Project: Feature Extraction and Price Prediction for Mobile Phones using Python

Project By :- Deepak Sharma

Project Description:- In this project, I worked with a dataset that contains detailed information about various mobile phones, including their model, color, memory, RAM, battery capacity, rear camera specifications, front camera specifications, presence of AI lens, mobile height, processor, and most importantly, the Price. My primary goal is to develop a predictive model for mobile phone prices.

Project Tasks

- Data Exploration
- Data Preprocessing
- Feature Extraction
- Model Building
- Model Evaluation
- Feature Importance Analysis
- Dashboard Creation Using Tableau
- Recommendations

Data Exploration

Project started by loading and exploring the dataset to understand its structure, data types, the range of values for each feature etc.

File Edit View Insert Cell Kernel Widgets Help   Trusted   Python 3 (ipykernel)

Run   Markdown

Import Necessary Packages

In [1]:

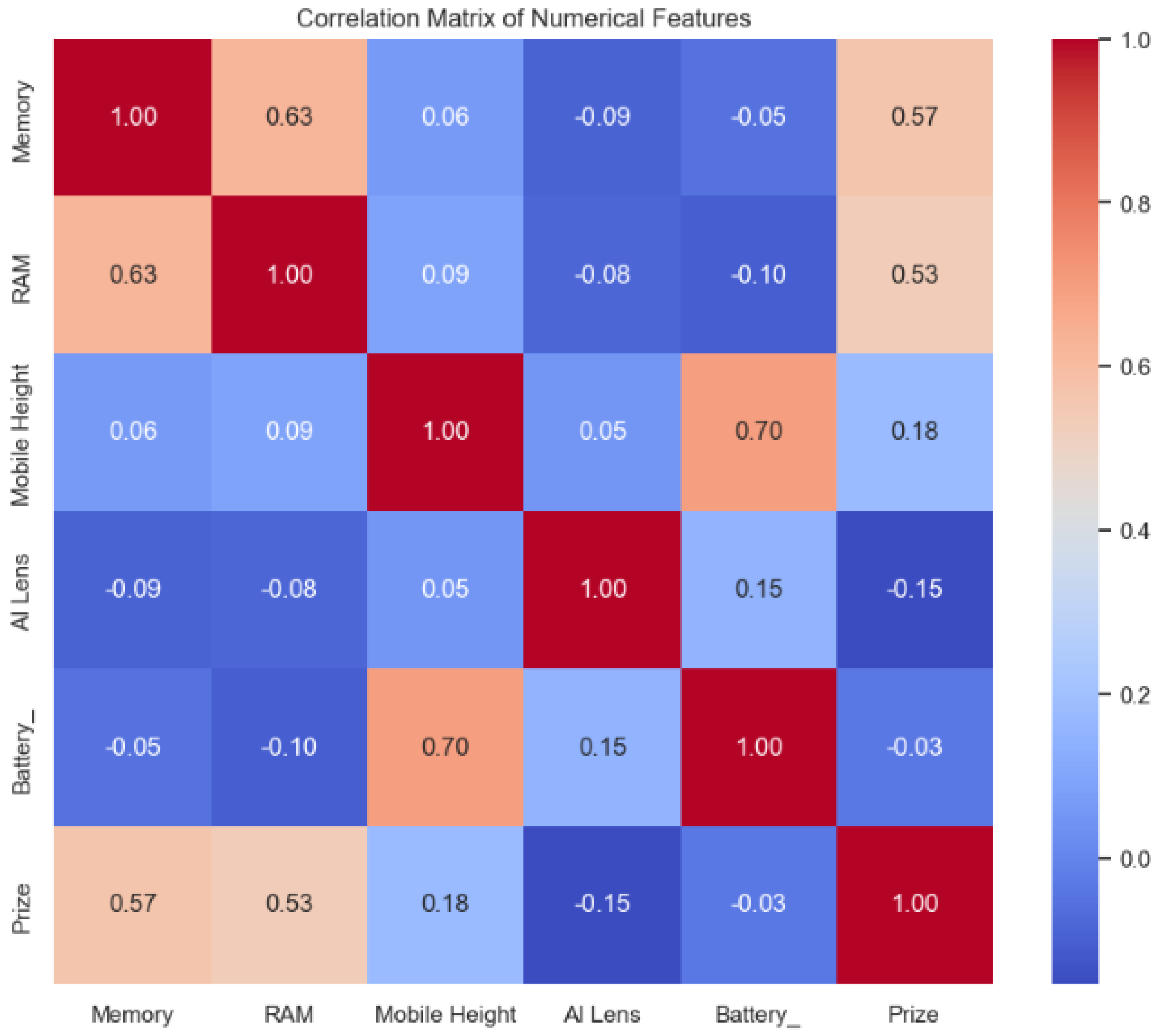
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.model_selection import train_test_split
```

Data Exploration

In [2]:

