

Course Title: Practical Machine Learning.

Lecturer: Dr.Mohammed ANBAR

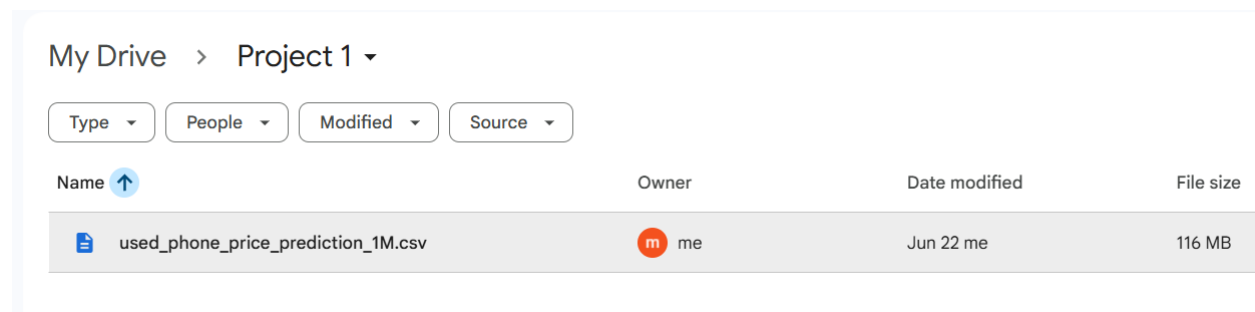
Title: Used Phone Price Prediction Dataset

Name: Maya ALhamedi

Id: 262000650

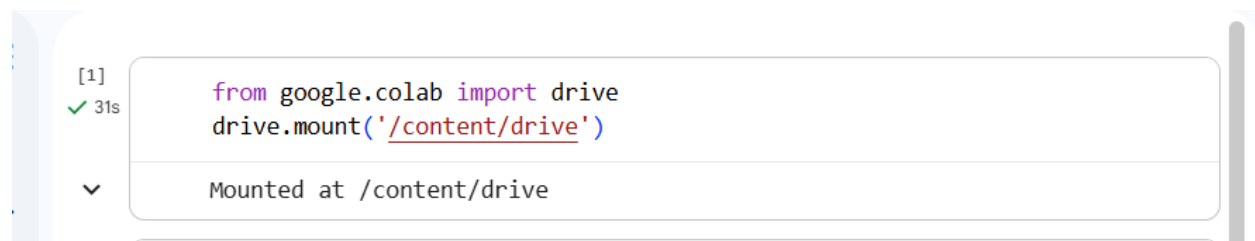
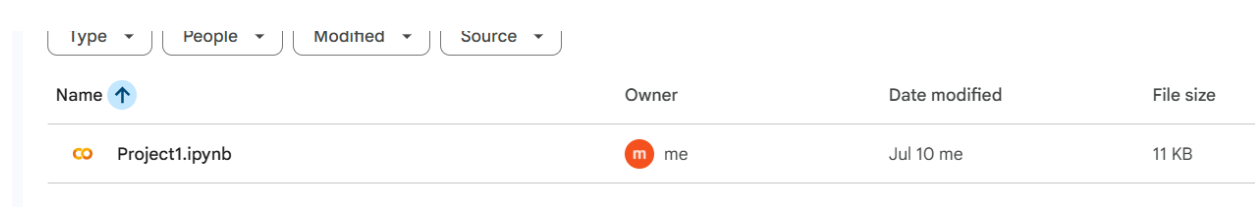
1- Open Google Drive and create a new folder.

Create a new folder in Google Drive to store the dataset and the project notebook.



