more than 4Gb RAM
- 75% of phones have less than 4000mAh of battery
- Average weight is 182 grams, 75% of devices weigh less than 185 grams
- Average device was released in 2015, 75% of devices were released before 2018
- Average device is 674 days old, 75% of devices are less than 868 days old
- Average price when device was new was $237, with 75% of devices being less than $291 when new
- Average used price is $92, with 75% of devices having a resale price below $116

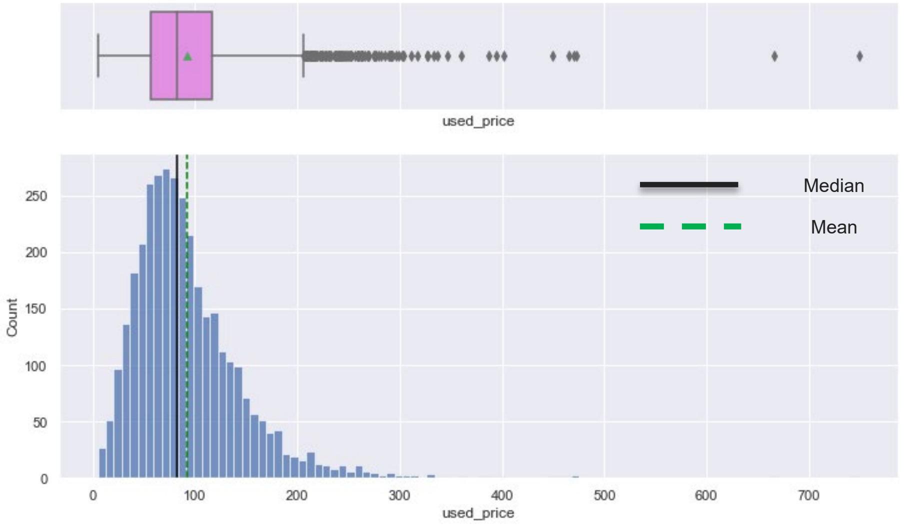
Link to Appendix slide on data background check

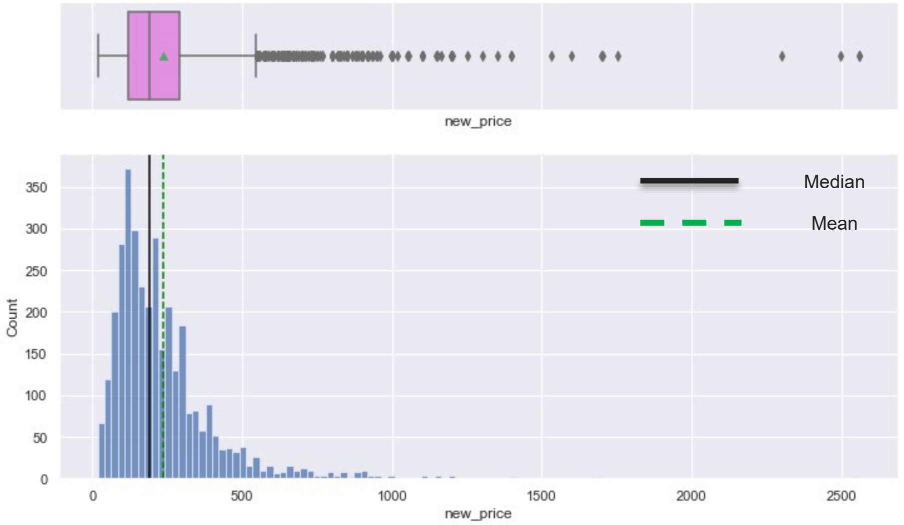
EDA: Attribute Distributions

Odds/Kid

- We observe right skewness in the distributions for “Used_price” and “New_price”
- Long tails to the right displace the mean, as it is sensitive to outliers

In general ML algorithms perform better with data that is normally distributed, we will use a log transformation for this variables to approach normality

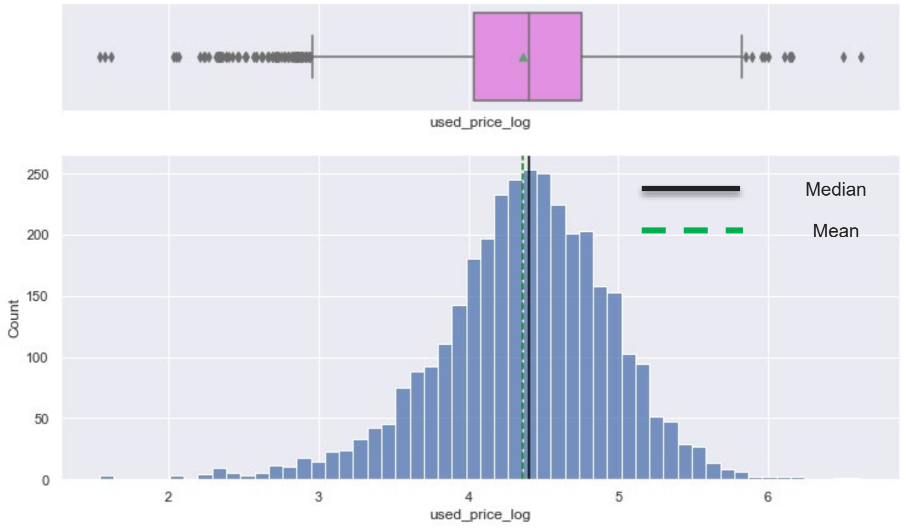


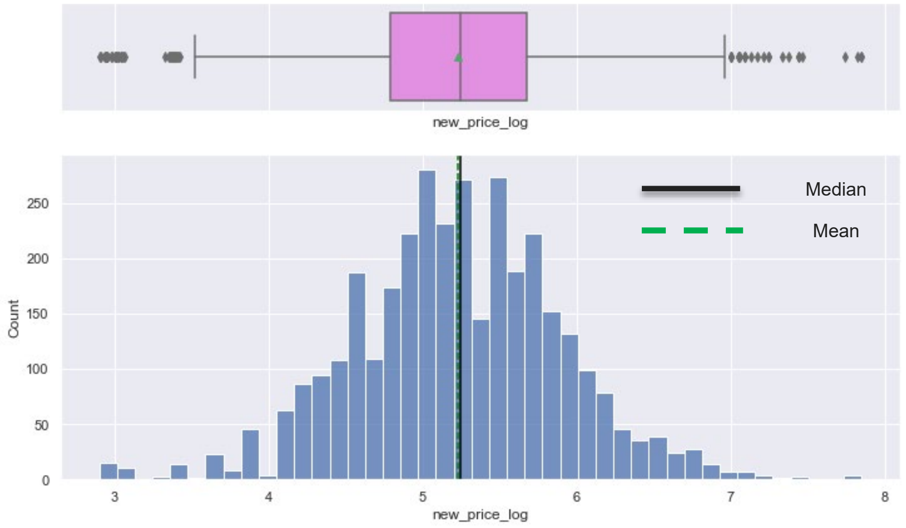


EDA: Attribute Distributions

Odds/Kid

- After the log transformation, we observe a reduction in the skewness of both distributions
- The shapes of the distributions are now closer to being normally distributed
- We observe that the mean is much closer to the median after transformation and the tails are shorter



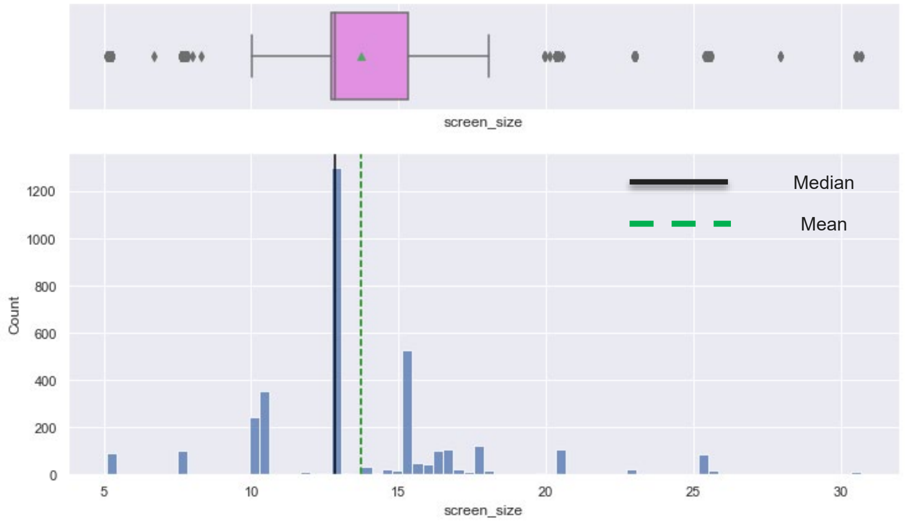


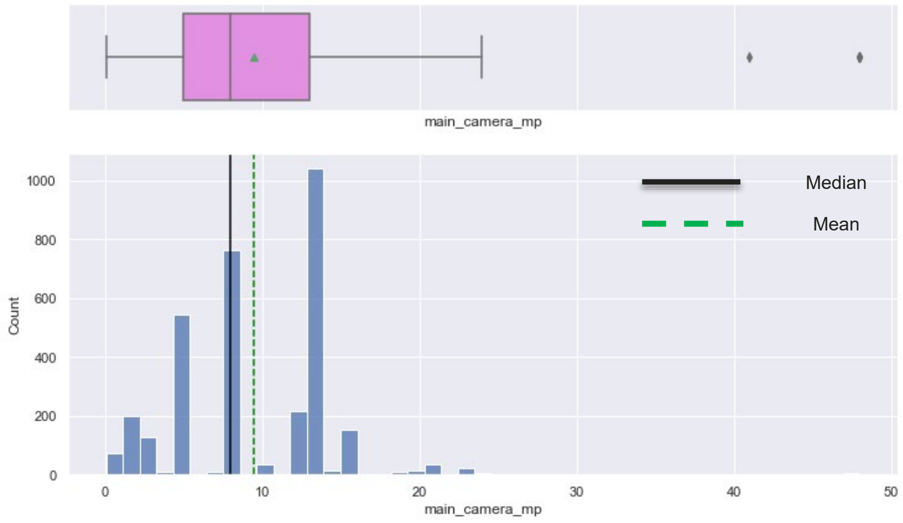
EDA: Attribute Distributions

Odds/Kid

- We observe that most devices have a screen size between 10-15 inches, this makes devices with larger screens appear as outliers, however this data set includes phones and tablets, so we can not drop this values

- We observe that most devices have main cameras between 5-18Mpx, but larger values are still valid as they might be from premium, photography-oriented devices

