Lambton College

Lights, Camera, Stocks: Exploring the Impact of Warner Bros. Movie Releases on Stock Market Fluctuations.

Submitted By: -	**Submitted To: -**
Rajan Ghimire	**Victoria Shtern**

Introduction

The entertainment industry, particularly the film sector, has long been recognized for its potential to influence various aspects of society, including consumer behavior and market trends. This project explores the relationship between movie releases and stock market fluctuations, focusing on the renowned production company Warner Bros. Leveraging data from The Movie Database (TMDb) and Yahoo Finance, we analyze the impact of Warner Bros.' movie releases on the performance of related companies' stocks. By employing sentiment analysis, stock market data analysis, and visualization techniques, this study aims to uncover insights into how blockbuster releases may affect investor sentiment and stock market dynamics. Understanding these dynamics can provide valuable insights for investors, industry stakeholders, and researchers alike.

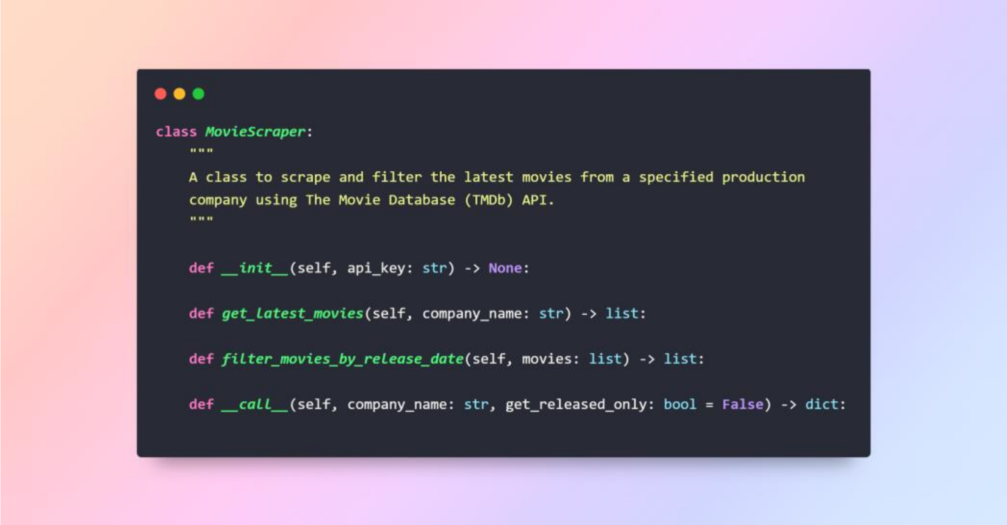
Methodology

1. Data Collection

Warner Bros. was chosen as the production company for analysis due to its prominence in the entertainment industry and its availability of stock market data on Yahoo Finance. Utilizing the MovieScraper class, movie details were collected from The Movie Database (TMDb) API. This involved downloading information on Warner Bros. movies, including titles, release dates, descriptions, and ratings.

1.1.Movie Data Acquisition

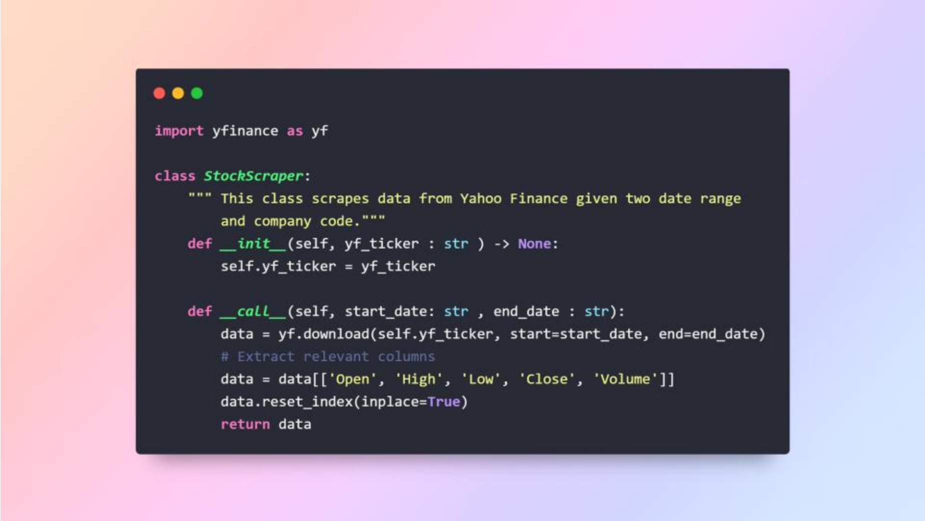
MovieScraper class was used to scrape the data.This class uses TMDb API to get all the movies based on the production company. If set the get_released_only variable to True, it will only give use the released movies before current date.



```
class MovieScraper:
    """
    A class to scrape and filter the latest movies from a specified production
    company using The Movie Database (TMDb) API.
    """

    def __init__(self, api_key: str) -> None:

    def get_latest_movies(self, company_name: str) -> list:

    def filter_movies_by_release_date(self, movies: list) -> list:

    def __call__(self, company_name: str, get_released_only: bool = False) -> dict:
```

1.2.Stock Market Data Retrieval

The StockScraper class was employed to gather historical stock market data for Warner Bros. (NYSE: WB) from Yahoo Finance. Data was collected for a time window spanning from 15 days before each movie's release date to 15 days after, capturing the potential impact of movie releases on stock performance.



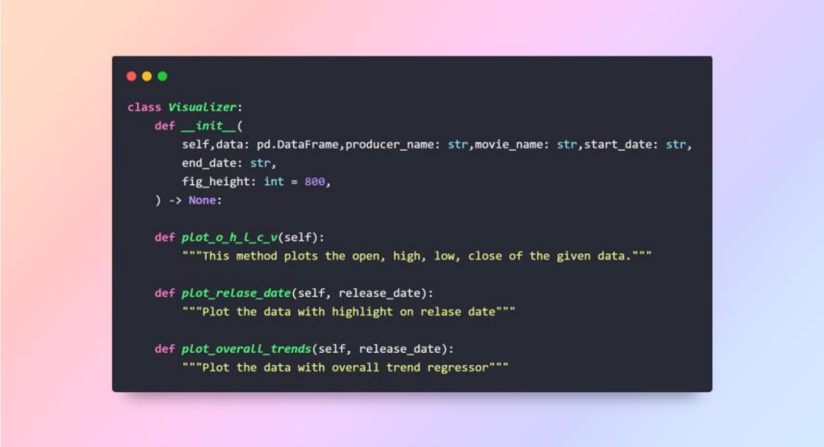
```
import yfinance as yf

class StockScraper:
    """ This class scrapes data from Yahoo Finance given two date range
        and company code."""
    def __init__(self, yf_ticker : str ) -> None:
        self.yf_ticker = yf_ticker

    def __call__(self, start_date: str , end_date : str):
        data = yf.download(self.yf_ticker, start=start_date, end=end_date)
        # Extract relevant columns
        data = data[['Open', 'High', 'Low', 'Close', 'Volume']]
        data.reset_index(inplace=True)
        return data
```