```
# Load the dataset

df = pd.read_excel("Processed_Flipdata.xlsx")
```

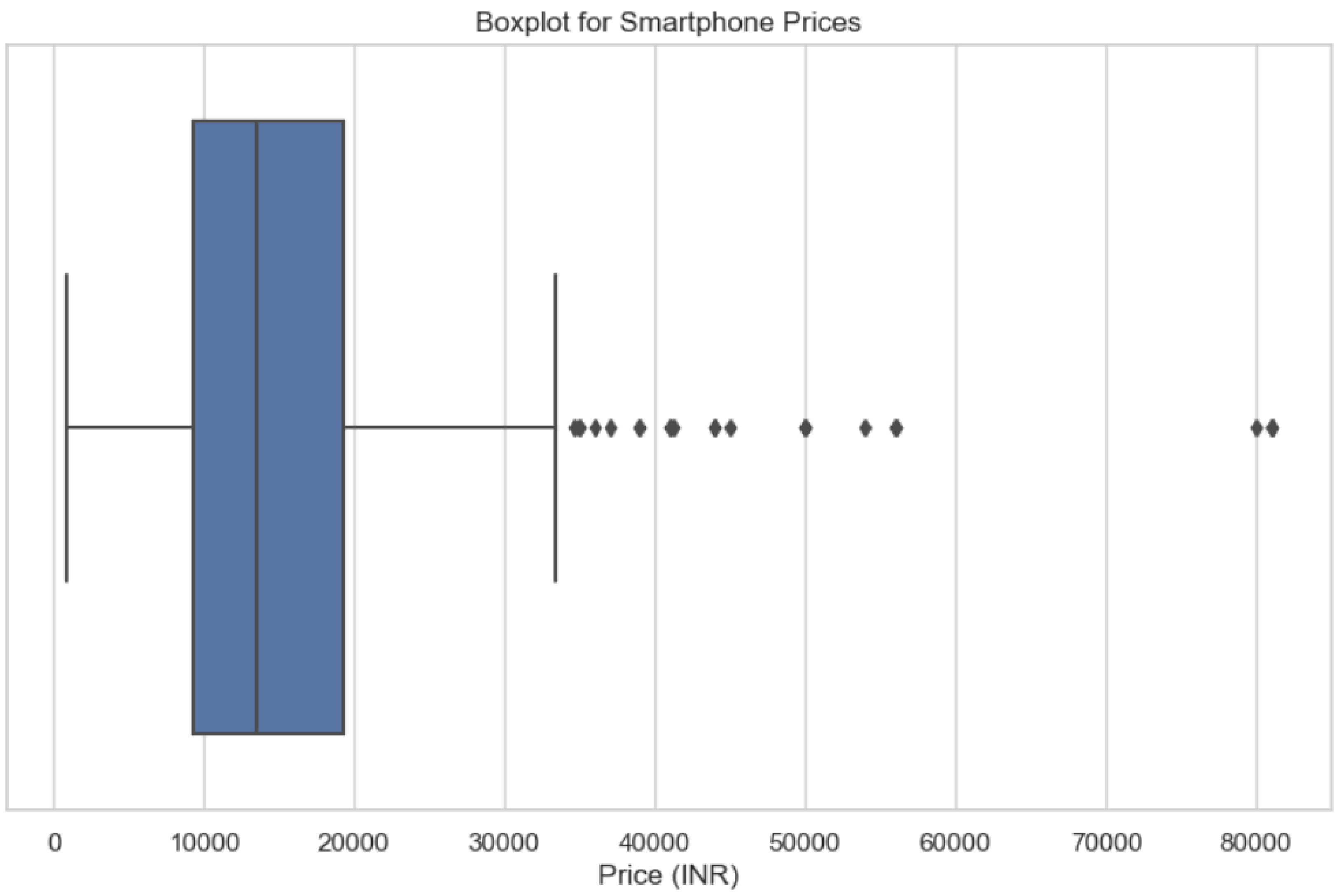
Correlation Matrix



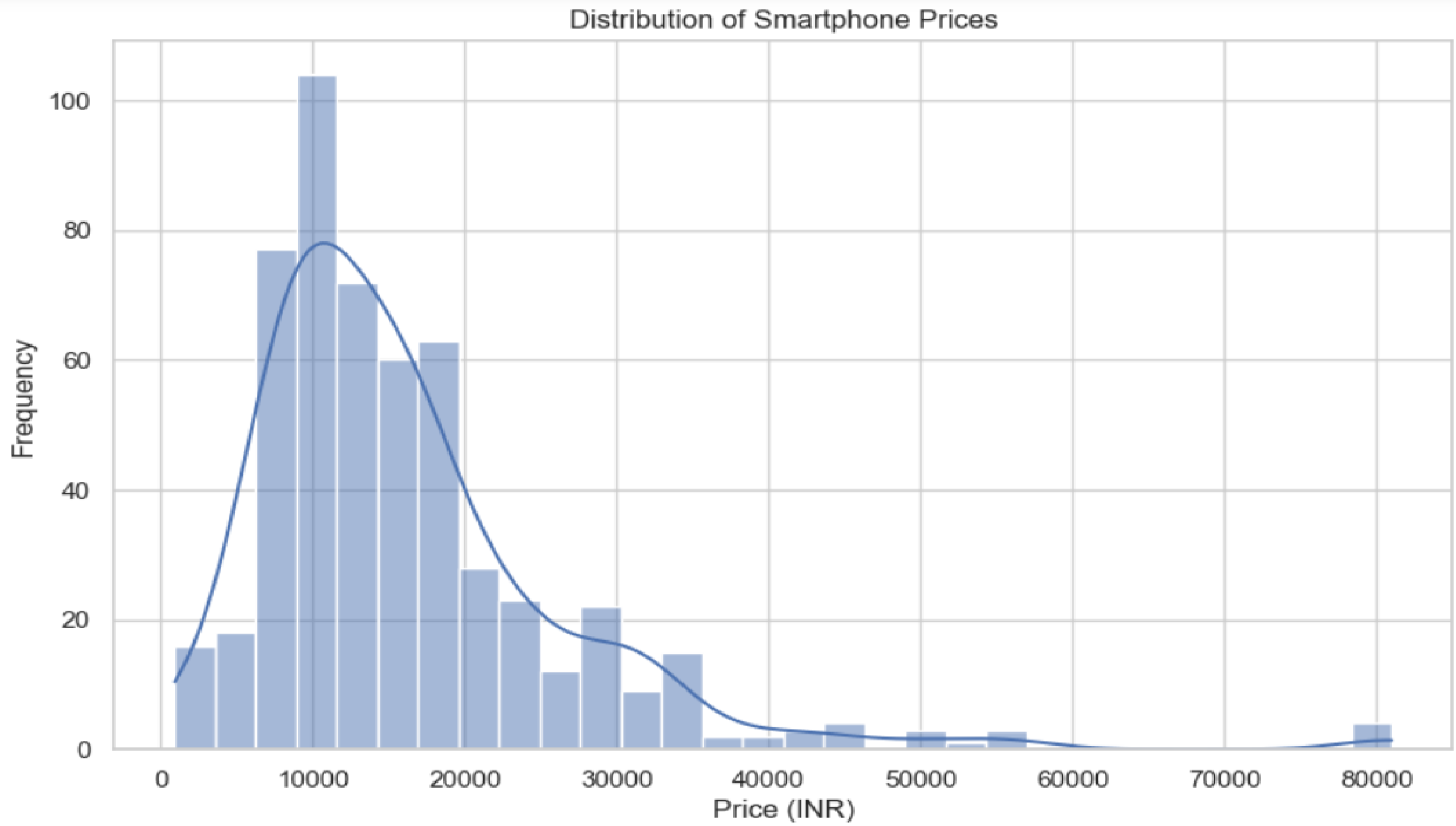
Observations :- The heatmap above represents the correlation matrix among the numerical features of the smartphones in the dataset. Here are some key observations.

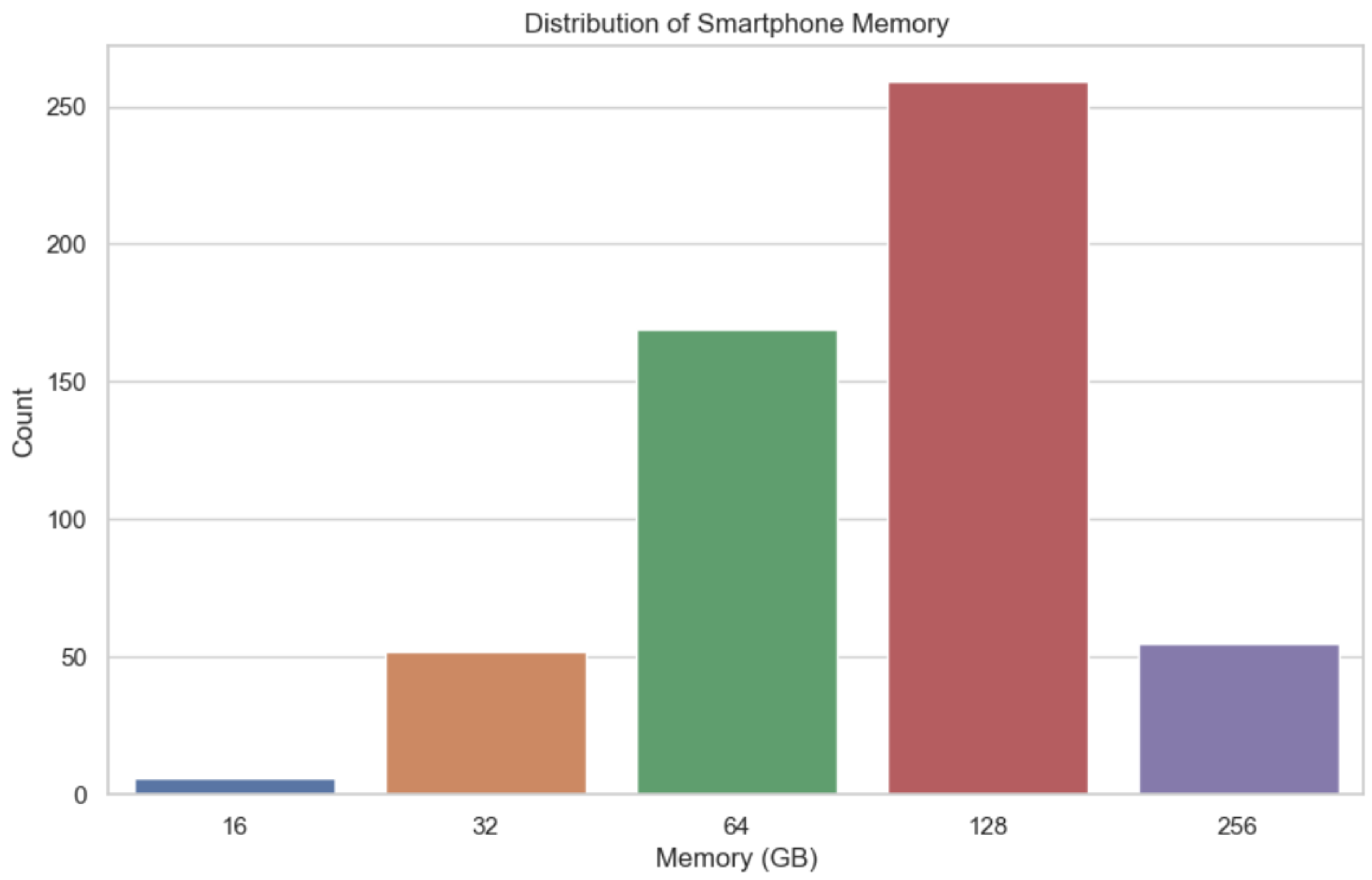
1. **Memory and Prize:** There is a moderate positive correlation between the memory of smartphones and it's prize. This suggests that as the internal storage increases, the prize of the smartphone tends to increase as well.
2. **RAM and Prize:** Similar to memory, There is also a moderate positive correlation between the RAM and prize. This indicates that smartphones with higher RAM are generally more expensive.
3. **Battery capacity and Prize:** The correlation between Battery capacity and Prize is relatively low, implying that the prize of smartphones doesn't increase significantly with larger battery capacity.
4. **Mobile height and other features:** The mobile height shows very low correlation with other features like memory, RAM, battery capacity and prize, indicating that the physical size of the phone is not strongly related to these specifications.
