PYTHON FOR DATA SCIENCE

PROJECT REPORT

EXPLORATORY DATA ANALYSIS(EDA) AND MODELING ON LAPTOP DATASET

Instructor: Dr. SREEJA SR

TEAM DETAILS:

NITYA N S S	S20200010143
GURRAM YASHASWI	S20200010073

INDEX

1. DATASET

The dataset is the laptop listings on Rukminim1, an ecommerce website in India. This is an unprocessed dataset. It contains 984 unique entries. This unprocessed dataset contains 12 features

Features of dataset

1. img_link : image link for the product
2. name : name of the laptop with version
3. price(in Rs.): price of the product
4. processor: processor present in product
5. ram: ram size in the product
6. os: operating system of the product
7. storage: max storage of the product
8. display(in inch): size of the display in inches
9. Rating: the overall rating given to the product
10. no_of_ratings: total number of ratings given to the product
11. no_of reviews: total number of review given to the product

2. EXPLORATORY DATA ANALYSIS

PREPROCESSING

2.1 LOAD DATASET

```
data = pd.read_csv('laptops.csv', index_col=0)
data.head()
```

	img_link	name	price(in Rs.)	processor	ram	os	storage	display(in inch)	rating	no_of_ratings	no_of_reviews
0	https://rukminim1.flixcart.com/image/312/312/x...	Lenovo Intel Core i5 11th Gen	62990	Intel Core i5 Processor (11th Gen)	16 GB DDR4 RAM	Windows 11 Operating System	512 GB SSD	15.6	4.5	14.0	1.0
1	https://rukminim1.flixcart.com/image/312/312/x...	Lenovo V15 G2 Core i3 11th Gen	37500	Intel Core i3 Processor (11th Gen)	8 GB DDR4 RAM	64 bit Windows 11 Operating System	1 TB HDD|256 GB SSD	15.6	4.4	53.0	3.0
2	https://rukminim1.flixcart.com/image/312/312/l...	ASUS TUF Gaming F15 Core i5 10th Gen	49990	Intel Core i5 Processor (10th Gen)	8 GB DDR4 RAM	Windows 11 Operating System	512 GB SSD	15.6	4.4	4733.0	463.0

2.2 Describe Data

Describing data by count, mean, standard deviation, min, 25%, 50%, 75%, max for numerical data

```
data.describe()
```

	price(in Rs.)	display(in inch)	rating	no_of_ratings	no_of_reviews
count	984.000000	984.000000	688.000000	688.000000	688.000000
mean	80960.720528	15.148374	4.284884	718.091570	83.898256
std	57421.220919	1.332078	0.330239	1750.817825	211.596726
min	15990.000000	11.600000	1.600000	1.000000	0.000000
25%	42655.000000	14.000000	4.100000	14.000000	2.000000
50%	62990.000000	15.600000	4.300000	90.000000	11.000000
75%	94990.000000	15.600000	4.500000	453.000000	53.500000
max	419990.000000	35.000000	5.000000	15492.000000	2054.000000

For Categorical data

We can describe by count, frequency, unique, top

```
data.describe(include=object)
```

	img_link	name	processor	ram	os	storage
count	984	984	984	984	984	984
unique	584	506	59	22	11	16
top	https://rukminim1.flixcart.com/image/312/312/x...	Lenovo Intel Core i5 11th Gen	Intel Core i3 Processor (11th Gen)	8 GB DDR4 RAM	64 bit Windows 11 Operating System	512 GB SSD
freq	41	43	131	463	527	575

2.3 Info of the data

Info give data type and count of non null data of the each feature

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 984 entries, 0 to 983
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   img_link          984 non-null    object 
 1   name              984 non-null    object 
 2   price(in Rs.)     984 non-null    int64  
 3   processor         984 non-null    object 
 4   ram               984 non-null    object 
 5   os                984 non-null    object 
 6   storage           984 non-null    object 
 7   display(in inch)  984 non-null    float64
 8   rating            688 non-null    float64
 9   no_of_ratings     688 non-null    float64
 10  no_of_reviews     688 non-null    float64
dtypes: float64(4), int64(1), object(6)
memory usage: 92.2+ KB
```

We observe there are null values in rating, no_of_ratings, no_of_reviews

2.4 Null values

```
data.isna().sum()
```

```
img_link              0
name                  0
price(in Rs.)         0
processor             0
ram                   0
os                    0
storage               0
display(in inch)      0
rating              296
no_of_ratings       296
no_of_reviews       296
dtype: int64
```

296 null values in each of the three

FILLING THE NULL VALUES

