RATINGS PREDICTION

We have a client who has a website where people write different reviews for technical products. Now they are adding a new feature to their website

The rating is out 5 stars and it only has 5 options available 1 star, 2 stars, 3 stars, 4 stars, 5 stars. Now they want to predict ratings for the reviews which were written in the past and they don't have a rating

o scrape the reviews of different laptops, Phones, Headphones, smart watches, Professional Cameras, Printers, Monitors, Home theater, Router from different ecommerce websites

we need these columns

1) reviews of the product.

2) rating of the product.

You can fetch other data as well, if you think data can be useful or can help in the project. It completely depends on your imagination or assumption

1. Data Cleaning

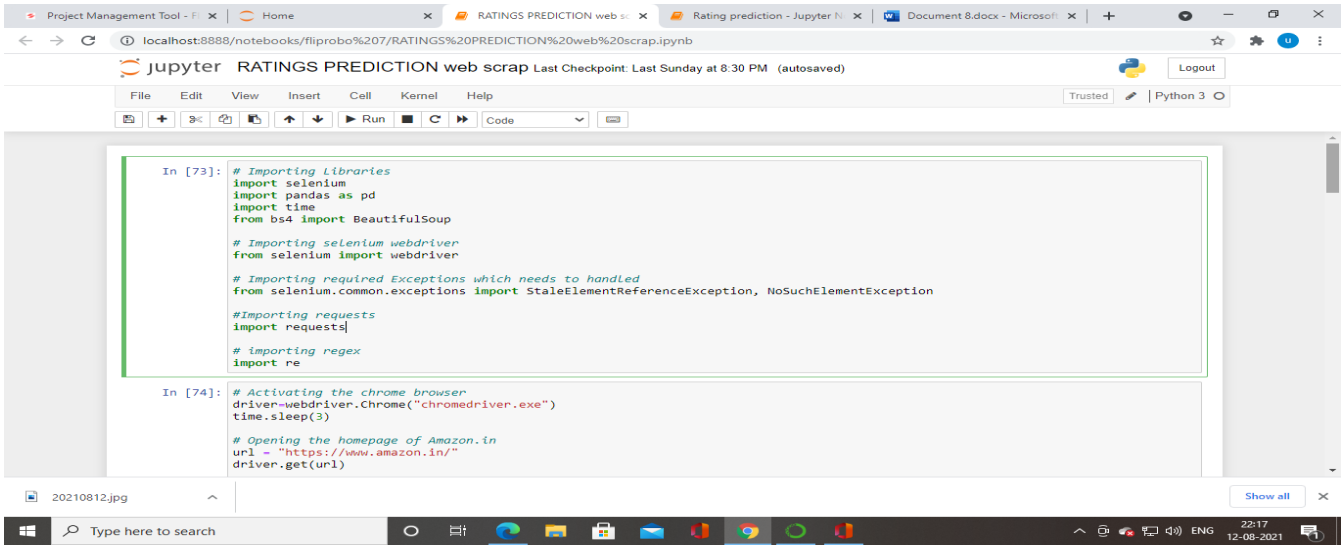
2. Exploratory Data Analysis

3. Data Preprocessing

4. Model Building

5. Model Evaluation

6. Selecting the best model

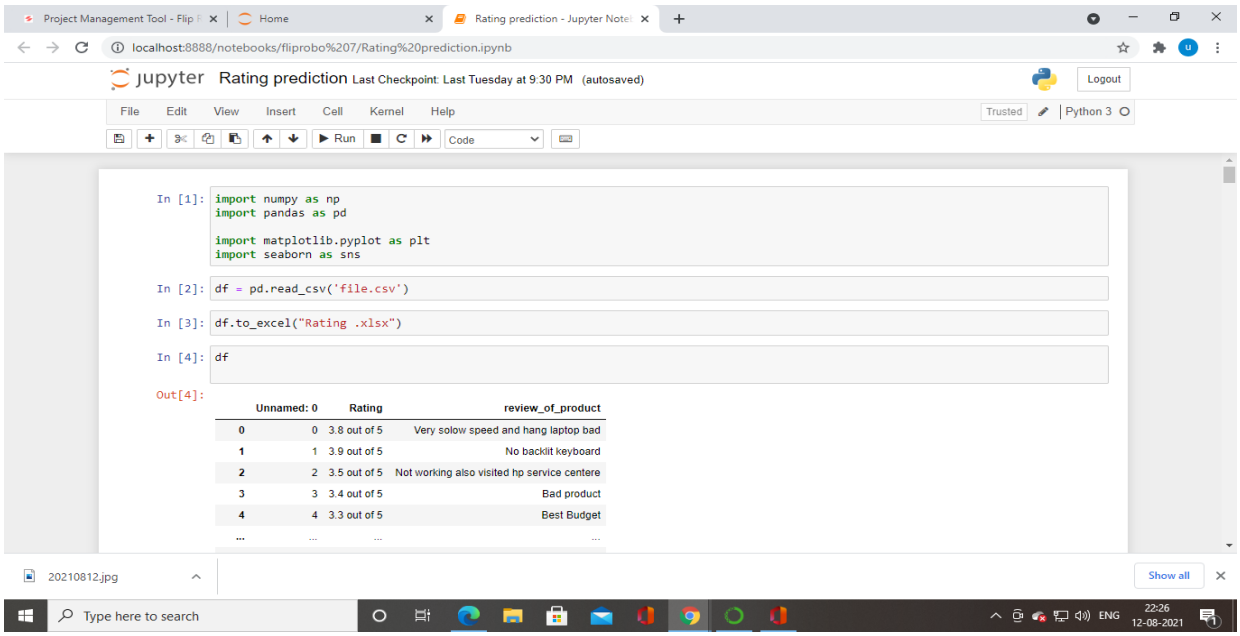


Selenium concept

From chrome browser we now activate and opening the homepage of Amazon.in

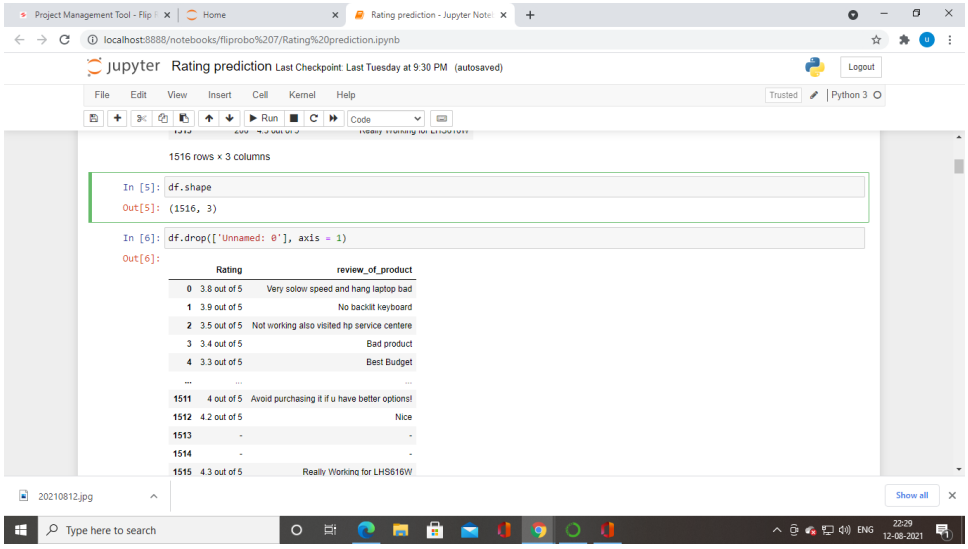
Asking the user input the keywords he/she wants to search