5. **Memory and RAM:** There is a significant positive correlation between memory and RAM. This makes sense as higher-end models often come with both higher RAM and memory.
6. **AI lens Vs Prize :** The correlation is very low, indicating that the presence of an AI lens feature does not significantly affect the prize

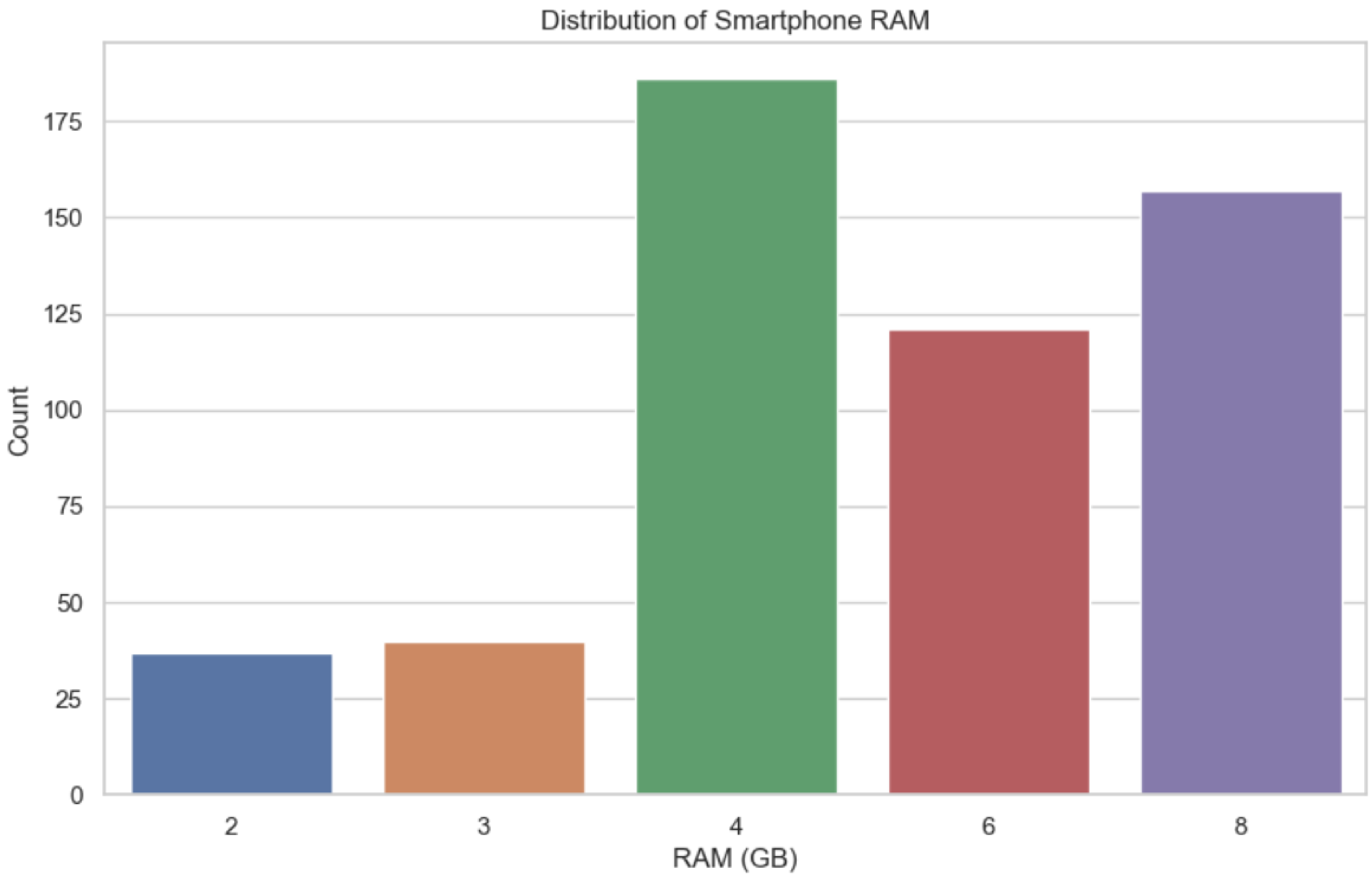
Plotting a boxlplot for the 'Prize' column to check for outliers

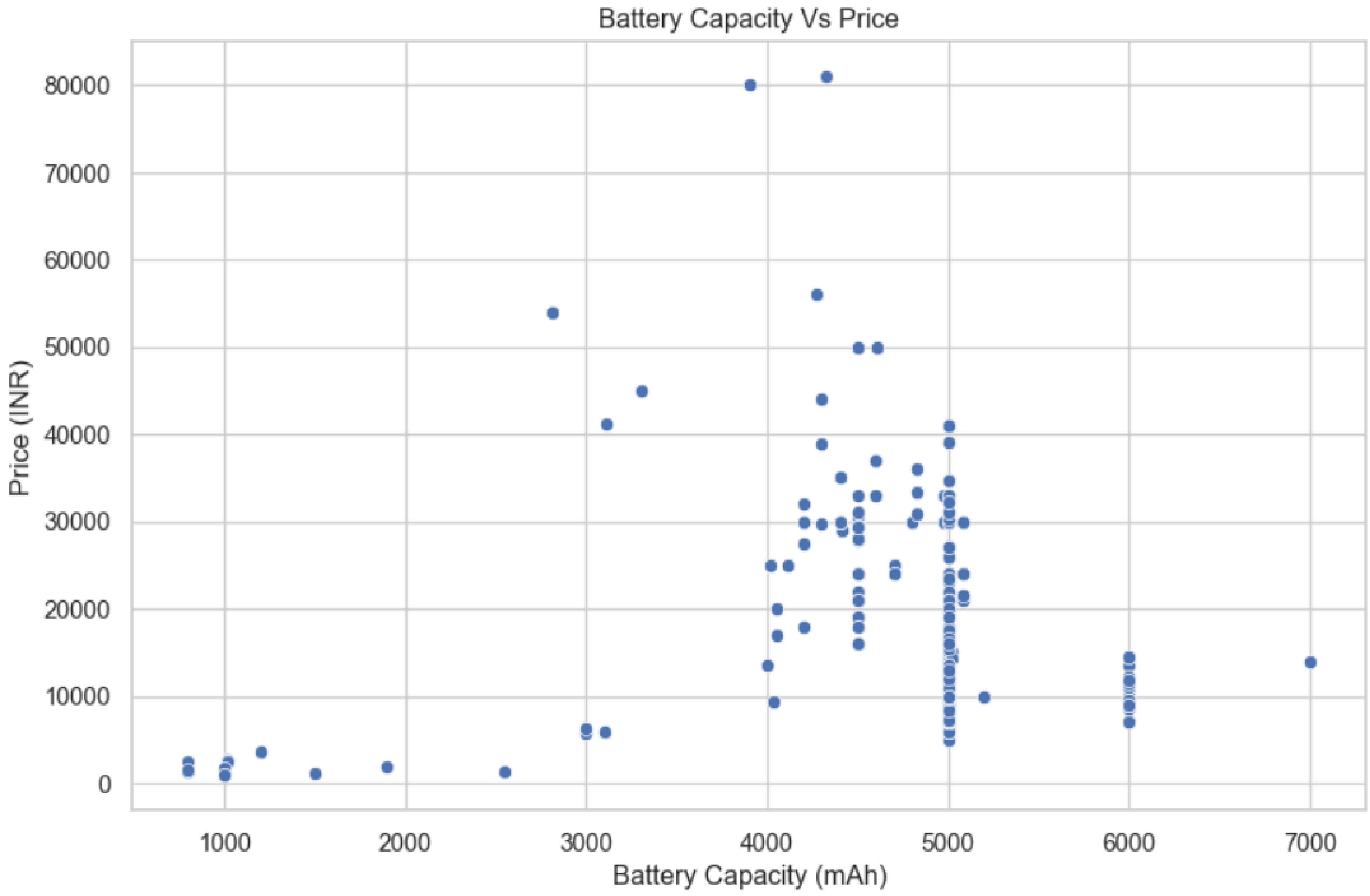


Some More Data Exploration By Visualization




Distribution of Smartphone Memory
250
200
150
100
50
0
Count
16
32
64
128
256
Memory (GB)


Distribution of Smartphone RAM
Count
175
150
125
100
75
50
25
0
2
3
4
6
8
RAM (GB)


Battery Capacity Vs Price
Price (INR)
80000
70000
60000
50000
40000
30000
20000
10000
0
1000
2000
3000
4000
5000
6000
7000
Battery Capacity (mAh)

Observations :- Here are some more data exploration based on visualization.

1. Distribution of Phone Prices : The price distribution shows a wide range of prices, with a concentration of smartphones in the lower to mid price range, there are fewer smartphones in the higher price range, indicating a skew towards more affordable models.
2. Distribution of Phone Memory : The most common memory sizes are 128 GB and 64 GB, with fewer smartphones offering 256 GB, 32 GB and 16 GB. This suggests a preference in the market for smartphones with moderate to high storage capacity.
3. Distribution of Phone RAM : The distribution of RAM shows a significant number of smartphones with 4 GB and 8 GB RAM, followed by 6 GB. Fewer models have 2GB and 3GB RAM.
4. Battery Capacity Vs Price : The Scatter plot of Battery Capacity Vs Price doesn't show a strong linear relationship. This suggests that battery capacity is not a primary factor driving the price of the smartphones. High capacity batteries are available in both lower and higher priced models.

Data Preprocessing

Handled missing values, outliers and inconsistencies in the dataset. Converted categorical variables into a suitable numerical format, using One-hot encoding, Ordinal encoding and Label Encoding.

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Markdown

Data Preprocessing

In [21]:
```
# Drop unnecessary columns
# The "Unnamed: 0" column seems redundant as it just replicates the index
# We can consider dropping it for analysis