2. Data Processing and Analysis

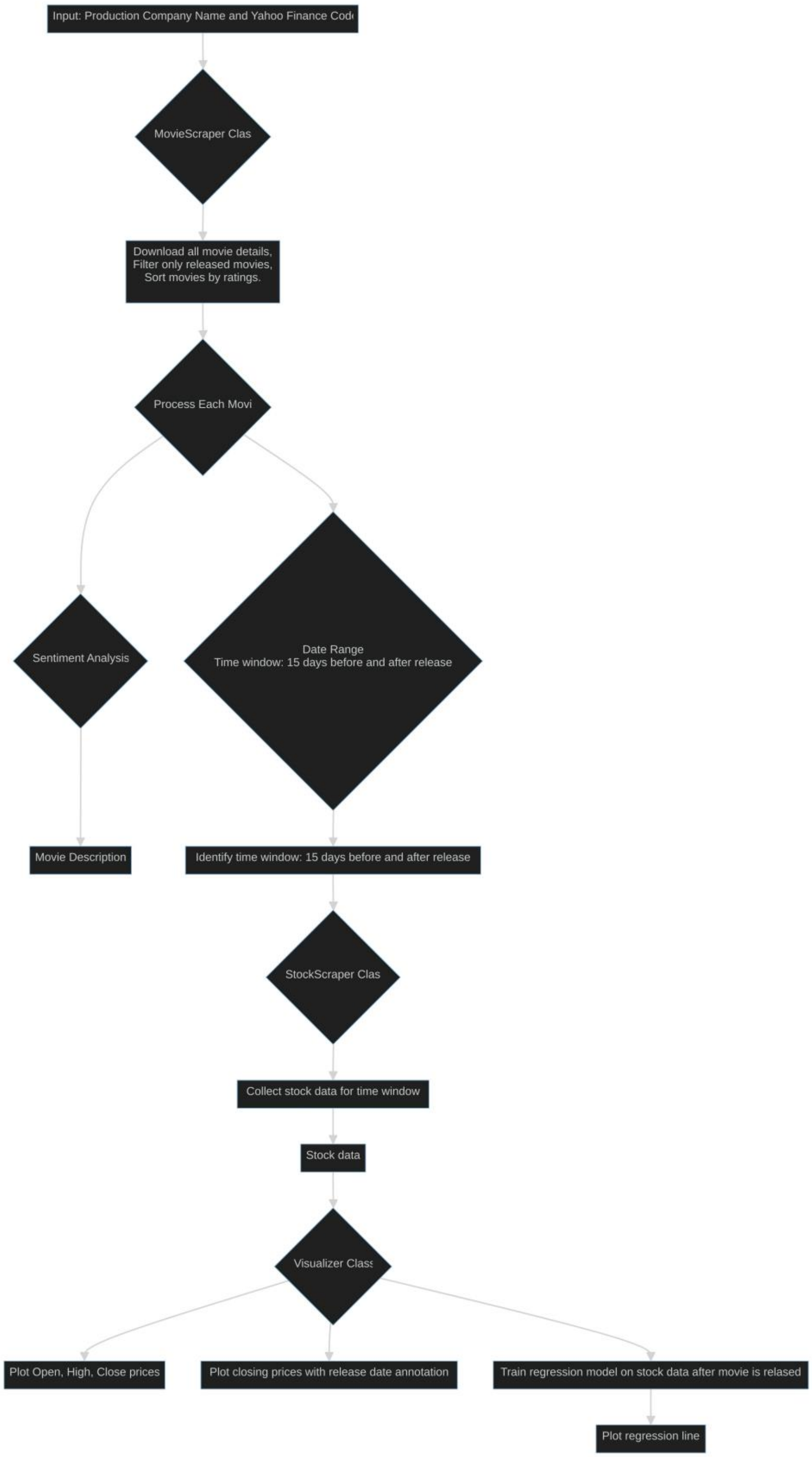
The descriptions of Warner Bros. movies were subjected to sentiment analysis using Natural Language Processing (NLP) techniques. Stock market data collected for Warner Bros. was analyzed to identify trends and patterns around the release dates of movies. Visualizations, including plots of stock prices and regression models, were generated to explore correlations between movie releases and stock market fluctuations.

3. Visualization and Interpretation

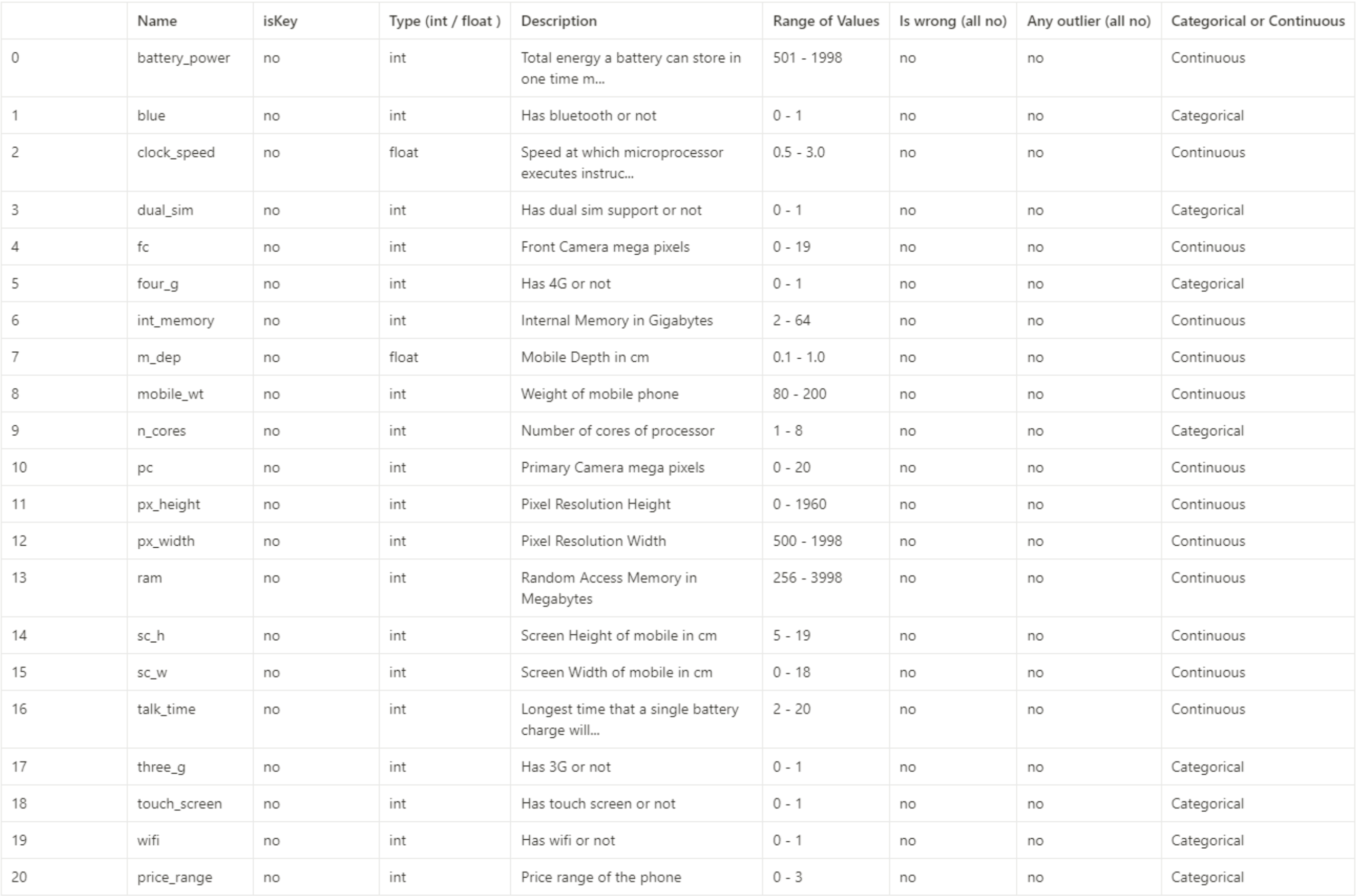
The Visualizer class was utilized to create visual representations of the data analysis findings. This included plotting stock price trends, annotating movie release dates, and visualizing regression models fitted to the stock data.



```
class Visualizer:
    def __init__(
        self,data: pd.DataFrame,producer_name: str,movie_name: str,start_date: str,
        end_date: str,
        fig_height: int = 800,
    ) -> None:

    def plot_o_h_l_c_v(self):
        """This method plots the open, high, low, close of the given data."""

    def plot_relase_date(self, release_date):
        """Plot the data with highlight on relase date"""

    def plot_overall_trends(self, release_date):
        """Plot the data with overall trend regressor"""
```