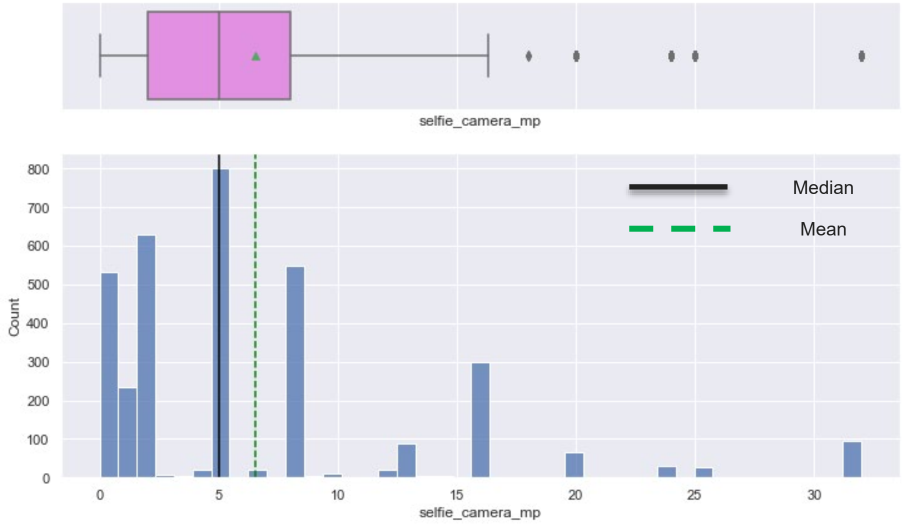


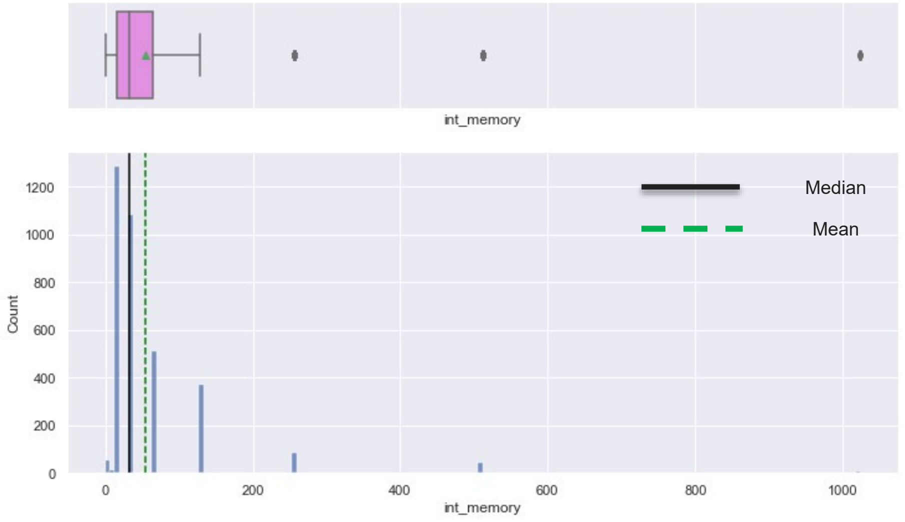


EDA: Attribute Distributions

- We observe that most devices have a selfie camera of 5Mpx and below17Mpx, but larger values are valid

- We observe the internal memory of most devices is below 100Gb, however we will accept larger values as they might be from premium devices with focus on storage



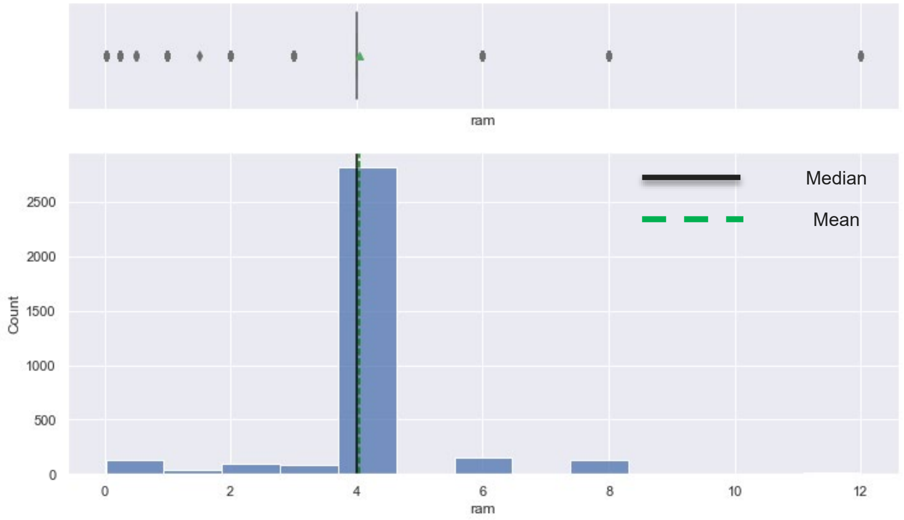


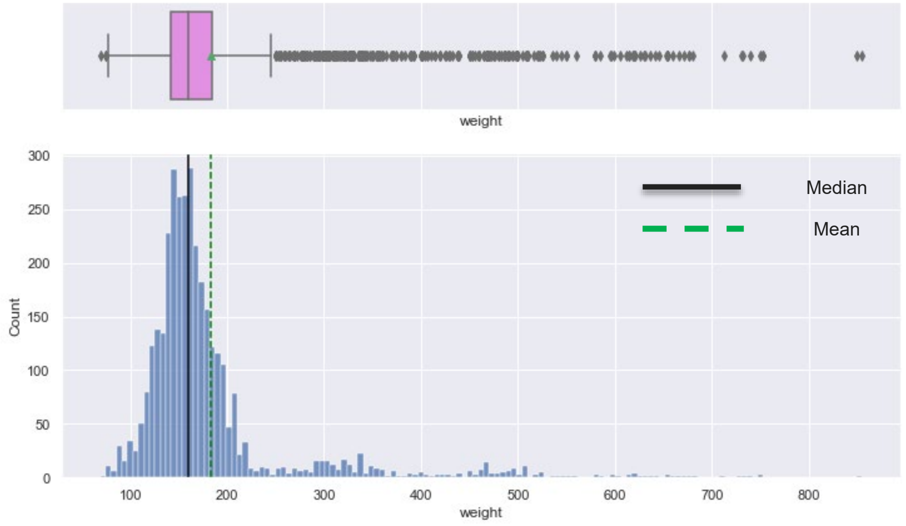
EDA: Attribute Distributions

Odds/Kid

- We observe that a vast majority of devices have 4Gb of Ram memory, this makes all other values appear as outliers, however we will not drop this values because they are valid

- We observe that most devices are lightweight, below 200 grams, however weight might increase with screen size, battery capacity and other features so we can not drop the heavy devices



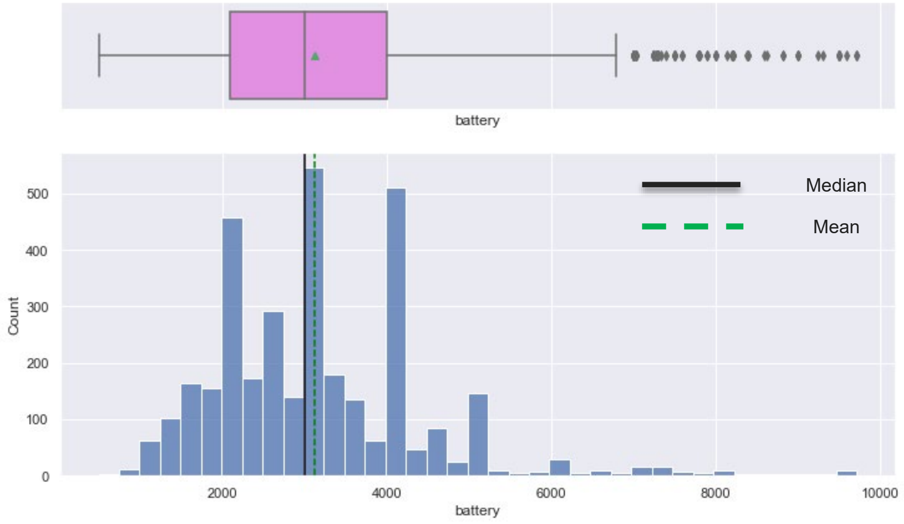


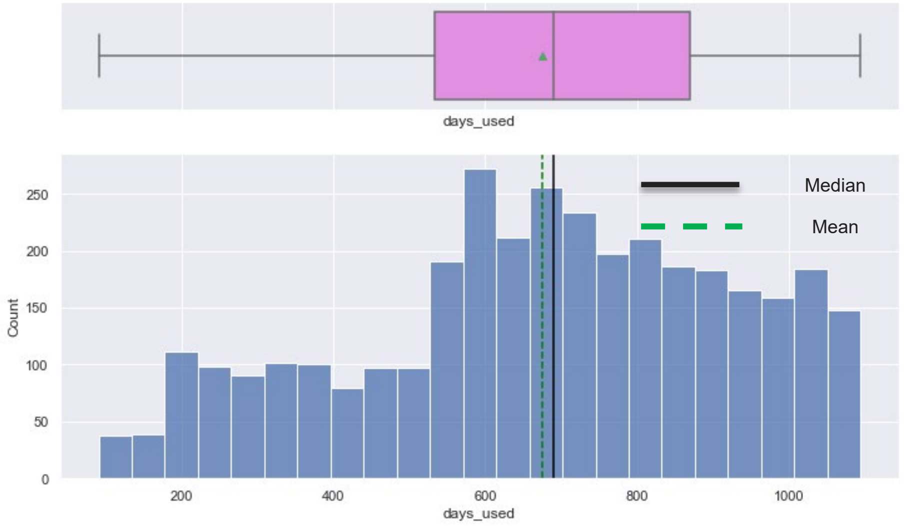
EDA: Attribute Distributions

Odds/Kid

- We observe that most devices have a battery between 2000-4000mAh, this makes devices with higher battery capacities appear as outliers, but we can not exclude this information

- “Days_used” distribution has no outlying values as per 1.5IQR definition, majority of phones have more than 600 days of use

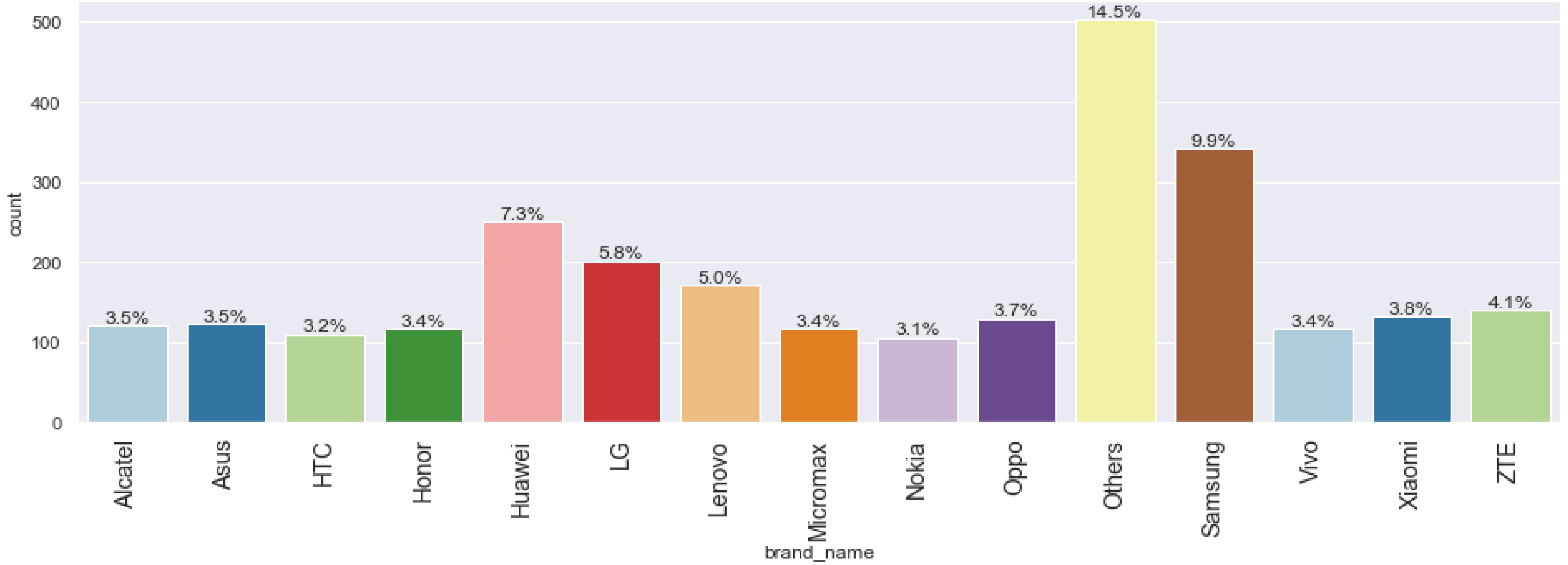




EDA: Manufacturers

Odds/Kid

- This are the top 15 manufactures in our data set, together they account for 77% of all devices
- The top 5: Others, Samsung, Huawei, LG and Lenovo have a share of 42.5% of all devices
- Apple does not appear as a brand in the data set, is dominant in the “Others” category? Let’s keep exploring!

