# ReCell Case Study Project 3

## Supervised Learning Foundation (SLF)

December 2, 2022

# Contents / Agenda

- Executive Summary
- Business Problem Overview and Solution Approach
- EDA Results
- Data Preprocessing
- Model Performance Summary
    - Model Building & Performance Check; Final Model
- Appendix
    - Data Background and Context
    - Model Assumptions Check

# Executive Summary

- The model is able to explain **~84%** of the variation in the data and is able to predict the normalized used price within **+/- 4.5%**
- The most significant predictor of the **normalized_used_price** is **normalized_new_price.** For every unit (euro) increase in **normalized_new_price**, the **normalized_used_price** will increase by 0.44.
- The same holds true for the other significant predictors of the **normalized_used_price.** Those are **years_since_release** (-0.03), **ram** (0.02), **main_camera_mp** (0.02), **selfie_camera_mp** (0.01), **4G** and **5G** networks.
- To capitalize on customer preferences, ***ReCell*** should look for phones that have a higher new price, recent release date, better cameras, more RAM, and have 5G.
- ***ReCell*** should also look for trends in technology that align with these preferences an identify where the peak of customer interest may shift to **the next tech advancement** in device features. Such as 4G phasing out to 5G. Or leveling off in demand for highest resolution cameras. Partner with device makers.
- ***ReCell*** may also want to overlay **customer demographics** and location to better target marketing campaigns. Such as customer more inclined to travel, share photos, or desire the latest technological capabilities and the means to buy.

# Business Problem Overview and Solution Approach

**Context**

- The **used and refurbished device (phones and tablets) market** has had an uptick in demand likely due to considerable savings compared with new models. Other **market drivers for buying and selling** used and refurbished devices include being sold with warranties, insured with proof of purchase, **attractive offers to customers for selling or trading in refurbished devices**, reduces their environmental impact.

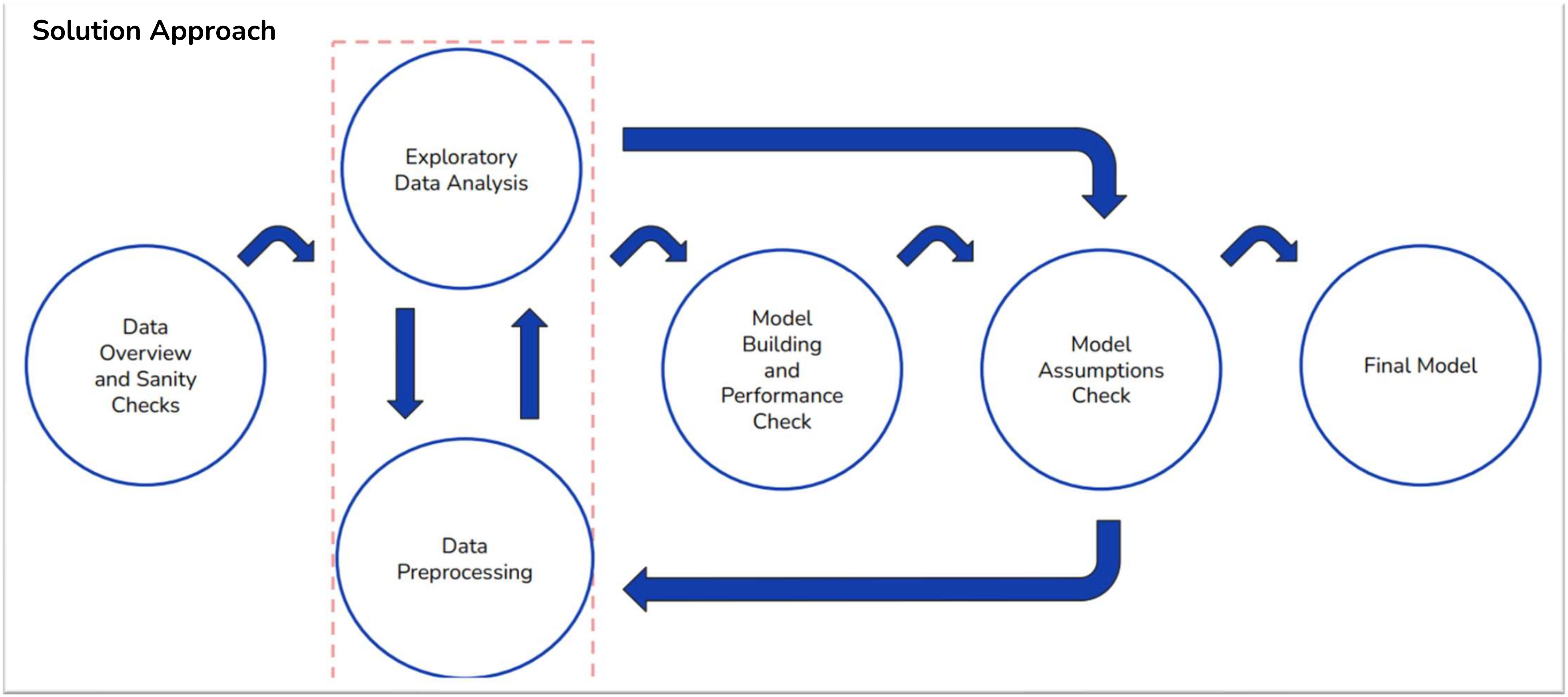
**Objective**

- **ReCell** needs an ML-based solution to **develop a dynamic pricing strategy** for used and refurbished devices. ReCell want to **(1) analyze the data** provided and **(2) build a linear regression model to predict the price of a used phone/tablet** and **(3) identify factors that significantly influence it.**

**Problem definition**

- Predict the price of a used phone/tablet and identify factors that significantly influence it

# Business Problem Overview and Solution Approach



# Business Problem Overview and Solution Approach

**Business Context**

- **Buying and selling used phones and tablets** used to be something that happened on a handful of online marketplace sites. But 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.

- **Refurbished and used devices** continue to provide **cost-effective** alternatives to both consumers and businesses that are looking to save money when purchasing one. There are plenty of **other benefits associated with the used device market**. Used and refurbished devices can be sold with **warranties** and can also be insured with proof of purchase. Third-party vendors/platforms, such as Verizon, Amazon, etc., provide **attractive offers** to customers for refurbished devices. Maximizing the longevity of devices through second-hand trade also reduces their **environmental impact** and helps in recycling and reducing waste. The impact of the COVID-19 outbreak may further boost this segment as consumers cut back on discretionary spending and buy phones and tablets only for immediate needs.

**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, has hired you as a data scientist. They want you to **analyze the data provided** and **build a linear regression model** to **predict the price of a used phone/tablet** and **identify factors that significantly influence** it.

# Business Problem Overview and Solution Approach

**ReCell Data Description**:

- The data contains the different attributes of used/refurbished phones and tablets. The data was collected in the year 2021.

**ReCell Data dictionary** is given below:

- **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 of the device in grams
- **release_year**: Year when the device model was released
- **days_used**: Number of days the used/refurbished device has been used
- **normalized_new_price**: Normalized price of a new device of the same model in euros
- **normalized_used_price**: Normalized price of the used/refurbished device in euros

# EDA Results – Overview & Questions

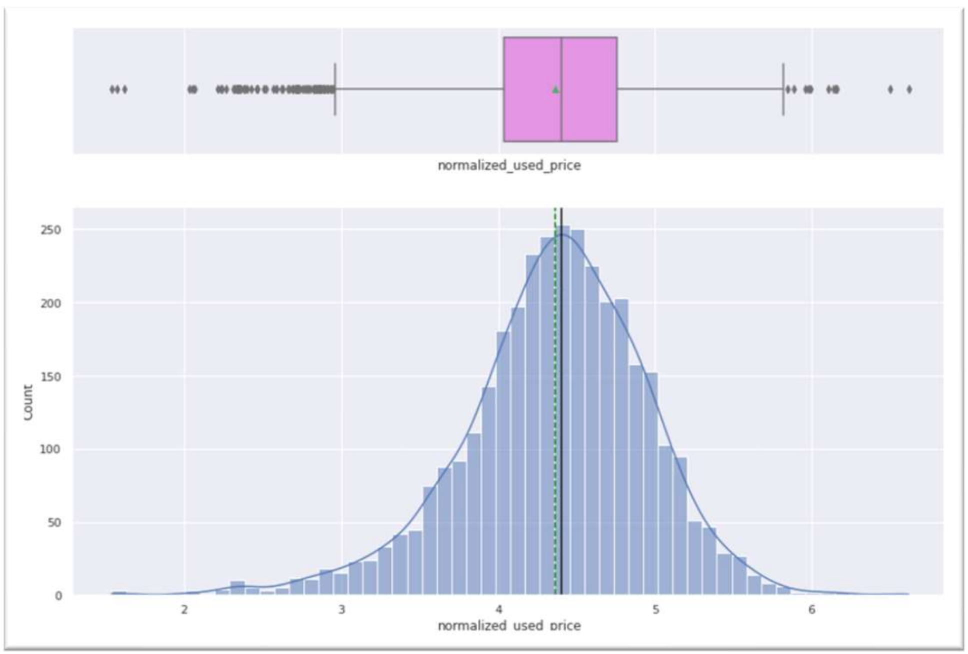
- Please mention the key results from EDA
- - Problem definition, questions to be answered - Data background and contents - Univariate analysis - Bivariate analysis - Insights based on EDA
- Please mention answers to the insight-based questions provided

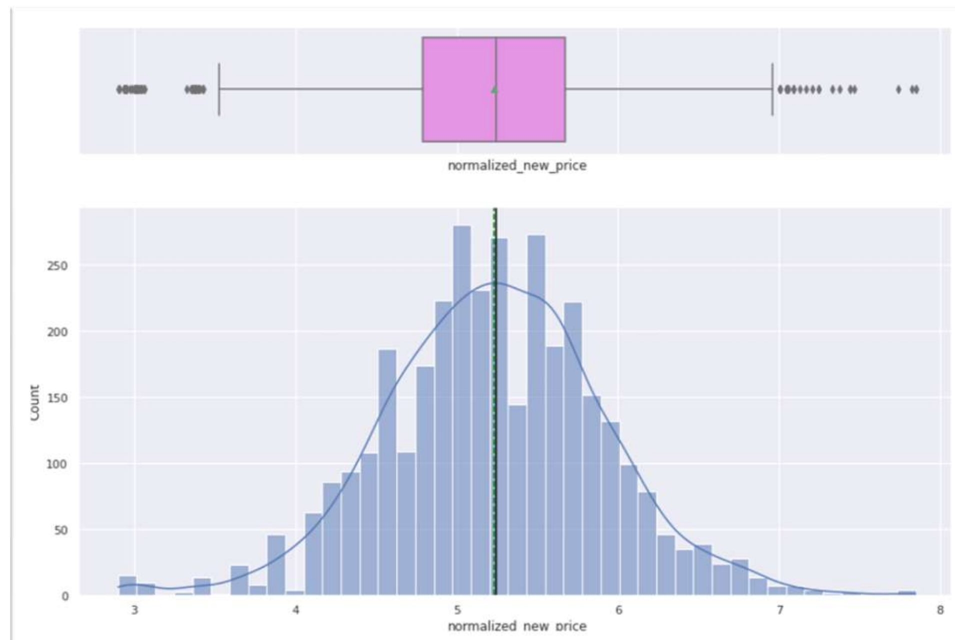
**Questions:**

1. What does the distribution of normalized used device prices look like?
2. What percentage of the used device market is dominated by Android devices?
3. The amount of RAM is important for the smooth functioning of a device. How does the amount of RAM vary with the brand?
4. A large battery often increases a device's weight, making it feel uncomfortable in the hands. How does the weight vary for phones and tablets offering large batteries (more than 4500 mAh)?
5. Bigger screens are desirable for entertainment purposes as they offer a better viewing experience. How many phones and tablets are available across different brands with a screen size larger than 6 inches?
6. A lot of devices nowadays offer great selfie cameras, allowing us to capture our favorite moments with loved ones. What is the distribution of devices offering greater than 8MP selfie cameras across brands?
7. Which attributes are highly correlated with the normalized price of a used device?

