



INNOVATION. AUTOMATION. ANALYTICS

# AC Price Analysis using Flipkart Data

Data-Driven Insights on Air Conditioner Pricing  
Using Python



# About us

- We are a team of B.Tech graduates in Electronics and Communication Engineering (ECE) and aspiring data analysts passionate about transforming raw data into meaningful insights.
- This project — “AC Price Analysis using Web Scraping and Pandas” — was developed to study market TRENDS of air conditioners on Flipkart.
- Our goal is to **analyze pricing patterns, brand comparisons**, and seasonal trends using data - driven techniques in Python.
- [Connect with me on LinkedIn Surya Namburi](https://www.linkedin.com/in/suryaprakashnamburi/)
- [Connect with me on LinkedIn Mani K](https://www.linkedin.com/in/mani-k-79b359345/)

# Introduction

- The air conditioner (AC) market is rapidly growing with diverse brands, models, and pricing strategies.
- Understanding pricing patterns helps identify market trends and consumer preferences.
- This project, “**AC Price Analysis using Web Scraping and Pandas,**” focuses on analyzing AC data scraped from **Flipkart** to uncover insights about pricing, brand comparisons, and seasonal variations.
- The analysis is performed using **Python** with tools like **BeautifulSoup** and **Pandas** for data extraction, cleaning, and visualization.

# Business Problem

- **Consumers face difficulty** in choosing the right AC due to varying **prices, brands, and features**.
- **Retailers and marketers** need insights into pricing trends to **remain competitive** in the market.
- The **AC market is highly dynamic**, with seasonal fluctuations affecting prices and consumer demand.

# Objective

- To **scrape real-time AC data** from **Flipkart** using Python-based web scraping tools.
- To **analyze pricing trends** across different **brands, capacities, and features**  
(e.g., star rating, inverter type).
- To **visualize data insights** through graphs and charts using **Pandas** and **Matplotlib/Seaborn**.
- To derive **data-driven insights** that help understand market behavior and consumer choices.

# Tools Used

- **1. Programming & Data Analysis:**
  - **Python** – Main programming language for scraping and analysis
  - **Pandas** – Data manipulation and analysis
  - **NumPy** – Numerical computations
- **2. Web Scraping:**
  - **BeautifulSoup** – Parsing HTML and extracting data
  - **Requests** – Sending HTTP requests to web pages
  - **re (Regex)** – Pattern matching for extracting features
- **3. Data Visualization:**
  - **Matplotlib** – Plotting graphs and charts
  - **Seaborn** – Enhanced statistical visualizations

# Data Collection (Web Scrapping)

- **Source:** Flipkart website – AC listings including price, brand, model, capacity, star rating, and inverter type.
- **Tools & Libraries:**
  - **Python**
  - **BeautifulSoup** – for parsing HTML pages
  - **Requests** – for sending HTTP requests
  - **Pandas** – for storing and managing the dataset
- **Process:**
  - Send HTTP requests to Flipkart AC pages.
  - Parse HTML to extract product details.
  - Store extracted data in a **structured CSV/Excel dataset**.
- **Challenges:**
  - Dynamic website content & pagination
  - Handling missing or inconsistent data
  - Avoiding IP blocks during scraping

# Web-Scrapping Code :

```
price = []
brand = []
model = []
ton = []
star = []
inv = []
units = []
room_size = []
ai = []

for i in range(1,77):
    url = ("https://www.flipkart.com/search?q=air+conditioner&otracker=
    "search&otracker1=search&marketplace=FLIPKART&as-show=on&as=off&page="+str(i))
    print(url)

    a = soup.find_all("div",class_="Nx9bqj _4b5DiR")
    b = soup.find_all("div",class_="KzDlHZ")
    c = soup.find_all("li",class_="J+igdf")

    # Price
    for i in a:
        price.append(i.text)

    #Brand
    for i in b:
        brand.append(re.findall(r"\b^\w+",i.text)[0])

    #model
    for i in b:
        q = re.findall(r"[20]{1}[0-9]{3}",i.text)
        if len(q)>0:
            model.append(q[0])
        else:
            model.append(np.nan)

    #ton
    for i in b:
        ton.append(re.findall(r"(\d+(?:\.\d+)?)\s*Ton",i.text,flags=re.I)[0])

    #star
    for i in b:
        star.append(re.findall(r"(\d+)\s\Star",i.text)[0])
```

```
#Inverter
for i in b:
    a = re.search(r"Split Inverter",i.text,flags=re.I)
    if a:
        inv.append("Yes")
    else:
        inv.append("No")

#units
for i in c:
    x = re.findall(r"(\d+(?:\.\d+)?)\s*(?:Units|kWh)", i.text, flags=re.I)
    if len(x)>0:
        units.append(x[0])

#Room_Size
for i in c:
    x = re.findall(r"Room Size:(.*)",i.text)
    if len(x)>0:
        room_size.append(x[0])

# AI
for i in b:
    a = re.search("AI",i.text)
    if a:
        ai.append("Yes")
    else:
        ai.append("No")
```



# Dataset Overview

- **Number of Records:** ~[1824 \* 10]
- **Key Features / Columns:**
  - **Brand** – Manufacturer of the AC
  - **Model** – AC model name/number
  - **Capacity** – Cooling capacity in tons
  - **Star Rating** – Energy efficiency (1★–5★)
  - **Inverter** – Yes/No
  - **Units** – Power Usage per annually
  - **Price** – Listed price in INR