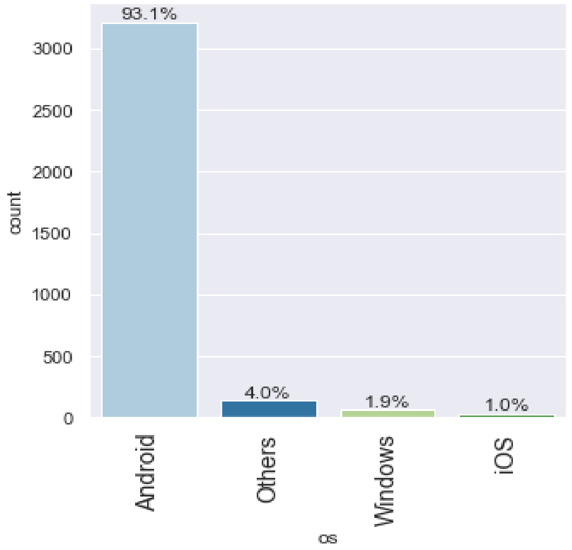


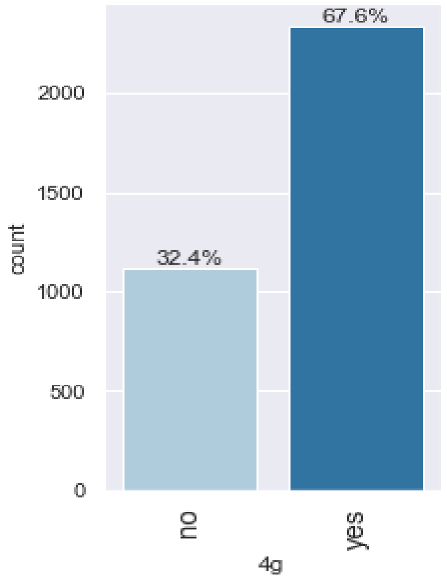
EDA: Operating System and Mobile Technology

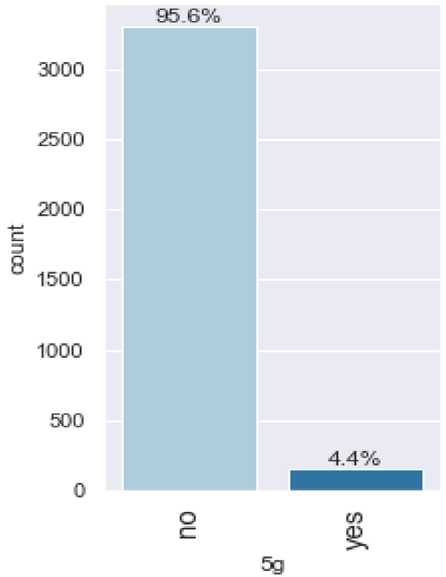
Odds/Kid

- We can see that only 1% of the devices are Apple, in contrast with Android's 93%. This means that the "Others" category must include a majority of other Android manufacturers

- 67% of devices have 4G technology while only 4.4% have 5G technology, the number of devices with pre 4G technology is important at around 32%



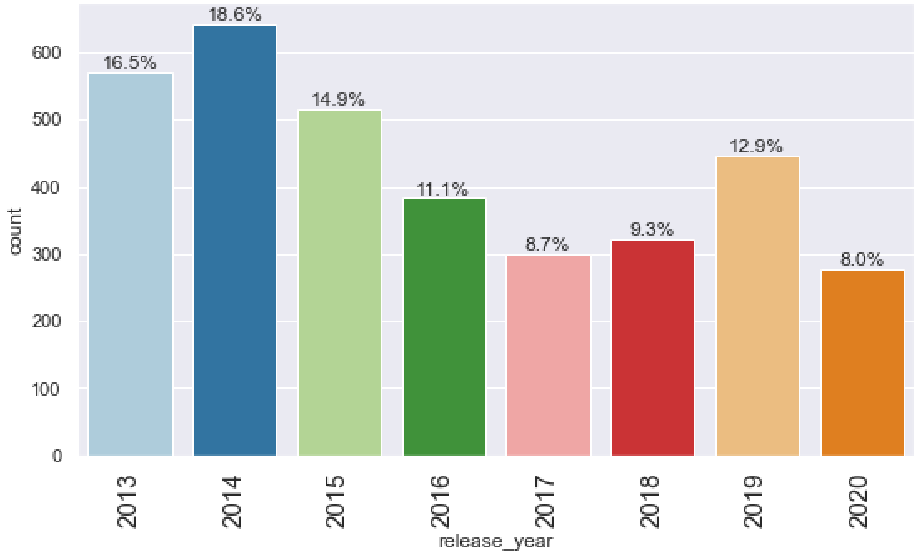


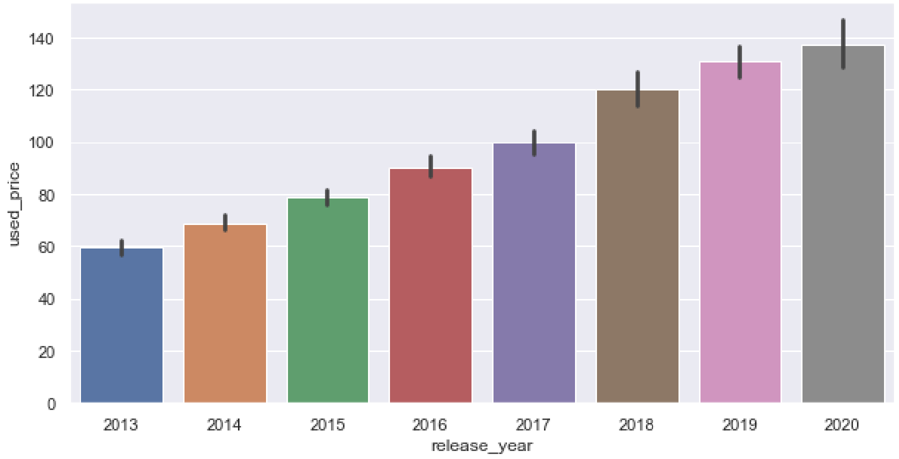


EDA: Release Year

Odds/Kid

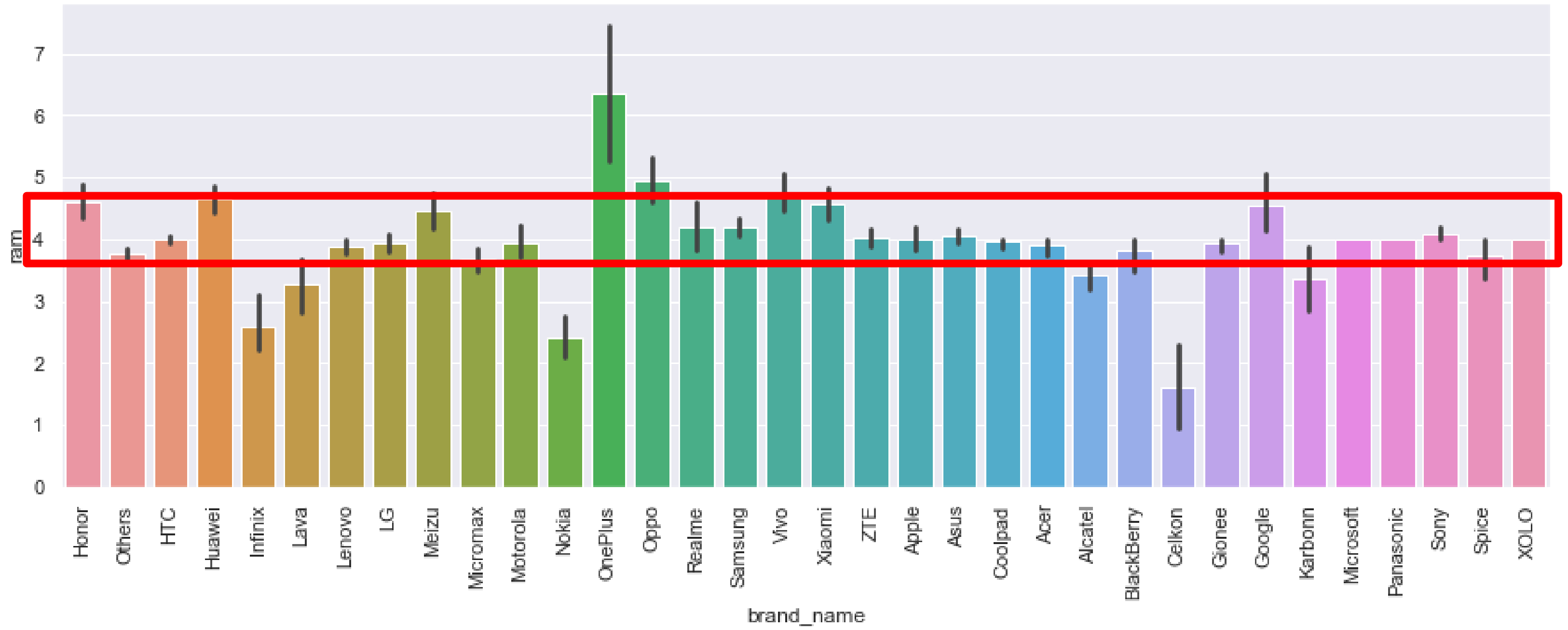
- Close to 70% of devices were released before 2017, the remaining 30% was released between 2017-2020
- The bar plot of “used_price” vs “release_year” uncovers a linear relationship between this two features, with newer devices having a higher resale price as used
- The same linear relationship is true for “new_price” vs “release_year”