# EDA Results – Univariate, price

- The **used price**, **normalized_used_price** and the **new price**, **normalized_new_price** have many outliers and appear to have a normal distribution. **Used price** has a slight left skew.

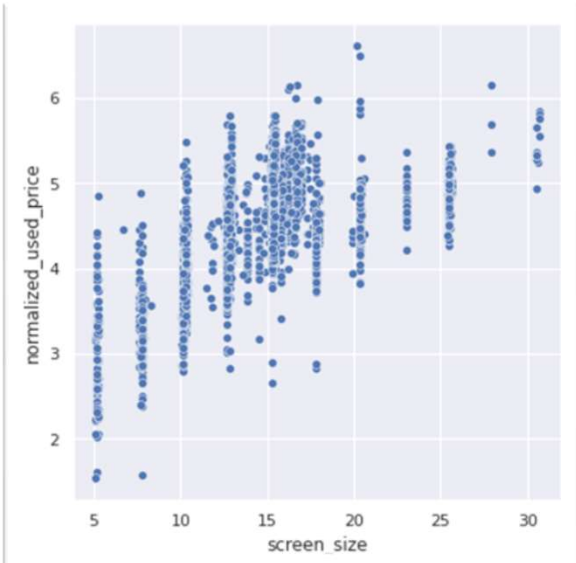


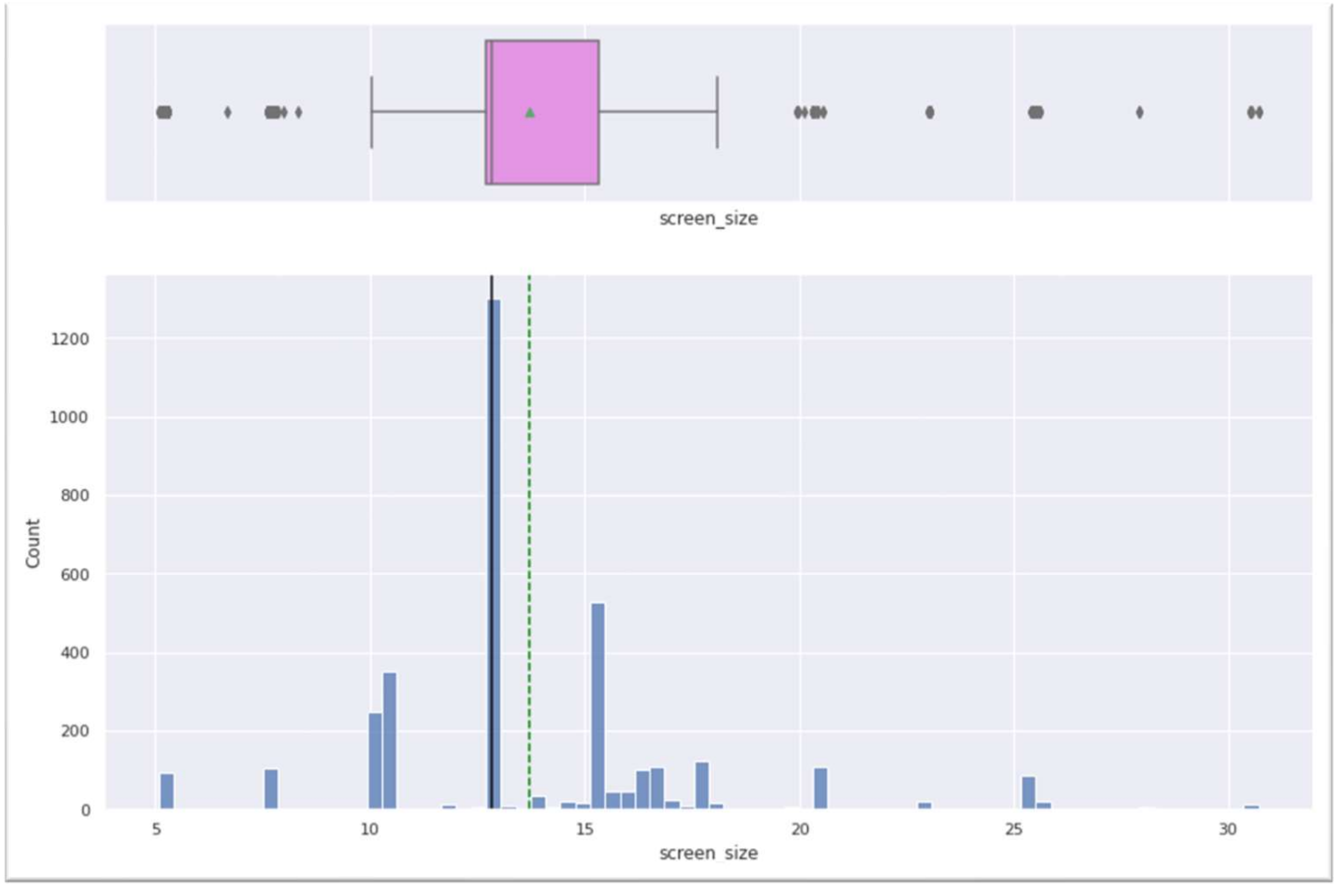




*Link to Appendix slide on data background check*

# EDA Results – Univariate, screen_size

- Screen size

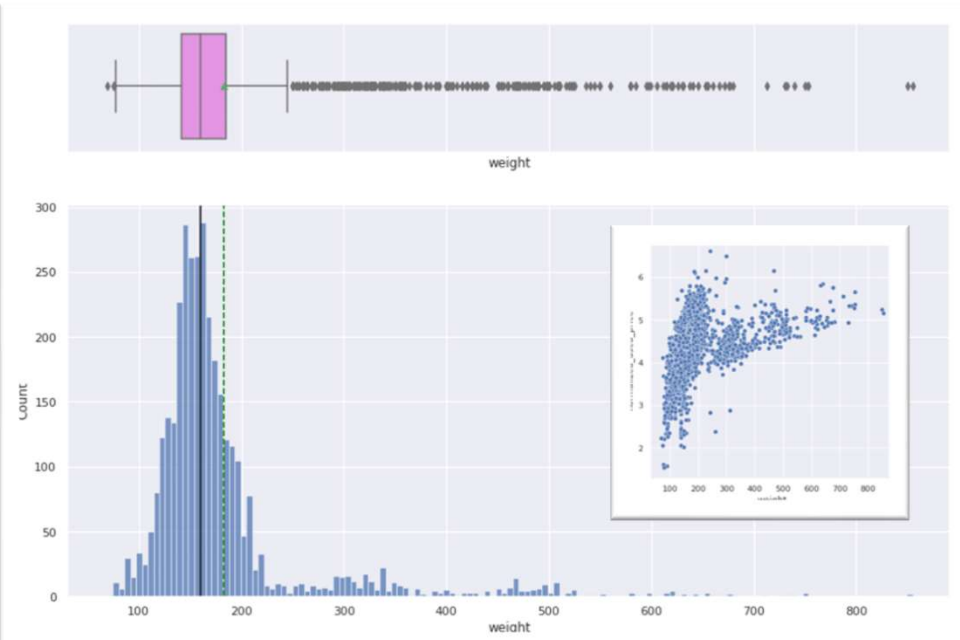


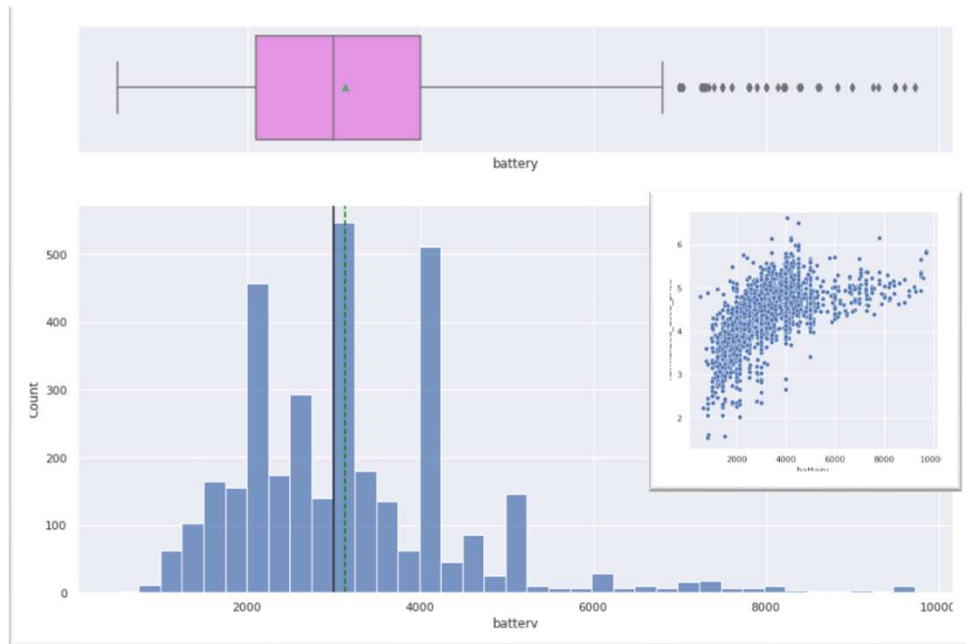




[Link to Appendix slide on data background check]()

# EDA Results – Univariate, weight and battery

- **Weight**, long right tail, many outliers concentrated to the right, possible exponential distribution
- **Battery**, long right tail, many outliers to the right, possible exponential distribution

