**** **Industrial Project Report**

*Submitted in partial fulfillment of the degree of*

**B-tech in Computer Science And Engineering**

**By**

***Poulami Sarkar[11900121034]***

***Shreyasi Talukdar[11900121017]***

***Sanchita Purkait[11900121039]***

***Bhavya Sidharth Verma [11900121066]***

***Arpan dey[11900121018]***

***Adrash Anand[11900121061]***

**Second-year student of**

**SILIGURI INSTITUTE OF TECHNOLOGY**

*THIS IS SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF*

**AFFILIATED TO**

**Maulana Abul Kalam Azad University of Technology**

**Under the supervision of :-** Mr. Ripam Kundu

***PROJECT ON :- MOBILE PRICE CLASSIFIACTION WITH MACHINE LEARNING***

By

UNDER THE GUIDANCE OF

**Mr. Ripam Kundu**

**Project Guide**

**Sikharthy Infotech Pvt. Ltd.**

*THIS IS SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF*

**B.Tech**

IN

Computer Science And Engineering

**SILIGURI INSTITUTE OF TECHNOLOGY**

**AFFILIATED TO**

**Maulana Abul Kalam Azad University of Technology**

**Department of Computer science Engineering**

I hereby forward the documentation prepared under my supervision by **Ripam Kundu Sir** entitled **Siliguri Institute Of Technology** to be accepted as fulfillment of the requirement for the Degree of Bachelor of Technology in Computer Science and Engineering, **Siliguri Institute Of Technology** affiliated to **Maulana Abul Kalam Azad University of Technology** (**MAKAUT**).

|  |  |
| --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_    **Mr.Ripam Kundu**  **(Software Developer)**  **Project Guide**  **Sikharthy Infotech Pvt. Ltd.**  **Shilpi Ghosal**  **(Director)**  **Sikharthy Infotech Pvt. Ltd.** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_    **HOD**  **Department Of Computer Science And Engineering SIT** |

**TPO**

**Siliguri Institute of Technology**

**Certificate of Approval**

The foregoing project is hereby approved as a creditable study for the B.Tech in Computer Science And Engineering presented in a manner of satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorsed or approved any statement made, opinion expressed or conclusion therein but approve this project only for the purpose for which it is submitted.

Final Examination for

Evaluation of the Project ----------------------------------------

-----------------------------------------

------------------------------------------

**Signatures of Examiners**

**ABSTRACT**

Our mobile price classification project in machine learning aims to provide an intelligent system that can accurately classify mobile phones into different price ranges based on their features and specifications. Our project utilizes machine learning algorithms and techniques to analyze large datasets of mobile phones and their corresponding prices, enabling us to develop a model that can predict the price of a mobile phone based on its features with high accuracy.

**ACKNOWLEDGEMENT**

It is a great pleasure for me to acknowledge the assistance and participation of a large number of individuals in this attempt. Our project report has been structured under the valued suggestion, support, and guidance of **Mr. Ripam Kundu**. Under his guidance, we have accomplished the challenging task in a very short time.

Finally, we express our sincere thankfulness to our family members for inspiring me all throughout and always encouraging us.

**Group Member Signature**

**TABLE OF CONTENTS**

* Introduction 8
* Importing Dataset 10
* Data Profiling 11-15
* Data Pre-processing 15 -17
* Data Visualization 17-20
* Code & Screenshot 20 -27
* Conclusion 29

**INTRODUCTION**

 We will use [Python](https://www.geeksforgeeks.org/python-programming-language/) and its different libraries to complete the uber data analysis

**WHAT LIBRARIES WE USED**

## **Importing Libraries**

The analysis will be done using the following libraries :

* [Pandas](https://www.geeksforgeeks.org/python-pandas-dataframe/):  This library helps to load the data frame in a 2D array format and has multiple functions to perform analysis tasks in one go.
* [NumPy](https://www.geeksforgeeks.org/python-numpy/): NumPy arrays are very fast and can perform large computations in a very short time.
* [Matplotlib](https://www.geeksforgeeks.org/matplotlib-tutorial/) / [Seaborn](https://www.geeksforgeeks.org/introduction-to-seaborn-python/): This library is used to draw visualizations.

To importing all these libraries, we can use the  below code :

|  |
| --- |
| import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  import seaborn as sns  from sklearn.model\_selection import train\_test\_split  from sklearn import metrics  from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix  import warnings  warnings.filterwarnings("ignore") |

## **Importing Dataset**

After importing all the libraries , you can import the dataset using the pandas library.

INPUT :

|  |
| --- |
| data = pd.read\_csv('train.csv')  print(data)  **So after importing the datasets the output we get is : -** |

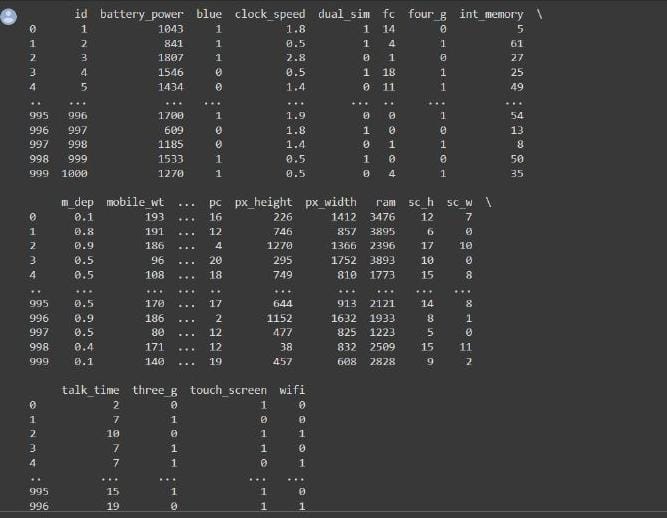
**TESTING DATA :**

**Input:**

test\_data = pd.read\_csv("test.csv”)

print(test\_data)

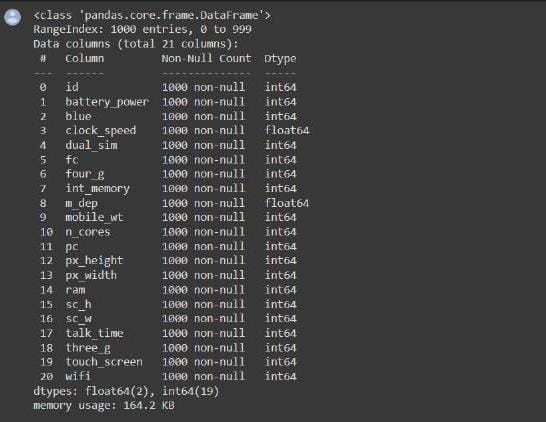
**OUTPUT:**



**INPUT:**

test\_data.info()

**OUTPUT:**

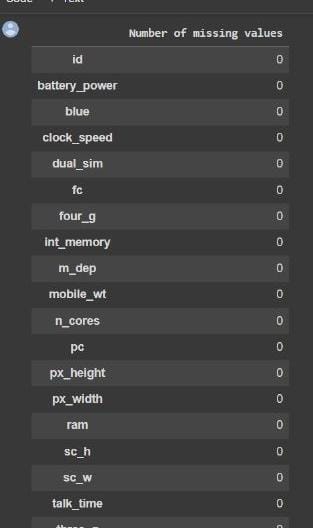


## MISSING VALUES FOR TEST DATA:

## Input:

pd.DataFrame(test\_data.isnull().sum(), columns= ['Number of missing values'])