```
data = data.fillna(0)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 984 entries, 0 to 983
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   img_link          984 non-null    object 
 1   name              984 non-null    object 
 2   price(in Rs.)     984 non-null    int64  
 3   processor         984 non-null    object 
 4   ram               984 non-null    object 
 5   os                984 non-null    object 
 6   storage           984 non-null    object 
 7   display(in inch)  984 non-null    float64
 8   rating            984 non-null    float64
 9   no_of_ratings     984 non-null    float64
 10  no_of_reviews     984 non-null    float64
dtypes: float64(4), int64(1), object(6)
memory usage: 92.2+ KB
```

2.5 Plots

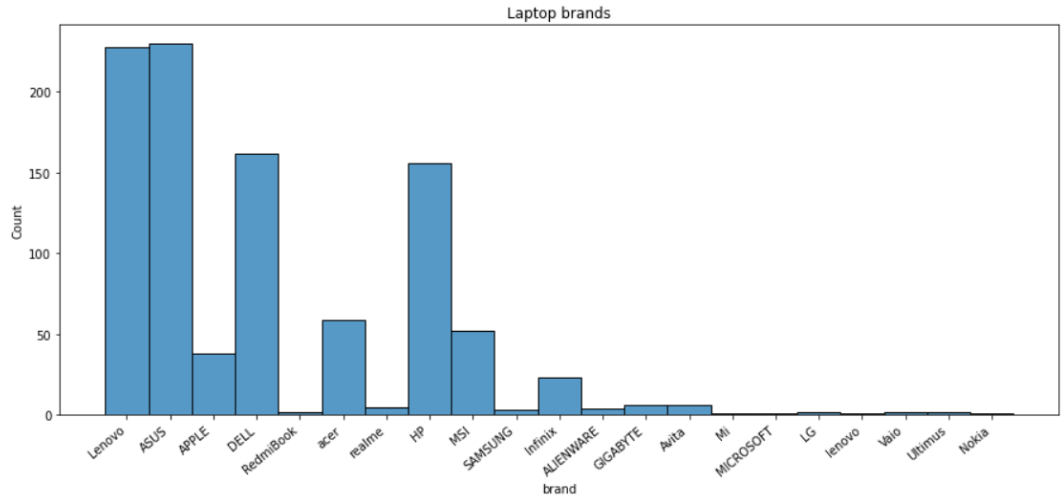
Plots are used for the better visualization and analysis of the data

We have different way to visualize the data we mainly use histograms, scatter plots, pair plot for the data analysis

2.5.1 *Histograms or Hist plots*

Plotting the counts of each brand taken into account

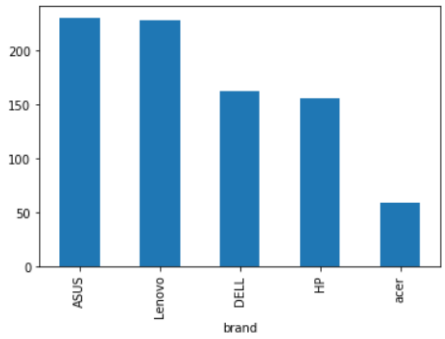
```
plt.figure(figsize=(15,6))
sns.histplot(data, x = 'brand')
plt.title('Laptop brands')
plt.xticks(rotation=40, horizontalalignment='right',
           fontsize=10)
plt.show()
```



- From here we can observe data is not balanced
- Ploting the top brands

```
data.groupby('brand').size().sort_values(ascending=False).head().plot(kind='bar')
```

```
<AxesSubplot:xlabel='brand'>
```



We have ASUS, Lenovo, DELL, HP and acer as top 5 brands

Using query to find min and max price is for which product

```
data.query('price == price.max()')
```

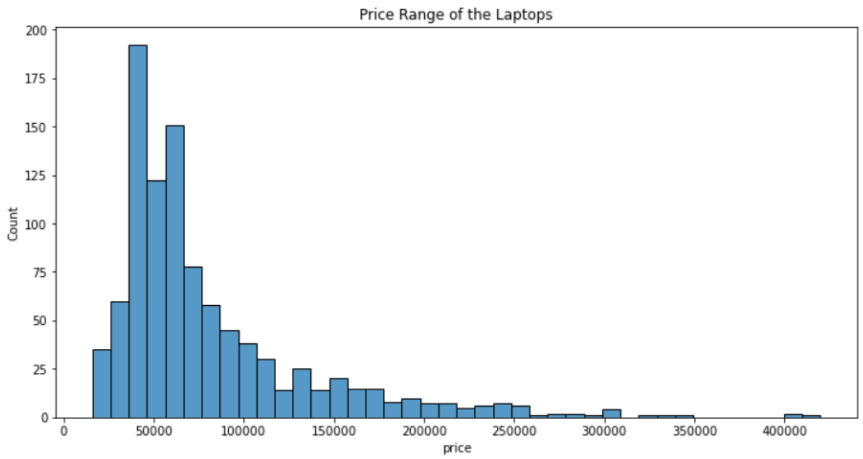
	img_link	name	price	processor	ram	os	storage	display	rating	no_of_ratings	no_of_reviews	brand
966	https://rukminim1.flixcart.com/image/312/312/x...	MSI Raider GE67 HX Core i9 12th Gen	419990	Intel Core i9 Processor (12th Gen)	32 GB DDR5 RAM	64 bit Windows 11 Operating System	2 TB SSD	15.6	0.0	0.0	0.0	MSI