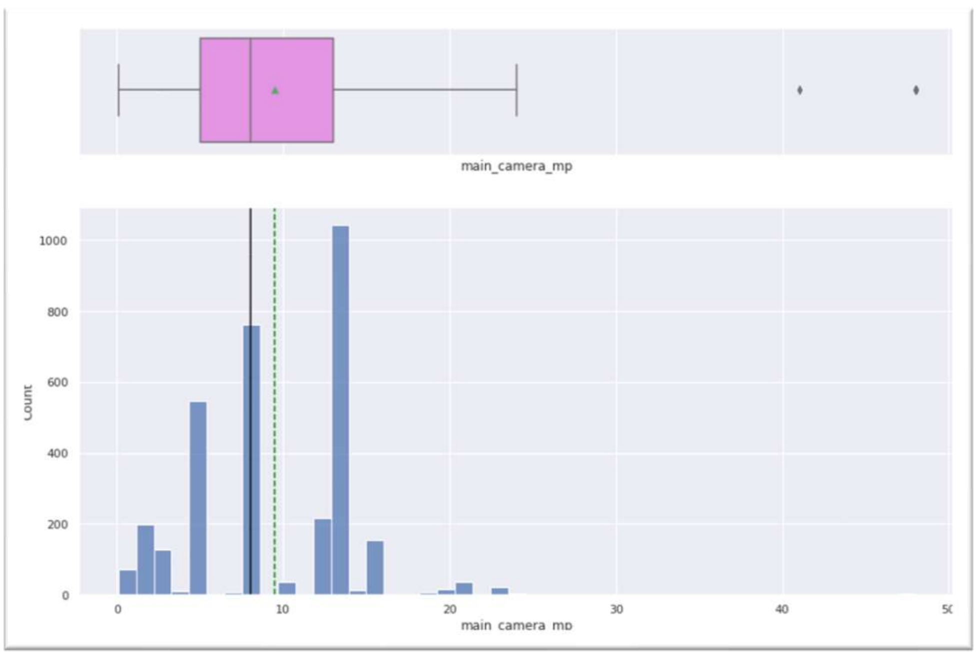


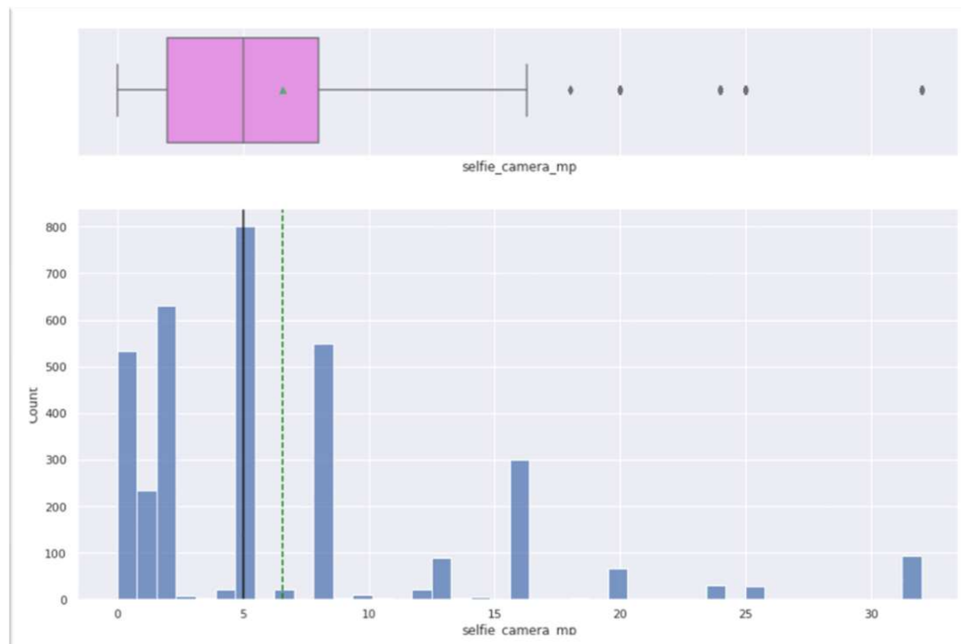




[Link to Appendix slide on data background check]

# EDA Results – Univariate, main camera, selfie camera

- Cameras: main camera (back), selfie camera (front)

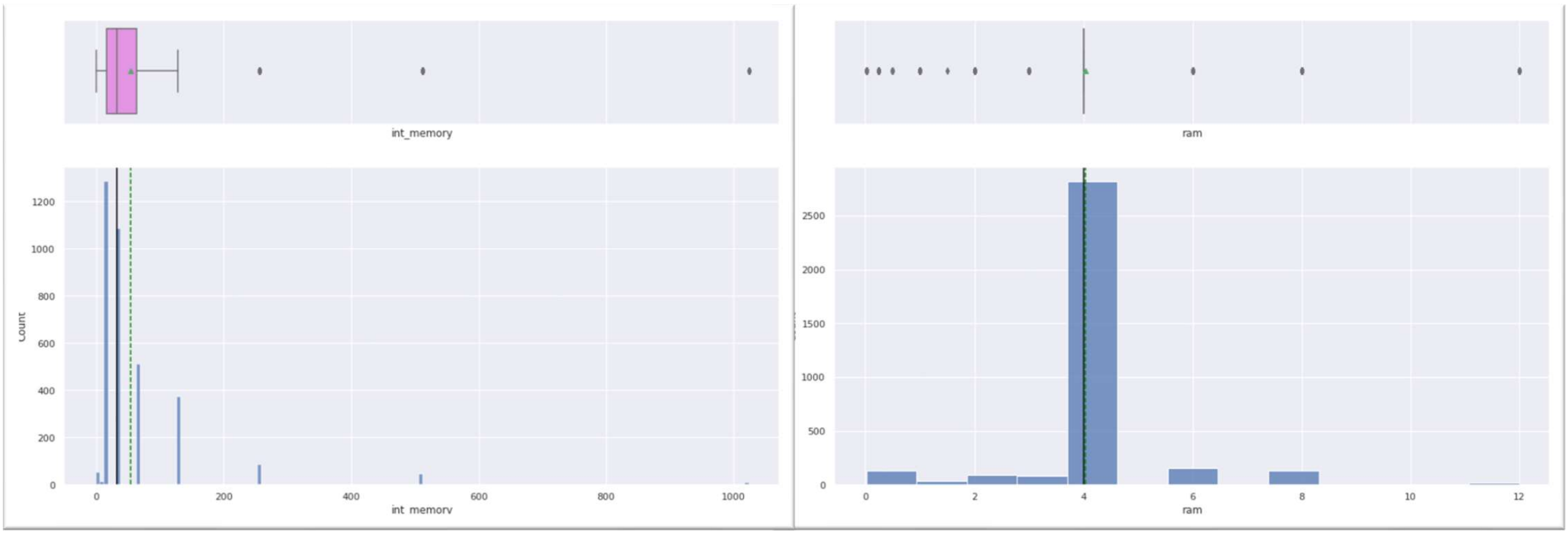



[Link to Appendix slide on data background check](#)

# EDA Results – Univariate

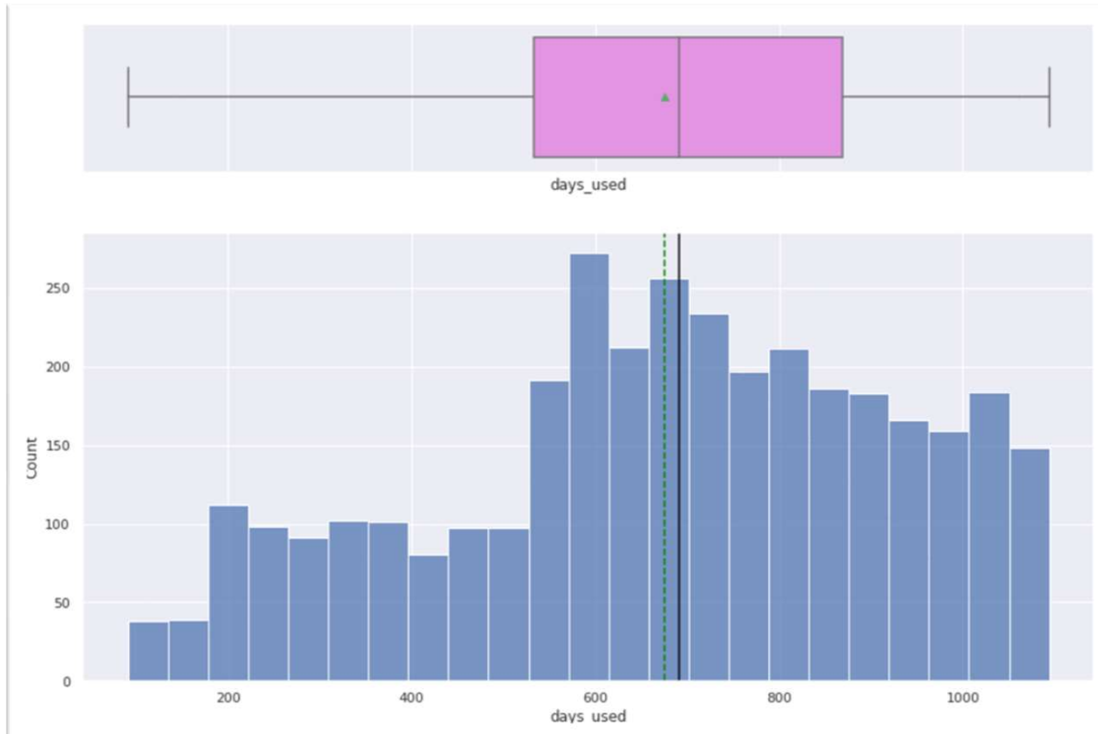
- Memory: internal and RAM




Link to Appendix slide on data background check

# EDA Results – Univariate, days_used

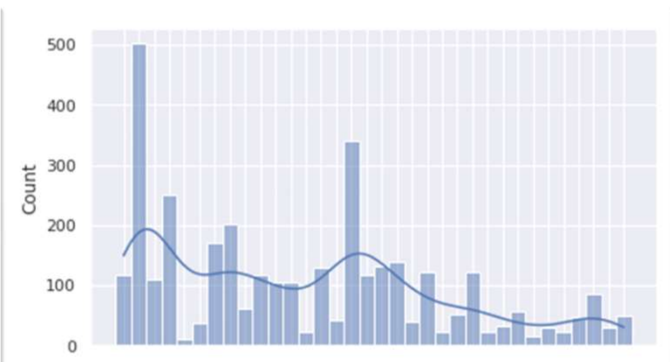
- Days Used

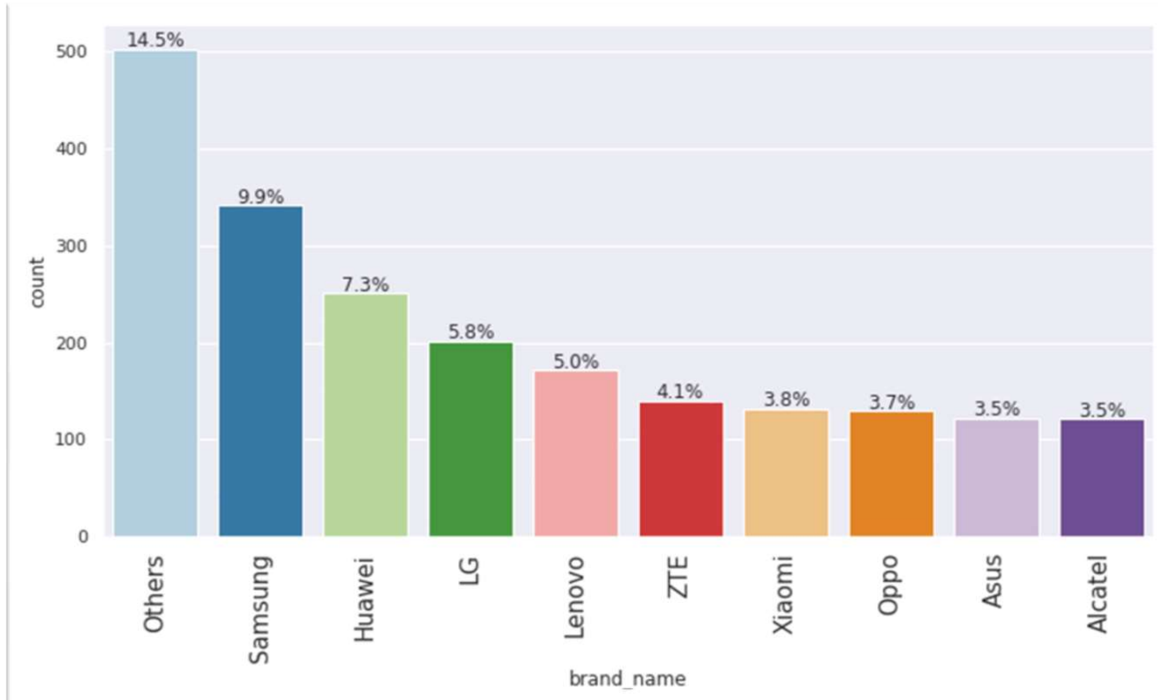



[Link to Appendix slide on data background check]