Input: Production Company Name and Yahoo Finance Cod
MovieScraper Clas
Download all movie details,
Filter only released movies,
Sort movies by ratings.
Process Each Movi
Sentiment Analysis
Date Range
Time window: 15 days before and after release
Movie Description
Identify time window: 15 days before and after release
StockScraper Clas
Collect stock data for time window
Stock data
Visualizer Class
Plot Open, High, Close prices
Plot closing prices with release date annotation
Train regression model on stock data after movie is relased
Plot regression line

Feature Analysis



	Name	isKey	Type (int / float)	Description	Range of Values	Is wrong (all no)	Any outlier (all no)	Categorical or Continuous
0	battery_power	no	int	Total energy a battery can store in one time m...	501 - 1998	no	no	Continuous
1	blue	no	int	Has bluetooth or not	0 - 1	no	no	Categorical
2	clock_speed	no	float	Speed at which microprocessor executes instruc...	0.5 - 3.0	no	no	Continuous
3	dual_sim	no	int	Has dual sim support or not	0 - 1	no	no	Categorical
4	fc	no	int	Front Camera mega pixels	0 - 19	no	no	Continuous
5	four_g	no	int	Has 4G or not	0 - 1	no	no	Categorical
6	int_memory	no	int	Internal Memory in Gigabytes	2 - 64	no	no	Continuous
7	m_dep	no	float	Mobile Depth in cm	0.1 - 1.0	no	no	Continuous
8	mobile_wt	no	int	Weight of mobile phone	80 - 200	no	no	Continuous
9	n_cores	no	int	Number of cores of processor	1 - 8	no	no	Categorical
10	pc	no	int	Primary Camera mega pixels	0 - 20	no	no	Continuous
11	px_height	no	int	Pixel Resolution Height	0 - 1960	no	no	Continuous
12	px_width	no	int	Pixel Resolution Width	500 - 1998	no	no	Continuous
13	ram	no	int	Random Access Memory in Megabytes	256 - 3998	no	no	Continuous
14	sc_h	no	int	Screen Height of mobile in cm	5 - 19	no	no	Continuous
15	sc_w	no	int	Screen Width of mobile in cm	0 - 18	no	no	Continuous
16	talk_time	no	int	Longest time that a single battery charge will...	2 - 20	no	no	Continuous
17	three_g	no	int	Has 3G or not	0 - 1	no	no	Categorical
18	touch_screen	no	int	Has touch screen or not	0 - 1	no	no	Categorical
19	wifi	no	int	Has wifi or not	0 - 1	no	no	Categorical
20	price_range	no	int	Price range of the phone	0 - 3	no	no	Categorical

Exploratory Data Analysis

Missing Values in Data:

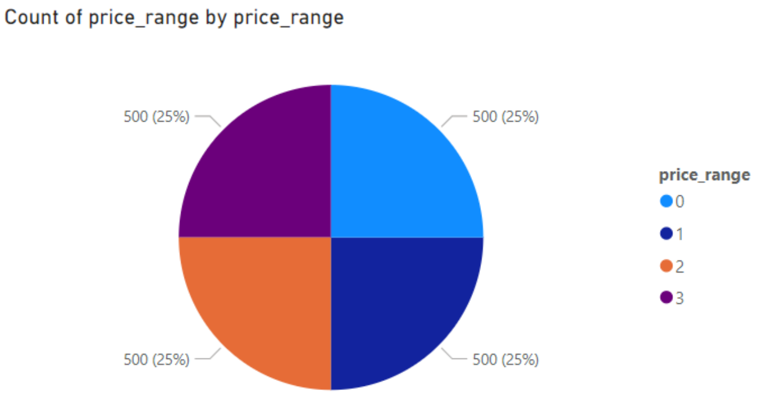
The dataset contains no such missing values.

	Column	Missing_Percentage
0	battery_power	0
1	blue	0
2	clock_speed	0
3	dual_sim	0
4	fc	0
5	four_g	0
6	int_memory	0
7	m_dep	0
8	mobile_wt	0
9	n_cores	0
10	pc	0
11	px_height	0
12	px_width	0
13	ram	0
14	sc_h	0
15	sc_w	0
16	talk_time	0
17	three_g	0
18	touch_screen	0
19	wifi	0
20	price_range	0