EDA: RAM

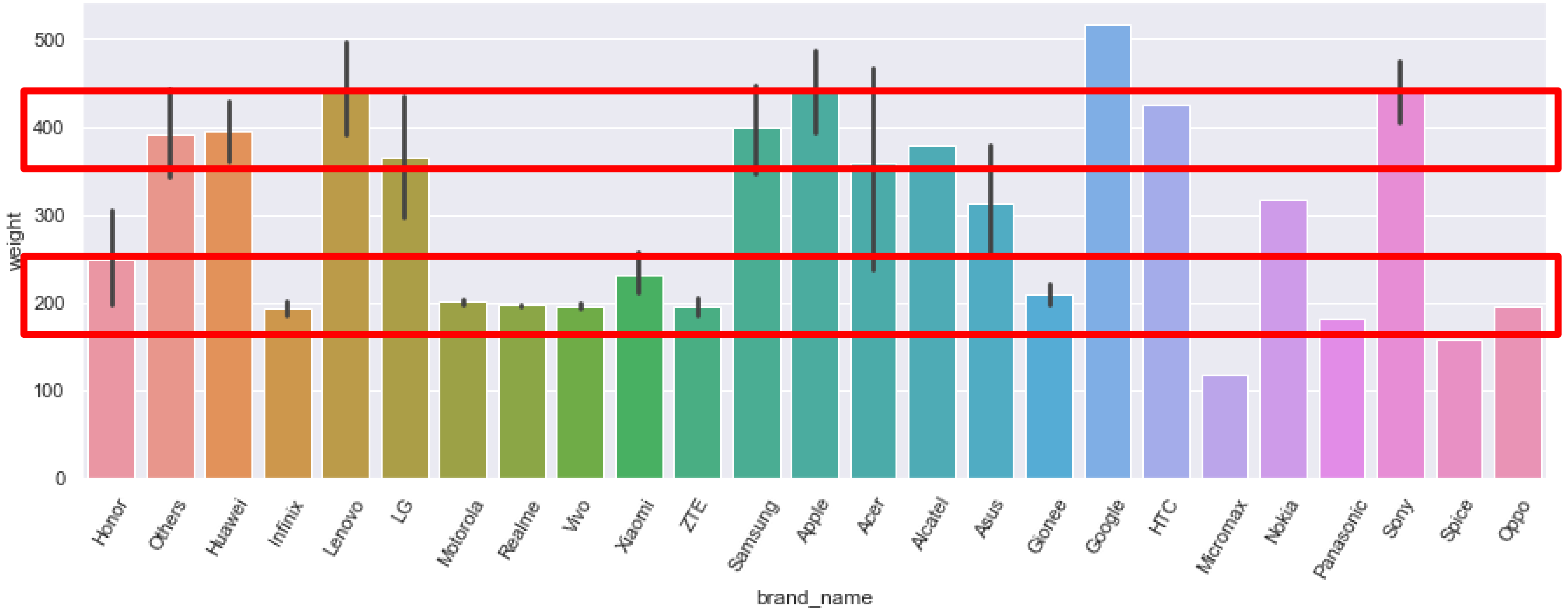
- We can see that most of the devices in the data set have 4Gb of ram, this is true across manufacturers
- In fact, 81% of devices have 4Gb of RAM



EDA: LARGE BATTERY

Odds/Kid

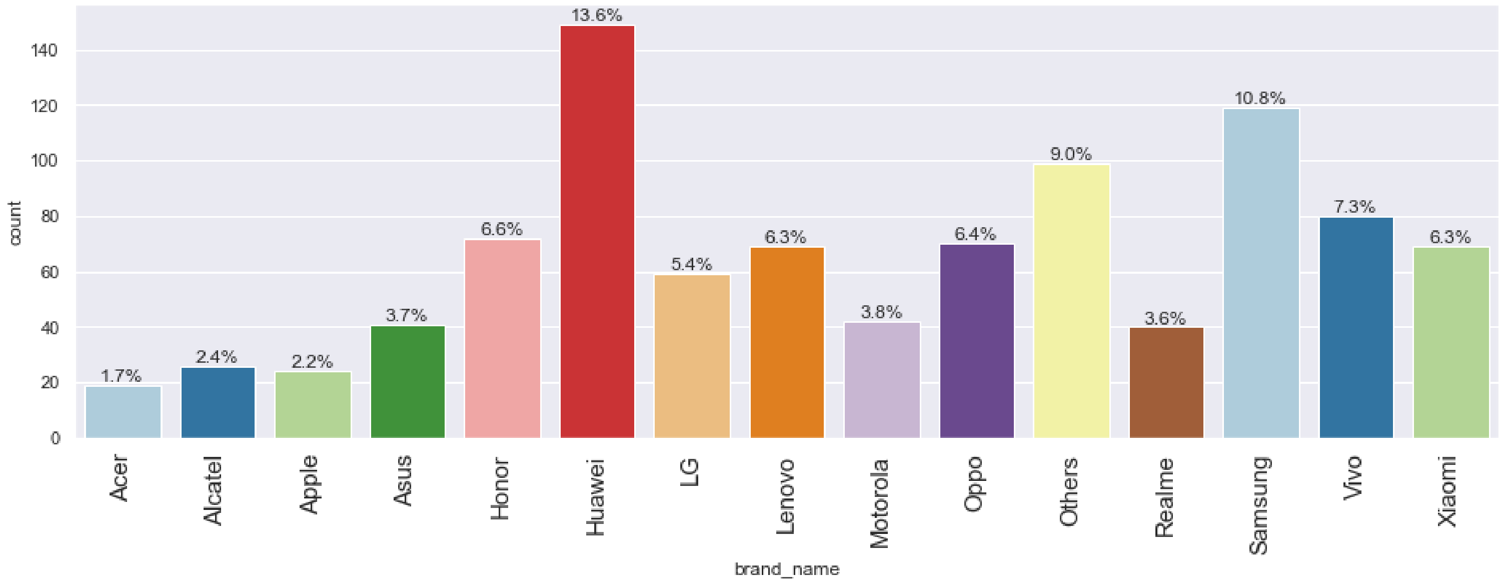
- Weight of devices with battery capacities higher than 4500mAh are concentrated around 200grams and 400grams



EDA: LARGE SCREEN

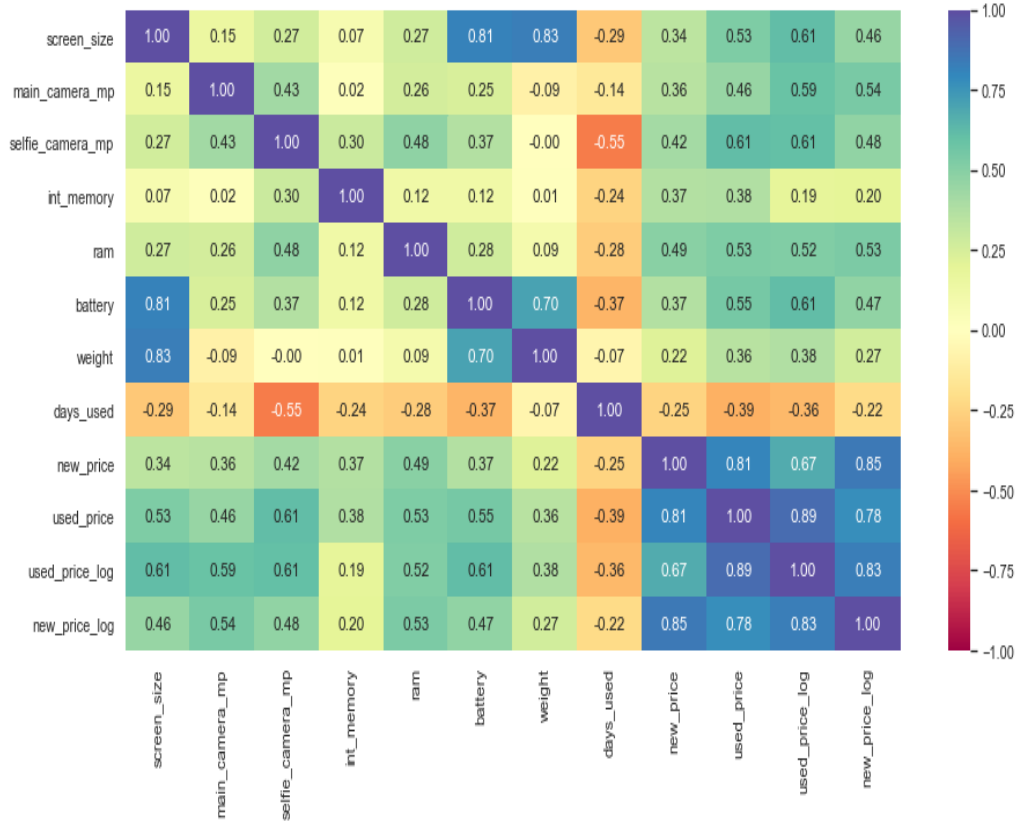
Odds/Kid

- Huawei stands out as the dominant manufacturer for devices with screens larger than 6 inches in this dataset
- Samsung and Others are next



EDA: Correlation

Odds/Kid



	screen_size	main_camera_mp	selfie_camera_mp	int_memory	ram	battery	weight	days_used	new_price	used_price	used_price_log	new_price_log
screen_size	1.00	0.15	0.27	0.07	0.27	0.81	0.83	-0.29	0.34	0.53	0.61	0.46
main_camera_mp	0.15	1.00	0.43	0.02	0.26	0.25	-0.09	-0.14	0.36	0.46	0.59	0.54
selfie_camera_mp	0.27	0.43	1.00	0.30	0.48	0.37	-0.00	-0.55	0.42	0.61	0.61	0.48
int_memory	0.07	0.02	0.30	1.00	0.12	0.12	0.01	-0.24	0.37	0.38	0.19	0.20
ram	0.27	0.26	0.48	0.12	1.00	0.28	0.09	-0.28	0.49	0.53	0.52	0.53
battery	0.81	0.25	0.37	0.12	0.28	1.00	0.70	-0.37	0.37	0.55	0.61	0.47
weight	0.83	-0.09	-0.00	0.01	0.09	0.70	1.00	-0.07	0.22	0.36	0.38	0.27
days_used	-0.29	-0.14	-0.55	-0.24	-0.28	-0.37	-0.07	1.00	-0.25	-0.39	-0.36	-0.22
new_price	0.34	0.36	0.42	0.37	0.49	0.37	0.22	-0.25	1.00	0.81	0.67	0.85
used_price	0.53	0.46	0.61	0.38	0.53	0.55	0.36	-0.39	0.81	1.00	0.89	0.78
used_price_log	0.61	0.59	0.61	0.19	0.52	0.61	0.38	-0.36	0.67	0.89	1.00	0.83
new_price_log	0.46	0.54	0.48	0.20	0.53	0.47	0.27	-0.22	0.85	0.78	0.83	1.00

Key Insights:

- Strong positive relation between used price and new price
- Weak negative relation between used price and days used
- Weak positive relations between used priced and: screen size, main camera, selfie camera, RAM, ROM and weight
- Strong positive relation between screen size and battery; screen size and weight

Data Preprocessing

- Duplicate value check
- Missing value treatment
- Outlier check (treatment if needed)
- Feature engineering
- Data preparation for modeling

Data Preprocessing*: Missing Value Treatment

Imputation

Checking:

	Missing
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
new_price	0
used_price	0
used_price_log	0
new_price_log	0
device_category	0

Imputing by median grouping by: release year & brand name

1

	Missing
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

Imputing by median grouping by: brand name

2

	Missing
brand_name	0
os	0
screen_size	0
4g	0
5g	0
main_camera_mp	10
selfie_camera_mp	0
int_memory	0
ram	0
battery	0
weight	0
release_year	0