```
data.query('price == price.min()')
```

	img_link	name	price	processor	ram	os	storage	display	rating	no_of_ratings	no_of_reviews	bran
74	https://rukminim1.flixcart.com/image/312/312/x...	ASUS Chromebook Flip Touch Celeron Dual Core	15990	Intel Celeron Dual Core Processor	4 GB LPDDR4 RAM	Chrome Operating System	2 TB SSD	11.6	4.0	1853.0	287.0	ASU

Lets observe the price range and number of products having that range using bins

```
plt.figure(figsize=(12,6))
sns.histplot(data, x = 'price',bins=40)
plt.title('Price Range of the Laptops')
plt.show()
```



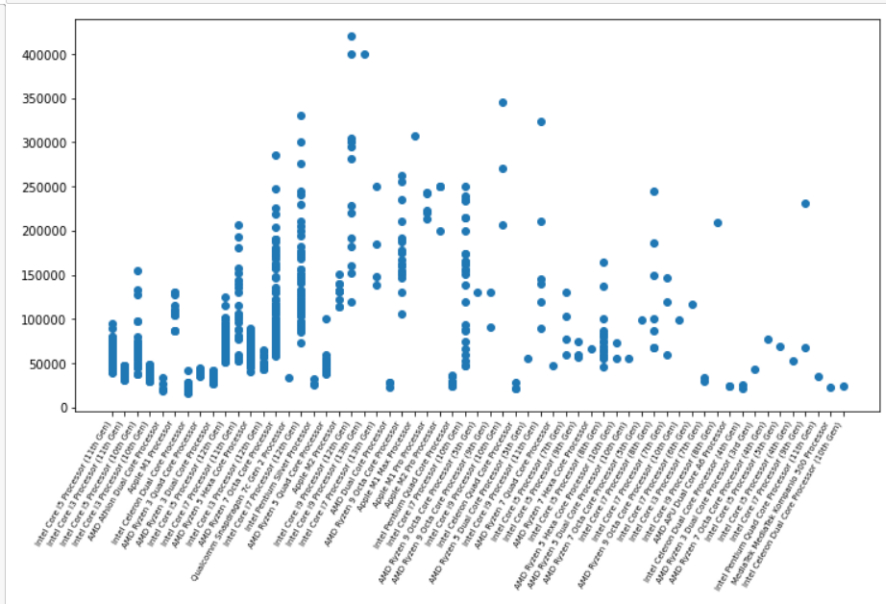
- We have laptops ranging over the price of Rs.15990-Rs.419990
- We can observe that most of the laptops lie in the price range of Rs.40000-Rs.70000

2.5.2 *Scatter plots*

Introducing scatter plot for better visualization with the histograms

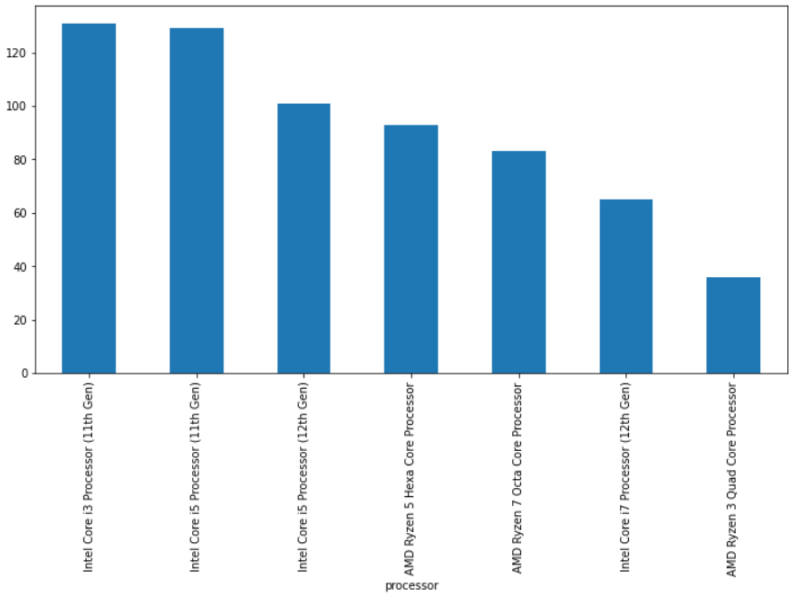
As we see processor while buying a laptop and price is also main concern lets check how they are varying