df2 = df.drop("Unnamed: 0", axis = 1)
```

In [22]:
```
df2
```

Out[22]:

	Model	Colour	Memory	RAM	Battery_	Rear Camera	Front Camera	AI Lens	Mobile Height	Processor_	Prize
0	Infinix SMART 7	Night Black	64	4	6000	13MP	5MP	1	16.76	Unisoc Spreadtrum SC9863A1	7299
1	Infinix SMART 7	Azure Blue	64	4	6000	13MP	5MP	1	16.76	Unisoc Spreadtrum SC9863A1	7299
2	MOTOROLA G32	Mineral Gray	128	8	5000	50MP	16MP	0	16.64	Qualcomm Snapdragon 680	11999
3	POCO C50	Royal Blue	32	2	5000	8MP	5MP	0	16.56	Mediatek Helio A22	5649
4	Infinix HOT 30i	Marigold	128	8	5000	50MP	5MP	1	16.76	G37	8999
...	...	...	...	...	...	...	...	...	...	...	...
536	SAMSUNG Galaxy S23 5G	Cream	256	8	3900	50MP	12MP	0	15.49	Qualcomm Snapdragon 8 Gen 2	79999
537	LAVA Z21	Cyan	32	2	3100	5MP	2MP	0	12.70	Octa Core	5998

Feature Extraction

Performed feature extraction to identify the most relevant features that strongly affect the price of mobile phones with Principal Component Analysis (PCA).

Principal Component Analysis (PCA)

In [35]:
```
from sklearn.decomposition import PCA