And next is scrape all the product urls and also extact data from the websites



By import numpy ,pandas matplotlib,seaborn we are using for the visual method

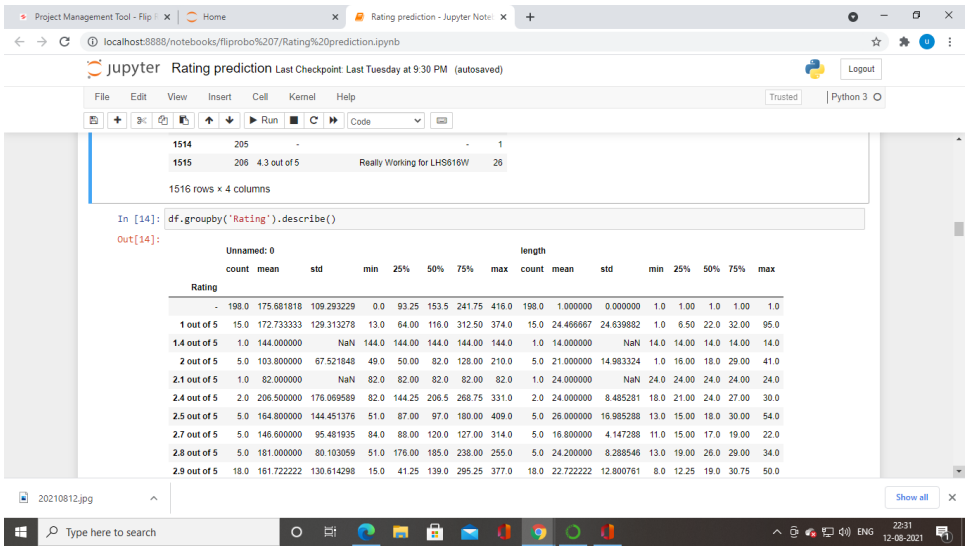
And then for reading the file



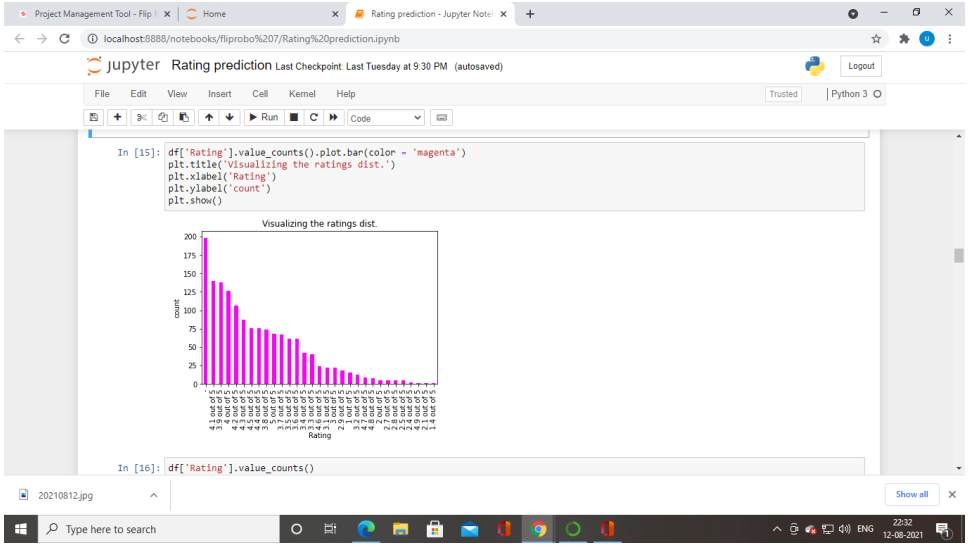
	Rating	review_of_product
0	3.8 out of 5	Very solow speed and hang laptop bad
1	3.9 out of 5	No backlit keyboard
2	3.5 out of 5	Not working also visited hp service centere
3	3.4 out of 5	Bad product
4	3.3 out of 5	Best Budget
...	...	...
1511	4 out of 5	Avoid purchasing it if u have better options!
1512	4.2 out of 5	Nice
1513	-	-
1514	-	-
1515	4.3 out of 5	Really Working for LHS616W