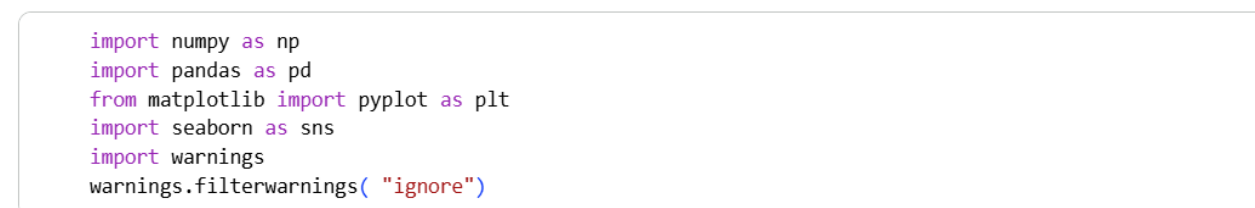
2- After creating the folder, connect Google Colab to Google Drive.

Connect Google Colab with Google Drive to access the dataset.



3 - Import the dataset (file).

Import the necessary Python libraries for data analysis and visualization.



4- Copy the file path.

Load the CSV file into a Pandas DataFrame.

```
df = pd.read_csv( '/content/drive/MyDrive/Project 1 /used_phone_price_prediction_1M.csv')
```

5- Display the first few rows of the dataset.

Display the first few rows of the dataset.

df.head()

	brand	model	release_year	ram_gb	storage_gb	screen_size_inches	battery_capacity	processor_score	camera_score	os_type	...	body_damage	repair_history	water_dam
0	Realme	Realme GT	2025	4	256	6.93	4130	47	100	Android	...	0	0	
1	Samsung	Galaxy S21	2024	16	64	6.29	5558	81	67	Android	...	0	0	
2	Apple	iPhone 11	2022	4	128	5.52	5945	59	67	iOS	...	0	0	
3	Google	Pixel 6	2024	16	512	5.99	5767	84	57	Android	...	0	0	
4	Google	Pixel 7	2021	4	128	6.59	5336	98	44	Android	...	0	0	

5 rows x 28 columns

6- Display. Dataset Information.

Display information about the dataset such as data types and the number of non-null values.

[6]

✓ 1s

df.info()

▼

```
... <class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000000 entries, 0 to 999999
Data columns (total 28 columns):
 #   Column                                  Non-Null Count  Dtype
---  -
 0   brand                                  1000000 non-null object
 1   model                                  1000000 non-null object
 2   release_year                          1000000 non-null int64
 3   ram_gb                                 1000000 non-null int64
 4   storage_gb                            1000000 non-null int64
 5   screen_size_inches                    1000000 non-null float64
 6   battery_capacity                       1000000 non-null int64
 7   processor_score                       1000000 non-null int64
 8   camera_score                          1000000 non-null int64
 9   os_type                                1000000 non-null object
10   has_5g                                 1000000 non-null int64
11   original_price                         1000000 non-null int64
12   purchase_year                         1000000 non-null int64
13   age_months                            1000000 non-null int64
14   usage_hours_per_day                   1000000 non-null float64
15   condition                              1000000 non-null object
16   battery_health                        1000000 non-null int64
17   screen_cracked                        1000000 non-null int64
18   body_damage                           1000000 non-null int64
19   repair_history                        1000000 non-null int64
20   water_damage                          1000000 non-null int64
21   city_tier                             1000000 non-null object
22   seller_type                           1000000 non-null object
23   warranty_remaining_months             1000000 non-null int64
```

[9]

✓ 0s

df.columns

▼

```
... Index(['brand', 'model', 'release_year', 'ram_gb', 'storage_gb',
         'screen_size_inches', 'battery_capacity', 'processor_score',
         'camera_score', 'os_type', 'has_5g', 'original_price', 'purchase_year',
         'age_months', 'usage_hours_per_day', 'condition', 'battery_health',
         'screen_cracked', 'body_damage', 'repair_history', 'water_damage',
         'city_tier', 'seller_type', 'warranty_remaining_months',
         'box_available', 'charger_available', 'market_demand_score',
         'resale_price'],
         dtype='object')
```

7-Check the missing values. Check whether the dataset contains missing values.

```
[10] df.isnull().sum()
```

...	0
brand	0
model	0
release_year	0
ram_gb	0
storage_gb	0
screen_size_inches	0
battery_capacity	0
processor_score	0
camera_score	0
os_type	0
has_5g	0
original_price	0
purchase_year	0
age_months	0

age_months	0
usage_hours_per_day	0
condition	0
battery_health	0
screen_cracked	0
body_damage	0
repair_history	0
water_damage	0
city_tier	0
seller_type	0
warranty_remaining_months	0
box_available	0
charger_available	0
market_demand_score	0
resale_price	0

dtype: int64

8-Handle Missing Values.