# Standardization of the data
scaler = StandardScaler()
scaled_data = scaler.fit_transform(df2)

# Applying PCA
pca = PCA(n_components=2)  # Using 2 components for visualization purposes
principal_components = pca.fit_transform(scaled_data)

# Creating a DataFrame with the principal components
pca_df = pd.DataFrame(data=principal_components, columns = ['Principal Component 1', 'Principal Component 2'])

pca_df.head()
```

Out[35]:

	Principal Component 1	Principal Component 2
0	-2.596461	-0.763273
1	-2.702775	-0.666949
2	0.867109	-0.145954
3	-2.444489	-0.155270
4	-0.707592	-1.539119

Observations :- After standardizing the data, We applied Principal Component Analysis (PCA) and reduced the dataset to two principal components. These components are linear combinations of our original features and are designed to capture as much of the variance in the data as possible.

Model Building and Evaluation

The dataset has been splitted into training and testing sets and Developed a machine learning model for price prediction using various algorithms such as Linear Regression, Decision tree, Random forest and Gradient Boosting and evaluated the performance metrics such as mean absolute error, root mean squared error, R2 score.

Model Building

In [37]:
```
# Import necessary packages

from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
from math import sqrt

# Splitting the dataset into features (X) and the target variable (y)

X = pca_df
y = df2['Prize']

# Splitting into training and testing sets

X_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialize the Linear Regression model

linear_model = LinearRegression()

# Train the model

linear_model.fit(X_train, y_train)

# Predict on the test set

y_pred = linear_model.predict(x_test)

# Calculate performance metrics

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
```

In [38]:
```
mse, r2
```

Out[38]:
```
(9332730.611560669, 0.8034560362164304)
```

The linear regression model is trained and evaluated on the test set, The performance metrices are as :-

1. Mean Squared Error (MSE) : Approximately 9332730.62
2. R2 score : Approximately 0.81

MSE :- This value indicates the average squared difference between the actual and the predicted value.

R2 :- This score represents the proportion of variance in the dependent variable that is predicatable from the independent variable. Here, an R2 of 0.81 means that about 81% of the variance in the 'Prize' can be predicted from the PCA components. This is a relatively strong score, suggesting that the model has good predicting power.

Model Evaluation

In [39]:
```
# Evaluating the performance metrics

mse = mean_squared_error(y_test, y_pred)
rmse = sqrt(mse)
mae = mean_absolute_error(y_test, y_pred)

mae, rmse
```

Out[39]:
```
(2111.5288275216917, 3054.9518182060856)
```

Observations :-

MAE : On average, the model's predictions are about 2111.53 units, away from the actual phone price.

EMSE : This is a more sensitive measure to larger errors, an RSME of 3054.95 suggests that the standard deviation of the prediction error is around this value.

Using Random Forest

In [40]:
```
# Import necessary packages first

from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
from sklearn.model_selection import cross_val_score

# Creating a pipeline that first scales the features and then apply Random Forest

pipeline = make_pipeline(StandardScaler(), RandomForestRegressor(random_state=42))

# Training the pipeline on the training data

pipeline.fit(X_train, y_train)

# Predicting on the test set

y_pred_rf = pipeline.predict(x_test)