**OUTPUT:**

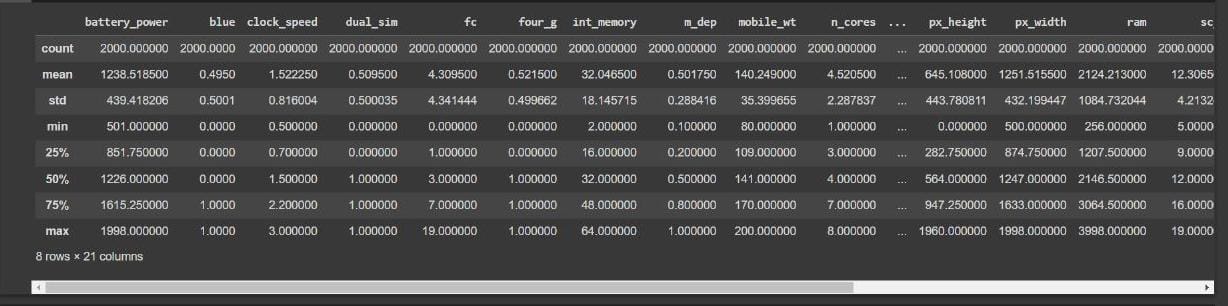


**TRAINING DATA:**

**INPUT:**

data.describe(include='all')

**OUTPUT:**

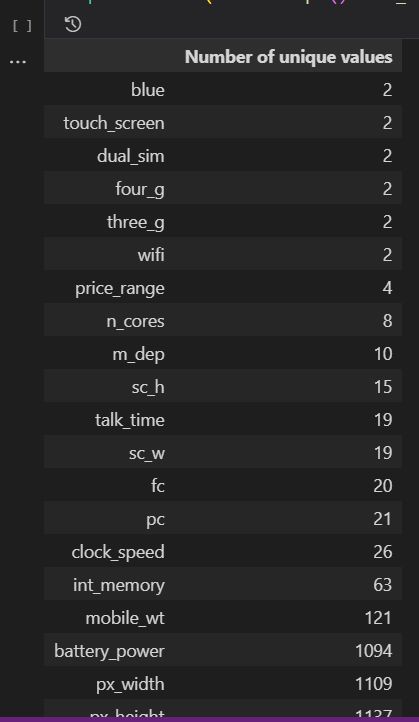


**MISSING VALUES FOR TRAIN DATA:**

**INPUT:**

pd.DataFrame(data.isnull().sum(), columns= ['Number of missing values'])

**OUTPUT:**



**CORRELATION OF ATTIRUBTUES:**

**INPUT:**

sns.set(style="white")

# Compute the correlation matrix

corr = data.corr()

# Generate a mask for the upper triangle

mask = np.zeros\_like(corr, dtype=np.bool)

mask[np.triu\_indices\_from(mask)] = True

# Set up the matplotlib figure

f, ax = plt.subplots(figsize=(27,18))

# Generate a custom diverging colormap

cmap = sns.diverging\_palette(15,963 ,as\_cmap=True)

# Draw the heatmap with the mask and correct aspect ratio

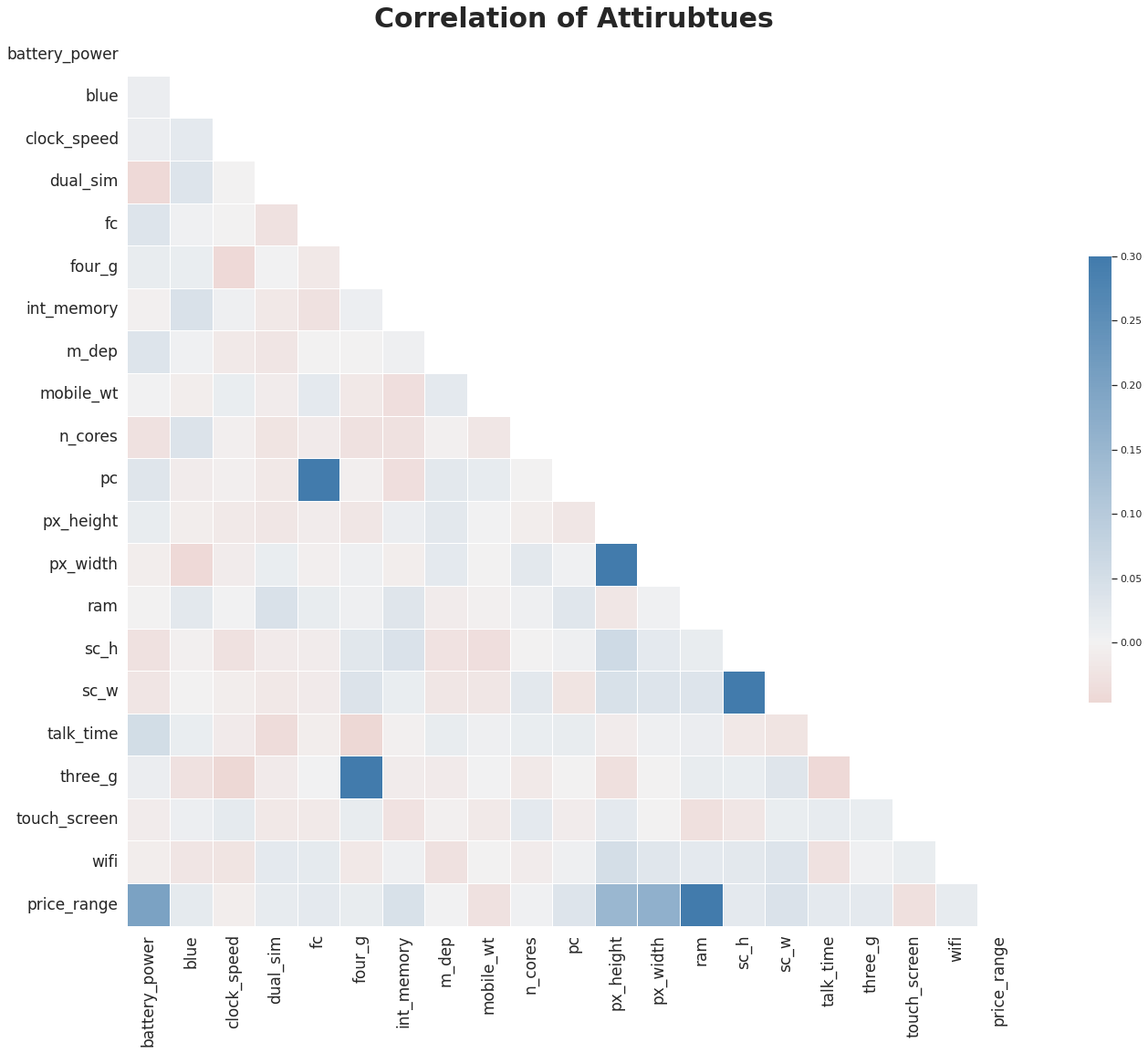
sns.heatmap(corr, mask=mask, cmap=cmap, vmax=.3, center=0,square=True, linewidths=.5, cbar\_kws={"shrink": .5})

plt.title("Correlation of Attirubtues",fontsize=30,fontweight='bold')

ax.xaxis.set\_tick\_params(labelsize=17)

ax.yaxis.set\_tick\_params(labelsize=17)