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 1824 entries, 0 to 1823  
Data columns (total 10 columns):  
#   Column      Non-Null Count  Dtype  
---  -  
0   Brand       1824 non-null   object  
1   Model       1824 non-null   int64  
2   Capacity    1824 non-null   float64  
3   Star        1824 non-null   int64  
4   Inverter    1824 non-null   object  
5   Units       1824 non-null   float64  
6   AI          1824 non-null   object  
7   Size        1824 non-null   object  
8   Price_tag   1824 non-null   object  
9   Price       1824 non-null   int64  
dtypes: float64(2), int64(3), object(5)  
memory usage: 142.6+ KB
```

# Sample Dataset

	Brand	Model	Capacity	Star	Inverter	Units	AI	Size	Price_tag	Price
0	Voltas	2024	1.00	5	Yes	511.13	No	90 sqft or Below	High price	30620
1	MarQ	2025	0.75	3	Yes	553.16	No	90 sqft or Below	Low price	19490
2	Midea	2025	1.00	3	Yes	685.62	Yes	90 sqft or Below	High price	25990
3	Lloyd	2025	1.50	3	Yes	941.76	No	111 - 150 sqft	High price	29490
4	Samsung	2025	1.50	5	Yes	751.24	Yes	111 - 150 sqft	High price	40990
5	Blue	2025	1.50	5	Yes	783.33	No	111 - 150 sqft	High price	39990
6	Panasonic	2025	1.50	3	Yes	977.16	No	111 - 150 sqft	High price	33490
7	Daikin	2024	1.50	3	Yes	966.47	No	111 - 150 sqft	High price	34490
8	Godrej	2025	1.50	3	Yes	951.91	No	111 - 150 sqft	High price	27990
9	LG	2025	1.50	5	No	744.75	Yes	111 - 150 sqft	High price	41490
10	realme	2025	1.50	5	Yes	781.88	No	111 - 150 sqft	High price	28990
11	IFB	2025	2.00	3	Yes	1252.53	Yes	151 - 200 sqft	High price	40590

# Data Cleaning & Preprocessing

## 1. Handling Missing Values:

- Replaced missing **model names** or **features** with NaN or default values.
- Removed rows with **critical missing data** (e.g., price).

## 2. Data Type Conversion:

- Converted **Price** and **Capacity** columns to numeric types for analysis.
- Converted **Star Rating** to integer.

## 3. Removing Duplicates:

- Checked for and removed **duplicate entries** to ensure data quality.

#### 4. Feature Standardization:

- Standardized **brand names** (e.g., “LG” vs “Lg”).
- Standardized **room sizes** and **inverter values** to consistent formats.

#### 5. Final Dataset:

- Clean, structured dataset ready for **analysis and visualization**.
- Columns: Brand, Model, Price, Capacity, Star\_Rating, Inverter, Units, Room\_Size, AI

# Exploratory Data Analysis (EDA)

## 1. Statistical Summary:

- Calculated **mean, median, and mode** of AC prices.
- Provides an overview of the **central tendency** and **spread** of the data.

### Numerical Data

```
df.describe()
```

	Capacity	Star	Units	Price
count	1824.000000	1824.000000	1824.000000	1824.000000
mean	1.427083	3.833333	831.204167	32599.583333
std	0.318606	0.986284	203.125218	6236.018960
min	0.750000	3.000000	511.130000	19490.000000
25%	1.375000	3.000000	712.185000	28990.000000
50%	1.500000	3.000000	781.880000	32490.000000
75%	1.500000	5.000000	964.692500	36990.000000
max	2.000000	5.000000	1252.530000	42390.000000

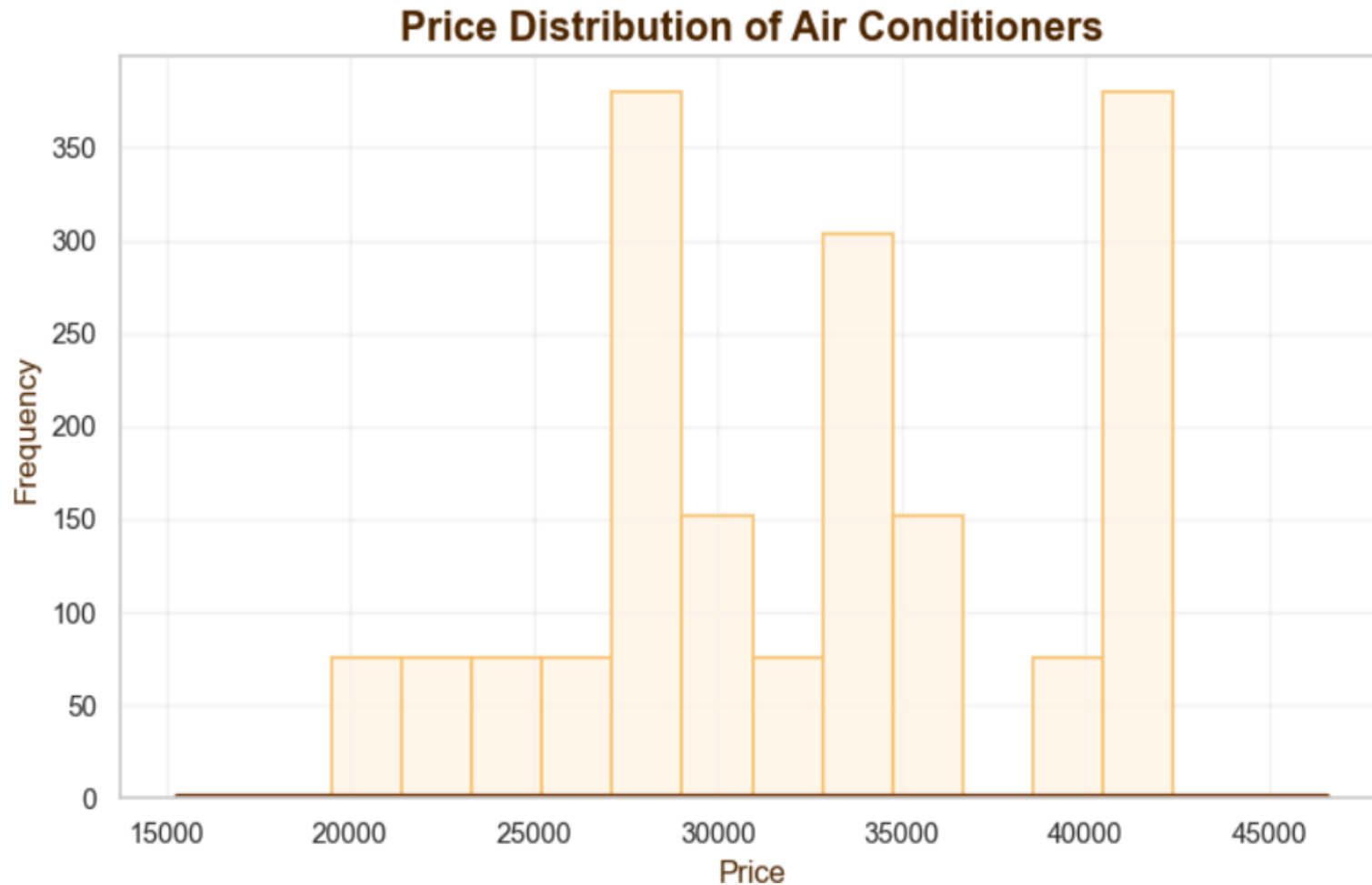
### Categorical Data

