Home | EDA_prj

localhost:8888/notebooks/EDA_prj.ipynb?

jupyter EDA_prj Last Checkpoint: 21 hours ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab | Python 3 (ipykernel)

[1]:
```
import pandas as pd
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
train_df=pd.read_csv("train.csv")
train_df.head()
```

[1]:

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touc
	842	0	2.2	0	1	0	7	0.6	188	2	...	20	756	2549	9	7	19	0	
	1021	1	0.5	1	0	1	53	0.7	136	3	...	905	1988	2631	17	3	7	1	
	563	1	0.5	1	2	1	41	0.9	145	5	...	1263	1716	2603	11	2	9	1	
	615	1	2.5	0	0	0	10	0.8	131	6	...	1216	1786	2769	16	8	11	1	
	1821	1	1.2	0	13	1	44	0.6	141	2	...	1208	1212	1411	8	2	15	1	

rows × 21 columns

[3]:
```
train_df.tail()
```

[3]:

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g
1995	794	1	0.5	1	0	1	2	0.8	106	6	...	1222	1890	668	13	4	19	1
1996	1965	1	2.6	1	0	0	39	0.2	187	4	...	915	1965	2032	11	10	16	1
1997	1911	0	0.9	1	1	1	36	0.7	108	8	...	868	1632	3057	9	1	5	1
1998	1512	0	0.9	0	4	1	46	0.1	145	5	...	336	670	869	18	10	19	1
1999	510	1	2.0	1	5	1	45	0.9	168	6	...	483	754	3919	19	4	2	1

5 rows × 21 columns

[8]:
```
train_df.shape
```

[8]:
```
(2000, 21)
```

[62]:
```
train_df.describe()
```

Home | EDA_prj

localhost:8888/notebooks/EDA_prj.ipynb?

jupyter EDA_prj Last Checkpoint: 21 hours ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

[62]:
```
train_df.describe()
```

[62]:

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	px_height	px_width
count	2000.000000	2000.0000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	...	2000.000000	2000.000000
mean	1238.518500	0.4950	1.522250	0.509500	4.309500	0.521500	32.046500	0.501750	140.249000	4.520500	...	645.108000	1251.515500
std	439.418206	0.5001	0.816004	0.500035	4.341444	0.499662	18.145715	0.288416	35.399655	2.287837	...	443.780811	432.199447
min	501.000000	0.0000	0.500000	0.000000	0.000000	0.000000	2.000000	0.100000	80.000000	1.000000	...	0.000000	500.000000
25%	851.750000	0.0000	0.700000	0.000000	1.000000	0.000000	16.000000	0.200000	109.000000	3.000000	...	282.750000	874.750000
50%	1226.000000	0.0000	1.500000	1.000000	3.000000	1.000000	32.000000	0.500000	141.000000	4.000000	...	564.000000	1247.000000
75%	1615.250000	1.0000	2.200000	1.000000	7.000000	1.000000	48.000000	0.800000	170.000000	7.000000	...	947.250000	1633.000000
max	1998.000000	1.0000	3.000000	1.000000	19.000000	1.000000	64.000000	1.000000	200.000000	8.000000	...	1960.000000	1998.000000

8 rows × 21 columns

[97]:
```
train_df.isnull().sum()
```

[97]:
```
battery_power    0
blue             0
clock_speed      0
dual_sim         0
fc               0
four_g           0
int_memory       0
m_dep            0
mobile_wt        0
n_cores          0
pc               0
px_height        0
px_width         0
ram              0
sc_h             0
sc_w             0
talk_time        0
three_g          0
touch_screen     0
wifi             0
price_range      0
dtype: int64
```

Home | EDA_prj

localhost:8888/notebooks/EDA_prj.ipynb?

jupyter EDA_prj Last Checkpoint: 21 hours ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab | Python 3 (ipykernel)