**OUTPUT:**

****

**CORRELATION BETWEEN ATTRIBUTES AND PRICE RANGE:**

**INPUT:**

corr = data.corr()

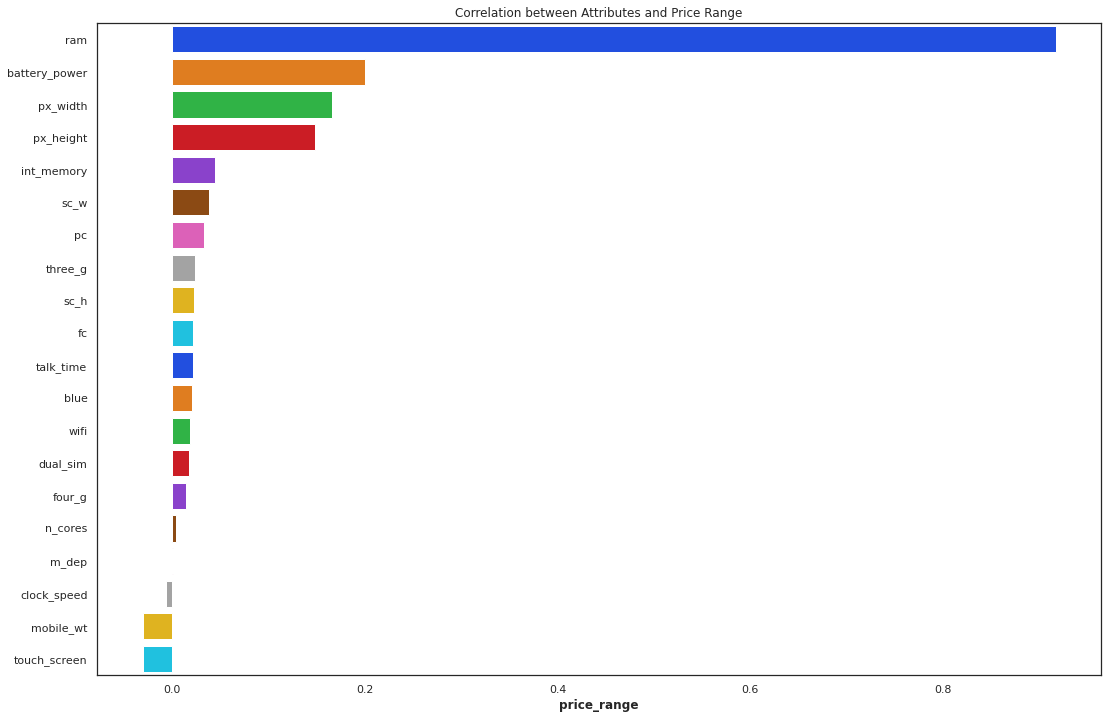
corr = corr.price\_range

cr = corr.sort\_values(ascending = False)[1:]

sns.barplot(x=cr, y=cr.index,palette = "bright")

plt.title("Correlation between Attributes and Price Range")

**OUTPUT:**

****

## **Data Visualization:**

Data visualization is the graphical representation of information and data. By using v[isual elements like charts, graphs, and maps](https://www.tableau.com/data-insights/reference-library/visual-analytics), data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. Additionally, it provides an excellent way for employees or business owners to present data to non-technical audiences without confusion**.**

In the world of Big Data, data visualization tools and technologies are essential to analyze massive amounts of information and make data-driven decisions.

**BATTERY POWER:**

**INPUT:**

sns.set(rc={"figure.dpi":100})

sns.set\_context('paper')

sns.set\_style("ticks")

fig = plt.figure(figsize=(15,12))

gs = fig.add\_gridspec(2, 2)

gs.update(wspace=0.3, hspace=0.4)

fig.text(0.085,0.95,'Battery power and its effect of the price range ', fontfamily='serif',fontsize=15, fontweight='bold')

sns.set\_palette('plasma')

ax0 = fig.add\_subplot(gs[0, 0])

ax1 = fig.add\_subplot(gs[0, 1])

ax2 = fig.add\_subplot(gs[1,0],ylim=(0, 3000),xlim=(0,5))

ax3 = fig.add\_subplot(gs[1,1],ylim=(0, 3000))

#Axis 0

ax0.grid(color='gray', linestyle=':', axis='y', zorder=0,  dashes=(1,5))

sns.kdeplot(x='battery\_power',

            data=data,

            shade=True,

            ax=ax0,

            linewidth = 0

            )

ax0.yaxis.set\_major\_formatter(mtick.PercentFormatter(1, decimals=3))

ax0.set\_xlabel("")

ax0.set\_ylabel("")

ax0.set\_title('Distribution',fontsize=12,fontfamily='serif',fontweight='bold',x=0,y=1)

ax0.spines['top'].set\_visible(False)

ax0.spines['right'].set\_visible(False)

ax0.spines['left'].set\_visible(False)

#Axis 1

ax1.grid(color='gray', linestyle=':', axis='y', zorder=0,  dashes=(1,5))

sns.kdeplot(x='battery\_power',

           hue='price\_range',

           shade=True,

           data=data,

           palette='plasma',

           ax=ax1,

           fill=True,

           alpha=.3,

           linewidth=0

           )

ax1.yaxis.set\_major\_formatter(mtick.PercentFormatter(1, decimals=3))

ax1.set\_xlabel("")

ax1.set\_ylabel("")

ax1.set\_title('Distribution by Price Range',fontsize=12,fontfamily='serif',fontweight='bold',x=0.1,y=1)

ax1.legend(['Low Cost ', 'Medium Cost','High Cost','Very High Cost'],fontsize=6,frameon=False)

ax1.spines['top'].set\_visible(False)

ax1.spines['right'].set\_visible(False)

ax1.spines['left'].set\_visible(False)

#Axis 2

ax2.grid(color='gray', linestyle=':', axis='y',  dashes=(1,5))

sns.boxenplot(y='battery\_power',

              data=data,

              ax=ax2,

              linewidth=0.4)

ax2.set\_xlabel("")

ax2.set\_ylabel("")

ax2.set\_title('Distribution',fontsize=12,fontfamily='serif',fontweight='bold',x=0,y=1.15)

ax2.spines['top'].set\_visible(False)

ax2.spines['right'].set\_visible(False)

ax2.spines['left'].set\_visible(False)

#Axis3

ax3.grid(color='gray', linestyle='-', axis='y', dashes=(1,5))

sns.boxenplot(x='price\_range',

              y='battery\_power',

              data=data,

              ax=ax3,

              linewidth=0.4

            )

ax3.set\_xlabel("")

ax3.set\_ylabel("")

ax3.set\_title('Distribution by Price Range',fontsize=12,fontfamily='serif',fontweight='bold',x=0.1,y=1.15)