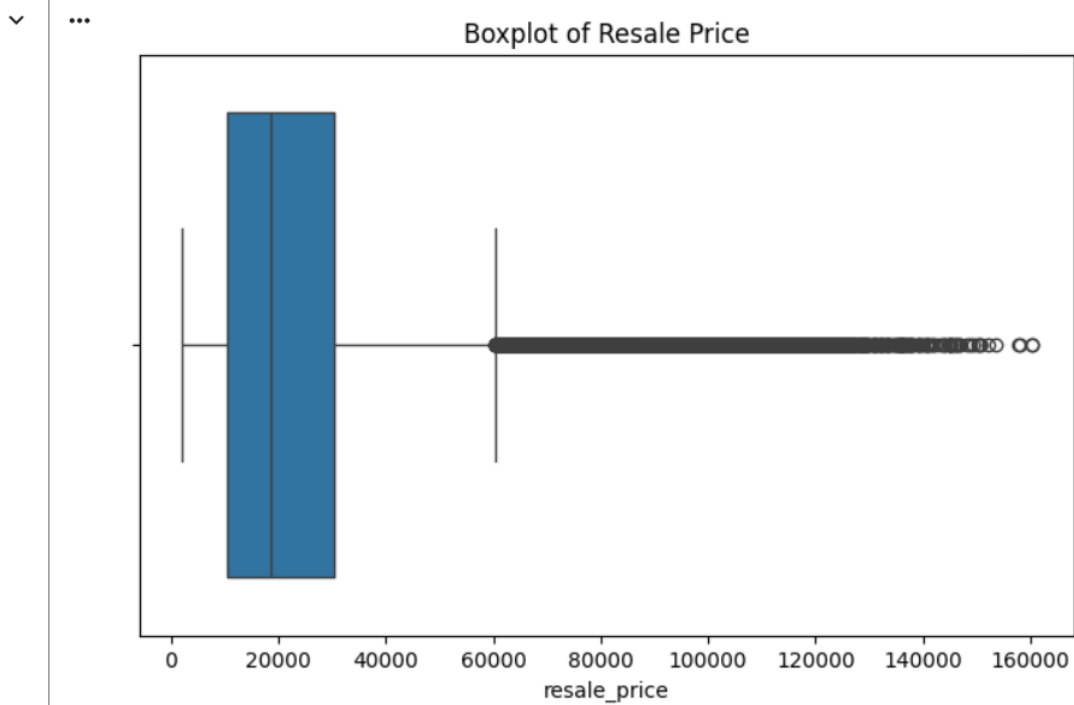
Missing values were replaced with the mean value of each numerical column.

```
[11] ✓ 0s df.fillna(df.mean(numeric_only=True), inplace=True)
```

9- Detect Outliers.

Use a boxplot to detect outliers in the resale price.

```
[12] ✓ 2s plt.figure(figsize=(8,5))  
sns.boxplot(x=df['resale_price'])  
plt.title("Boxplot of Resale Price")  
plt.show()
```



10- Handle Outliers Using The IQR Method.

Outliers were removed using the Interquartile Range (IQR) method.

[13]

✓ 0s



```
Q1 = df['resale_price'].quantile(0.25)
```

```
Q3 = df['resale_price'].quantile(0.75)
```

```
IQR = Q3 - Q1
```

```
lower = Q1 - 1.5 * IQR
```

```
upper = Q3 + 1.5 * IQR
```

```
df = df[(df['resale_price'] >= lower) & (df['resale_price'] <= upper)]
```

11- Data Visualization 1.