# EDA Results – Univariate, brand_name

- For **brand_name** count, **Samsung has the most (9.9%)** of the devices, next is **Huawei**, **LG**, and **Lenovo** with **7.3%, 5.8%, 5.0%** respectively
- Others is 14%
- Brand is ad has a wide distribution

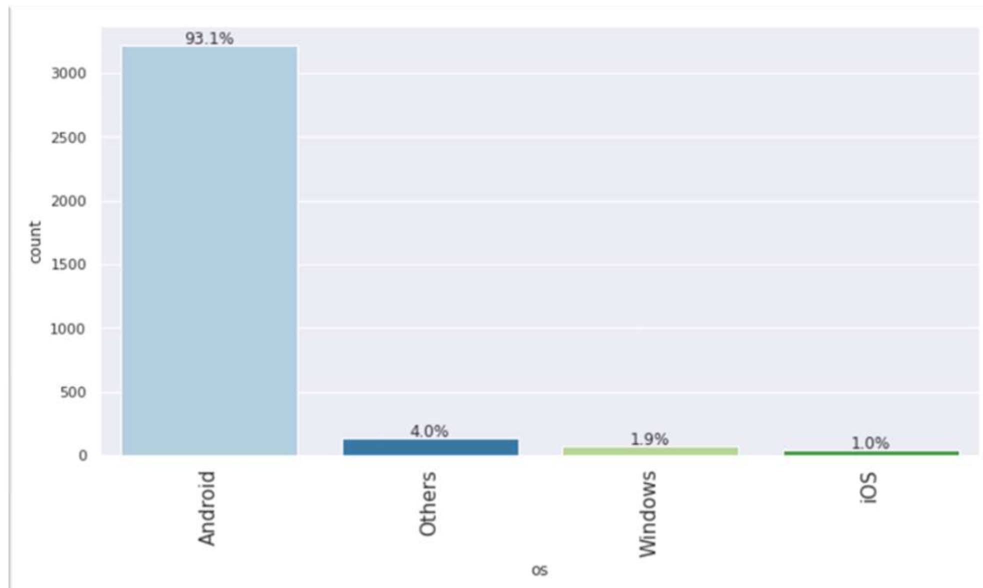



[Link to Appendix slide on data background check]

# EDA Results – Univariate, os operating system

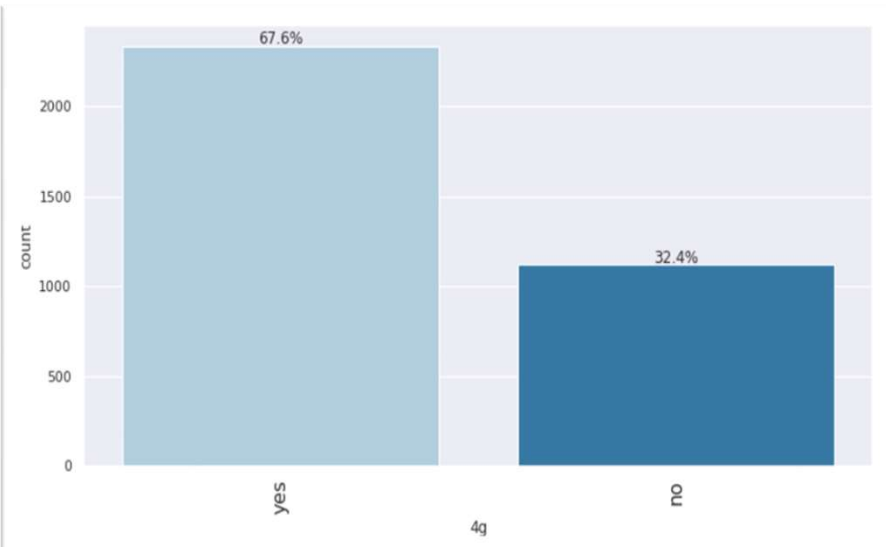
- For OS operating system, **Android dominates the sample dataset with 93%.**
- Then Windows (1.9%) and iOS (1.0%), others collectively is 4.0%

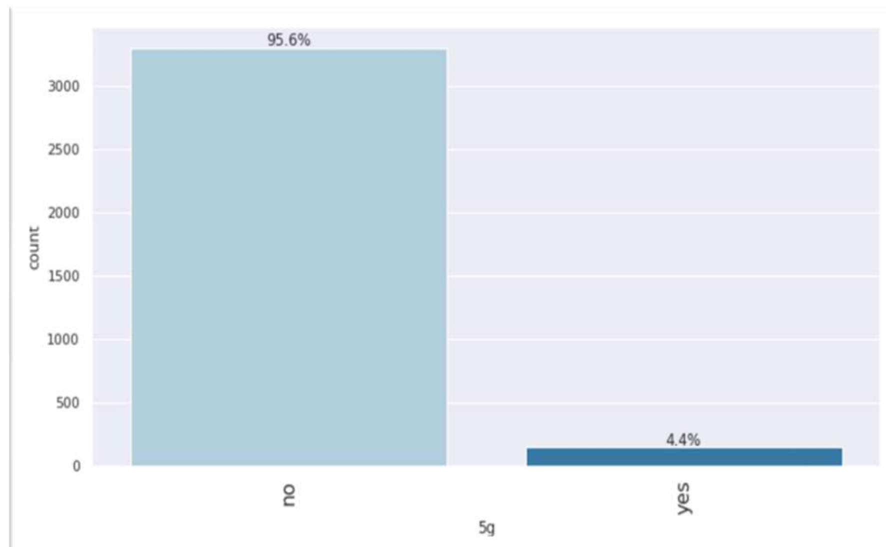



[Link to Appendix slide on data background check]

# EDA Results – Univariate, 4g, 5g networks

- For 4g, yes is 67% no is 32.4%. Yes is twice as much as no. **2 / 3rd of the dataset has 4G.**
- For 5g, no is 95.6%, yes is 4.4%. No is overwhelmingly dominant. Only **less than 5%** of the dataset has 5g



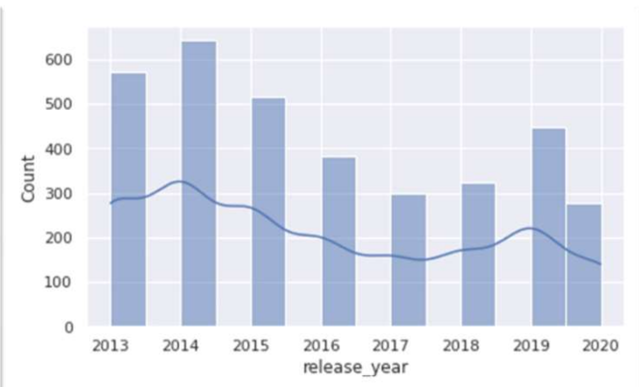




*Link to Appendix slide on data background check*

# EDA Results – Univariate, release_year

- For **release_year**, generally, the earlier years have more of the used and refurbished devices (URD)

- Uptick peaks in 2014 and 2019
- Highest year is 2014 at 18.6%
- Highest increase from 2018 to 2019
- Highest increase from at 12.9% to at 9.3%