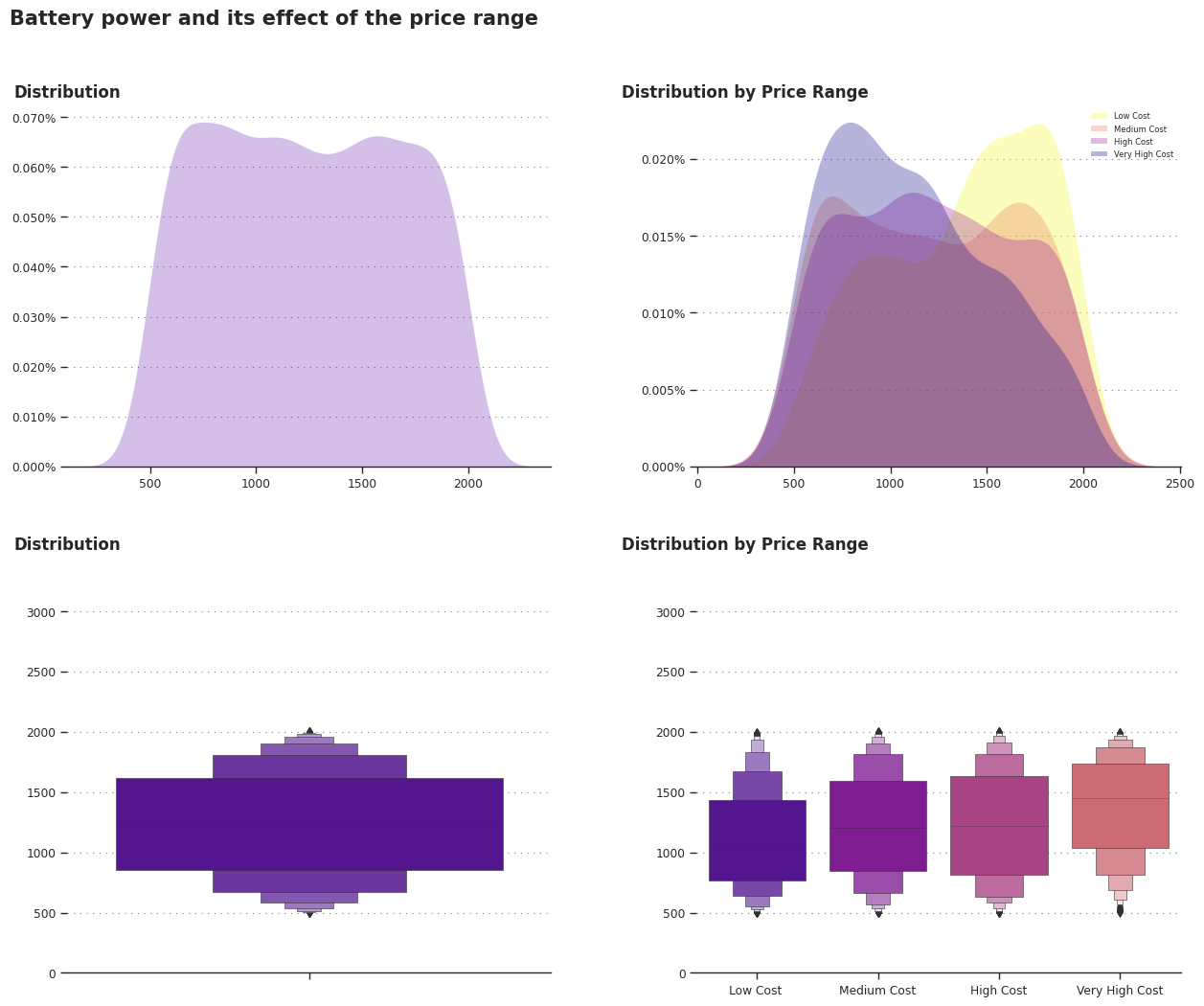
ax3.set\_xticklabels(['Low Cost ', 'Medium Cost','High Cost','Very High Cost'])

ax3.spines['top'].set\_visible(False)

ax3.spines['right'].set\_visible(False)

ax3.spines['left'].set\_visible(False)

fig.show()

**OUTPUT: **

**FRONT CAMERA MEGA PIXELS:**

**INPUT:**

fig = plt.figure(figsize=(15,12))

gs = fig.add\_gridspec(2, 2)

gs.update(wspace=0.3, hspace=0.4)

fig.text(0.085,0.95,'Front Camera mega-pixels and its effect on the price range', fontfamily='serif',fontsize=15, fontweight='bold')

sns.set\_palette('plasma')

ax0 = fig.add\_subplot(gs[0, 0])

ax1 = fig.add\_subplot(gs[0, 1])

ax2 = fig.add\_subplot(gs[1,0],ylim=(0,20))

ax3 = fig.add\_subplot(gs[1,1],ylim=(0,20))

#Axis 0

ax0.grid(color='gray', linestyle=':', axis='y', zorder=0,  dashes=(1,5))

sns.kdeplot(x='fc',

            data=data,

            shade=True,

            ax=ax0,

            linewidth = 0

            )

ax0.yaxis.set\_major\_formatter(mtick.PercentFormatter(1, decimals=3))

ax0.set\_xlabel("")

ax0.set\_ylabel("")

ax0.set\_title('Distribution',fontsize=12,fontfamily='serif',fontweight='bold',x=0,y=1.05)

ax0.spines['top'].set\_visible(False)

ax0.spines['right'].set\_visible(False)

ax0.spines['left'].set\_visible(False)

#Axis 1

ax1.grid(color='gray', linestyle=':', axis='y', zorder=0,  dashes=(1,5))

sns.kdeplot(x='fc',

           hue='price\_range',

           shade=True,

           data=data,

           palette='plasma',

           ax=ax1,

           fill=True,

           alpha=.3,

           linewidth=0

           )

ax1.yaxis.set\_major\_formatter(mtick.PercentFormatter(1, decimals=3))

ax1.set\_xlabel("")

ax1.set\_ylabel("")

ax1.set\_title('Distribution by Price Range',fontsize=12,fontfamily='serif',fontweight='bold',x=0.1,y=1.05)

ax1.legend(['Low Cost ', 'Medium Cost','High Cost','Very High Cost'],fontsize=6,frameon=False)

ax1.spines['top'].set\_visible(False)

ax1.spines['right'].set\_visible(False)

ax1.spines['left'].set\_visible(False)

#Axis 2

ax2.grid(color='gray', linestyle=':', axis='y',  dashes=(1,5))

sns.boxenplot(y='fc',

              data=data,

              ax=ax2,

              linewidth=0.4)

ax2.set\_xlabel("")

ax2.set\_ylabel("")

ax2.set\_title('Distribution',fontsize=12,fontfamily='serif',fontweight='bold',x=0,y=1.15)

ax2.spines['top'].set\_visible(False)

ax2.spines['right'].set\_visible(False)

ax2.spines['left'].set\_visible(False)

#Axis3

ax3.grid(color='gray', linestyle='-', axis='y',  dashes=(1,5))

sns.boxenplot(x='price\_range',

              y='fc',

              data=data,

              ax=ax3,

              linewidth=0.4

            )

ax3.set\_xlabel("")

ax3.set\_ylabel("")

ax3.set\_title('Distribution by Price Range',fontsize=12,fontfamily='serif',fontweight='bold',x=0.1,y=1.15)

ax3.set\_xticklabels(['Low Cost ', 'Medium Cost','High Cost','Very High Cost'])