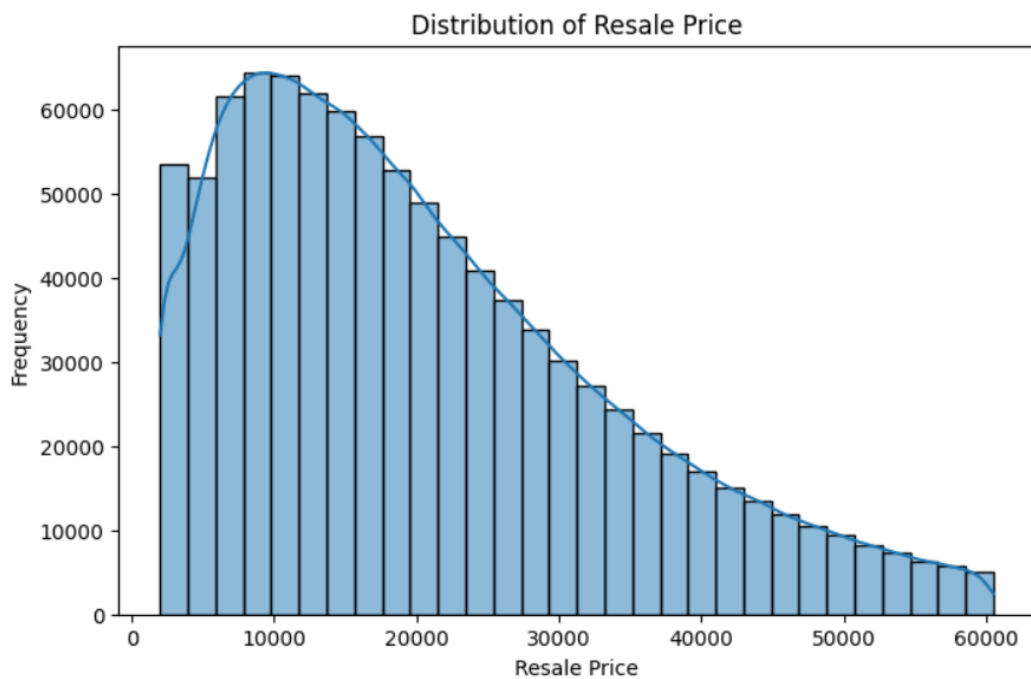
This histogram shows how resale prices are distributed across the dataset.

[14]

✓ 5s



```
plt.figure(figsize=(8,5))
sns.histplot(df['resale_price'], bins=30, kde=True)
plt.title("Distribution of Resale Price")
plt.xlabel("Resale Price")
plt.ylabel("Frequency")
plt.show()
```



Visualization 2: Number Of Phone By Brand.

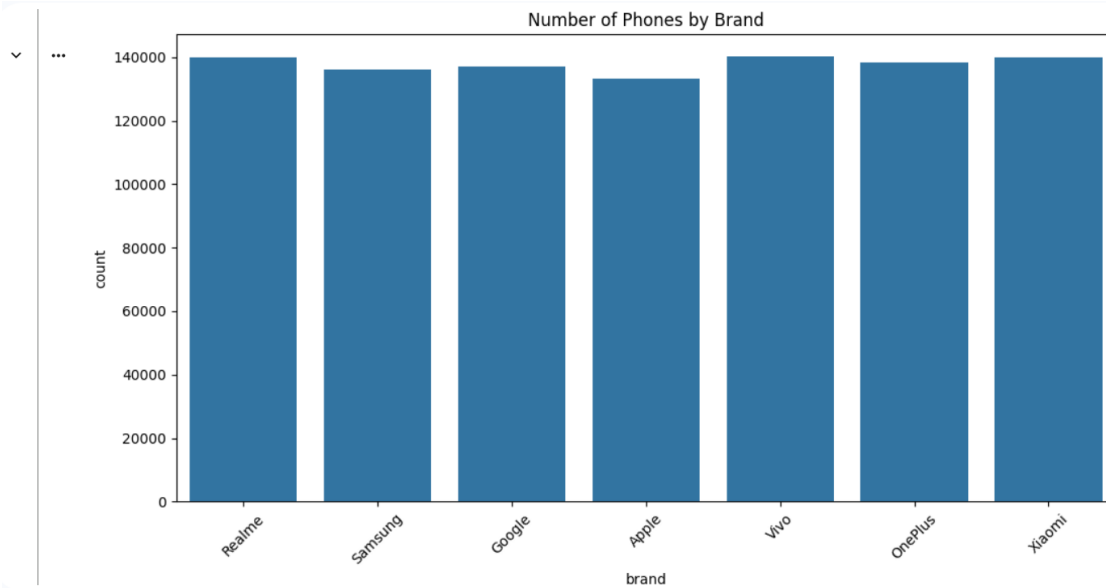
This chart displays the number of phones available for each brand.