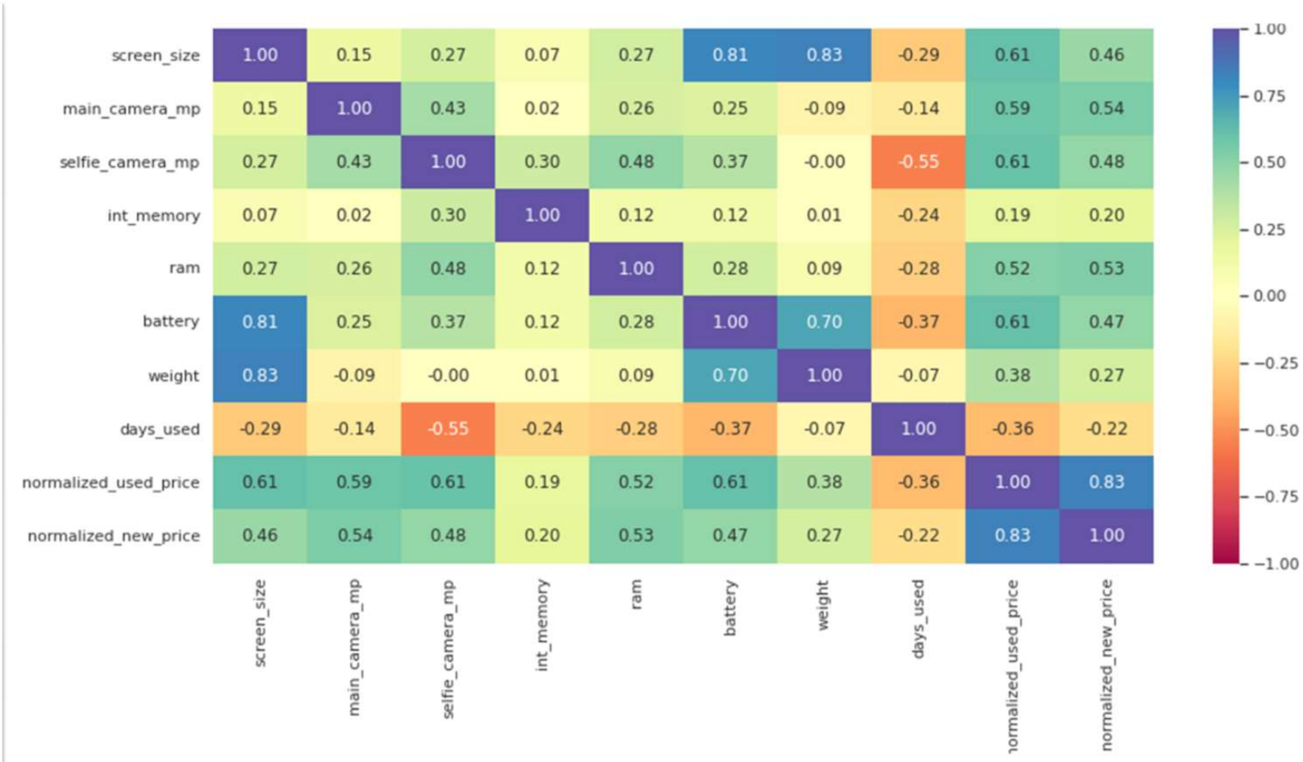




*Link to Appendix slide on data background check*

# EDA Results – Bivariate, Correlation

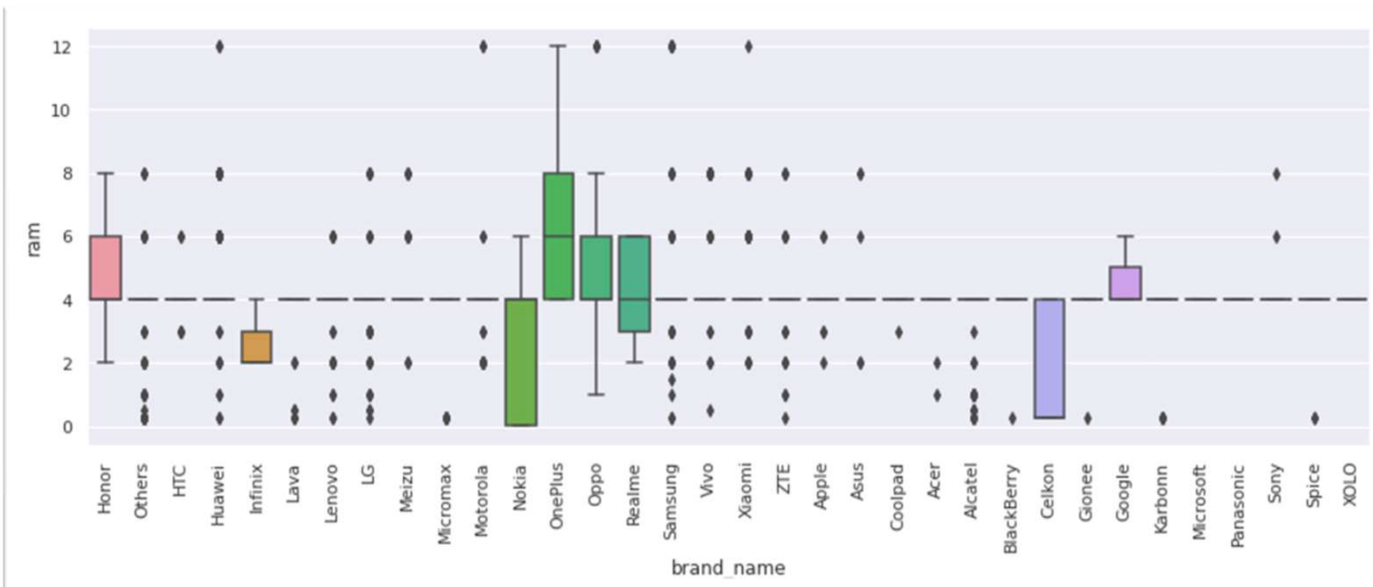
- The **normalized_used_price** and **normalized_new_price** are highly correlated **(0.83)**.
- Three, **screen_size** and **weight** and **battery**, are closely correlated with each other. (0.83), (0.81),(0.70)
- The **days_used is inversely** correlated with all other numerical features; most **inversely** correlated with **selfie_camera (-0.55**)




[Link to Appendix slide on data background check]

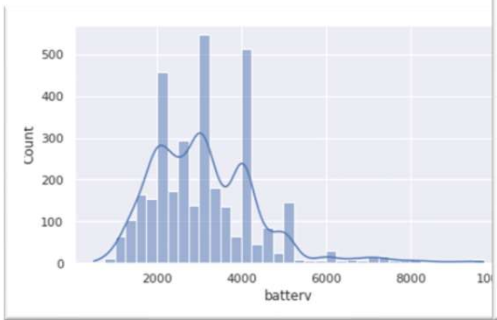
# EDA Results – Bivariate, brand_name vs ram

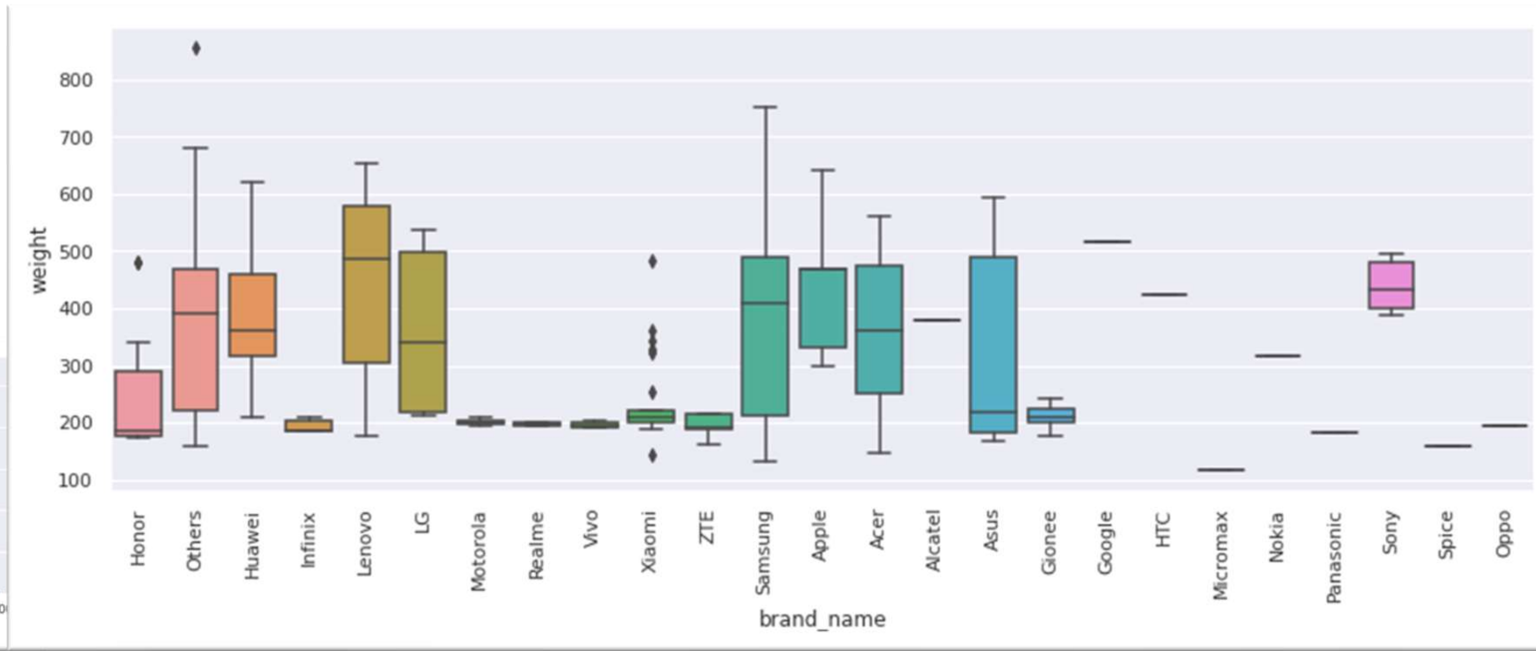
- The amount of RAM variation across brands
- Most all the companies' phones are 4GB for at least 50% of the devices; the smashed interquartile range (between 25 and 75).
- 75% **OnePlus** phones are 4GB or higher, 50% of **OnePlus** phones are 6GB or higher
- 75% of phones are at or below 4GB for **Infinix and Celkon**



# EDA Results – Bivariate, brand_name vs weight, large battery

- On battery capacity above 4500 mAh.
- Many brands with variety of weight options. Motorola, Realme, Vivo, Gionee, and many others have light weight phones with battery capacity above 4500 mAh.

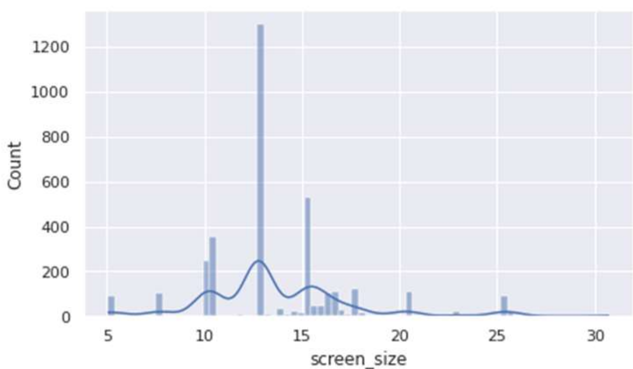


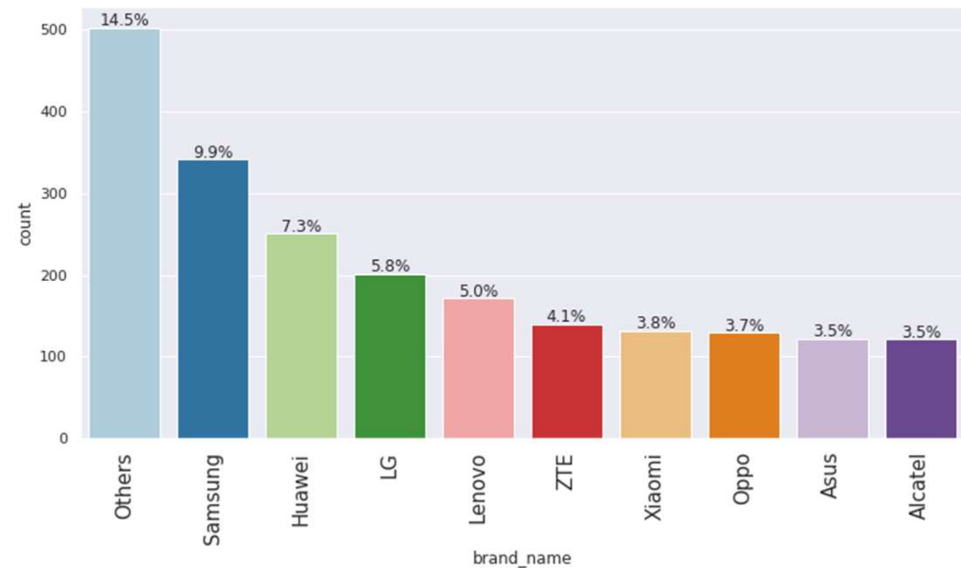




*Link to Appendix slide on data background check*

# EDA Results – Bivariate, brand_name vs count, on screen_size

- For screen size above 6 inches, The highest percentage count is **Huawei at ~14%.** Followed by **Samsung** at **~11%,** and others at **9%**, **Vito at ~7%**
- Four in the middle were all **~6.5 (Honor ,Oppo, Xiaomi)**
- The lowest was **Motorola at ~ 4%,** followed by **LG at 5.4%.**