[10]:
```
train_df.info()
```

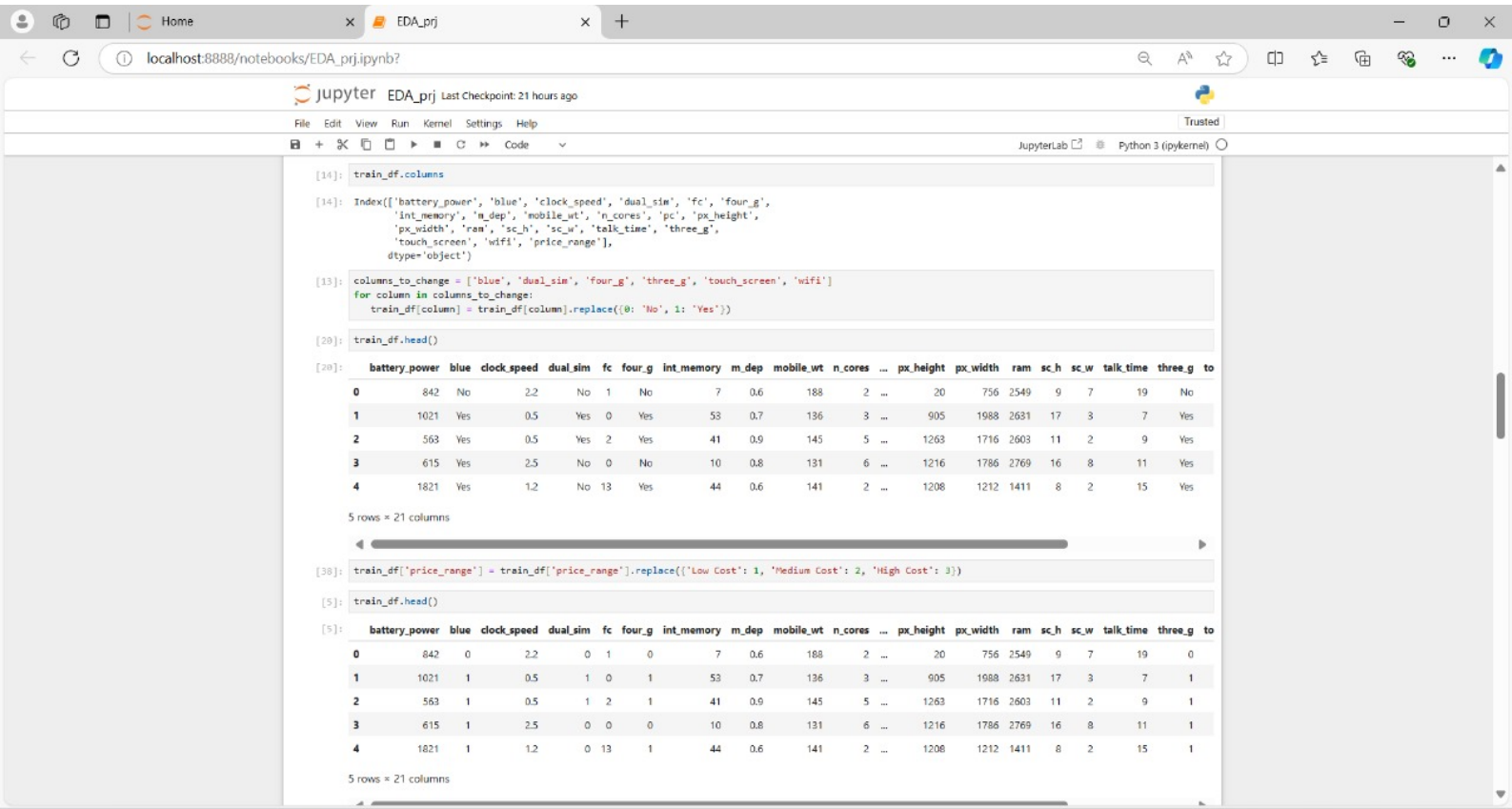
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   battery_power  2000 non-null   int64  
 1   blue           2000 non-null   int64  
 2   clock_speed    2000 non-null   float64
 3   dual_sim       2000 non-null   int64  
 4   fc             2000 non-null   int64  
 5   four_g         2000 non-null   int64  
 6   int_memory     2000 non-null   int64  
 7   m_dep          2000 non-null   float64
 8   mobile_wt      2000 non-null   int64  
 9   n_cores        2000 non-null   int64  
 10  pc             2000 non-null   int64  
 11  px_height      2000 non-null   int64  
 12  px_width       2000 non-null   int64  
 13  ram            2000 non-null   int64  
 14  sc_h           2000 non-null   int64  
 15  sc_w           2000 non-null   int64  
 16  talk_time      2000 non-null   int64  
 17  three_g        2000 non-null   int64  
 18  touch_screen   2000 non-null   int64  
 19  wifi           2000 non-null   int64  
 20  price_range    2000 non-null   int64  
dtypes: float64(2), int64(19)
memory usage: 328.3 KB
```

[12]:
```
train_df=pd.read_csv('train.csv',encoding='latin-1')
train_df.head()
```

[12]:

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	to
0	842	0	2.2	0	1	0	7	0.6	188	2	...	20	756	2549	9	7	19	0	
1	1021	1	0.5	1	0	1	53	0.7	136	3	...	905	1988	2631	17	3	7	1	
2	563	1	0.5	1	2	1	41	0.9	145	5	...	1263	1716	2603	11	2	9	1	
3	615	1	2.5	0	0	0	10	0.8	131	6	...	1216	1786	2769	16	8	11	1	
4	1821	1	1.2	0	13	1	44	0.6	141	2	...	1208	1212	1411	8	2	15	1	

5 rows × 21 columns


Home
EDA_prj
localhost:8888/notebooks/EDA_prj.ipynb?
jupyter EDA_prj Last Checkpoint: 21 hours ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[14]: train_df.columns
[14]: Index(['battery_power', 'blue', 'clock_speed', 'dual_sim', 'fc', 'four_g',