```
df.describe(include="O")
```

	Brand	Model	Inverter	AI	Size	Price_tag
count	1824	1824	1824	1824	1824	1824
unique	12	2	2	2	3	2
top	Voltas	2025	Yes	No	111 - 150 sqft	High price
freq	304	1444	1748	1216	1140	1748

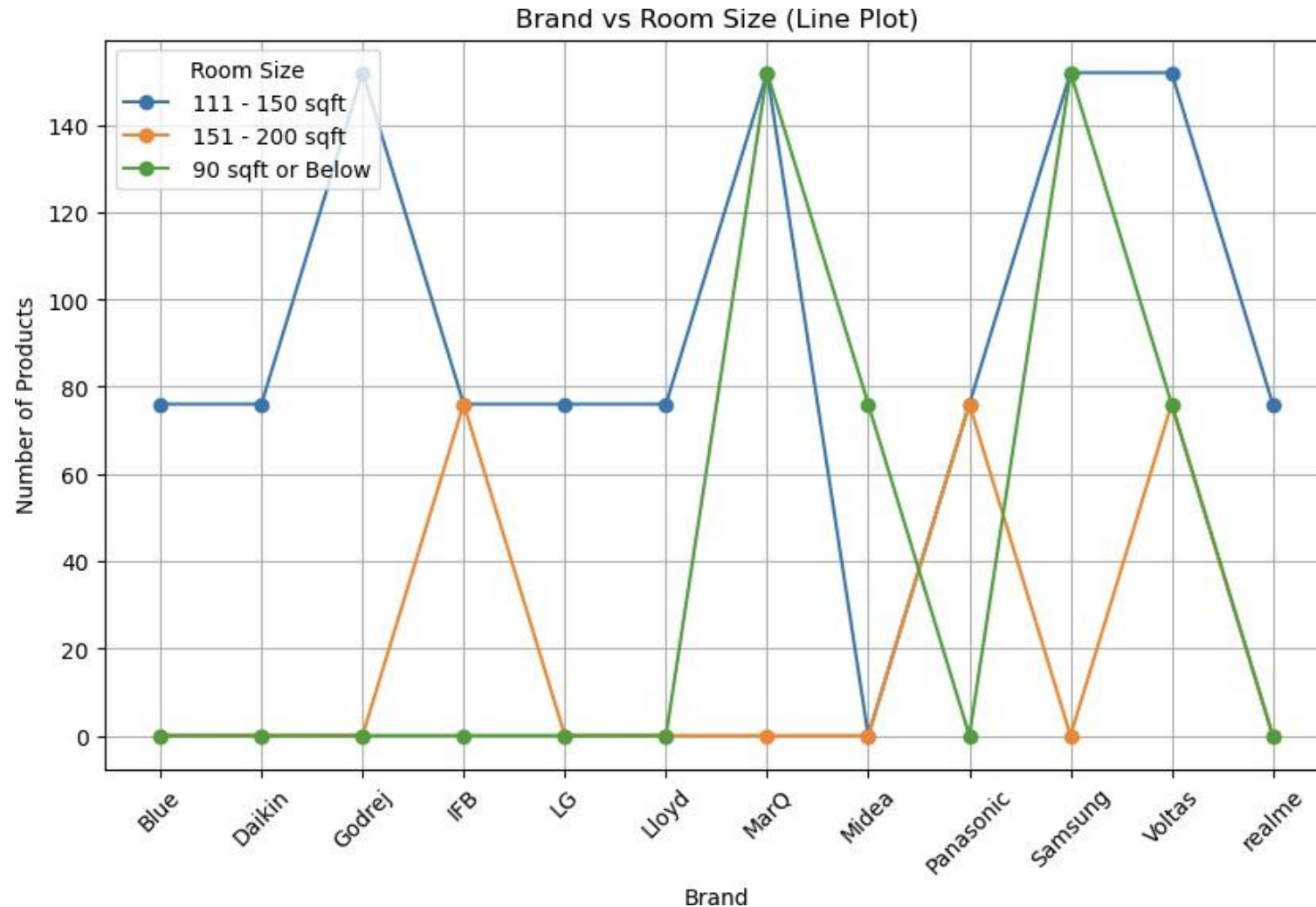
## 2. Price Distribution:

- Visualized **AC prices** using **bar plot**.
- Identifying **price ranges**.



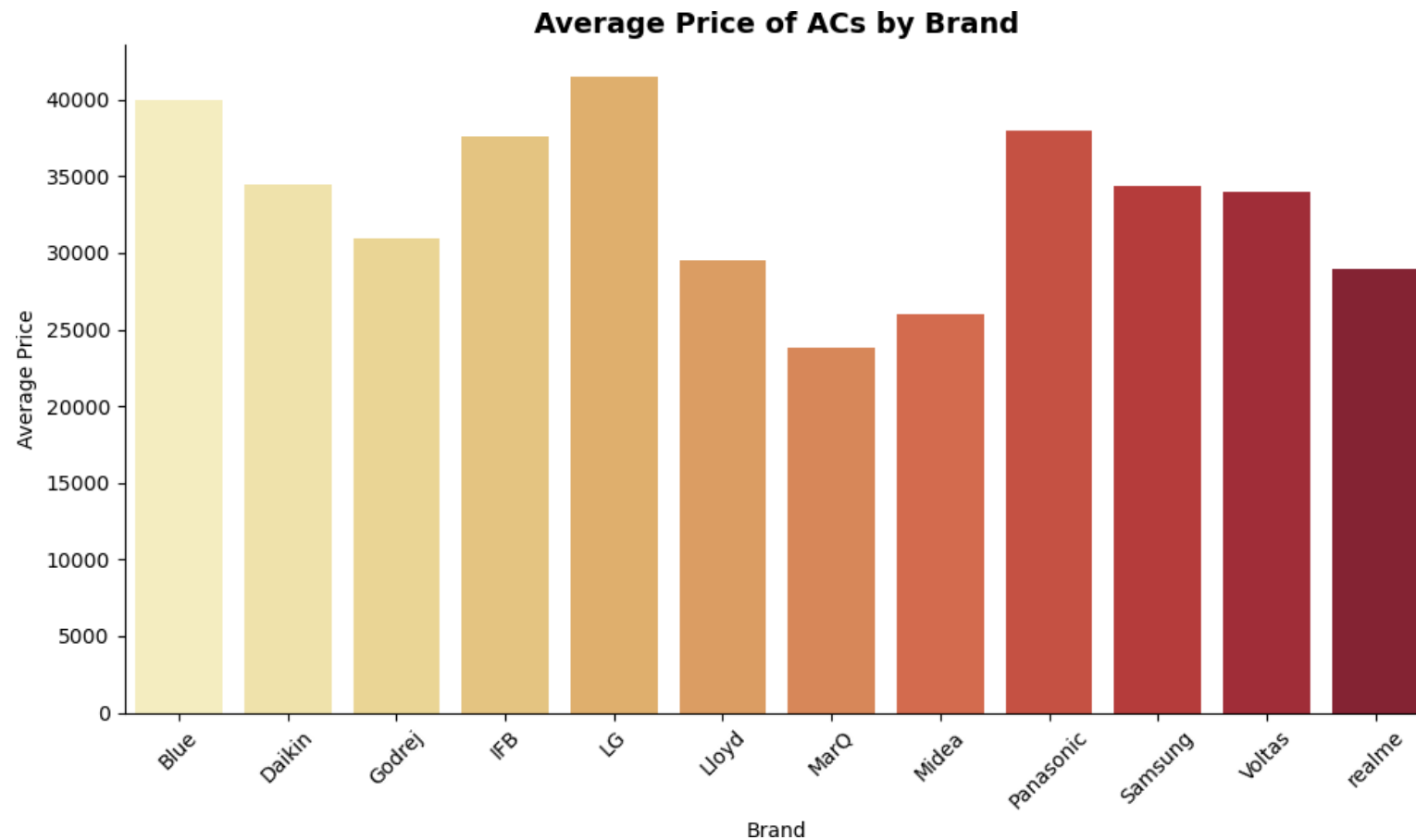
### 3. Brand Distribution across Area:

- Visualized **AC Brand** using **Line plot**.
- Identifying various **Room Sizes**.