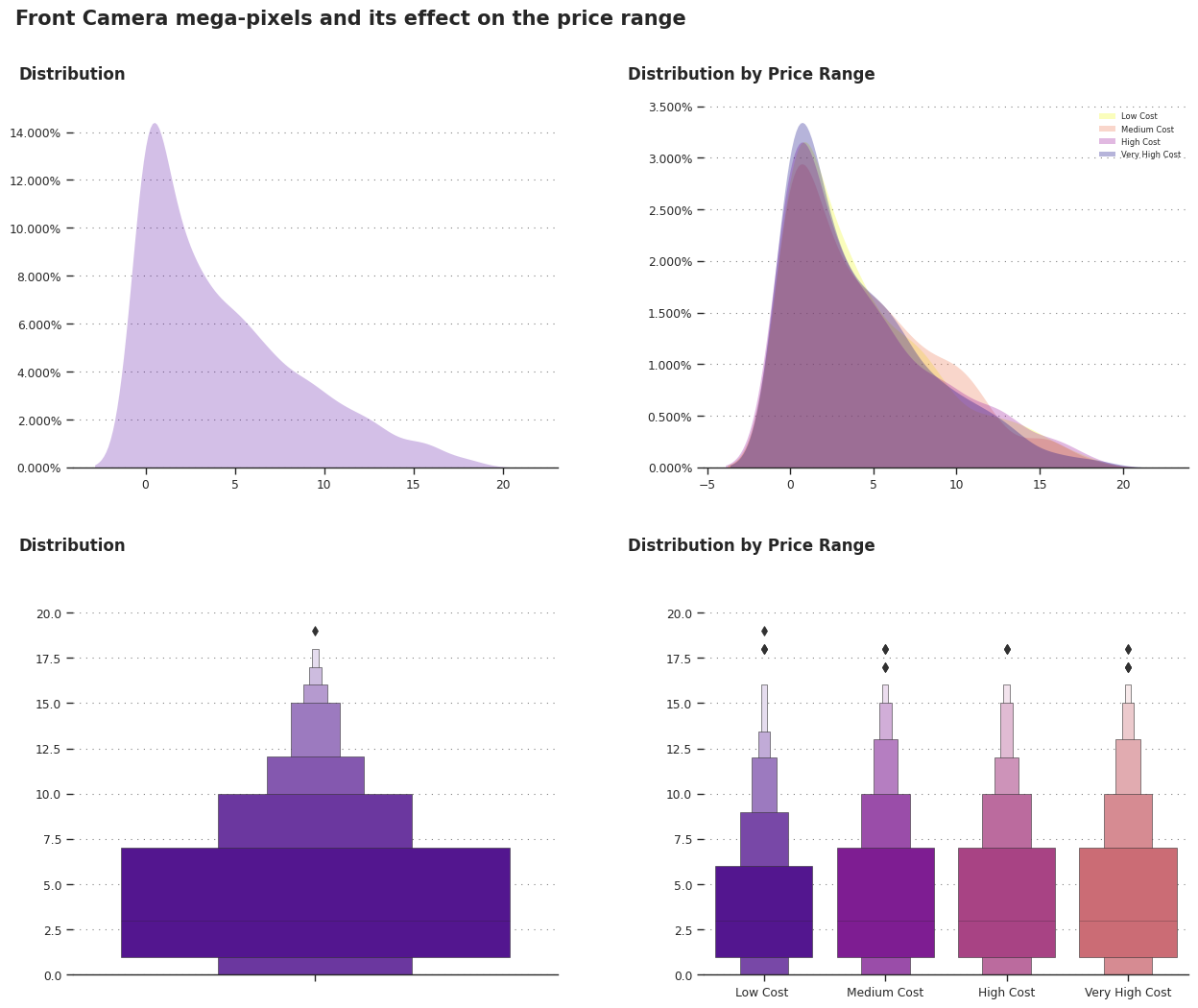
ax3.spines['top'].set\_visible(False)

ax3.spines['right'].set\_visible(False)

ax3.spines['left'].set\_visible(False)

fig.show()

**OUTPUT:**



**4G:**

**INPUT**:

fig = plt.figure(figsize=(15,4))

gs = fig.add\_gridspec(1, 2)

gs.update(wspace=0.3, hspace=0)

fig.text(0.120,1.1,'4G and its effect on the price ', fontfamily='serif',fontsize=15, fontweight='bold')

ax0 = fig.add\_subplot(gs[0, 0])

ax1 = fig.add\_subplot(gs[0, 1])

sns.countplot(x='four\_g',

           data=data,

            palette='bone',

           ax=ax0)

ax0.grid(color='gray', linestyle=':', axis='y', zorder=0,  dashes=(1,5))

ax0.set\_title('Distribution',fontsize=12,fontfamily='serif',fontweight='bold',x=0.1,y=1)

ax0.spines['top'].set\_visible(False)

ax0.spines['right'].set\_visible(False)

ax0.spines['left'].set\_visible(False)

ax0.set\_xticklabels(["Doesnt Support","Support"])

ax0.set\_xlabel("")

ax0.set\_ylabel("")

sns.countplot(x='four\_g',

             data=data,

             hue='price\_range',

             ax=ax1)

ax1.grid(color='gray', linestyle=':', axis='y', zorder=0,  dashes=(1,5))

ax1.set\_title('Distribution by Price Range',fontsize=12,fontfamily='serif',fontweight='bold',x=0.25,y=1)

ax1.spines['top'].set\_visible(False)

ax1.spines['right'].set\_visible(False)

ax1.spines['left'].set\_visible(False)

ax1.get\_legend().remove()

legend\_labels, \_= ax1.get\_legend\_handles\_labels()

ax1.legend(legend\_labels, ['Low Cost ', 'Medium Cost','High Cost','Very High Cost'], ncol=4, bbox\_to\_anchor=(-0.30, 1.22))

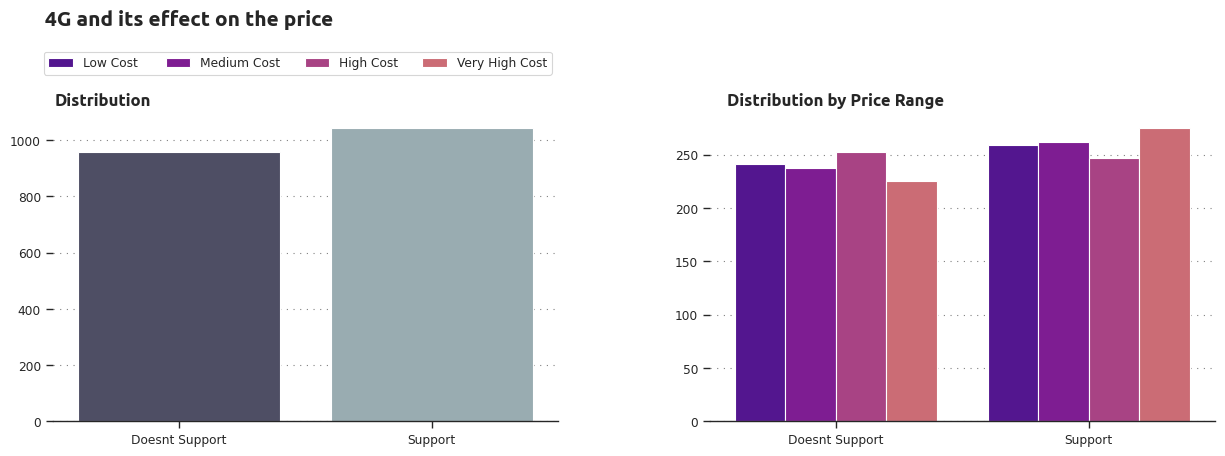
ax1.set\_xticklabels(["Doesnt Support","Support"])

ax1.set\_xlabel("")

ax1.set\_ylabel("")

fig.show()