Imputing by median in main camera column

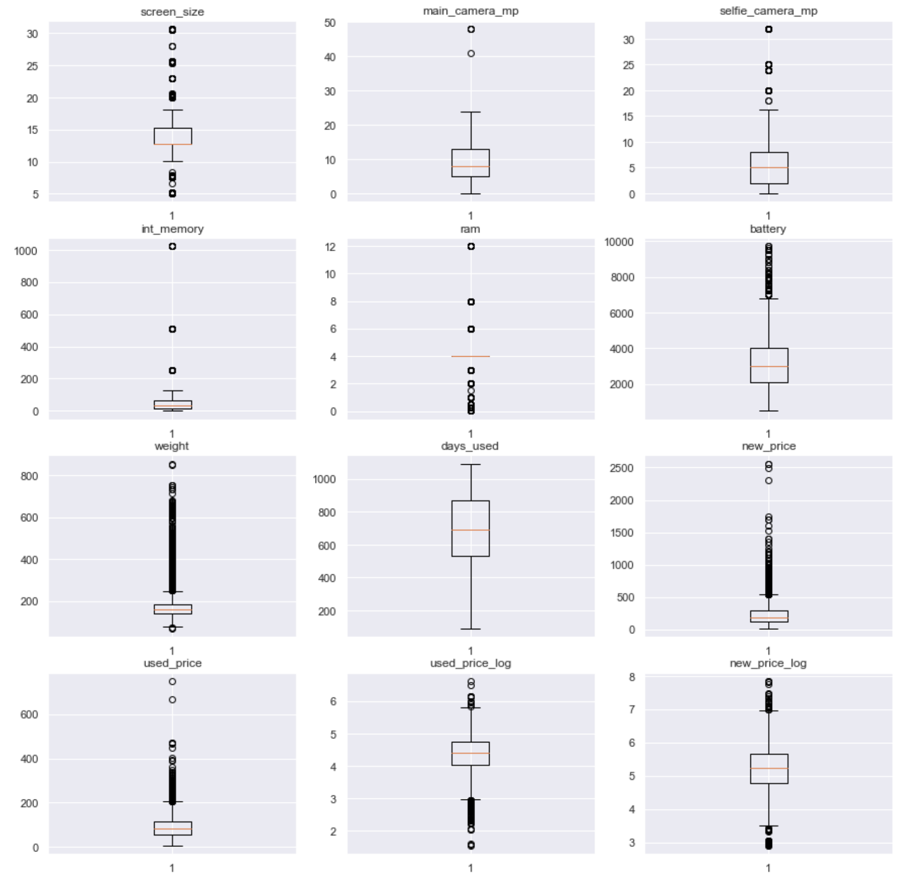
3

No missing:

	Missing
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
new_price	0
used_price	0
used_price_log	0
new_price_log	0
device_category	0

*No duplicate values were found

Data Preprocessing: Outlier Treatment



- We can see that there are outliers in all distributions as per the 1.5IQR rule, however we have no reason to believe that this values are invalid

- Values that appear as outliers contain valid and valuable information about screen sizes, main camera, selfie camera, ROM, RAM, battery, weight and prices

- One of the reasons they might be outlying is that most devices share common traits, while premium or niche devices have high values for specific features

- No outlier treatment was performed in the dataset

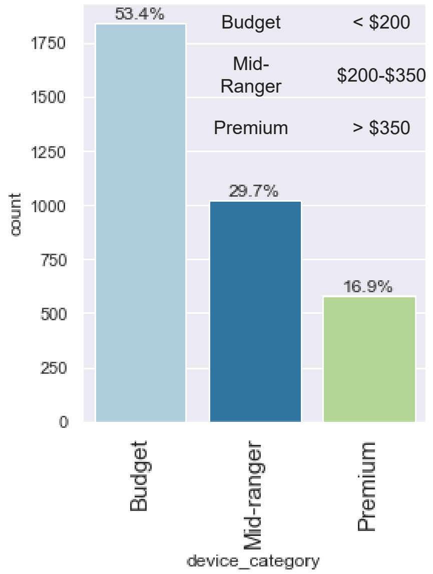
Data Preprocessing: Feature Engineering

Price Bins & Selfie Camera Resolution above 8Mpx

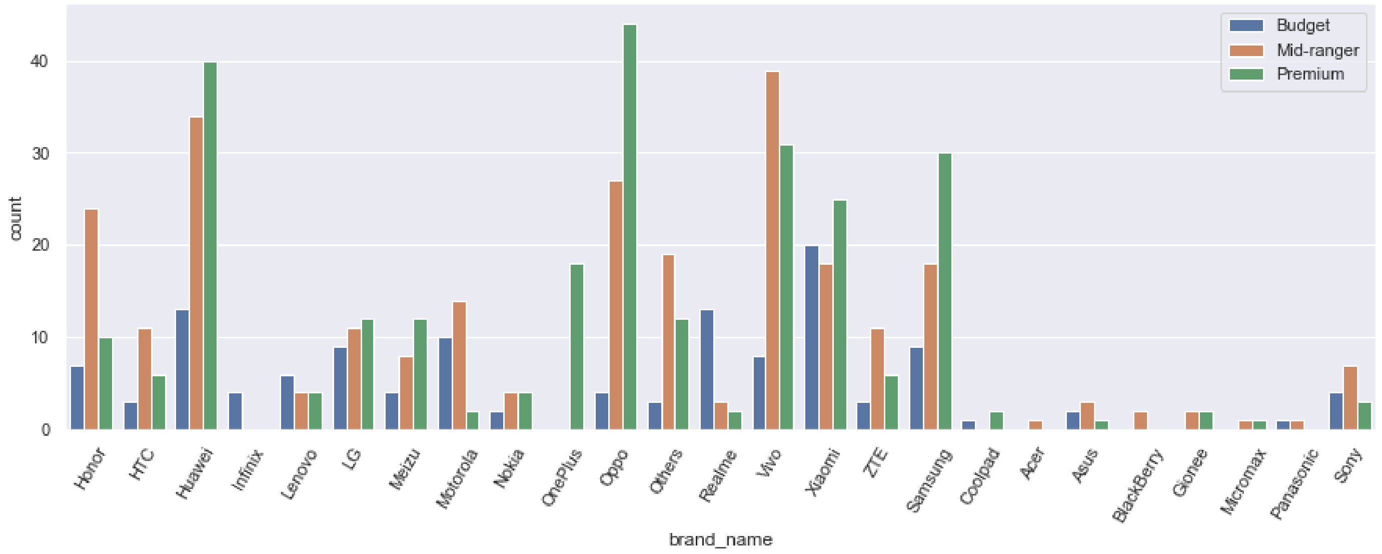
Odds/Kid

- Most devices with selfie camera resolution higher than 8Mpx are in the Mid-Ranger to Premium category, although Budget alternatives can be found across most of the brands

- Xiaomi stands out as the brand with more Budget devices with selfie camera resolution above 8Mpx in this dataset



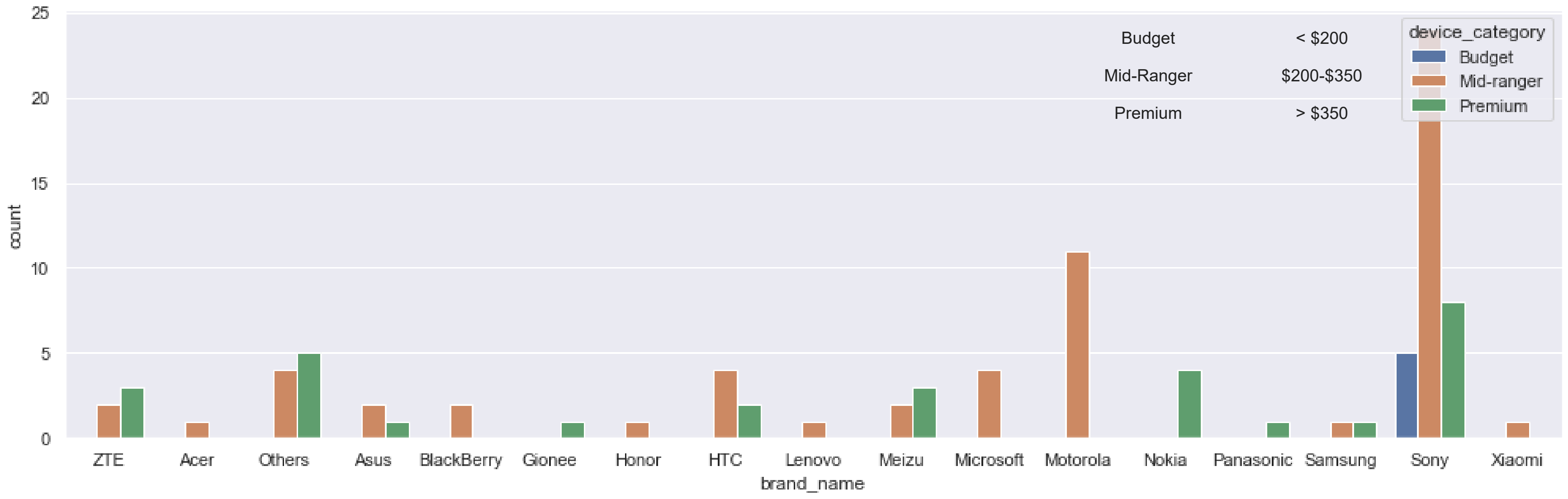
Budget	< $200
Mid-Ranger	$200-$350
Premium	> $350



Data Preprocessing: Feature Engineering

Price Bins & Main Camera Resolution above 16Mpx

- Sony stands out as the only brand with a Budget device with main camera resolution above 16Mpx in this dataset
- Devices with main camera resolutions above 16Mpx are Midrange and Premium devices, Motorola and Sony stand out as the brands with most devices in this category that is focused on clients interested in photography



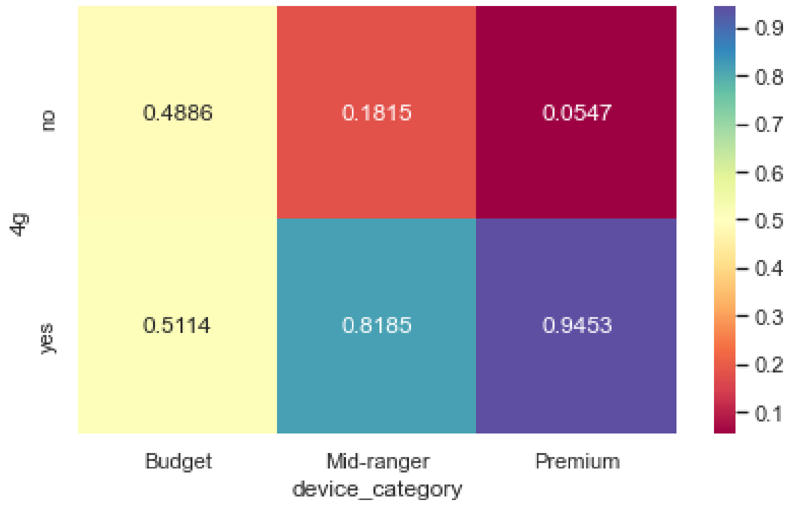
Budget	< $200
Mid-Ranger	$200-$350
Premium	> $350

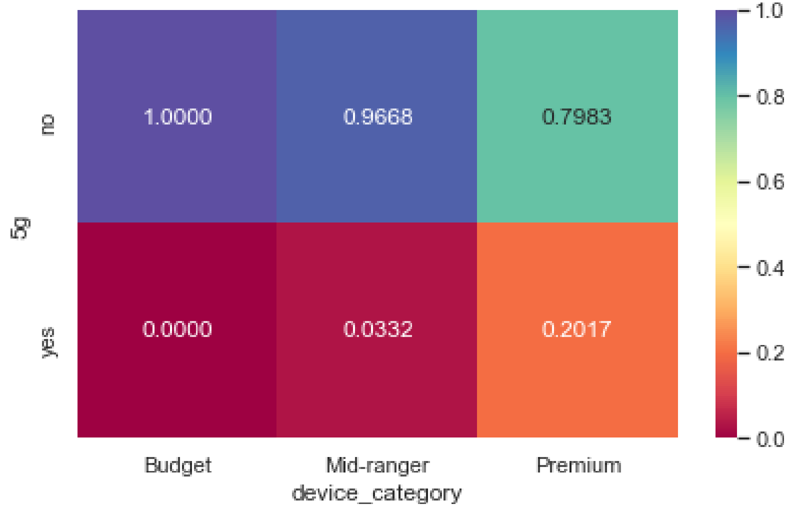
Data Preprocessing: Feature Engineering

Price Bins & Mobile Technology

Odds/Kid

- 94% of premium devices in this dataset have 4G capabilities, in contrast only 20% of premium devices have 5G
- Only 3% of Midrange devices have 5G, no Budget devices have 5G
- 48% of Budgets devices have pre 4G technology, while only 18% and 5% of Midrange and Premium devices are not 4G





Data Preprocessing: Data Preparation for Modeling

Encoding Categorical Variables, Splitting Data, Adding the Intercept

Odds/Kid

$$y = a + b_1 x_1 + \cdots + b_k x_k$$

- We will build the model for the transformed independent variable ln(used_price) so we drop the not transformed
- We will use the transformed predictor ln(new_price) so we drop the not transformed
- Encoded categorical variables and dropped the first encoded variable as it is redundant
- Split data in to train and test sets with a 70:30 ratio using
- Added the intercept for train and test independent variables
- Data is now ready to fit a Linear Regression model!