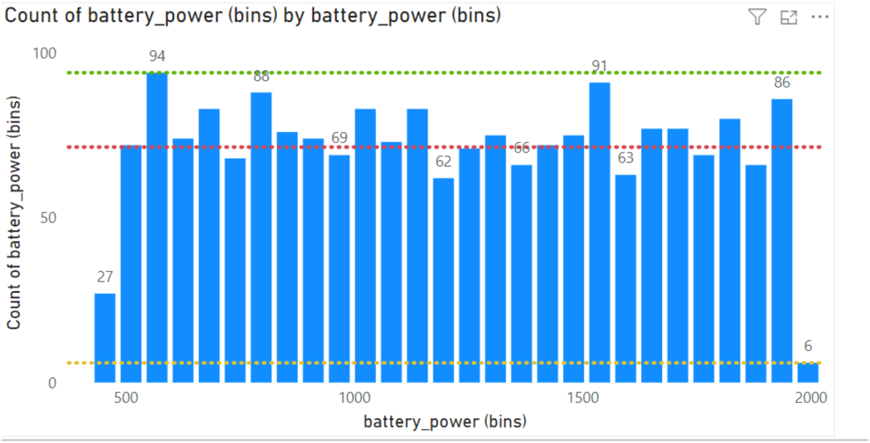
Analysis of class variable ‘price range’:

price_range	Count of price_range
3	500
2	500
0	500
1	500
Grand Total	**2000**

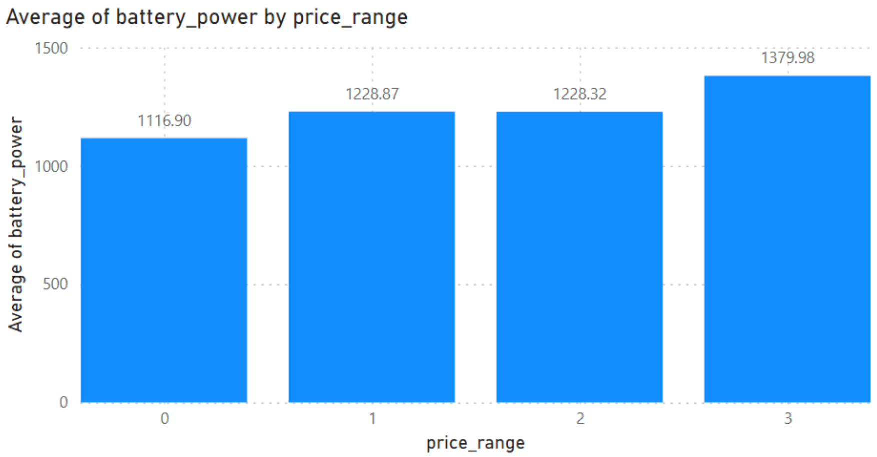
The price range is equally distributed among 4 classes. 0 means low cost, 1 means Medium Cost, 2 means high cost and 3 means very high cost.



Analysis of battery power:

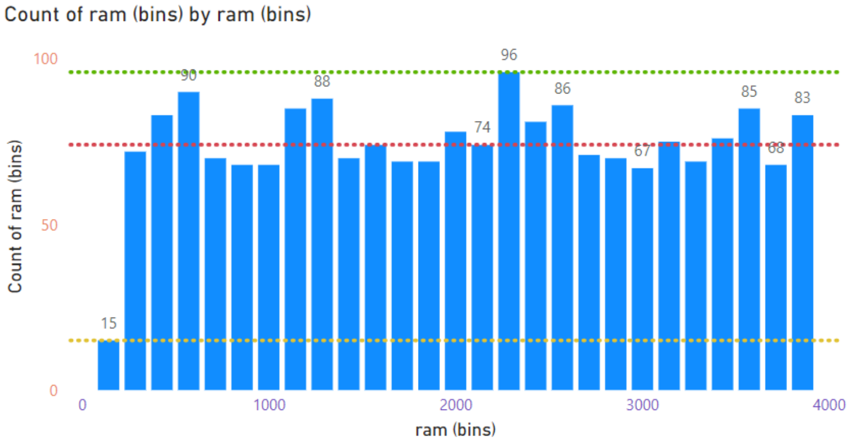


The graph shows the distribution of battery power. Here, the yellow line represents the minimum values if the distribution, red line depicts the average of the distribution and green line shows the maximum values of the distribution.

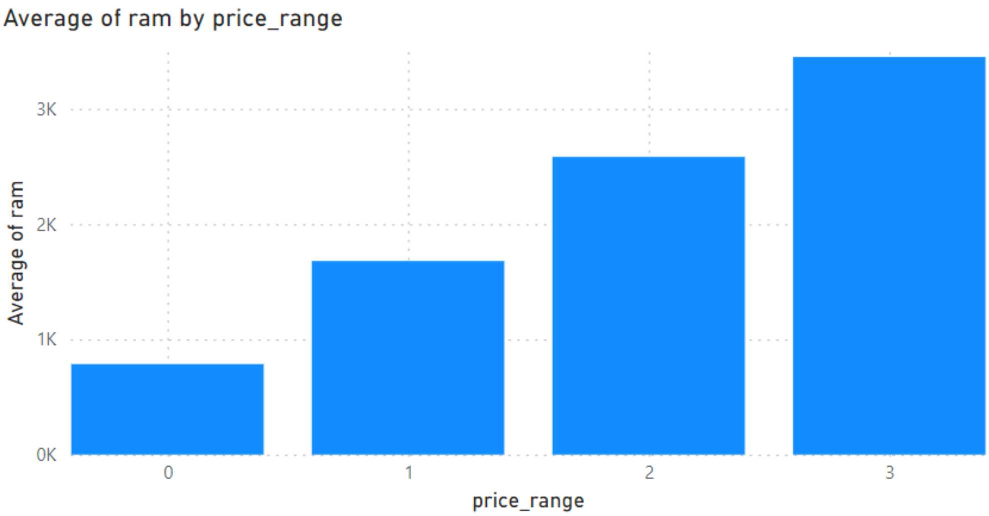


Here, after looking at average battery power per each price range we can interpret that: There is positive co-relation between price range and battery power. As the battery power increases, the price of the mobile also increases.

Analysis of ram:



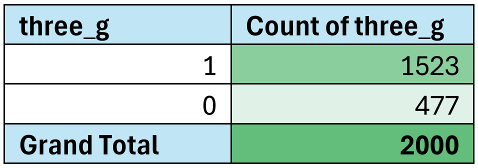
The histogram shows the distribution of ram. Here, the yellow line represents the minimum values if the distribution, red line depicts the average of the distribution and green line shows the maximum values of the distribution.



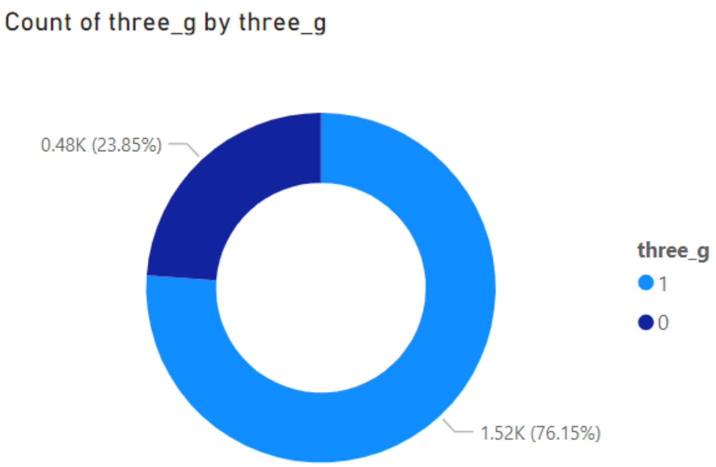
Here, after looking at the average ram per each price range we can interpret that: There is positive co-relation between price range and ram. As the size of ram increases, the price of the mobile also increases.