'int_memory', 'm_dep', 'mobile_wt', 'n_cores', 'pc', 'px_height',
'px_width', 'ram', 'sc_h', 'sc_w', 'talk_time', 'three_g',
'touch_screen', 'wifi', 'price_range'],
dtype='object')
[13]: columns_to_change = ['blue', 'dual_sim', 'four_g', 'three_g', 'touch_screen', 'wifi']
for column in columns_to_change:
train_df[column] = train_df[column].replace({0: 'No', 1: 'Yes'})
[20]: train_df.head()
[20]:
battery_power blue clock_speed dual_sim fc four_g int_memory m_dep mobile_wt n_cores ... px_height px_width ram sc_h sc_w talk_time three_g to
0 842 No 2.2 No 1 No 7 0.6 188 2 ... 20 756 2549 9 7 19 No
1 1021 Yes 0.5 Yes 0 Yes 53 0.7 136 3 ... 905 1988 2631 17 3 7 Yes
2 563 Yes 0.5 Yes 2 Yes 41 0.9 145 5 ... 1263 1716 2603 11 2 9 Yes
3 615 Yes 2.5 No 0 No 10 0.8 131 6 ... 1216 1786 2769 16 8 11 Yes
4 1821 Yes 1.2 No 13 Yes 44 0.6 141 2 ... 1208 1212 1411 8 2 15 Yes
5 rows × 21 columns
[38]: train_df['price_range'] = train_df['price_range'].replace({'Low Cost': 1, 'Medium Cost': 2, 'High Cost': 3})
[5]: train_df.head()
[5]:
battery_power blue clock_speed dual_sim fc four_g int_memory m_dep mobile_wt n_cores ... px_height px_width ram sc_h sc_w talk_time three_g to
0 842 0 2.2 0 1 0 7 0.6 188 2 ... 20 756 2549 9 7 19 0
1 1021 1 0.5 1 0 1 53 0.7 136 3 ... 905 1988 2631 17 3 7 1
2 563 1 0.5 1 2 1 41 0.9 145 5 ... 1263 1716 2603 11 2 9 1
3 615 1 2.5 0 0 0 10 0.8 131 6 ... 1216 1786 2769 16 8 11 1
4 1821 1 1.2 0 13 1 44 0.6 141 2 ... 1208 1212 1411 8 2 15 1
5 rows × 21 columns

Home | EDA_prj

localhost:8888/notebooks/EDA_prj.ipynb?

jupyter EDA_prj Last Checkpoint: 21 hours ago

File Edit View Run Kernel Settings Help

Trusted

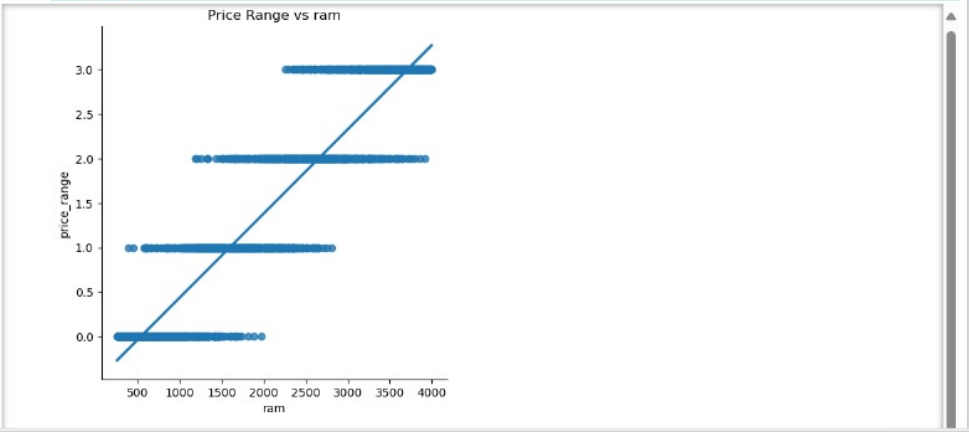
Code

JupyterLab Python 3 (ipykernel)

*[17]:
```
import seaborn as sns
import matplotlib.pyplot as plt
for column in columns_to_change:
    train_df[column] = train_df[column].replace({'No': 0, 'Yes': 1})