Df.shape which is used for size of the dataset

We used the drop the columns unnamed value



	Unnamed: 0								length							
Rating	count	mean	std	min	25%	50%	75%	max	count	mean	std	min	25%	50%	75%	max
-	198.0	175.681818	109.293229	0.0	93.25	153.5	241.75	416.0	198.0	1.000000	0.000000	1.0	1.00	1.0	1.00	1.0
1 out of 5	15.0	172.733333	129.313278	13.0	64.00	116.0	312.50	374.0	15.0	24.466667	24.639882	1.0	6.50	22.0	32.00	95.0
1.4 out of 5	1.0	144.000000	NaN	144.0	144.00	144.0	144.00	144.0	1.0	14.000000	NaN	14.0	14.00	14.0	14.00	14.0
2 out of 5	5.0	103.800000	67.521848	49.0	50.00	82.0	128.00	210.0	5.0	21.000000	14.983324	1.0	16.00	18.0	29.00	41.0
2.1 out of 5	1.0	82.000000	NaN	82.0	82.00	82.0	82.00	82.0	1.0	24.000000	NaN	24.0	24.00	24.0	24.00	24.0
2.4 out of 5	2.0	206.500000	176.069589	82.0	144.25	206.5	268.75	331.0	2.0	24.000000	8.485281	18.0	21.00	24.0	27.00	30.0
2.5 out of 5	5.0	164.800000	144.451376	51.0	87.00	97.0	180.00	409.0	5.0	26.000000	16.985288	13.0	15.00	18.0	30.00	54.0
2.7 out of 5	5.0	146.600000	95.481935	84.0	88.00	120.0	127.00	314.0	5.0	16.800000	4.147288	11.0	15.00	17.0	19.00	22.0
2.8 out of 5	5.0	181.000000	80.103059	51.0	176.00	185.0	238.00	255.0	5.0	24.200000	8.288546	13.0	19.00	26.0	29.00	34.0
2.9 out of 5	18.0	161.722222	130.614298	15.0	41.25	139.0	295.25	377.0	18.0	22.722222	12.800761	8.0	12.25	19.0	30.75	50.0

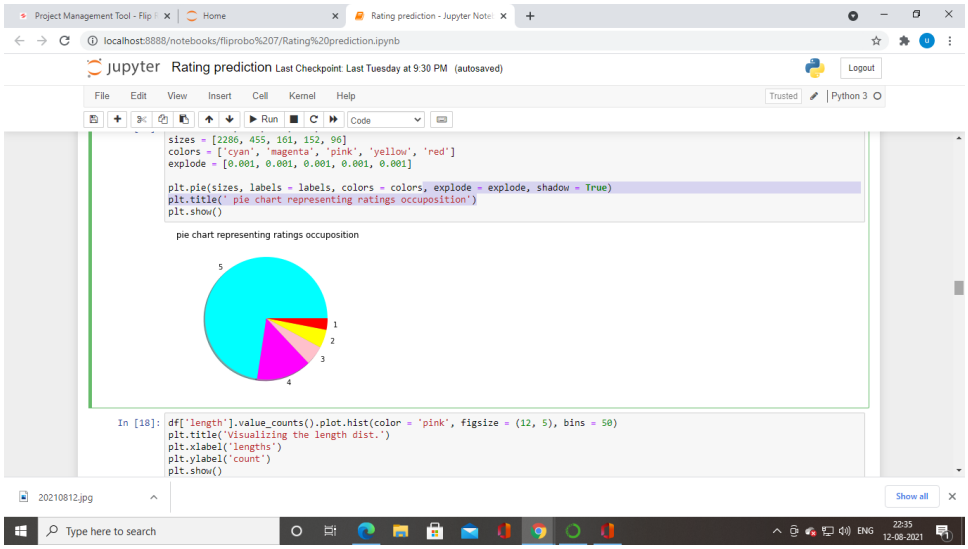


The value of count is high

df['Rating'].value_counts()