# Calculate performance metrics for Random Forest

mse_rf = mean_squared_error(y_test, y_pred_rf)
rmse_rf = sqrt(mse_rf)
r2_rf = r2_score(y_test, y_pred_rf)
```

Comparing metrics of different models

- The linear regression model is trained and evaluated on the test set, The performance metrices are as :-

1. Mean Squared Error (MSE) : Approximately 9332730.62
2. R2 score : Approximately 0.81

- MSE :- This value indicates the average squared difference between the actual and the predicted value.
- R2 :- This score represents the proportion of variance in the dependent variable that is predicatable from the independent variable. Here, an R2 of 0.81 means that about 81% of the variance in the 'Prize' can be predicted from the PCA components. This is a relatively strong score, suggesting that the model has good predicting power.
- The Random forest model is trained and evaluated on the test set, The performance metrices are as :-

1. Mean Squared Error (MSE) : Approximately 6635072.091
2. R2 score : Approximately 0.86

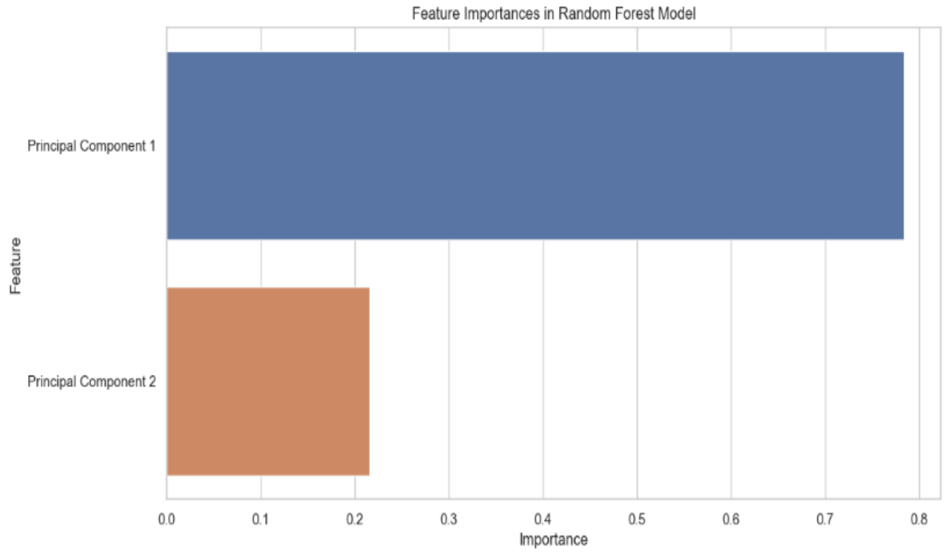
Observations :- Comparing these metrics to those from the Linear Regression Model, We notice improvements.

Feature Importance Analysis

Analyzed the feature importance obtained from the model to confirm the significance of the features identified during the feature extraction phase.

Feature Importance Analysis

In [46]:
```
# Extracting feature importance from the Random Forest model

feature_importances = pipeline.named_steps['randomforestregressor'].feature_importances_

# Creating a DataFrame for better visualization

features_df = pd.DataFrame({
    'Feature': X.columns,
    'Importance': feature_importances
}).sort_values(by="Importance", ascending=False)

# Plotting the feature importance

plt.figure(figsize=(12,6))
sns.barplot(x='Importance', y="Feature", data=features_df)
plt.title('Feature Importances in Random Forest Model')
plt.xlabel('Importance')
plt.ylabel('Feature')
plt.show()