#### 4. Brand-wise Price Comparison:

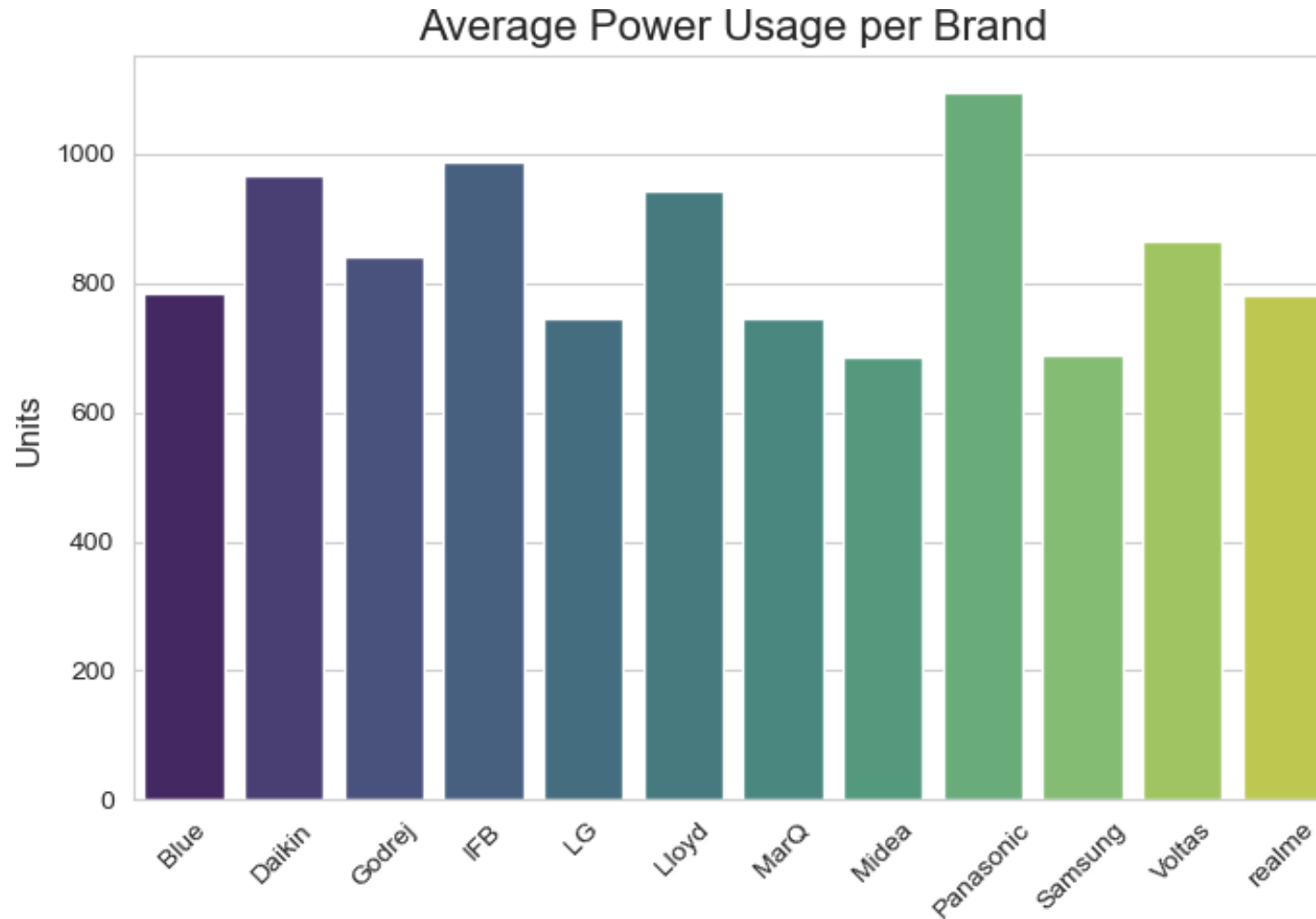
- Visualized **Average AC prices** using **bar plot** across **Brands**.





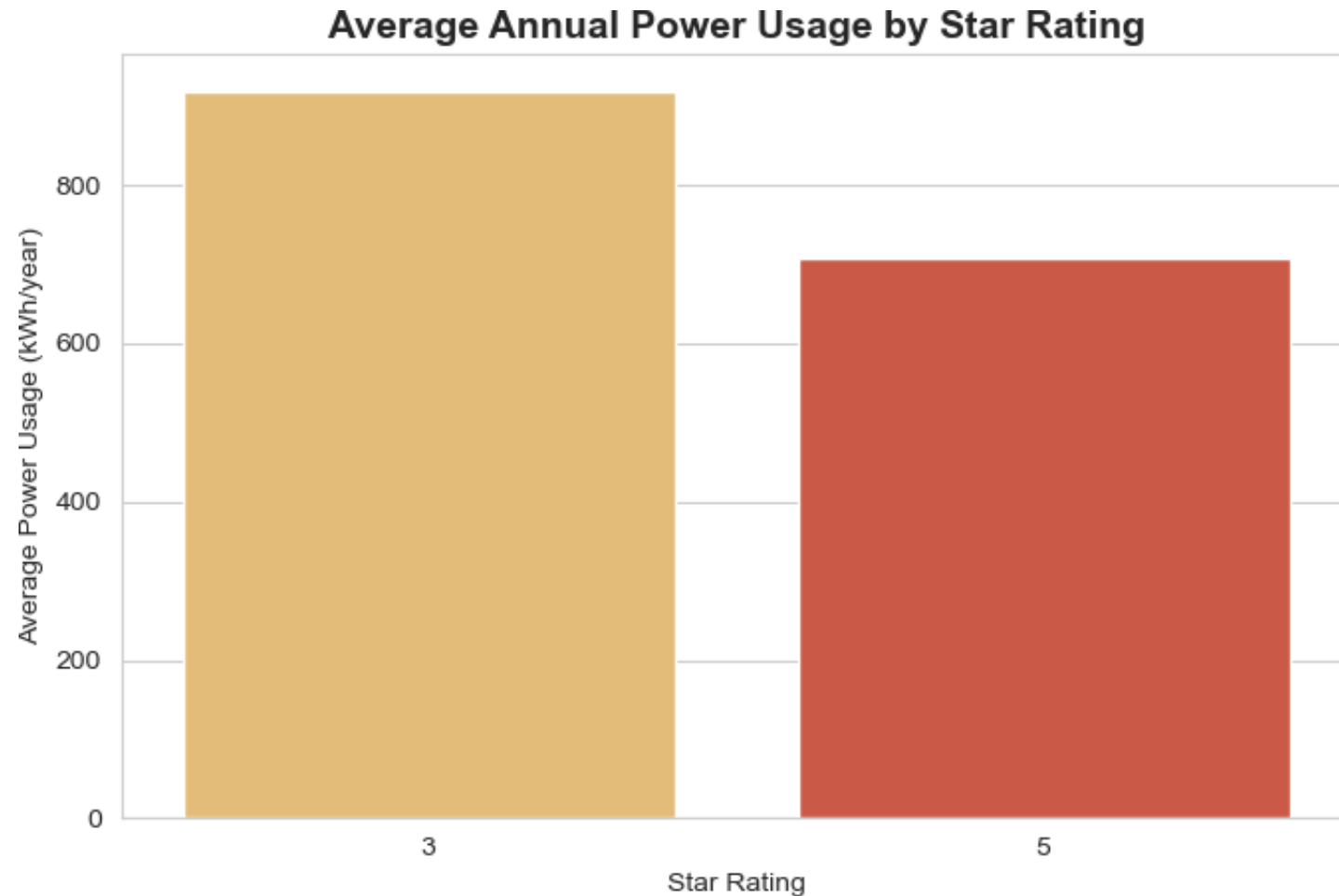
## 5. Brand-wise Power Distribution:

- Visualized **Average power usage** using **bar plot** across **Brands**.



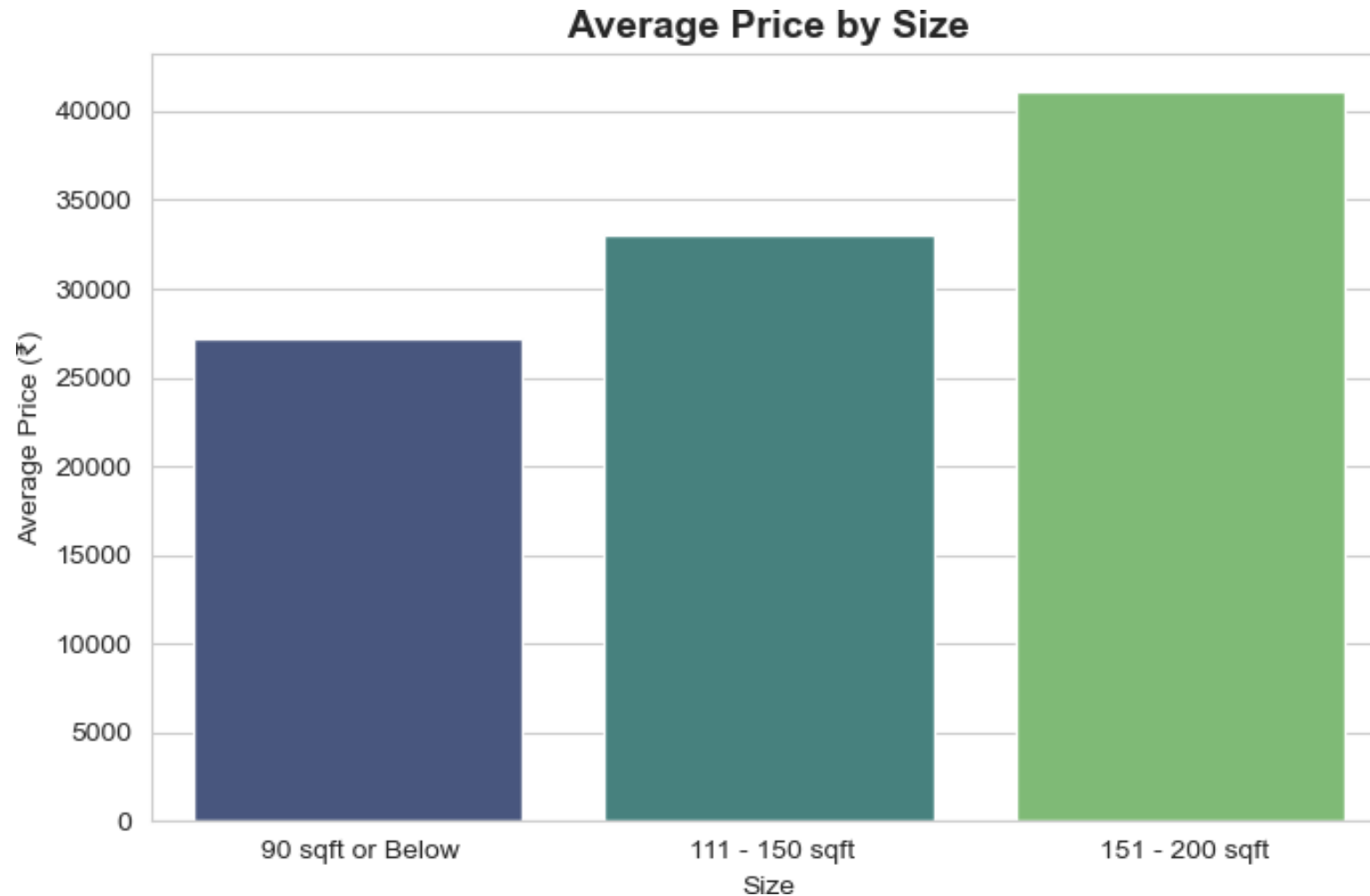
## 6. Power Distribution based on Star :

- Visualized **Average power Distribution** using **bar plot** across **Star rating**.



## 7. Price comparison based on Area :

- Visualized **Average Price Comparison** using **bar plot** across **Different areas**.