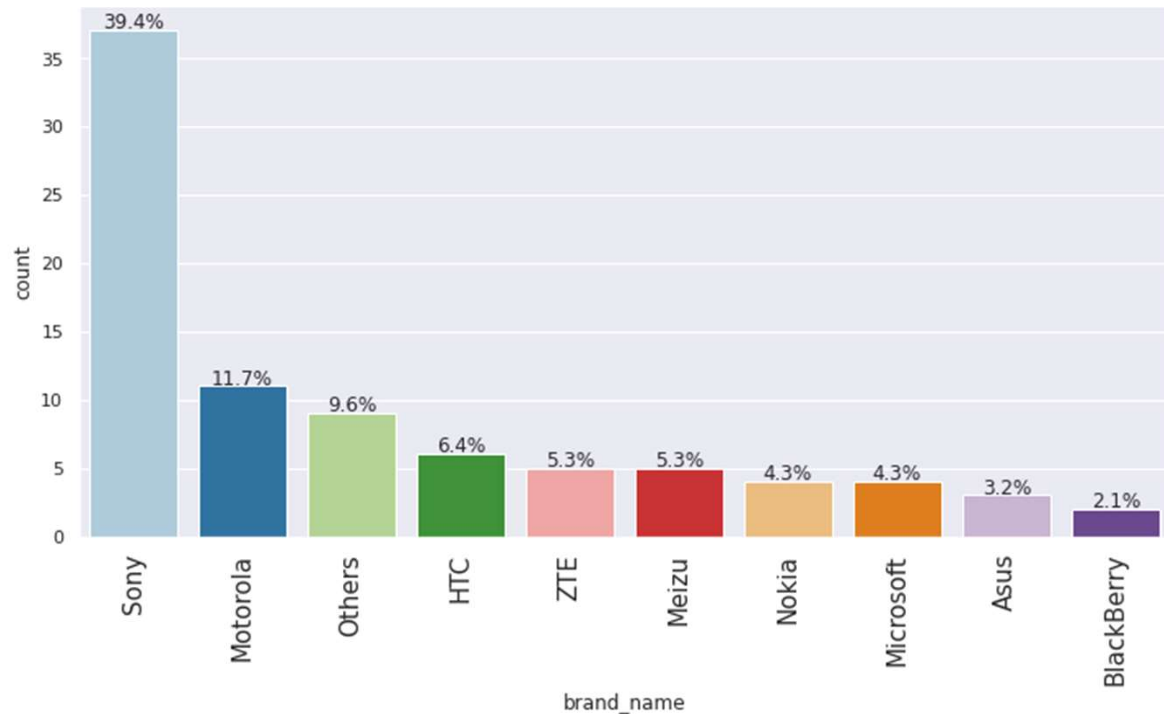



*Link to Appendix slide on data background check*

# EDA Results – Bivariate – Brand main camera above 16 mp

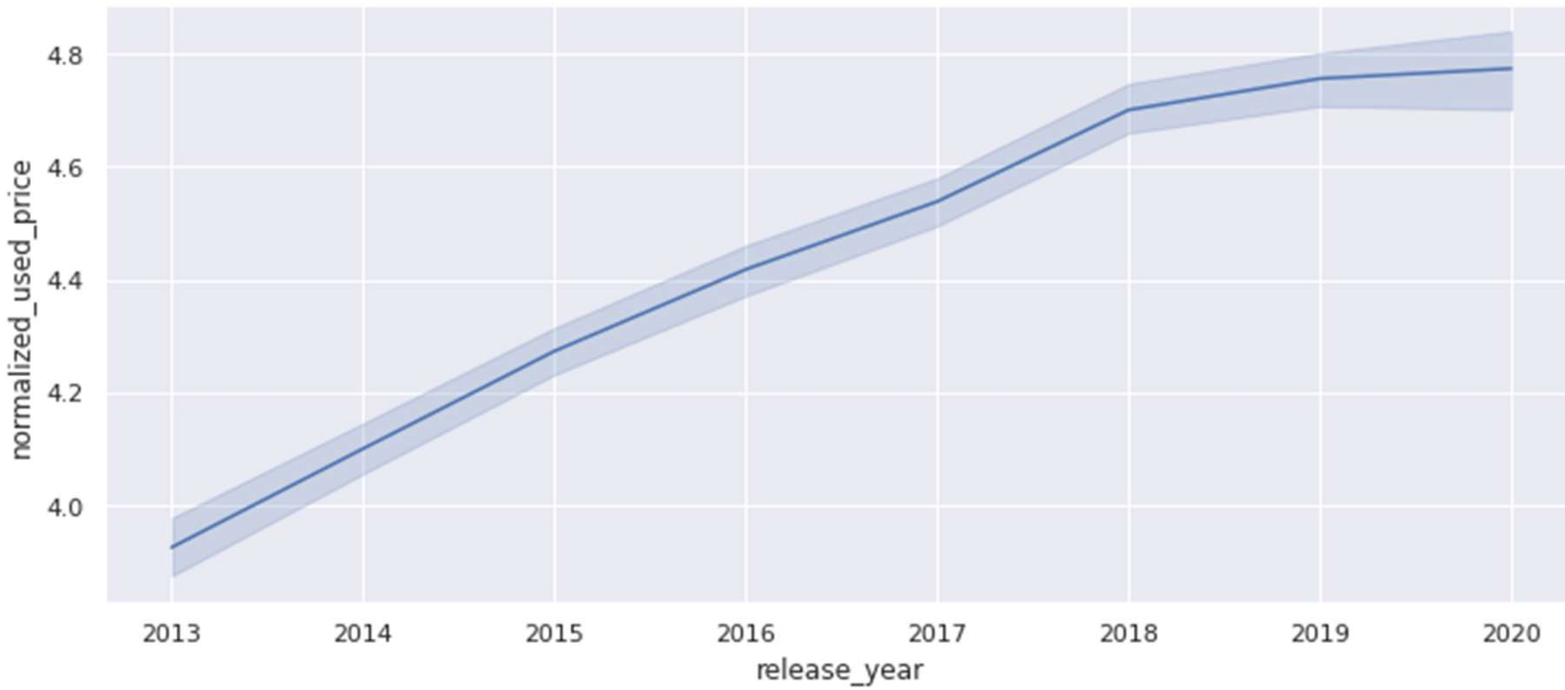
- Main camera above 16 MP
- Sony ~40% of the devices with the main camera above 16 MP




Link to Appendix slide on data background check

# EDA Results – Bivariate

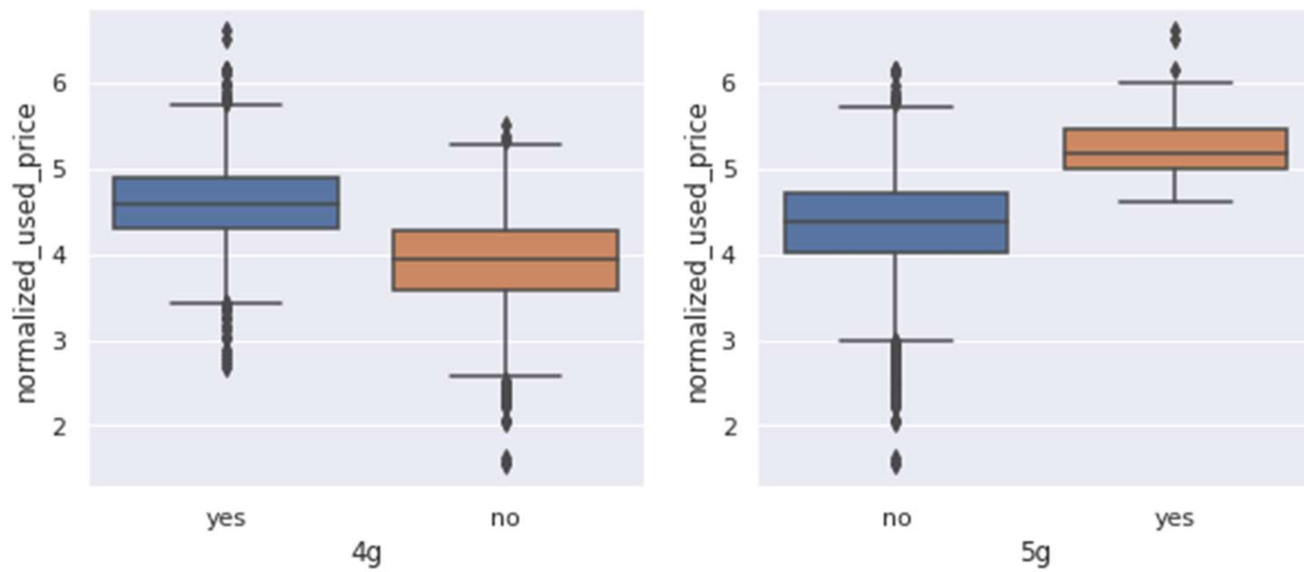
- The normalized used price has been increasing steadily between 2013 and 2020, slightly leveled off in recent years.




*Link to Appendix slide on data background check*

# EDA Results – Bivariate, price variation, 4g & 5g networks

- The used price is higher for phones with at least 4G is higher than those without 4G
- Those used devices that do have 5G have a higher used price than those without 5G
  - As stated in the univariate, less than 5% of the dataset has 5g.
  - This may change in the coming years as 4G is phased out and more phone have 5G




*Link to Appendix slide on data background check*

# Data Preprocessing – Duplicate, Missing Values, Imputation

- There are **no duplicate** values. --------→ --------→ --------→ --------→

```
data.duplicated().sum()
0
```

- **There are missing values**. --------→
  - main_camera_mp 179
  - selfie_camera_mp 2
  - int_memory 4
  - ram 4
  - battery 6
  - weight 7

| | Count | Percentage |
|---|---|---|
| main_camera_mp | 179 | 5.182397 |
| selfie_camera_mp | 2 | 0.057904 |
| int_memory | 4 | 0.115808 |
| ram | 4 | 0.115808 |
| battery | 6 | 0.173712 |
| weight | 7 | 0.202664 |

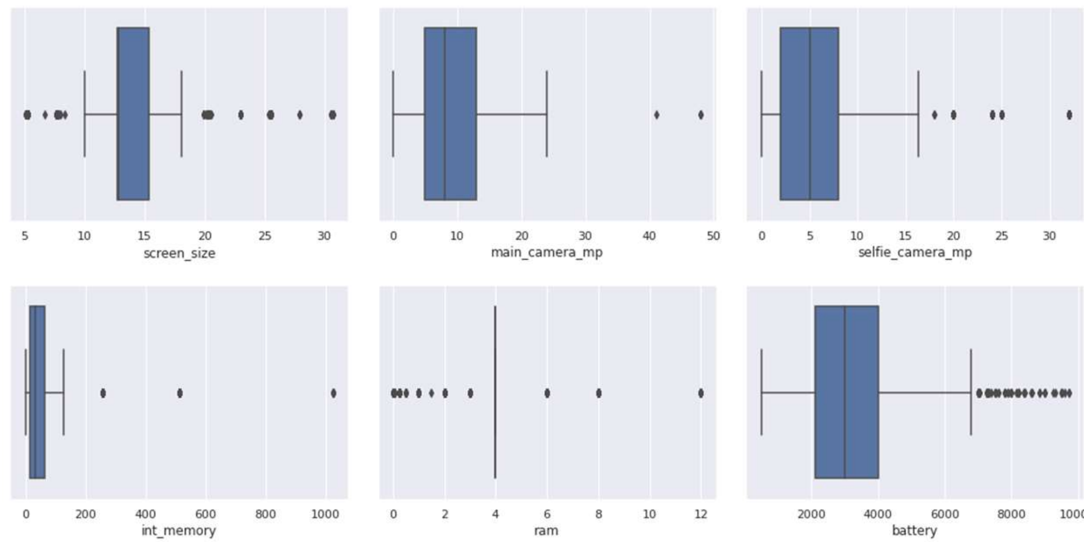
- The most missing values are for **main_camera_mp** at 179. These are spread over 15 different brands. But increasingly concentrated over the release years 2017 to 2020. --------→

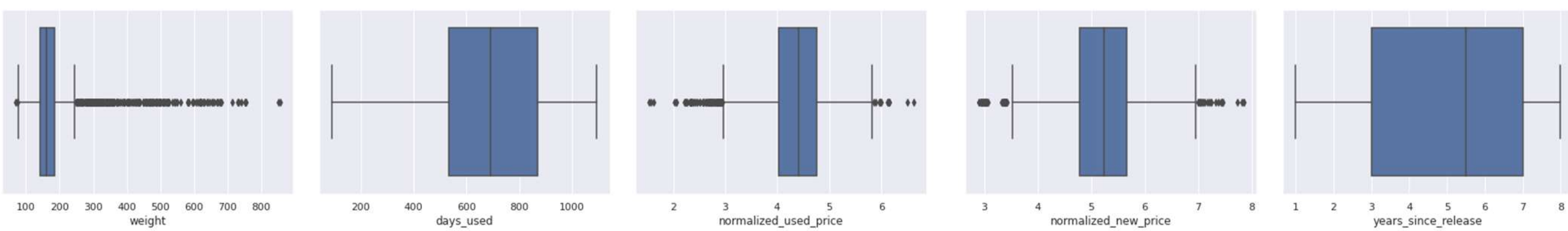
```
2020    122
2019     45
2018     10
2017      2
```

- The missing values were **imputed** by the column **medians** each column grouped by **release_year** and **brand_name.** This helped fill **int_memory** and **ram**. Used medians because of many outliers
- Then imputed by the column **medians** grouped by **brand_name** only. This helped all but 10 in **main_camera_mp**.
- Then filled the remaining 10 in **main_camera_mp** by the column **median.**

# Data Preprocessing – Outlier check (treatment if needed)

- Outliers in nearly all columns, but we are not treating them because we see no curent evedence that they are erronious.