# Assuming 'train_df' is your dataset
# Calculate the correlation matrix
corr_matrix = train_df.corr()
price_range_corr = corr_matrix['price_range'].sort_values(ascending=False)

# Iterate over variables and create lmplots
for column in price_range_corr.index:
  if column != 'price_range':  # Exclude plotting price_range against itself
    sns.lmplot(x=column, y='price_range', data=train_df)
    plt.title(f'Price Range vs {column}')
    plt.show()
```



Home | EDA_prj

localhost:8888/notebooks/EDA_prj.ipynb?

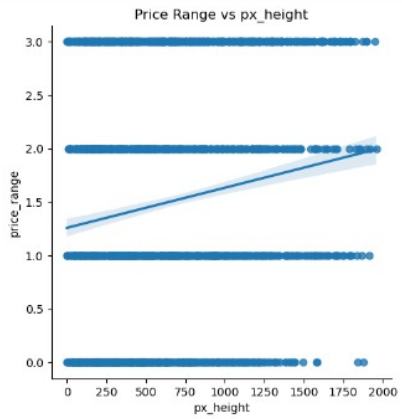
jupyter EDA_prj Last Checkpoint: 21 hours ago

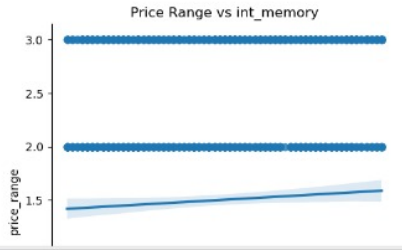
File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)





Home | EDA_prj

localhost:8888/notebooks/EDA_prj.ipynb?

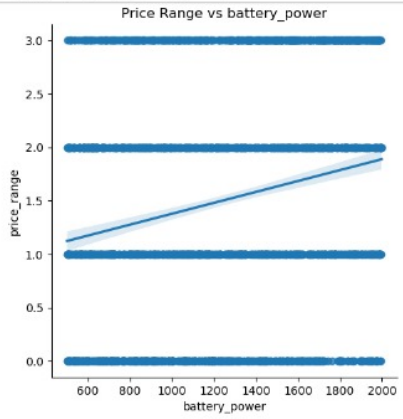
jupyter EDA_prj Last Checkpoint: 21 hours ago

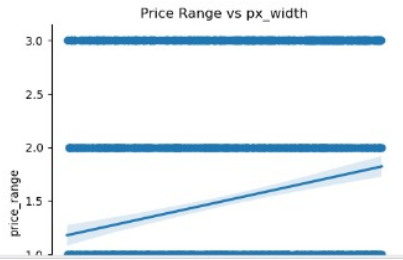
File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)





New tab | Esakkiraj_eda_prj.ipynb - Colab | Home | mobile_price

localhost:8888/notebooks/mobile_price.ipynb?

jupyter mobile_price Last Checkpoint: 23 hours ago

File Edit View Run Kernel Settings Help

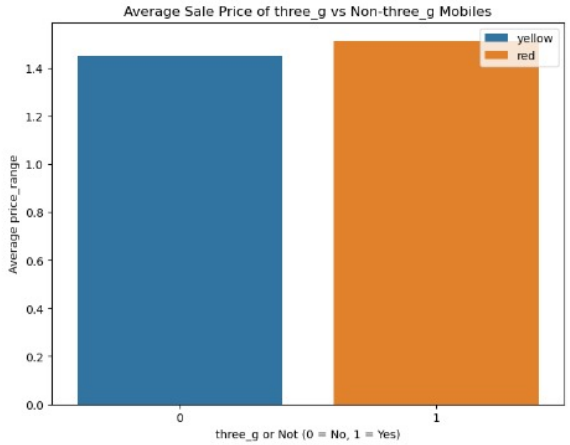
Trusted

Code

JupyterLab Python 3 (ipykernel)

[21]:
```
plt.figure(figsize=(8, 6))
sns.barplot(x='three_g', y='price_range', data=grouped_data, hue=['yellow','red'])