OUTPUT:



**3G:**

**INPUT:**

fig = plt.figure(figsize=(15,4))

gs = fig.add\_gridspec(1, 2)

gs.update(wspace=0.3, hspace=0)

fig.text(0.120,1.1,'3G and its effect on the price ', fontfamily='serif',fontsize=15, fontweight='bold')

ax0 = fig.add\_subplot(gs[0, 0])

ax1 = fig.add\_subplot(gs[0, 1])

sns.countplot(x='three\_g',

           data=data,

            palette='bone',

           ax=ax0)

ax0.grid(color='gray', linestyle=':', axis='y', zorder=0,  dashes=(1,5))

ax0.set\_title('Distribution',fontsize=12,fontfamily='serif',fontweight='bold',x=0.1,y=1)

ax0.spines['top'].set\_visible(False)

ax0.spines['right'].set\_visible(False)

ax0.spines['left'].set\_visible(False)

ax0.set\_xticklabels(["Doesnt Support","Support"])

ax0.set\_xlabel("")

ax0.set\_ylabel("")

sns.countplot(x='three\_g',

             data=data,

             hue='price\_range',

             ax=ax1)

ax1.grid(color='gray', linestyle=':', axis='y', zorder=0,  dashes=(1,5))

ax1.set\_title('Distribution by Price Range',fontsize=12,fontfamily='serif',fontweight='bold',x=0.25,y=1)

ax1.spines['top'].set\_visible(False)

ax1.spines['right'].set\_visible(False)

ax1.spines['left'].set\_visible(False)

ax1.get\_legend().remove()

legend\_labels, \_= ax1.get\_legend\_handles\_labels()

ax1.legend(legend\_labels, ['Low Cost ', 'Medium Cost','High Cost','Very High Cost'], ncol=4, bbox\_to\_anchor=(-0.30, 1.22))

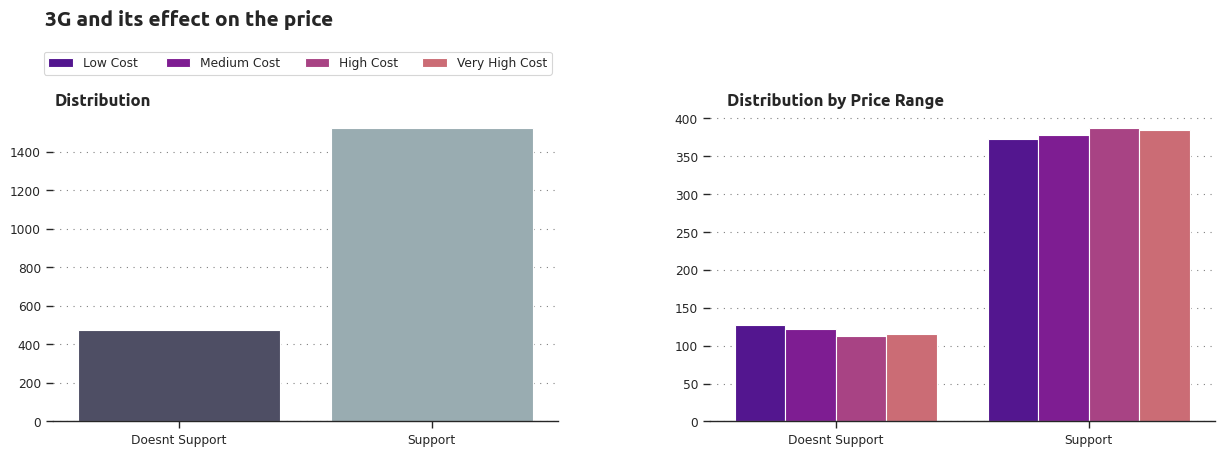
ax1.set\_xticklabels(["Doesnt Support","Support"])

ax1.set\_xlabel("")

ax1.set\_ylabel("")

fig.show()

**OUTPUT:**

****

**TOUCH SCREEN:**

**INPUT:**

fig = plt.figure(figsize=(15,4))

gs = fig.add\_gridspec(1, 2)

gs.update(wspace=0.3, hspace=0)

fig.text(0.120,1.1,'Touch Screen and its effect on the price ', fontfamily='serif',fontsize=15, fontweight='bold')

ax0 = fig.add\_subplot(gs[0, 0])

ax1 = fig.add\_subplot(gs[0, 1])

sns.countplot(x='touch\_screen',

           data=data,

            palette='bone',

           ax=ax0)

ax0.grid(color='gray', linestyle=':', axis='y', zorder=0,  dashes=(1,5))

ax0.set\_title('Distribution',fontsize=12,fontfamily='serif',fontweight='bold',x=0.1,y=1)

ax0.spines['top'].set\_visible(False)

ax0.spines['right'].set\_visible(False)

ax0.spines['left'].set\_visible(False)

ax0.set\_xticklabels(["Doesnt Support","Support"])

ax0.set\_xlabel("")

ax0.set\_ylabel("")

sns.countplot(x='touch\_screen',

             data=data,

             hue='price\_range',

             ax=ax1)

ax1.grid(color='gray', linestyle=':', axis='y', zorder=0,  dashes=(1,5))

ax1.set\_title('Distribution by Price Range',fontsize=12,fontfamily='serif',fontweight='bold',x=0.25,y=1)

ax1.spines['top'].set\_visible(False)

ax1.spines['right'].set\_visible(False)

ax1.spines['left'].set\_visible(False)

ax1.get\_legend().remove()

legend\_labels, \_= ax1.get\_legend\_handles\_labels()

ax1.legend(legend\_labels, ['Low Cost ', 'Medium Cost','High Cost','Very High Cost'], ncol=4, bbox\_to\_anchor=(-0.30, 1.22))

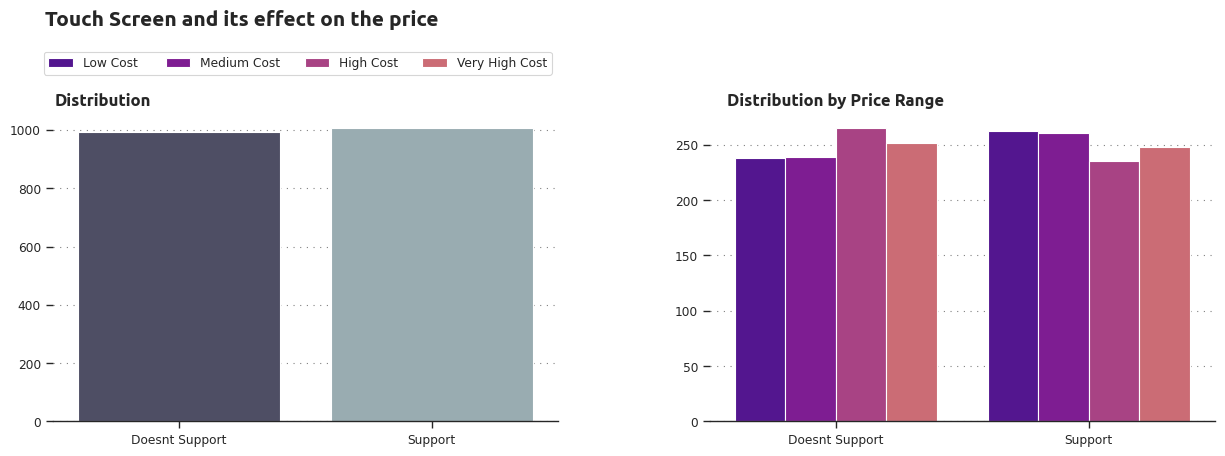
ax1.set\_xticklabels(["Doesnt Support","Support"])

ax1.set\_xlabel("")

ax1.set\_ylabel("")

fig.show()