# Data Preprocessing – Feature engineering

- Created a new column **years_since_release** from the **release_year column**.
- Consider the year of data collection, 2021, as the baseline, 0; 2020 as 1 year, 2019 as 2 years and so on. Dropped the **release_year** column.

```
years_since_release
```

| | |
|---|---|
| count | 3454.000000 |
| mean | 5.034742 |
| std | 2.298455 |
| min | 1.000000 |
| 25% | 3.000000 |
| 50% | 5.500000 |
| 75% | 7.000000 |
| max | 8.000000 |

- Data processing post-model and assumption checks may require transforming variables. To be determined in next iteration.

# Data Preprocessing – Data preparation for modeling

- The objective is to predict the **normalized_used_price** of devices so we define that as **y, dependent variable, target variable.**
- All other columns as **X, independent variables**.
- Added intercept to data.
- Used one hot encoding on categorical features to create dummy variables
- Split the data into train (70% of data) and test (30% of data).

```
X.columns
```

```
Index(['const', 'screen_size', 'main_camera_mp', 'selfie_camera_mp',
       'int_memory', 'ram', 'battery', 'weight', 'days_used',
       'normalized_new_price', 'years_since_release', 'brand_name_Alcatel',
       'brand_name_Apple', 'brand_name_Asus', 'brand_name_BlackBerry',
       'brand_name_Celkon', 'brand_name_Coolpad', 'brand_name_Gionee',
       'brand_name_Google', 'brand_name_HTC', 'brand_name_Honor',
       'brand_name_Huawei', 'brand_name_Infinix', 'brand_name_Karbonn',
       'brand_name_LG', 'brand_name_Lava', 'brand_name_Lenovo',
       'brand_name_Meizu', 'brand_name_Micromax', 'brand_name_Microsoft',
       'brand_name_Motorola', 'brand_name_Nokia', 'brand_name_OnePlus',
       'brand_name_Oppo', 'brand_name_Others', 'brand_name_Panasonic',
       'brand_name_Realme', 'brand_name_Samsung', 'brand_name_Sony',
       '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'],
      dtype='object')
```

```
Number of rows in train data = 2417
Number of rows in test data = 1037
```

# Model Building – Linear Regression Results – Baseline

- Built model of ordinary least squares (OLS) regressions
- The value for **R-squared** is **0.845**
- The value for **adjusted R**-squared is **0.842.** The **baseline model is able to explain ~85% of the variance.**

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: | Mon, 28 Nov 2022 | Prob (F-statistic): | 0.00 |
| Time: | 18:11:28 | 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 |
| 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 Building – Linear Regression Results – Baseline

- Baseline **coefficients (coef)** of the predictor variable excluding dummy variables are shown here.
- The highest coefficient is **normalized_new_price** with **0.43**, meaning this is the most **significant predictor**.
- For every unit (euro) increase in **normalized_new_price**, the **normalized_used_price** we are trying to predict will increases by **0.43.**
- **P-value** is **0.000**, meaning we **can trust the 0.43 coef as significant**.

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: | Mon, 28 Nov 2022 | Prob (F-statistic): | 0.00 |
| Time: | 18:11:28 | 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 |
| 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 Building – Linear Regression Results – Baseline

- **Baseline significance coef and 0.000 p-value.**

- normalized_new_price 0.4311
- screen_size, 0.0244
- years_since_release -0.0237
- ram 0.0230
- main_camera_mp 0.0208
- selfie_camera_mp 0.0135
- 4g 0.0528
- 5g -0.0714

- **Might remove the non-significant predictor variables due to negligible coeff and or, high p-value indicating insignificance.**

- int_memory
- days_used
- battery

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: | Mon, 28 Nov 2022 | Prob (F-statistic): | 0.00 |
| Time: | 18:11:28 | 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 |
| 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 Building – Linear Regression Results – Baseline

- Baseline for brand dummy variables

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| 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 |

# Model Building – Linear Regression Results – Final

- Tested baseline model performance
- Tested assumptions.
- **See appendix for assumption details.**
- **Removed multicollinearity**.
- **Dropped high p-value variables**.
- Re-ran the model.
- Compared statistics
- Re-evaluated model performance
- Reduced the predictive variables while maintain the performance

Training Performance

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

Test Performance

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

```
                            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:                   Sun, 04 Dec 2022   Prob (F-statistic):               0.00
Time:                           07:40:16   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.
```

# Model Building – Linear Regression Results – Final

- Final results have reduced the predictive variables while maintain the performance.
- Final **coefficients (coef)** and the **most significant predictors** of the **normalized_used_price** variable are shown here.

- normalized_new_price 0.4415
- years_since_release, -0.0292
- ram 0.0207
- main_camera_mp 0.0210
- selfie_camera_mp 0.0138
- weight 0.0017
- 4g 0.0502
- 5g -0.0673

- For every unit (euro) increase in **normalized_new_price**, the **normalized_used_price** we are trying to predict will increases by **0.44.**
- The same holds true for the other predictors.

target = intercept + constant1*feature1 + constant2*feature2 + constant3*feature3 + .....

```
                            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:                   Sun, 04 Dec 2022   Prob (F-statistic):               0.00
Time:                           07:40:16   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.
```

# Model Performance Summary –

## Evaluation Metrics for a Regression Model

| R-squared | Adjusted R-squared | Mean Absolute Error | Root Mean Square Error |
|---|---|---|---|
| • Measure of the % of variance in the target variable explained by the model<br>• Generally the first metric to look at for linear regression model performance<br>• Higher the better | • Conceptually, very similar to R-squared but penalizes for the addition of too many variables<br>• Generally used when you have too many variables as adding more variables always increases R^2 but not Adjusted R^2<br>• Higher the better | • Simplest metric to check prediction accuracy<br>• Same unit as the dependent variable<br>• Not sensitive to outliers i.e. errors doesn't increase too much if there are outliers<br>• Difficult to optimize from a mathematical point of view (pure maths logic)<br>• Lower the better | • Another metric to measure the accuracy of prediction<br>• Same unit as the dependent variable<br>• Sensitive to outliers - errors will be magnified due to the square function<br>• But has other mathematical advantages that will be covered later<br>• Lower the better |

*Link to Appendix slide on model assumptions*

# Model Performance Summary – Baseline vs Final

- The **training R-squared is 0.844** so the model is **not underfitting.**
- The **train and test RMSE and MAE** are **comparable**, so the model is **not overfitting**

- **MAE** suggests that the model can **predict the normalized used price** within a **mean error of 0.18** on the test data. That is +/- 0.18.
- **MAPE** of 4.5 on the test data indicates the model is able to **predict the normalized used price within 4.5%.** That is +/- 4.5%

- See chart of test vs training and compare the **adjusted R-squared is 84.2% vs 83.5%**.

Training Performance

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

Test Performance

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

- The training and testing metrics for the model were the **same as the baseline model within rounding.** The model was not underfitting, not overfitting, and able to predict the normalized used price within +/- 4.5%.
- See chart of test vs training and compare the **adjusted R-squared is 83.8% vs 83.6%**.

Training Performance

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

Test Performance

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

# APPENDIX

# Data Background and Context – Appendix

- Please mention about the data background and contents

# Data Background and Context – Business Context & Objectives

**Context**

- Buying and selling used phones and tablets used to be something that happened on a handful of online marketplace sites. But 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.

- Refurbished and used devices continue to provide cost-effective alternatives to both consumers and businesses that are looking to save money when purchasing one. There are plenty of other benefits associated with the used device market. Used and refurbished devices can be sold with warranties and can also be insured with proof of purchase. Third-party vendors/platforms, such as Verizon, Amazon, etc., provide attractive offers to customers for refurbished devices. Maximizing the longevity of devices through second-hand trade also reduces their environmental impact and helps in recycling and reducing waste. The impact of the COVID-19 outbreak may further boost this segment as consumers cut back on discretionary spending and buy phones and tablets only for immediate needs.

**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, has hired you as a data scientist. They want you to analyze the data provided and build a linear regression model to predict the price of a used phone/tablet and identify factors that significantly influence it.

# Data Background and Context – ReCell Dataset

**Data Description**:
- The data contains the different attributes of used/refurbished phones and tablets. The data was collected in the year 2021.

**Data dictionary** is given below:
- **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 of the device in grams
- **release_year**: Year when the device model was released
- **days_used**: Number of days the used/refurbished device has been used
- **normalized_new_price**: Normalized price of a new device of the same model in euros
- **normalized_used_price**: Normalized price of the used/refurbished device in euros

# DATA OVERVIEW – Rows, Columns, Data types

**Rows and Columns**

- 3,454 rows,
- 15 columns

**Data types include**

- Eleven (11) **numeric**, consisting of
- nine (9) float64 and
- two (2) int64.

- Four (4) **categorical**, consisting of
- four (4) object type.

**There are missing values**

```
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)
```

# DATA OVERVIEW – Statistical Summary, Numerical

Numerical columns descriptive statistics summary

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| screen_size | 3454.0 | 13.713115 | 3.805280 | 5.080000 | 12.700000 | 12.830000 | 15.340000 | 30.710000 |
| main_camera_mp | 3275.0 | 9.460208 | 4.815461 | 0.080000 | 5.000000 | 8.000000 | 13.000000 | 48.000000 |
| selfie_camera_mp | 3452.0 | 6.554229 | 6.970372 | 0.000000 | 2.000000 | 5.000000 | 8.000000 | 32.000000 |
| int_memory | 3450.0 | 54.573099 | 84.972371 | 0.010000 | 16.000000 | 32.000000 | 64.000000 | 1024.000000 |
| ram | 3450.0 | 4.036122 | 1.365105 | 0.020000 | 4.000000 | 4.000000 | 4.000000 | 12.000000 |
| battery | 3448.0 | 3133.402697 | 1299.682844 | 500.000000 | 2100.000000 | 3000.000000 | 4000.000000 | 9720.000000 |
| weight | 3447.0 | 182.751871 | 88.413228 | 69.000000 | 142.000000 | 160.000000 | 185.000000 | 855.000000 |
| release_year | 3454.0 | 2015.965258 | 2.298455 | 2013.000000 | 2014.000000 | 2015.500000 | 2018.000000 | 2020.000000 |
| days_used | 3454.0 | 674.869716 | 248.580166 | 91.000000 | 533.500000 | 690.500000 | 868.750000 | 1094.000000 |
| normalized_used_price | 3454.0 | 4.364712 | 0.588914 | 1.536867 | 4.033931 | 4.405133 | 4.755700 | 6.619433 |
| normalized_new_price | 3454.0 | 5.233107 | 0.683637 | 2.901422 | 4.790342 | 5.245892 | 5.673718 | 7.847841 |