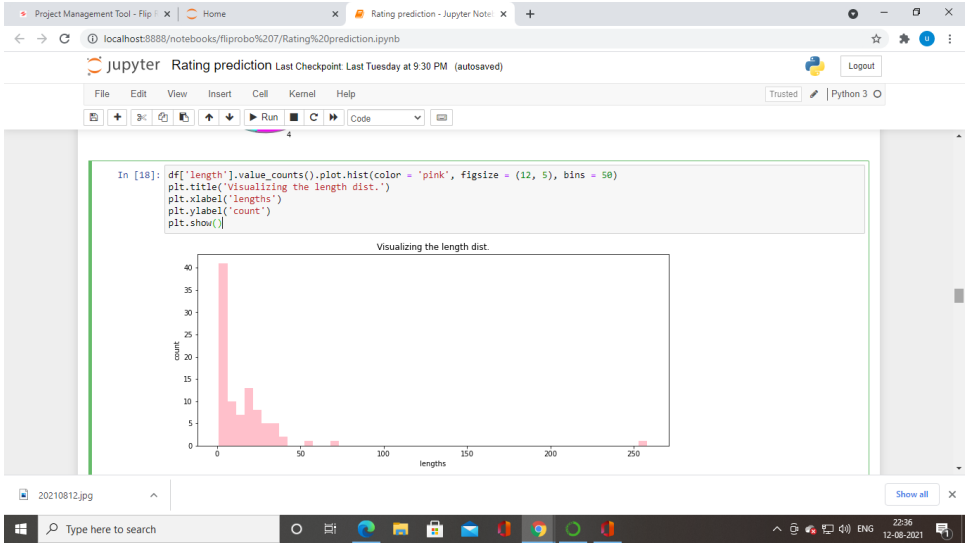
```
-                 198
4.1 out of 5    140
3.9 out of 5     138
4 out of 5       126


4.9 out of 5
      12.1 out of 5
      11.4 out of 5
```