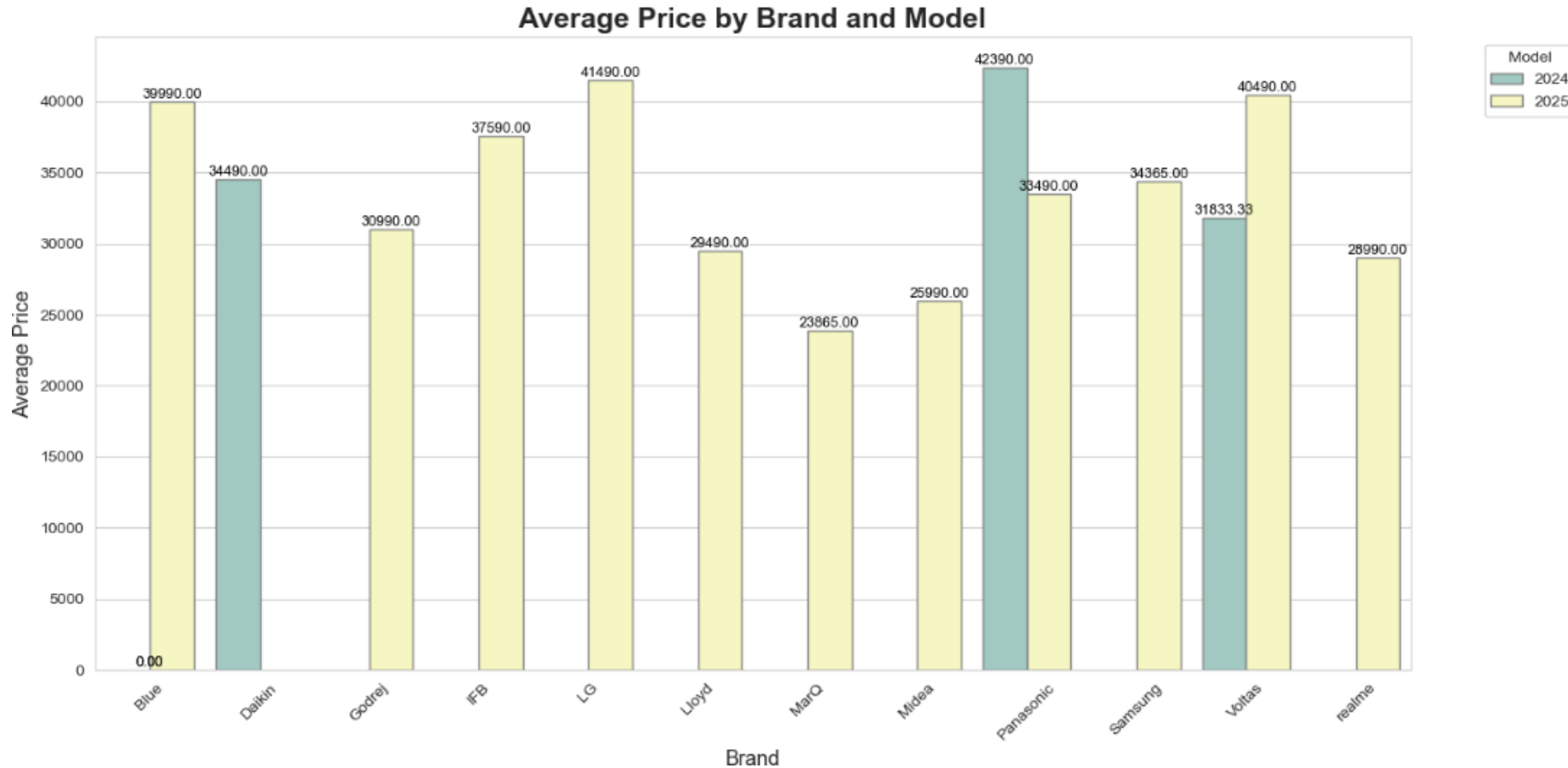
## 8. Price comparison based on Brands :

- Visualized **Average Price Comparison** using **bar plot** across **Different Brands**.



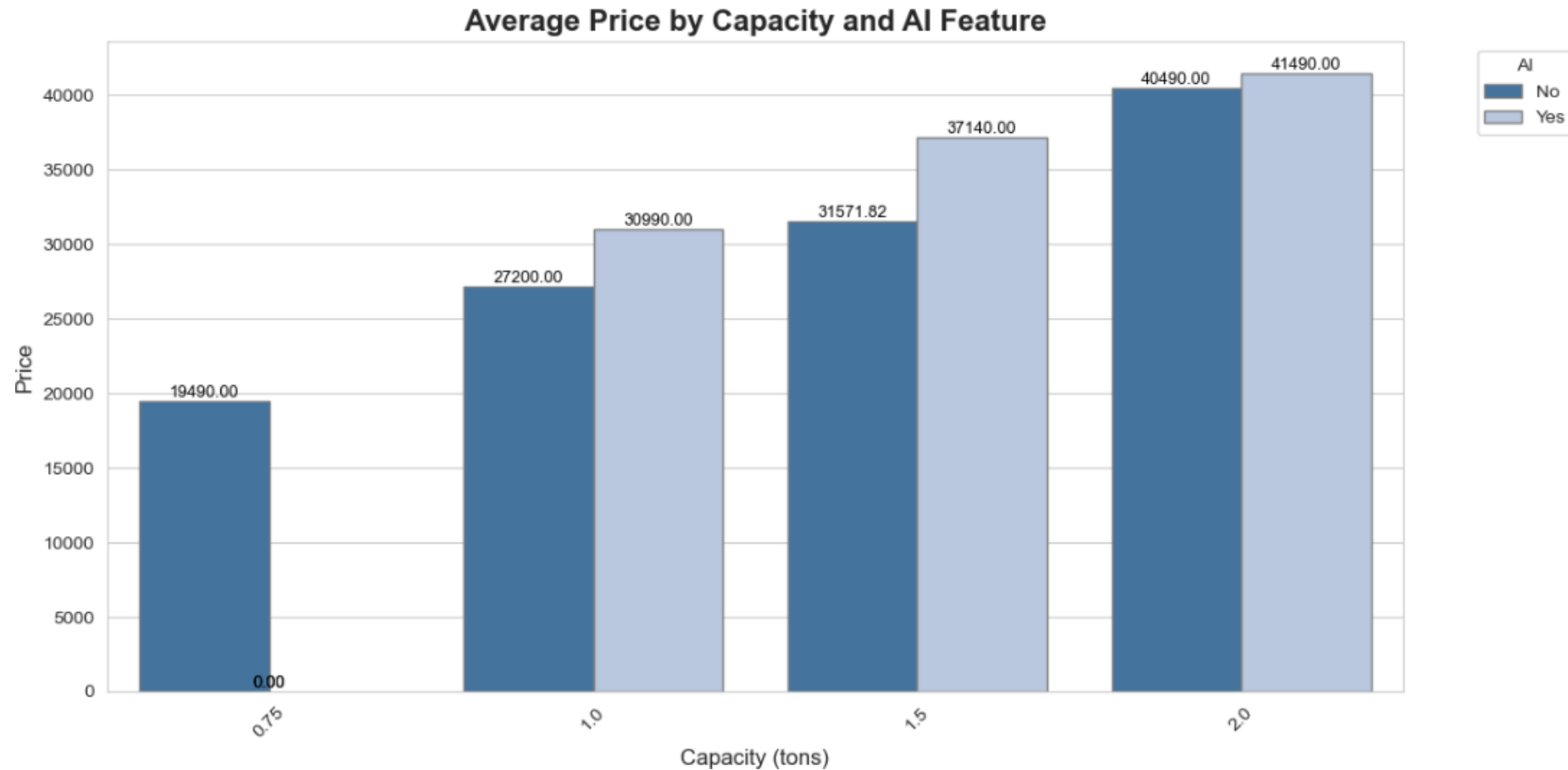
## 9. Price comparison based on Model :

- Visualized **Average Price Comparison** using **bar plot** across **Different Models**.

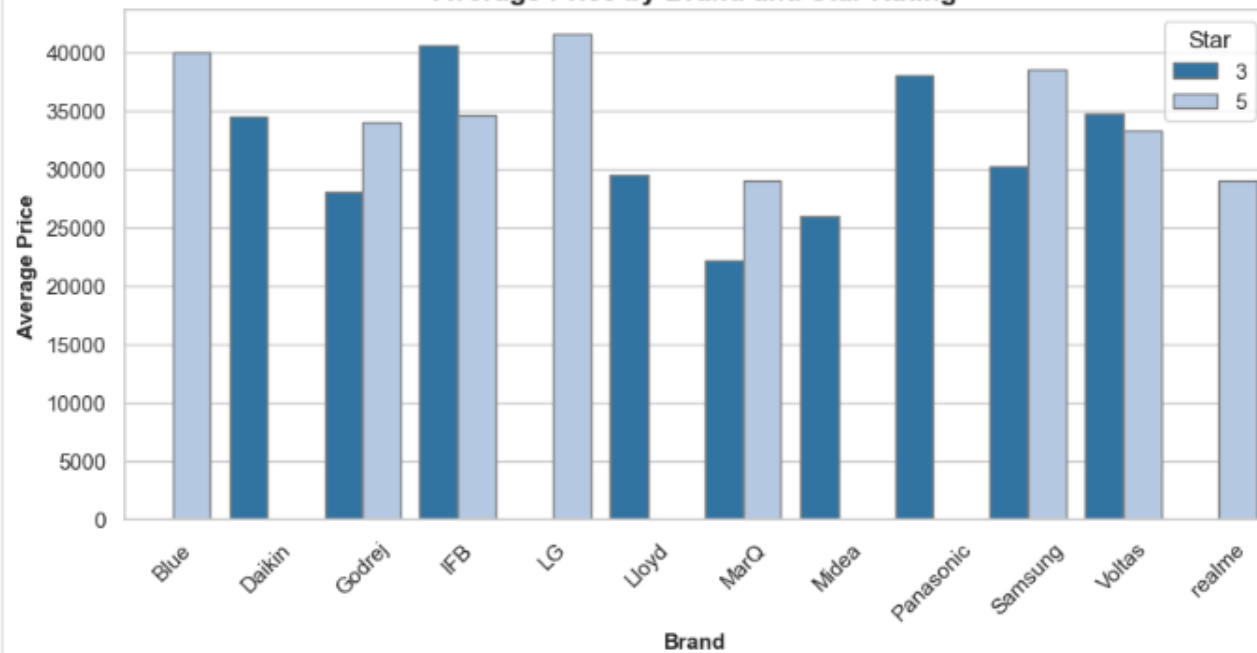


## 10. Price comparison based on AI :

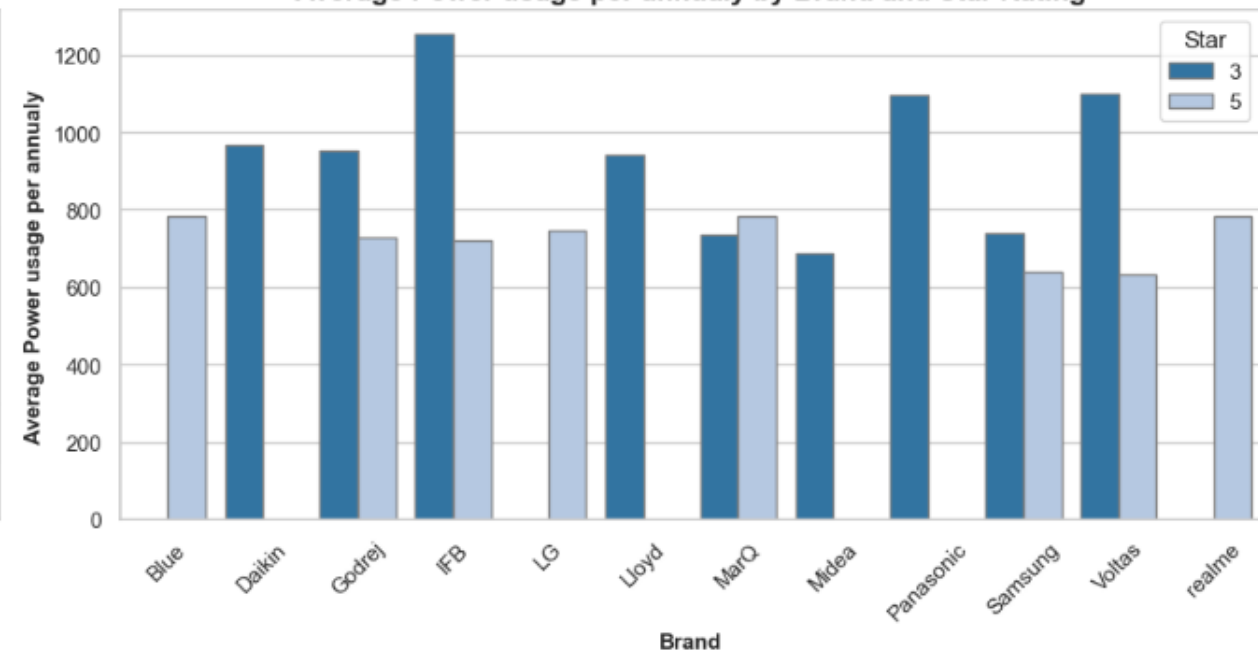
- Visualized **Average Price Comparison** using **bar plot** across **AI**.