Analysis of 3G:

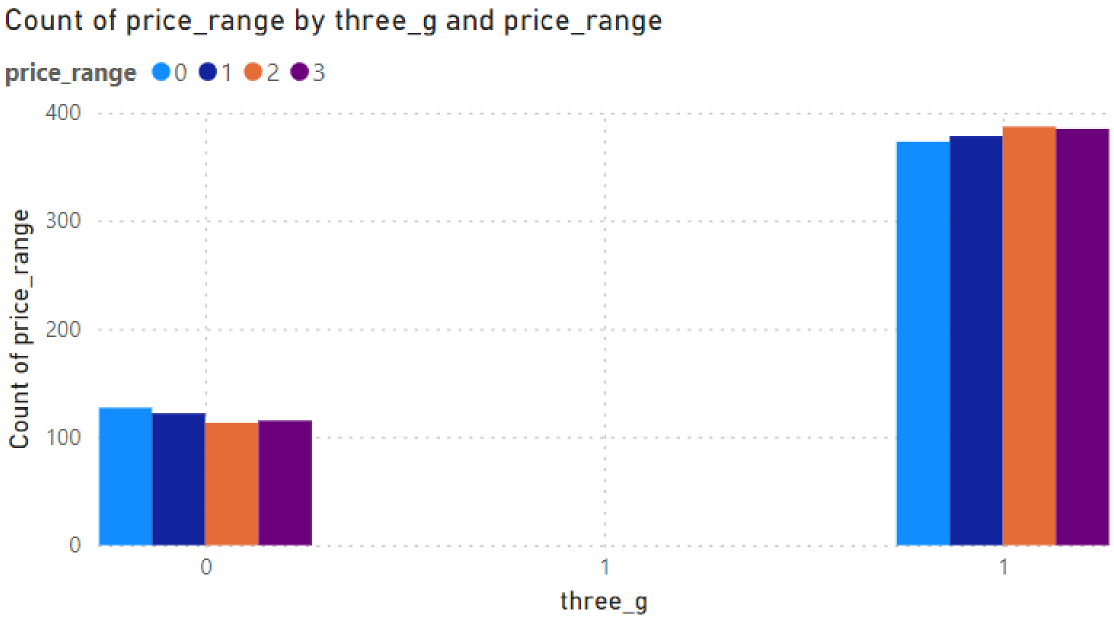
The feature three_g is a categorical variable with two classes 1 and 0. 1 represents presence of 3G feature in phone and 0 represents absence of meeting.



three_g	Count of three_g
1	1523
0	477
Grand Total	**2000**



From the given donut plot, we can understand that most of the phones have 3G.

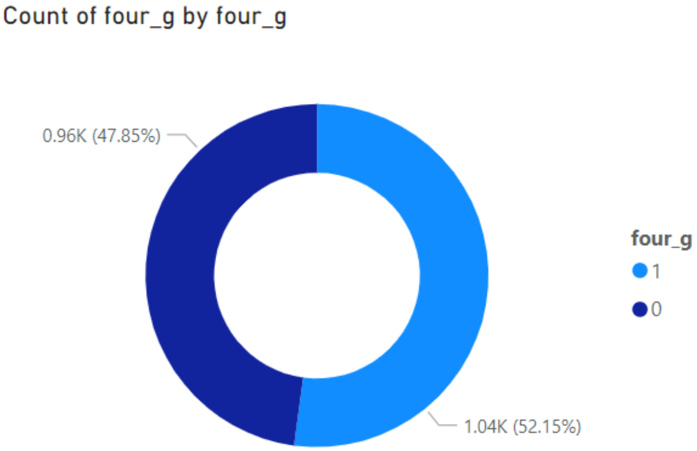


Now, when we look at the count of 3G per price category, we can interpret that the price goes up when the phone has 3G.

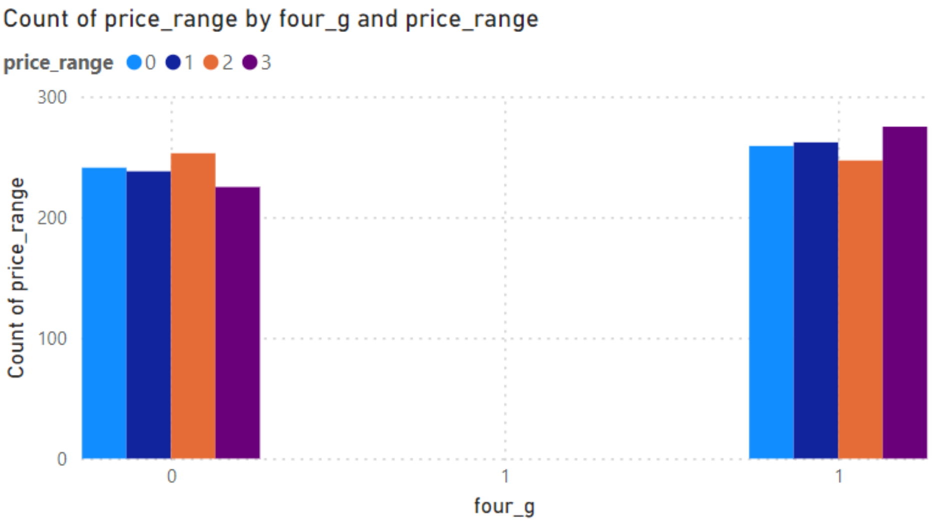
Analysis of 4G:

The feature four_g is a categorical variable with two classes 1 and 0. 1 represents presence of 4G feature in phone and 0 represents absence of meeting.

four_g	Count of four_g
1	1043
0	957
Grand Total	**2000**



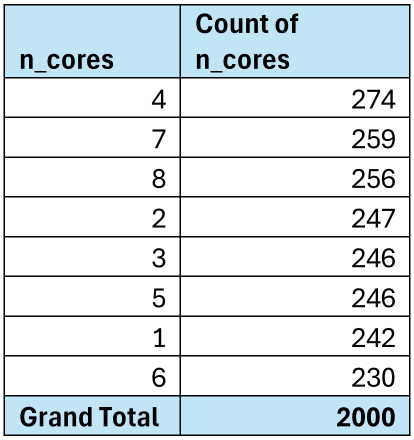
From the given donut plot, we can understand that there is equal distribution between phone with and without 4G.