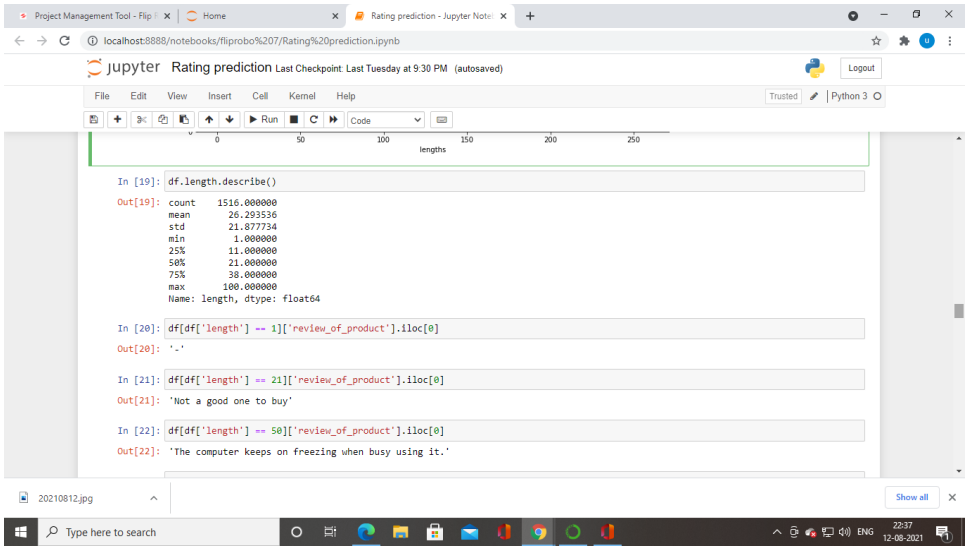


5 point which has maxium

1 point which has minium



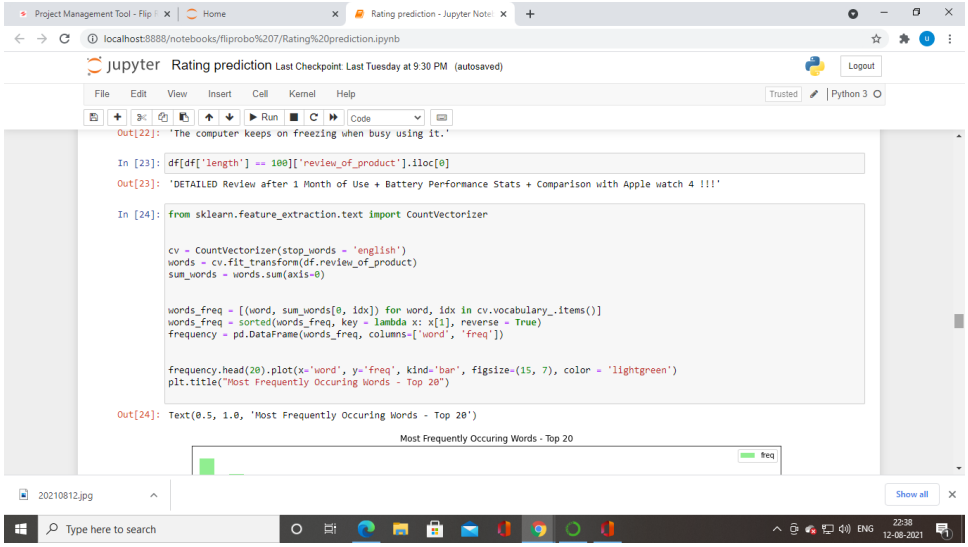
The length of the value which show above

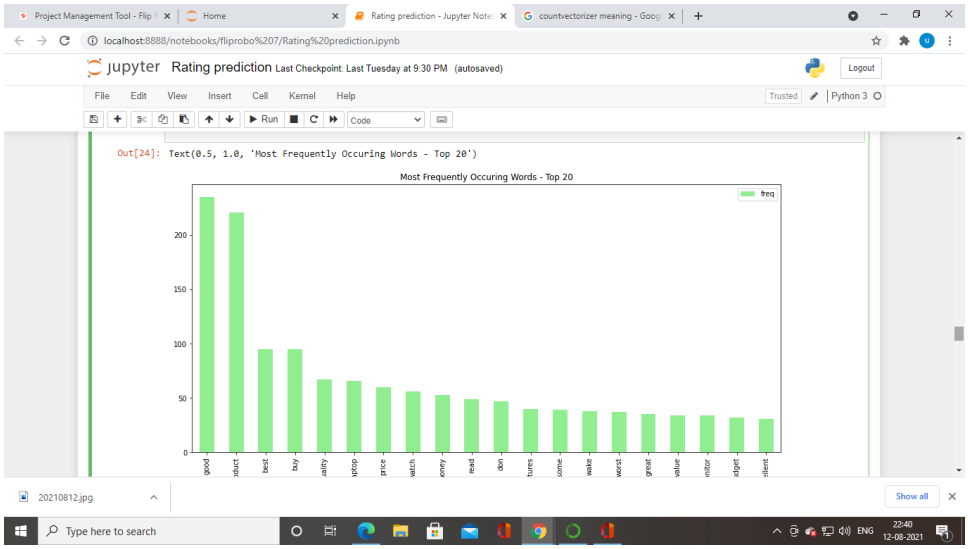


The describe which show mean,standard deviation

CountVectorizer

is a great tool provided by the scikit-learn library in Python. It is used to transform a given text into a vector on the basis of the frequency (count) of each word that occurs in the entire text.