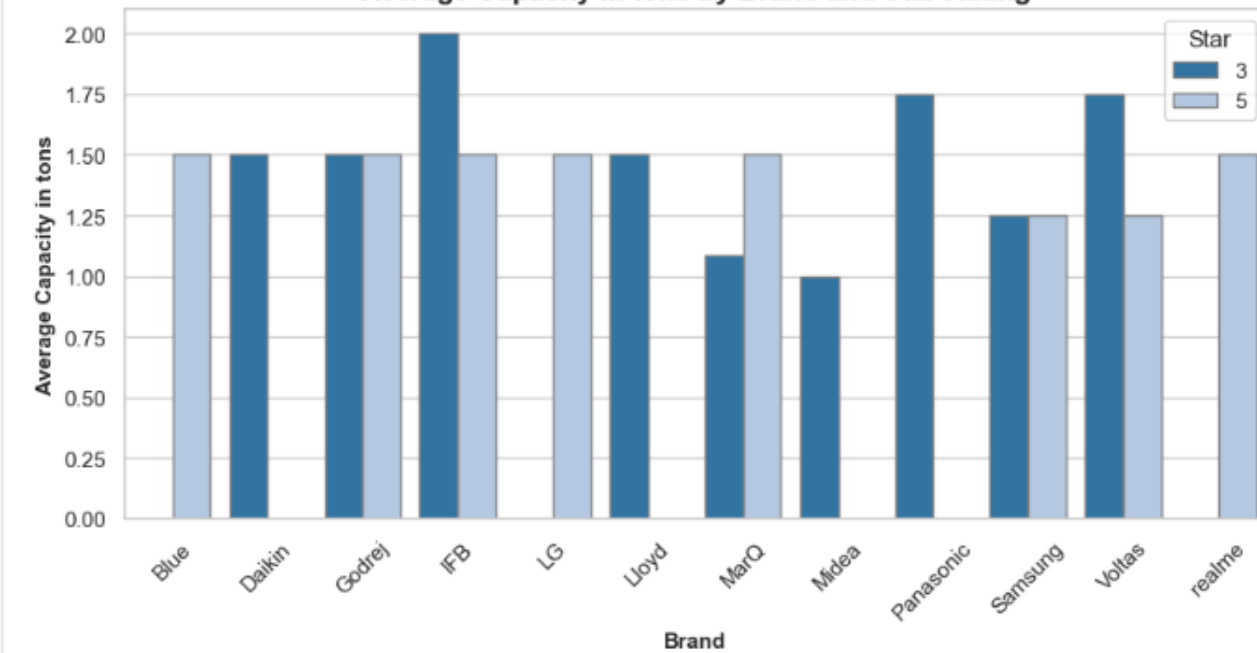
Average Price by Brand and Star Rating



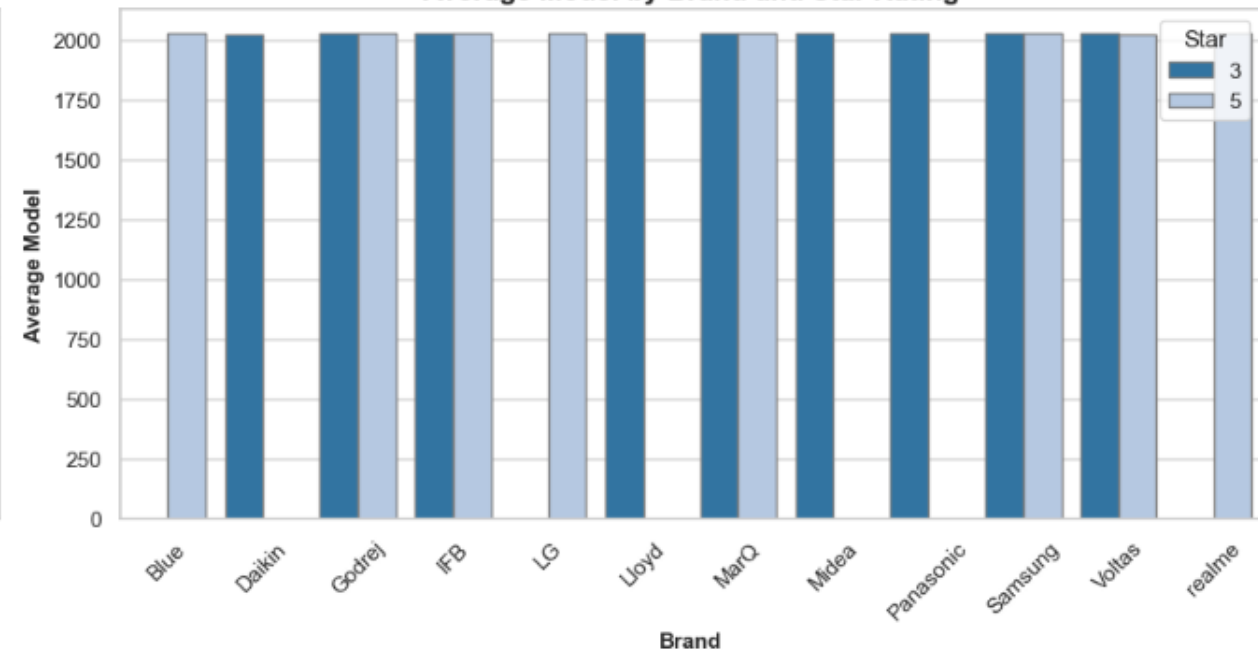
Average Power usage per annually by Brand and Star Rating



Average Capacity in tons by Brand and Star Rating



Average Model by Brand and Star Rating



# General Data Patterns

Metric	Observation
Average Price	₹27,000 – ₹32,000
Price Range	₹15,000 – ₹60,000
Most Common Capacity	<b>1.0 Ton</b> (around 50% of all ACs)
Most Common Star Rating	<b>3-Star</b>
Inverter AC Share	~70% of products
AI Feature Share	~30% of products

## Interpretation:

Mid-range 1.0-ton, 3-star inverter ACs dominate the market, reflecting consumer preference for energy efficiency and moderate cooling capacity.



# Brand Analysis

Brand	Median Price (₹)	Segment
LG	~36,000	Premium
Samsung	~34,000	Premium
Voltas	~29,000	Mid-range
MarQ	~21,000	Budget
Midea	~26,000	Budget-Mid

## Insights:

- **Voltas** and **MarQ** dominate the budget segment (<₹25k).
- **LG** and **Samsung** target the premium inverter AC market (>₹32k).
- **Midea** offers competitive pricing with AI and inverter features at mid-level prices.

# Capacity vs. Price

Capacity (Ton)	Avg Price (₹)	Trend
0.75	~19,000	Budget compact models
1.0	~27,000	Standard segment
1.5	~32,000	Premium segment
2.0	~38,000	High-end models

## Observation:

Price **increases proportionally with capacity**, showing a clear linear trend (correlation  $\approx +0.7$ ).

# Key Questions:

- Which brands offer the **best value for money**?
- How do features like **capacity, star rating, and inverter technology** impact price?
- What are the **seasonal trends** in AC pricing?

THANK  
YOU