**OUTPUT:**

****

**WIFI:**

**INPUT:**

fig = plt.figure(figsize=(15,4))

gs = fig.add\_gridspec(1, 2)

gs.update(wspace=0.3, hspace=0)

fig.text(0.120,1.1,'Wi-fi and its effect on the price ', fontfamily='serif',fontsize=15, fontweight='bold')

ax0 = fig.add\_subplot(gs[0, 0])

ax1 = fig.add\_subplot(gs[0, 1])

sns.countplot(x='wifi',

           data=data,

            palette='bone',

           ax=ax0)

ax0.grid(color='gray', linestyle=':', axis='y', zorder=0,  dashes=(1,5))

ax0.set\_title('Distribution',fontsize=12,fontfamily='serif',fontweight='bold',x=0.1,y=1)

ax0.spines['top'].set\_visible(False)

ax0.spines['right'].set\_visible(False)

ax0.spines['left'].set\_visible(False)

ax0.set\_xticklabels(["Doesnt Support","Support"])

ax0.set\_xlabel("")

ax0.set\_ylabel("")

sns.countplot(x='wifi',

             data=data,

             hue='price\_range',

             ax=ax1)

ax1.grid(color='gray', linestyle=':', axis='y', zorder=0,  dashes=(1,5))

ax1.set\_title('Distribution by Price Range',fontsize=12,fontfamily='serif',fontweight='bold',x=0.25,y=1)

ax1.spines['top'].set\_visible(False)

ax1.spines['right'].set\_visible(False)

ax1.spines['left'].set\_visible(False)

ax1.get\_legend().remove()

legend\_labels, \_= ax1.get\_legend\_handles\_labels()

ax1.legend(legend\_labels, ['Low Cost ', 'Medium Cost','High Cost','Very High Cost'], ncol=4, bbox\_to\_anchor=(-0.30, 1.22))

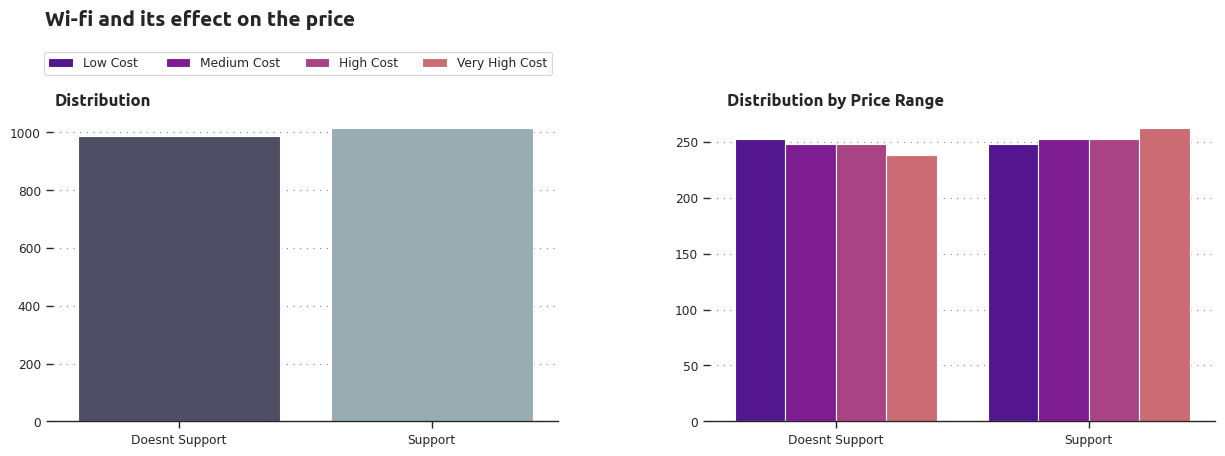
ax1.set\_xticklabels(["Doesnt Support","Support"])

ax1.set\_xlabel("")

ax1.set\_ylabel("")

fig.show()

OUTPUT:

****

**FREQUENCY DISTRIBUTION:**

**INPUT:**

fig, ax = plt.subplots(nrows=3, ncols=3, figsize=(15,10), dpi=100)

colors = ['#0055ff', '#ff7000', '#23bf00']

CustomPalette = sns.set\_palette(sns.color\_palette(colors))

for i in range(len(CatCols)):

    row = i//3

    col = i%3

    graph = sns.countplot(x=CatCols[i],hue=data['price\_range'],data=data, ax=ax[row,col])

    ax[row,col].set\_xlabel(CatCols[i], fontsize=15)

    ax[row,col].set\_ylabel('Count', fontsize=12)

    ax[row,col].set\_xticklabels(ax[row,col].get\_xticks())

    ax[row,col].grid(color='lightgrey')

    for j,p in enumerate(graph.patches):

        ax[row,col].annotate('{}'.format(p.get\_height()), (p.get\_x()+p.get\_width()/2, p.get\_height()+1),

                             ha='center', fontsize=10 ,fontweight="bold")

plt.suptitle('Frequency Distribution of Categorical Variables', fontsize=20)