# Adding labels and title
plt.xlabel('three_g or Not (0 = No, 1 = Yes)')
plt.ylabel('Average price_range')
plt.title('Average Sale Price of three_g vs Non-three_g Mobiles')
plt.show()
```



[76]:
```
# Assuming train_df is your dataset and 'bluetooth' and 'price_range' are the relevant columns
plt.figure(figsize=(8, 6))

# Create a count plot
```

Home | EDA_prj

localhost:8888/notebooks/EDA_prj.ipynb?

jupyter EDA_prj Last Checkpoint: 21 hours ago

File Edit View Run Kernel Settings Help

Trusted

Code

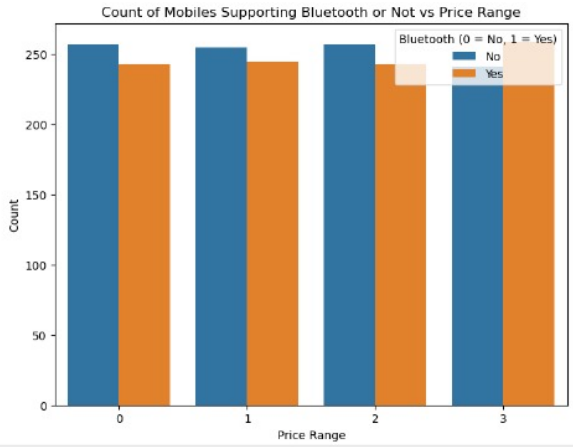
JupyterLab Python 3 (ipykernel)

[76]:
```
# Assuming train_df is your dataset and 'bluetooth' and 'price_range' are the relevant columns
plt.figure(figsize=(8, 6))

# Create a count plot
sns.countplot(x='price_range', hue='blue', data=train_df)

# Adding labels and title
plt.xlabel('Price Range')
plt.ylabel('Count')
plt.title('Count of Mobiles Supporting Bluetooth or Not vs Price Range')

# Display the plot
plt.legend(title='Bluetooth (0 = No, 1 = Yes)')
plt.show()
```



Home | EDA_prj

localhost:8888/notebooks/EDA_prj.ipynb?

jupyter EDA_prj Last Checkpoint: 21 hours ago

File Edit View Run Kernel Settings Help

Trusted

Code | JupyterLab | Python 3 (ipykernel)

[78]:
```
# Assuming 'pixel_resolution_height', 'pixel_resolution_width', and 'price_range' are columns in train_df
plt.figure(figsize=(8, 6))

# Create a scatterplot
sns.scatterplot(x='px_width', y='px_height', hue='price_range', data=train_df, palette='viridis')

# Adding labels and title
plt.xlabel('Pixel Resolution Width')
plt.ylabel('Pixel Resolution Height')
plt.title('Relation of Pixel Resolution (Height vs Width) with Price Range')

# Display the plot
plt.legend(title='Price Range')
plt.show()
```

Home | EDA_prj

localhost:8888/notebooks/EDA_prj.ipynb?

jupyter EDA_prj Last Checkpoint: 21 hours ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab | Python 3 (ipykernel)

[80]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Assuming 'screen_height', 'screen_width', and 'price_range' are columns in train_df
plt.figure(figsize=(8, 6))
# Create a scatterplot
sns.scatterplot(x='sc_w', y='sc_h', hue='price_range', data=train_df, palette='coolwarm')
# Adding labels and title
plt.xlabel('Screen Width')
plt.ylabel('Screen Height')
plt.title('Relation of Screen Height and Screen Width with Price Range')
# Display the plot
plt.legend(title='Price Range')
plt.show()
```