```
plt.figure(figsize=(12,6))
plt.scatter(data.processor, data['price'])
plt.xticks(rotation=60, horizontalalignment='right',
           fontsize=7);
```



- Intel Core i9 Processor(12th Gen) has the greatest price of Rs.419990
- We can't come in to any conclusion based on the processor and the price. They are not much related.

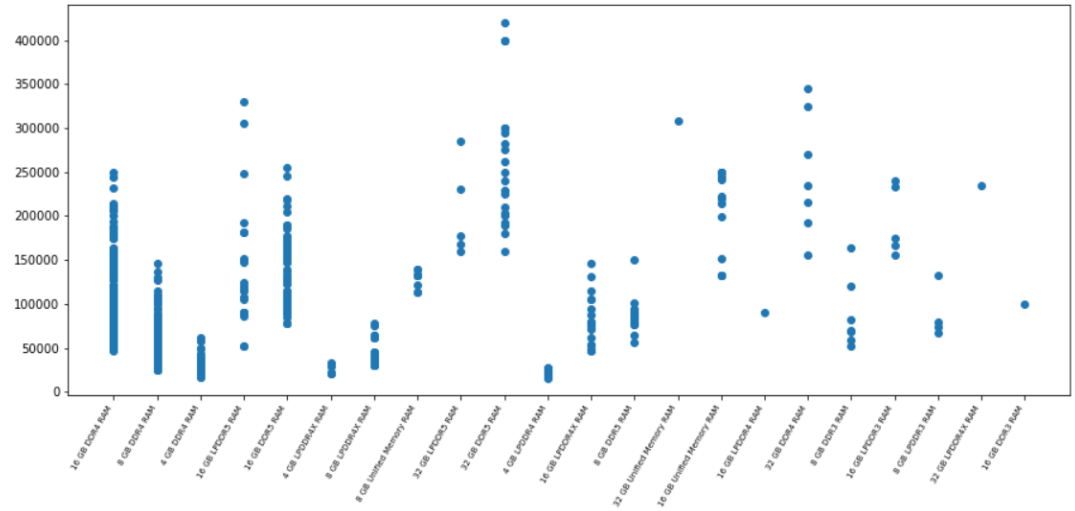
Let’s examine the frequency



- We examined that most of the laptops range over the price 40000-70000.
- Therefore there is high chance that these laptops have the processor among those mentioned above.

After processor we search for ram, So lets compare for price and ram

```
plt.figure(figsize=(15, 6))
plt.scatter(data['ram'],data['price'])
plt.xticks(rotation=60, horizontalalignment='right',
           fontsize=7);
```

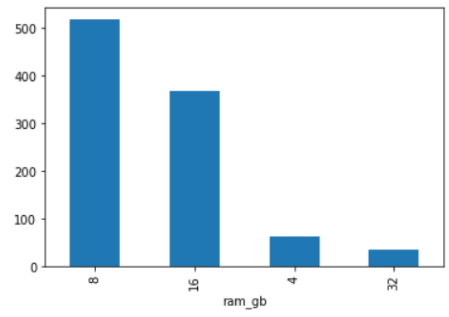


- We can't exactly find any relation between RAM and the price.
- But we can find there is only single laptop with a certain price for particular RAM.
- 16 GB DDR3 RAM, 32 GB LPDDR4X RAM, 16 GB LPDDR4 RAM, 32 GB Unified Memory RAM these are of particular prices and may be with a single piece laptop.

Let’s get into frequency

```
data.groupby('ram_gb').size().sort_values(ascending=False).plot(kind='bar')
```

```
<AxesSubplot:xlabel='ram_gb'>
```

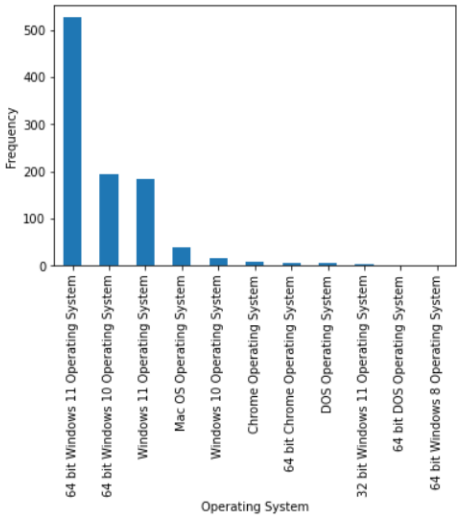


- We have laptops with ram ranging from 4 to 32 GB.
- Most of the laptops are of 8 GB and 16 GB RAM.
- As we examined that most of the laptops range over the price 40000-70000, there is high chance that these laptops have the ram as 8 and 16 GB.

Now checking for operating system

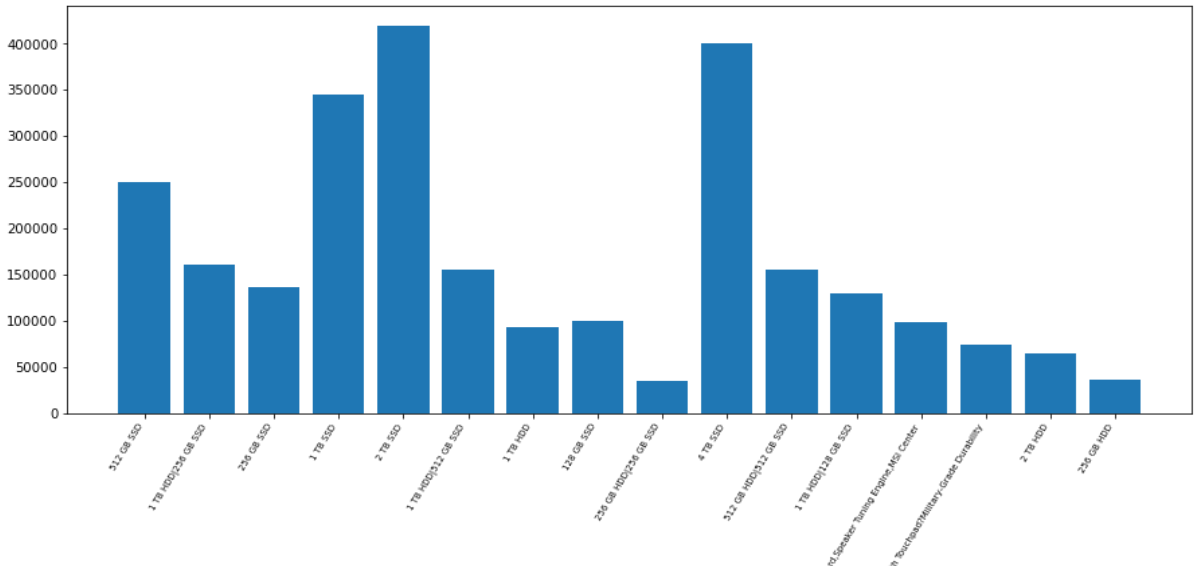
```
data.groupby('os').size().sort_values(ascending=False).plot(kind='bar');
plt.xlabel("Operating System")
plt.ylabel("Frequency")
```