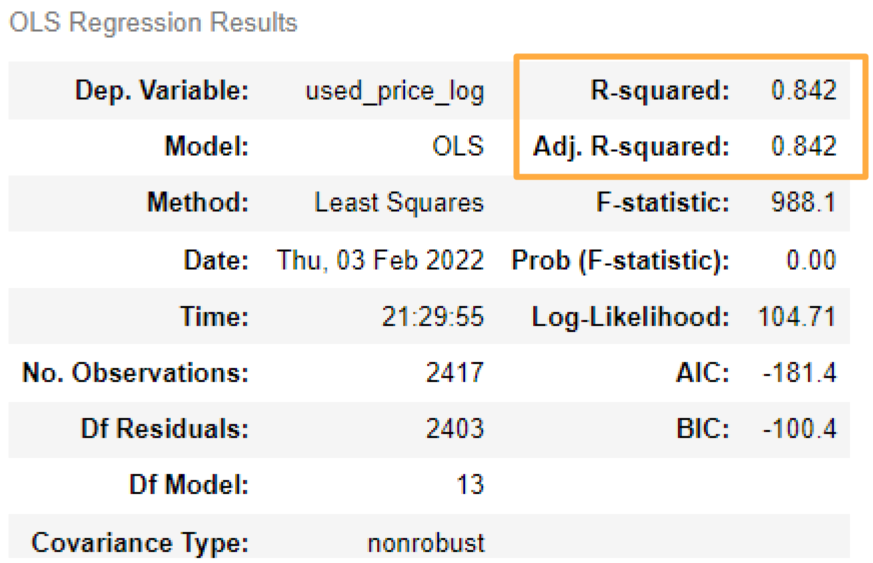
Link to Appendix slide

Odds/Kid

Model Performance Summary

- Overview of ML model and its parameters
- Summary of most important factors used by the ML model for prediction
- Summary of key performance metrics for training and test data:
  - Root Mean Squared Error
  - Mean Absolute Error
  - Mean Absolute Percentage Error

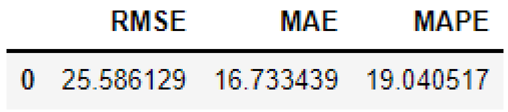
Model Performance Summary



OLS Regression Results

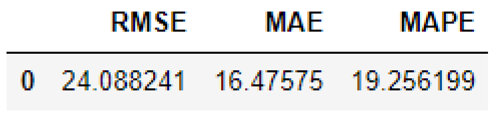
Dep. Variable:	used_price_log	**R-squared:**	0.842
Model:	OLS	**Adj. R-squared:**	0.842
Method:	Least Squares	**F-statistic:**	988.1
Date:	Thu, 03 Feb 2022	**Prob (F-statistic):**	0.00
Time:	21:29:55	**Log-Likelihood:**	104.71
No. Observations:	2417	**AIC:**	-181.4
Df Residuals:	2403	**BIC:**	-100.4
Df Model:	13		
Covariance Type:	nonrobust		

Performance on train set:



	RMSE	MAE	MAPE
0	25.586129	16.733439	19.040517

Performance on test set:

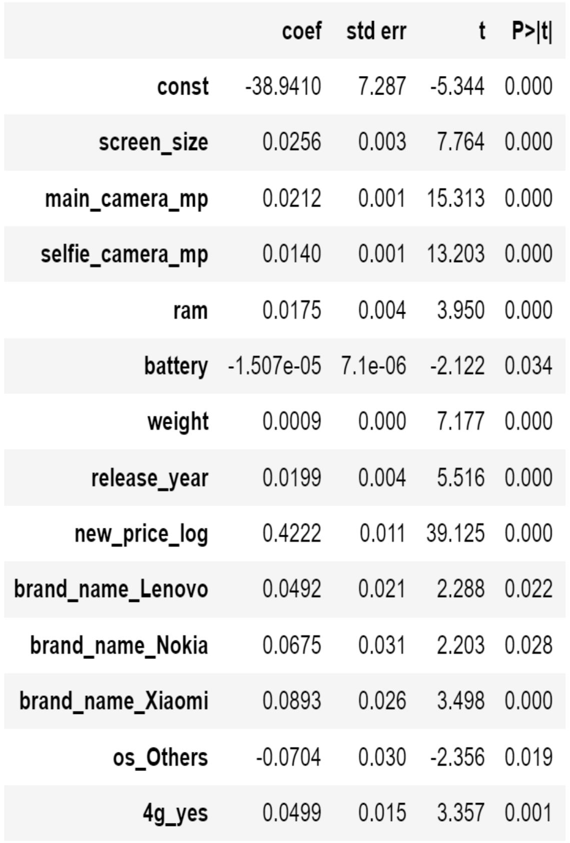


	RMSE	MAE	MAPE
0	24.088241	16.47575	19.256199

- Model explains 84.2% of the variance in the training set
- RMSE on train and test set are comparable, so our model is not suffering from overfitting or underfitting
- MAE indicates that our model can predict used price within a mean error of $19.25 on test data
- Initial model had 48 predictors, after programmatically dropping predictors with p-values > 0.05 we have 13 predictors
- We can conclude that the model is good for prediction!

Odds/Kid

Model Performance Summary: Coefficients
After removing all coefficients with p-val



	coef	std err	t	P>|t|
const	-38.9410	7.287	-5.344	0.000
screen_size	0.0256	0.003	7.764	0.000
main_camera_mp	0.0212	0.001	15.313	0.000
selfie_camera_mp	0.0140	0.001	13.203	0.000
ram	0.0175	0.004	3.950	0.000
battery	-1.507e-05	7.1e-06	-2.122	0.034
weight	0.0009	0.000	7.177	0.000
release_year	0.0199	0.004	5.516	0.000
new_price_log	0.4222	0.011	39.125	0.000
brand_name_Lenovo	0.0492	0.021	2.288	0.022
brand_name_Nokia	0.0675	0.031	2.203	0.028
brand_name_Xiaomi	0.0893	0.026	3.498	0.000
os_Others	-0.0704	0.030	-2.356	0.019
4g_yes	0.0499	0.015	3.357	0.001

- A unit increase in screen size will result in a 2.59% increase in used price
- A unit increase in main camera resolution will result in a 2.14% increase in used price
- A unit increase in selfie camera resolution will result in a 1.40% increase in used price
- A unit increase in RAM will result in a 1.76% increase in used price
- A unit increase in battery will result in a 0.001% decrease in used price
- A unit increase in weight will result in a 0.09% increase in used price
- A unit increase in release year will result in a 2.0% increase in used priced
- A 1% increase in new price will result in a 0.42% increase in used price
- A device from Lenovo will result in a 5.04% increase in used price
- A device from Nokia will result in a 7% increase in used price
- A device from Xiaomi will result in a 9% increase in used price
- A device that is not Android nor IOS will decrease used price by 6.8%
- A device having 4G will increase used price by 5.11%

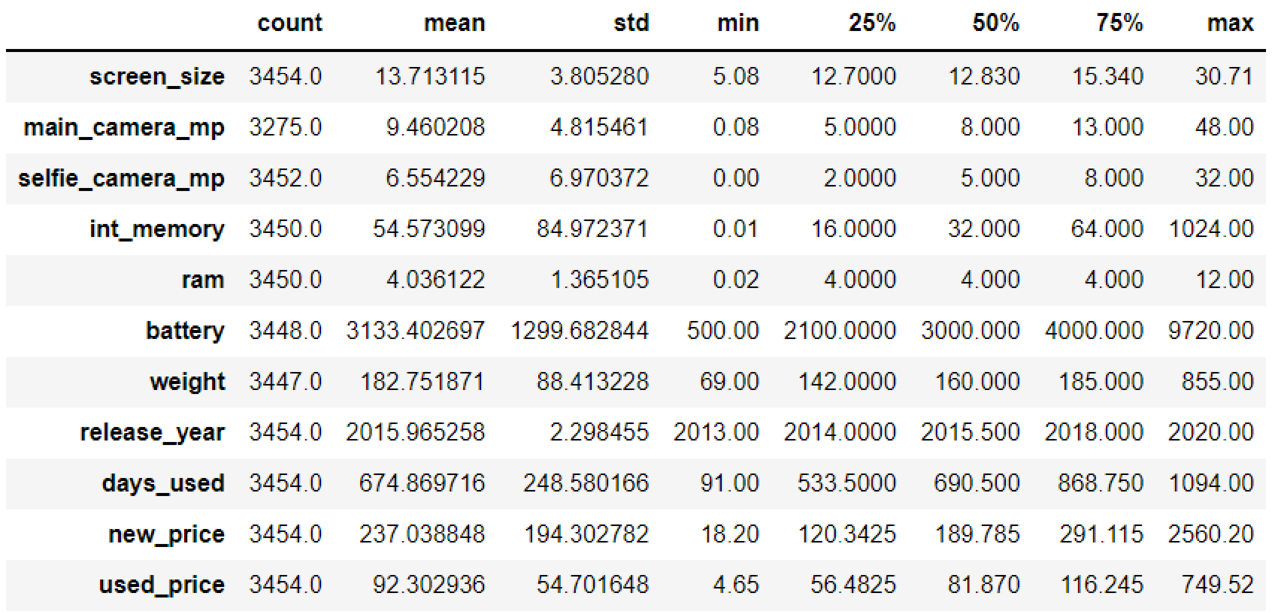
Link to Appendix slide on model assumptions

Odds/Kid

APPENDIX: Data Background

Data Background and Contents

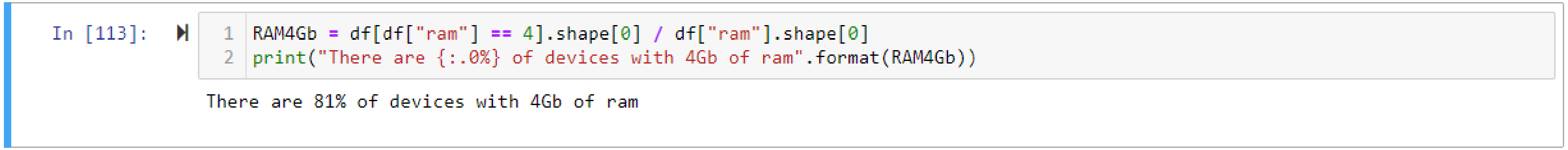
- 5-point summary of raw dataset



	count	mean	std	min	25%	50%	75%	max
screen_size	3454.0	13.713115	3.805280	5.08	12.7000	12.830	15.340	30.71
main_camera_mp	3275.0	9.460208	4.815461	0.08	5.0000	8.000	13.000	48.00
selfie_camera_mp	3452.0	6.554229	6.970372	0.00	2.0000	5.000	8.000	32.00
int_memory	3450.0	54.573099	84.972371	0.01	16.0000	32.000	64.000	1024.00
ram	3450.0	4.036122	1.365105	0.02	4.0000	4.000	4.000	12.00
battery	3448.0	3133.402697	1299.682844	500.00	2100.0000	3000.000	4000.000	9720.00
weight	3447.0	182.751871	88.413228	69.00	142.0000	160.000	185.000	855.00
release_year	3454.0	2015.965258	2.298455	2013.00	2014.0000	2015.500	2018.000	2020.00
days_used	3454.0	674.869716	248.580166	91.00	533.5000	690.500	868.750	1094.00
new_price	3454.0	237.038848	194.302782	18.20	120.3425	189.785	291.115	2560.20
used_price	3454.0	92.302936	54.701648	4.65	56.4825	81.870	116.245	749.52