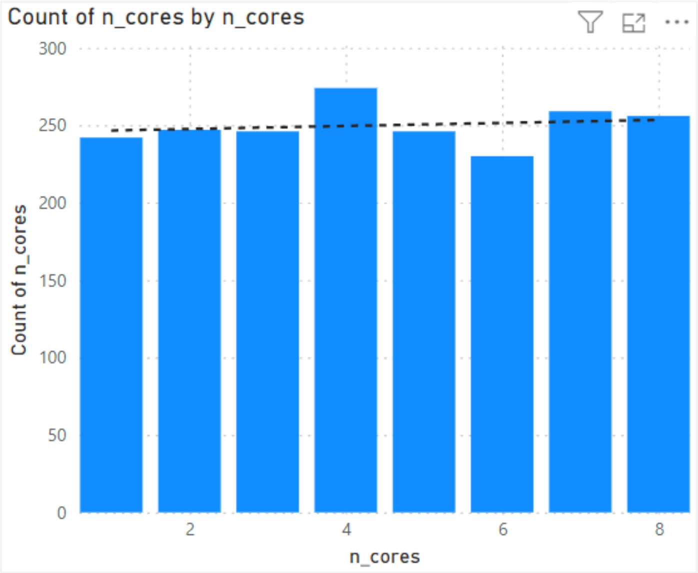
Now, when we look at the count of 4G per price category, we can interpret that 4G feature does not have that much of difference in price of the mobile phone.

Analysis of n_cores:

The n_cores features represent the total number of processing cores within the system. The higher the number of processing cores, the better the performance.

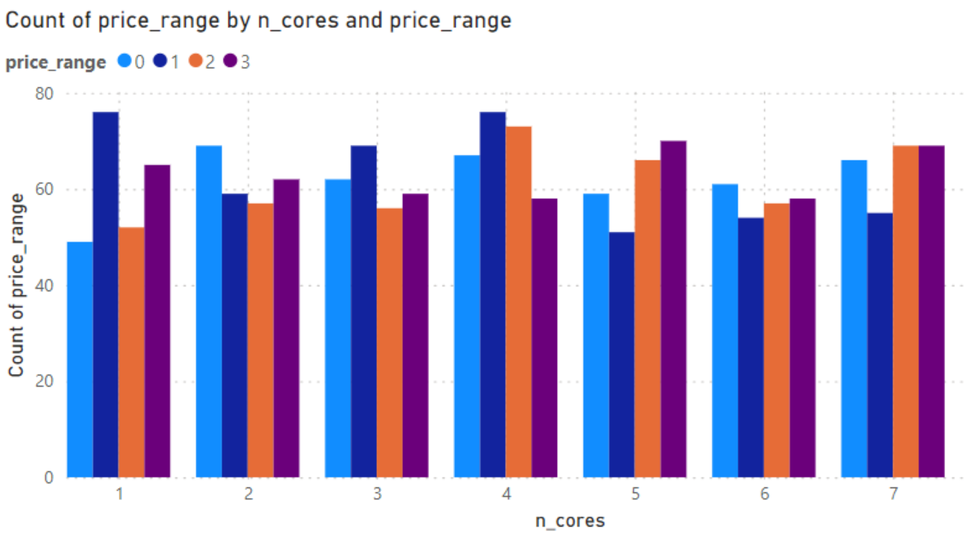


n_cores	Count of n_cores
4	274
7	259
8	256
2	247
3	246
5	246
1	242
6	230
Grand Total	**2000**



The histogram shows the distribution of n_cores. The black dotted line represents overall trend of the distribution.

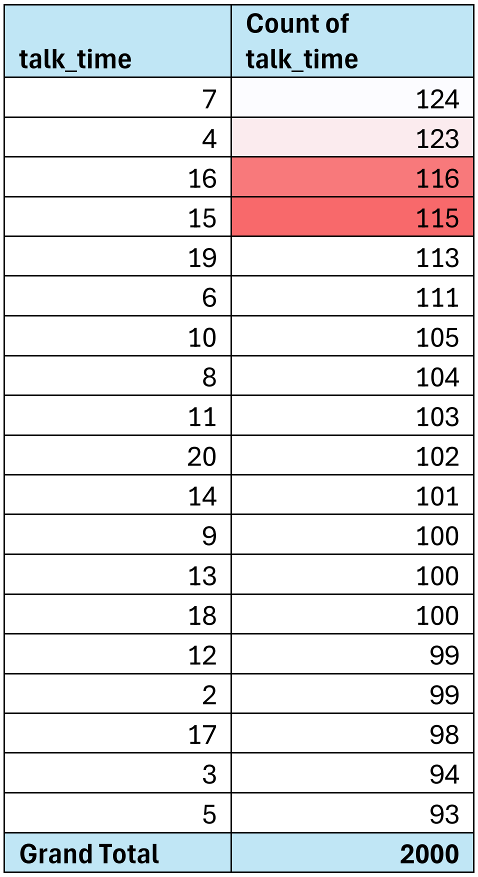
Now, when we look at the count of n_cores per price category, we can interpret that n_cores feature does not have that much of difference in price of the mobile phone. This means, mobile phones of different price range have same number of cores.



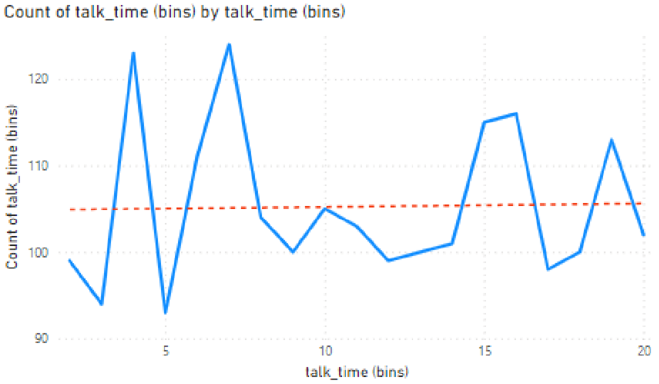
Count of price_range by n_cores and price_range

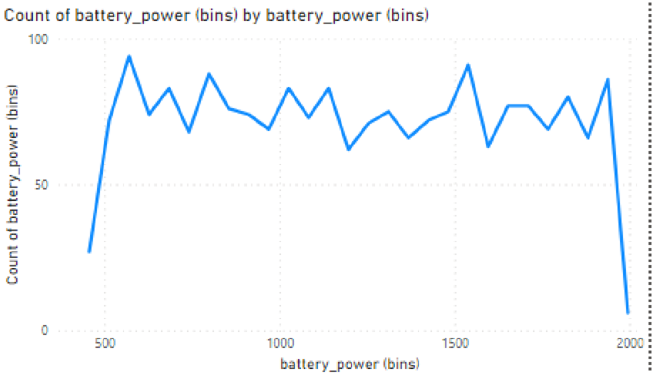
Analysis of talk_time vs battery_power:

In this analysis, we are trying to find out if mobile phones with longer talk time have longer battery life. This is a confirmatory data analysis where we want to confirm this hypothesis.