```
Text(0, 0.5, 'Frequency')
```



- From this analysis, we can find the most probable Operating system we choose is 64 bit Windows 11 Operating System

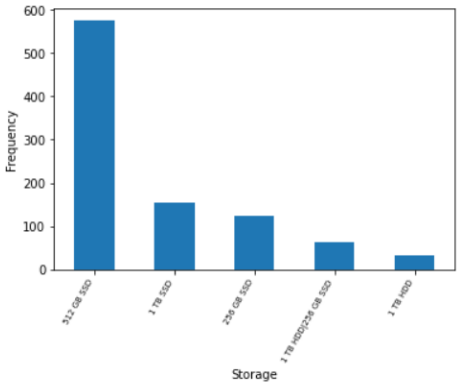
The most vital need is also storage, so lets have a look at price vs storage



- We can observe that, higher the storage, higher the price

Coming to frequency

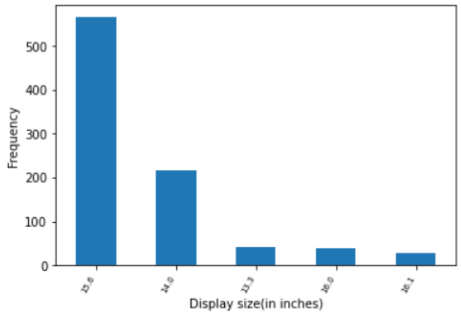
```
data.groupby('storage').size().sort_values(ascending=False).head(5).plot(kind='bar');
plt.xlabel("Storage")
plt.ylabel("Frequency")
plt.xticks(rotation=60, horizontalalignment='right',
           fontsize=7);
```



- We can see 512 GB SSD is more frequent when compared to even more storage laptops.
- People may not use very high storage like 1,2,4 TB SSD laptops which are are costlier until they require higher specifications and storage, so they prefer the laptops which can be of enough storage to them.
- Also, many of them don't prefer less stoarge laptops even though they are cheaper, because they can't come upto their expectations. They can be of very limited storage and may get get into the situation of out of storage.
- Hence we can say that 512 GB SSD(considering size and price) are more frequent due to this reason

Frequency of display(in inch)

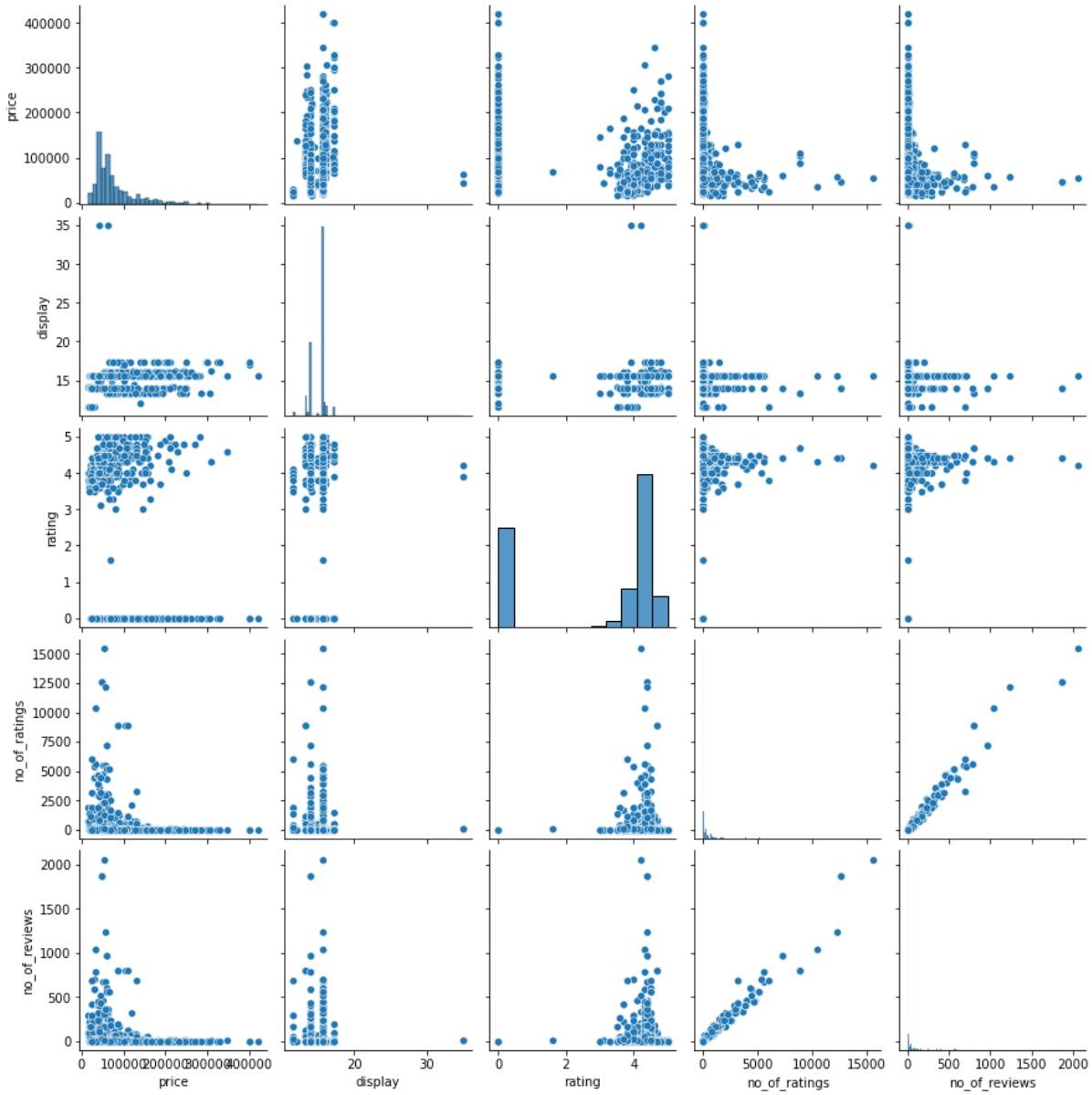
```
data.groupby('display').size().sort_values(ascending=False).head(5).plot(kind='bar');