EDA: RAM

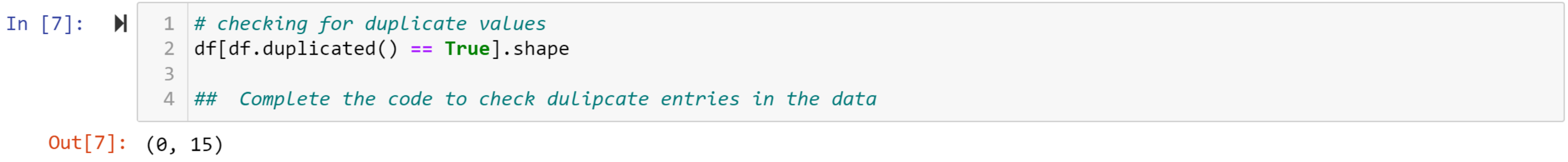
Odds/Kid



```
In [113]:
RAM4Gb = df[df["ram"] == 4].shape[0] / df["ram"].shape[0]
print("There are {:.0%} of devices with 4Gb of ram".format(RAM4Gb))
```

```
There are 81% of devices with 4Gb of ram
```

EDA: Duplicates



```
In [7]:
# checking for duplicate values
df[df.duplicated() == True].shape

##  Complete the code to check dulipcate entries in the data
```

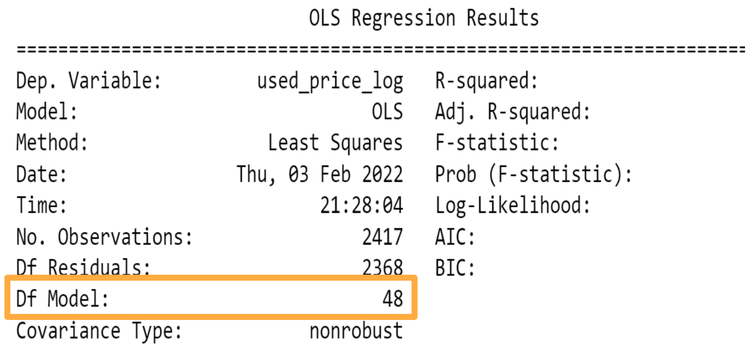
```
Out[7]: (0, 15)
```

Link back to EDA

Odds/Kid

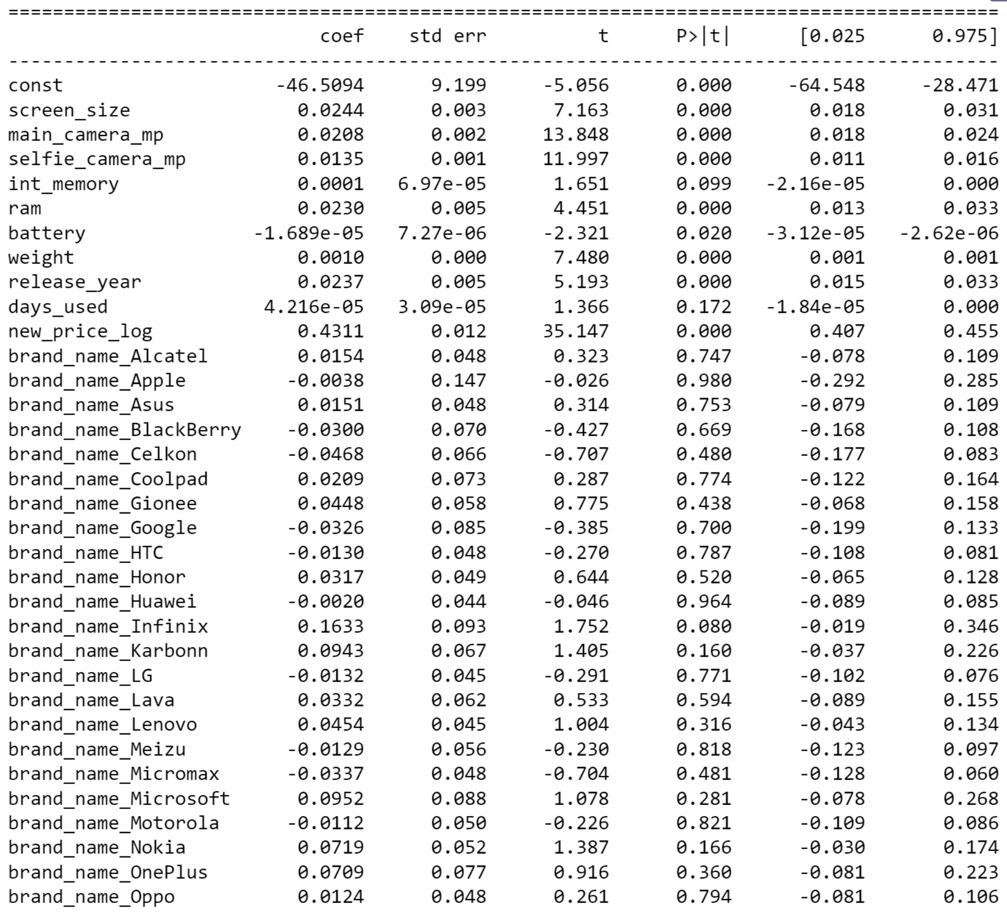
APPENDIX: Model Performance

Model Assumptions: Multicollinearity



```
                          OLS Regression Results
==============================================================================
Dep. Variable:          used_price_log   R-squared:
Model:                             OLS   Adj. R-squared:
Method:                  Least Squares   F-statistic:
Date:                 Thu, 03 Feb 2022   Prob (F-statistic):
Time:                         21:28:04   Log-Likelihood:
No. Observations:                 2417   AIC:
Df Residuals:                     2368   BIC:
Df Model:                           48
Covariance Type:             nonrobust
```

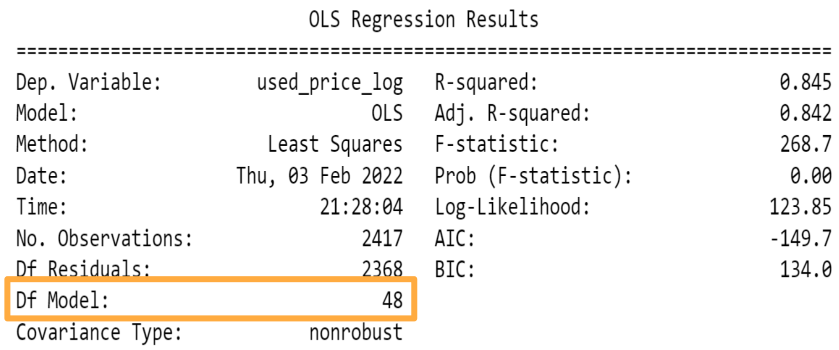
After encoding categorical variables, the first model that was fitted had 48 predictors, with an Adj. R-squared of 0.842



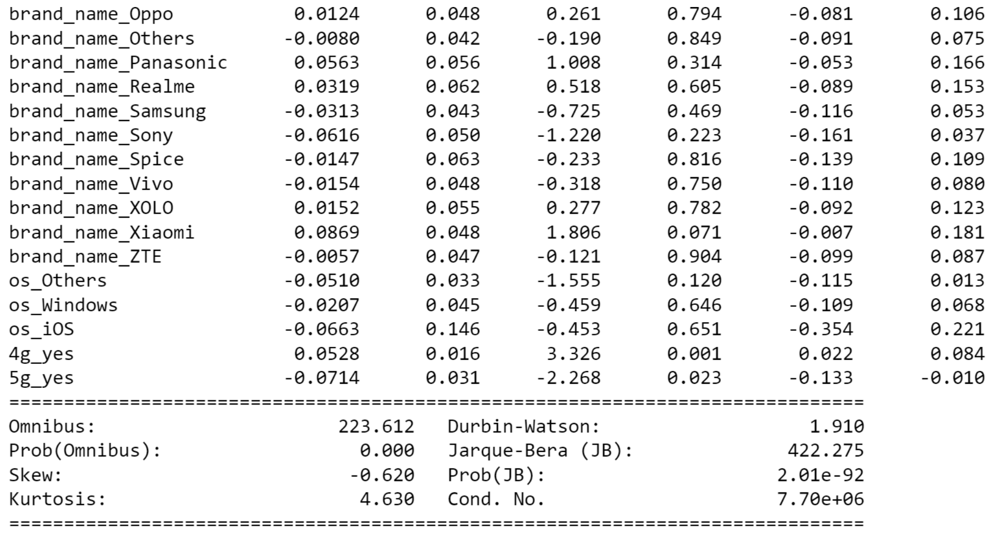
```
==========================================================================================
                             coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------
const                    -46.5094      9.199     -5.056      0.000     -64.548     -28.471
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
release_year               0.0237      0.005      5.193      0.000       0.015       0.033
days_used               4.216e-05   3.09e-05      1.366      0.172   -1.84e-05       0.000
new_price_log              0.4311      0.012     35.147      0.000       0.407       0.455
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
```

Model Assumptions: Multicollinearity

Odds/Kid



```
                            OLS Regression Results
==============================================================================
Dep. Variable:          used_price_log   R-squared:                       0.845
Model:                             OLS   Adj. R-squared:                  0.842
Method:                  Least Squares   F-statistic:                     268.7
Date:                 Thu, 03 Feb 2022   Prob (F-statistic):               0.00
Time:                         21:28:04   Log-Likelihood:                 123.85
No. Observations:                 2417   AIC:                            -149.7
Df Residuals:                     2368   BIC:                             134.0
Df Model:                           48
Covariance Type:             nonrobust
```



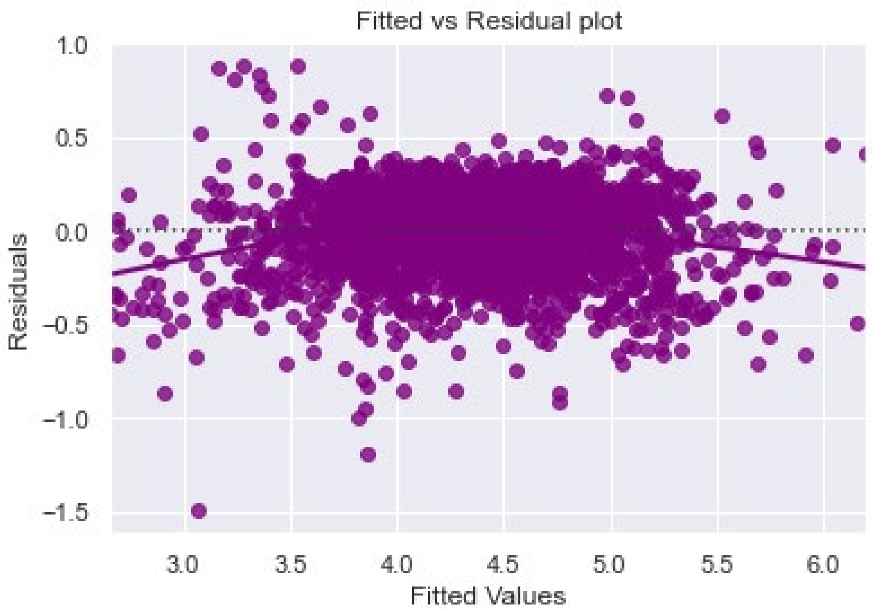
```
brand_name_Oppo          0.0124      0.048      0.261      0.794     -0.081      0.106
brand_name_Others       -0.0080      0.042     -0.190      0.849     -0.091      0.075
brand_name_Panasonic     0.0563      0.056      1.008      0.314     -0.053      0.166
brand_name_Realme        0.0319      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
==============================================================================
Omnibus:                      223.612   Durbin-Watson:                   1.910
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              422.275
Skew:                          -0.620   Prob(JB):                     2.01e-92
Kurtosis:                       4.630   Cond. No.                     7.70e+06
==============================================================================
```