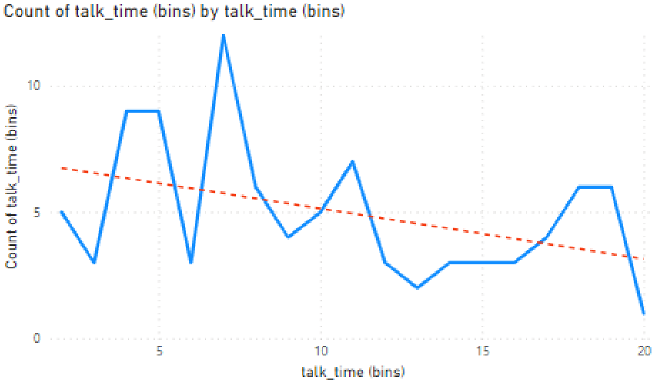


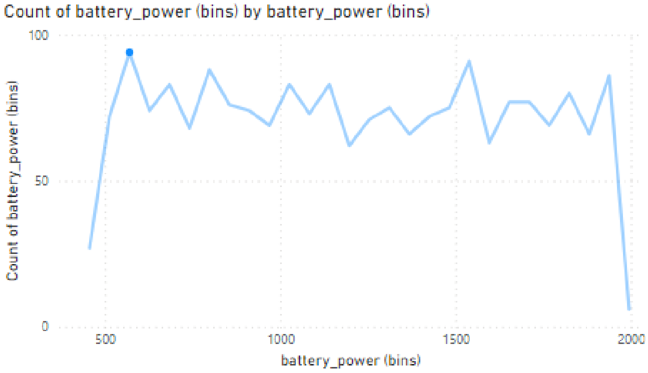
talk_time	Count of talk_time
7	124
4	123
16	116
15	115
19	113
6	111
10	105
8	104
11	103
20	102
14	101
9	100
13	100
18	100
12	99
2	99
17	98
3	94
5	93
Grand Total	**2000**



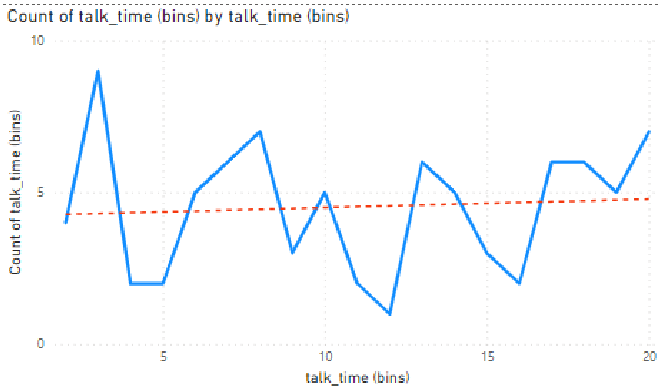


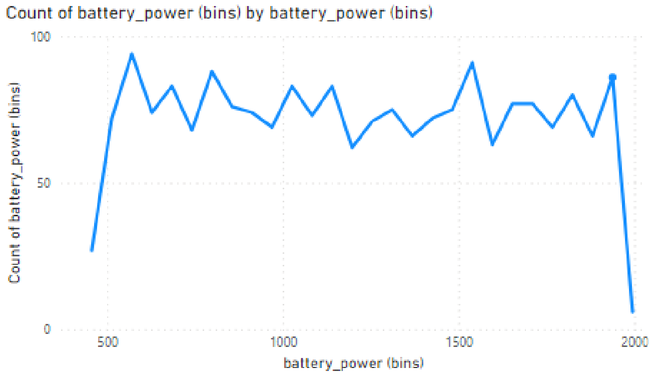
Here we have two-line charts that show count of talk time and battery for different bins. The red line is to indicate overall trend in the talk time.





Here If we select lower battery power, the overall talk time is decreasing shown by the red trend line in talk time.





Similarly, when we select higher talk time, we can see that overall talk time is increasing shown by the red trend line which proves our hypothesis.