plt.xlabel("Display size(in inches)")
plt.ylabel("Frequency")
plt.xticks(rotation=60, horizontalalignment='right',
           fontsize=7);
```



- We can see 15.6 Inch display is more frequent because it is more comfortable to use.
- People may need to work with 2 or more tabs at a time and it may not be comfortable with lesser display size to view them clearly.
- Also, greater than 16 inch size are less frequent. This may be due to the reason that it is hard to carry all the time easily from one place to another.

2.5.3 *Pair Plots*

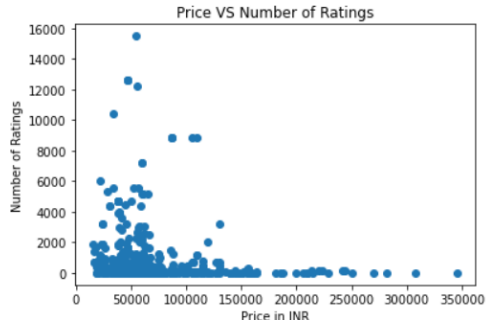
Pair Plots the best way to check on two features combination on overall dataset in a single go so lets check out them



Lets have a look at some more hist plots and scatter plots for more analysis

We can have a look at price vs rating to check price worthy or not

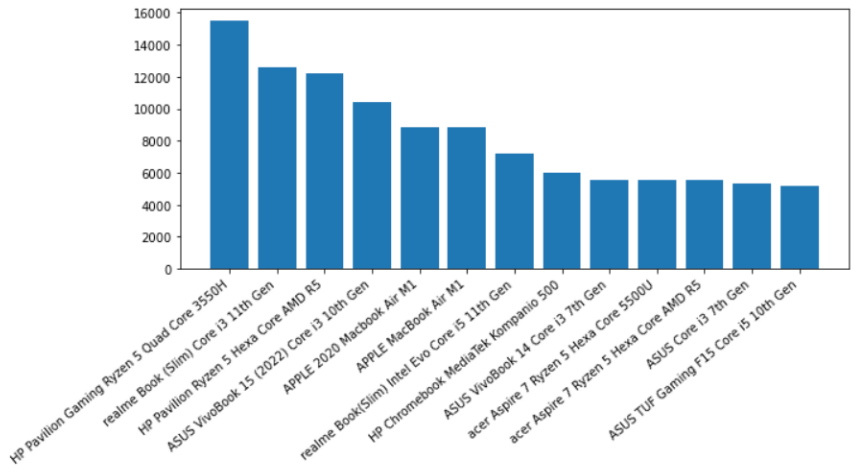
```
new_df = data[data['no_of_ratings'] != 0]
plt.scatter(new_df['price'],new_df['no_of_ratings'])
plt.xlabel('Price in INR')
plt.ylabel("Number of Ratings")
plt.title("Price VS Number of Ratings")
plt.show()
```



- When the price is between 25,000 to 1,00,000 there are more number of rating which means that this is the price range in which most people are interested upon

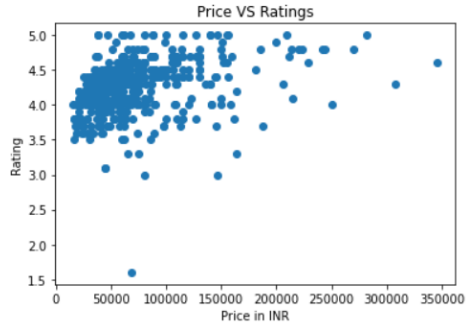
Let see what the most rated product