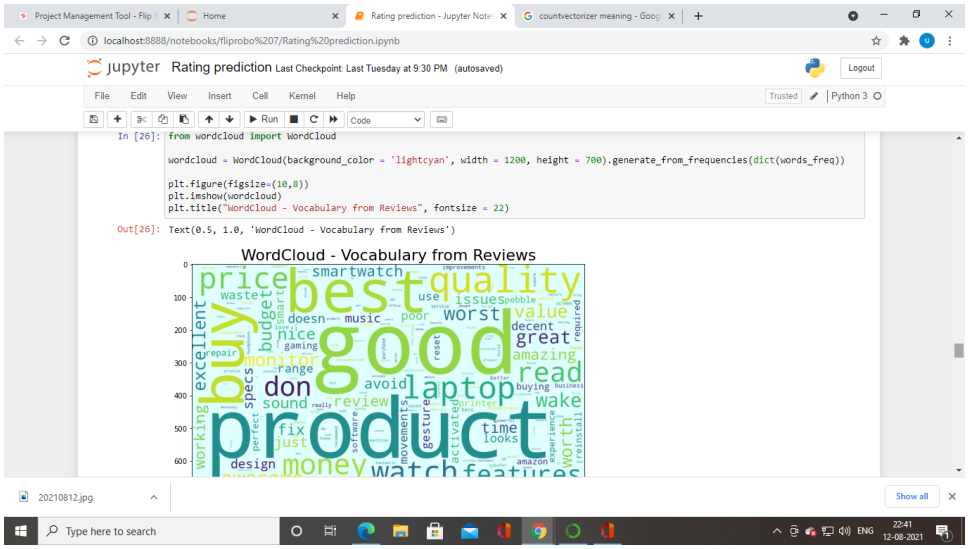


Good

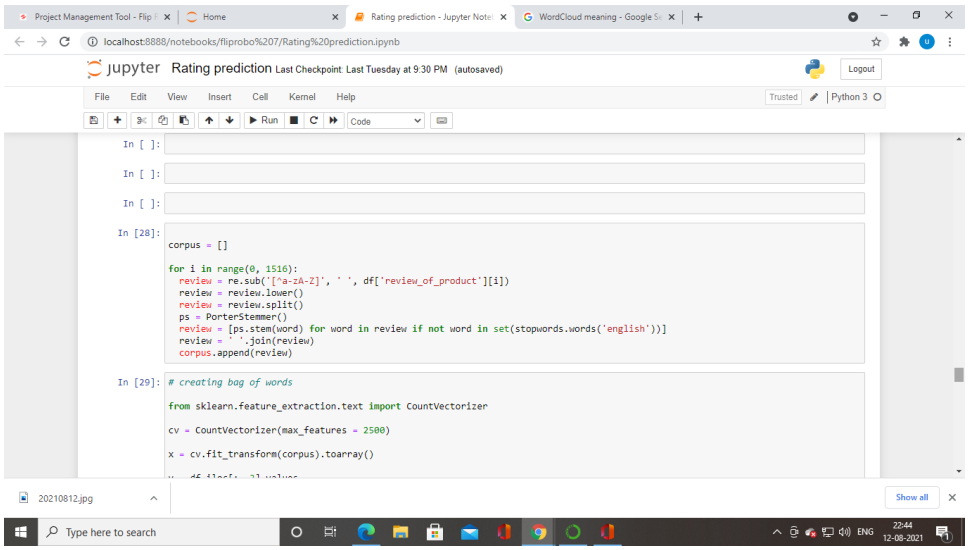
Product are used commonly

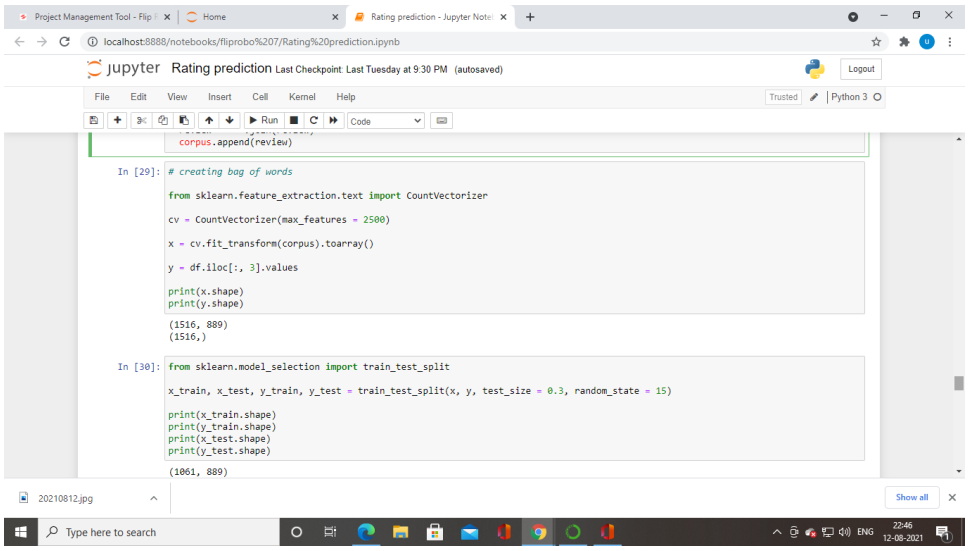
Word clouds or tag clouds

are graphical representations of word frequency that give greater prominence to words that appear more frequently in a source text. ... Most word cloud generators have features that allow users to change colors, font, and exclude common or similar



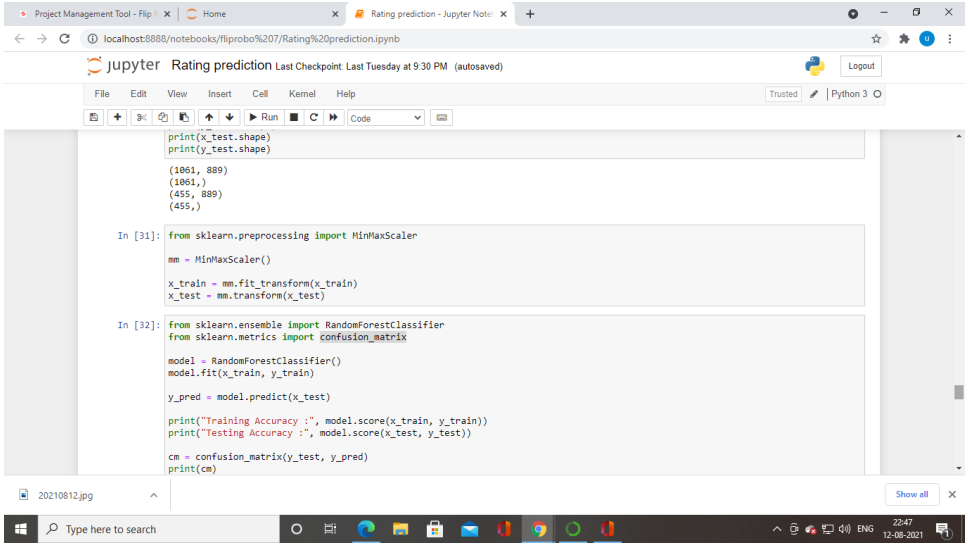
A collection of written texts, especially the entire works of a particular author or a body of writing on a particular subject.





Transform features by scaling each feature to a given range.

This estimator scales and translates each feature individually such that it is in the given range on the training set, e.g. between zero and one.



Training Accuracy : 0.942507068803016

Testing Accuracy : 0.4835164835164835

[[76 0 0 ... 0 0 0]

[1 0 0 ... 0 0 0]

[5 0 2 ... 0 0 0]

...

[0 0 0 ... 0 0 0]

[0 0 0 ... 0 0 0]

[0 0 0 ... 0 0 0]]

cross_val_score

Accuracy : 0.48636924704637624

Standard Variance : 0.031545429501507696

Mean Cross Validation Accuracy - Train Set : 87.85437320115126

Mean Cross Validation Accuracy - Validation Set : 40.599438581530045

Accuracy Score for Test Set : 0.4835164835164835

DecisionTreeClassifier

Training Accuracy : 0.943449575871819

Testing Accuracy : 0.4945054945054945

```
[[76  0  0 ...  0  0  0]
 [ 1  0  0 ...  0  0  0]
 [ 5  0  2 ...  0  0  0]
 ...
 [ 0  0  0 ...  0  0  0]
 [ 0  0  0 ...  0  0  0]
 [ 0  0  0 ...  0  0  0]]
```

Random forest is the best model