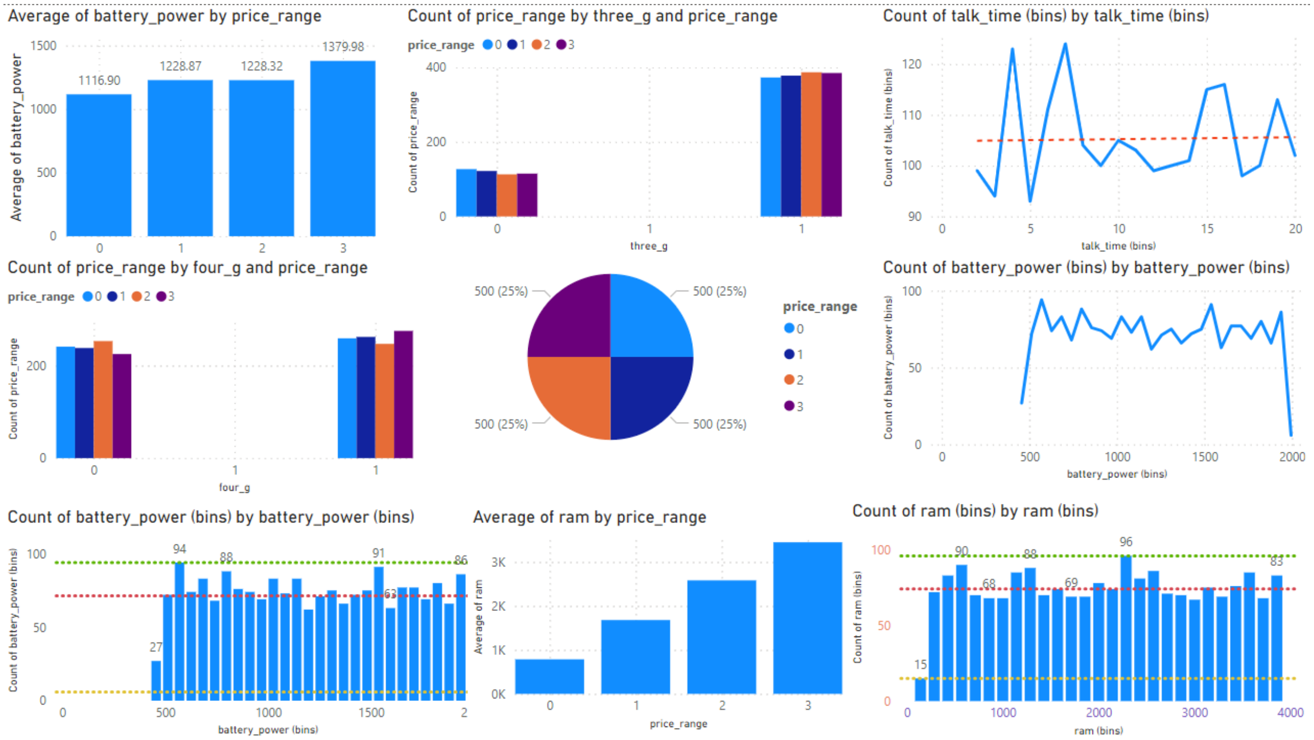
Conclusion

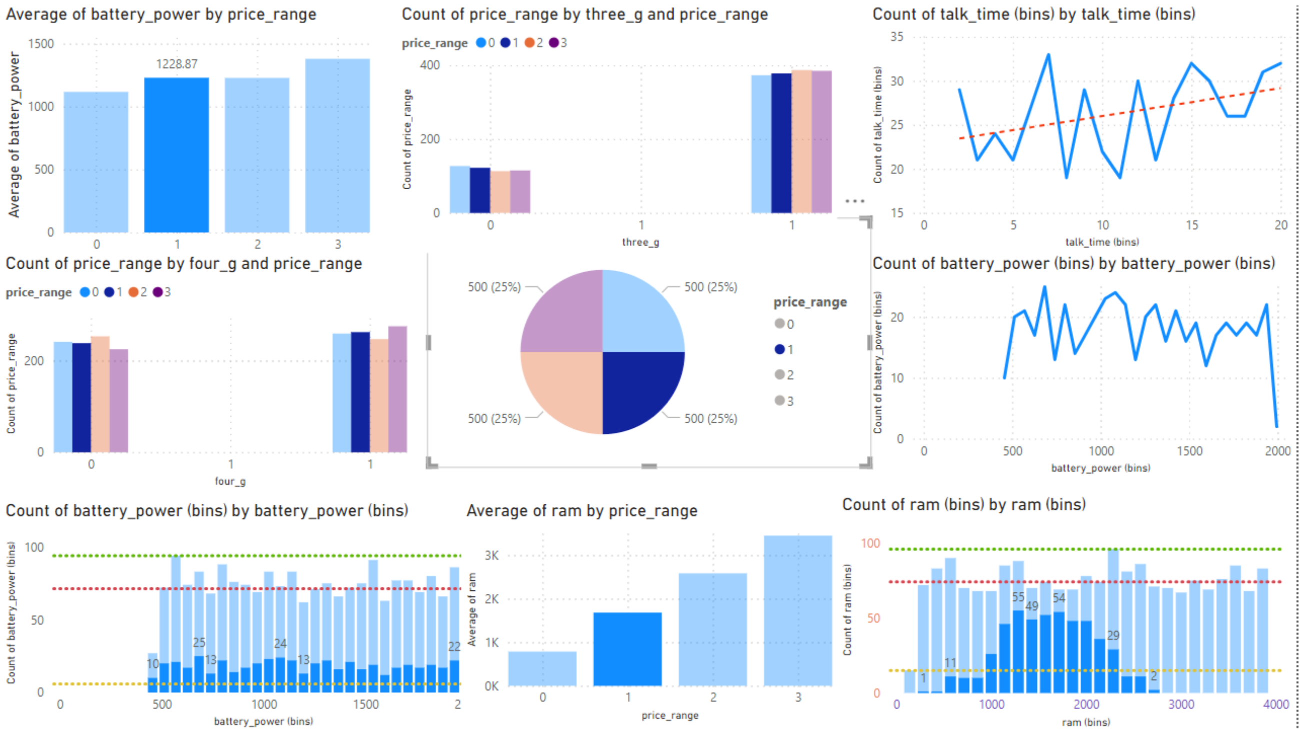
The exploratory data analysis on the mobile device dataset has yielded valuable insights into the relationships between various features and the price range of the devices. The analysis revealed a positive correlation between battery power, RAM capacity, and the device's price range, suggesting that higher-end devices tend to offer larger batteries and more RAM. Interestingly, the presence of 3G connectivity appeared to influence pricing, while 4G connectivity and the number of processor cores did not significantly impact pricing across different price categories. Additionally, the analysis confirmed the expected behavior that lower battery power corresponds to shorter talk times, and vice versa. These findings can inform product development strategies, pricing decisions, and marketing efforts, enabling manufacturers and retailers to better align their offerings with consumer preferences and market dynamics within the highly competitive mobile device.

References

Iabhishekofficial. (n.d.). Mobile Price Classification. Kaggle. Retrieved June 5, 2024, from https://www.kaggle.com/datasets/iabhishekofficial/mobile-price-classification

Dashboard


Average of battery_power by price_range
Average of battery_power
1500
1000
500
0
1116.90
1228.87
1228.32
1379.98
0
1
2
3
Count of price_range by three_g and price_range
price_range ● 0 ● 1 ● 2 ● 3
Count of price_range
400
200
0
0
1
1
three_g
Count of talk_time (bins) by talk_time (bins)
Count of talk_time (bins)
120
110
100
90
0
5
10
15
20
talk_time (bins)
Count of price_range by four_g and price_range
price_range ● 0 ● 1 ● 2 ● 3
Count of price_range
200
0
0
1
1
four_g
500 (25%)
500 (25%)
500 (25%)
500 (25%)
price_range
● 0
● 1
● 2
● 3
Count of battery_power (bins) by battery_power (bins)
Count of battery_power (bins)
100
50
0
0
500
1000
1500
2000
battery_power (bins)
Count of battery_power (bins) by battery_power (bins)
Count of battery_power (bins)
100
50
0
94
88
91
86
27
63
0
500
1000
1500
2
battery_power (bins)
Average of ram by price_range
Average of ram
3K
2K
1K
0K
0
1
2
3
price_range
Count of ram (bins) by ram (bins)
Count of ram (bins)
100
50
0
90
88
96
83
68
69
15
0
1000
2000
3000
4000
ram (bins)


Average of battery_power by price_range
Average of battery_power
1500
1000
500
0
1228.87
0
1
2
3
Count of price_range by three_g and price_range
price_range ● 0 ● 1 ● 2 ● 3
Count of price_range
400
200
0
0
1
1
three_g
Count of talk_time (bins) by talk_time (bins)
Count of talk_time (bins)
35
30
25
20
15
0
5
10
15
20
talk_time (bins)
Count of price_range by four_g and price_range
price_range ● 0 ● 1 ● 2 ● 3
Count of price_range
200
0
0
1
1
four_g
500 (25%)
500 (25%)
500 (25%)
500 (25%)
price_range
● 0
● 1
● 2
● 3
Count of battery_power (bins) by battery_power (bins)
Count of battery_power (bins)
20
10
0
0
500
1000
1500
2000
battery_power (bins)
Count of battery_power (bins) by battery_power (bins)
Count of battery_power (bins)
100
50
0
10
25
13
24
13
22
0
500
1000
1500
2
battery_power (bins)
Average of ram by price_range
Average of ram
3K
2K
1K
0K
0
1
2
3
price_range
Count of ram (bins) by ram (bins)
Count of ram (bins)
100
50
0
55
49
54
29
11
1
2
0
1000
2000
3000
4000
ram (bins)