[15]

✓ 2s

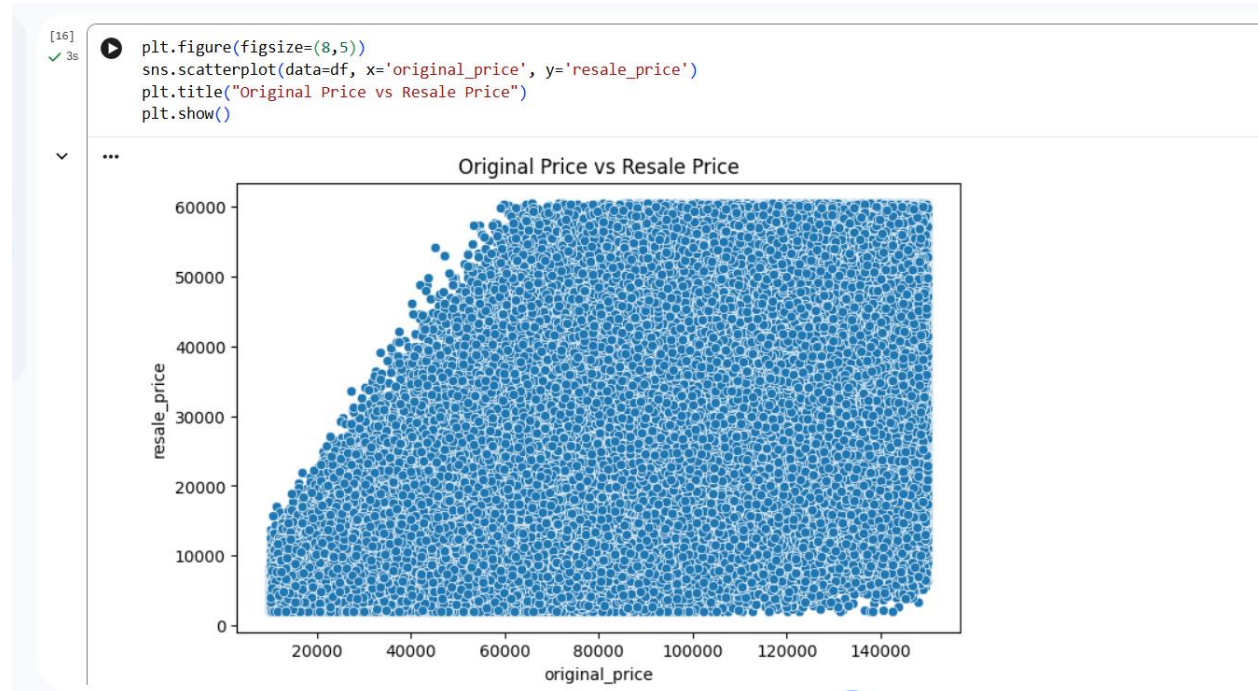


```
plt.figure(figsize=(12,6))
sns.countplot(data=df, x='brand')
plt.xticks(rotation=45)
plt.title("Number of Phones by Brand")
plt.show()
```



Visualization 3: Original Price VS Resale Price.

This scatter plot illustrates the relationship between the original price and the resale price.



Conclusion:

- The dataset was successfully loaded and analyzed.
- Missing values were checked and handled.
- Outliers were detected and removed using the IQR method.
- Three visualizations were created to explore the dataset.
- The results provide insights into resale prices and phone brands.