```
laptops_sorted_by_rating = data.sort_values(['no_of_ratings'], ascending=False)

plt.figure(figsize=(10,4))
plt.bar(laptops_sorted_by_rating['name'][:20], laptops_sorted_by_rating['no_of_ratings'][:20])
plt.xticks(rotation=40, horizontalalignment='right',
           fontsize=10)

print()
```

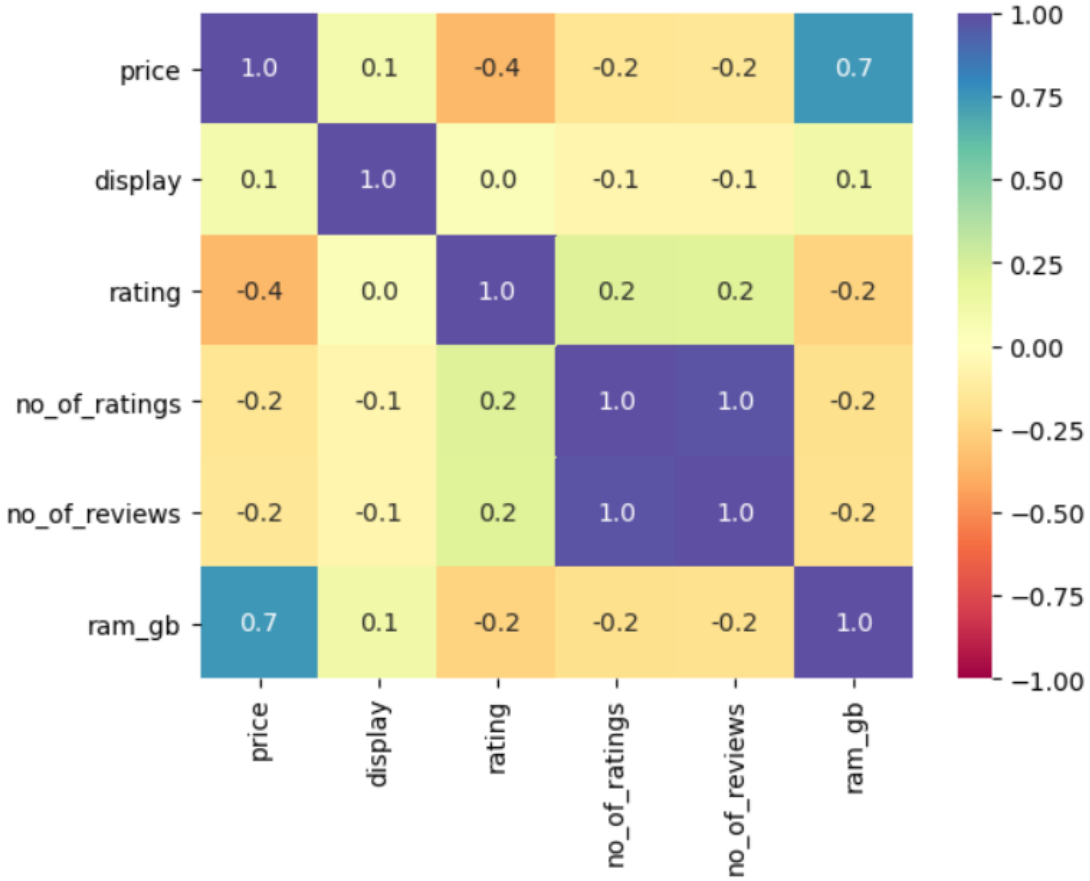


Ratings

```
new_df = data[data['rating'] != 0]
plt.scatter(new_df['price'],new_df['rating'])
plt.xlabel('Price in INR')
plt.ylabel("Rating")
plt.title("Price VS Ratings")
plt.show()
```



2.5.4 Correlation Heat Map



From the heatmap, we can observe a positive correlation between ram and the price of the laptop.

We can say that there is an increase in price with the increase in ram.

3. MODELING

3.1 Features

The following numerical features are taken into consideration. Some categorical features are transformed into numerical values into a pandas Data Frame. This is done by creating a new feature based on the uniqueness of the values in each categorical feature.

- processor
- ram
- os
- storage
- display
- rating
- no_of_ratings
- no_of_reviews

3.2 Target

- price

The target value is the price of the laptop based on the specifications mentioned.

3.3 Split to train & test

- train_test_split is imported from sklearn.model_selection
- The preprocessed data is split into training and testing of test size 0.3

3.4 Fit with Random Forest Regressor

Random Forest Regression is a machine learning algorithm used for regression problems, which involves predicting a continuous output variable. It is an ensemble method that combines multiple decision trees to create a more robust and accurate prediction model.

Steps involved are:

- Building the model