# DATA OVERVIEW – Statistical Summary, Categorical

## Categorical columns, Unique values and value counts

- brand_name --------→ --------→ --------→ --------→
- os ---------→

```
----------------------------
Android    3214
Others      137
Windows      67
iOS          36
Name: os, dtype: int64
Android    0.930515
Others     0.039664
Windows    0.019398
iOS        0.010423
Name: os, dtype: float64
```

- 4g --------→ --------→ --------→

```
----------------------------
yes    2335
no     1119
Name: 4g, dtype: int64
yes    0.676028
no     0.323972
Name: 4g, dtype: float64
----------------------------
```

- 5g ----→

```
----------------------------
no     3302
yes     152
Name: 5g, dtype: int64
no     0.955993
yes    0.044007
Name: 5g, dtype: float64
----------------------------
```

```
Others        502
Samsung       341
Huawei        251
LG            201
Lenovo        171
ZTE           140
Xiaomi        132
Oppo          129
Asus          122
Alcatel       121
Micromax      117
Vivo          117
Honor         116
HTC           110
Nokia         106
Motorola      106
Sony           86
Meizu          62
Gionee         56
Acer           51
XOLO           49
Panasonic      47
Realme         41
Apple          39
Lava           36
Celkon         33
Spice          30
Karbonn        29
Coolpad        22
BlackBerry     22
Microsoft      22
OnePlus        22
Google         15
Infinix        10
```

```
Others        0.145339
Samsung       0.098726
Huawei        0.072669
LG            0.058193
Lenovo        0.049508
ZTE           0.040533
Xiaomi        0.038217
Oppo          0.037348
Asus          0.035321
Alcatel       0.035032
Micromax      0.033874
Vivo          0.033874
Honor         0.033584
HTC           0.031847
Nokia         0.030689
Motorola      0.030689
Sony          0.024899
Meizu         0.017950
Gionee        0.016213
Acer          0.014765
XOLO          0.014186
Panasonic     0.013607
Realme        0.011870
Apple         0.011291
Lava          0.010423
Celkon        0.009554
Spice         0.008686
Karbonn       0.008396
Coolpad       0.006369
BlackBerry    0.006369
Microsoft     0.006369
OnePlus       0.006369
Google        0.004343
Infinix       0.002895
```

# DATA OVERVIEW – Rows, Columns, Data types

**Rows and Columns**

- 3,454 rows,
- 15 columns

**Data types include**

- Eleven (11) **numeric**, consisting of
- nine (9) float64 and
- two (2) int64.

- Four (4) **categorical**, consisting of
- four (4) object type.

**There are no duplicate values**

**There are missing values**

```
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)
```

# DATA OVERVIEW – Missing Values, Check, Percentage

Checked for missing values and percentages

```
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
```

| | Count | Percentage |
|---|---|---|
| main_camera_mp | 179 | 5.182397 |
| selfie_camera_mp | 2 | 0.057904 |
| int_memory | 4 | 0.115808 |
| ram | 4 | 0.115808 |
| battery | 6 | 0.173712 |
| weight | 7 | 0.202664 |

# Model Assumptions – Overview

- Checking the following Linear Regression assumptions:
    - **No Multicollinearity** among independent variables
    - **Linearity of variables**. There should be a linear relationship between dependent and independent variables.
    - **Independence of error terms**. The residuals should be independent of each other.
    - **Normality of error terms**. The residuals must be normally distributed.
    - **No Heteroscedasticity**. The residuals must have constant variance.
- Tests conducted for checking model assumptions and the Results obtained

# Model Assumptions – No Multicollinearity

- There should be **no multicollinearity** among independent variables.
- Tested using VIF. Dummy variables were excluded from consideration. Two showed moderate multicollinearity (above 5), **screen_size** (~7.7) and **weight** ( ~6.4).
- Dropped **screen_size** because it had the **least impact on the adjusted R-squared** of the model and re-ran VIF. The resulting VIF of **weight** decreased to 2.99.

| | 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 |

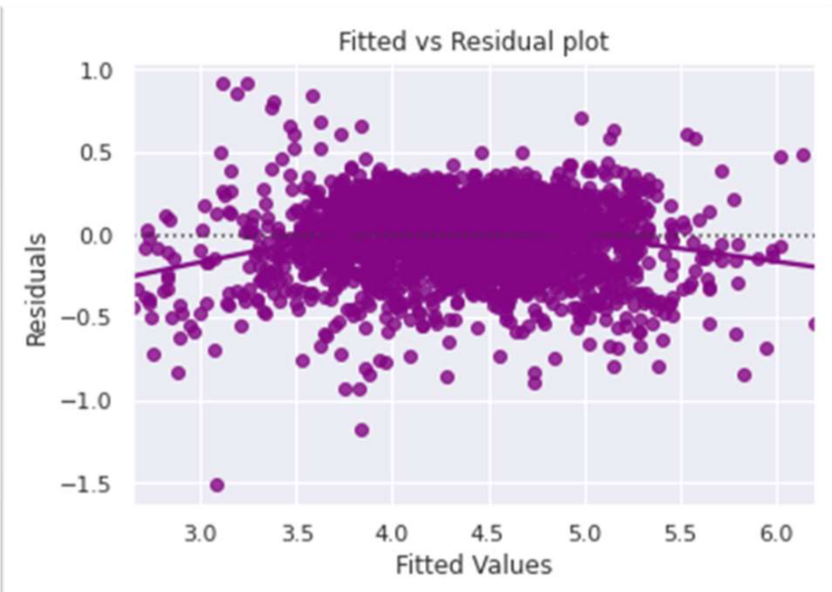
| | col | Adj. R-squared after_dropping col | RMSE after dropping col |
|---|---|---|---|
| 0 | screen_size | 0.838381 | 0.234703 |
| 1 | weight | 0.838071 | 0.234928 |

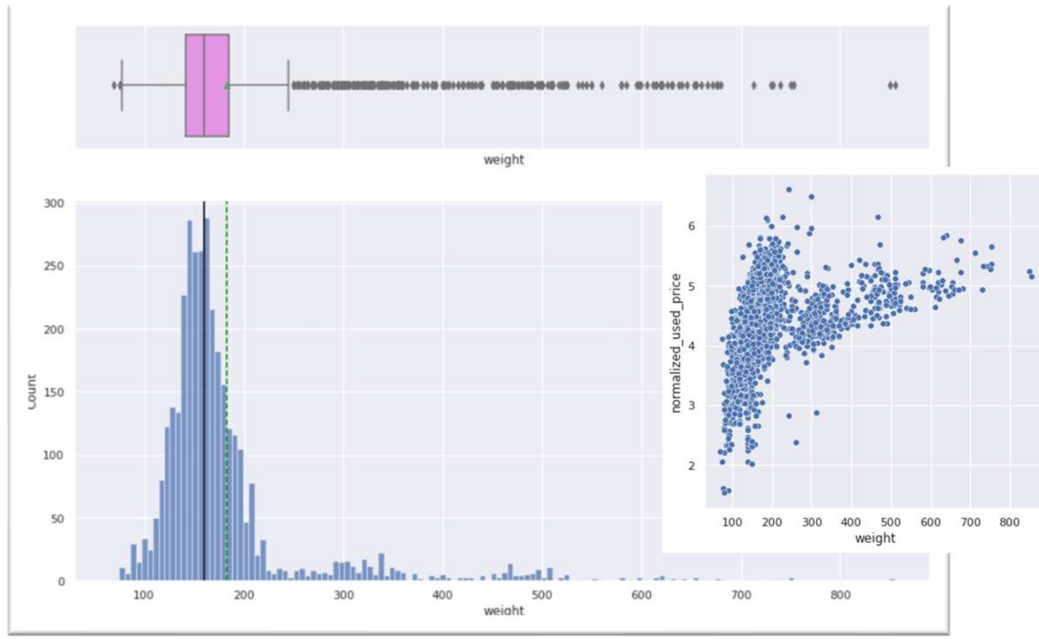
VIF after dropping screen_size

| | feature | VIF |
|---|---|---|
| 0 | const | 202.673906 |
| 1 | main_camera_mp | 2.281835 |
| 2 | selfie_camera_mp | 2.809009 |
| 3 | int_memory | 1.362043 |
| 4 | ram | 2.282350 |
| 5 | battery | 3.842989 |
| 6 | weight | 2.993855 |
| 7 | days_used | 2.648929 |
| 8 | normalized_new_price | 3.077650 |
| 9 | years_since_release | 4.730315 |

# Model Assumptions – Linearity of variables

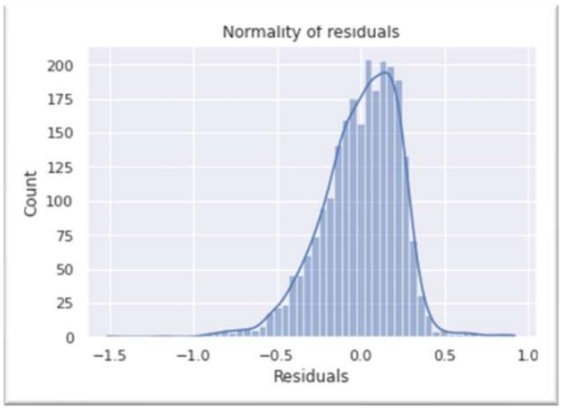
- There should be a **linear relationship between dependent and independent variables**.
- Plotted the fitted values vs residuals. **Saw no pattern**, so the **model is assumed linear.**
- However, **weight** may warrant further investigation due to curved patter exhibited in scatterplot with **used price**.

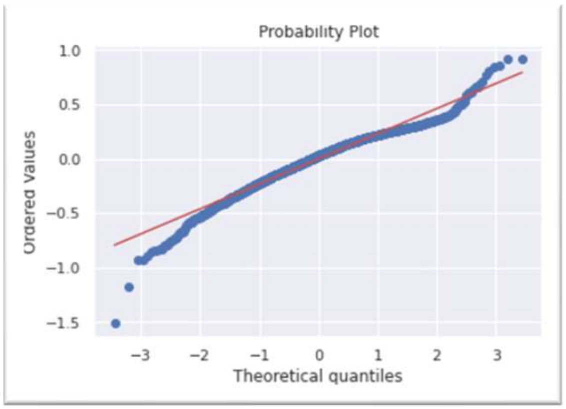


# Model Assumptions – Normality of error terms

- The error terms, **residuals** must be **normally distributed**. Tested for normality by: (1) checking the **distribution of residuals**, (2) by checking the **Q-Q plot of residuals**, and (3) by using the **Shapiro-Wilk test**.
  - (1) The residuals follow a normal distribution with a **slight left- skew**.
  - (2) The Q-Q plot of residuals **generally make a straight-line plot**. **This could be improved.**
  - (3) Shapiro-Wilk test, resulted in **statistic**=0.9676972031593323 **pvalue**=6.995328206686811e-23).
    - P-vale is 0.0000 ( **NOT** greater than 0.05), we can **NOT** say the residuals are normally distributed. The **residuals are NOT normal as per Shapiro-Wilk test**. **This may need further investigation.**



# Model Assumptions – No Heteroscedasticity

- The **residuals** must have **constant variance**. Tested for **homoscedasticity** by using the **goldfeldquandt test**. The Goldfeldquandt test, resulted in F statistic =~1.00875..., p-value = ~ 0.4402. Since the p-value was greater than 0.05, we can say that the residuals are homoscedastic.

Great Learning

# Happy Learning !