features_df
```



Out[46]:

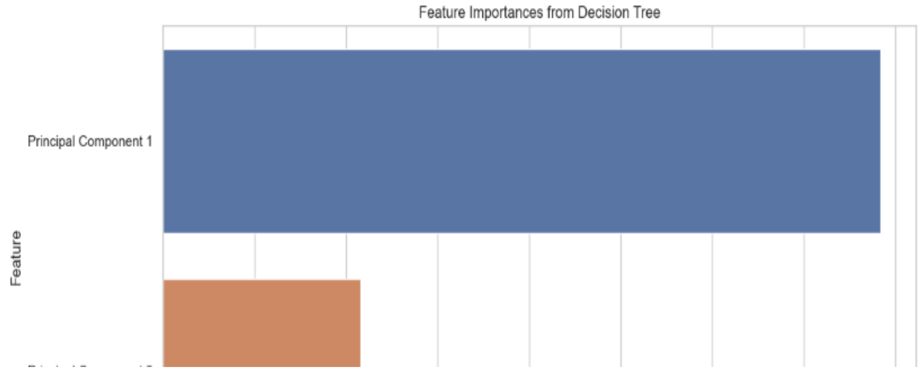
	Feature	Importance
0	Principal Component 1	0.783945
1	Principal Component 2	0.216055

Observations :-

PC1(Principal Component1) : This component is the most important feature, with an importance score of approximately 0.784. It indicates that the first principal component captures the most significant variance and patterns in the data.

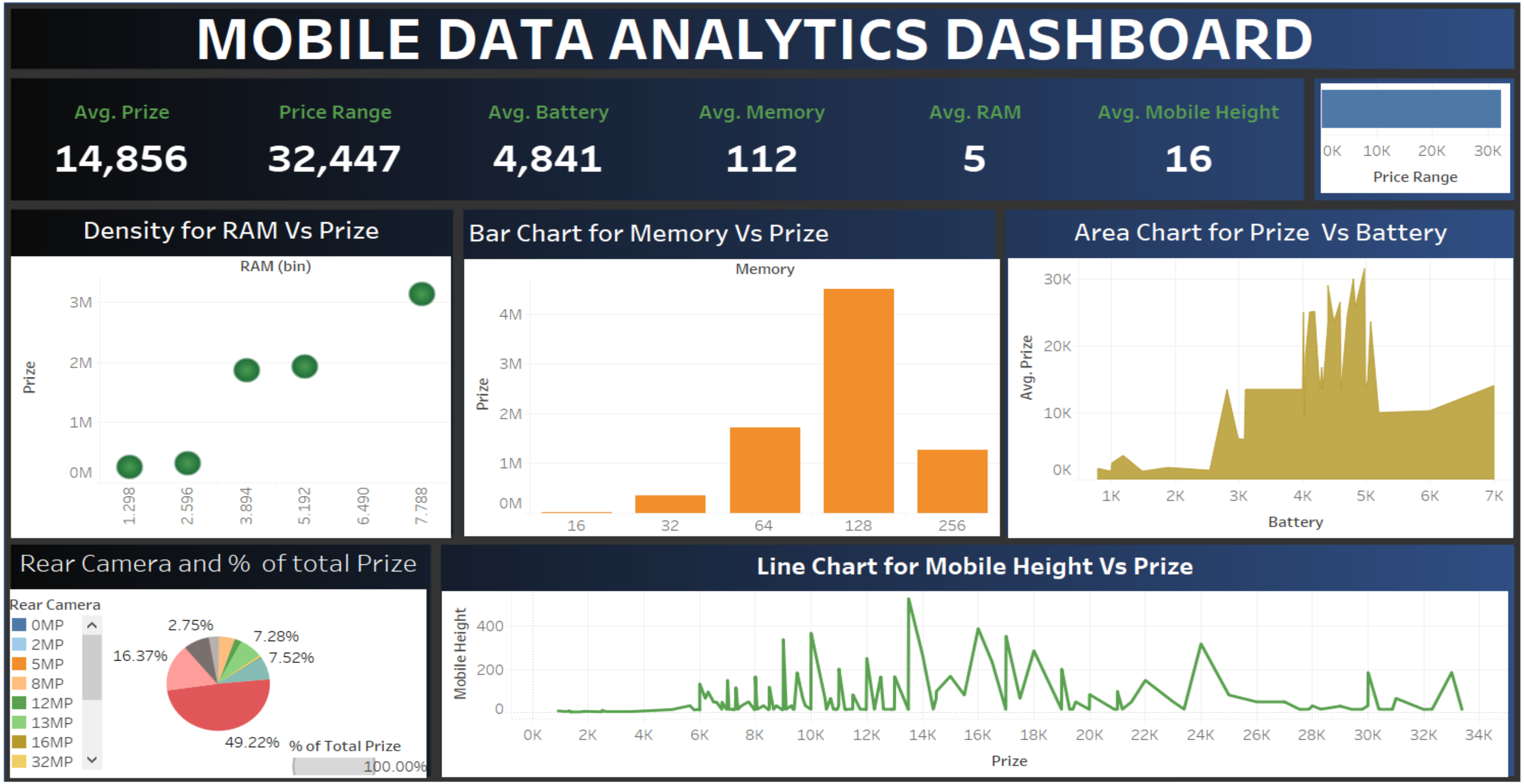
PC2(Principal Component2) : The second principal component is also valuable but less so than the first, with an importance score of about 0.216.

In [47]:
```
# Feature importance from the Decision Tree

dt_feature_importance = dt_model.feature_importances_

# Creating a DataFrame for better visualization

dt_features_df = pd.DataFrame({
    'Feature': X.columns,
    'Importance': dt_feature_importance
}).sort_values(by="Importance", ascending=False)

# Plotting the feature importance

plt.figure(figsize=(12,6))
sns.barplot(x='Importance', y="Feature", data=features_df)
plt.title('Feature Importances from Decision Tree')
plt.xlabel('Importance')
plt.ylabel('Feature')
plt.show()

dt_features_df
```



Dashboard Creation Using Tableau

Created a Dynamic Dashboard including multiple sheets with a minimum of 6-7 important data visualization charts to showcase the project requirement.



Python Libraries Used

Data science libraries such as NumPy, Pandas, Matplotlib, Seaborn and Machine learning libraries such as Scikit-learn (for Python) for building and evaluating the predictive model in the Jupyter Notebook Environment.

Project Conclusion

- The model's reliance on PC1 more than PC2 aligns with the nature of PCA, where the first few components are designed to capture the most variance. Both the components are used (with significant weights) indicates that the feature extraction phase was successful in identifying meaningful patterns in the data. The dominance of PC1 suggests that most information relevant for predicting the mobile phone prices is encapsulated in this single component. However, with an average importance of approximately 0.21 (21 %), PC2 also plays a significant role in predictions.
- Memory Size (RAM and Storage) : High positive loading in PC1. It means that Phone with more Memory and Storage tends to be priced higher and are important for performance-focused consumers.

Recommendations

- Prioritize increasing Memory in higher-end models and highlight this in marketing to target performance-sensitive segments.
- Recognize that different features will appeal to different segments. For instance, gamers might value processor speed and RAM, while travellers might look for battery life and durability.
- Regularly assess how competitors are positioning their products concerning these key features. This can help in understanding market trends and customer expectations.
- Engage with your customer base to get direct feedback on which feature they value most and how they perceive your offering.