  In this model, multiple decision trees are created, and each tree is trained on a different subset of the training data. Each tree makes a prediction, and the final prediction is obtained by averaging the predictions from all the trees.
  It is imported from sklearn.ensemble

- Tuning the model

  Number of estimators are taken as 200.

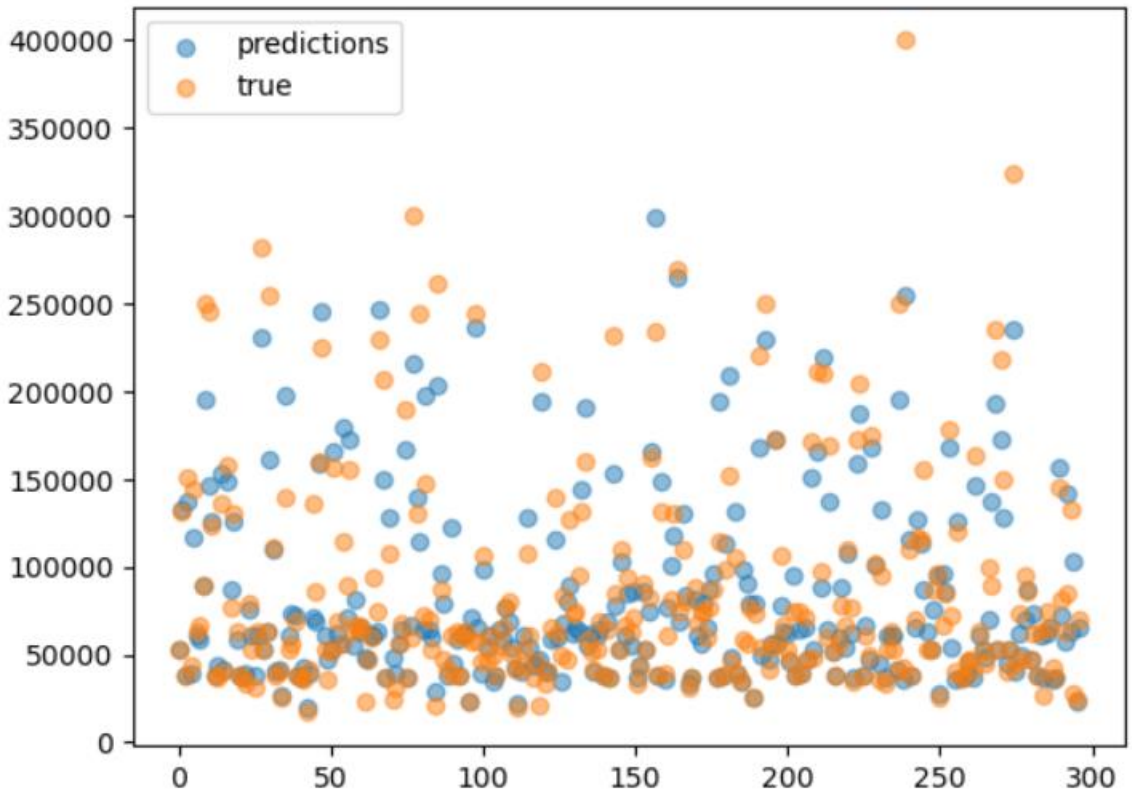
- Evaluating the model

  It is evaluated on the test set to see how well it generalizes to new data. The evaluation metrics include MSE, R-squared, and other regression metrics.

3.5 Test

- Testing is done on the preprocessed data set used n_estimators as 200 with Random Forest Regressor.

  The predictions versus the true values is plotted and shown below.



3.6 Evaluation metrics

3.6.1 *R2 score*

It is also known as the coefficient of determination and is commonly used for evaluating the performance of regression models. It measures the proportion of the variance in which the target variable that is explained by the independent variables in the model.

R2 score ranges from 0 to 1, where 0 indicates that the model does not explain any of the variability in the target variable, and 1 indicates that the model explains all the variability in the target variable. A higher R2 score (>0.75) indicates a better fit between the model and the data.

R2 score of our model:

```
r2 score:  0.8334682200143083
```

4. Conclusion

After performing EDA, proper data preprocessing and building model, we have got the model score as 0.833 which indicates the good fit of the model. Random Forest Regressor performs well when compared to other models and the R2 score indicates the good performance of the model.