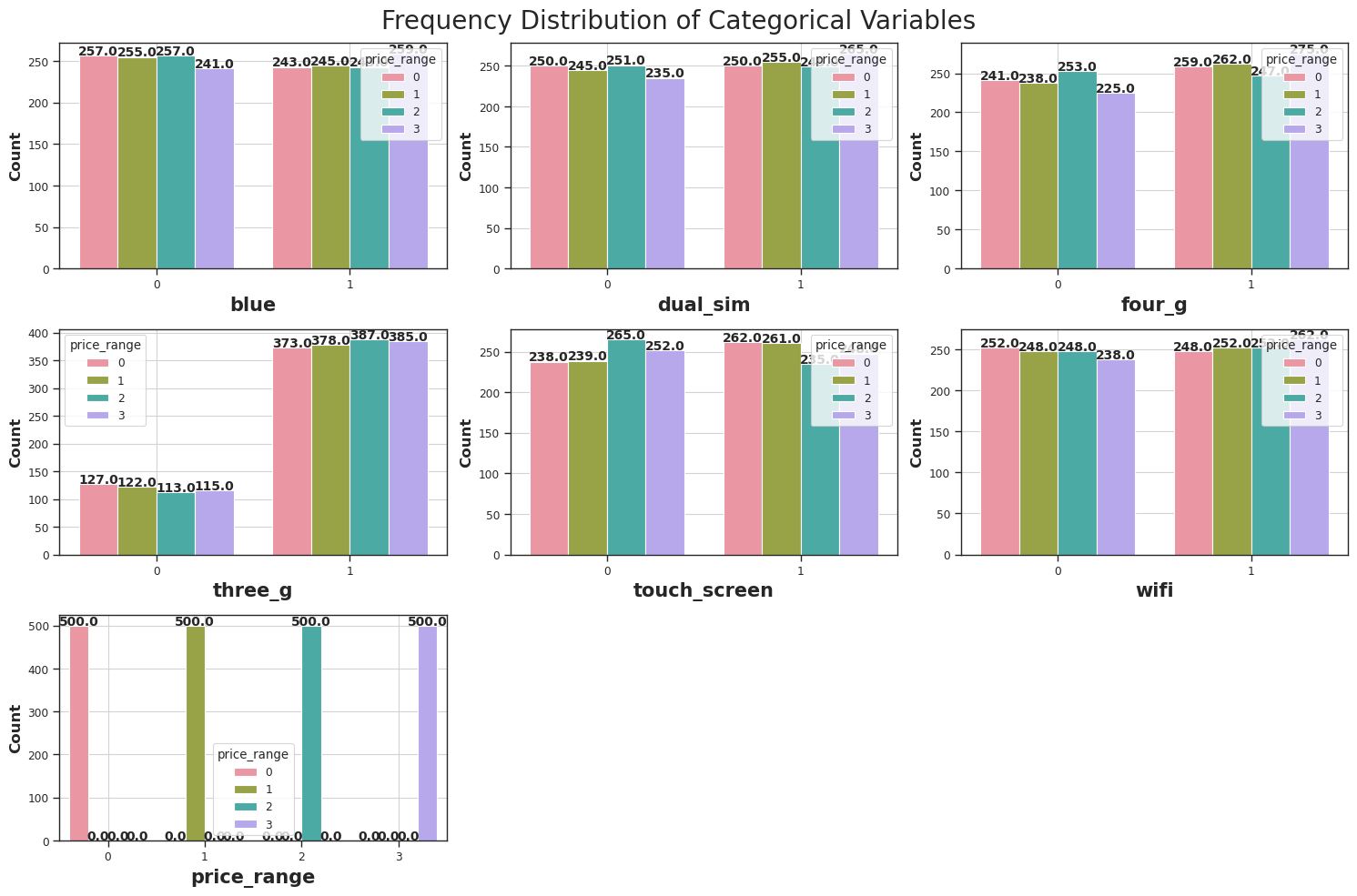
ax[-1,2].axis('off')

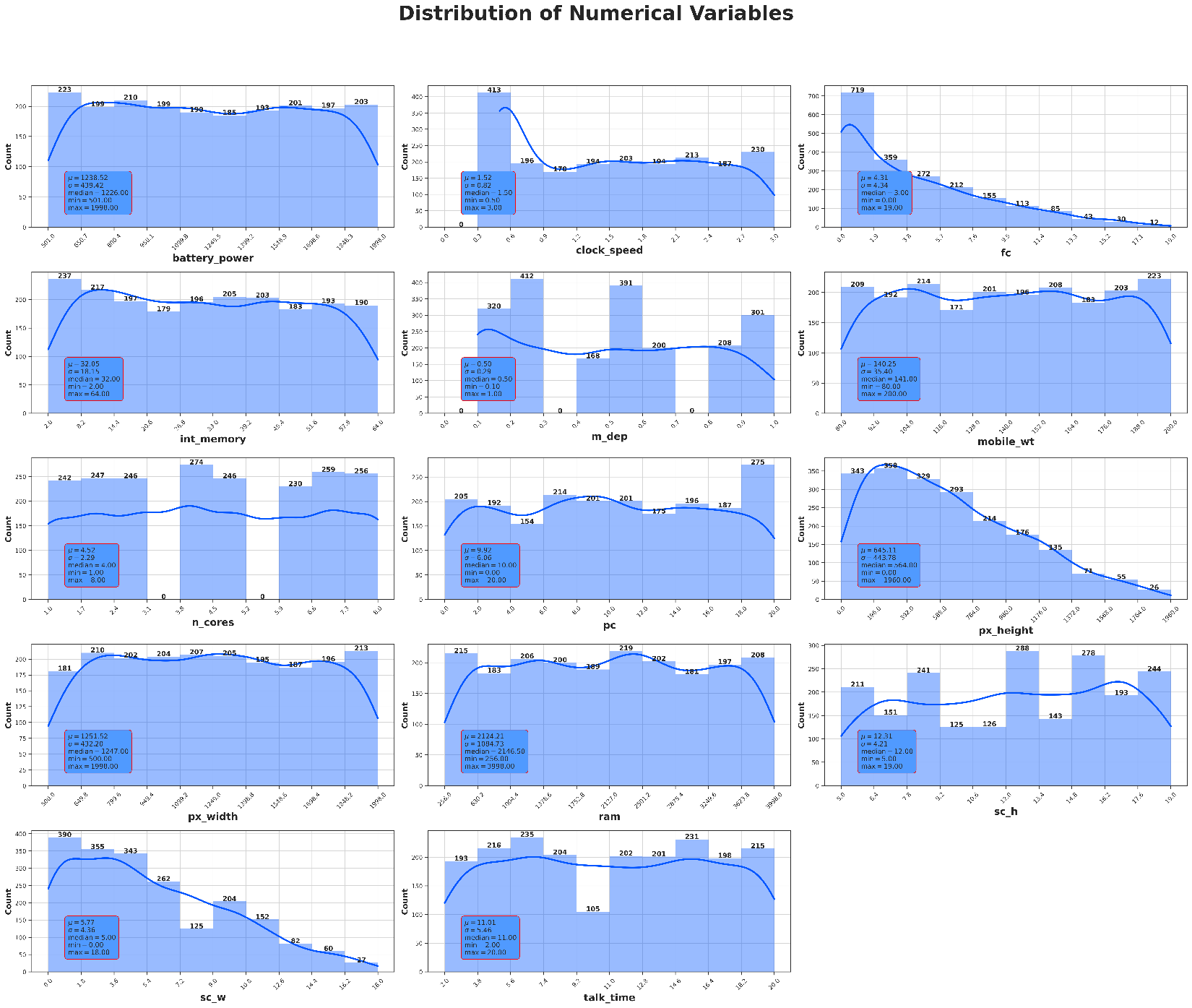
ax[-1,1].axis('off')

plt.tight\_layout()

plt.show()

OUTPUT:

****

****

**FUNCTIONAL REQUIREMENTS OF THE SYSTEM**

***SOFTWARE:***

* *Operating System*
* Windows OS 11

***WEB BROWSER:***

* Internet Explorer 7
* Google Chrome

***CODING LANGUAGE :***

* Python

**Conclusion :**

The project "Mobile Price Classification" involves predicting the price range of mobile phones based on their features. The dataset used in this project contains various features such as battery capacity, camera quality, and brand name, and their corresponding price ranges.

The project involves several steps such as data exploration, preprocessing, and model selection. In the data exploration step, various data visualization techniques are used to understand the relationship between different features and the price range of mobile phones. Preprocessing steps involve cleaning and transforming the data to make it suitable for training the model.

**REFERENCE:**

[**www.interview.projectideas**](http://www.interview.projectideas)

[**www.geesksforgeesks.com**](http://www.geesksforgeesks.com)

[**www.3Dsurfaceplot.com**](http://www.3Dsurfaceplot.com)