After checking this model for multicollinearity most predictors had high variance inflation factors, confirming that the model had strong multicollinearity issues

For the final model we programmatically dropped all predictors with p-values > 0.05 This reduced the model from 48 predictors to

Model Assumptions: Test for linearity and Independence

Odds/Kid



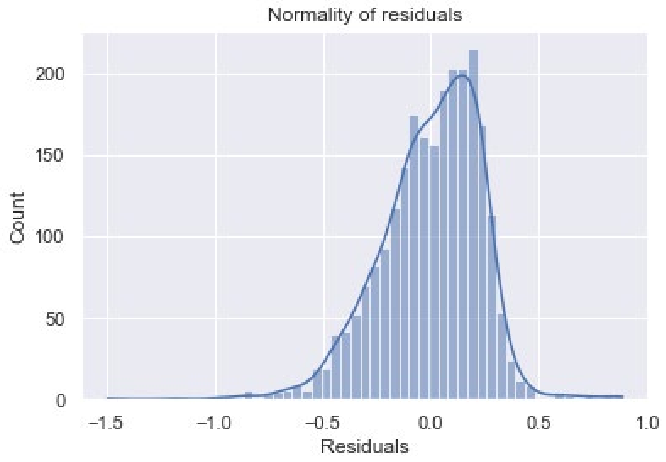
We know that after removing linearity from the data, the residuals should not exhibit any pattern, as they should be entirely noise.

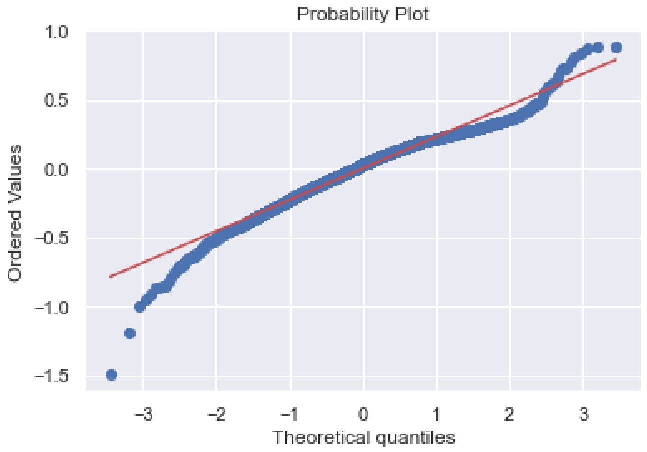
We also know that no point should influence any other point, as noise is random.

- Residual plot is acceptable for us to say that there is no pattern in the residual plot
- Residual plot is acceptable for us as to say that there is independence among the residuals
- Thus, this model passes the test for linearity and independence !

Model Assumptions: Test for Normality

Odds/Kid

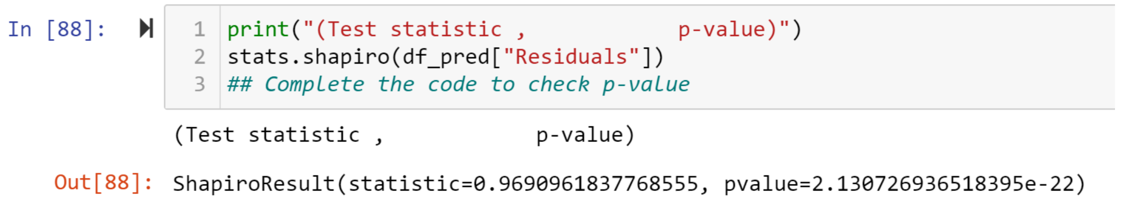




We want our residuals to be normally distributed, we can visually explore this by plotting the distribution of residuals or using a QQ plot with normality in the 45-degree diagonal.

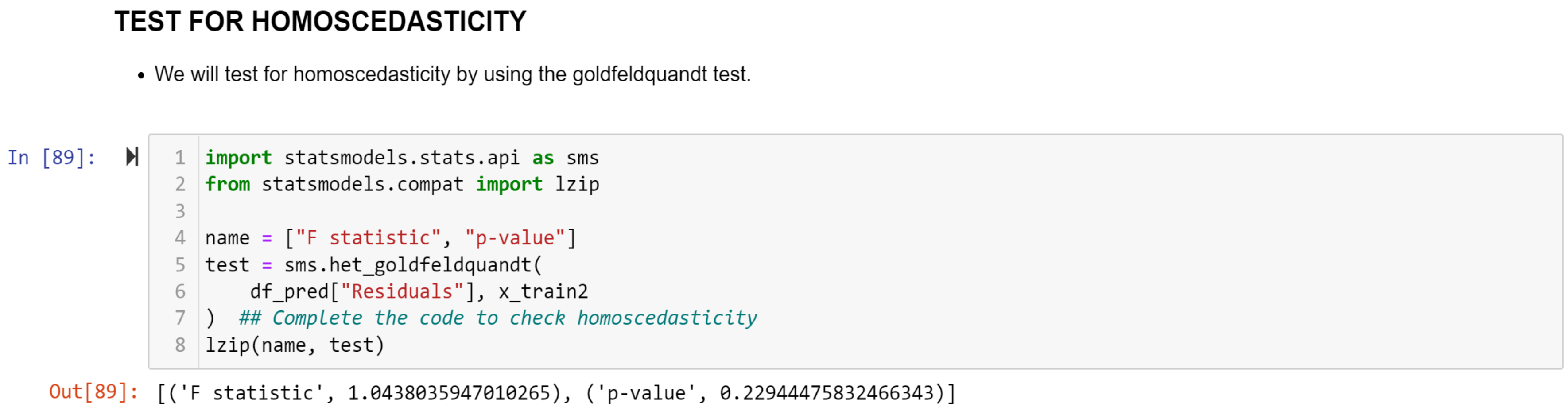
We can also use the Shapiro test; however, this is a statistical test, and it tends to be very strict.

- Based on the distribution plot and QQ plot we can accept this residuals as normally distributed, as they are close enough, specially coming from real world data.
- However, this residuals do not pass the Shapiro test this is because statistical test are very strict.
- For practical purposes we accept normality of residuals using visual exploration and thus the model passes the test for normality!



```
In [88]:  print("(Test statistic ,           p-value)")
          stats.shapiro(df_pred["Residuals"])
          ## Complete the code to check p-value

          (Test statistic ,           p-value)

Out[88]:  ShapiroResult(statistic=0.9690961837768555, pvalue=2.130726936518395e-22)
```

Model Assumptions: Test for Homoscedasticity



TEST FOR HOMOSCEDASTICITY

- We will test for homoscedasticity by using the goldfeldquandt test.

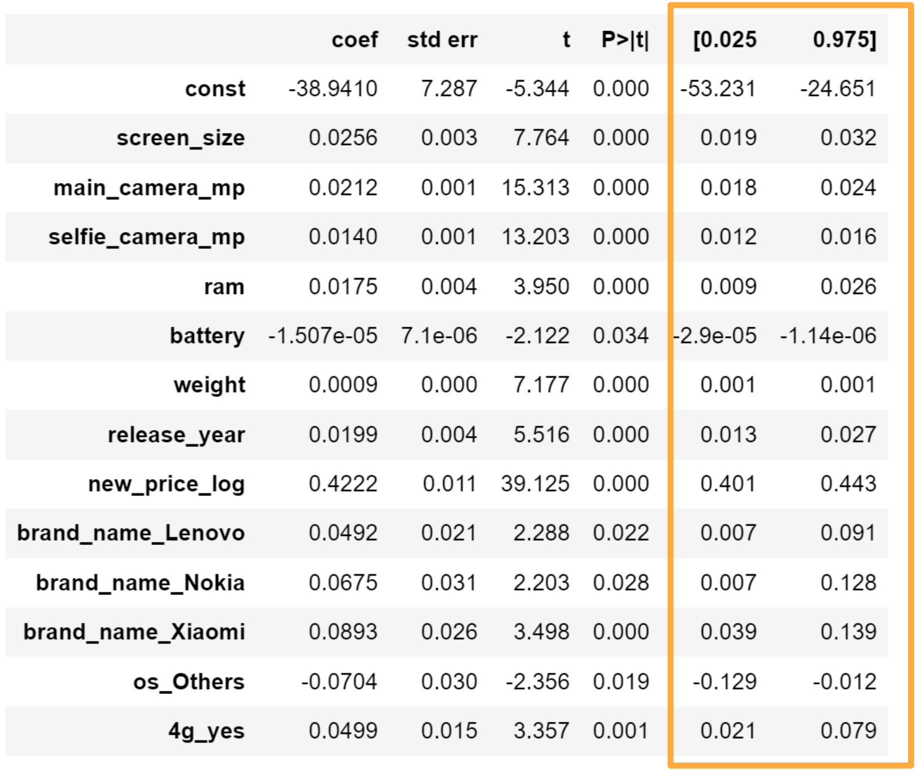
```
In [89]:
import statsmodels.stats.api as sms
from statsmodels.compat import lzip

name = ["F statistic", "p-value"]
test = sms.het_goldfeldquandt(
    df_pred["Residuals"], x_train2
)  ## Complete the code to check homoscedasticity
lzip(name, test)
```

```
Out[89]: [('F statistic', 1.0438035947010265), ('p-value', 0.22944475832466343)]
```

- Since p-value > 0.05 we can say that residuals are homoscedastic

All the assumptions of linear regression are now satisfied!

Confidence intervals



	coef	std err	t	P>|t|	[0.025	0.975]
const	-38.9410	7.287	-5.344	0.000	-53.231	-24.651
screen_size	0.0256	0.003	7.764	0.000	0.019	0.032
main_camera_mp	0.0212	0.001	15.313	0.000	0.018	0.024
selfie_camera_mp	0.0140	0.001	13.203	0.000	0.012	0.016
ram	0.0175	0.004	3.950	0.000	0.009	0.026
battery	-1.507e-05	7.1e-06	-2.122	0.034	-2.9e-05	-1.14e-06
weight	0.0009	0.000	7.177	0.000	0.001	0.001
release_year	0.0199	0.004	5.516	0.000	0.013	0.027
new_price_log	0.4222	0.011	39.125	0.000	0.401	0.443
brand_name_Lenovo	0.0492	0.021	2.288	0.022	0.007	0.091
brand_name_Nokia	0.0675	0.031	2.203	0.028	0.007	0.128
brand_name_Xiaomi	0.0893	0.026	3.498	0.000	0.039	0.139
os_Others	-0.0704	0.030	-2.356	0.019	-0.129	-0.012
4g_yes	0.0499	0.015	3.357	0.001	0.021	0.079

We can be 95% certain that the true value for the coefficient of the predictors lies between the lower and upper values of the confidence intervals.

Link back to Model

Great Learning

Happy Learning !